



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、仕様の開発元企業として「スパンション」, 「Spansion」, 「富士通」または「Fujitsu」の名が記載されておりますが、これらの製品は Cypress が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。

#### 商品仕様の継続性について

Cypress 製品として提供することに伴う商品仕様としての変更はなく、ドキュメントとしての変更もありません。また本ページのお知らせは、変更情報として追記いたしません。本ドキュメントに変更情報が記載されている場合、それは本お知らせを除いた前版からの変更点です。なお、今後改訂は必要に応じて行われますが、その際の変更内容は改訂後のドキュメントに記載いたします。

#### オーダ型格および品名について

Cypress は既存のオーダ型格および品名を引き続きサポートいたします。これらの製品をご注文の際は、このドキュメントに記載されているオーダ型格および品名をご使用ください。

#### 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 (銘柄コード: CY) は、車載や産業機器、ネットワーキング プラットフォームから高機能民生機器およびモバイル機器まで、今日の最先端組み込みシステム向けに高性能で高品質のソリューションを提供します。NOR フラッシュ メモリや F-RAM<sup>TM</sup>、SRAM、Traveo<sup>TM</sup> マイクロコントローラー、業界唯一の PSoC<sup>®</sup> プログラマブル システムオンチップ ソリューション、アナログおよび PMIC Power Management IC、CapSense<sup>®</sup> 静電容量タッチセンシング コントローラー、Wireless BLE Bluetooth<sup>®</sup> Low-Energy、USB コネクティビティ ソリューションなど、幅広い差別化製品ポートフォリオを、一貫した革新性と業界最高クラスの技術サポート、比類のないシステム バリューとともにグローバルに提供します。

# F<sup>2</sup>MC-16LX

16 ビット・マイクロコントローラ

## MB90460/465 シリーズ ハードウェアマニュアル



# F<sup>2</sup>MC-16LX

## 16 ビット・マイクロコントローラ

# MB90460/465 シリーズ

# ハードウェアマニュアル

富士通マイクロエレクトロニクス社のマイコンを効率的に開発するための情報を下記 URL にてご紹介いたします。  
ご採用を検討中、またはご採用いただいたお客様に有益な情報を公開しています。

**開発における最新の注意事項に関しては、必ず「Check Sheet」を参照してください。**  
「Check Sheet」はシステム開発において、問題を未然に防ぐことを目的として、最低限必要と思われるチェック項目をリストにしたものです。

<http://edevic.e.fujitsu.com/micom/jp-support/>

富士通マイクロエレクトロニクス株式会社





# はじめに

## ■ 本書の目的と対象読者

富士通マイクロエレクトロニクス製品につきまして，平素より格別のご愛読を賜り厚くお礼申し上げます。

MB90460/465 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 F<sup>2</sup>MC-16LX 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，MB90460/465 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。

## ■ 商標

F<sup>2</sup>MC は，FUJITSU Flexible Microcontroller の略で富士通マイクロエレクトロニクス株式会社の商標です。

その他の記載されている社名および製品名などの固有名詞は，各社の商標または登録商標です。

## ■ 本書の全体構成

本書は、以下に示す 24 の章および付録から構成されています。

### 第 1 章 概要

MB90460/465 シリーズの主な特長と基本仕様について説明します。

### 第 2 章 デバイスの取扱いについて

デバイスの取扱い上の注意について説明します。

### 第 3 章 CPU

MB90460/465 シリーズの機能と動作について説明します。

### 第 4 章 リセット

MB90460/465 シリーズのリセットについて説明します。

### 第 5 章 クロック

MB90460/465 シリーズのクロックについて説明します。

### 第 6 章 低消費電力モード

MB90460/465 シリーズ低消費電力モードについて説明します。

### 第 7 章 割込み

MB90460/465 シリーズの割込みと拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) について説明します。

### 第 8 章 モード設定

MB90460/465 シリーズの動作モードについて説明します。

### 第 9 章 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。

### 第 10 章 タイムベースタイマ

タイムベースタイマの機能と動作について説明します。

### 第 11 章 ウォッチドッグタイマ

ウォッチドッグタイマの機能と動作について説明します。

### 第 12 章 16 ビットリロードタイマ

16 ビットリロードタイマの機能と動作について説明します。

### 第 13 章 16 ビット PPG タイマ

16 ビット PPG タイマの機能と動作について説明します (MB90467 では、PPG1 タイマ ch.1 未搭載となります)。

### 第 14 章 多機能タイマ

多機能タイマの機能と動作について説明します。

### 第 15 章 マルチパルスジェネレータ

マルチパルスジェネレータの機能と動作について説明します (MB90467 では、未搭載となります)。

### 第 16 章 PWC タイマ

PWC タイマの機能と動作について説明します (MB90467 では、PWC タイマ ch.0 未搭載となります)。

## 第 17 章 UART

UART の機能と動作について説明します。

## 第 18 章 DTP/ 外部割込み

DTP/ 外部割込みの機能と動作について説明します。

## 第 19 章 遅延割込み発生モジュール

遅延割込み発生モジュールの機能と動作について説明します。

## 第 20 章 8/10 ビット A/D コンバータ

8/10 ビット A/D コンバータの機能と動作について説明します。

## 第 21 章 アドレス一致検出機能

アドレス一致検出機能の機能と動作について説明します。

## 第 22 章 ROM 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の機能と動作について説明します。

## 第 23 章 512K/1024K ビット フラッシュメモリ

512K/1024K ビットフラッシュメモリの機能と動作について説明します。

## 第 24 章 フラッシュシリアル 書込み接続例

横河ディジタルコンピュータ株式会社製フラッシュマイコンプログラムを用いた場合の、シリアル書込み接続例について説明します。

## 付録

付録では、I/O マップ、命令などを記載します。

- 本資料の記載内容は、予告なしに変更することがありますので、ご用命の際は営業部門にご確認ください。
- 本資料に記載された動作概要や応用回路例は、半導体デバイスの標準的な動作や使い方を示したもので、実際に使用する機器での動作を保証するものではありません。したがって、これらを使用するにあたってはお客様の責任において機器の設計を行ってください。これらの使用に起因する損害などについては、当社はその責任を負いません。
- 本資料に記載された動作概要・回路図を含む技術情報は、当社もしくは第三者の特許権、著作権等の知的財産権やその他の権利の使用権または実施権の許諾を意味するものではありません。また、これらの使用について、第三者の知的財産権やその他の権利の実施ができることの保証を行うものではありません。したがって、これらの使用に起因する第三者の知的財産権やその他の権利の侵害について、当社はその責任を負いません。
- 本資料に記載された製品は、通常の産業用、一般事務用、パーソナル用、家庭用などの一般的用途に使用されることを意図して設計・製造されています。極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、社会的に重大な影響を与えかつ直接生命・身体に対する重大な危険性を伴う用途（原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療機器、兵器システムにおけるミサイル発射制御をいう）、ならびに極めて高い信頼性が要求される用途（海底中継器、宇宙衛星をいう）に使用されるよう設計・製造されたものではありません。したがって、これらの用途にご使用をお考えのお客様は、必ず事前に営業部門までご相談ください。ご相談なく使用されたことにより発生した損害などについては、責任を負いかねますのでご了承ください。
- 半導体デバイスはある確率で故障が発生します。当社半導体デバイスが故障しても、結果的に人身事故、火災事故、社会的な損害を生じさせないよう、お客様は、装置の冗長設計、延焼対策設計、過電流防止対策設計、誤動作防止設計などの安全設計をお願いします。
- 本資料に記載された製品を輸出または提供する場合は、外国為替及び外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

# 目次

<b>第 1 章</b>	<b>概要</b>	<b>1</b>
1.1	MB90460/465 シリーズの特長	2
1.2	MB90460/465 シリーズの製品一覧	5
1.3	MB90460/465 シリーズのブロックダイアグラム	7
1.4	端子配列図	8
1.5	外形寸法図	11
1.6	I/O 端子と端子機能説明	14
1.7	I/O 回路の種類	19
<b>第 2 章</b>	<b>デバイスの取扱いについて</b>	<b>23</b>
2.1	デバイス取扱い上の注意事項	24
<b>第 3 章</b>	<b>CPU</b>	<b>27</b>
3.1	CPU の概要	28
3.2	メモリ空間	29
3.3	メモリマップ	31
3.4	アドレス指定	33
3.4.1	リニア方式によるアドレス指定	34
3.4.2	バンク方式によるアドレス指定	35
3.5	多バイト長データのメモリ配置	37
3.6	レジスタ	39
3.7	専用レジスタ	40
3.7.1	アキュムレータ (A)	42
3.7.2	スタックポインタ (USP, SSP)	45
3.7.3	プロセッサステータス (PS)	47
3.7.4	コンディションコードレジスタ (PS:CCR)	48
3.7.5	レジスタバンクポインタ (PS:RP)	50
3.7.6	割込みレベルマスクレジスタ (PS:ILM)	51
3.7.7	プログラムカウンタ (PC)	52
3.7.8	ダイレクトページレジスタ (DPR)	53
3.7.9	バンクレジスタ (PCB, DTB, USB, SSB, ADB)	54
3.8	汎用レジスタ	55
3.9	プリフィックスコード	57
3.9.1	バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)	58
3.9.2	コモンレジスタバンクプリフィックス (CMR)	60
3.9.3	フラグ変化抑止プリフィックス (NCC)	61
3.9.4	プリフィックスコードに関する制約	62
3.9.5	「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意	64
<b>第 4 章</b>	<b>リセット</b>	<b>67</b>
4.1	リセットの概要	68
4.2	リセット要因と発振安定待ち時間	70
4.3	外部リセット端子	71
4.4	リセット動作	72

4.5	リセット要因フラグビット.....	74
4.6	リセットによる各端子の状態.....	77
<b>第5章</b>	<b>クロック.....</b>	<b>79</b>
5.1	クロックの概要 .....	80
5.2	クロック発生部のブロックダイアグラム.....	82
5.3	クロック選択レジスタ (CKSCR).....	84
5.4	クロックモード .....	86
5.5	発振安定待ち時間.....	88
5.6	振動子と外部クロックの接続.....	89
<b>第6章</b>	<b>低消費電力モード .....</b>	<b>91</b>
6.1	低消費電力モードの概要 .....	92
6.2	低消費電力制御回路のブロックダイアグラム.....	95
6.3	低消費電力モード制御レジスタ (LPMCR).....	97
6.4	CPU 間欠動作モード .....	100
6.5	スタンバイモード.....	101
6.5.1	スリープモード .....	102
6.5.2	タイムベースタイマモード .....	105
6.5.3	ストップモード .....	107
6.6	状態遷移図.....	109
6.7	スタンバイモード, リセットの端子状態.....	110
6.8	低消費電力モード使用上の注意 .....	111
<b>第7章</b>	<b>割込み .....</b>	<b>113</b>
7.1	割込み.....	114
7.2	割込み要因と割込みベクタ.....	116
7.3	割込み制御レジスタとリソース .....	119
7.3.1	割込み制御レジスタ (ICR00 ~ ICR15).....	120
7.3.2	割込み制御レジスタの機能 .....	122
7.4	ハードウェア割込み .....	125
7.4.1	ハードウェア割込み動作.....	128
7.4.2	割込み処理 .....	130
7.4.3	ハードウェア割込み使用手順.....	131
7.4.4	複数の割込み.....	132
7.4.5	ハードウェア割込みの処理に要する時間.....	134
7.5	ソフトウェア割込み .....	136
7.6	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) からの割込み.....	138
7.6.1	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) のディスクリプタ (ISD).....	140
7.6.2	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) のディスクリプタ (ISD) の レジスタ構成 .....	141
7.6.3	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の動作 .....	144
7.6.4	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の使用手順 .....	145
7.6.5	拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の処理時間 .....	146
7.7	例外処理割込み .....	148
7.8	割込み処理時のスタック動作.....	149
7.9	割込み処理のプログラム例.....	151

<b>第 8 章</b>	<b>モード設定 .....</b>	<b>155</b>
8.1	モード設定 .....	156
8.2	モード端子 ( MD2 ~ MD0 ) .....	157
8.3	モードデータレジスタ .....	158
<b>第 9 章</b>	<b>I/O ポート .....</b>	<b>161</b>
9.1	I/O ポートの概要 .....	162
9.2	I/O ポートのレジスタ .....	164
9.3	ポート 0 .....	165
9.3.1	ポート 0 のレジスタ ( PDR0, DDR0, RDR0 ) .....	167
9.3.2	ポート 0 の動作 .....	169
9.4	ポート 1 .....	171
9.4.1	ポート 1 のレジスタ ( PDR1, DDR1, RDR1 ) .....	173
9.4.2	ポート 1 の動作 .....	175
9.5	ポート 2 .....	177
9.5.1	ポート 2 のレジスタ ( PDR2, DDR2 ) .....	179
9.5.2	ポート 2 の動作 .....	180
9.6	ポート 3 .....	182
9.6.1	ポート 3 のレジスタ ( PDR3, DDR3 ) .....	184
9.6.2	ポート 3 の動作 .....	185
9.7	ポート 4 .....	187
9.7.1	ポート 4 のレジスタ ( PDR4, DDR4 ) .....	189
9.7.2	ポート 4 の動作 .....	190
9.8	ポート 5 .....	192
9.8.1	ポート 5 のレジスタ ( PDR5, DDR5, ADER ) .....	194
9.8.2	ポート 5 の動作 .....	196
9.9	ポート 6 .....	198
9.9.1	ポート 6 のレジスタ ( PDR6, DDR6 ) .....	200
9.9.2	ポート 6 の動作 .....	201
9.10	I/O ポートのプログラム例 .....	203
<b>第 10 章</b>	<b>タイムベースタイマ .....</b>	<b>205</b>
10.1	タイムベースタイマの概要 .....	206
10.2	タイムベースタイマの構成 .....	208
10.3	タイムベースタイマ制御レジスタ ( TBTC ) .....	209
10.4	タイムベースタイマ割込み .....	211
10.5	タイムベースタイマの動作 .....	212
10.6	タイムベースタイマ使用上の注意 .....	215
10.7	タイムベースタイマのプログラム例 .....	216
<b>第 11 章</b>	<b>ウォッチドッグタイマ .....</b>	<b>217</b>
11.1	ウォッチドッグタイマの概要 .....	218
11.2	ウォッチドッグタイマの構成 .....	219
11.3	ウォッチドッグタイマ制御レジスタ ( WDTC ) .....	220
11.4	ウォッチドッグタイマの動作 .....	222
11.5	ウォッチドッグタイマ使用上の注意 .....	224
11.6	ウォッチドッグタイマのプログラム例 .....	225



<b>第 12 章</b>	<b>16 ビットリロードタイマ</b>	<b>227</b>
12.1	16 ビットリロードタイマの概要	228
12.2	16 ビットリロードタイマのブロックダイアグラム	231
12.3	16 ビットリロードタイマの端子	233
12.4	16 ビットリロードタイマのレジスタ	234
12.4.1	タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1)	235
12.4.2	タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1)	237
12.4.3	16 ビットタイマレジスタ上位, 下位 (TMRH0, TMRH1, TMRL0, TMRL1)	239
12.4.4	16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1)	240
12.5	16 ビットリロードタイマの割込み	241
12.6	16 ビットリロードタイマの動作	242
12.6.1	内部クロックモード (リロードモード)	244
12.6.2	内部クロックモード (ワンショットモード)	247
12.6.3	イベントカウントモード	250
12.7	16 ビットリロードタイマ使用上の注意	252
12.8	16 ビットリロードタイマのプログラム例	253
<b>第 13 章</b>	<b>16 ビット PPG タイマ</b>	<b>257</b>
13.1	16 ビット PPG タイマの概要	258
13.2	16 ビット PPG タイマのブロックダイアグラム	259
13.3	16 ビット PPG タイマの端子	260
13.4	16 ビット PPG タイマのレジスタ	262
13.4.1	PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2)	264
13.4.2	PPG 周期設定バッファレジスタ (PCSRH, PCSRL0 ~ PCSRL2)	265
13.4.3	PPG デューティ設定バッファレジスタ (PDUTH, PDUTL0 ~ PDUTL2)	266
13.4.4	PPG 状態制御レジスタ (PCNTH, PCNTL0 ~ PCNTL2)	267
13.5	16 ビット PPG タイマ割込み	271
13.6	16 ビット PPG タイマの動作	273
13.7	16 ビット PPG タイマ使用上の注意	277
13.8	16 ビット PPG タイマのプログラム例	278
<b>第 14 章</b>	<b>多機能タイマ</b>	<b>281</b>
14.1	多機能タイマの概要	282
14.2	多機能タイマのブロックダイアグラム	284
14.3	多機能タイマの端子	288
14.4	多機能タイマのレジスタ	290
14.4.1	コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL) / コンペアクリアレジスタ (CPCLRH, CPCLRL)	294
14.4.2	タイマデータレジスタ (TCDTH, TCDTL)	295
14.4.3	タイマ状態制御レジスタ (TCCSH, TCCSL)	296
14.4.4	アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL0 ~ OCCPBL5) / アウトプットコンペアレジスタ (OCCPH, OCCPL0 ~ OCCPL5)	302
14.4.5	コンペア制御レジスタ (OCSH, OCSL0 ~ OCSL5)	304
14.4.6	インプットキャプチャデータレジスタ (IPCPH, IPCPL0 ~ IPCPL3)	310
14.4.7	インプットキャプチャ状態制御 / PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01)	311
14.4.8	16 ビットデッドタイマレジスタ (TMRRH, TMRRL0 ~ TMRRL2)	318
14.4.9	16 ビットデッドタイマ制御レジスタ (DTCR0 ~ DTCR2)	319
14.4.10	波形制御レジスタ (SIGCR)	328

14.5	多機能タイマ割込み .....	330
14.6	多機能タイマの動作 .....	334
14.6.1	16 ビットフリーランタイマの動作 .....	335
14.6.2	16 ビットアウトプットコンペアの動作 .....	341
14.6.3	16 ビットインプットキャプチャの動作 .....	346
14.6.4	波形ジェネレータの動作 .....	348
14.7	多機能タイマ使用上の注意 .....	360
14.8	多機能タイマのプログラム例 .....	363
<b>第 15 章</b>	<b>マルチパルスジェネレータ .....</b>	<b>367</b>
15.1	マルチパルスジェネレータの概要 .....	368
15.2	マルチパルスジェネレータのブロックダイアグラム .....	372
15.3	マルチパルスジェネレータの端子 .....	379
15.4	マルチパルスジェネレータのレジスタ .....	381
15.4.1	出力制御レジスタ上位, 下位 (OPCUR, OPCLR) .....	384
15.4.2	出力データレジスタ上位, 下位 (OPDRH, OPDRL) .....	388
15.4.3	出力データバッファレジスタ (OPDBR0 ~ OPDBRB) .....	391
15.4.4	入力制御レジスタ上位, 下位 (IPCUR, IPCLR) .....	394
15.4.5	コンペアクリアレジスタ上位, 下位 (CPCRH, CPCRL) .....	398
15.4.6	タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL) .....	399
15.4.7	ノイズキャンセル制御レジスタ (NCCR) .....	400
15.4.8	タイマ状態制御レジスタ (TCSR) .....	402
15.5	マルチパルスジェネレータの割込み .....	404
15.6	マルチパルスジェネレータの動作 .....	407
15.6.1	位置検出の動作 .....	409
15.6.2	データ書込み制御ユニットの動作 .....	411
15.6.3	出力データバッファレジスタの動作 .....	415
15.6.4	出力データレジスタへのデータ転送動作 .....	417
15.6.5	DTTI 入力制御の動作 .....	431
15.6.6	ノイズ除去機能の動作 .....	434
15.6.7	16 ビットタイマの動作 .....	435
15.7	マルチパルスジェネレータ使用上の注意 .....	439
15.8	マルチパルスジェネレータのプログラム例 .....	441
<b>第 16 章</b>	<b>PWC タイマ .....</b>	<b>443</b>
16.1	PWC タイマの概要 .....	444
16.2	PWC タイマのブロックダイアグラム .....	445
16.3	PWC タイマの端子 .....	446
16.4	PWC タイマのレジスタ .....	448
16.4.1	PWC 状態制御レジスタ (PWCSH0, PWCSH1, PWCSL0, PWCSL1) .....	449
16.4.2	PWC データバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1) .....	454
16.4.3	分周比制御レジスタ (DIV0, DIV1) .....	455
16.5	PWC タイマの割込み .....	456
16.6	PWC タイマの動作 .....	458
16.6.1	動作モードの選択 .....	461
16.6.2	タイマとパルス幅測定の起動 / 停止と, タイマクリア .....	462
16.6.3	タイマモード時の動作 .....	464
16.6.4	パルス幅測定モード時の動作 .....	467
16.7	PWC タイマ使用上の注意 .....	472

16.8 PWC タイマのプログラム例 .....	474
<b>第 17 章 UART .....</b>	<b>477</b>
17.1 UART の概要 .....	478
17.2 UART のブロックダイアグラム .....	480
17.3 UART の構成 .....	483
17.3.1 シリアル制御レジスタ 0, 1 ( SCR0, SCR1 ) .....	485
17.3.2 シリアルモードレジスタ 0, 1 ( SMR0, SMR1 ) .....	487
17.3.3 シリアルステータスレジスタ 0, 1 ( SSR0, SSR1 ) .....	489
17.3.4 シリアル入力データレジスタ 0, 1 ( SIDR0, SIDR1 ) , シリアル出力データレジスタ 0, 1 ( SODR0, SODR1 ) .....	492
17.3.5 通信プリスケール制御レジスタ ( CDCR0, CDCR1 ) .....	494
17.4 UART の割り込み .....	495
17.4.1 受信割り込み発生とフラグ設定のタイミング .....	497
17.4.2 送信割り込み発生とフラグ設定のタイミング .....	499
17.5 UART のボーレート .....	500
17.5.1 専用ボーレートジェネレータによるボーレート .....	502
17.5.2 内部タイマ ( 16 ビットリロードタイマ ) によるボーレート .....	505
17.5.3 外部クロックによるボーレート .....	507
17.6 UART の動作説明 .....	508
17.6.1 非同期モード ( 動作モード 0, 1 ) 時の動作 .....	510
17.6.2 クロック同期モード ( 動作モード 2 ) 時の動作 .....	514
17.6.3 双方向通信機能 ( 動作モード 0, 2 ) .....	516
17.6.4 マスタ / スレーブ型通信機能 ( マルチプロセッサモード時 ) .....	518
17.7 UART 使用上の注意 .....	521
17.8 UART のプログラム例 .....	522
<b>第 18 章 DTP/ 外部割り込み .....</b>	<b>525</b>
18.1 DTP/ 外部割り込みの概要 .....	526
18.2 DTP/ 外部割り込みのブロックダイアグラム .....	527
18.3 DTP/ 外部割り込みの構成 .....	529
18.3.1 DTP/ 外部割り込み要因レジスタ ( EIRR ) .....	530
18.3.2 DTP/ 外部割り込み許可レジスタ ( ENIR ) .....	531
18.3.3 検出レベル設定レジスタ, 上位 ( ELVRH ) .....	533
18.3.4 検出レベル設定レジスタ, 下位 ( ELVRL ) .....	534
18.4 DTP/ 外部割り込みの動作説明 .....	535
18.4.1 外部割り込み機能 .....	538
18.4.2 DTP 機能 .....	539
18.5 DTP/ 外部割り込み使用上の注意 .....	540
18.6 DTP/ 外部割り込み回路のプログラム例 .....	542
<b>第 19 章 遅延割り込み発生モジュール .....</b>	<b>545</b>
19.1 遅延割り込み発生モジュールの概要 .....	546
19.2 遅延割り込み発生モジュールのブロックダイアグラム .....	547
19.3 遅延割り込み発生モジュールの構成 .....	548
19.3.1 遅延割り込み要求発生 / 解除レジスタ ( DIRR ) .....	549
19.4 遅延割り込み発生モジュールの動作説明 .....	550
19.5 遅延割り込み発生モジュール使用上の注意 .....	551
19.6 遅延割り込み発生モジュールのプログラム例 .....	552

<b>第 20 章</b>	<b>8/10 ビット A/D コンバータ</b>	<b>553</b>
20.1	8/10 ビット A/D コンバータの概要	554
20.2	8/10 ビット A/D コンバータのブロックダイアグラム	556
20.3	8/10 ビット A/D コンバータの端子	558
20.4	8/10 ビット A/D コンバータのレジスタ構成	560
20.4.1	A/D 制御ステータスレジスタ 1 ( ADCS1 )	561
20.4.2	A/D 制御ステータスレジスタ 0 ( ADCS0 )	564
20.4.3	A/D データレジスタ ( ADCR0, ADCR1 )	567
20.5	8/10 ビット A/D コンバータの割込み	569
20.6	8/10 ビット A/D コンバータの動作	570
20.6.1	EI <sup>2</sup> OS を使用した場合の変換動作	573
20.6.2	A/D 変換データ保護機能	574
20.7	8/10 ビット A/D コンバータ使用上の注意	576
20.8	8/10 ビット A/D コンバータのプログラム例 1 ( EI <sup>2</sup> OS を使用した場合の単発変換モード )	577
20.9	8/10 ビット A/D コンバータのプログラム例 2 ( EI <sup>2</sup> OS を使用した場合の連続変換モード )	579
20.10	8/10 ビット A/D コンバータのプログラム例 3 ( EI <sup>2</sup> OS を使用した場合の停止変換モード )	582
<b>第 21 章</b>	<b>アドレス一致検出機能</b>	<b>585</b>
21.1	アドレス一致検出機能の概要	586
21.2	アドレス一致検出機能のブロックダイアグラム	587
21.3	アドレス一致検出機能のレジスタ構成	588
21.3.1	プログラムアドレス検出制御ステータスレジスタ ( PACSR )	589
21.3.2	プログラムアドレス検出レジスタ ( PADR0H, M, L, PADR1H, M, L )	591
21.4	アドレス一致検出機能の動作説明	593
21.4.1	アドレス一致検出機能の使用例	594
21.5	アドレス一致検出機能のプログラム例	598
<b>第 22 章</b>	<b>ROM ミラー機能選択モジュール</b>	<b>599</b>
22.1	ROM ミラー機能選択モジュールの概要	600
22.2	ROM ミラー機能選択モジュールレジスタ ( ROMM )	602
<b>第 23 章</b>	<b>512K/1024K ビットフラッシュメモリ</b>	<b>605</b>
23.1	512K/1024K ビットフラッシュメモリの概要	606
23.2	フラッシュメモリのレジスタとセクタ構成	607
23.3	フラッシュメモリ制御ステータスレジスタ ( FMCS )	608
23.4	フラッシュメモリ自動アルゴリズム起動方法	611
23.5	自動アルゴリズム実行状態の確認	613
23.5.1	データポーリングフラグ ( DQ7 )	615
23.5.2	トグルビットフラグ ( DQ6 )	617
23.5.3	タイミングリミット超過フラグ ( DQ5 )	618
23.5.4	セクタ消去タイマフラグ ( DQ3 )	619
23.5.5	トグルビット 2 フラグ ( DQ2 )	620
23.6	フラッシュメモリ書込み / 消去の詳細説明	622
23.6.1	フラッシュメモリを読出し / リセット状態にする	623
23.6.2	フラッシュメモリへデータを書き込む	624
23.6.3	フラッシュメモリのデータを消去する ( チップ消去 )	626

23.6.4	フラッシュメモリの任意のデータを消去する（セクタ消去）.....	627
23.6.5	フラッシュメモリのセクタ消去を一時停止する .....	629
23.6.6	フラッシュメモリのセクタ消去を再開する .....	630
23.7	フラッシュセキュリティの特長 .....	631
23.8	512K ビットフラッシュメモリのプログラム例.....	632
<b>第 24 章</b>	<b>フラッシュシリアル書込み接続例 .....</b>	<b>637</b>
24.1	F <sup>2</sup> MC-16LX ファミリ MB90F462/F462A/F463A シリアル書込み接続の基本構成 .....	638
24.2	シングルチップモード時の接続例（ユーザ電源）.....	641
24.3	シングルチップモード時の接続例（ライタ電源）.....	643
24.4	フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源）.....	645
24.5	フラッシュマイコンプログラマとの最小限の接続例（ライタ電源）.....	647
<b>付録</b>	<b>.....</b>	<b>649</b>
付録 A	I/O マップ .....	650
付録 B	命令 .....	660
<b>索引</b>	<b>.....</b>	<b>721</b>

# 本版での主な変更内容

ページ		変更内容（詳細は本文を参照してください。）
-		MB90460/465 シリーズ に統一
25	第 2 章 デバイスの取扱いについて 2.1 デバイス取扱い上の注意事項	水晶発振回路を変更 ・各量産品において、ご使用される発振子メーカーに発振評価依頼をしてください。を追加
85	第 5 章 クロック 5.3 クロック選択レジスタ (CKSCR)	表 5.3-1 を変更 Bit14 の MCM の機能に「・書込みは動作に影響ありません。」を追加
109	第 6 章 低消費電力モード 6.6 状態遷移図	図 6.6-1 を変更
110	第 6 章 低消費電力モード 6.7 スタンバイモード、リセットの端子状態	表 6.7-1 の下の説明 *3 を変更
122	第 7 章 割込み 7.3.2 割込み制御レジスタの機能	サマリ文を変更
229	第 12 章 16 ビットリロードタイマ 12.1 16 ビットリロードタイマの概要	外部トリガ入力動作を変更
253	第 12 章 16 ビットリロードタイマ 12.8 16 ビットリロードタイマのプログラム例	コーディング例を変更
300	第 14 章 多機能タイマ 14.4.3 タイマ状態制御レジスタ (TCCSH, TCCSL)	表 14.4-2 の bit3 に「1」を書き込んだ後、次のカウントクロックまでにこのビットへの「0」書込みを行うと、カウンタ値の初期化は行われません。を追加
530	第 18 章 DTP/ 外部割込み 18.3.1 DTP/ 外部割込み要因レジスタ (EIRR)	DTP/ 外部割込み要因レジスタ (EIRR) を変更 < 注意事項 > を追加
531	第 18 章 DTP/ 外部割込み 18.3.2 DTP/ 外部割込み許可レジスタ (ENIR)	DTP/ 外部割込み許可レジスタ (ENIR) を変更 < 注意事項 > を追加
535	第 18 章 DTP/ 外部割込み 18.4 DTP/ 外部割込みの動作説明	設定の手順を変更 1. 外部割込み入力として使用する端子と兼用する汎用入出力ポートを入力ポートに設定する。を追加

ページ		変更内容（詳細は本文を参照してください。）
540	第 18 章 DTP/ 外部割込み 18.5 DTP/ 外部割込み使用上の注意	外部割込みの入力極性を変更 要求入力レベルがレベル設定の場合、パルス幅はデータシートに記載されている最少パルス幅以上の期間を必要とします。また割込み入力端子がアクティブレベルを保持しつづけている限りは、DTP/ 外部割込み要因レジスタをクリアしても、割込みコントローラへの割込み要求は発生し続けます。を追加
566	第 20 章 8/10 ビット A/D コンバータ 20.4.2 A/D 制御ステータスレジスタ 0 (ADCS0)	表 20.4-2 を変更 bit5 ~ bit3 に（注意事項）を追加
629	第 23 章 512K/1024K ビットフラッシュメモリ 23.6.5 フラッシュメモリのセクタ消去を一時停止する	フラッシュメモリのセクタ消去を一時停止するを変更 最後の文を次のとおり変更 ・最大 15 $\mu$ s 最大 20 $\mu$ s ・セクタ消去一時停止コマンドは、セクタ消去コマンドあるいはセクタ消去再開コマンド発行後、20 $\mu$ s 以上後に行ってください。
639	第 24 章 フラッシュシリアル書込み接続例 24.1 F <sup>2</sup> MC-16LX ファミリ MB90F462/F462A/F463A シリアル書込み接続の基本構成	表 24.1-1 を変更

変更箇所は、本文中のページ左側の によって示しています。

# 第1章

---

## 概要

**MB90460/465 シリーズの主な特長と基本仕様について説明します。**

- 1.1 MB90460/465 シリーズの特長
- 1.2 MB90460/465 シリーズの製品一覧
- 1.3 MB90460/465 シリーズのブロックダイアグラム
- 1.4 端子配列図
- 1.5 外形寸法図
- 1.6 I/O 端子と端子機能説明
- 1.7 I/O 回路の種類



## 1.1 MB90460/465 シリーズの特長

MB90460/465 シリーズは、高速リアルタイム処理を必要とするアプリケーション用に設計された 16 ビット汎用マイクロコントローラであり、さまざまな工業用機械やモータ（交流インダクションモータやブラシレス直流モータ）の制御に適した製品です。このマイクロコントローラは、交流 / 直流モータ制御用の多機能タイマと、さまざまな型の波形を生成できる直流モータ制御用のマルチパルスジェネレータから構成されています。

命令セットは、コントローラのアプリケーションに対して最適化されるように設計されており、かつ F<sup>2</sup>MC-16LX ファミリの AT アーキテクチャを継承していることにより、広範にわたる制御作業を効率的かつ高速で処理することを可能とします。

### ■ MB90460/465 シリーズの特長

- クロック
  - ・ 組込み PLL クロック逡倍回路
  - ・ 動作クロック（PLL クロック）としては、原発振の 2 分周かあるいは原発振の 1 から 4 逡倍した周波数（原発振が 4MHz の場合は 4MHz, 8MHz, 12MHz, 16MHz のいずれか）から選択できます。
  - ・ 最小命令実行時間は 62.5 ns（原発振が 4MHz, PLL クロックが 4 逡倍, V<sub>CC</sub> が 5.0V の時）
- 16M バイトの CPU アドレス空間
  - ・ 24 ビットの内部アドレス
- アプリケーションに対して最適化された命令セット
  - ・ 豊富なデータタイプ（ビット、バイト、ワード、ロングワード）
  - ・ 豊富なアドレス指定モード（23 種類）
  - ・ 高いコード効率
  - ・ 32 ビットアキュムレータにより実現される高精度演算
- 高級言語（C）とマルチタスク処理用に設計された命令セット
  - ・ システムスタックポインタを採用
  - ・ 高度なポインタ間接命令
  - ・ バレルシフト命令
- プログラムパッチ機能（2 アドレスポインタ）
- 向上した実行速度
  - ・ 4 バイト命令キュー
- 強力な割込み機能
  - ・ 優先順位を設定可能：8 レベル
  - ・ 32 の強力な割込み要因

- CPU 動作に依存しない自動データ転送機能
  - ・ 拡張インテリジェント I/O サービス機能 (EI<sup>2</sup>OS)
  - ・ 最大 16 チャンネル
- 低消費電力モード (スタンバイモード)
  - ・ スリープモード (CPU 動作クロックが停止しているモード)
  - ・ タイムベースタイマモード (発振とタイムベースタイマ以外が停止しているモード)
  - ・ ストップモード (発振が停止しているモード)
  - ・ CPU 間欠動作モード
- パッケージ
  - ・ LQFP-64 (FPT-64P-M09: 0.65mm ピッチ)
  - ・ LQFP-64 (FPT-64P-M06: 1.00mm ピッチ)
  - ・ SDIP-64 (DIP-64P-M01: 1.78mm ピッチ)
- プロセス
  - ・ CMOS

## ■ リソース

- I/O ポート
  - ・ 最大 51 ポート
- 18 ビットタイムベースカウンタ / ウォッチドッグタイマ : 1 チャンネル
- ウォッチドッグタイマ : 1 チャンネル
- PWC: 2 チャンネル
- 16 ビットリロードタイマ : 1 チャンネル
- 16 ビット PPG タイマ : 1 チャンネル
- 多機能タイマ (交流 / 直流モータ制御用) : 1 チャンネル
  - ・ 16 ビットフリーランタイマ (アップモードまたはアップ / ダウンモードの選択が可能であり、バッファ付き) : 1 チャンネル
  - ・ 16 ビットアウトプットコンペア (バッファ付き) : 6 チャンネル
  - ・ 16 ビットインプットキャプチャ : 4 チャンネル
  - ・ 16 ビット PPG タイマ : 1 チャンネル
  - ・ 波形ジェネレータ (バッファとコンペアクリア機能付き 16 ビットタイマを搭載)
- マルチパルスジェネレータ (直流モータ制御用) : 1 チャンネル
  - ・ 16 ビットリロードタイマ : 1 チャンネル
  - ・ 16 ビット PPG タイマ : 1 チャンネル
  - ・ 波形シーケンサ (バッファとコンペアクリア機能付き 16 ビットタイマを搭載)

## 第 1 章 概要

### ● UART: 2 チャンネル

- 全二重の二重バッファ (8 ビット長) を搭載
- クロック非同期転送またはクロック同期転送 (スタートビットとストップビットを使用) を選択的に使用可能

### ● DTP/ 外部割込み回路: 8 チャンネル

- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の起動および外部入力をトリガとする外部割込み生成用モジュール

### ● 遅延割込み生成モジュール

- タスク切換え割込み要求を生成

### ● 8/10 ビット A/D コンバータ: 8 チャンネル

- 8/10 ビット分解能

## 1.2 MB90460/465 シリーズの製品一覧

MB90460/465 シリーズの製品一覧を表 1.2-1 に示します。

### ■ MB90460/465 シリーズの製品一覧

表 1.2-1 MB90460/465 シリーズの製品一覧 ( 1 / 2 )

項目	型格					
	MB90V460	MB90F462	MB90F462A	MB90F463A	MB90462	MB90467
分類	-	フラッシュ ROM			マスク ROM	
ROM 容量	-	64K バイト		128K バイト	64K バイト	
RAM 容量	8K バイト	2K バイト				
エミュレータ専用電源 *	無	-	-	-	-	-
CPU 機能	命令数 : 351 最大実行時間 : 62.5ns/4MHz ( PLL: 4MHz × 4 ) アドレス指定モード : 23 データビット長 : 1, 8, 16 ビット 最大メモリ空間 : 16M バイト					
I/O ポート	I/O ポート ( CMOS ) : 51					
PWC	パルス幅カウンタタイマ : 2 チャンネル					1 チャンネル
UART	全二重の二重バッファ ( 8 ビット長 ) を搭載 クロック非同期転送またはクロック同期転送 ( スタートビットとストップビット使用 ) を選択的に使用可能					
16 ビットリロードタイマ	リロードタイマ : 2 チャンネル リロードモード , シングルショットモードまたはイベントカウントモードを選択可能 マルチパルスジェネレータと一緒に動作することが可能。また単独での動作も可能					
16 ビット PPG タイマ	PPG タイマ : 3 チャンネル PWM モードまたはシングルショットモードを選択可能 多機能タイマ / マルチパルスジェネレータと一緒に動作することが可能。また単独での動作も可能					2 チャンネル
多機能タイマ ( 交流 / 直流モータ制御用 )	16 ビットフリーランタイマ ( アップモードまたはアップ / ダウンモード選択可能。バッファ付き ) : 1 チャンネル 16 ビットアウトプットコンペア : 6 チャンネル 16 ビットインプットキャプチャ : 4 チャンネル 16 ビット PPG タイマ : 1 チャンネル 波形ジェネレータ ( 16 ビットタイマ : 3 チャンネル。3 位相波形またはデッドタイム )					

表 1.2-1 MB90460/465 シリーズの製品一覧 ( 2 / 2 )

項目	型格					
	MB90V460	MB90F462	MB90F462A	MB90F463A	MB90462	MB90467
マルチパルスジェネレータ ( 直流モータ制御用 )	16 ビット PPG タイマ : 1 チャンネル 16 ビットリロードタイマ動作 ( トグル出力またはワンショット出力を選択可能 ) イベントカウンタ機能 ( 1 チャンネル内蔵 ) 波形シーケンサ ( バッファとコンペアクリア機能付き 16 ビットタイマを搭載 )					-
8/10 ビット A/D コンバータ	8/10 ビット分解能 ( 8 チャンネル ) 変換時間 : 6.13μs 未満 ( 16 MHz 内部クロック )					
外部割込み	独立した 8 チャンネル 選択可能な要因 : 立上がり , 立下がり , "L" レベルまたは "H" レベル					
低消費電力	停止モード , スリープモード , CPU 間欠動作モード					
プロセス	CMOS					
パッケージ	PGA256	LQFP-64 ( FPT-64P-M09: 0.65mm ピッチ ) LQFP-64 ( FPT-64P-M06: 1.00mm ピッチ ) SDIP-64 ( DIP-64P-M01: 1.78mm ピッチ )				
動作電圧	5V ± 10% @16MHz					

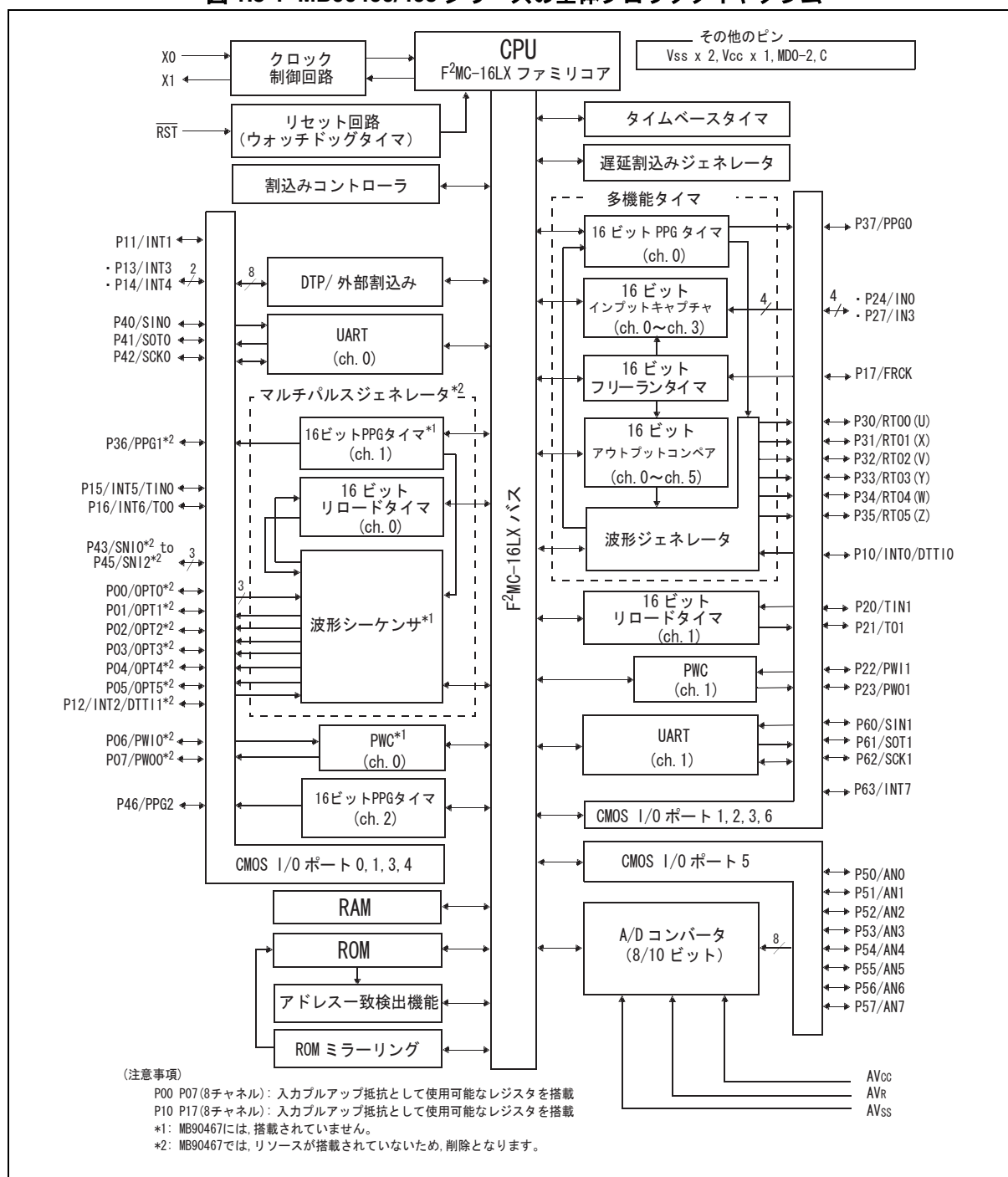
\*: エミュレーションポッド ( MB2145-507 ) を使用頂く際のディップスイッチ ( S2 ) の設定です。詳細につきましては『MB2145-507 ハードウェアマニュアル ( 2.7 エミュレータ専用電源端子 )』をご参照ください。

### 1.3 MB90460/465 シリーズのブロックダイアグラム

**MB90460/465 シリーズの全体ブロックダイヤグラムを図 1.3-1 に示します。**

## ■ MB90460/465 シリーズの全体ブロックダイヤグラム

**図 1.3-1 MB90460/465 シリーズの全体ブロックダイヤグラム**

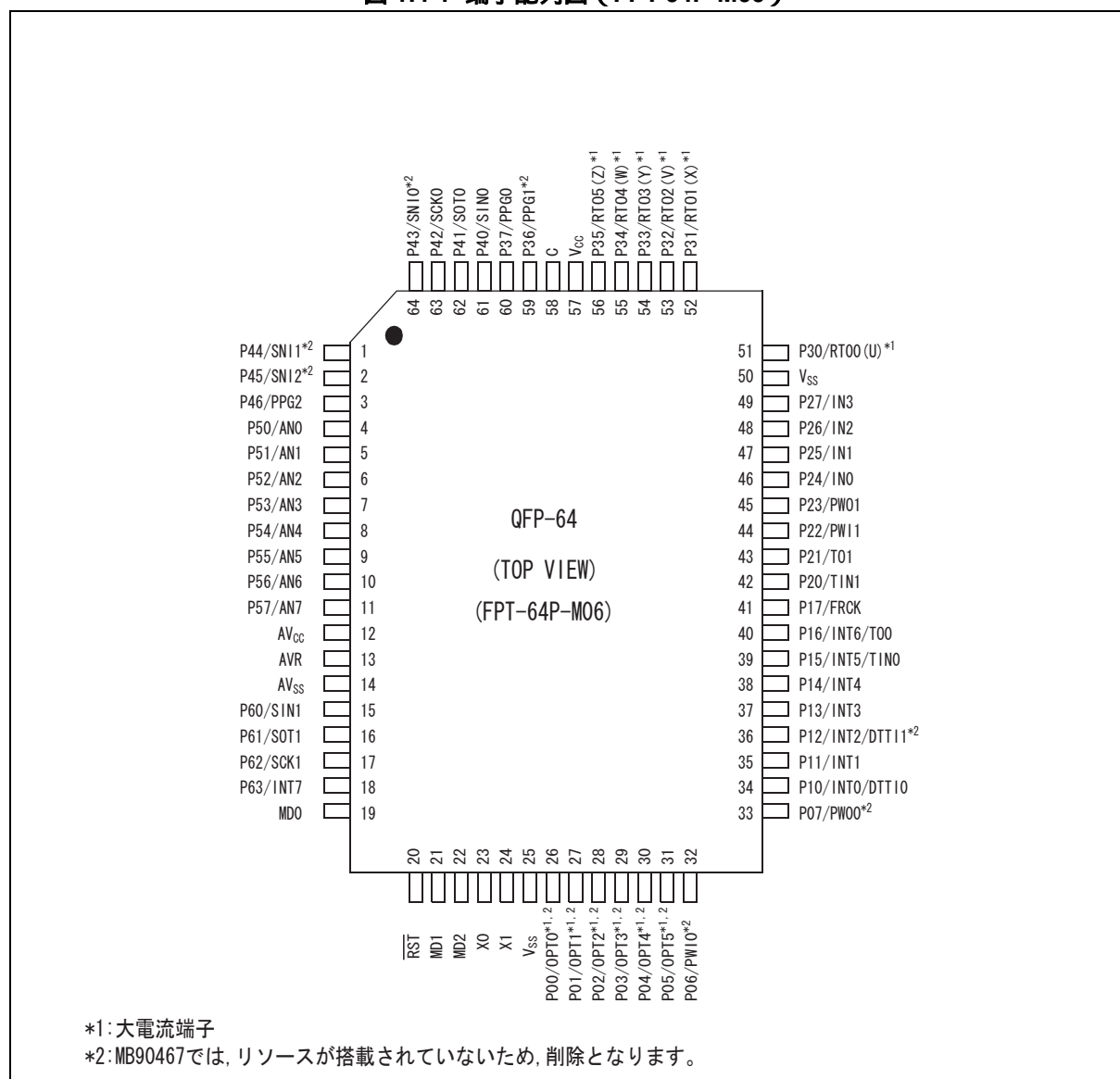


## 1.4 端子配列図

MB90460/465 シリーズの端子配列図を図 1.4-1 ~ 図 1.4-3 に示します。

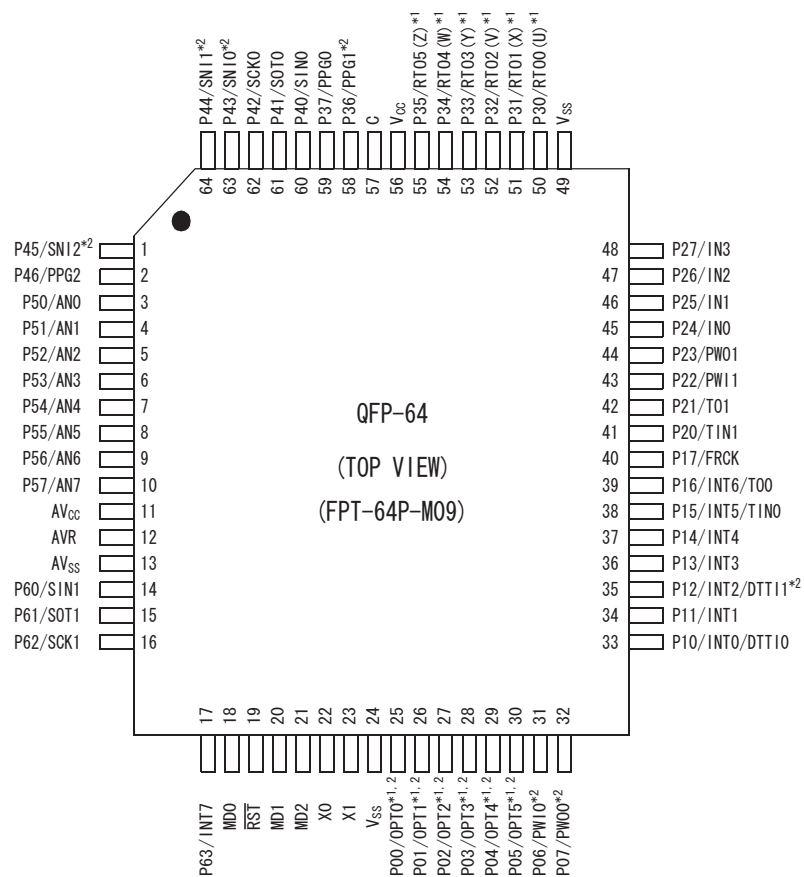
### ■ 端子配列図 (FPT-64P-M06)

図 1.4-1 端子配列図 (FPT-64P-M06)



## ■ 端子配列図 (FPT-64P-M09)

図 1.4-2 端子配列図 (FPT-64P-M09)



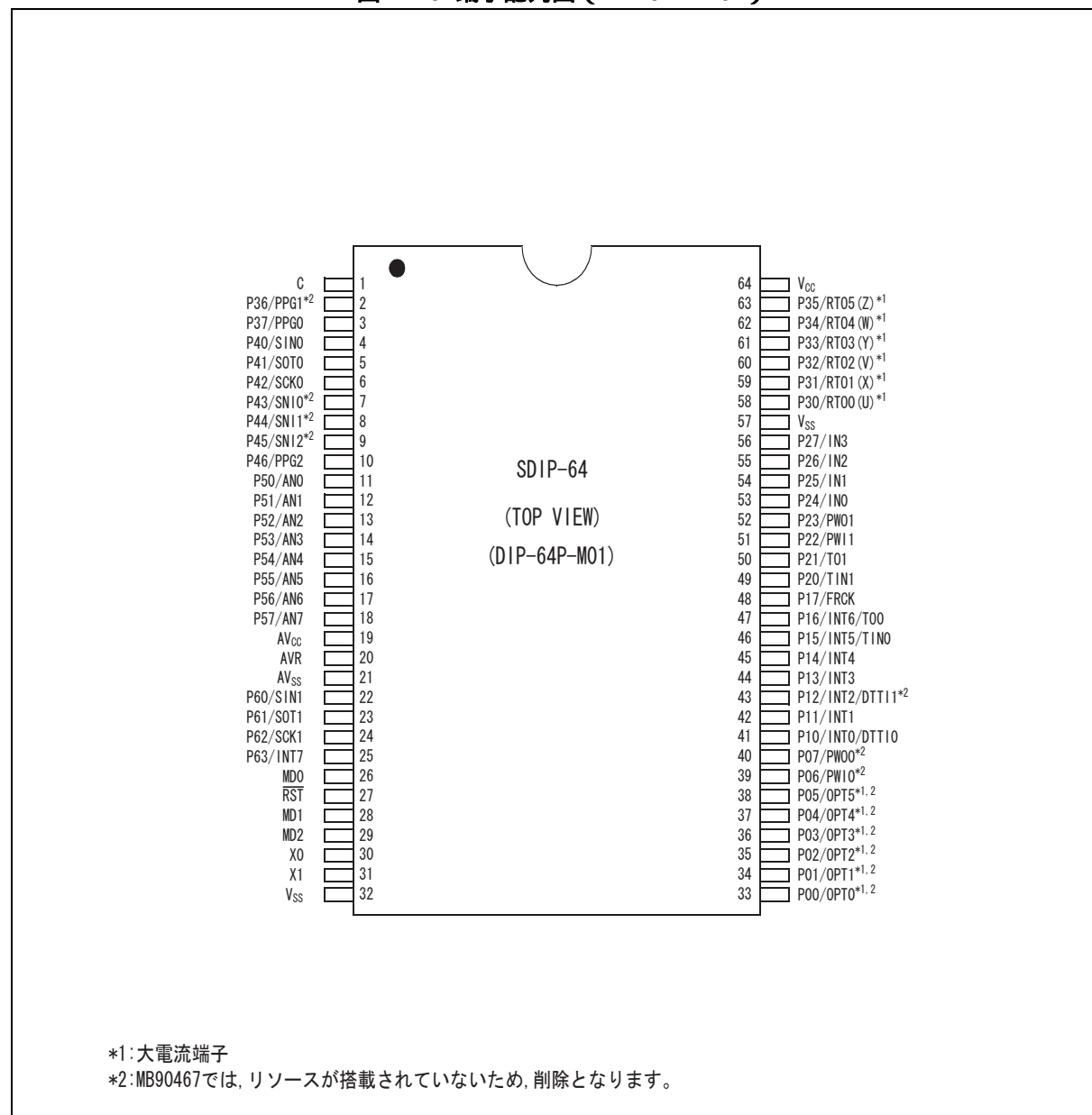
\*1:大電流端子

\*2:MB90467では、リソースが搭載されていないため、削除となります。



## ■ 端子配列図 (DIP-64P-M01)

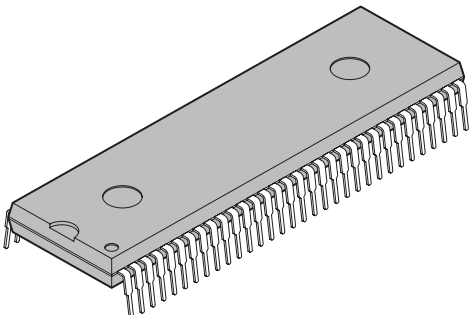
図 1.4-3 端子配列図 (DIP-64P-M01)

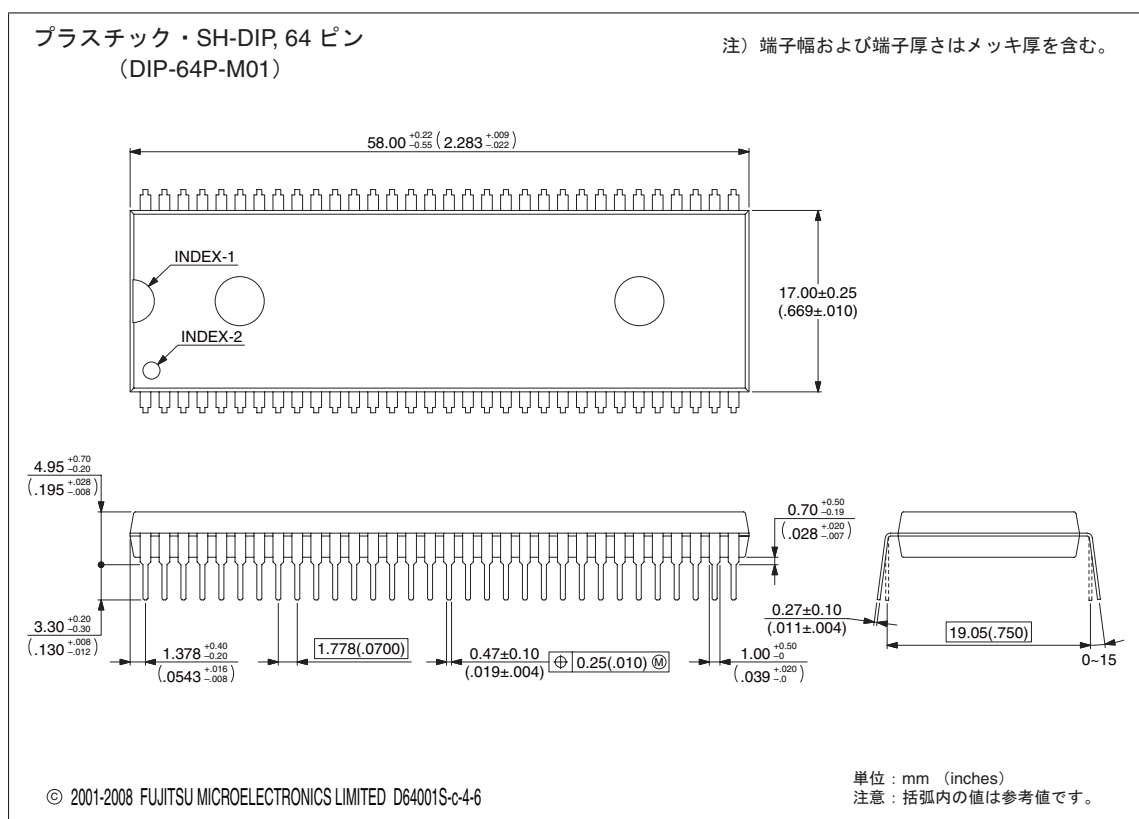


## 1.5 外形寸法図

MB90460/465 シリーズには、3 種類のパッケージが用意されています。  
 なお、本外形寸法図は参考用です。正式版につきましては別途ご相談ください。

### ■ DIP-64P-M01 の外形寸法図

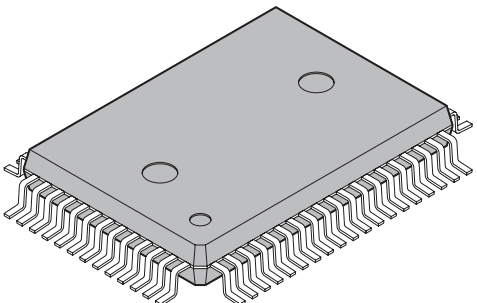
<p>プラスチック・SH-DIP, 64 ピン</p>  <p>(DIP-64P-M01)</p>	リードピッチ	1.778mm
	パッケージ幅× パッケージ長さ	17 × 58mm
	封止方法	プラスチックモールド
	取付け高さ	5.65mm MAX

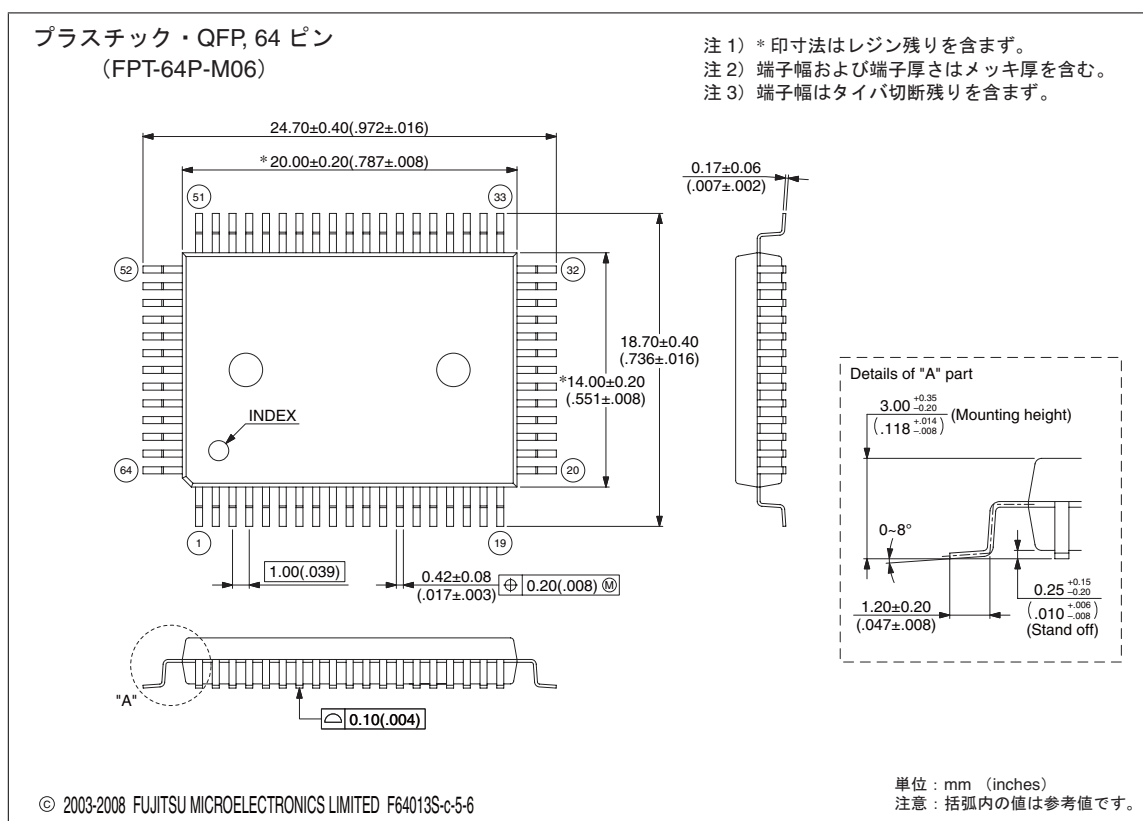


最新の外形寸法図については、下記の URL にてご確認ください。

<http://edevic.fujitsu.com/package/jp-search/>

## ■ FPT-64P-M06 の外形寸法図

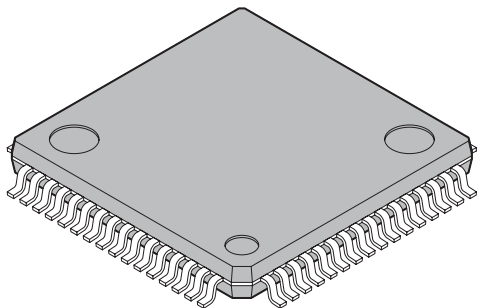
<p>プラスチック・QFP, 64 ピン</p>  <p>(FPT-64P-M06)</p>	リードピッチ	1.00mm
	パッケージ幅× パッケージ長さ	14 × 20mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	3.35mm MAX
	コード (参考)	P-QFP64-14×20-1.00

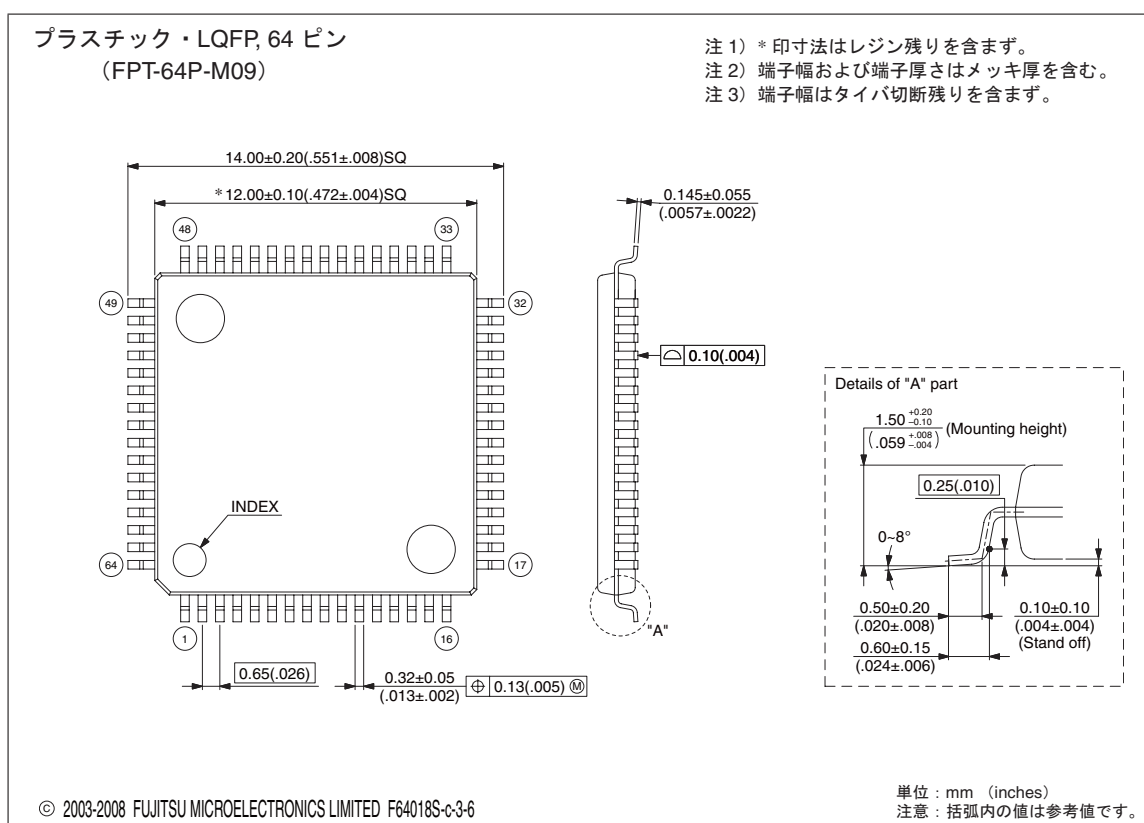


最新の外形寸法図については、下記の URL にてご確認ください。

<http://edevic.fujitsu.com/package/jp-search/>

## ■ FPT-64P-M09 の外形寸法図

<p>プラスチック・LQFP, 64 ピン</p>  <p>(FPT-64P-M09)</p>	リードピッチ	0.65mm
	パッケージ幅× パッケージ長さ	12 × 12mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	1.70mm MAX
	コード (参考)	P-LQFP64-12×12-0.65



最新の外形寸法図については, 下記の URL にてご確認ください。

<http://edevic.fujitsu.com/package/jp-search/>

## 1.6 I/O 端子と端子機能説明

MB90460/465 シリーズの I/O 端子とこれらの端子の機能を表 1.6-1 に示し、I/O 回路の種類を表 1.7-1 に示します。

表 1.6-1 における “ I/O 回路の種類 ” 欄の英字は、表 1.7-1 における “ 分類 ” 欄の英字を示します。

### ■ I/O 端子と端子機能

表 1.6-1 I/O 端子と端子機能 (1 / 5)

端子番号			端子名	I/O 回路	リセット 時の端子 の状態	機能
QFP- M09*1	QFP- M06*2	SDIP*3				
22, 23	23, 24	30, 31	X0, X1	A	発振中	発振中発振入力端子です。
19	20	27	$\overline{\text{RST}}$	B	リセット 入力	リセット入力外部リセット入力端子です。
25 ~ 30	26 ~ 31	33 ~ 38	P00 ~ P05	D	入力 ポート	汎用入出力ポートです。
			OPT0 ~ OPT5*4			波形シーケンサの出力端子（OPTn）です。これ らの端子は、波形シーケンサ回路の出力データレ ジスタで指定された波形を出力します。波形は OPCR の OPnE を有効にすると出力されます。*4
31	32	39	P06	E		汎用入出力ポートです。
			PWIO*4			PWC 0 信号入力端子です。*4
32	33	40	P07	E		汎用入出力ポートです。
			PWO0*4			PWC 0 信号出力端子です。*4
33	34	41	P10	C		汎用入出力ポートです。
			INT0			外部割込み要求入力チャネル 0 として使用できま す。スタンバイモード時には EN0 に "1" を設定 して、入力ポートに設定した場合に有効になりま す。
			DTTI0			RTO0 ~ RTO5 端子使用時の端子レベル固定入力 端子です。この機能は、波形ジェネレータで各入 力を有効にすると有効になります。
34	35	42	P11	C		汎用入出力ポートです。
			INT1		外部割込み要求入力チャネル 1 として使用できま す。スタンバイモード時には EN1 に "1" を設定 して、入力ポートに設定した場合に有効になりま す。	

表 1.6-1 I/O 端子と端子機能 (2 / 5)

端子番号			端子名	I/O 回路	リセット 時の端子 の状態	機能
QFP- M09 <sup>*1</sup>	QFP- M06 <sup>*2</sup>	SDIP <sup>*3</sup>				
35	36	43	P12	C	入力 ポート	汎用入出力ポートです。
			INT			外部割込み要求入力チャネル 2 として使用できます。スタンバイモード時には EN2 に "1" を設定して、入力ポートに設定した場合に有効になります。
			DTTI1 <sup>*4</sup>			OPTO0 ~ OPTO5 端子使用時の端子レベル固定入力端子です。この機能は、波形シーケンサで各入力を有効にすると有効になります。 <sup>*4</sup>
36, 37	37, 38	44, 45	P13, P14	C		汎用入出力ポートです。
			INT3, INT4			外部割込み要求入力チャネル 3, 4 として使用できます。スタンバイモード時には EN3 と EN4 に "1" を設定して、入力ポートに設定した場合に有効になります。
38	39	46	P15	C		汎用入出力ポートです。
			INT5			外部割込み要求入力チャネル 5 として使用できます。スタンバイモード時には EN5 に "1" を設定して、入力ポートに設定した場合に有効になります。
			TIN0			リロードタイマ 0 用の外部クロック入力端子です。
39	40	47	P16	C		汎用入出力ポートです。
			INT6			外部割込み要求入力チャネル 6 として使用できます。スタンバイモード時には EN6 に "1" を設定して、入力ポートに設定した場合に有効になります。
			TO0			リロードタイマ 0 のイベント出力端子です。
40	41	48	P17	C		汎用入出力ポートです。
			FRCK			フリーランタイム用の外部クロック入力端子です。
41	42	49	P20	F		汎用入出力ポートです。
			TIN1			リロードタイマ 1 用の外部クロック入力端子です。
42	43	50	P21	F		汎用入出力ポートです。
			TO1			リロードタイマ 1 用のイベント出力端子です。
43	44	51	P22	F		汎用入出力ポートです。
			PWI1			PWC 1 信号入力端子です。

## 第 1 章 概要

表 1.6-1 I/O 端子と端子機能 (3 / 5)

端子番号			端子名	I/O 回路	リセット 時の端子 の状態	機能
QFP- M09*1	QFP- M06*2	SDIP*3				
44	45	52	P23	F	入力 ポート	汎用入出力ポートです。
			PW01			PWC 1 信号出力端子です。
45 ~ 48	46 ~ 49	53 ~ 56	P24 ~ P27	F		汎用入出力ポートです。
			IN0 ~ IN3			インプットキャプチャチャネル 0 ~ 3 のトリガ入 力端子です。インプットキャプチャトリガ入力 を設定し、入力ポートを設定した場合に入力でき ます。
50 ~ 55	51 ~ 56	58 ~ 63	P30 ~ P35	G		汎用入出力ポートです。
			RTO0 ~ RTO5			波形ジェネレータ出力端子です。これらの端子は、 波形ジェネレータに指定した波形を出力します。 出力波形は、波形ジェネレータ出力を有効にす ると生成されます。
58, 59	59, 60	2, 3	P36, P37	H		汎用入出力ポートです。
			PPG1*5, PPG0			PPG チャネル 1, 0 の出力端子です。この機能は、 PPG チャネル 1, 0 の出力を有効にすると有効にな ります。*5
60	61	4	P40	F		汎用入出力ポートです。
			SIN0			UART チャネル 0 のシリアルデー タ入力端子で す。UART チャネル 0 が入力動作中は、シリアル データ以外は入力しないでください。
61	62	5	P41	F		汎用入出力ポートです。
			SOT0			UART チャネル 0 のシリアルデー タ出力端子で す。この機能は、UART チャネル 0 のデー タ出力 を有効にすると有効になります。
62	63	6	P42	F		汎用入出力ポートです。
			SCK0			UART チャネル 0 のシリアルクロック I/O 端子で す。この機能は、UART チャネル 0 のクロック出 力を有効にすると有効になります。
63	64	7	P43	F		汎用入出力ポートです。
			SNIO*4			波形シーケンサの位置検出用トリガ入力端子で す。端子を入力として設定した場合は、トリガ入 力以外は入力しないでください。*4
64	1	8	P44	F	汎用入出力ポートです。	
			SNII*4		マルチパルスジェネレータの位置検出用トリガ入 力端子です。端子を入力として設定した場合は、 トリガ入力以外は入力しないでください。*4	

表 1.6-1 I/O 端子と端子機能 (4 / 5)

端子番号			端子名	I/O 回路	リセット 時の端子 の状態	機能
QFP- M09 <sup>*1</sup>	QFP- M06 <sup>*2</sup>	SDIP <sup>*3</sup>				
1	2	9	P45	F	入力 ポート	汎用入出力ポートです。
			SNI2 <sup>*4</sup>			マルチパルスジェネレータの位置検出用トリガ入力端子です。端子を入力として設定した場合は、トリガ入力以外は入力しないでください。 <sup>*4</sup>
2	3	10	P46	F		汎用入出力ポートです。
			PPG2			PPG チャネル 2 の出力端子です。この機能は、PPG チャネル 2 の出力を有効にすると有効になります。
3 ~ 10	4 ~ 11	11 ~ 18	P50 ~ P57	I	アナログ 入力	汎用入出力ポートです。
			AN0 ~ AN7			A/D コンバータアナログ入力端子です。この機能は、アナログ入力を有効すると有効になります (ADER)。
11	12	19	AV <sub>CC</sub>	-	電源入力	A/D コンバータの電源入力端子です。
12	13	20	AVR	J		A/D コンバータのリファレンス入力端子です。この電圧は V <sub>CC</sub> を超えてはいけません。
13	14	21	AV <sub>SS</sub>	-		A/D コンバータの電源入力端子 (0V) です。
14	15	22	P60	F	入力 ポート	汎用入出力ポートです。
			SIN1			UART チャネル 1 のシリアルデータ入力端子です。UART チャネルが入力動作中はシリアルデータ以外は入力しないでください。
15	16	23	P61	F		汎用入出力ポートです。
			SOT1			UART チャネル 1 のシリアルデータ出力端子です。この機能は、UART チャネル 1 のデータ出力を有効にすると有効になります。
16	17	24	P62	F		汎用入出力ポートです。
			SCK1			UART チャネル 1 のシリアルクロック I/O 端子です。この機能は、UART チャネル 1 のクロック出力を有効にすると有効になります。
17	18	25	P63	F		汎用入出力ポートです。
			INT7			外部割込み要求入力チャネル 7 として使用できます。スタンバイモード時には EN7 に "1" を設定して、入力ポートに設定した場合に有効になります。
18	19	26	MD0	L	モード 入力	動作モードの指定用入力端子です。この端子は、V <sub>CC</sub> または V <sub>SS</sub> に直結してください。
20, 21	21, 22	28, 29	MD1, MD2	L		動作モードの指定用入力端子です。この端子は、V <sub>CC</sub> または V <sub>SS</sub> に直結してください。



## 第 1 章 概要

表 1.6-1 I/O 端子と端子機能 (5 / 5)

端子番号			端子名	I/O 回路	リセット 時の端子 の状態	機能
QFP- M09 <sup>*1</sup>	QFP- M06 <sup>*2</sup>	SDIP <sup>*3</sup>				
24, 49	25, 50	32, 57	V <sub>SS</sub>	-	電源入力	電源（0V）入力端子です。
56	57	64	V <sub>CC</sub>	-		電源（5V）入力端子です。

\*1: FPT-64P-M09

\*2: FPT-64P-M06

\*3: DIP-64P-M01

\*4:MB90467 では、リソースが搭載されていないため、削除となります。

\*5:MB90467 では、リソースが搭載されていないため、PPG1 は削除となります。

## 1.7 I/O 回路の種類

MB90460/465 シリーズの I/O 回路の種類を表 1.7-1 に示します。

### ■ I/O 回路の種類

表 1.7-1 入出力回路形式 ( 1 / 3 )

分類	回路	備考
A		発振回路 ・ 発振帰還抵抗は約 $1\text{M}\Omega$
B		・ ヒステリシス入力 ・ 抵抗は約 $50\text{k}\Omega$
C		・ CMOS 出力 ・ ヒステリシス入力 ・ プルアップ抵抗選択可能 (約 $50\text{k}\Omega$ ) ・ $I_{OL}=4\text{mA}$
D		・ CMOS 出力 ・ CMOS 入力 ・ プルアップ抵抗選択可能 (約 $50\text{k}\Omega$ ) ・ $I_{OL}=12\text{mA}$

表 1.7-1 入出力回路形式 ( 2 / 3 )

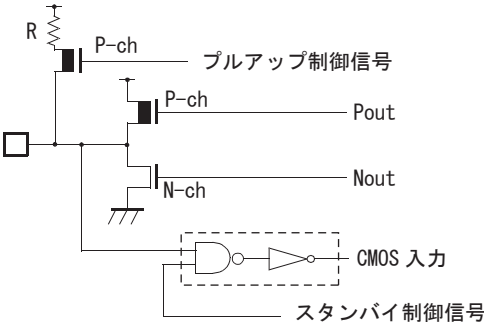
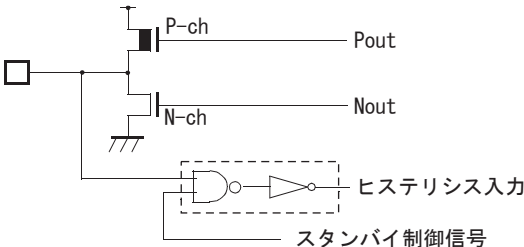
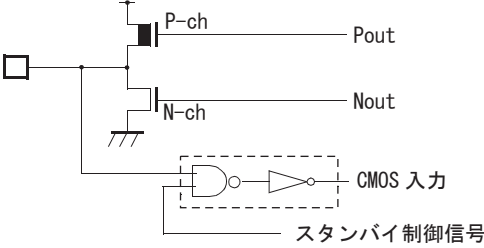
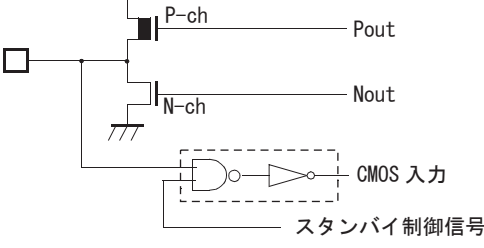
分類	回路	備考
E		<ul style="list-style-type: none"> <li>• CMOS 出力</li> <li>• CMOS 入力</li> <li>• プルアップ抵抗選択可能 ( 約 50kΩ )</li> <li>• <math>I_{OL}=4mA</math></li> </ul>
F		<ul style="list-style-type: none"> <li>• CMOS 出力</li> <li>• ヒステリシス入力</li> <li>• <math>I_{OL}=4mA</math></li> </ul>
G		<ul style="list-style-type: none"> <li>• CMOS 出力</li> <li>• CMOS 入力</li> <li>• <math>I_{OL}=12mA</math></li> </ul>
H		<ul style="list-style-type: none"> <li>• CMOS 出力</li> <li>• CMOS 入力</li> <li>• <math>I_{OL}=4mA</math></li> </ul>

表 1.7-1 入出力回路形式 ( 3 / 3 )

分類	回路	備考
I		<ul style="list-style-type: none"> <li>• CMOS 出力</li> <li>• CMOS 入力</li> <li>• A/D コンバータアナログ入力</li> <li>• <math>I_{OL}=4mA</math></li> </ul>
J		<ul style="list-style-type: none"> <li>• 保護回路付き A/D コンバータ基準電圧 (AVR) 入力端子</li> </ul>
K		<ul style="list-style-type: none"> <li>• ヒステリシス入力</li> </ul>



## 第2章

---

# デバイスの取扱い について

デバイスの取扱い上の注意について説明します。

### 2.1 デバイス取扱い上の注意事項

## 2.1 デバイス取扱い上の注意事項

---

デバイスを取扱う際には、以下の項目について特に注意してください。

- 最大定格電圧の厳守（ラッチアップの防止）
  - 供給電圧の安定化
  - 電源端子
  - 水晶発振回路
  - 外部クロックを使用する場合
  - PLL クロックモード動作中の注意について
  - 未使用入力端子の処理
  - N.C. 端子の処理
  - A/D コンバータを使用しない場合の端子処理
- 

### ■ デバイス取扱い上の注意事項

#### ● 最大定格電圧の厳守（ラッチアップの防止）

- CMOS IC においては、中・高耐圧以外の入力端子や出力端子に、 $V_{CC}$  よりも高い電圧や  $V_{SS}$  より低い電圧が印加された場合または、 $V_{CC}$  に定格を超える電圧が印加された場合に、ラッチアップ現象を発生する場合があります。  
ラッチアップが発生した場合、電源電流が激増し、素子の熱破壊に至る場合がありますので、使用に際しては最大定格を超えないように注意してください。
- アナログ電源を投入または切断する場合、アナログ供給電圧（ $AV_{CC}$ ,  $AVR$ ）とアナログ入力電圧は、デジタル電源電圧（ $V_{CC}$ ）を超えないように注意してください。

#### ● 供給電圧の安定化

- $V_{CC}$  電源電圧の動作保障範囲内であっても、電源電圧が急激に変化した場合は、誤動作を起こす場合がありますので、 $V_{CC}$  電源電圧を安定させてください。
- 安定化の基準としましては、商用周波数（50Hz ~ 60Hz）での  $V_{CC}$  リップル変動（peak to peak 値）を標準  $V_{CC}$  電源電圧値の "10%" 以下に、また、電源を切り換える瞬時変化におきましては過渡変動率が 0.1V/ms 以下になるように電源電圧を安定させてください。

#### ● 電源端子

- 電流供給源は、低インピーダンスでデバイスの  $V_{CC}$  へ接続してください。
- 電源ノイズ対策として、 $V_{CC}$  や  $V_{SS}$  端子間（デバイスの  $V_{CC}$  や  $V_{SS}$  端子の近く）に 0.1  $\mu$ F 程度のコンデンサをバイパスコンデンサとして接続してください。

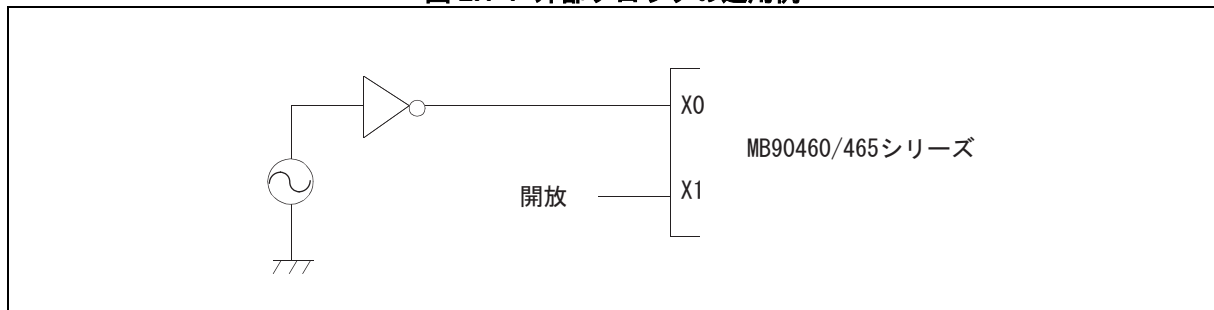
● 水晶発振回路

- X0 や X1 端子へのノイズは、デバイスの誤動作の原因となります。X0 端子、水晶振動子（あるいはセラミック振動子）およびグランドへのバイパスコンデンサは、X0 や X1 端子の近くに接続してください。  
また、X0 や X1 端子の配線は、ほかの配線と交差しないようにプリント基板を設計してください。
- X0 や X1 端子の回りをグランドで囲むようなプリント基板アートワークは、安定した動作を期待できます。
- 各量産品において、ご使用される発振子メーカーに発振評価依頼をしてください。

● 外部クロックを使用する場合

- 外部クロックを使用する場合は、X0 端子のみを駆動し、X1 端子は開放してください。外部クロック使用例を図 2.1-1 に示します。

図 2.1-1 外部クロックの適用例



● PLL クロックモード動作中の注意について

- 本マイコンで PLL クロックを選択しているときに発振子が外れたり、あるいはクロック入力が停止した場合、本マイコンは PLL 内部の自励発振回路の自走周波数で動作を継続し続ける場合があります。この動作は保証外の動作です。

● 未使用入力端子の処理

- 使用していない入力端子を開放のままにした場合、誤動作およびラッチアップによる永久破壊の原因となることがあります。2kΩ 以上の抵抗を介してプルアップまたはプルダウン処理してください。  
また、使用していない入出力端子が存在する場合は、出力状態に設定して開放するか、入力状態に設定して入力端子と同じように処理してください。

● N.C. 端子の処理

- N.C.（内部接続）端子は、必ず開放にして使用してください。

● A/D コンバータを使用しない場合の端子処理

- A/D コンバータを使用しない場合は、 $AV_{CC} = AVR = V_{CC}$ 、 $AV_{SS} = V_{SS}$  となるように接続してください。





# 第3章

---

## CPU

MB90460/465 シリーズの機能と動作について説明します。

- 3.1 CPU の概要
- 3.2 メモリ空間
- 3.3 メモリマップ
- 3.4 アドレス指定
- 3.5 多バイト長データのメモリ配置
- 3.6 レジスタ
- 3.7 専用レジスタ
- 3.8 汎用レジスタ
- 3.9 プリフィックスコード

## 3.1 CPU の概要

---

F<sup>2</sup>MC-16LX ファミリ CPU コアは、民生用や車載用機器等の高速リアルタイム処理が要求されるアプリケーション用に設計された 16 ビット CPU です。F<sup>2</sup>MC-16LX ファミリ命令セットは制御アプリケーション用に設計されており、各種制御の高速かつ高効率処理が可能です。

F<sup>2</sup>MC-16LX ファミリ CPU コアは、内部 32 ビットアキュムレータを搭載しているので 32 ビット処理も可能です。メモリ空間は最大 16M バイトであり、このメモリ空間へはリニア方式およびバンク方式のいずれでもアクセス可能です。また、命令体系は、F<sup>2</sup>MC-8L ファミリおよび F<sup>2</sup>MC-16L ファミリの AT アーキテクチャをベースにして、C 言語対応命令の追加、アドレス指定モードの拡張、乗除算命令の強化およびビット処理の充実化により強化されています。以下に、F<sup>2</sup>MC-16LX ファミリ CPU の特長を示します。

---

### ■ CPU の概要

- 最小命令実行時間 :62.5ns ( 原発振 4MHz, PLL クロック 4 通倍のとき )
- 最大メモリ空間 :16M バイト, リニア / バンク方式アドレス指定にてアクセス
- 命令体系
  - ・ データタイプ : ビット, バイト, ワード, ロングワード
  - ・ アドレス指定モード :23 種類
  - ・ 32 ビットアキュムレータ採用による演算精度の強化
  - ・ 符号付き乗除算, 拡張 RETI 命令
- 割込み機能
  - 8 つのプライオリティレベル ( プログラマブル )
- CPU に依存しない自動転送機能
  - 最大 16 チャンネルまでの拡張インテリジェント I/O サービス
- C 言語, マルチタスクに対応した命令体系
  - システムスタックポイントの採用, 命令セットの対称性, パレルシフト命令
- 実行速度の向上 :4 バイトの命令キュー

---

< 注意事項 > MB90460/465 シリーズは、シングルチップモードによる使用となるため、内蔵 ROM, 内蔵 RAM, リソースの空間のみへアクセスでき、ほかへはアクセスできません。

---

## 3.2 メモリ空間

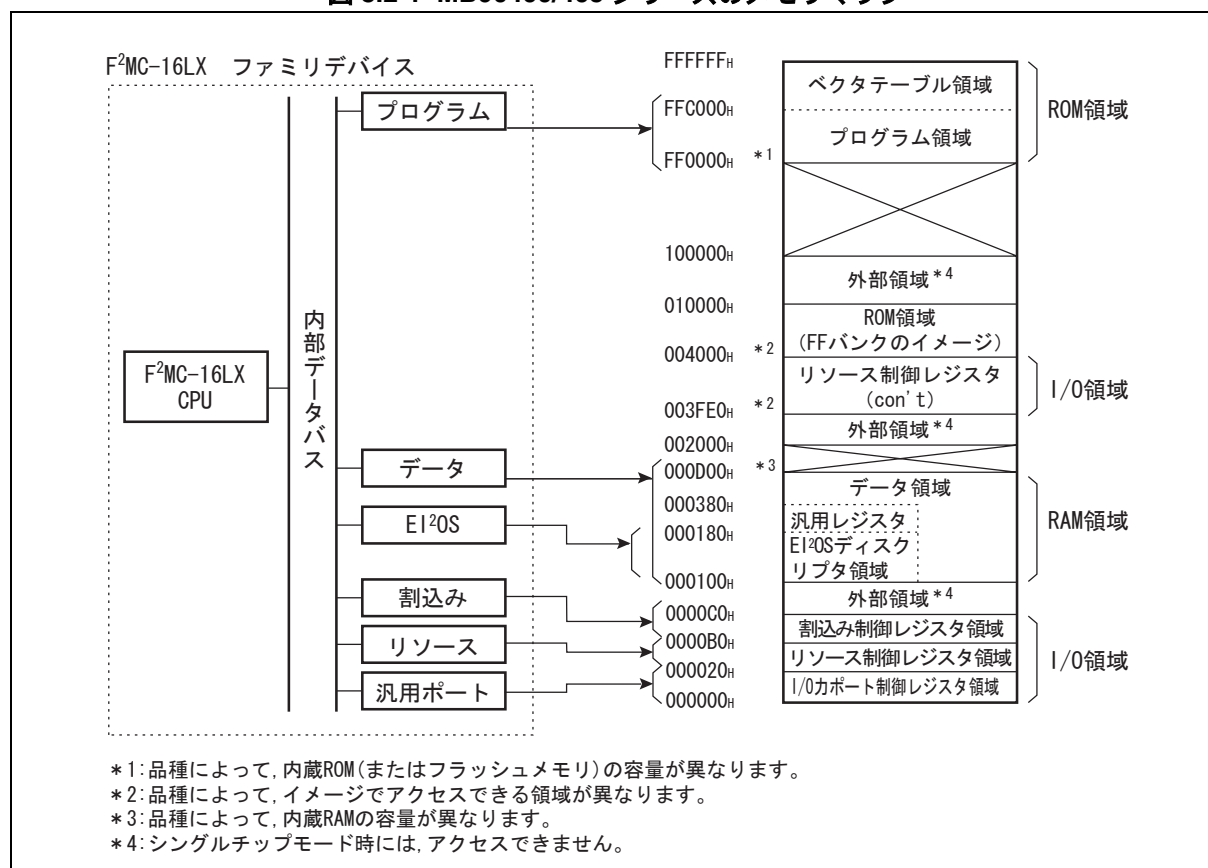
F<sup>2</sup>MC-16LX ファミリの I/O, プログラムおよびデータは, 16M バイトのメモリ空間に配置されます。RAM 領域は, 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) ディスクリプタおよび汎用レジスタに使用されます。

### ■ メモリ空間

I/O, プログラムおよびデータは, F<sup>2</sup>MC-16LX ファミリ CPU が持つ 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によってメモリ空間のアドレスを示し, 各リソースへアクセスできます。

MB90460/465 シリーズのメモリマップを図 3.2-1 に示します。

図 3.2-1 MB90460/465 シリーズのメモリマップ



### ■ ROM 領域

#### ● ベクタテーブル領域 (アドレス: "FFFC00H ~ FFFFFFFH")

- ベクタコール命令, 割込みベクタおよびリセットベクタの各々のベクタテーブルとして使用します。
- ROM 領域の最上位に割当てられており, 対応する処理ルーチンの開始アドレスをベクタコール命令, 割込みベクタおよびリセットベクタのベクタテーブルのアドレスにデータとして設定します。

### ● プログラム領域 (アドレス : " ~ FFFBFF<sub>H</sub>" )

- ・ マスク ROM ( またはフラッシュメモリ ) が , 内部プログラム領域として内蔵されています。
- ・ 内蔵 ROM 容量 ( またはフラッシュメモリ容量 ) は , 品種によって異なります。

## ■ RAM 領域

### ● データ領域 ( アドレス : "000100<sub>H</sub>" ~ )

- ・ スタティック RAM が , 内部データ領域として内蔵されています。
- ・ 内蔵 RAM 容量は , 品種によって異なります。

### ● 汎用レジスタ領域 ( アドレス : "000180<sub>H</sub> ~ 00037F<sub>H</sub>" )

- ・ 8 ビット , 16 ビット , 32 ビットの演算や転送に使用する汎用レジスタが配置されています。
- ・ 汎用レジスタとして使用しない場合は , 通常の RAM として使用できます。
- ・ 汎用レジスタとして使用した場合 , 汎用レジスタアドレス指定は , 短い命令による高速アクセスを可能とします。

### ● 拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) ディスクリプタ領域 ( アドレス : "000100<sub>H</sub> ~ 00017F<sub>H</sub>" )

- ・ 拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) の転送モード , I/O のアドレス , 転送数およびバッファアドレスを設定します。
- ・ 拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) を使用しない場合は , 通常の RAM として使用できます。

## ■ I/O 領域

### ● 割込み制御レジスタ領域 ( アドレス : "0000B0<sub>H</sub> ~ 0000BF<sub>H</sub>" )

割込み制御レジスタ ( ICR00 ~ ICR15 ) は , 割込み機能を持つ全てのリソースに対応しています。これらのレジスタは割込レベルと拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) を設定します。

### ● リソース制御レジスタ領域 ( アドレス : "000080<sub>H</sub> ~ 00000F<sub>H</sub>" , "000019<sub>H</sub> ~ 0000AF<sub>H</sub>" , "003FE0<sub>H</sub> ~ 003FFF<sub>H</sub>" )

これらのレジスタはリソースを制御し , かつデータを入出力します。I/O アドレスを指定する命令 ( MOV A, io ) は 003FE0<sub>H</sub> ~ 003FFF<sub>H</sub> レジスタ領域をサポートしていません。

### ● I/O ポート制御レジスタ領域 ( アドレス : "000000<sub>H</sub> ~ 000006<sub>H</sub>" , "000010<sub>H</sub> ~ 000017<sub>H</sub>" )

これらのレジスタは I/O ポートを制御し , かつデータを入出力します。

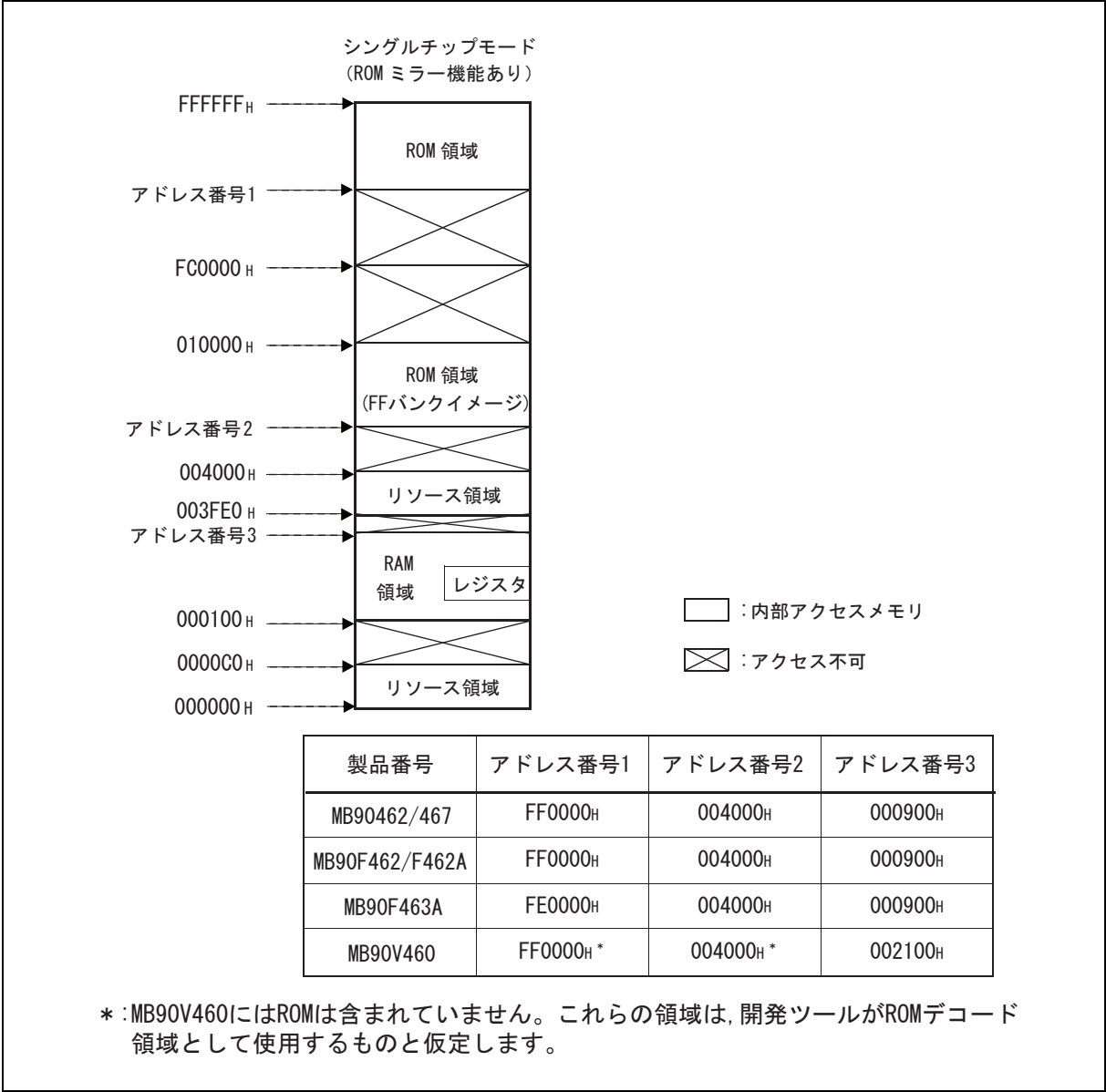
### 3.3 メモリマップ

MB90460/465 シリーズの各製品のメモリマップを示します。

#### ■ メモリマップ

MB90460/465 シリーズのメモリマップを図 3.3-1 に示します。

図 3.3-1 メモリマップ



< 注意事項 > ROM ミラー機能レジスタを設定した場合，00 バンクの上位側（"004000<sub>H</sub> ~ 00FFFF<sub>H</sub>"）に FF バンクの上位側（"FF0000<sub>H</sub> ~ FFFFFF<sub>H</sub>"）のデータがミラーイメージで見えるようになります。  
ROM ミラー機能の設定は，「第 22 章 ROM ミラー機能選択モジュール」をご参照ください。

---

< 参考 >      ROM ミラー機能は、C コンパイラのスモールモデルを使用するための機能です。FF バンクの下位 16 ビットアドレスは、00 バンクの下位 16 ビットアドレスと同じになります。ただし、FF バンクの ROM 領域は 48K バイトを超えますので、00 バンクには ROM 領域の全データをミラーイメージで見せることはできません。C コンパイラのスモールモデルをご使用の場合は、データテーブルを "FF4000<sub>H</sub> ~ FFFFFFF<sub>H</sub>" に格納しておくことで "004000<sub>H</sub> ~ 00FFFF<sub>H</sub>" にミラーイメージでデータテーブルを見せることができます。したがって、ポインタで far 指定を宣言することなく、ROM 領域内のデータテーブルを参照できます。

---

## 3.4 アドレス指定

アドレス指定には、リニア方式とバンク方式があります。

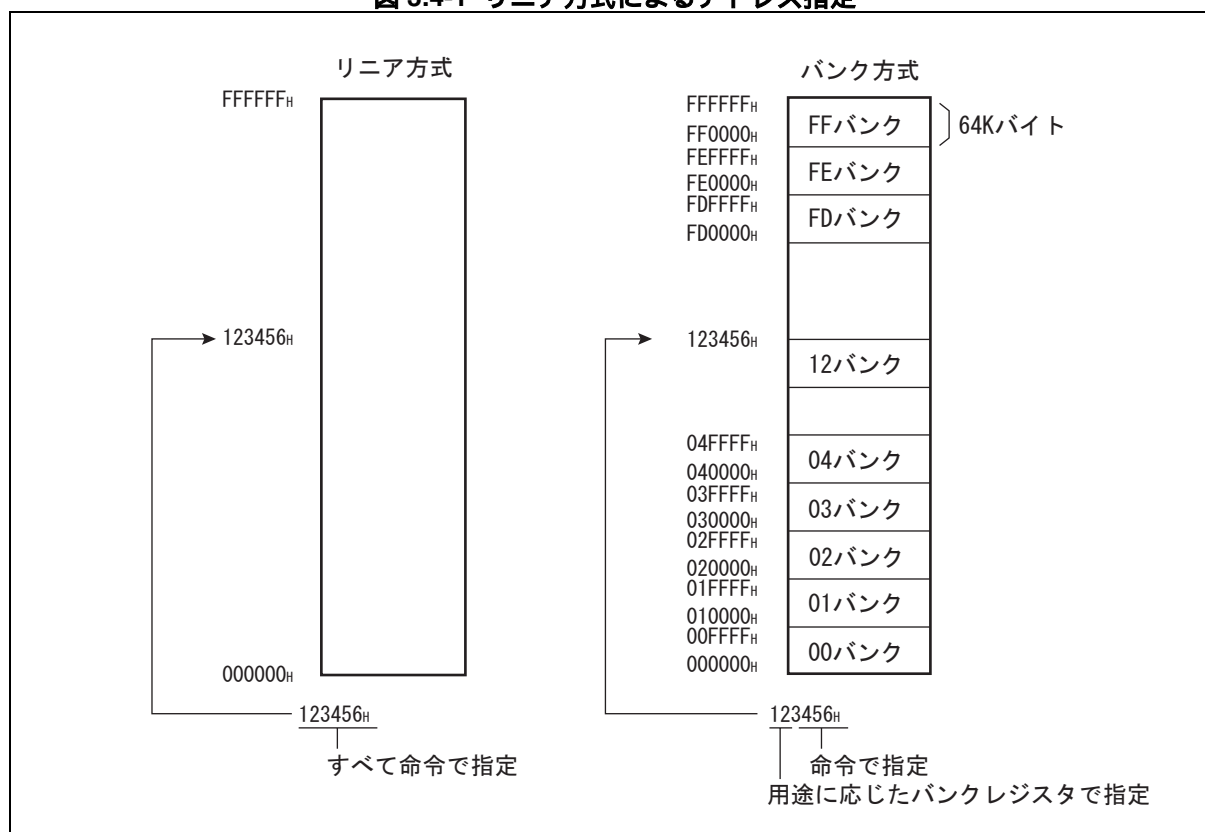
リニア方式は、16M バイトの空間において連続している 24 ビットアドレスに対し、命令により直接アドレスを指定します。

バンク方式は、64K バイトごとの 256 バンクに分割された 16M バイト空間において、上位 8 ビットアドレスをバンクレジスタにより指定し、下位 16 ビットアドレスを命令により直接指定します。

F<sup>2</sup>MC-16LX ファミリは、アドレス生成方式により命令の使用方法が異なります。

### ■ リニア方式とバンク方式のメモリ指定

図 3.4-1 リニア方式によるアドレス指定



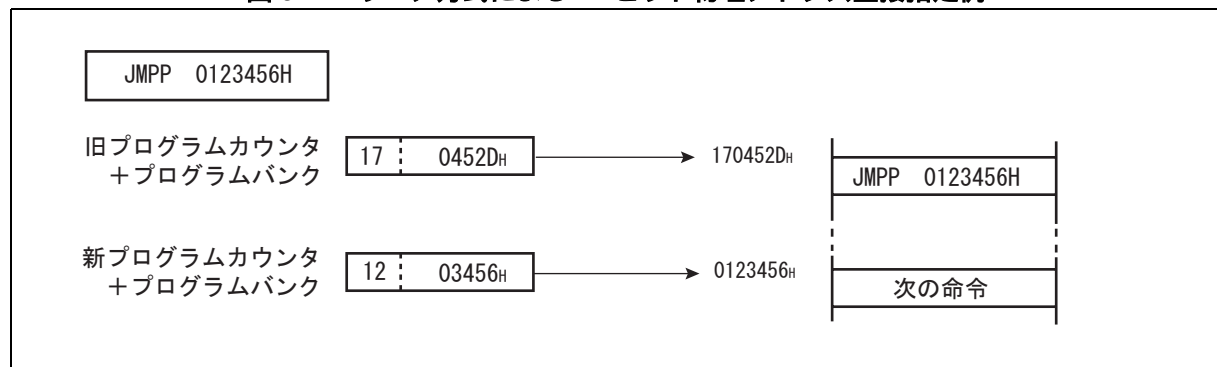


### 3.4.1 リニア方式によるアドレス指定

リニア方式によるアドレス指定には、さらにオペランドで 24 ビットのアドレスを直接指定する方法と、32 ビット汎用レジスタの下位 24 ビットをアドレスとして引用指定（間接指定）する方法があります。

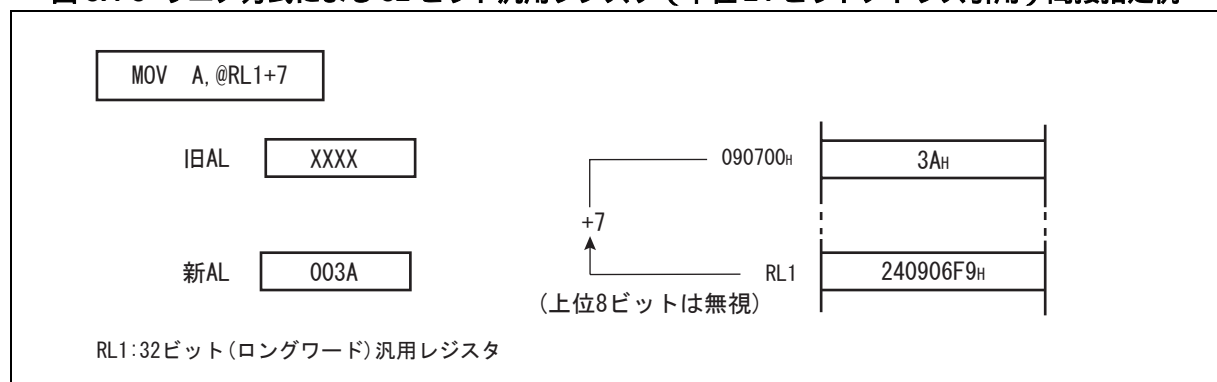
#### ■ オペランドによる 24 ビットアドレス直接指定

図 3.4-2 リニア方式による 24 ビット物理アドレス直接指定例



#### ■ 32 ビット汎用レジスタの下位 24 ビットアドレス引用指定（間接指定）

図 3.4-3 リニア方式による 32 ビット汎用レジスタ（下位 24 ビットアドレス引用）間接指定例



### 3.4.2 バンク方式によるアドレス指定

バンク方式は、16M バイトのメモリ空間を 64K バイトごとの 256 バンクに分割し、これらのバンクアドレスを、アドレスの上位 8 ビットを示すバンクレジスタによって指定します。下位 16 ビットアドレスは命令により直接指定します。

バンクレジスタには、用途別に以下に示す 5 種類があります。

- プログラムバンクレジスタ (PCB)
- データバンクレジスタ (DTB)
- 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)
- アディショナルバンクレジスタ (ADB)

#### ■ バンクレジスタとアクセス空間

各バンクレジスタのアクセス空間と主な用途について表 3.4-1 に示します。

表 3.4-1 各バンクレジスタのアクセス空間と主な用途

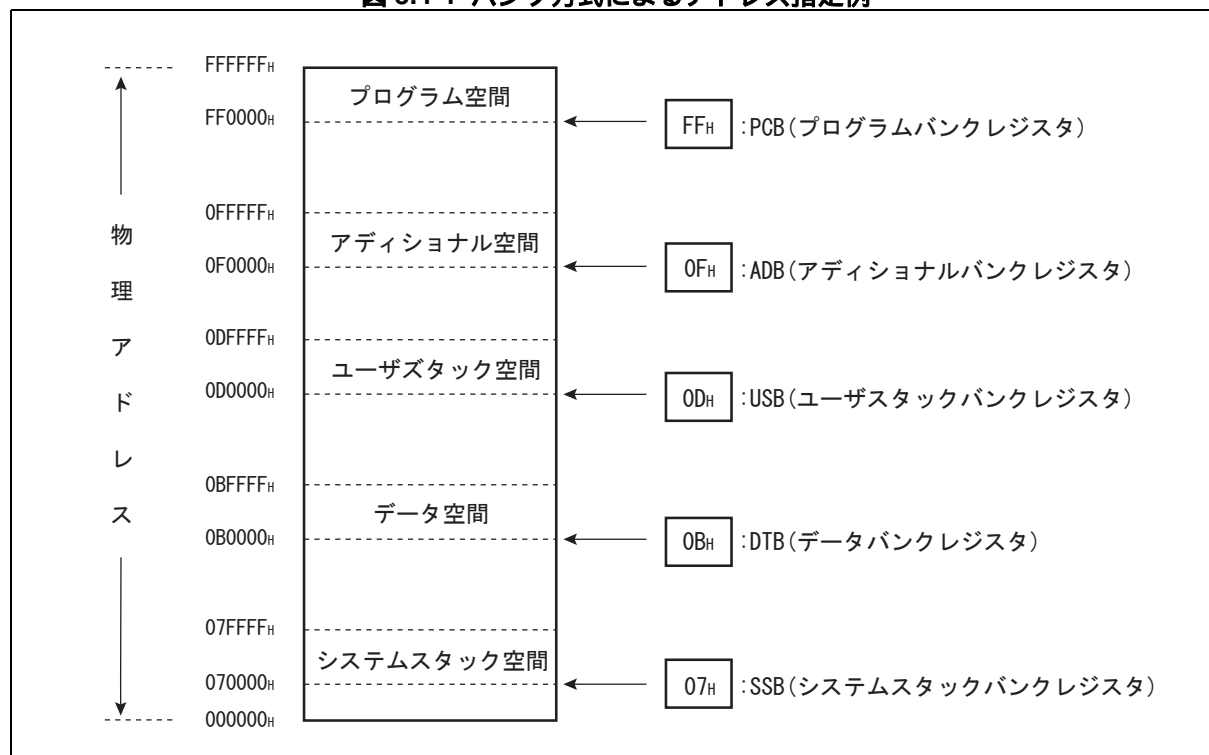
バンクレジスタ名	アクセス空間	主な用途	リセット時初期値
プログラムバンクレジスタ (PCB)	プログラム空間	命令コード、ベクタテーブル、即値データの格納用	FF <sub>H</sub>
データバンクレジスタ (DTB)	データ空間	読み書き可能なデータを格納。内外周辺の制御レジスタやデータレジスタへのアクセス用	00 <sub>H</sub>
ユーザスタックバンクレジスタ (USB)	スタック空間	PUSH/POP 命令や、割込みのレジスタ退避などのスタックアクセス用。コンディションレジスタ内のスタックフラグ (CCR:S) が "1" の場合は SSB レジスタを、"0" の場合は USB レジスタを使用 *	00 <sub>H</sub>
システムスタックバンクレジスタ (SSB)*			00 <sub>H</sub>
アディショナルバンクレジスタ (ADB)	アディショナル空間	データ (DT) 空間でオーバフローしたデータの格納用	00 <sub>H</sub>

\*: 割込み時のスタックには、必ず SSB レジスタが使用されます。

バンクレジスタの詳細は、「3.7.9 バンクレジスタ (PCB, DTB, USB, SSB, ADB)」をご参照ください。

バンク方式によるアドレス指定例を図 3.4-4 に示します。

図 3.4-4 バンク方式によるアドレス指定例



### ■ バンク方式によるアドレス指定とデフォルト空間

命令のコード効率向上のために、各命令には、アドレス指定方式ごとに表 3.4-2 に示すようなデフォルト空間が決められています。デフォルト以外の空間を使用したい場合は、各バンクに対応しているプリフィックスコードを、命令に先行して設定してください。これにより、プリフィックスコードに対応したバンク空間へアクセスできます。プリフィックスコードの詳細は、「3.9 プリフィックスコード」をご参照ください。

表 3.4-2 アドレス指定とデフォルト空間

デフォルト空間	アドレス指定
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, dir を用いたアドレス指定
スタック空間	PUSHW, POPW, @RW3, @RW7 を用いたアドレス指定
アディショナル空間	@RW2, @RW6 を用いたアドレス指定

### 3.5 多バイト長データのメモリ配置

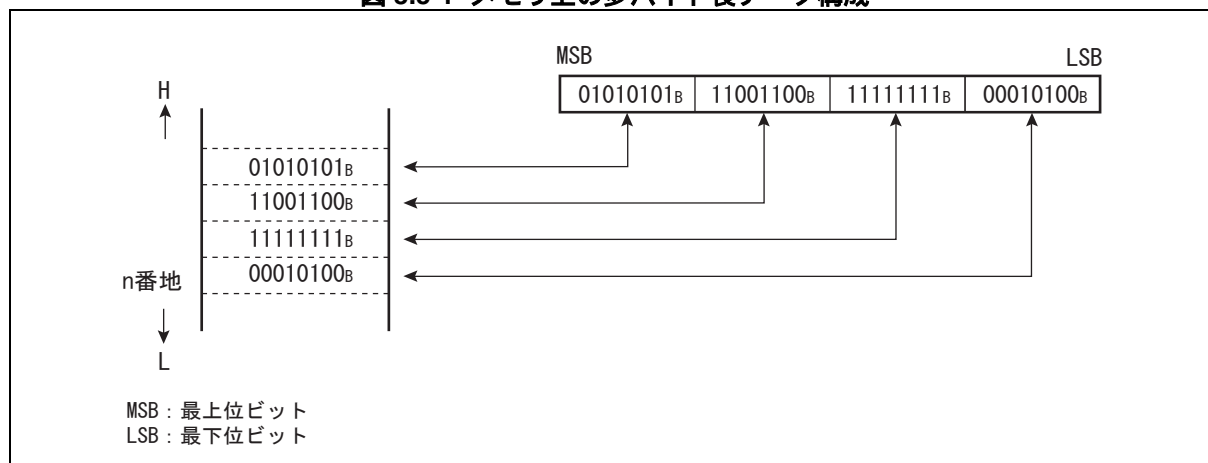
多バイト長データは、下位アドレスから順にメモリへ書き込まれます。多バイト長データが 32 ビットであれば、下位 16 ビットの次に上位 16 ビットの順に転送されます。ただし、下位データを書き込んだ直後に外部リセット信号が入力された場合、上位データが書き込まれない場合があります。

#### ■ メモリ上の多バイト長データの格納状態

データは下位 8 ビットが  $n$  番地に、次に  $n + 1$  番地、 $n + 2$  番地、 $n + 3$  番地の順に配置されます。

メモリ上における多バイト長データの構成を図 3.5-1 に示します。

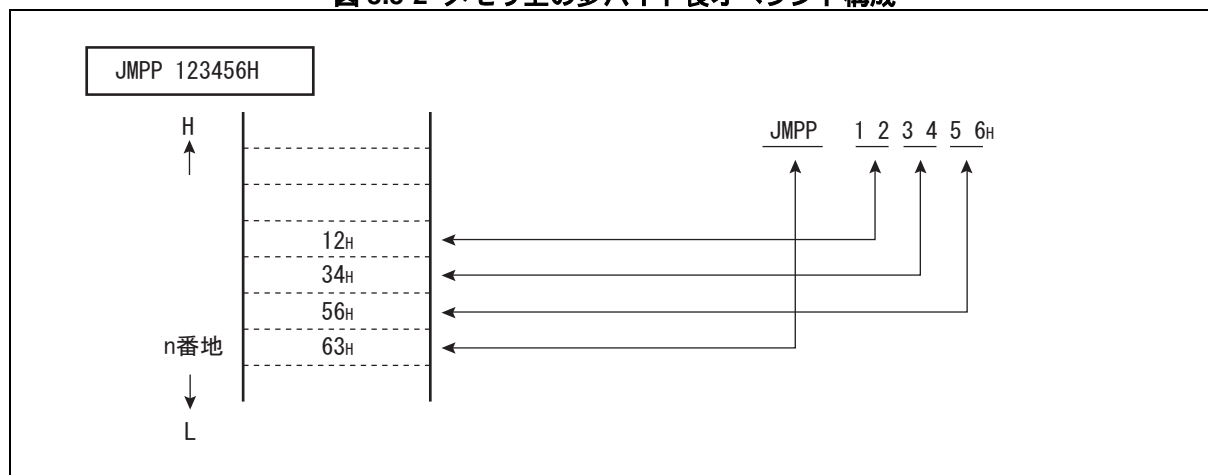
図 3.5-1 メモリ上の多バイト長データ構成



#### ■ 多バイト長オペランドの構成

メモリ上における多バイト長オペランドの構成を図 3.5-2 に示します。

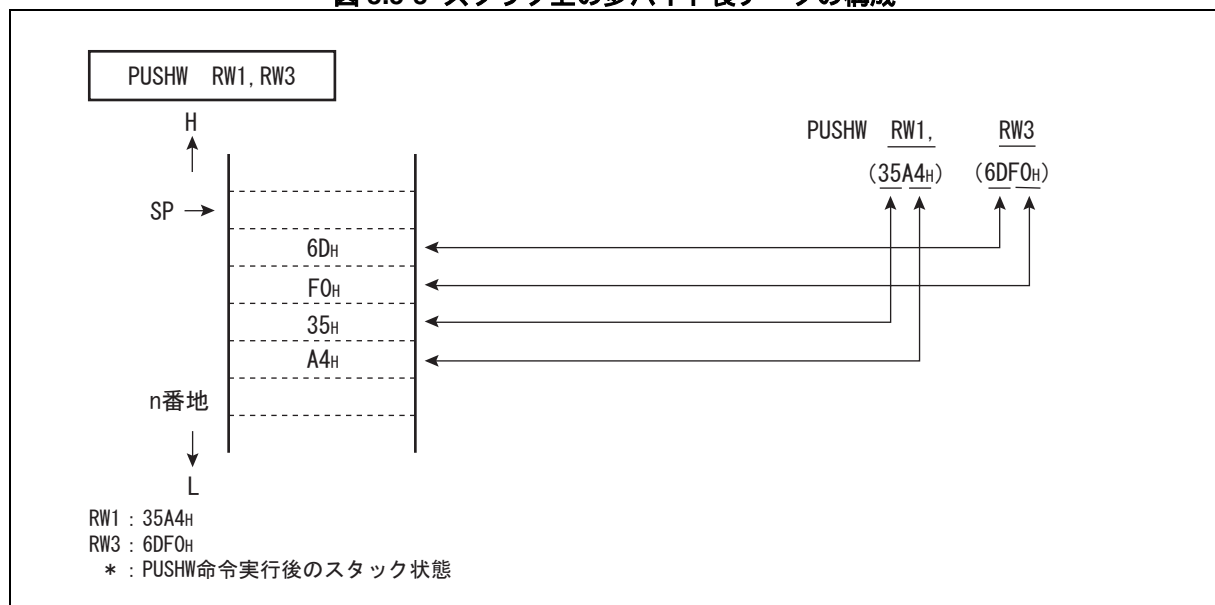
図 3.5-2 メモリ上の多バイト長オペランド構成



## ■ スタック上の多バイト長データの構成

スタック上の多バイト長データ構成を図 3.5-3 に示します。

図 3.5-3 スタック上の多バイト長データの構成

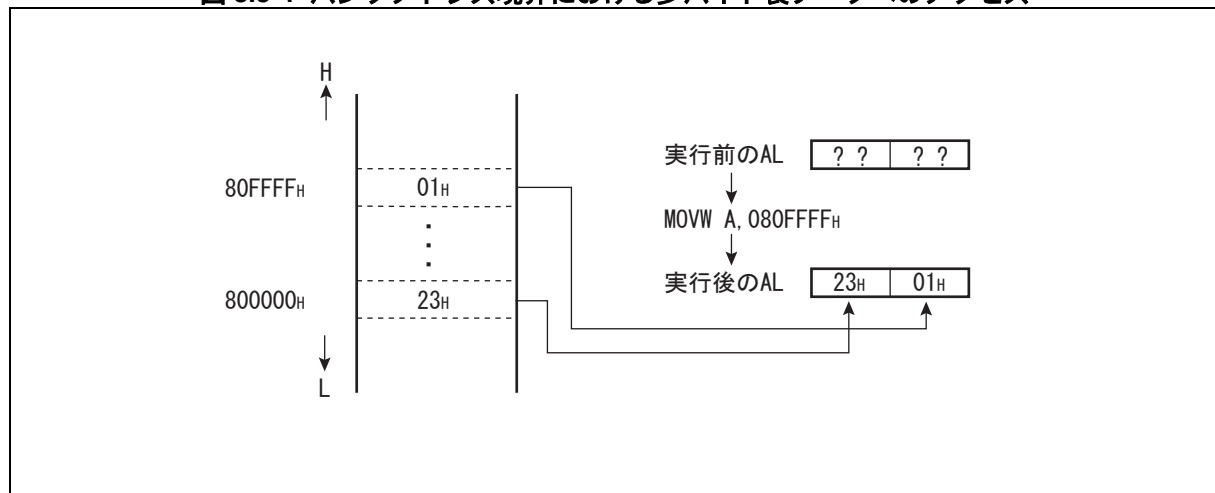


## ■ 多バイト長データへのアクセス

多バイト長データのアクセスは、バンク内が基本です。多バイト長データへアクセスする命令を実行した場合は、"FFFF<sub>H</sub>" 番地の次のアドレスは同じバンクの "0000<sub>H</sub>" 番地になります。

バンクアドレス境界における多バイト長データへのアクセスを図 3.5-4 に示します。

図 3.5-4 バンクアドレス境界における多バイト長データへのアクセス



## 3.6 レジスタ

F<sup>2</sup>MC-16LX ファミリのレジスタには、CPU に内蔵されている専用レジスタと、内蔵 RAM 上に配置されている汎用レジスタがあります。

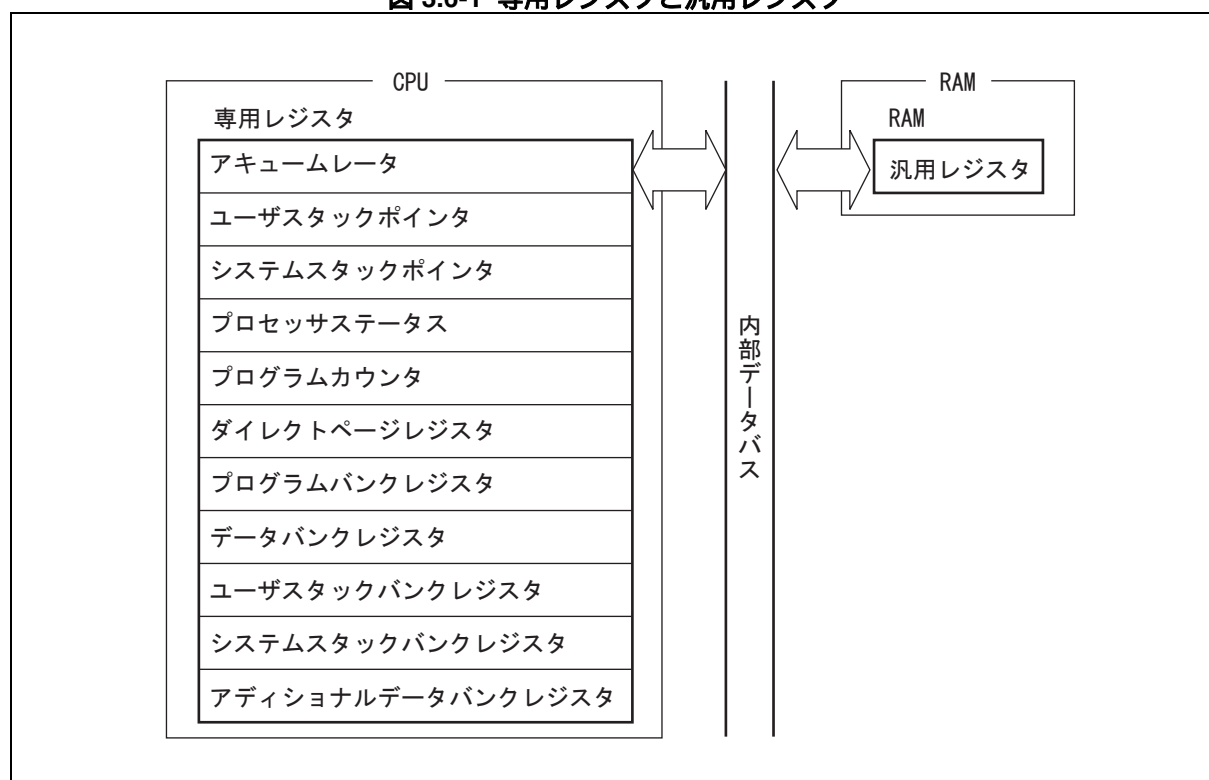
### ■ 専用レジスタと汎用レジスタ

専用レジスタは、CPU に内蔵されています。

汎用レジスタは、CPU のアドレス空間上に RAM と共存しています。

専用レジスタと汎用レジスタを図 3.6-1 に示します。

図 3.6-1 専用レジスタと汎用レジスタ



## 3.7 専用レジスタ

専用レジスタは、以下に示す 11 種類のレジスタで構成されています。

- アキュムレータ (A)
- ユーザスタックポインタ (USP)
- システムスタックポインタ (SSP)
- プロセッサステータス (PS)
- プログラムカウンタ (PC)
- ダイレクトページレジスタ (DPR)
- プログラムバンクレジスタ (PCB)
- データバンクレジスタ (DTB)
- 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)
- アディショナルデータバンクレジスタ (ADB)

### ■ 専用レジスタの構成

専用レジスタの構成を図 3.7-1 に示します。

図 3.7-1 専用レジスタの構成

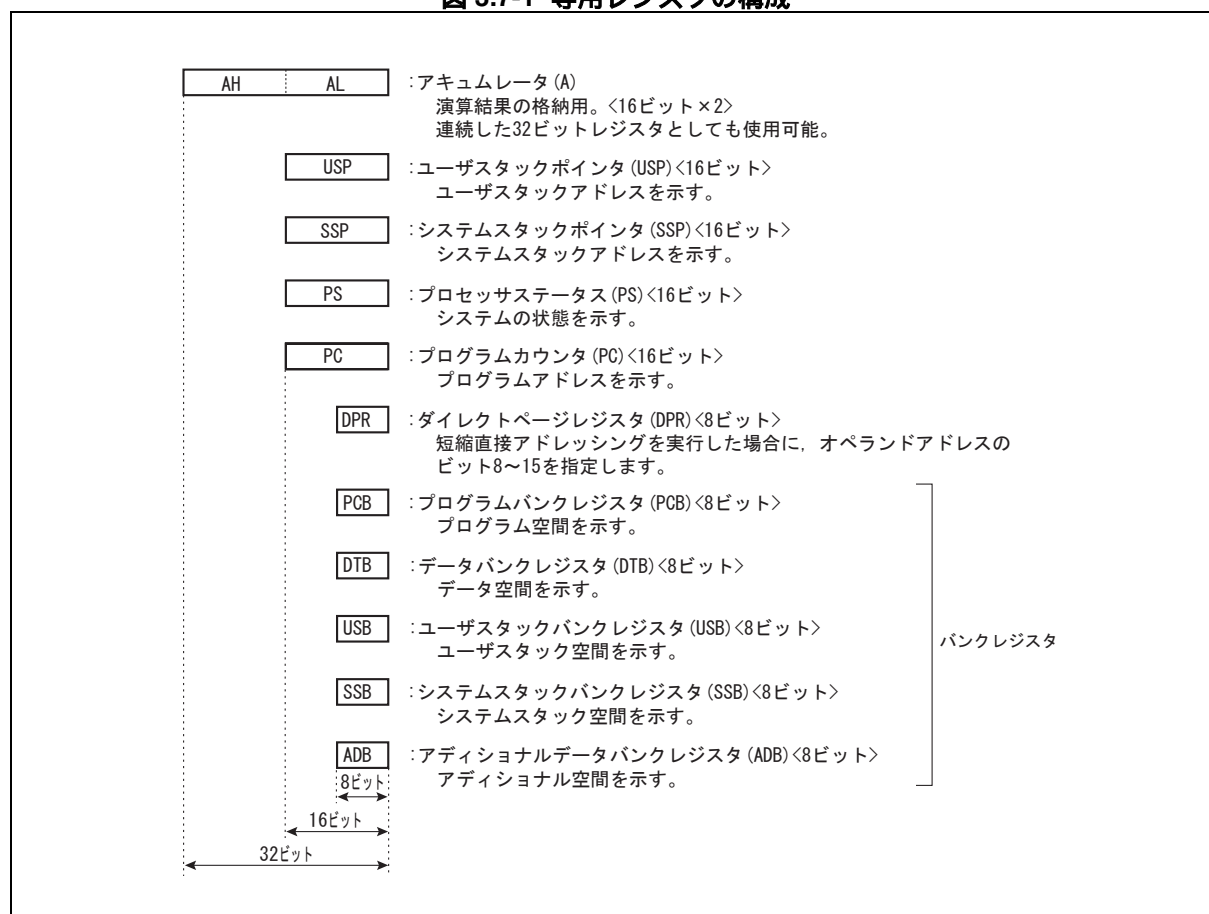


表 3.7-1 専用レジスタの初期値

専用レジスタ	初期値
アキュムレータ ( A )	不定
ユーザスタックポインタ ( USP )	不定
システムスタックポインタ ( SSP )	不定
プロセッサステータス ( PS )	<div><div><div>bit 15 ~ 13 12 ~ 8 7 ~ 0</div><div>PS<div><div>ILM</div><div>RP</div><div>CCR</div></div><div>0 0 0 0 0 0 0 0 x 0 1 x x x x x x</div></div></div><div>リセットベクタ中の値 (FFFFDC<sub>H</sub>, FFFFDD<sub>H</sub>のデータ)</div></div>
プログラムカウンタ ( PC )	リセットベクタ中の値 (FFFFDC <sub>H</sub> , FFFFDD <sub>H</sub> のデータ)
ダイレクトページレジスタ ( DPR )	01 <sub>H</sub>
プログラムバンクレジスタ ( PCB )	リセットベクタ中の値 (FFFFDE <sub>H</sub> のデータ)
データバンクレジスタ ( DTB )	00 <sub>H</sub>
ユーザスタックバンクレジスタ ( USB )	00 <sub>H</sub>
システムスタックバンクレジスタ ( SSB )	00 <sub>H</sub>
アディショナルデータバンクレジスタ ( ADB )	00 <sub>H</sub>



### 3.7.1 アキュムレータ (A)

アキュムレータ (A) は、二つの 16 ビット長演算用レジスタ (AH/AL) から構成されています。演算結果やデータの一時記憶に使用されるレジスタです。

アキュムレータ (A) は、32/16/8 ビットのレジスタとして使用できます。メモリとほかのレジスタ間、あるいは上位側の 16 ビット長演算レジスタ (AH) と下位側の 16 ビット長演算レジスタ (AL) 間で演算できます。また、ワード長以下 (16 ビット長以下) のデータを下位側の 16 ビット長演算レジスタ (AL) へ転送した場合、それまでに存在した下位側の 16 ビット長演算レジスタ (AL) のデータが、上位側の 16 ビット長演算レジスタ (AH) へ転送されるデータ保持機能があります (一部の命令ではこの "データ保持機能" は無効となります)。

#### ■ アキュムレータ (A)

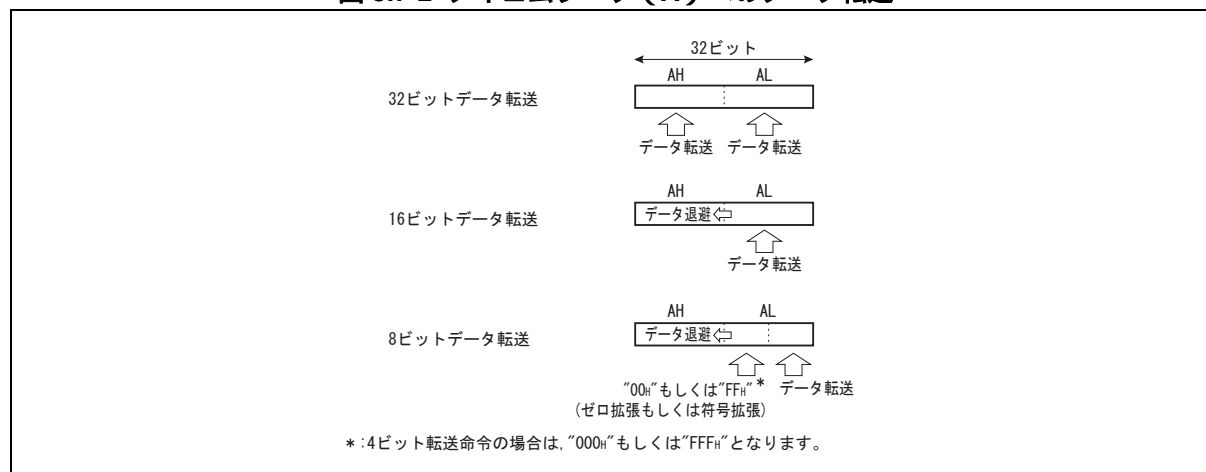
##### ● アキュムレータへのデータ転送

アキュムレータは、32 ビット長 (ロングワード)、16 ビット長 (ワード)、8 ビット長 (バイト) のデータを処理します。例外として、4 ビットデータ転送命令 (MOVN) も 8 ビットデータと同様に処理します。

- 32 ビットデータの処理時は、上位演算用レジスタ (AH) と下位演算用レジスタ (AL) が連結して使用されます。
- 16 ビットデータや 8 ビットデータの処理時は、下位演算用レジスタ (AL) が使用されます。上位演算用レジスタ (AH) は、その処理前まで存在した下位演算用レジスタ (AL) のデータを保持します。
- 下位演算用レジスタ (AL) へバイト長以下 (8 ビット長以下) のデータが転送された場合は、そのデータは 16 ビット長へ符号拡張またはゼロ拡張され、下位演算用レジスタ (AL) へ格納されます。また、下位演算用レジスタ (AL) へ格納されたデータは、16 ビットデータまたは 8 ビットデータとして処理できます。

具体的な転送例を図 3.7-2 ~ 図 3.7-6 に示します。

図 3.7-2 アキュムレータ (A) へのデータ転送



● アキュムレータ (A) のバイト処理算術演算

バイト処理の算術演算命令を下位演算用レジスタ (AL) に対して実行した場合、下位演算用レジスタ (AL) の上位 8 ビットは無視されます ("00<sub>H</sub>" が挿入されます)。

● アキュムレータ (A) の初期値

リセット後の初期値は不定です。

図 3.7-3 アキュムレータ (A) の AL-AH レジスタ間の転送例 (8 ビット即値, ゼロ拡張)

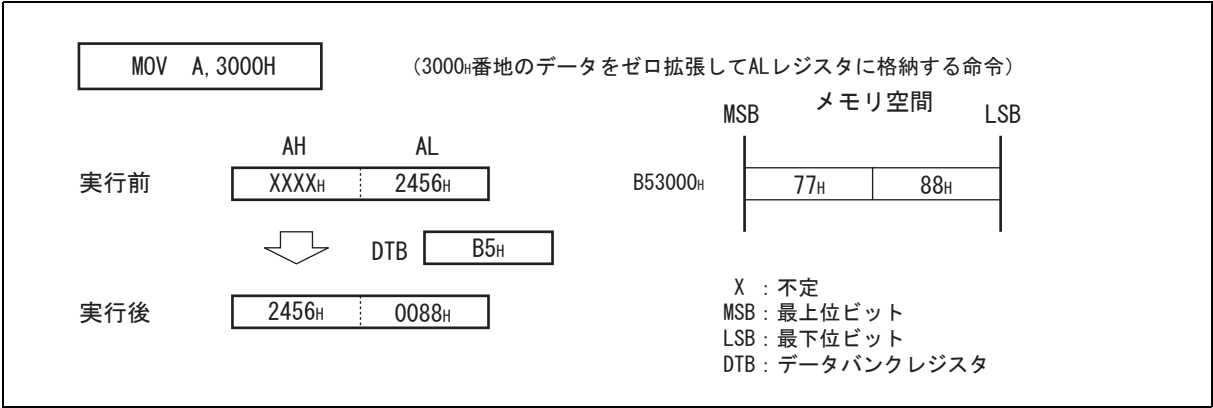


図 3.7-5 アキュムレータ (A) への 32 ビットデータ転送例 (レジスタ間接)

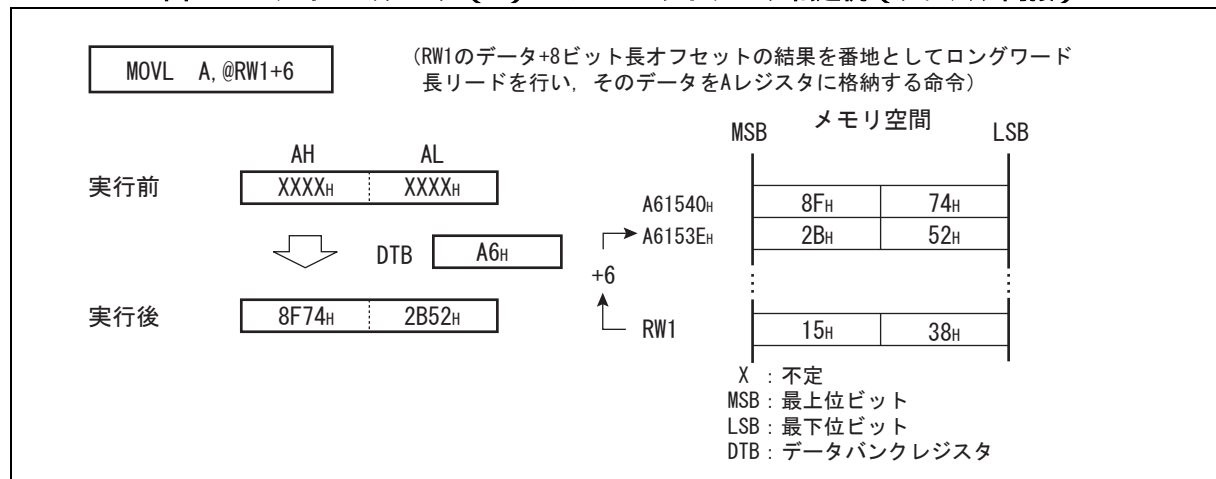
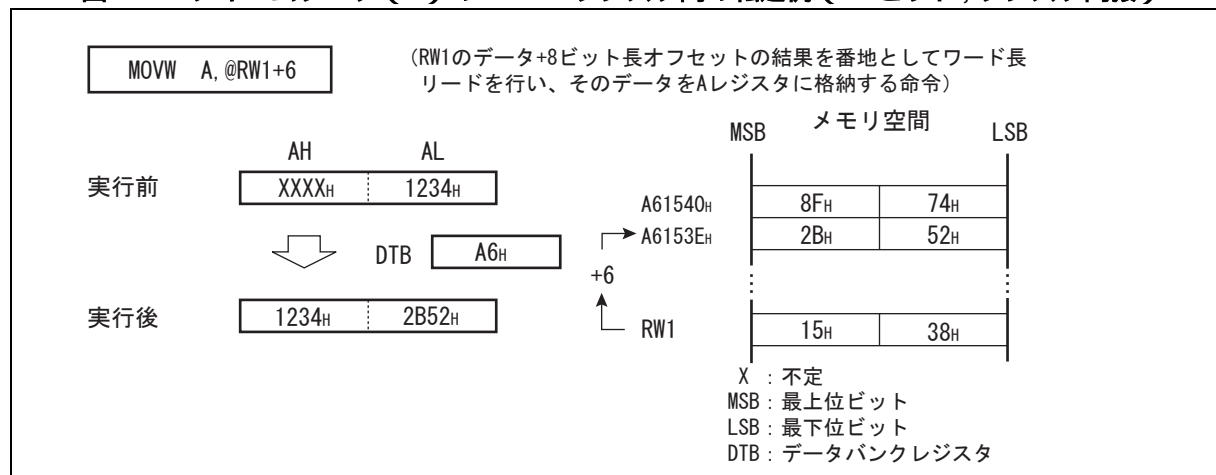


図 3.7-6 アキュムレータ (A) の AL-AH レジスタ間の転送例 (16 ビット, レジスタ間接)



3.7.2      スタックポインタ（USP, SSP）

スタックポインタには、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）があります。これらのレジスタは、PUSH 命令や POP 命令およびサブルーチンを実行する場合に、データの退避先や復帰先のメモリアドレスを示す 24 ビットのレジスタです。スタックアドレス（24 ビット）の上位 8 ビットは、ユーザスタックバンクレジスタ（USB）もしくはシステムスタックバンクレジスタ（SSB）で指定されます。

コンディションコードレジスタ（CCR）の S フラグが "0" の場合は、ユーザスタックポインタ（USP）およびユーザスタックバンクレジスタ（USB）が有効になります。S フラグが "1" の場合は、システムスタックポインタ（SSP）およびシステムスタックバンクレジスタ（SSB）が有効になります。

■ スタックの設定

F<sup>2</sup>MC-16LX ファミリでは、システムスタックとユーザスタックの 2 種類のスタックを使用できます。スタックアドレスは、プロセッサステータスレジスタ（PS:CCR）内の S フラグによって、表 3.7-2 のように決定されます。

表 3.7-2 スタックアドレスの指定

S フラグ	スタックアドレス（24 ビット）	
	上位 8 ビット	下位 16 ビット
0	ユーザスタックバンクレジスタ（USB）	ユーザスタックポインタ（USP）
1*	システムスタックバンクレジスタ（SSB）	システムスタックポインタ（SSP）

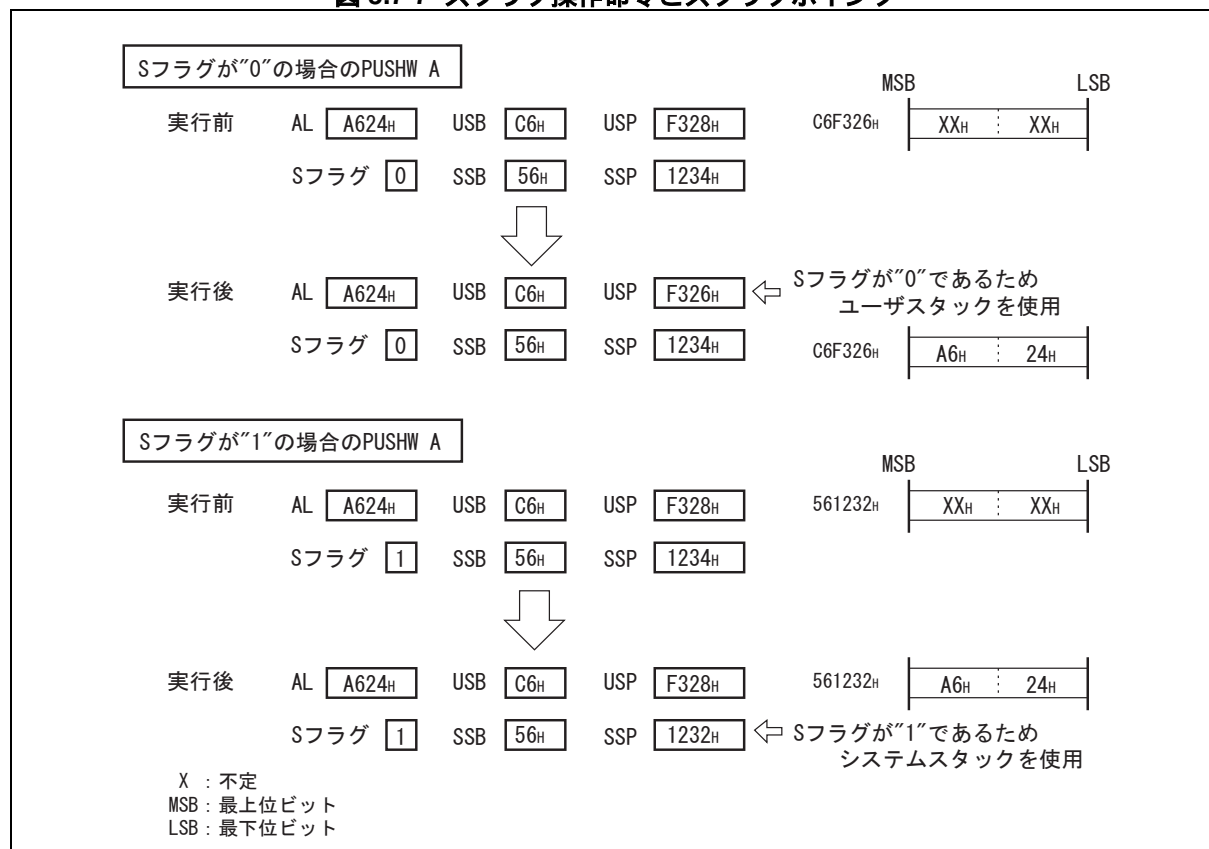
\*: 初期値

リセットによって、S フラグは "1" に初期化されます。したがって、初期設定ではシステムスタックが使用されます。ただし、割込みルーチンのスタック操作時はシステムスタックが使用され、割込みルーチン以外のスタック操作時にはユーザスタックが使用されます。スタック空間を分割しないのであれば、システムスタックを使用してください。

< 注意事項 >      割込みが受け付けられた場合、S フラグは "1" にセットされ、システムスタックが使用されます。

スタック操作命令とスタックポインタを図 3.7-7 に示します。

図 3.7-7 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スタックアドレスを設定する場合は、偶数アドレスを設定してください。  
奇数アドレスを設定した場合は、ワードアクセスが2回に分割されます。
  - USP レジスタおよび SSP レジスタの初期値は不定です。

## ■ システムスタックポインタ (SSP)

システムスタックポインタ (SSP) を設定する場合は、コンディションコードレジスタ (CCR) 内の S フラグに “1” を設定してください。S フラグに “1” を設定した場合、スタックを操作する際に使用されるアドレスの上位8ビットは、システムスタックバンクレジスタ (SSB) により示されます。

コンディションコードレジスタ (CCR) の詳細は、「3.7.4 コンディションコードレジスタ (PS:CCR)」を、システムスタックバンクレジスタ (SSB) の詳細は、「3.7.9 バンクレジスタ (PCB, DTB, USB, SSB, ADB)」をご参照ください。

## ■ ユーザスタックポインタ (USP)

ユーザスタックポインタ (USP) を設定する場合は、コンディションコードレジスタ (CCR) 内の S フラグに “0” を設定してください。S フラグに “0” を設定した場合、スタックを操作する際に使用されるアドレスの上位8ビットは、ユーザスタックバンクレジスタ (USB) により示されます。

コンディションコードレジスタ (CCR) の詳細は、「3.7.4 コンディションコードレジスタ (PS:CCR)」を、システムスタックバンクレジスタ (SSB) の詳細は、「3.7.9 バンクレジスタ (PCB, DTB, USB, SSB, ADB)」をご参照ください。

### 3.7.3 プロセッサステータス (PS)

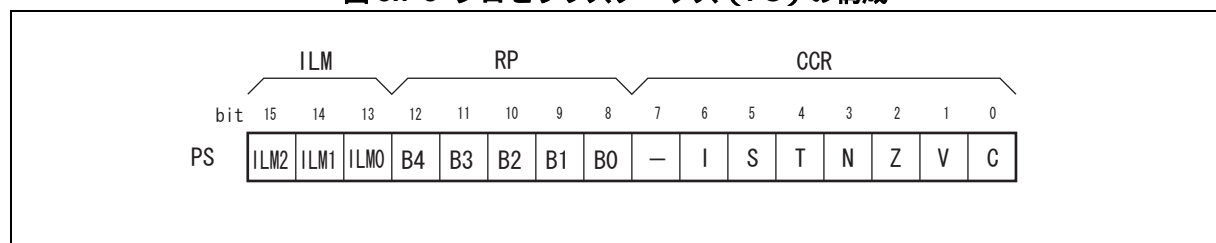
プロセッサステータス (PS) は、CPU を制御するビットと CPU の状態を示すビットから構成されています。プロセッサステータス (PS) は、以下に示す 3 つのレジスタで構成されています。

- 割込みレベルマスクレジスタ (ILM)
- レジスタバンクポインタ (RP)
- コンディションコードレジスタ (CCR)

#### ■ プロセッサステータス (PS) の構成

プロセッサステータス (PS) の構成を図 3.7-8 に示します。

図 3.7-8 プロセッサステータス (PS) の構成



#### ● コンディションコードレジスタ (CCR)

命令実行結果や割込み出力によって "1" が設定または "0" にクリアされる各種フラグから構成されています。

各種フラグの詳細は、「3.7.4 コンディションコードレジスタ (PS:CCR)」をご参照ください。

#### ● レジスタバンクポインタ (RP)

RAM 領域の中で汎用レジスタとして使用するメモリブロック (レジスタバンク) の先頭アドレスを設定するポインタです。

汎用レジスタは 32 バンクで構成されています。レジスタバンクポインタ (RP) に "00<sub>H</sub> ~ 1F<sub>H</sub>" の値を設定してバンクを指定してください。

設定方法および詳細は、「3.7.5 レジスタバンクポインタ (PS:RP)」をご参照ください。

#### ● 割込みレベルマスクレジスタ (ILM)

CPU が現在受付けている割込みのレベルを示します。またリソースごとの割込み要求に対応して設定されている割込み制御レジスタ (ICR00 ~ ICR15) の割込みレベル設定ビット (ICR:IL0 ~ IL2) 値と比較します。

設定方法および詳細は、「3.7.6 割込みレベルマスクレジスタ (PS:ILM)」をご参照ください。

### 3.7.4 コンディションコードレジスタ (PS:CCR)

コンディションコードレジスタ (CCR) は、以下に示す 8 ビットで構成されています。

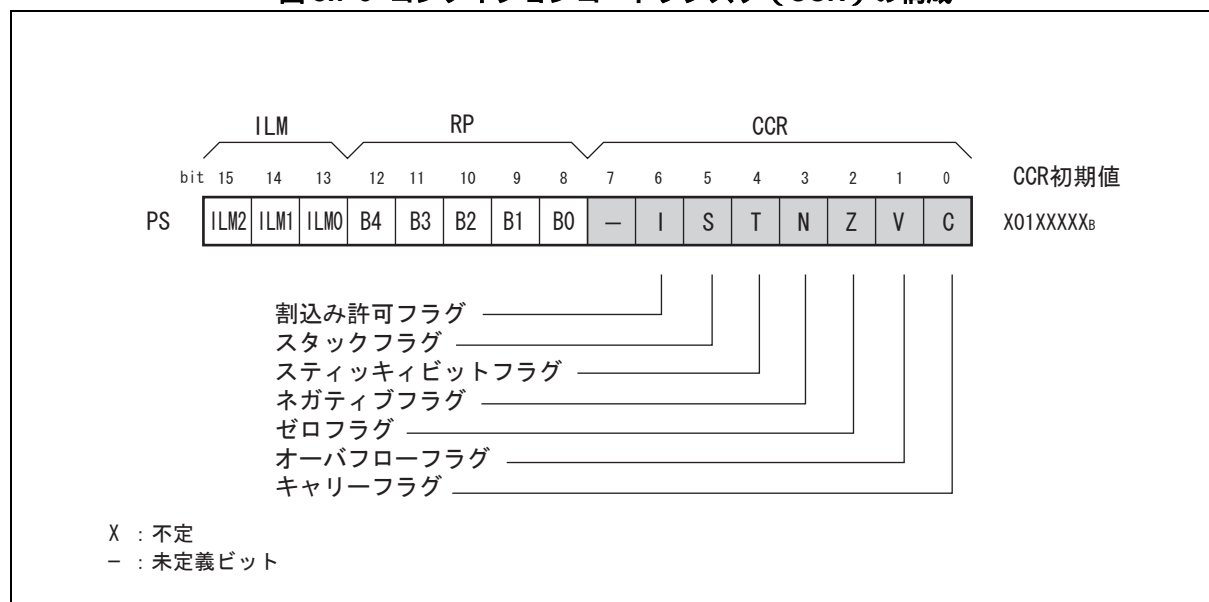
- 演算結果や転送データを示すビット
- 割込み要求の受け付けを制御するビット

#### ■ コンディションコードレジスタ (CCR) の構成

命令実行時のコンディションコードレジスタ (CCR) の状態については、『F<sup>2</sup>MC-16LX ファミリ プログラミングマニュアル』をご参照ください。

コンディションコードレジスタ (CCR) の構成を図 3.7-9 に示します。

図 3.7-9 コンディションコードレジスタ (CCR) の構成



#### ● 割込み許可フラグ (I)

割込み許可フラグ (I) に "1" が設定された場合はソフトウェア割込み以外の割込み要求を許可し、割込み許可フラグ (I) が "0" にクリアされた場合はソフトウェア割込み以外の割込み要求を禁止します。外部リセットおよびソフトウェアリセットにより "0" にクリアされます。

#### ● スタックフラグ (S)

スタック操作に用いられるポインタを示します。スタックフラグ (S) が "0" にクリアされた場合はユーザスタックポインタ (USP) が有効になり、スタックフラグ (S) に "1" が設定された場合はシステムスタックポインタ (SSP) が有効になります。割込みが受け付けられた場合または外部リセットおよびソフトウェアリセットがアサートされた場合は、"1" が設定されます。

スタックポインタの詳細は、「3.7.2 スタックポインタ (USP, SSP)」をご参照ください。

- スティッキビットフラグ (T)

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、キャリーからシフトアウトされたデータ内に "1" があればスティッキビットフラグ (T) に "1" が設定され、"1" がなければスティッキビットフラグ (T) は "0" にクリアされます。また、シフト量がゼロの場合でも "0" にクリアされます。

- ネガティブフラグ (N)

演算結果が格納される汎用レジスタ (RL0 ~ RL3) の最上位ビット (MSB) が "1" の場合は、ネガティブフラグ (N) に "1" が設定されます。演算結果が格納される汎用レジスタ (RL0-RL3) の最上位ビット (MSB) が "0" の場合は、ネガティブフラグ (N) が "0" にクリアされます。

汎用レジスタの詳細は、「3.8 汎用レジスタ」をご参照ください。

- ゼロフラグ (Z)

演算結果が格納される汎用レジスタ (RL0 ~ RL3) の値が "0000<sub>H</sub>" の場合は、ゼロフラグ (Z) に "1" が設定されます。演算結果が格納される汎用レジスタ (RL0 ~ RL3) の値が "0000<sub>H</sub>" でない場合は、ゼロフラグ (Z) は "0" にクリアされます。

汎用レジスタの詳細は、「3.8 汎用レジスタ」をご参照ください。

- オーバフローフラグ (V)

演算の結果、符号付き数値としてオーバーフローが発生した場合は、オーバーフローフラグ (V) に "1" が設定されます。オーバーフローが発生しなかった場合は、オーバーフローフラグ (V) は "0" にクリアされます。

- キャリーフラグ (C)

演算の結果、最上位ビットからの桁上がりまたは最上位ビットへの桁下がりが発生した場合は、キャリーフラグ (C) に "1" が設定されます。発生しなかった場合は、キャリーフラグ (C) は "0" にクリアされます。



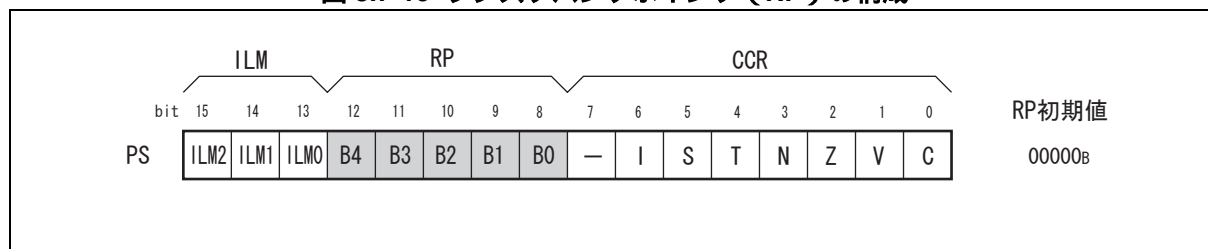
### 3.7.5 レジスタバンクポインタ (PS:RP)

レジスタバンクポインタ (RP) は、現在使用している汎用レジスタバンクの先頭アドレスを示す 5 ビットのレジスタです。

#### ■ レジスタバンクポインタ (RP)

レジスタバンクポインタ (RP) を図 3.7-10 に示します。

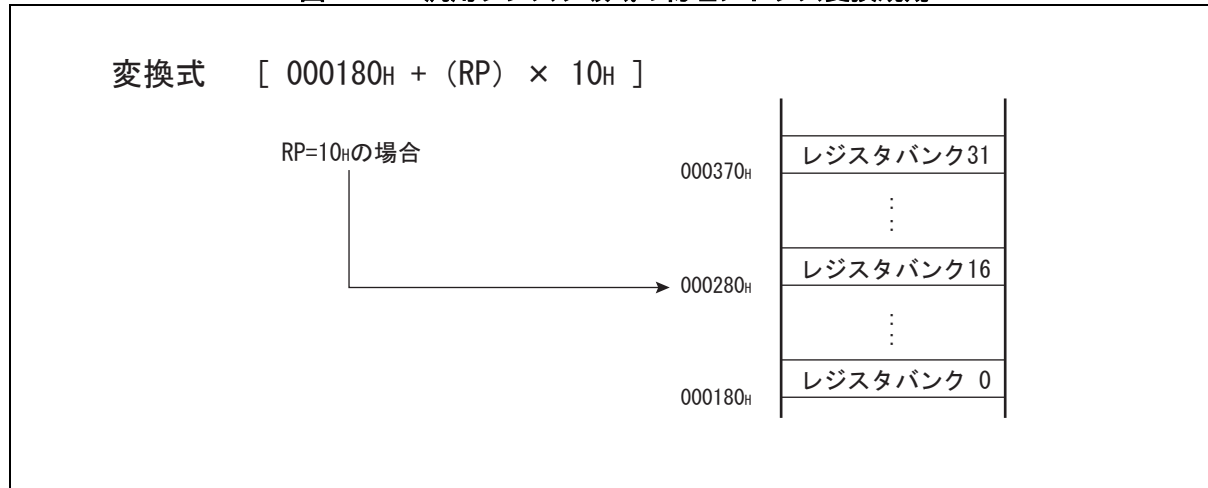
図 3.7-10 レジスタバンクポインタ (RP) の構成



#### ■ 汎用レジスタ領域とレジスタバンクポインタ (RP)

レジスタバンクポインタ (RP) は、F<sup>2</sup>MC-16LX ファミリが持っている汎用レジスタと、内部 RAM のアドレスとの関係を示すポインタです。レジスタバンクポインタ (RP) とアドレスは、図 3.7-11 に示す変換規則の関係です。

図 3.7-11 汎用レジスタ領域の物理アドレス変換規則



- アセンブラ命令では、レジスタバンクポインタ (RP) へ転送する 8 ビットの即値転送命令を使用できます。ただし、下位 5 ビットのデータが有効となります。

### 3.7.6 割込みレベルマスクレジスタ (PS:ILM)

割込みレベルマスクレジスタ (ILM) は、現在設定されている割込みレベルを示します。

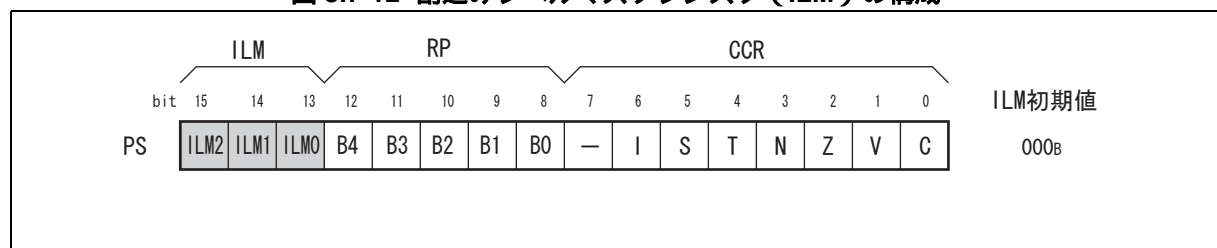
割込みレベルマスクレジスタ (ILM) には、受付け可能な割込みレベルを設定できます。割込みレベルマスクレジスタ (ILM) に設定された割込みレベル値よりも弱い割込みレベル値の割込みは受付けません。

#### ■ 割込みレベルマスクレジスタ (ILM)

割込みの詳細は、「第7章 割込み」をご参照ください。

割込みレベルマスクレジスタ (ILM) の構成を図 3.7-12 に示します。

図 3.7-12 割込みレベルマスクレジスタ (ILM) の構成



割込みレベルマスクレジスタ (ILM) は、現在設定されている割込みレベルを示します。割込みレベルマスクレジスタ (ILM) には、受付け可能な割込みレベルを設定できます。割込みレベルマスクレジスタ (ILM) に設定された割込みレベル値よりも弱い割込みレベル値の割込みは受付けません。

- リセットにより割込みレベルマスクレジスタ (ILM) は、最強の割込みレベルが設定され、割込みを受付けません。
- アセンブラ命令では、割込みレベルマスクレジスタ (ILM) へ転送する 8 ビットの即値転送命令を使用できます。ただし、下位 3 ビットが有効となります。

表 3.7-3 割込みレベルマスクレジスタ (ILM) と割込みレベルの強弱

ILM2	ILM1	ILM0	割込みレベル	割込みレベルの強弱
0	0	0	0	強（割込み禁止）
0	0	1	1	
0	1	0	2	
0	1	1	3	
1	0	0	4	
1	0	1	5	
1	1	0	6	
1	1	1	7	弱

### 3.7.7 プログラムカウンタ (PC)

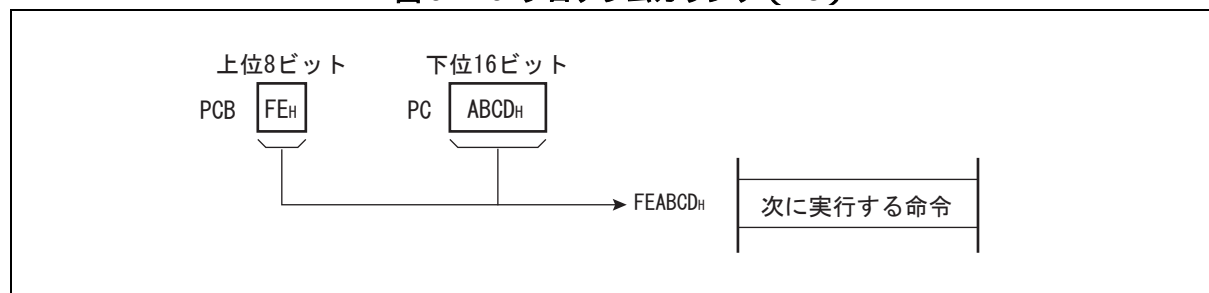
プログラムカウンタ (PC) は、CPU が次に実行する命令アドレスの下位 16 ビットを示す 16 ビットカウンタです。

#### ■ プログラムカウンタ (PC)

CPU が次に実行する命令アドレスの、上位 8 ビットをプログラムバンクレジスタ (PCB) で設定し、下位 16 ビットをプログラムカウンタ (PC) で設定します。次に実行する命令アドレスは、図 3.7-13 のように決定します。プログラムカウンタ (PC) は条件分岐命令、サブルーチンコール命令、割込みやりセットでも内容が更新されます。また、オペランドを読み出す場合のベースポインタとしても使用できます。

プログラムバンクレジスタ (PCB) の詳細は、「3.7.9 バンクレジスタ (PCB, DTB, USB, SSB, ADB)」をご参照ください。

図 3.7-13 プログラムカウンタ (PC)



< 注意事項 > プログラムカウンタ (PC) やプログラムバンクレジスタ (PCB) をプログラム (MOV PC, #0FFH などの命令) で直接書き換えることはできません。

### 3.7.8      ダイレクトページレジスタ（DPR）

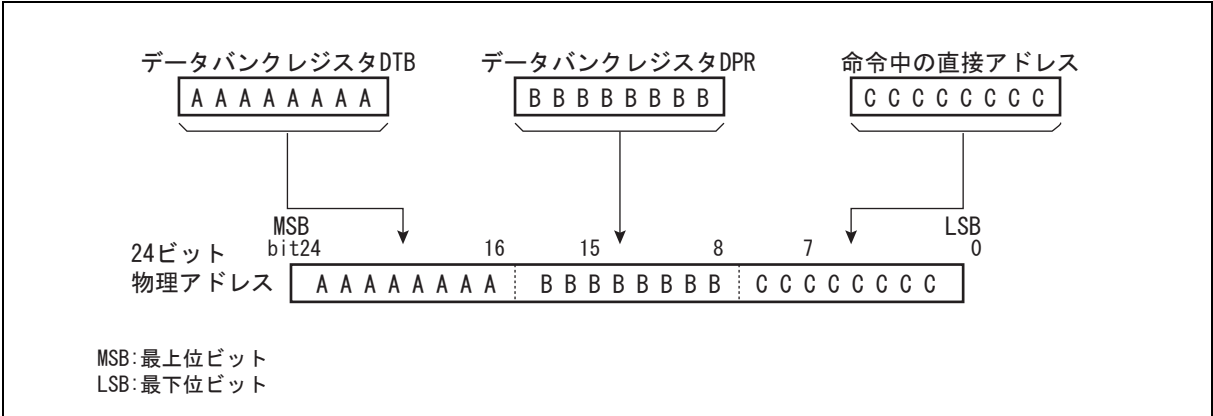
ダイレクトページレジスタ（DPR）は、短縮直接アドレス指定方式の命令実行時に、オペランドアドレスのビット 8 ～ 15（addr8 ～ addr15）を指定する 8 ビットレジスタです。リセットにより "01<sub>H</sub>" に初期化されます。

■    ダイレクトページレジスタ（DPR）

短縮直接アドレス指定方式については、「付録 B.3 直接アドレッシング」をご参照ください。

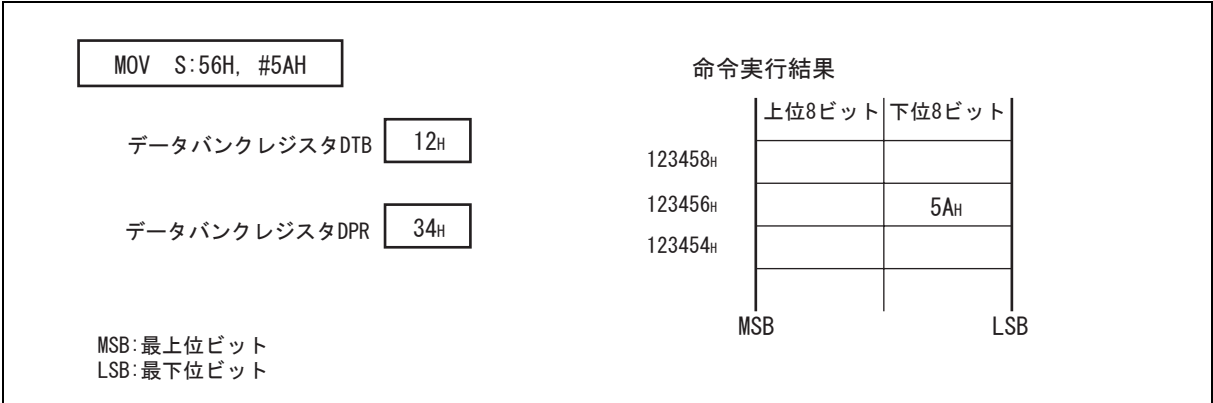
ダイレクトページレジスタ（DPR）による物理アドレスの生成を図 3.7-14 に示します。

図 3.7-14    ダイレクトページレジスタ（DPR）による物理アドレスの生成



ダイレクトページレジスタ（DPR）の設定とデータアクセス例を図 3.7-15 に示します。

図 3.7-15    ダイレクトページレジスタ（DPR）の設定とデータアクセス例



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

---

バンクレジスタは、バンク方式アドレス指定の最上位 8 ビットアドレスを指定するレジスタであり、以下に示す 5 種類のレジスタで構成されます。

- プログラムバンクレジスタ (PCB)
- データバンクレジスタ (DTB)
- 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)
- アディショナルバンクレジスタ (ADB)

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、システムスタック空間およびアディショナル空間に配置されるメモリバンクを示します。

---

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

- プログラムバンクレジスタ (PCB)

プログラム (PC) 空間を指定するバンクレジスタです。

- データバンクレジスタ (DTB)

データ (DT) 空間を指定するバンクレジスタです。

- 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)

スタック (SP) 空間を指定するバンクレジスタです。

- アディショナルバンクレジスタ (ADB)

アディショナル (AD) 空間を指定するバンクレジスタです。

- 各バンクの設定とデータアクセス

バンクレジスタは、8 ビット長です。リセットによりプログラムバンクレジスタ (PCB) は "FF<sub>H</sub>" に初期化され、データバンクレジスタ (DTB)、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) およびアディショナルバンクレジスタ (ADB) は "00<sub>H</sub>" に初期化されます。プログラムバンクレジスタ (PCB) は、読み出しはできますが書き込みはできません。プログラムバンクレジスタ (PCB) 以外のバンクレジスタは、読み書き可能です。

---

< 注意事項 > MB90460/465 シリーズは、デバイスに内蔵されるメモリ空間までをサポートします。各レジスタ動作の詳細は、「3.4.2 バンク方式によるアドレス指定」をご参照ください。

---

## 3.8 汎用レジスタ

汎用レジスタは、RAM 上の "000180<sub>H</sub> ~ 00037F<sub>H</sub>" に、16 ビット× 8 本を 1 つのレジスタバンクとして割当てたメモリブロックです。汎用の 8 ビットレジスタ（バイトレジスタ R0 ~ R7）、16 ビットレジスタ（ワードレジスタ RW0 ~ RW7）もしくは 32 ビットレジスタ（ロングワードレジスタ RL0 ~ RL7）として使用できます。汎用レジスタへは、短い命令で高速にアクセスできます。また、レジスタバンクによってブロック化されているために、データの保護や機能単位の分割が容易です。また、ロングワードレジスタとして使用した場合は、全空間へ直接アクセスするリニアポインタとしても使用できます。

### ■ 汎用レジスタの構成

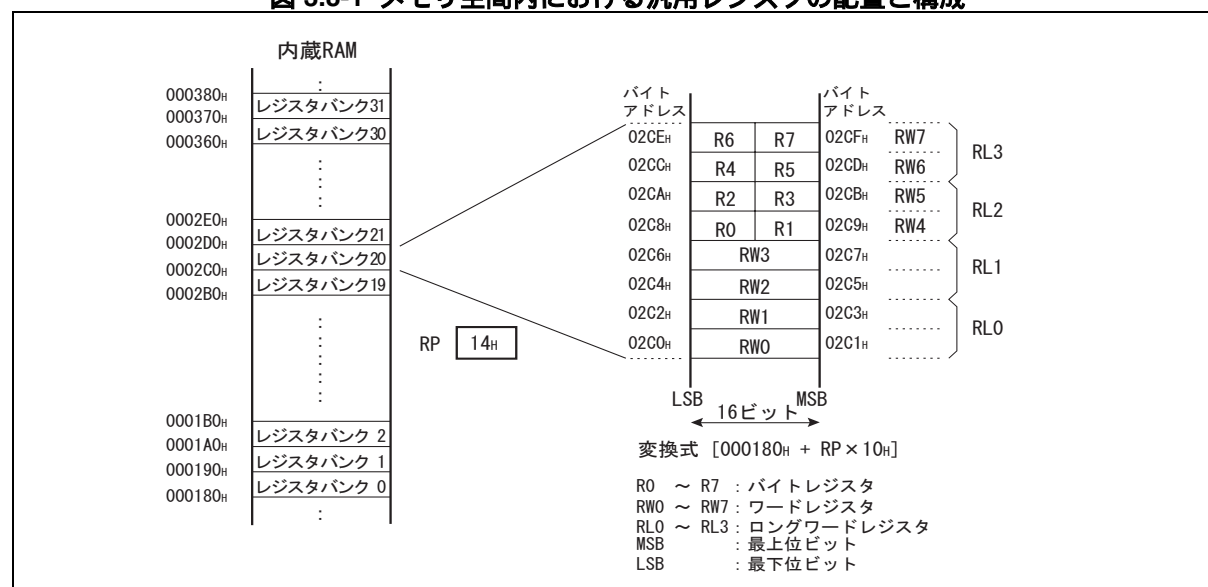
汎用レジスタは、RAM 上の "000180<sub>H</sub> ~ 00037F<sub>H</sub>" に全部で 32 バンク存在し、レジスタバンクポインタ（RP）でバンクを指定します。レジスタバンクポインタ（RP）で設定するバンクの先頭アドレスは、下式のようにになります。16 ビット× 8 本を 1 つのレジスタバンクとして定義します。

$$\text{汎用レジスタの先頭アドレス} = 000180_{\text{H}} + \text{レジスタバンクポインタ (RP)} \times 10_{\text{H}}$$

レジスタバンクポインタ（PR）の詳細は、「3.7.5 レジスタバンクポインタ（PS:RP）」をご参照ください。

メモリ空間内における汎用レジスタの配置と構成を図 3.8-1 に示します。

図 3.8-1 メモリ空間内における汎用レジスタの配置と構成



< 注意事項 > レジスタバンクポインタ（RP）は、リセット後 "00<sub>H</sub>" に初期化されます。

## ■ レジスタバンク

レジスタバンク内の汎用レジスタは、RAM と同様にリセットでは初期化されず、リセット前のデータを保持します。ただし、パワーオンリセット時には不定となります。

表 3.8-1 汎用レジスタの代表的な機能

レジスタ名	機能
R0 to R7	各種命令のオペランドとして使用 (注意事項) R0 はパレルシフトのカウンタおよびノーマライズ (正規化) 命令のカウンタとしても使用
RW0 to 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
RL0 to 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

## 3.9 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設定した場合は、プリフィックスコード直後の命令動作を変更できます。プリフィックスコードには、以下に示す3種類があります。

- バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)
  - コモンレジスタバンクプリフィックス (CMR)
  - フラグ変化抑止プリフィックス (NCC)
- 

### ■ プリフィックスコード

#### ● バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)

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。

詳細は、「3.9.1 バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)」をご参照ください。

#### ● コモンレジスタバンクプリフィックス (CMR)

レジスタバンクをアクセスする命令の前にコモンレジスタバンクプリフィックスを設定した場合は、レジスタバンクポインタ (RP) の値とは関係なく、レジスタアクセスを "000180<sub>H</sub> ~ 00018F<sub>H</sub>" に存在するコモンバンク (RP=00<sub>H</sub> の場合に設定されるレジスタバンク) へ変更できます。

詳細は、「3.9.2 コモンレジスタバンクプリフィックス (CMR)」をご参照ください。

#### ● フラグ変化抑止プリフィックス (NCC)

フラグ変化を抑止したい命令の前に、フラグ変化抑止プリフィックスコードを設定した場合は、命令の実行に伴うフラグ変化を抑止できます。

詳細は、「3.9.3 フラグ変化抑止プリフィックス (NCC)」をご参照ください。



### 3.9.1 バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)

データアクセス時のメモリ空間は、アドレス指定方式ごとに定められています。しかし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、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。

#### ■ バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)

表 3.9-1 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	設定される空間
PCB	プログラム空間
DTB	データ空間
ADB	アディショナル空間
SPB	コンディションコードレジスタ (CCR) の S フラグの値が "0" の場合はユーザスタック空間が、"1" の場合はシステムスタック空間が用いられます。

バンクセレクトプリフィックスを使用した場合、例外的に動作する命令があります。

表 3.9-2 バンクセレクトプリフィックスの影響を受けない命令

命令の種類	命 令	バンクセレクトプリフィックスの効果
ストリング 命令	MOVS SCEQ FILS MOVSW SCWEQ FILSW	プリフィックスの有無にかかわらず、オペランドで指定されたバンクレジスタが使用されます。
スタック操 作命令	PUSHW POPW	プリフィックスの有無にかかわらず、S フラグが "0" の場合はユーザスタックバンクレジスタ (USB) が、S フラグが "1" の場合はシステムスタックバンクレジスタ (SSB) が使用されます。
I/O アクセ ス命令	MOV A MOVW A, io MOV io, A MOV io, #imm8 MOVB A, io:bp SETB io:bp BBC io:bp, rel WBTC io, bp MOVX A, io MOVW io, A MOVW io, #imm16 MOVB io:bp, A CLRB io:bp BBS io:bp, rel WBTS io:bp	プリフィックスの有無にかかわらず、I/O 空間 ("000000 <sub>H</sub> " ~ "0000FF <sub>H</sub> ") がアクセスされます。
割込み復帰 命令	RETI	プリフィックスの有無にかかわらず、システムスタックバンクレジスタ (SSB) が使用されます。

表 3.9-3 バンクセレクトプリフィックスの使用時に注意すべき命令

命令の種類	命 令	説 明
フラグ変更命令	AND CCR, #imm8 OR CCR, #imm8	プリフィックスの効果は次の命令までおよびます。
ILM 設定命令	MOV ILM, #imm8	プリフィックスの効果は次の命令までおよびます。
PS 復帰命令	POPW PS	PS 復帰命令に対してはバンクセレクトプリフィックスを付加しないでください。

## 3.9.2 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、レジスタバンクをアクセスする命令の前に設定した場合は、レジスタバンクポインタ (RP) の値とは関係なく、レジスタアクセスを、"000180<sub>H</sub> ~ 00018F<sub>H</sub>" に存在するコモンバンク (RP=00<sub>H</sub> の場合に設定されるレジスタバンク) へ変更できます。

### ■ コモンレジスタバンクプリフィックス (CMR)

複数のタスク間におけるデータ交換を容易にするために、F<sup>2</sup>MC-16LX ファミリでは各タスクで共通に使用できるコモンバンクを用意しています。コモンバンクは、"000180<sub>H</sub> ~ 00018F<sub>H</sub>" に存在します。

ただし、コモンレジスタバンクプリフィックス (CMR) を使用する際は表 3.9-4 に示した命令に注意してください。

表 3.9-4 コモンレジスタバンクプリフィックス (CMR) 使用時に注意すべき命令

命令の種類	命 令	説 明
ストリング命令	MOVS MOVSW SCEQ SCWEQ FILS FILSW	ストリング命令に対しては、CMR プリフィックスを付加しないでください。
フラグ変更命令	AND CCR, #imm8 OR CCR, #imm8	プリフィックスの効果は、次の命令までおよびます。
PS 復帰命令	POPW PS	プリフィックスの効果は、次の命令までおよびます。
ILM 設定命令	MOV ILM, #imm8	プリフィックスの効果は、次の命令までおよびます。

### 3.9.3 フラグ変化抑止プリフィックス (NCC)

フラグ変化を抑止したい命令の前に、NCC プリフィックスを設定した場合は、命令の実行に伴うフラグ変化を抑止できます。

#### ■ フラグ変化抑止プリフィックス (NCC)

不要なフラグ変化を抑止するために、フラグ変化抑止プリフィックス (NCC) を用います。フラグ変化の抑止対象となるフラグは T, N, Z, V, C です。

ただし、フラグ変化抑止プリフィックスを使用する際は表 3.9-5 に示した命令に注意してください。

T, N, Z, V, C の各フラグの詳細は、「3.7.4 コンディションコードレジスタ (PS:CCR)」をご参照ください。

表 3.9-5 フラグ変化抑止プリフィックス (NCC) の使用時に注意すべき命令

命令の種類	命 令	説 明
ストリング命令	MOVS MOVSW SCEQ SCWEQ FILS FILSW	ストリング命令に対しては、NCC プリフィックスを付加しないでください。
フラグ変更命令	AND CCR, #imm8 OR CCR, #imm8	プリフィックスの有無にかかわらず、コンディションコードレジスタ (CCR) は命令の仕様どおりに変化します。プリフィックスの効果は、次の命令までおよびます。
PS 復帰命令	POPW PS	プリフィックスの有無にかかわらず、コンディションコードレジスタ (CCR) は命令の仕様どおりに変化します。プリフィックスの効果は、次の命令までおよびます。
ILM 設定命令	MOV ILM, #imm8	プリフィックスの効果は、次の命令までおよびます。
割込み命令、 割込み復帰命令	INT #vct8 INT9 INT adder16 INTP addr24 RETI	プリフィックスの有無にかかわらず、コンディションコードレジスタ (CCR) は命令の仕様どおりに変化します。
コンテキスト スイッチ命令	JCTX @A	プリフィックスの有無にかかわらず、コンディションコードレジスタ (CCR) は命令の仕様どおりに変化します。

### 3.9.4 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の使用には、以下のような制約があります。**

- プリフィックスコードや割込み / ホールド抑止命令の実行中は、割込み / ホールド要求を受付けません。
- 割込み / ホールド命令の前にプリフィックスコードを置いた場合、プリフィックスコードの効果は遅延します。
- 競合するプリフィックスコードが連続している場合は、最後のプリフィックスコードが有効となります。

## ■ プリフィックスコードと割込み抑止命令

プリフィックスコードと割込み抑止命令を表 3.9-6 に示します。

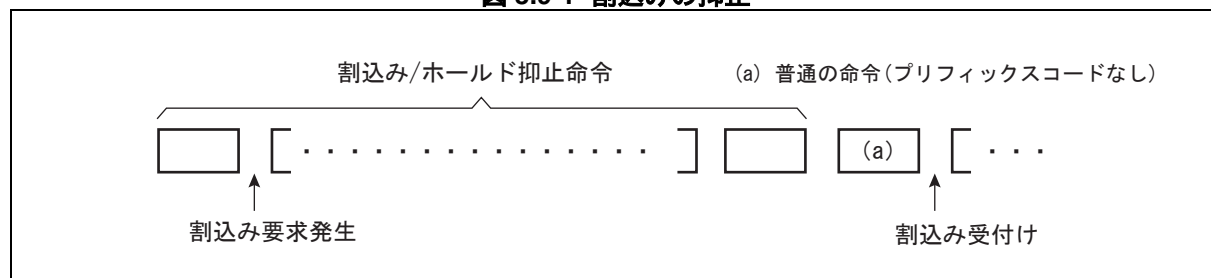
**表 3.9-6 プリフィックスコードと割込み抑止命令**

	プリフィックス コード	割り込み / ホールド抑止命令 (プリフィックスコードの効果を遅延させる命令)
割り込みやホールド 要求を受付けない	PCB DTB ADB SPB CMR NCC	MOV ILM, #imm8 OR CCR, #imm8 AND CCR, #imm8 POPW PS

**< 注意事項 > 割込みの抑止**

図 3.9-1 に示すように、プリフィックスコードや割込み / ホールド命令の実行中は、割込みやホールド要求が発生しても受け付けられません。これらの要求が受け付けられない場合、次に割込み / ホールドが処理されるのは、プリフィックスコードや割込み / ホールド抑止命令以降、初めてプリフィックスコードや割込み / ホールド命令以外の命令が実行された後になります。

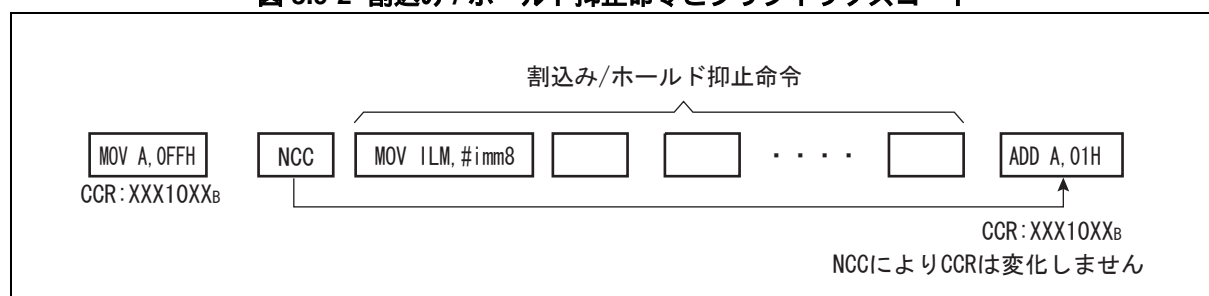
**図 3.9-1 割込みの抑止**



● プリフィックスコードの効果遅延

割込み/ホールド抑止命令の前にプリフィックスコードを設定した場合、プリフィックスコードの効果は、割込み/ホールド抑止命令後の命令に対して有効となります。

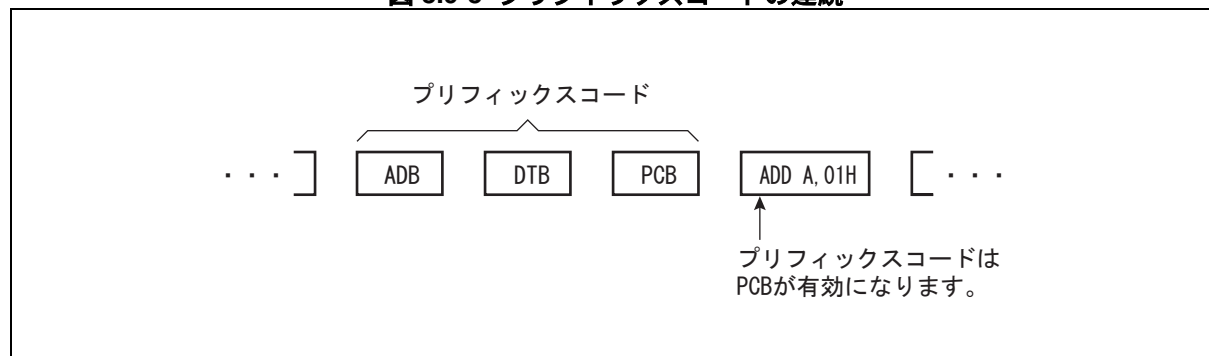
図 3.9-2 割込み/ホールド抑止命令とプリフィックスコード



■ 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（PCB, ADB, DTB, SPB）が連続していた場合、最後のプログラム空間（PCB）に設定された値が有効になります。

図 3.9-3 プリフィックスコードの連続



### 3.9.5 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意

「DIV A, Ri」, 「DIVW A, RWi」命令を使用する場合は, バンクレジスタに "00<sub>H</sub>" を設定してください。

#### ■「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意

「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意を表 3.9-7 に示します。

表 3.9-7 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意 (i=0 ~ 7)

命 令	左記命令実行時に 影響を受ける バンクレジスタ名	余りが格納されるアドレス
DIV A, R0	DTB	(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +8 <sub>H</sub> : 下位 16 ビット)
DIV A, R1		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +9 <sub>H</sub> : 下位 16 ビット)
DIV A, R4		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +C <sub>H</sub> : 下位 16 ビット)
DIV A, R5		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +D <sub>H</sub> : 下位 16 ビット)
DIVW A, RW0		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +0 <sub>H</sub> : 下位 16 ビット)
DIVW A, RW1		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +2 <sub>H</sub> : 下位 16 ビット)
DIVW A, RW4		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +8 <sub>H</sub> : 下位 16 ビット)
DIVW A, RW5		(DTB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +A <sub>H</sub> : 下位 16 ビット)
DIV A, R2	ADB	(ADB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +A <sub>H</sub> : 下位 16 ビット)
DIV A, R6		(ADB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +E <sub>H</sub> : 下位 16 ビット)
DIVW A, RW2		(ADB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +4 <sub>H</sub> : 下位 16 ビット)
DIVW A, RW6		(ADB: 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +E <sub>H</sub> : 下位 16 ビット)
DIV A, R3	USB SSB <sup>*1</sup>	(USB <sup>*2</sup> : 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +B <sub>H</sub> : 下位 16 ビット)
DIV A, R7		(USB <sup>*2</sup> : 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +F <sub>H</sub> : 下位 16 ビット)
DIVW A, RW3		(USB <sup>*2</sup> : 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +6 <sub>H</sub> : 下位 16 ビット)
DIVW A, RW7		(USB <sup>*2</sup> : 上位 8 ビット) + (0180 <sub>H</sub> +RP × 10 <sub>H</sub> +E <sub>H</sub> : 下位 16 ビット)

\*1:CCR レジスタの S ビットによる

\*2:CCR レジスタの S ビットが "0" の場合

バンクレジスタ (DTB, ADB, USB, SSB) の値が "00<sub>H</sub>" の場合は、除算結果の余りは命令オペランドのレジスタに格納されます。バンクレジスタの値が "00<sub>H</sub>" 以外の場合は、上位 8 ビットアドレスは命令オペランドのレジスタに対応したバンクレジスタにより指定され、下位 16 ビットアドレスは命令オペランドのレジスタのアドレスと同じアドレスとなり、上位 8 ビットで指定されたバンクのレジスタに余りが格納されます。

**【例】**

DTB=053<sub>H</sub>, RP=03<sub>H</sub> の場合で「DIV A, R0」を実行すると、R0 のアドレスは "0180<sub>H</sub>" + RP ("03<sub>H</sub>") × "10<sub>H</sub>" + "08<sub>H</sub>" (R0 相当アドレス) = 0001B8<sub>H</sub> です。

ここで、「DIV A, R0」で指定されるバンクレジスタはデータバンクレジスタ (DTB) ですので、バンクアドレス "053<sub>H</sub>" を付加したアドレス "05301B8<sub>H</sub>" に余りが格納されます (Ri および RWi のレジスタについては、「3.8 汎用レジスタ」をご参照ください)。

## ■ 注意事項の回避について

「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意事項を回避してプログラムを開発していただくために、コンパイラでは表 3.9-7 の命令を生成しないように変更し、アセンブラでは表 3.9-7 の命令を同等の命令列に置換える機能を追加したバージョンが用意されています。コンパイラおよびアセンブラは、以下のバージョンをご使用ください。

### ● コンパイラ

- cc907 の V02L06 以降のバージョンおよび fcc907s の V30L02 以降のバージョン

### ● アセンブラ

- asm907a の V03L04 以降のバージョンおよび fasm907s の V30L04 (Rev. 300004) 以降のバージョン





# 第4章

---

## リセット

**MB90460/465 シリーズのリセットについて説明します。**

- 4.1 リセットの概要
- 4.2 リセット要因と発振安定待ち時間
- 4.3 外部リセット端子
- 4.4 リセット動作
- 4.5 リセット要因フラグビット
- 4.6 リセットによる各端子の状態

## 4.1 リセットの概要

リセット要因が発生した場合，CPU は現在実行中の処理を中断し，リセット解除待ち状態になります。リセット解除後は，リセットベクタが示すアドレスから処理を開始します。

リセットには，以下に示す4種類の要因があります。

- パワーオンリセットの発生（電源投入時）
- ウォッチドッグタイマのオーバフロー（ウォッチドッグタイマを使用している場合）
- 端子から外部リセット信号を入力した場合
- 低消費電力モード制御レジスタ（LPMCR）の内部リセット信号発生ビット（RST）に "0" を設定した場合（ソフトウェアリセット）

### ■ リセット要因

表 4.1-1 リセット要因

リセット	発生要因	マシクロック	ウォッチドッグタイマ	発振安定待ち
外部リセット	$\overline{\text{RST}}$ 端子への "L" レベル入力	メインクロック (MCLK)	カウント停止	なし
ソフトウェアリセット	低消費電力モード制御レジスタ (LPMCR) の内部リセット信号発生ビット (RST) に "0" を設定	メインクロック (MCLK)	カウント停止	なし
ウォッチドッグタイマリセット	ウォッチドッグ機能が許可されている場合に，ウォッチドッグタイマがオーバフローした場合	メインクロック (MCLK)	カウント停止	なし
パワーオンリセット	電源投入時	メインクロック (MCLK)	カウント停止	あり

MCLK: メインクロック周波数（発振クロックの2分周クロック :2/HCLK）

HCLK: 発振クロック周波数

#### ● 外部リセット

外部リセットは，外部リセット端子（端子）に "L" レベルを設定した場合に発生します。"L" レベル入力時間は，16 マシンサイクル（16/φ）以上必要です。マシクロックで動作している場合には，外部リセット端子に "L" レベルを設定してリセットが発生しても，発振安定待ち時間を取りません。

#### < 参考 >

##### 命令実行中のリセット

命令実行中（転送系命令実行中 MOV 命令など）に外部リセット端子をアサートした場合，実行中の命令が終了した後にリセット入力の有効になります。ただし，ストリング系命令（MOVS 命令など）の実行中は，設定したカウンタ値による転送が完了する前にリセット入力がある場合があります。外部リセット端子をアサートした場合，ポート端子は命令実行サイクルに関係なくリセット状態になります（アサートされた場合は非同期です）。

### ● ソフトウェアリセット

ソフトウェアリセットは、低消費電力モード制御レジスタ (LPMCR) の内部リセット信号発生ビット (RST) に "0" を設定することにより、3 マシンサイクル ( $3/\phi$ ) のリセットを発生します。ソフトウェアリセット時は、発振安定待ち時間を取りません。

### ● ウォッチドッグリセット

ウォッチドッグリセットは、ウォッチドッグタイマ起動後から、ウォッチドッグタイマ制御レジスタ (WDTC) のインターバル時間設定ビット (WT1, WT0) で設定した時間内に、ウォッチドッグタイマ制御レジスタ (WDTC) のウォッチドッグ制御ビット (WTE) に "0" を設定しなかった場合に発生します。

発振安定待ち時間はクロック選択レジスタ (CKSCR) によって設定できます。

### ● パワーオンリセット

パワーオンリセットは、電源投入時に発生するリセットです。

発振安定待ち時間は  $2^{18}/\text{HCLK}$  (約 65.54ms: 発振クロック 4MHz 時) の固定です。発振安定待ち時間が経過した後、リセットが動作します。

#### < 参考 >

#### クロックの定義

HCLK: 発振クロック周波数 (発振端子から供給されるクロック)

MCLK: メインクロック周波数 (発振クロックの2分周クロック)

$\phi$ : マシンクロック周波数 (CPU 動作クロック)

$1/\phi$ : マシンサイクル (CPU 動作クロック周期)

マシンクロックの詳細は、「5.1 クロックの概要」ご参照ください。

## 4.2 リセット要因と発振安定待ち時間

F<sup>2</sup>MC-16LX ファミリには4種類のリセット要因があり、リセット時の発振安定待ち時間はリセット要因によって異なります。

### ■ リセット要因と発振安定待ち時間

リセット要因と発振安定待ち時を表 4.2-1 に示します。

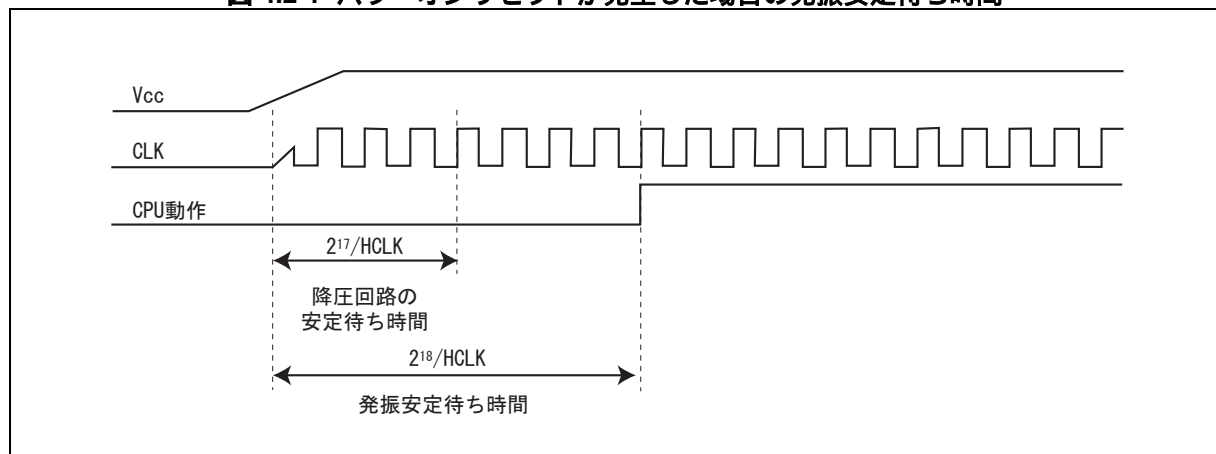
表 4.2-1 リセット要因と発振安定待ち時間

リセット要因	発振安定待ち時間 ( ) 内は発振クロック周波数 4MHz 時
パワーオンリセット	218/HCLK ( 約 65.54ms )
ウォッチドッグタイマ	なし ( WS1, WS0 ビットは "11 <sub>B</sub> " に初期化されます )
外部リセット	なし ( WS1, WS0 ビットは "11 <sub>B</sub> " に初期化されます )
ソフトウェアリセット	なし ( WS1, WS0 ビットは "11 <sub>B</sub> " に初期化されます )

HCLK: 発振クロック周波数 ( MHz )

パワーオンリセットが発生した場合の発振安定待ち時間を図 4.2-1 に示します。

図 4.2-1 パワーオンリセットが発生した場合の発振安定待ち時間



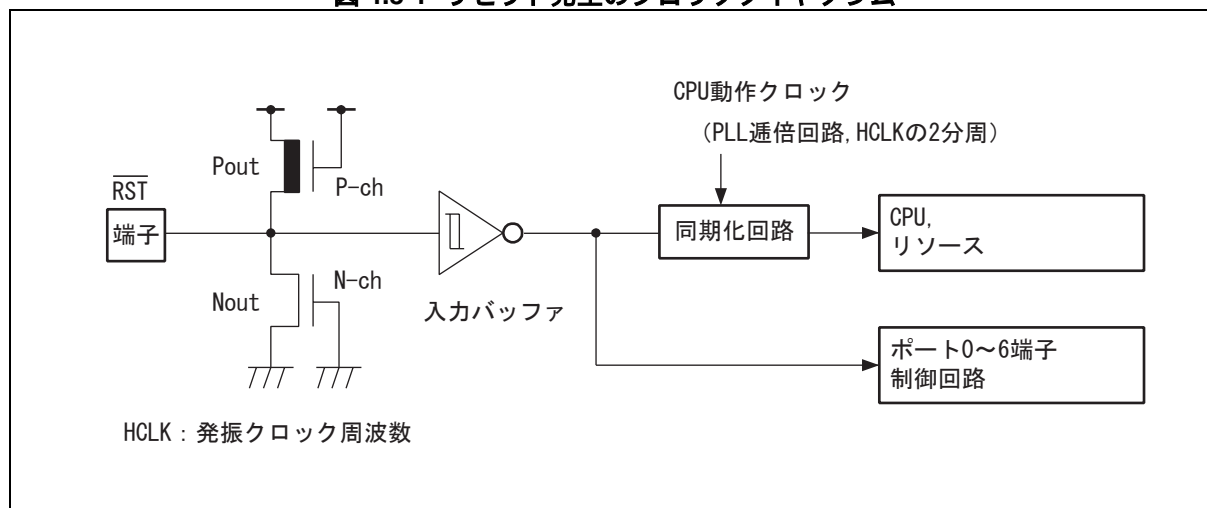
< 注意事項 > 発振クロック振動子は、発振を開始してから固有振動数で安定するまでに、振動子固有の発振安定待ち時間が必要です。使用する振動子に合わせた発振安定待ち時間を設定してください。発振安定待ち時間の詳細は、「5.5 発振安定待ち時間」をご参照ください。

### 4.3 外部リセット端子

外部リセット端子 ( $\overline{\text{RST}}$  端子) へ "L" レベル入力を設定することにより, リセットが発生します。

#### ■ 外部リセット端子のブロックダイアグラム

図 4.3-1 リセット発生ブロックダイアグラム



< 注意事項 > 内部回路を初期化するには, マシンクロックが必要です。リセット入力時には, 発振端子からクロックが供給されている必要があります。

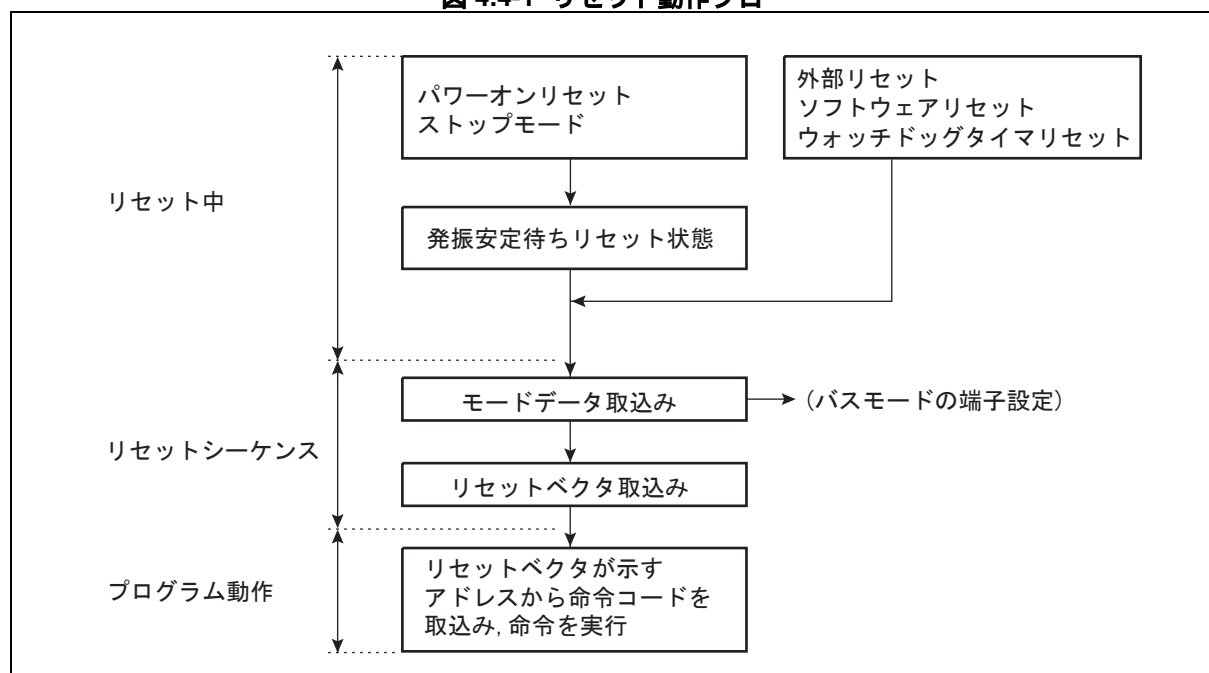
## 4.4 リセット動作

リセットが解除された場合，モード端子の設定により，内部メモリまたは外部メモリに設定されたモードデータとリセットベクタを取込みます。モードデータレジスタにより CPU の動作モードが設定され，リセットベクタによりリセットシーケンス終了後の実行開始アドレスが設定されます。

### ■ リセット動作の概要

リセット動作フローを図 4.4-1 に示します。

図 4.4-1 リセット動作フロー



### ■ モード端子

モード端子（MD2 ～ MD0）は，モードデータとリセットベクタの取込み方法を事前に設定します。モードデータとリセットベクタを取込む場合は，リセットシーケンスを実行します。詳細は，「8.2 モード端子（MD2 ～ MD0）」をご参照ください。

### ■ モードデータの取込み

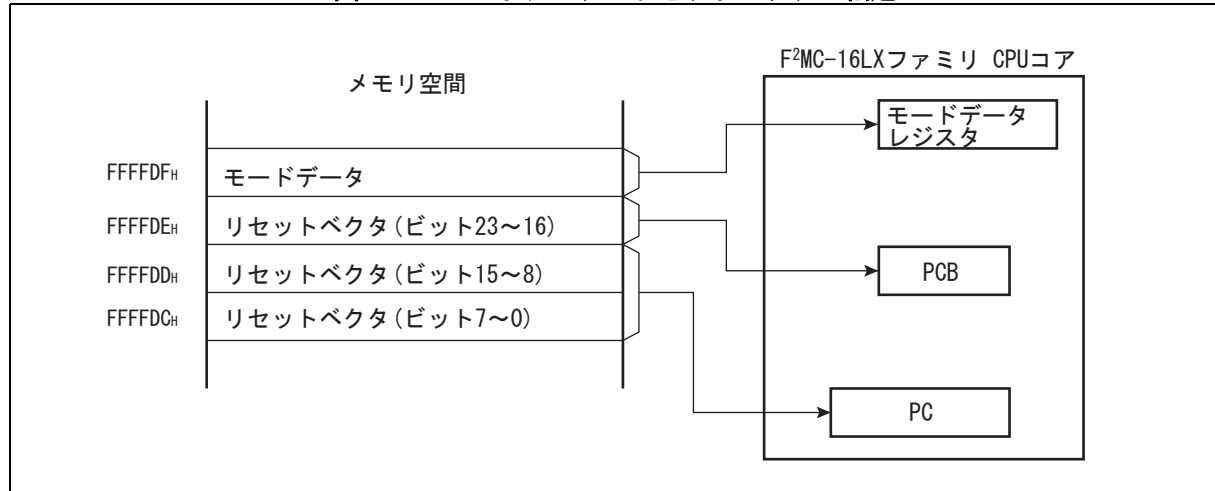
リセットが解除された場合，CPU はモードデータをモードデータレジスタに取込みます。モードデータを取込んだ後に，リセットベクタをプログラムカウンタ（PC）とプログラムカウンタバンクレジスタ（PCB）に取込みます。

モードデータレジスタは，バスモードとバス幅を設定できます。また，リセットベクタでは，プログラムの開始アドレスを指定できます。

モードデータ取込みの詳細は，「第8章 モード設定」をご参照ください。

モードデータとリセットベクタの転送を，図 4.4-2 に示します。

図 4.4-2 モードデータとリセットベクタの転送



## &lt; 参考 &gt;

**モード端子の設定**

モードデータとリセットベクタを、内蔵 ROM (またはフラッシュメモリ)、あるいは外部メモリから読み出すかは、モード端子の設定により決まります。モード端子を外部メモリに設定した場合は、外部メモリからモードデータとリセットベクタを読み出します。また、モード端子を内蔵 ROM (またはフラッシュメモリ) に設定した場合は、内蔵 ROM (またはフラッシュメモリ) からモードデータとリセットベクタを読み出します。

シングルチップモードの場合は、モード端子を内蔵 ROM (またはフラッシュメモリ) に設定してください。

詳細は、「8.2 モード端子 (MD2 ~ MD0)」をご参照ください。

- モードデータレジスタ (アドレス: FFFFDF<sub>H</sub>)

モードデータレジスタは、リセットシーケンス実行中に設定を変更できます。

モードデータレジスタの設定は、リセットベクタ取込み後から有効になります。

また、命令を使用して "FFFFDF<sub>H</sub>" にモードデータを設定しても、モードデータレジスタを書換えることはできません。

詳細は、「8.3 モードデータレジスタ」をご参照ください。

- リセットベクタ (アドレス: FFFFDC<sub>H</sub> ~ FFFFDE<sub>H</sub>)

リセット解除後のプログラム開始アドレスを設定します。リセットベクタで設定されたアドレスからプログラムを実行します。



## 4.5 リセット要因フラグビット

リセット発生要因は、ウォッチドッグタイマ制御レジスタ (WDTC) 内のフラグを読み出すことで判別できます。

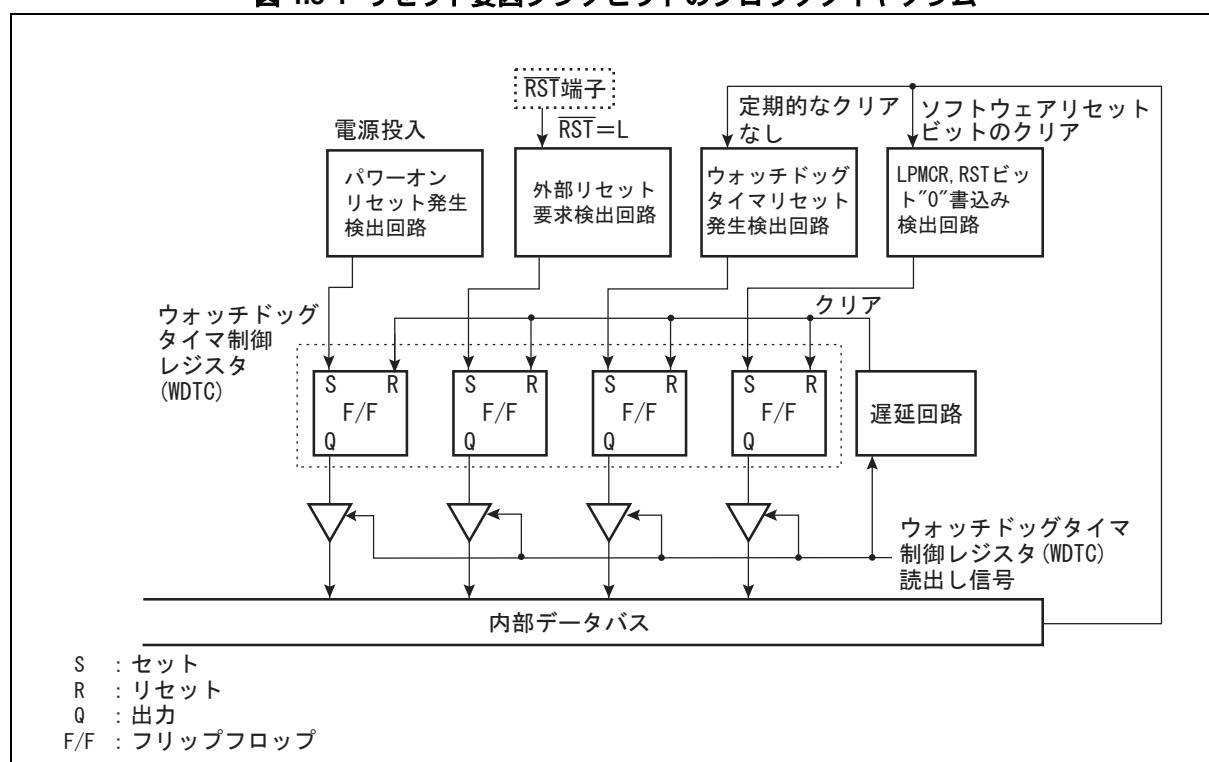
### ■ リセット要因フラグビット

ウォッチドッグタイマ制御レジスタ (WDTC) のリセット要因フラグビット (PONR, WRST, ERST, SRST) でリセット要因を確認できます。リセット解除後に、リセット発生要因を判別する必要がある場合には、ウォッチドッグタイマ制御レジスタ (WDTC) のリセット要因フラグビット (PONR, WRST, ERST, SRST) を読み出してください。

ただし、リセット要因フラグビット (PONR, WRST, ERST, SRST) は、ウォッチドッグタイマ制御レジスタ (WDTC) を読み出した場合、"0" にクリアされます。

リセット要因フラグビットのブロックダイアグラムを、図 4.5-1 に示します。

図 4.5-1 リセット要因フラグビットのブロックダイアグラム



■ リセット要因フラグビットとリセット要因の対応

リセット要因フラグビットの構成を、図 4.5-2 に示します。

図 4.5-2 リセット要因フラグビットの構成（ウォッチドッグタイマ制御レジスタ）

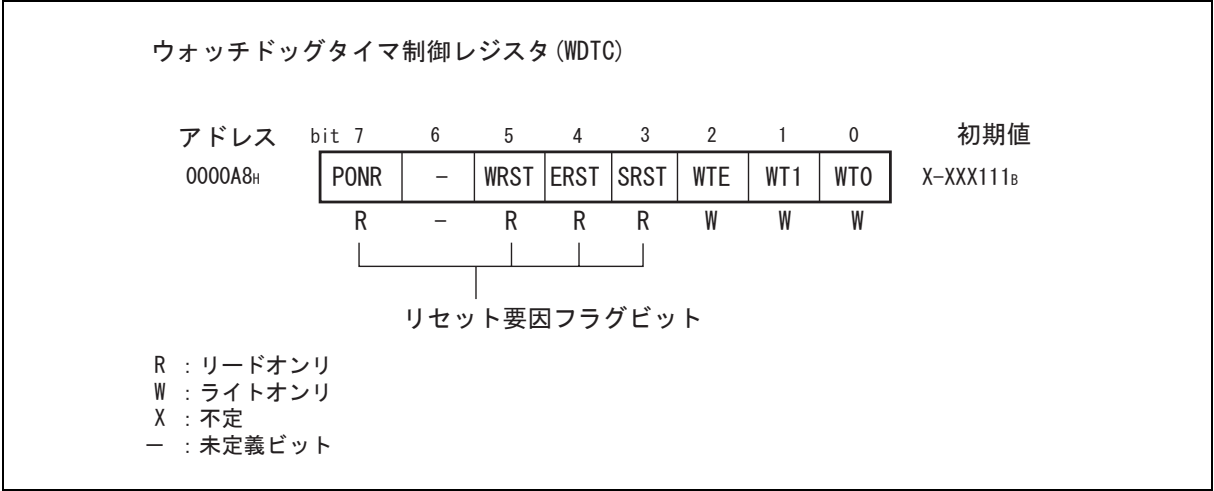


表 4.5-1 リセット要因フラグビットの値とリセット要因の対応

リセット要因	PONR	WRST	ERST	SRST
パワーオンリセット	1	×	×	×
ウォッチドッグタイマリセット	*	1	*	*
外部リセット (RST 端子入力)	*	*	1	*
ソフトウェアリセット	*	*	*	1

※: 前の状態を保持  
× : 不定

■ リセット要因フラグビットの注意事項

● 複数のリセット要因が発生した場合

リセット要因を複数検出した場合でも、ウォッチドッグタイマ制御レジスタ (WDTC) の対応するリセット要因フラグビット (PONR, WRST, ERST, SRST) には "1" が設定されます。

< 参考 > 外部リセット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、ウォッチドッグタイマ制御レジスタ (WDTC) のリセット要因フラグビット (ERST, WRST) に "1" が設定されます。

● パワーオンリセットが発生した場合

パワーオンリセットが発生した場合は、ウォッチドッグタイマ制御レジスタ (WDTC) の PONR ビットに "1" が設定されます。しかし WRST, ERST, SRST ビットは不定値となります。

PONR ビットに "1" が設定されている場合は、WRST, ERST, SRST ビットの値を無視してください。

## 第4章 リセット

### ● リセット要因フラグビットのクリア

リセット要因フラグビット (PONR, WRST, ERST, SRST) は、ウォッチドッグタイマ制御レジスタ (WDTC) を読み出した場合、全て "0" にクリアされます。リセットが発生してもウォッチドッグタイマ制御レジスタ (WDTC) を読み出さなければ、リセット要因フラグビットは "0" にクリアされません。

## 4.6 リセットによる各端子の状態

---

リセットによる各端子の状態について説明します。

---

### ■ リセットによる各端子の状態

リセットによる各端子の状態は、モード端子(MD2～MD0)の設定によって異なります。

- 内部ベクタモード設定時 (MD=0, MD1=1, MD0=1)

ポート0～6はハイインピーダンス出力となります。

モードデータは内蔵ROM(フラッシュメモリ)から読み出されます。

### ■ モードデータ読出し後の端子状態

モードデータ読出し後の端子状態は、モードデータレジスタのバスモード設定ビット(M1, M0)によって設定されます。

- シングルチップモード設定時 (モードデータレジスタ:M1, M0ビット=00<sub>B</sub>)

ポート0～6はハイインピーダンス出力となります。

モードデータは、内蔵ROM(またはフラッシュメモリ)から読み出されます。

---

< 注意事項 > 外部端子レベルを、外部回路が動作しないように設定してください。

---

リセット中の各端子状態は、「6.7 スタンバイモード、リセットの端子状態」の表 6.7-1 をご参照ください。



# 第5章

---

## クロック

**MB90460/465 シリーズのクロックについて説明します。**

- 5.1 クロックの概要
- 5.2 クロック発生部のブロックダイアグラム
- 5.3 クロック選択レジスタ (CKSCR)
- 5.4 クロックモード
- 5.5 発振安定待ち時間
- 5.6 振動子と外部クロックの接続

## 5.1 クロックの概要

---

クロック発生部は、CPU およびリソースの動作クロックを制御します。この動作クロックを " マシナクロック " とよびます。

クロックとしては、以下の 4 種類があります。

- 発振クロック (HCLK)
  - メインクロック (MCLK)
  - PLL クロック (PCLK)
  - マシナクロック ( $\phi$ )
- 

### ■ クロックの概要

クロック生成部には、発振クロックを生成する発振回路が含まれています。外部発振器は、この回路に接続します。発振クロックの供給は、クロック生成部に外部クロックを入力することによっても行うことができます。

クロック生成部にはまた、PLL クロック逡倍回路 (発振クロックの逡倍である 4 つのクロックを生成する) も含まれています。

クロック生成部は、発振安定待ち時間および PLL クロック逡倍を制御します。また、クロックセクタでクロックを切り換えることによって動作クロックの動作も制御します。

#### ● 発振クロック (HCLK)

X0, X1 端子に振動子を接続するかまたは外部クロックを入力して発生させたクロックです。

#### ● メインクロック (MCLK)

発振クロックの 2 分周クロックであり、タイムベースタイマおよびクロックセクタへの入力クロックとなります。

#### ● PLL クロック (PCLK)

発振クロックを、PLL 逡倍回路 (PLL 発振回路) により逡倍したクロックです。4 種類 (1 逡倍 ~ 4 逡倍) のクロックを発生します。

#### ● マシナクロック ( $\phi$ )

CPU およびリソースの動作クロックです。マシナクロックの 1 周期が 1 マシナサイクルです。メインクロック (2 分周されたソースクロックが生成) および 4 種類のクロック (ソースクロックの逡倍) を設定します。

---

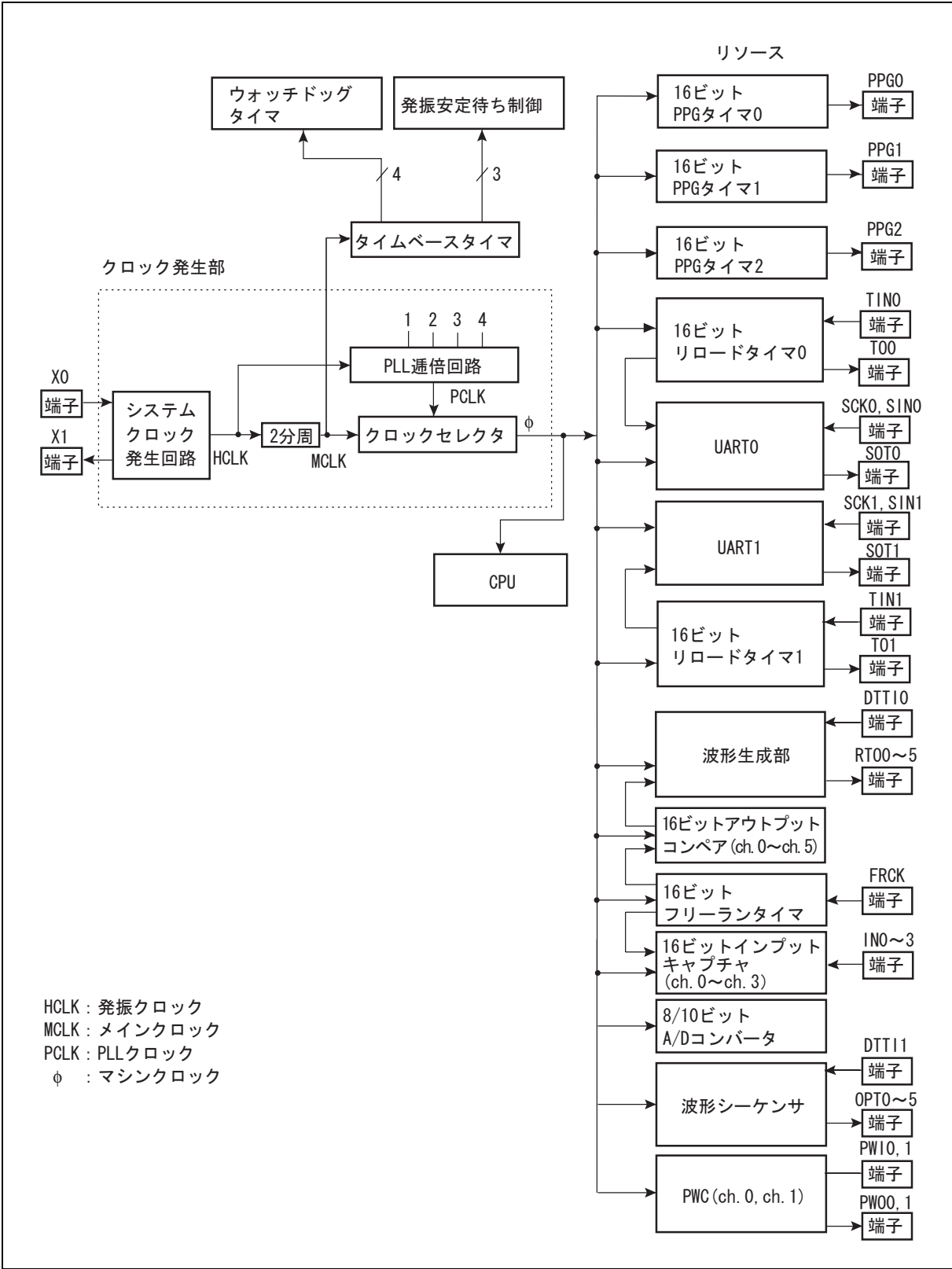
< 注意事項 > 動作電圧が 5V の場合、発振クロック周波数は 3MHz ~ 32MHz の範囲で使用できます。CPU およびリソースの最大動作周波数は 16MHz です。最大動作周波数を超える逡倍率を設定した場合、デバイスは正常に動作しません。例えば、発振クロック周波数が 16MHz の場合には、1 逡倍が設定可能です。

---

■ クロック供給マップ

クロック供給マップを図 5.1-1 に示します。

図 5.1-1 クロック供給マップ





## 5.2 クロック発生部のブロックダイアグラム

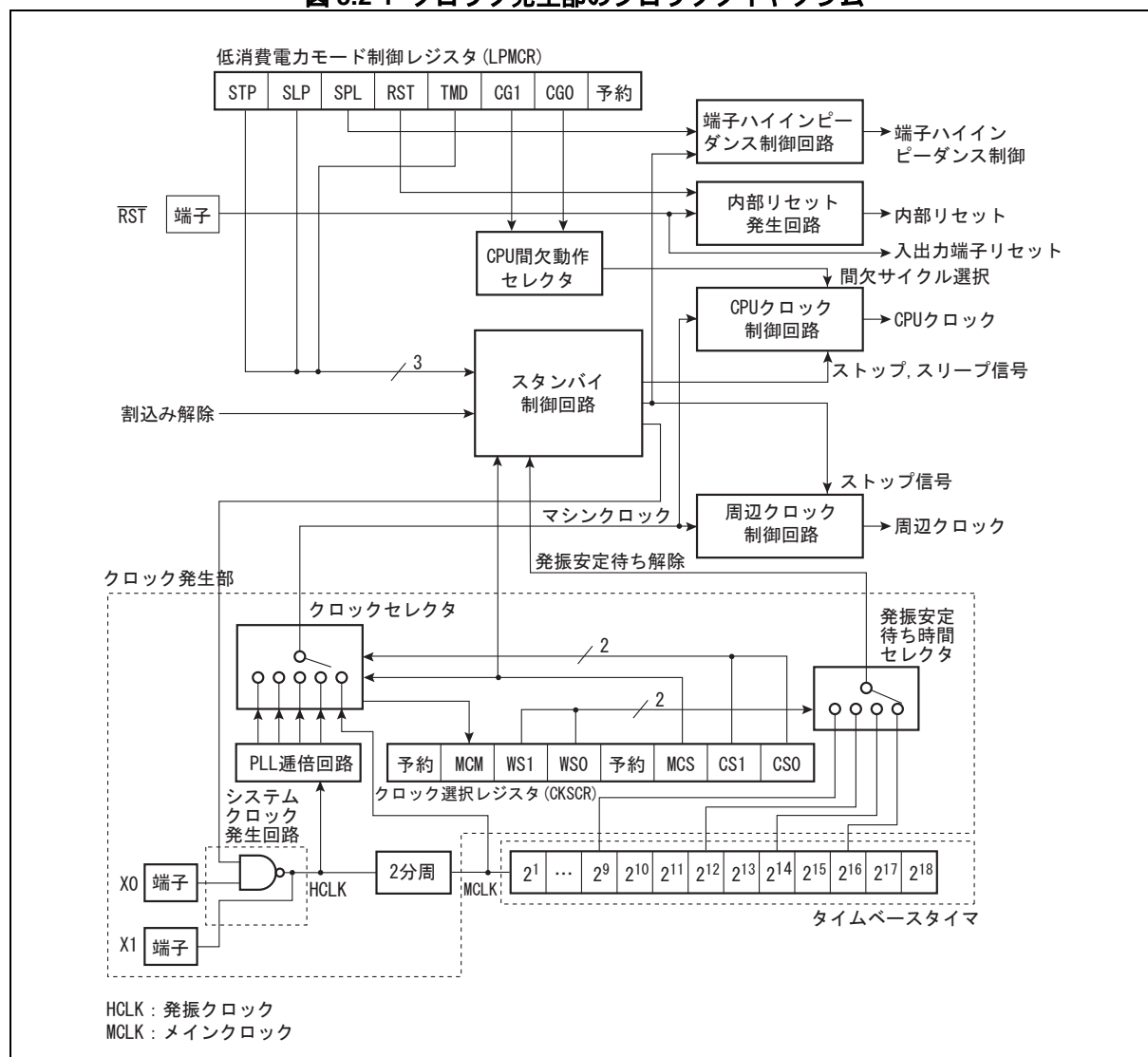
クロック発生部は、以下の5種類のブロックから構成されています。

- システムクロック発生回路
- PLL 通倍回路
- クロックセクタ
- クロック選択レジスタ (CKSCR)
- 発振安定待ち時間セクタ

### ■ クロック発生部のブロックダイアグラム

クロック発生部のブロックダイアグラムを図 5.2-1 に示します。

図 5.2-1 クロック発生部のブロックダイアグラム



● システムクロック発生回路

X0, X1 端子に振動子を接続するかまたは外部クロックを入力し、発振クロックを発生します。

● PLL 逡倍回路

発振クロックを逡倍し、クロックセクタへ供給します。

● クロックセクタ

メインクロックおよび4種類のPLLクロックから、CPUクロック制御回路および周辺クロック制御回路へ供給するクロックを設定します。

● クロック選択レジスタ (CKSCR)

マシクロック (メインクロック /PLL クロック) の切換え、発振安定待ち時間およびPLLクロックの逡倍率を設定します。

● 発振安定待ち時間セクタ

ストップモードの解除またはウォッチドッグタイマがリセットされた場合の発振クロックの発振安定待ち時間を設定します。4種類のタイムベースタイマ出力から発振安定待ち時間を設定します。

## 5.3 クロック選択レジスタ (CKSCR)

クロック選択レジスタ (CKSCR) は、マシナクロックの切換え、発振安定待ち時間の設定および PLL クロックの通倍率を設定するレジスタです。

### ■ クロック選択レジスタ (CKSCR)

クロック選択レジスタ (CKSCR) の構成を図 5.3-1 に示します。

また、クロック選択レジスタ (CKSCR) の各ビットの機能説明を表 5.3-1 に示します。

図 5.3-1 クロック選択レジスタ (CKSCR) の構成

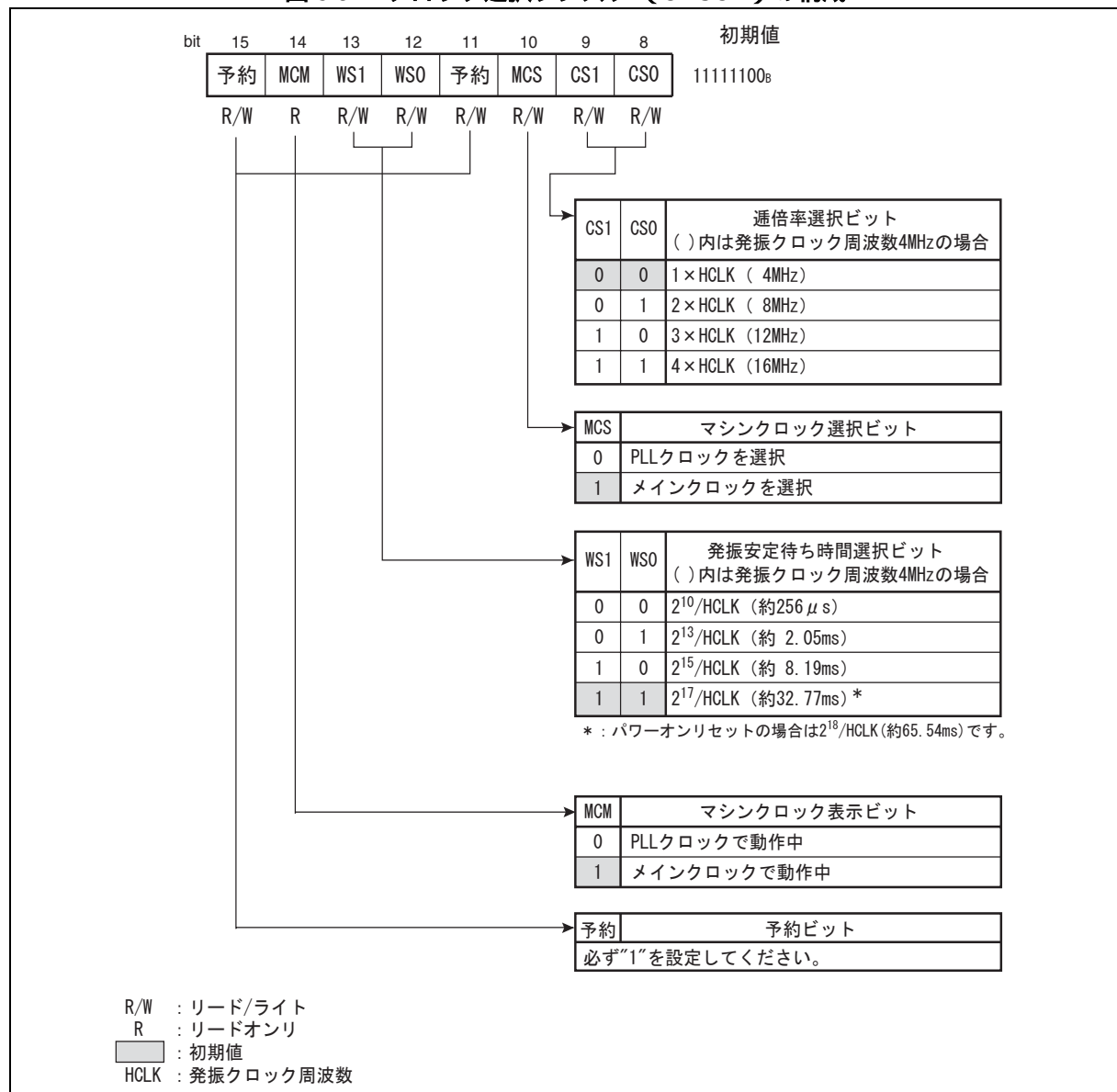


表 5.3-1 クロック選択レジスタ (CKSCR) の各ビットの機能説明

ビット名		機 能
bit15, bit11	予約: 予約ビット	<ul style="list-style-type: none"> <li>必ず "1" を設定してください。</li> </ul>
bit14	MCM: マシナクロック 表示ビット	<ul style="list-style-type: none"> <li>マシナクロックとして、メインクロックと PLL クロックのどちらが設定されているかを示します。</li> <li>このビットが "0" の場合: PLL クロックが設定されていることを示します。</li> <li>このビットが "1" の場合: メインクロックが設定されていることを示します。</li> <li>マシナクロック選択ビット (MCS) に "0" を設定後に, "1" がセットされている場合は, PLL クロックの発振安定待ち時間中であることを示します。</li> <li>書込みは動作に影響ありません。</li> </ul>
bit13, bit12	WS1, WS0: 発振安定待ち 時間選択ビット	<ul style="list-style-type: none"> <li>ストップモードを外部割込みにより解除した場合の、発振クロックの発振安定待ち時間を設定します。</li> <li>リセット要因により "11<sub>B</sub>" に初期化されます。</li> <li>発振安定待ち時間は、使用する振動子に合わせて設定してください。</li> </ul> <p>(注意事項)</p> <p>「4.2 リセット要因と発振安定待ち時間」をご参照ください。</p> <p>(参考)</p> <p>PLL 発振安定化待ち時間は、<math>2^{14}/HCLK</math> 固定です。</p>
bit10	MCS: マシナクロック 設定ビット	<ul style="list-style-type: none"> <li>マシナクロックとして、メインクロックまたは PLL クロックを設定します。</li> <li>このビットが "0" の場合: PLL クロックを設定します。</li> <li>このビットが "1" の場合: メインクロックを設定します。</li> <li>"1" が設定されている場合に, "0" を設定すると, PLL クロックの発振安定待ち時間が発生するため、タイムベースタイマカウンタおよびタイムベースタイマ制御レジスタ (TBTC) の割込み要求フラグビット (TBOF) は "0" にクリアされます。</li> <li>PLL クロックの発振安定待ち時間は、<math>2^{14}/HCLK</math> に固定です。発振クロック周波数が 4MHz の場合、発振安定待ち時間は約 4ms です。</li> <li>メインクロックを設定した場合のマシナクロックは、発振クロックを 2 分周したクロックとなります。発振クロック周波数が 4MHz の場合、マシナクロック周波数、2MHz です。</li> <li>リセットで "1" に初期化されます。</li> </ul> <p>(注意事項)</p> <p>"1" が設定されている場合に, "0" を設定するには、タイムベースタイマ制御レジスタ (TBTC) の割込み要求許可ビット (TBIE) または割込みレベルマスクレジスタ (ILM) により、タイムベースタイマ割込み要求が禁止されている状態に限られます。</p> <p>"1" を設定してから 8 マシンサイクルの間は、本ビットへ "0" を設定できない場合があります。本ビットへの設定は、8 マシンサイクル経過後に行ってください。</p>
bit9, bit8	CS1, CS0: 通倍率設定 ビット	<ul style="list-style-type: none"> <li>PLL クロックの通倍率を設定します。</li> <li>4 種類の通倍率から設定できます。</li> <li>リセットで "00<sub>B</sub>" に初期化されます。</li> </ul> <p>(注意事項)</p> <p>マシナクロック選択ビット (MCS) または、マシナクロック表示ビット (MCM) に "0" が設定されている場合は設定できません。マシナクロック選択ビット (MCS) に "1" を設定後に設定してください。</p>

HCLK: 発振クロック周波数

## 5.4 クロックモード

---

クロックモードとしては、メインクロックモードと PLL クロックモードがあります。

---

### ■ メインクロックモードと PLL クロックモード

#### ● メインクロックモード

メインクロックモード時は、CPU およびリソースのマシナクロックとしてメインクロック（発振クロックの 2 分周）を使用し、PLL クロックを停止させます。

#### ● PLL クロックモード

PLL クロックモード時は、CPU およびリソースのマシナクロックとして PLL クロックを使用します。PLL クロックの逡倍率は、クロック選択レジスタ（CKSCR）の逡倍率選択ビット（CS1, CS0:bit9, bit8）で設定します。

### ■ クロックモードの遷移

クロック選択レジスタ（CKSCR）のマシナクロック選択ビット（MCS:bit10）の設定により、メインクロックモードまたは PLL クロックモードへ遷移します。

#### ● メインクロックモードから PLL クロックモードへの遷移

クロック選択レジスタ（CKSCR）のマシナクロック選択ビット（MCS:bit10）を "1" から "0" に設定した場合は、PLL クロックの発振安定待ち時間（ $2^{14}/HCLK$ ）後、メインクロックから PLL クロックに切り換わります。

#### ● PLL クロックモードからメインクロックモードへの遷移

クロック選択レジスタ（CKSCR）のマシナクロック選択ビット（MCS:bit10）を "0" から "1" に設定した場合は、PLL クロックとメインクロックのエッジが一致するタイミング（1 ~ 8PLL クロック経過後）で、PLL クロックからメインクロックに切り換わります。

---

< 注意事項 > マシナクロックの切換え後、リソースを動作させる場合は、クロック選択レジスタ（CKSCR）のマシナクロック表示ビット（MCM:bit14）により、マシナクロックが切り換わったことを確認したあと動作させてください。  
クロックモードを切り換えた場合、切換えが完了するまではほかのクロックモードおよび低消費電力モードへは切り換えないようにしてください。  
切換えの完了はクロック選択レジスタ（CKSCR）の MCM ビットを参照して確認してください。切換えが完了する前に、ほかのクロックモードおよび低消費電力モードへの切換えを行った場合、切り換わらない場合があります。

---

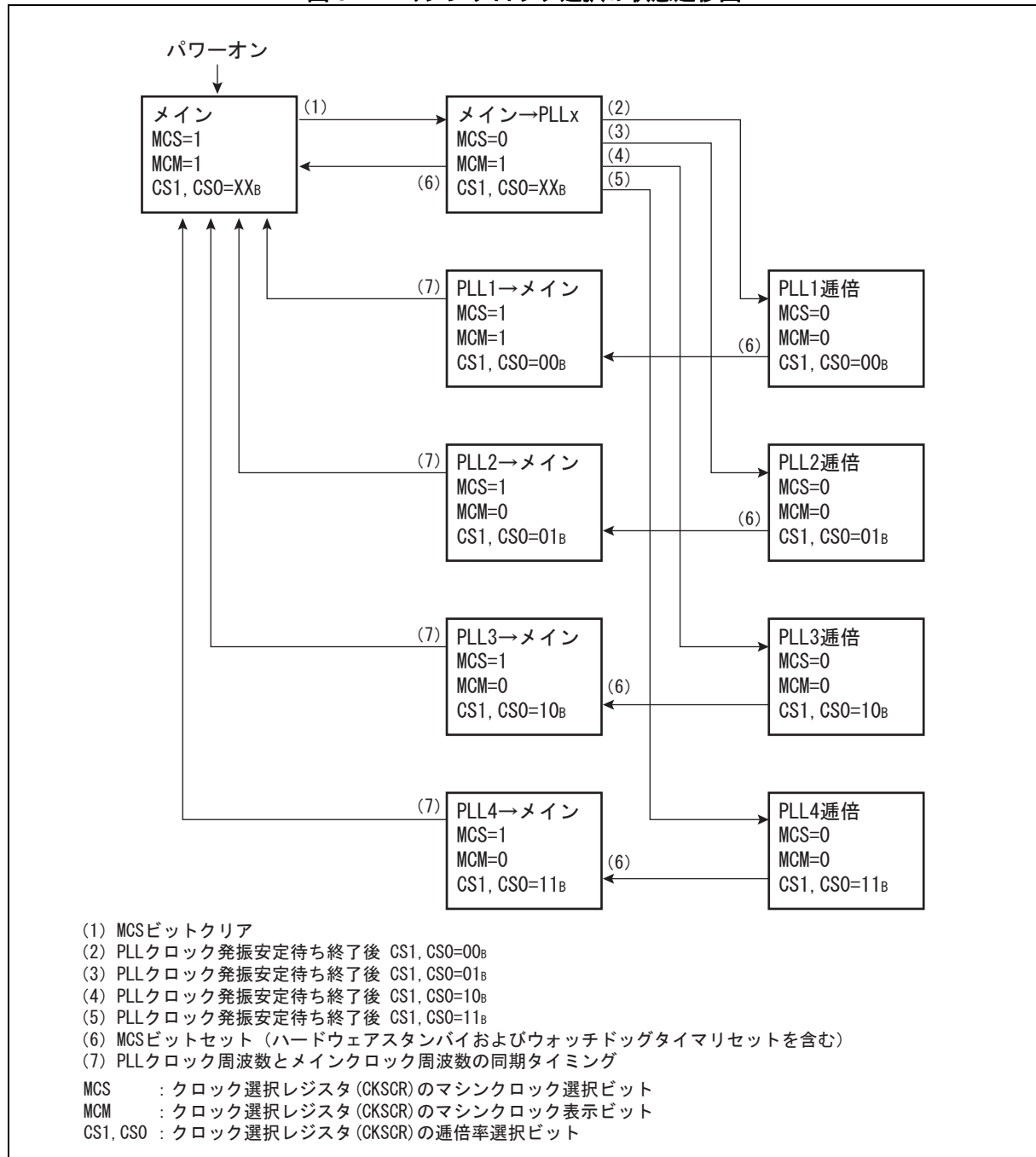
### ■ PLL クロック逡倍率の設定

クロック選択レジスタ（CKSCR）の逡倍率選択ビット（CS1, CS0:bit9, bit8）に "00<sub>B</sub>" ~ "11<sub>B</sub>" を設定し、1 ~ 4 逡倍の 4 種類から PLL クロック逡倍率を設定します。

## ■ マシニングロック

メインクロックまたは PLL クロックが、マシニングロックとなります。マシニングロックは CPU およびリソースの動作クロックです。メインクロックまたは PLL クロックは、クロック選択レジスタ (CKSCR) のマシニングロック選択ビット (MCS) で設定します。マシニングロックを切り換えることによって発生する状態遷移を図 5.4-1 に示します。

図 5.4-1 マシニングロック選択の状態遷移図



< 注意事項 > マシニングロックの初期値は、メインクロック (CKSCR:MCS=1) です。

## 5.5 発振安定待ち時間

電源投入，ストップモードの解除およびウォッチドッグタイマリセットの場合は，発振停止から動作が開始するため，発振開始後に発振安定待ち時間が必要です。また，メインクロックから PLL クロックへ切り換える場合も，PLL クロックの発振開始後に発振安定待ち時間が必要です。

### ■ 発振安定待ち時間

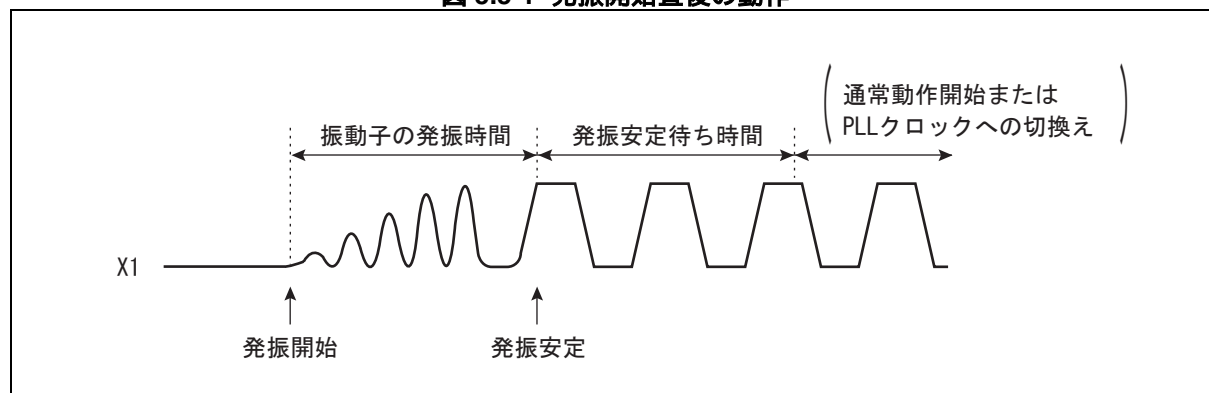
セラミック発振子および水晶発振子では一般に，発振を開始してから固有周波数に安定するまでに，数ミリ秒から数十ミリ秒の発振安定化待ち時間が必要です。

CPU 動作は，発振が完全に安定した後で行う必要があるため，CPU の動作は発振安定化待ち時間が経過した後で行われます。

発振安定化待ち時間は発振子のタイプ（水晶，セラミックなど）によって異なるので，使用する発振子に適切な発振安定化待ち時間を選択しなければなりません。発振安定化待ち時間は，クロック選択レジスタ（CKSCR）を設定することによって選択できます。

メインクロックから PLL クロックへの切換え時，CPU は発振安定化待ち時間中はメインクロックで動作を継続します。動作クロックの PLL クロックへの切換えは，この時間が経過した後で行われます。発振開始後の動作を図 5.5-1 に示します。

図 5.5-1 発振開始直後の動作



## 5.6 振動子と外部クロックの接続

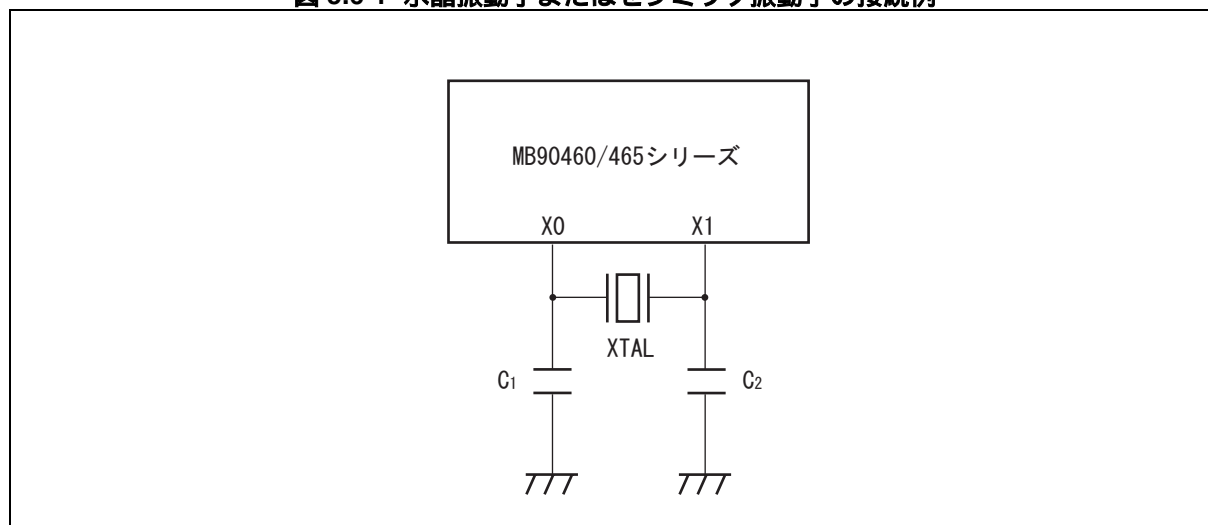
MB90460/465 シリーズは、システムクロック発生回路を内蔵しており、X0, X1 端子に振動子を接続するかまたは外部クロックを入力できます。

### ■ 振動子と外部クロックの接続

#### ● 水晶振動子またはセラミック振動子の接続例

水晶振動子またはセラミック振動子は、図 5.6-1 の例で示すように接続してください。

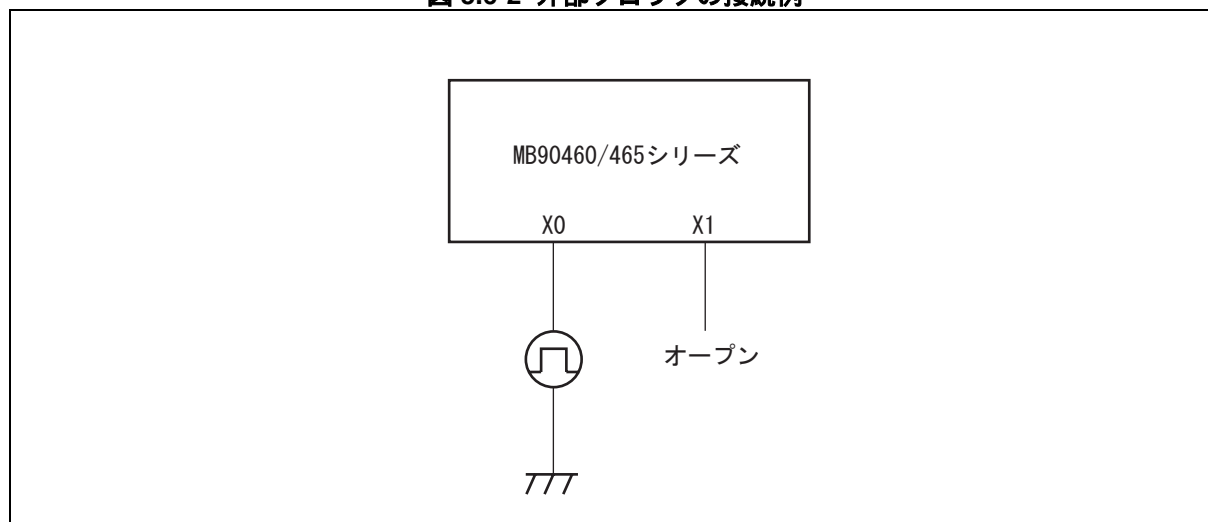
図 5.6-1 水晶振動子またはセラミック振動子の接続例



#### ● 外部クロックの接続例

図 5.6-2 の例に示すように、外部クロックは X0 端子に接続し、X1 端子はオープン（開放）にしてください。

図 5.6-2 外部クロックの接続例







# 第6章

---

## 低消費電力モード

MB90460/465 シリーズ低消費電力モードについて説明します。

- 6.1 低消費電力モードの概要
- 6.2 低消費電力制御回路のブロックダイアグラム
- 6.3 低消費電力モード制御レジスタ（LPMCR）
- 6.4 CPU 間欠動作モード
- 6.5 スタンバイモード
- 6.6 状態遷移図
- 6.7 スタンバイモード，リセットの端子状態
- 6.8 低消費電力モード使用上の注意

## 6.1 低消費電力モードの概要

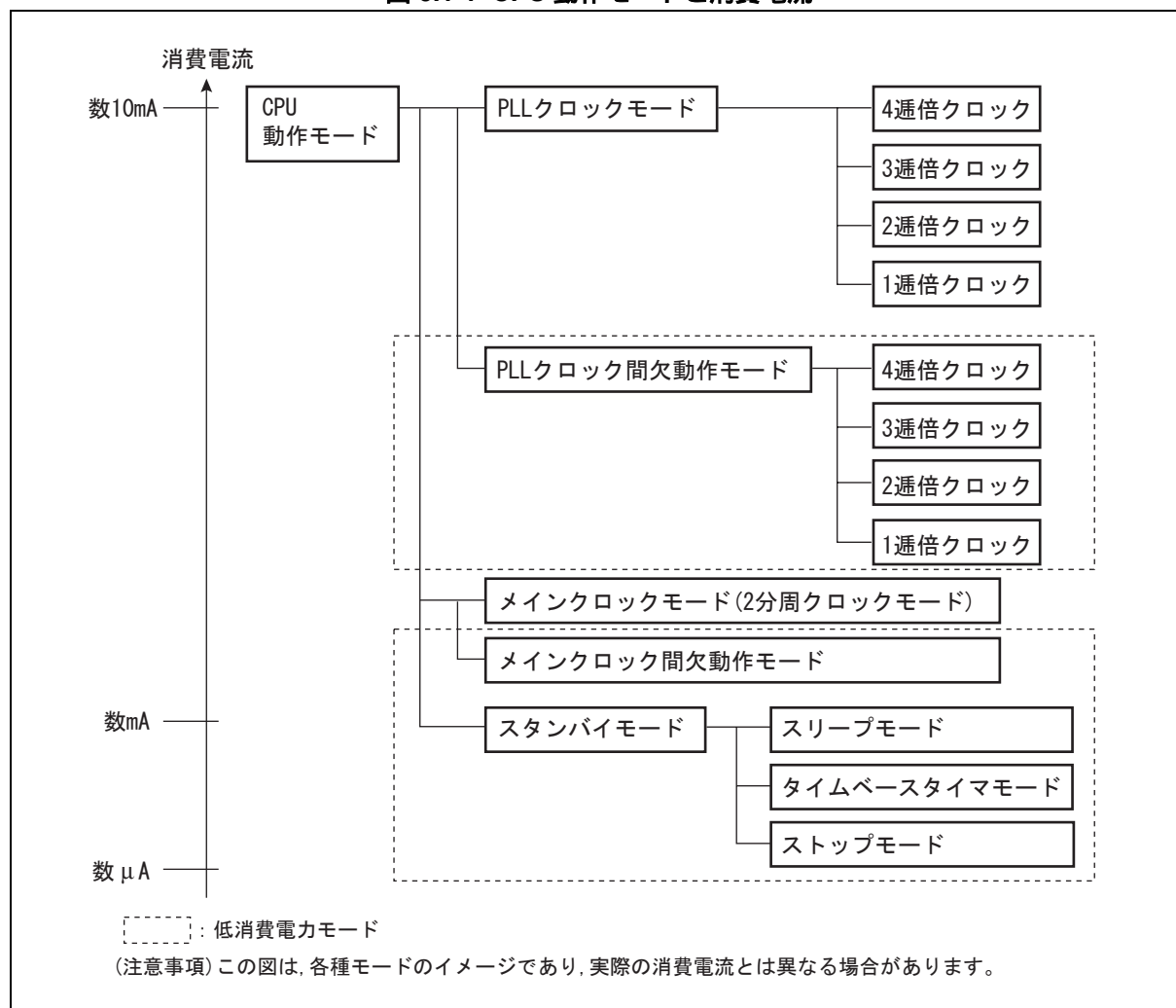
MB90460/465 シリーズは、動作クロックの設定とクロックの動作制御により、以下の低消費電力モードがあります。

- CPU 間欠動作モード（PLL クロック間欠動作モードおよびメインクロック間欠動作モード）
- スタンバイモード（スリープモード、タイムベースタイムモードおよびストップモード）

### ■ CPU 動作モードと消費電流

CPU 動作モードと消費電流の関係を図 6.1-1 に示します。

図 6.1-1 CPU 動作モードと消費電流



## ■ クロックモード

### ● PLL クロックモード

PLL クロックは、発振クロック（HCLK）を 1 ～ 4 通倍（CKSCR レジスタの CS1, CS0 ビットで設定）した周波数のクロックであり、CPU およびリソースを動作させます。

### ● メインクロックモード

メインクロックは、発振クロック（HCLK）を 2 分周した周波数のクロックであり、CPU およびリソースを動作させます。メインクロックモード時は、PLL 通倍回路が停止します。

---

< 参照 >      クロックモードの詳細については、「5.1 クロックの概要」をご参照ください。

---

## ■ CPU 間欠動作モード

CPU 間欠動作モード時は、CPU が間欠的に動作すると同時に、高速クロックパルスがリソースへ供給され、消費電力が抑えられます。CPU 間欠動作モード時、間欠クロックパルスが CPU に供給されるのは、レジスタ、内部メモリ、リソースまたは外部ユニットをアクセスするときのみです。

## ■ スタンバイモード

スタンバイモードにおける低消費電力制御回路は、CPU へのクロック供給を停止（スリープモード）、CPU およびリソースへのクロック供給を停止（タイムベースタイマモード）または発振クロックを完全に停止（ストップモード）することにより消費電力を抑えます。

### ● PLL スリープモード

PLL スリープモードが起動され、PLL クロックモードになると、CPU の動作クロックが停止します。ほかのリソースは、PLL クロックで動作を継続します。

### ● メインスリープモード

メインスリープモードが起動され、メインクロックモードになると、CPU の動作クロックが停止します。ほかのリソースは、メインクロックで動作を継続します。

### ● PLL タイムベースタイマモード

PLL タイムベースタイマモード時は、発振クロック、PLL クロックおよびタイムベースタイマ以外の動作が停止します。

### ● メインタイムベースタイマモード

メインタイムベースタイマモード時は、発振クロック、メインクロックおよびタイムベースタイマ以外の動作が停止します。

### ● ストップモード

ストップモード時は、発振クロックが停止し、すべての機能が停止します。

ストップモード時は発振クロックが停止するため、最も低消費電力でデータを保持できます。

---

< 注意事項 > クロックモードを切り換えた場合，切換えが完了するまではほかのクロックモードおよび低消費電力モードへ切り換えないようにしてください。  
切換えの完了はクロック選択レジスタ（CKSCR）の MCM ビットを参照して確認してください。切換えが完了する前に，ほかのクロックモードおよび低消費電力モードへの切換えを行った場合，切り換わらない場合があります。

---

## 6.2 低消費電力制御回路のブロックダイアグラム

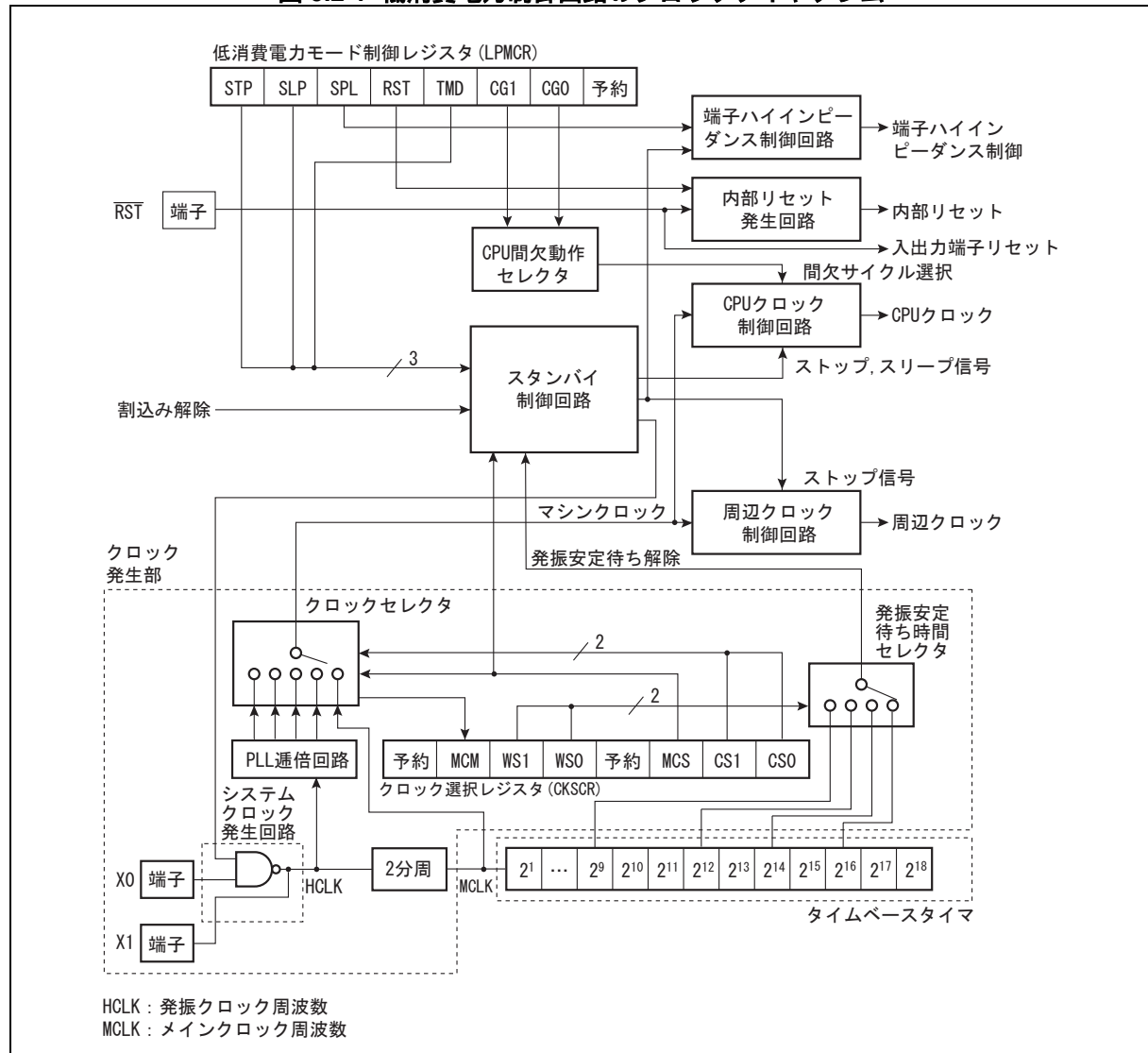
低消費電力制御回路は、以下の7つのブロックから構成されています。

- CPU 間欠動作セクタ
- 待機クロック制御回路
- CPU クロック制御回
- 周辺クロック制御回路
- 端子ハイインピーダンス制御回路
- 内部リセット発生回路
- 低消費電力モード制御レジスタ (LPMCR)

### ■ 低消費電力制御回路のブロックダイアグラム

低消費電力制御回路のブロックダイアグラムを図 6.2-1 に示します。

図 6.2-1 低消費電力制御回路のブロックダイアグラム



- CPU 間欠動作セレクト

CPU 間欠動作モード時に動作中の CPU 動作クロック停止サイクル数を設定します。

- スタンバイ制御回路

CPU クロック制御回路，周辺クロック制御回路および端子ハイインピーダンス制御回路を制御し，低消費電力モードへの遷移および解除を行います。

- CPU クロック制御回路

CPU に供給するクロックを制御します。この回路は，周辺クロック制御のリソースへ供給されるクロックを制御します。

- 周辺クロック制御回路

リソースへ供給するクロックを制御します。

- 端子ハイインピーダンス制御回路

タイムベースタイムモードおよびストップモード時に，設定により入出力端子をハイインピーダンスにします。

プルアップ抵抗を接続している入出力端子は，ストップモード時にプルアップ抵抗を切離します。

- 内部リセット生成回路

内部リセット信号を発生させます。

- 低消費電力モード制御レジスタ (LPMCR)

低消費電力モードへの遷移 / 解除および CPU 間欠動作モードで動作中の CPU 動作クロック停止サイクル数などを設定します。

## 6.3 低消費電力モード制御レジスタ（LPMCR）

低消費電力モード制御レジスタ（LPMCR）は、低消費電力モードへの遷移 / 解除および CPU 間欠動作モードで動作中の CPU 動作クロック停止サイクル数を設定します。

### ■ 低消費電力モード制御レジスタ（LPMCR）

低消費電力モード制御レジスタ（LPMCR）の構成を図 6.3-1 に示します。

図 6.3-1 低消費電力モード制御レジスタ（LPMCR）の構成

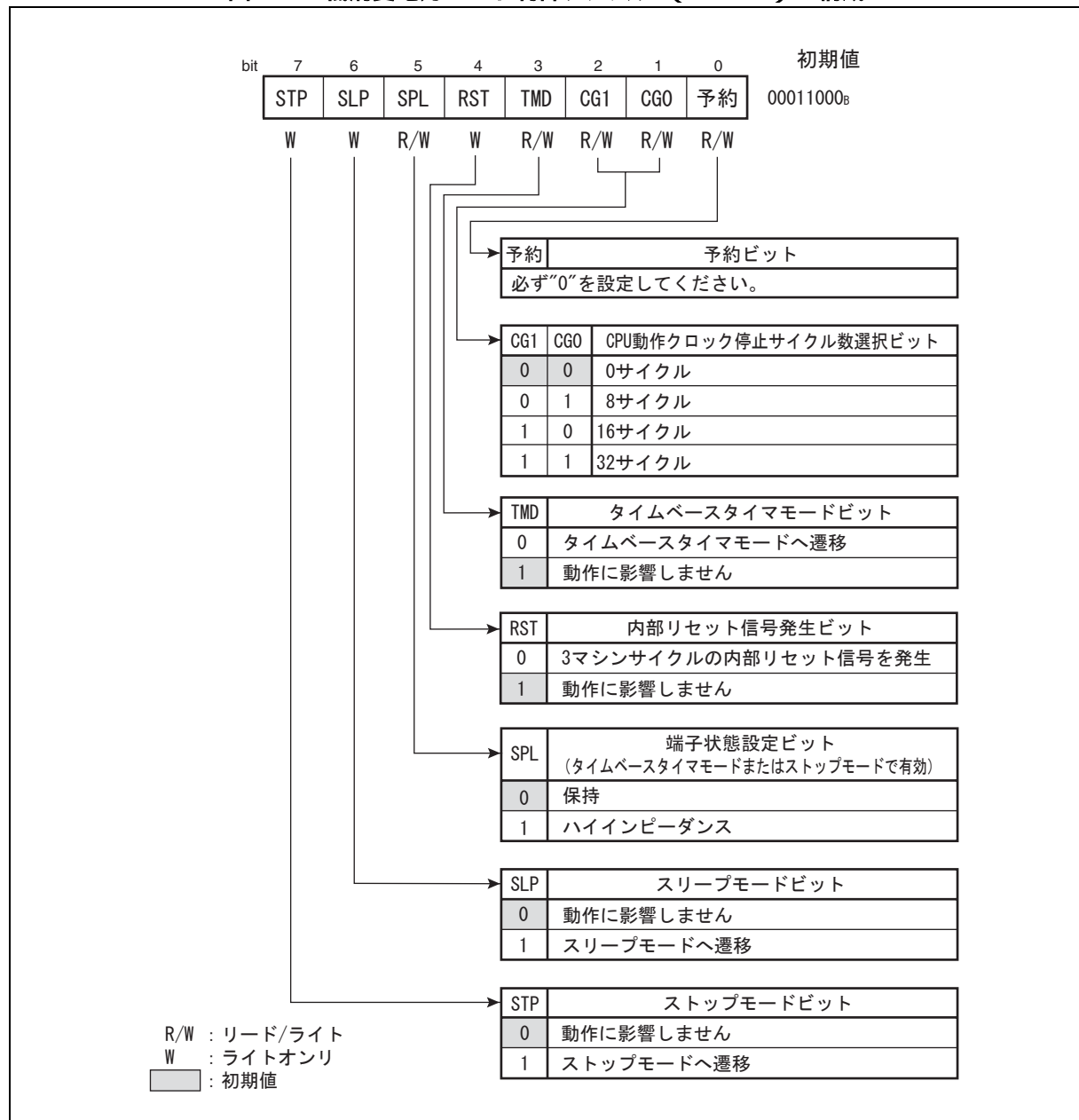




表 6.3-1 低消費電力モード制御レジスタ (LPMCR) の各ビットの機能説明

ビット名		機 能
bit7	STP: ストップモード ビット	<ul style="list-style-type: none"> <li>• ストップモードを設定するビットです。</li> <li>• このビットに "0" を設定した場合: 動作に影響しません。</li> <li>• このビットに "1" を設定した場合: ストップモードへ遷移します。</li> <li>• 外部リセットまたはハードウェア割込み出力により, "0" にクリアされます。</li> <li>• 読出し値は常に, "0" です。</li> </ul>
bit6	SLP: スリープモード ビット	<ul style="list-style-type: none"> <li>• スリープモードを設定するビットです。</li> <li>• このビットに "0" を設定した場合: 動作に影響しません。</li> <li>• このビットに "1" を設定した場合: スリープモードへ遷移します。</li> <li>• 外部リセットまたはハードウェア割込み出力により, "0" にクリアされます。</li> <li>• 読出し値は常に, "0" です。</li> </ul>
bit5	SPL: 端子状態設定ビット (タイムベースタイ マモードまたはス トップモードで有効)	<ul style="list-style-type: none"> <li>• タイムベースタイマモードまたはストップモード時の端子状態を設定するビットです。</li> <li>• このビットに "0" を設定した場合: 入出力端子のレベルを保持します。</li> <li>• このビットに "1" を設定した場合: 入出力端子をハイインピーダンスにします。</li> <li>• 外部リセットで "0" にクリアされます。</li> </ul>
bit4	RST: 内部リセット信号発 生ビット	<ul style="list-style-type: none"> <li>• 内部リセットを設定するビットです。</li> <li>• このビットに "0" を設定した場合: 3 マシンサイクルの内部リセット信号を発生します。</li> <li>• このビットに "1" を設定した場合: 動作に影響しません。</li> <li>• 読出し値は常に, "1" です。</li> </ul>
bit3	TMDX: タイムベースタイマ モードビット	<ul style="list-style-type: none"> <li>• タイムベースタイマモードへの遷移を設定するビットです。</li> <li>• このビットに "0" を設定した場合: タイムベースタイマモードへに遷移します。</li> <li>• このビットに "1" を設定した場合: 動作に影響しません。</li> <li>• 外部リセットまたはタイムベースタイマ割込み出力により, "1" に初期化されます。</li> <li>• 読出し値は "1" です。</li> </ul>
bit2, bit1	CG1, CG0: CPU 動作クロック 停止サイクル数選択 ビット	<ul style="list-style-type: none"> <li>• CPU 間欠動作モードで動作中の CPU 動作クロック停止サイクル数を設定するビットです。</li> <li>• 1 命令ごとに設定サイクル数の CPU クロック供給を停止します。</li> <li>• 4 種類のサイクル数から設定します。</li> <li>• リセットにより "00<sub>B</sub>" に初期化されます。</li> </ul>
bit0	予約: 予約ビット	<ul style="list-style-type: none"> <li>• 必ず "0" を設定してください。</li> </ul>

## ■ 低消費電力モード制御レジスタへのアクセス

低消費電力モード制御レジスタを設定するには、表 6.3-2 に示す命令を使用してください。表 6.3-2 に示す命令以外で低消費電力モードへ遷移した場合は、動作を保証できません。

ワード長で低消費電力モード制御レジスタへ設定する場合は、低消費電力制御回路（0000A0<sub>H</sub>, 0000A1<sub>H</sub>）にワードアクセスしてください。

表 6.3-2 低消費電力モードへ遷移する場合に使用する命令一覧

MOV io, #imm8	MOV dir, #imm8	MOV eam, #imm8	MOV eam, Ri
MOV io, A	MOV dir, A	MOV addr16, A	MOV eam, A
MOV @RLi+disp8, A			
MOVW io, #imm16	MOVW dir, #imm16	MOVW eam, #imm16	MOVW eam, RWi
MOVW io, A	MOVW dir, A	MOVW addr16, A	MOVW eam, A
MOVW @RLi+disp8, A			
SETB io:bp	SETB dir:bp	SETB addr16:bp	
CLRB io:bp	CLRB dir:bp	CLRB addr16:bp	

## ■ スタンバイモードの優先順位

ストップモード（LPMCR:STP）、スリープモード（LPMCR:SLP）およびタイムベースタイマモード（LPMCR:TMD）を同時に設定した場合は、ストップモード＞タイムベースタイマモード＞スリープモードの優先順位で遷移します。

## 6.4 CPU 間欠動作モード

CPU 間欠動作モードは、CPU の間欠動作を行うと同時にリソースの高速動作を継続させるために使用されます。CPU 間欠動作を行うことにより消費電力を抑えることができます。

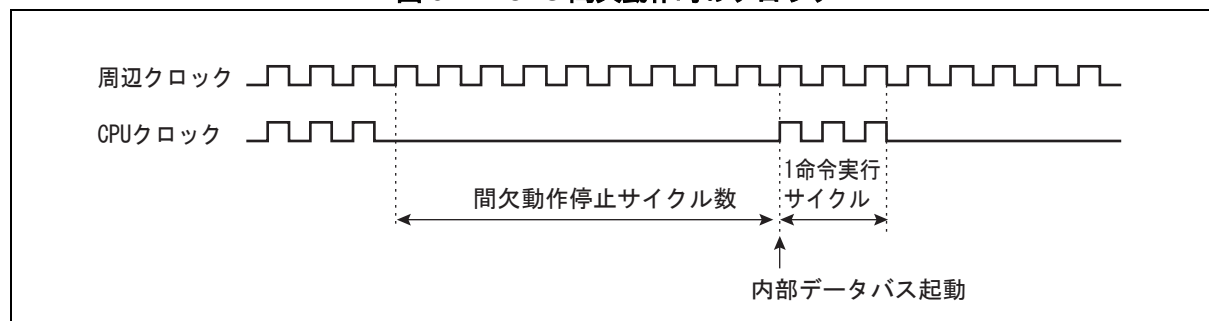
### ■ CPU 間欠動作モード

CPU 間欠動作モードは、CPU へのクロック供給をある一定期間停止します。この停止が発生するのは、レジスタ、内部メモリ (ROM および RAM)、I/O およびリソースをアクセスする各命令が実行された後です。したがって、内部バスサイクルが起動される際には遅延が生じます。ある一定速度の周辺クロックパルスがリソースに供給されている間、CPU 実行速度が下げられ、低消費電力での処理が可能になります。

- 低消費電力モード制御レジスタ (LPMCR) の CG1 ビットと CG0 ビットは、CPU に供給されるクロックの 1 停止サイクル当りのクロックパルス数を選択するために使用します。
- CPU 間欠モード時の命令実行時間は、次のようにして算出できます。レジスタ、内部メモリ、リソースおよび外部バスをアクセスする命令が実行された回数に 1 停止サイクル当りのクロックパルス数を掛けることによって、補正値を求めます。この補正値を正常実行時間に加算します。

CPU 間欠動作モード時の動作クロックパルスを図 6.4-1 に示します。

図 6.4-1 CPU 間欠動作時のクロック



## 6.5 スタンバイモード

スタンバイモードとしては、スリープモード（PLL スリープモード，メインスリープモード），タイムベースタイマモード，ストップモードがあります。

### ■ スタンバイモードでの動作状態

スタンバイモードの動作状態の概要を表 6.5-1 に示します。

表 6.5-1 スタンバイモードの動作状態

スタンバイモード		遷移条件	発振 クロック	マシン クロック	CPU	リソース	端子	解除方法	
スリープ モード	PLL スリー プモード	MCS=0 SLP=1	動作	動作	停止	動作	動作	外部リセッ トまたは 割込み	
	メインスリー プモード	MCS=1 SLP=1							
タイム ベース タイマ モード	タイムベース タイマモード ( SPL=0 )	TMD=0		停止		停止	停止 *  	保持	外部リセッ トまたは 割込み * <sup>1</sup>
	タイムベース タイマモード ( SPL=1 )	TMD=0						Hi-Z	
ストップ モード	ストップモー ド ( SPL=0 )	STP=1	停止			停止	停止	保持	外部リセッ トまたは 割込み * <sup>2</sup>
	ストップモー ド ( SPL=1 )	STP=1						Hi-Z	

\*1：タイムベースタイマ

\*2：外部割込み

SPL：低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット

SLP：低消費電力モード制御レジスタ（LPMCR）のスリープモードビット

STP：低消費電力モード制御レジスタ（LPMCR）のストップモードビット

TMD：低消費電力モード制御レジスタ（LPMCR）のタイムベースタイマモードビット

MCS：クロック選択レジスタ（CKSCR）のマシクロック選択ビット

Hi-Z：ハイインピーダンス

## 6.5.1 スリープモード

スリープモードは、CPU の動作クロックを停止させます。CPU 以外は動作を続けます。スリープモードへの遷移を設定すると、PLL クロックモードを設定している場合は PLL スリープモードへ遷移し、メインクロックモードを設定している場合はメインスリープモードへ遷移します。

### ■ スリープモードへの遷移

低消費電力モード制御レジスタ (LPMCR) のスリープモードビット (SLP:bit6) に "1" を、タイムベースタイマモードビット (TMD:bit3) に "1" を、ストップモードビット (STP:bit5) に "0" を設定した場合は、スリープモードへ遷移します。スリープモードへの遷移を設定すると、クロック選択レジスタ (CKSCR) のマシナクロック選択ビット (MCS:bit10) に "0" が設定されている場合は、PLL スリープモードへ遷移します。マシナクロック選択ビット (MCS:bit10) に "1" が設定されている場合は、メインスリープモードへ遷移します。

SLP ビットと STP ビットへの "1" の書込みと TMD ビットへの "0" の書込みが同時に行われると、STP/TMD ビット設定値は SLP ビット設定値を無効にするので、ストップモードまたはタイムベースタイマモードへの切り換えが発生します。

#### ● データ保持機能

スリープモード時は、専用レジスタ (アキュムレータや内部 RAM など) のデータを保持します。

専用レジスタの詳細は、「3.7 専用レジスタ」をご参照ください。

#### ● 割り込み要求時の動作

割り込み要求時に LPMCR の SLP ビットに "1" を書き込んでも、スリープモードへの切り換えは発生しません。CPU が割り込みを受け付けない場合は、CPU は当該命令のその次の命令を実行します。CPU が割り込みを受け付けた場合は、CPU 動作は直ちに割り込み処理ルーチンに分岐します。

#### ● ピンの状態

スリープモード時には、すべての端子 (バス入出力またはバス制御を行うために使用される端子は除く) は、スリープモードへの切り換えが発生する直前の状態を保持します。

## ■ スリープモードの解除

スリープモードは、外部リセットまたは割込みにより解除できます。

### ● リセットによるスリープモードの解除

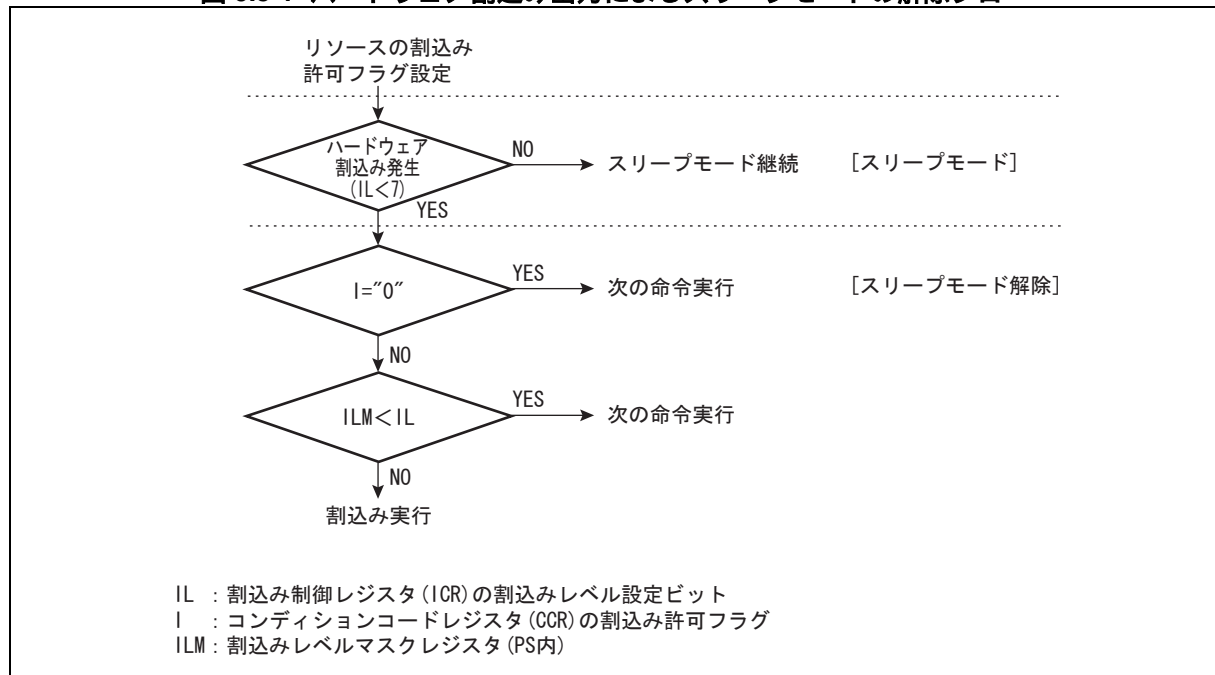
スリープモードがリセットによって解除されると、スリープモードが解除されると同時にリセット状態になります。

### ● 割込みによるスリープモードの解除

スリープモード時にリソースからレベル7より高い割込み要求が発生されると、スリープモードが解除されます。スリープモードが解除されると、CPU はこの割込みをほかの割込みと同様な方法で処理します。CPU は、コンディションコードレジスタ (CCR)、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) の設定に従って処理を実行します。この割込みが受け付けられた場合は、CPU は割込み処理を実行します。この割込みが受け付けられなかった場合は、CPU はスリープモードへの切換えが指定されている命令のすぐ次の命令を実行します。

割込みによるスリープモードの解除を図 6.5-1 に示します。

図 6.5-1 ハードウェア割込み出力によるスリープモードの解除フロー

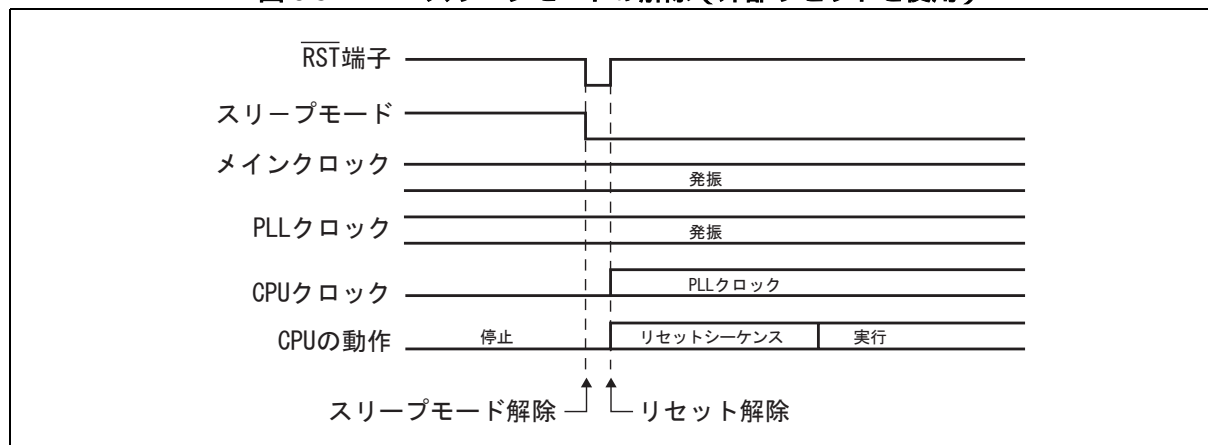


割込み処理が正常に実行されると、CPU はまず、スリープモードへの切換えが指定されている命令のすぐ次の命令を実行します。CPU は次に、割込み処理に移ります。

### ● 外部リセットによる PLL スリープモードからノーマルモードへの復帰

PLL スリープモード時には、メインクロックと PLL クロックがクロックパルスを生成します。外部リセットはクロック選択レジスタ (CKSCR) の MCS ビットを "1" に初期化しないので、PLL ロックモードが選択された状態 (CKSCR の MCS=0) のままとなります。外部リセットで PLL スリープモードからノーマルモードに復帰すると (すなわち、PLL スリープモードが解除された後)、CPU は直ちに PLL クロックで動作を開始します (図 6.5-2 を参照)。

図 6.5-2 PLL スリープモードの解除 (外部リセットを使用)



## 6.5.2 タイムベースタイマモード

タイムベースタイマモードは、発振クロックとタイムベースタイマ以外の動作を停止させます。

### ■ タイムベースタイマモードへの遷移

低消費電力モード制御レジスタ (LPMCR) のタイムベースタイマモードビット (TMD:bit3) に "0" を設定した場合は、タイムベースタイマモードへ遷移します。

LPMCR の TMD ビットと STP ビットにそれぞれ "0" を書き込むと、タイムベースタイマモードへの切換えが発生します。

この時点でクロック選択レジスタ (CKSCR) の MCS ビットに "0" が設定されている場合は、PLL タイムベースタイマモードになり、クロック選択レジスタ (CKSCR) の MCS ビットに "1" が設定されている場合は、メインタイムベースタイマモードになります。

TMD ビットと STP ビットへの "0" の書き込みが同時に行われると、STP ビット設定値は TMD ビットの設定値を無効にするので、ストップモードへの切換えが発生します。

#### ● データ保持機能

タイムベースタイマモード時は、専用レジスタと (アキュムレータや内部 RAM など) のデータを保持します。

専用レジスタの詳細は、「3.7 専用レジスタ」をご参照ください。

#### ● 割込み要求時の動作

割込み要求時に LPMCR の TMD ビットに "0" を書き込んでも、タイムベースタイマモードへの切換えは発生しません。

#### ● 端子の状態

タイムベースタイマモードへの切換えが発生する直前の状態を外部端子に保持させるかまたはこのモードへの切換えによって外部端子をハイインピーダンスに設定するかを選択に関する制御は、LPMCR の SPL ビットで行うことができます。



## ■ タイムベースタイマモードの解除

タイムベースタイマモードは、外部リセットまたはタイムベースタイマ割込みで解除できます。

### ● リセットによるタイムベースタイマモードの解除

タイムベースタイマモードがリセットによって解除されると、タイムベースタイマモードから解除された後でリセット状態になります。

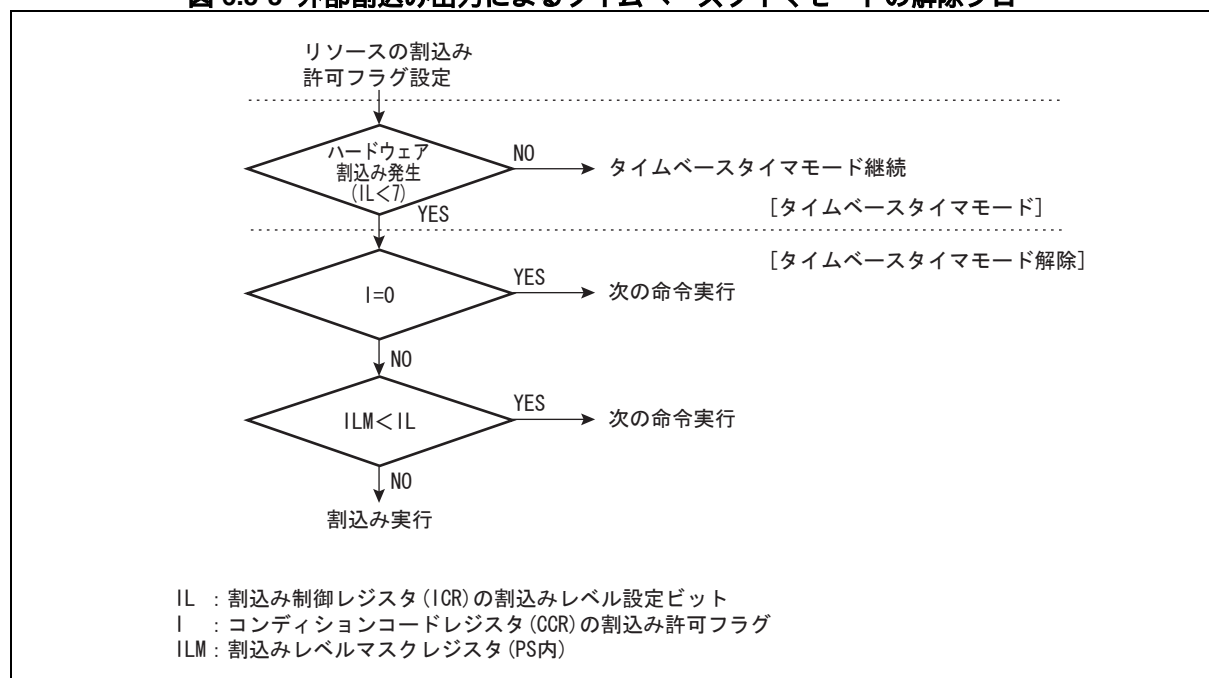
### ● 外部リセットによるタイムベースタイマモードの解除

外部リセットはクロック選択レジスタ (CKSCR) の MCS ビットを "1" に初期化しないため、PLL クロックモードが選択された状態 (CKSCR の MCS=0) のままとなるかまたはメインクロックモードが選択された状態 (CKSCR の MCS=1) のままとなります。外部リセットでタイムベースタイマモードから復帰すると (すなわち、タイムベースタイマモードが解除された後)、CPU は直ちに PLL/ メインクロックで動作を開始します。外部リセットによるタイムベースタイマモードからノーマルモードへの復帰動作を図 6.5-3 に示します。

### ● タイムベースタイマ割込みによる解除

タイムベースタイマモード中に割込みレベルが 7 より強く設定されたタイムベースタイマ割込みが発生すると、低消費電力制御回路はタイムベースタイマモードを解除します。タイムベースタイマモードの解除後は、通常の割込み処理と同じ扱いとなります。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) の設定により割込みが受け付けられない場合は、タイムベースタイマモードに入る前の次の命令から処理を続行します。

図 6.5-3 外部割込み出力によるタイムベースタイマモードの解除フロー



### 6.5.3 ストップモード

---

ストップモードは、発振クロックを停止させます。すべての機能が停止しますので、最も低消費電力でデータを保持できます。

---

#### ■ ストップモードへの遷移

低消費電力モード制御レジスタ (LPMCR) のストップモードビット (STP:bit7) に "1" を設定した場合は、ストップモードへ遷移します。

この時点でクロック選択レジスタ (CKSCR) の MCS ビットに "0" が設定されている場合は、PLL ストップモードになり、クロック選択レジスタ (CKSCR) の MCS ビットに "1" が設定されている場合は、メインストップモードになります。

#### ● データ保持機能

ストップモード時は、専用レジスタと (アキュムレータや内部 RAM など) のデータを保持します。

専用レジスタの詳細は、「3.7 専用レジスタ」をご参照ください。

#### ● 割込み要求出力中の動作

割込み要求を出力している場合は、低消費電力モード制御レジスタ (LPMCR) のストップモードビット (STP:bit7) に "1" を設定しても、ストップモードへ遷移しません。

#### ● 端子状態の設定

ストップモードの入出力端子は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL:bit5) により、直前のレベルを保持するか、ハイインピーダンスにするかを設定します。

## ■ ストップモードの解除

ストップモードは、外部リセットまたはハードウェア割込み出力で解除できます。

### ● リセットによるストップモードの解除

ストップモードのリセットによって解除されると、ストップモードから解除された後、発振安定化待ちおよびリセット状態になります。リセットシーケンスには、発振安定化待ち時間経過後に移ります。

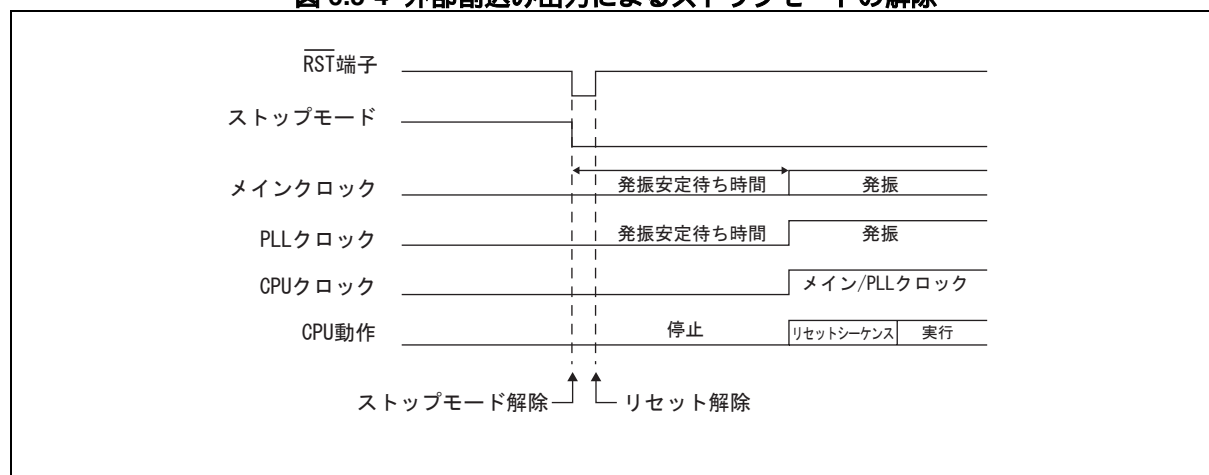
### ● 割込みによるストップモードの解除

( 割込み制御レジスタ (ICR) の IL2, IL1, IL0 が "111<sub>B</sub>" 以外の値に設定されているとき )  
ストップモード時にリソースからレベル7より高い割込み要求が発生されると、低消費電力制御回路はストップモードを解除します。ストップモードが解除されると、CPU はこの割込みをほかの割込みと同様な方法で処理します。ただし、CPU の起動は、クロック選択レジスタ (CKSCR) の WS1 ビットと WS0 ビットで指定されたメインクロック発振安定化待ち時間が経過した後で行われます。CPU は、コンディションコードレジスタ (CCR)、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) の設定に従って処理を実行します。この割込みが受け付けられた場合は、CPU は割込み処理を実行します。この割込みが受け付けられなかった場合は、CPU はストップモードへの切換えが指定されている命令のすぐ次の命令を実行します。

割込み処理が正常に実行されると、CPU はまず、ストップモードへの切換えが指定されている命令のすぐ次の命令を実行します。CPU は次に、割込み処理に移ります。

ストップモードからノーマルモードへの復帰動作を図 6.5-4 に示します。

図 6.5-4 外部割込み出力によるストップモードの解除

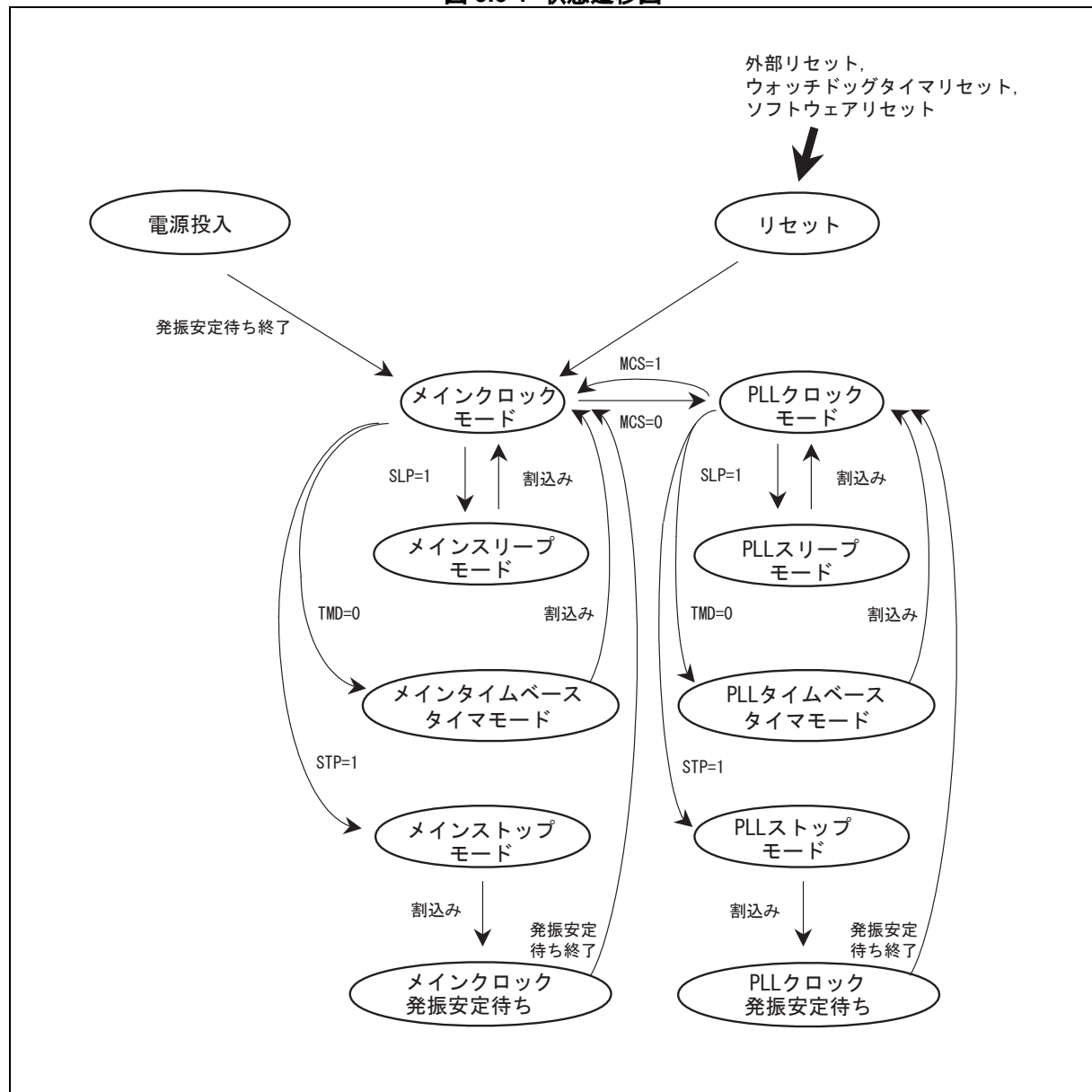


## 6.6 状態遷移図

MB90460/465 シリーズ CPU 動作モードの状態遷移図を示します。

### ■ 状態遷移図

図 6.6-1 状態遷移図



< 注意事項 > クロックモードを切り換えた場合、切り換えが完了するまではほかのクロックモードおよび低消費電力モードへ切り換えないようにしてください。切り換えの完了はクロック選択レジスタ (CKSCR) の MCM ビットを参照して確認してください。切り換えが完了する前に、ほかのクロックモードおよび低消費電力モードへの切り換えを行った場合、切り換わらない場合があります。

## 6.7 スタンバイモード，リセットの端子状態

スタンバイモード，リセットの端子状態を示します。

### ■ ソフトウェアブルアップ抵抗

ソフトウェアでプルアップ抵抗を接続している入出力端子は，出力設定にすることでプルアップ抵抗は切離されます。

### ■ 端子状態

各端子の状態を表 6.7-1 に示します。

表 6.7-1 各端子状態

端子名	スタンバイモード			リセット時
	スリープ	ストップモード		
		SPL=0	SPL=1	
P00 ~ P07 P17 P20 ~ P27 P30 ~ P37 P40 P42 ~ P47 P50 ~ P57 P60 ~ P63	直前の状態を保持 <sup>*2</sup>	直前の状態を保持 <sup>*2</sup>	入力遮断 <sup>*3</sup> / 出力 Hi-Z <sup>*4</sup>	出力 Hi-Z <sup>*4</sup>
P10 ~ P16 P63		入力可 <sup>*1</sup>		

\*1: "入力可" とは，入力機能が可能な状態であることを意味します。ただし，部割込みを許可している場合に限りです。出力ポートとして使用している場合は，低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）の設定に従います。

\*2: "直前の状態を保持" とは，スタンバイモードに遷移する直前に出力していた状態をそのまま保持<sup>\*5</sup>することを意味します。ただし，入力状態であった場合は，入力不可<sup>\*6</sup>となるので注意してください。

\*3: 入力遮断状態では，入力はマスクされ "L" レベルが内部に伝わります。

\*4: "出力 Hi-Z" とは，端子駆動用トランジスタを駆動禁止状態にし，端子をハイインピーダンスにすることを意味します。

\*5: "出力していた状態をそのまま保持" とは，リソースの出力値またはポートの出力値を保持することを意味します。

\*6: "入力不可" とは，内部回路が動作していないため，端子への入力値が内部で受け付けられない状態を意味します。

## 6.8 低消費電力モード使用上の注意

低消費電力モードを使用する場合は、以下の注意が必要です。

- スタンバイモードへの遷移と割込み
- 割込みによるスタンバイモードの解除
- スタンバイモードの設定
- 外部割込みによるストップモードの解除
- タイムベースタイマモードの解除
- 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遷移と割込み

リソースが割込み要求を出力している場合は、低消費電力モード制御レジスタ (LPMCR) のストップモードビット (STP:bit7) に "1", スリープモードビット (SLP:bit5) に "1", あるいはタイムベースタイマモードビット (TMD:bit3) に "0" を設定しても、スタンバイモードへは遷移しません。

ただし、CPU による割込み処理が実行中、割込み要求フラグがクリアされ、かつほかの割込み要求が送出されていない場合は、スタンバイモードへ遷移できます。

### ■ 割込みによるスタンバイモードの解除

スリープモード、タイムベースタイマモードおよびストップモード中に、動作しているリソースおよび外部割込みから割込みレベルが "7" より強い割込み要求 (割込み制御レジスタ ICR:IL2, IL1, IL0=000<sub>B</sub> ~ 110<sub>B</sub>) が出力されると、スタンバイモードは解除されます。

待機モードが解除された後、正常割込み処理が実行されます。CPU は割込み処理ルーチンへ分岐します (割込みレベル設定ビット (ICR の IL2, IL1, IL0) で示された割込み要求優先順位が割込みレベルマスケジスタ (ILM) より高い場合)。コンディショニングコードレジスタ (CCR) の割込み許可フラグ (I) は "1" (許可) に設定されます。割込みが受け付けられない場合は、CPU はスタンバイモードへの遷移が指定されている命令のすぐ次の命令を実行します。

割込み処理が正常に実行されると、CPU はまず、スタンバイモードへの遷移が指定されている命令のすぐ次の命令を実行します。CPU は次に、割込み処理に移ります。ただし、スタンバイモードへの遷移が実行されたときの状態によっては、CPU はその次の命令を実行する前に割込み処理に移ることがあります。

スタンバイモードからノーマルモードへ復帰した直後に CPU が割込み処理ルーチンへ分岐しない場合は、スタンバイモードが設定される前に割込みを停止する処置をしなければなりません。

### ■ スタンバイモードの設定

LPMCR の STP ビットと SLP ビットに同時に "1" を書き込むと、スタンバイモードへの遷移が実行されます。クロック選択レジスタ (CKSCR) の MCS ビットが "0" の場合は、タイムベースタイマモードへの切換えが実行されます。このビットが "1" の場合は、ストップモードへの切換えが実行されます。

## ■ 外部割込みによるストップモードの解除

ストップモードを解除するために外部割込みを使用する場合は、システムがストップモードに遷移する前に割込み入力要因として設定した入力を使用してください。入力要因として "H" レベル, "L" レベル, 立上りエッジまたは立下りエッジを選択できます。

## ■ タイムベースタイマモードの解除

PLL タイムベースタイマモードが解除されると、PLL クロック発振安定化待ち状態になります。メインタイムベースタイマモードの場合は、リセット直後かまたは割込みからの復帰と同時に実行される命令でクロック選択レジスタ (CKSCR) の MCS ビットを "1" に書き換えてください。

外部割込みを使用してタイムベースタイマモードを解除する場合は、入力要因として "H" レベル, "L" レベル, 立上りエッジまたは立下りエッジを選択できます。

## ■ 発振安定待ち時間

### ● 発振クロックの発振安定化待ち時間

ストップモード時は原発振の発振子は停止するので、発振安定化待ち時間を設けなければなりません。CKSCR の WS1 ビットと WS0 ビットで選択された時間幅は、発振安定化待ち時間として使用されます。

### ● PLL クロック発振安定化待ち時間

CPU はメインクロックで動作でき、PLL クロックは停止できます。CPU およびリソースが PLL クロックで動作するモードになると、PLL クロックは最初に発振安定化待ち状態になります。発振安定化待ち状態においては、CPU はメインクロックで動作します。

PLL クロック発振安定化待ち時間は、 $2^{14}/\text{HCLK}$  固定です (HCLK: 発振クロック周波数)。

## ■ クロックモードの切換え

クロックモードを切り換えた場合、切換えが完了するまでは低消費電力モードへの切換えおよびほかのクロックモードへ切り換えないようにしてください。切換えの完了はクロック選択レジスタ (CKSCR) の MCM ビットを参照して確認してください。

切換えが完了する前に、ほかのクロックモードおよび低消費電力モードへの切換えを行った場合、切り換わらない場合があります。

# 第7章

---

## 割込み

**MB90460/465 シリーズの割込みと拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) について説明します。**

- 7.1 割込み
- 7.2 割込み要因と割込みベクタ
- 7.3 割込み制御レジスタとリソース
- 7.4 ハードウェア割込み
- 7.5 ソフトウェア割込み
- 7.6 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) からの割込み
- 7.7 例外処理割込み
- 7.8 割込み処理時のスタック動作
- 7.9 割込み処理のプログラム例



## 7.1 割込み

---

MB90460/465 シリーズの割込みと拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) について説明します。

- ハードウェア割込み
  - ソフトウェア割込み
  - 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) からの割込み
  - 例外処理
- 

### ■ 割込みの種類と機能

#### ● ハードウェア割込み

ハードウェア割込みは、リソースから割込み要求が発生すると、ユーザ定義の割込み処理プログラムへ制御を移します。

#### ● ソフトウェア割込み

ソフトウェア割込みは、専用ソフトウェア割込み命令 (INT 命令など) の実行により起動されたユーザ定義の割込み処理プログラムへ制御を移します。

#### ● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) による割込み

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) は、リソースとメモリ間における自動データ転送機能です。EI<sup>2</sup>OS の起動時の設定プログラムと終了プログラムを作成するだけで、データを転送できます。データ転送処理を終了すると、自動的に割込み処理プログラムを実行します。

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) による割込みは、上記のハードウェア割込みの一種です。

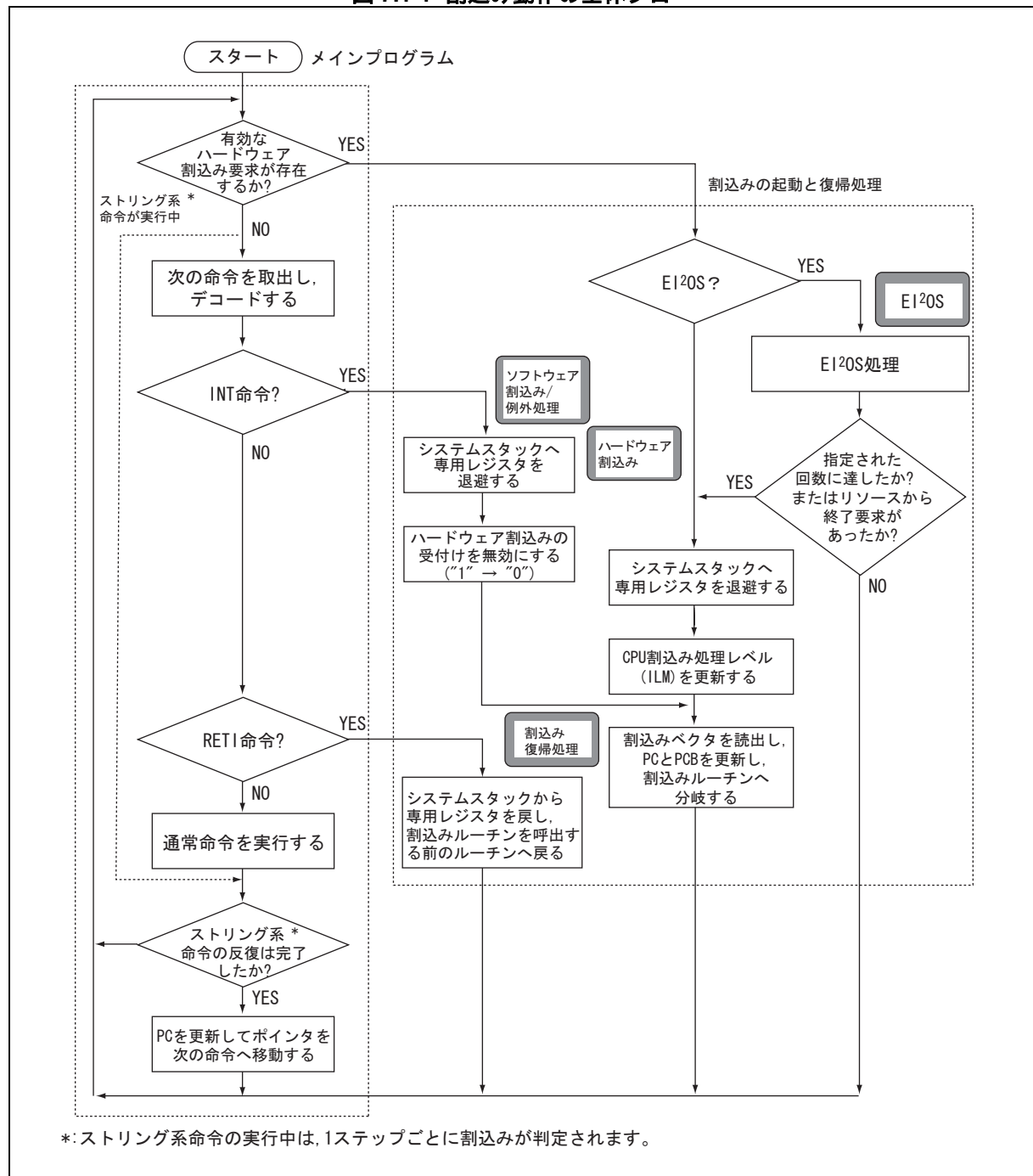
#### ● 例外処理

例外処理は、命令間で例外事項 (未定義命令の実行) の発生が検出された場合に、通常処理を中断して実行されます。上記のソフトウェア割込み命令の "INTI0" と等価です。

## ■ 割り込み動作

4種類の割り込み機能の起動と復帰処理を図7.1-1に示します。

図7.1-1 割り込み動作の全体フロー



## 7.2 割込み要因と割込みベクタ

F<sup>2</sup>MC-16LX ファミリは、256 種類の割込み要因を処理する機能を備えています。256 の割込みベクタテーブルは、最上位アドレスのメモリに割当てられています。これらの割込みベクタは、すべての割込みで共用されます。

ソフトウェア割込みは、これらすべての割込みベクタ（INT0 ～ INT255）を使用できます。ソフトウェア割込みは、同じ割込みベクタをハードウェア割込みおよび例外処理割込みと共用します。ハードウェア割込みは、各リソースの固定割込みベクタと割込み制御レジスタ（ICR）を使用します。

### ■ 割込みベクタ

割込み処理時に参照される割込みベクタテーブルは、メモリ領域の最上位アドレス（FFFC00<sub>H</sub> ～ FFFFFF<sub>H</sub>）に割当てられています。割込みベクタは、同じ領域を EI<sup>2</sup>OS、例外処理、ハードウェア割込み、ソフトウェア割込みが共用します。

割込み番号および割込みベクタの割当てを、表 7.2-1 に示します。

表 7.2-1 割込みベクタ

ソフトウェア 割込み命令	ベクタ アドレス L	ベクタ アドレス M	ベクタ アドレス H	モード データ	割込み 番号	ハードウェア 割込み
INT0	FFFFFC <sub>H</sub>	FFFFFD <sub>H</sub>	FFFFFE <sub>H</sub>	未使用	#0	なし
:	:	:	:	:	:	:
INT7	FFFFE0 <sub>H</sub>	FFFFE1 <sub>H</sub>	FFFFE2 <sub>H</sub>	未使用	#7	なし
INT8	FFFFDC <sub>H</sub>	FFFFDD <sub>H</sub>	FFFFDE <sub>H</sub>	FFFFDE <sub>H</sub>	#8	（リセットベクタ）
INT9	FFFFD8 <sub>H</sub>	FFFFD9 <sub>H</sub>	FFFFDA <sub>H</sub>	未使用	#9	なし
INT10	FFFFD4 <sub>H</sub>	FFFFD5 <sub>H</sub>	FFFFD6 <sub>H</sub>	未使用	#10	< 例外処理 >
INT11	FFFFD0 <sub>H</sub>	FFFFD1 <sub>H</sub>	FFFFD2 <sub>H</sub>	未使用	#11	ハードウェア割込み番号 0
INT12	FFFFCC <sub>H</sub>	FFFFCD <sub>H</sub>	FFFFCE <sub>H</sub>	未使用	#12	ハードウェア割込み番号 1
INT13	FFFFC8 <sub>H</sub>	FFFFC9 <sub>H</sub>	FFFFCA <sub>H</sub>	未使用	#13	ハードウェア割込み番号 2
INT14	FFFFC4 <sub>H</sub>	FFFFC5 <sub>H</sub>	FFFFC6 <sub>H</sub>	未使用	#14	ハードウェア割込み番号 3
:	:	:	:	:	:	:
INT254	FFFC04 <sub>H</sub>	FFFC05 <sub>H</sub>	FFFC06 <sub>H</sub>	未使用	#254	なし
INT255	FFFC00 <sub>H</sub>	FFFC01 <sub>H</sub>	FFFC02 <sub>H</sub>	未使用	#255	なし

L: 下位アドレス  
M: 中位アドレス  
H: 上位アドレス

< 参考 >      使用されない割込みベクタは、例外処理などのアドレスに設定することを推奨します。

## ■ 割り込み要因，割り込みベクタおよび割り込み制御レジスタ

割り込み要因（ソフトウェア割り込みを除く），割り込みベクタおよび割り込み制御レジスタの関係を表 7.2-2 に示します。

表 7.2-2 割り込み要因，割り込みベクタおよび割り込み制御レジスタ（1 / 2）

割り込み要因	EI <sup>2</sup> OS 対応	割り込みベクタ			割り込み制御レジスタ		優先度 <sup>*2</sup>
		番号		アドレス	ICR	アドレス	
リセット	×	#08	08 <sub>H</sub>	FFFFDC <sub>H</sub>	-	-	高い
INT9 命令	×	#09	09 <sub>H</sub>	FFFFD8 <sub>H</sub>	-	-	
例外処理	×	#10	0A <sub>H</sub>	FFFFD4 <sub>H</sub>	-	-	
A/D コンバータによる変換終了		#11	0B <sub>H</sub>	FFFFD0 <sub>H</sub>	ICR00	0000B0 <sub>H</sub> <sup>*1</sup>	
アウトプットコンペアチャンネル 0 一致		#12	0C <sub>H</sub>	FFFFCC <sub>H</sub>			
PWC0 タイマ測定終了， PWC0 タイマオーバフロー		#13	0D <sub>H</sub>	FFFFC8 <sub>H</sub>	ICR01	0000B1 <sub>H</sub> <sup>*1</sup>	
16 ビット PPG タイマ 0		#14	0E <sub>H</sub>	FFFFC4 <sub>H</sub>			
アウトプットコンペアチャンネル 1 一致		#15	0F <sub>H</sub>	FFFFC0 <sub>H</sub>	ICR02	0000B2 <sub>H</sub> <sup>*1</sup>	
16 ビット PPG タイマ 1		#16	10 <sub>H</sub>	FFFFBC <sub>H</sub>			
アウトプットコンペアチャンネル 2 一致		#17	11 <sub>H</sub>	FFFFB8 <sub>H</sub>	ICR03	0000B3 <sub>H</sub> <sup>*1</sup>	
16 ビットリロードタイマ 1 アンダ フロー		#18	12 <sub>H</sub>	FFFFB4 <sub>H</sub>			
アウトプットコンペアチャンネル 3 一致		#19	13 <sub>H</sub>	FFFFB0 <sub>H</sub>	ICR04	0000B4 <sub>H</sub> <sup>*1</sup>	
DTP/ 外部割り込みチャンネル 0/1 検出		#20	14 <sub>H</sub>	FFFFAC <sub>H</sub>			
DTTI0							
アウトプットコンペアチャンネル 4 一致		#21	15 <sub>H</sub>	FFFFA8 <sub>H</sub>	ICR05	0000B5 <sub>H</sub> <sup>*1</sup>	
DTP/ 外部割り込みチャンネル 2/3 検出		#22	16 <sub>H</sub>	FFFFA4 <sub>H</sub>			
DTTI1							
アウトプットコンペアチャンネル 5 一致		#23	17 <sub>H</sub>	FFFFA0 <sub>H</sub>	ICR06	0000B6 <sub>H</sub> <sup>*1</sup>	
PWC1 タイマ測定終了， PWC1 タイマオーバフロー		#24	18 <sub>H</sub>	FFFF9C <sub>H</sub>			
DTP/ 外部割り込みチャンネル 4/5 検出		#25	19 <sub>H</sub>	FFFF98 <sub>H</sub>	ICR07	0000B7 <sub>H</sub> <sup>*1</sup>	
マルチパルスジェネレータタイマ コンペア一致 / 書込みタイミング		#26	1A <sub>H</sub>	FFFF94 <sub>H</sub>			
							低い

表 7.2-2 割り込み要因，割り込みベクタおよび割り込み制御レジスタ ( 2 / 2 )

割り込み要因	EI <sup>2</sup> OS 対応	割り込みベクタ		割り込み制御レジスタ		優先度 <sup>*2</sup>	
		番号	アドレス	ICR	アドレス		
DTP/ 外部割り込みチャネル 6/7 検出		#27	1B <sub>H</sub>	FFFF90 <sub>H</sub>	ICR08	0000B8 <sub>H</sub> <sup>*1</sup>	高い
マルチパルスジェネレータ位置検出 / コンペア割り込み		#28	1C <sub>H</sub>	FFFF8C <sub>H</sub>			
波形生成部 16 ビットタイマ 0/1/2 アンダフロー		#29	1D <sub>H</sub>	FFFF88 <sub>H</sub>	ICR09	0000B9 <sub>H</sub> <sup>*1</sup>	
16 ビットリロードタイマ 0 アンダフロー		#30	1E <sub>H</sub>	FFFF84 <sub>H</sub>			
16 ビットフリーランタイマ 0 検出		#31	1F <sub>H</sub>	FFFF80 <sub>H</sub>	ICR10	0000BA <sub>H</sub> <sup>*1</sup>	
16 ビット PPG タイマ 2		#32	20 <sub>H</sub>	FFFF7C <sub>H</sub>			
インプットキャプチャチャネル 0/1		#33	21 <sub>H</sub>	FFFF78 <sub>H</sub>	ICR11	0000BB <sub>H</sub> <sup>*1</sup>	
16 ビットフリーランタイマコンペアクリア		#34	22 <sub>H</sub>	FFFF74 <sub>H</sub>			
アウトプットキャプチャチャネル 2/3		#35	23 <sub>H</sub>	FFFF70 <sub>H</sub>	ICR12	0000BC <sub>H</sub> <sup>*1</sup>	
タイムベースタイマ		#36	24 <sub>H</sub>	FFFF6C <sub>H</sub>			
UART1 受信完了		#37	25 <sub>H</sub>	FFFF68 <sub>H</sub>	ICR13	0000BD <sub>H</sub> <sup>*1</sup>	
UART1 送信開始		#38	26 <sub>H</sub>	FFFF64 <sub>H</sub>			
UART0 受信完了		#39	27 <sub>H</sub>	FFFF60 <sub>H</sub>	ICR14	0000BE <sub>H</sub> <sup>*1</sup>	
UART0 送信開始		#40	28 <sub>H</sub>	FFFF5C <sub>H</sub>			
フラッシュメモリステータス		#41	29 <sub>H</sub>	FFFF58 <sub>H</sub>	ICR15	0000BF <sub>H</sub> <sup>*1</sup>	
遅延割り込み発生モジュール		#42	2A <sub>H</sub>	FFFF54 <sub>H</sub>			低い

:EI<sup>2</sup>OS は使用可能であり，割り込み要求フラグは EI<sup>2</sup>OS 割り込みクリア信号でクリアされます。

× :EI<sup>2</sup>OS は使用不可。

:EI<sup>2</sup>OS は使用可能であり，EI<sup>2</sup>OS 停止要求もサポートされています。

:EI<sup>2</sup>OS は，ICR を共用する割り込み要因を使用しない場合，使用可能です。

\*1:

- ICR レジスタを共用するリソースの場合，割り込みレベルは同じです。

- EI<sup>2</sup>OS を，別のリソースと，ICR レジスタを共用しているリソースと一緒に使用する場合は，EI<sup>2</sup>OS はこれら 2 つのリソースのどちらかで起動できます。また，EI<sup>2</sup>OS クリアをサポートしている場合は，2 つの割り込み要因の各割り込み要求フラグが EI<sup>2</sup>OS 割り込みクリア信号でクリアされます。EI<sup>2</sup>OS の使用中は，どちらか一方の割り込み要求をマスクすることを推奨します。

- 複数の EI<sup>2</sup>OS を複数回，同時に起動することはできません。EI<sup>2</sup>OS の動作中は，動作中の割り込み以外の割り込みはマスクされます。EI<sup>2</sup>OS の使用中は，どちらか一方の割り込み要求をマスクすることを推奨します。

\*2: この優先度は，同じレベルの割り込みが同時に発生する場合に適用されます。

## 7.3 割り込み制御レジスタとリソース

割り込み制御レジスタ (ICR00 ~ ICR15) は、割り込みコントローラ内に存在します。割り込み制御レジスタは、割り込み機能を持つすべてのリソースに対応しています。これらのレジスタは、割り込みと拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を制御します。

### ■ 割り込み制御レジスタ

割り込み制御レジスタと、これらのレジスタに対応しているリソースを表 7.3-1 に示します。

表 7.3-1 割り込み制御レジスタ

アドレス	レジスタ	略語	対応するリソース
0000B0 <sub>H</sub>	割り込み制御レジスタ 00	ICR00	A/D コンバータ, アウトプットコンペア 0
0000B1 <sub>H</sub>	割り込み制御レジスタ 01	ICR01	PWC0, 16 ビット PPG タイマ 0
0000B2 <sub>H</sub>	割り込み制御レジスタ 02	ICR02	アウトプットコンペア 1, 16 ビット PPG タイマ 1
0000B3 <sub>H</sub>	割り込み制御レジスタ 03	ICR03	アウトプットコンペア 2, 16 ビットリロードタイマ 1
0000B4 <sub>H</sub>	割り込み制御レジスタ 04	ICR04	アウトプットコンペア 3, DTP/ 外部割り込み 0/1, DTTI0
0000B5 <sub>H</sub>	割り込み制御レジスタ 05	ICR05	アウトプットコンペア 4, DTP/ 外部割り込み 2/3, DTTI1
0000B6 <sub>H</sub>	割り込み制御レジスタ 06	ICR06	アウトプットコンペア 5, PWC1 タイマ
0000B7 <sub>H</sub>	割り込み制御レジスタ 07	ICR07	DTP/ 外部割り込み 4/5, マルチパルスジェネレータ
0000B8 <sub>H</sub>	割り込み制御レジスタ 08	ICR08	DTP/ 外部割り込み 6/7, マルチパルスジェネレータ
0000B9 <sub>H</sub>	割り込み制御レジスタ 09	ICR09	波形生成部, 16 ビットリロードタイマ 0
0000BA <sub>H</sub>	割り込み制御レジスタ 10	ICR10	16 ビットフリーランタイマ 0 検出, 16 ビット PPG タイマ 2
0000BB <sub>H</sub>	割り込み制御レジスタ 11	ICR11	インプットキャプチャ 0/1, 16 ビットフリーランタイマコンペアクリア
0000BC <sub>H</sub>	割り込み制御レジスタ 12	ICR12	インプットキャプチャ 2/3, タイムベースタイマ
0000BD <sub>H</sub>	割り込み制御レジスタ 13	ICR13	UART1
0000BE <sub>H</sub>	割り込み制御レジスタ 14	ICR14	UART0
0000BF <sub>H</sub>	割り込み制御レジスタ 15	ICR15	フラッシュメモリ, 遅延割り込みジェネレータモジュール

### ■ 割り込み制御レジスタの機能

すべての割り込み制御レジスタ (ICR) は、以下を処理します。

- 対応するリソースの割り込みレベルを設定する
- 対応するリソースの割り込みとして, " 通常割り込み " または " 拡張インテリジェント I/O サービス " を選択する
- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のチャンネルを選択する
- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の状態を表示する

割り込み制御レジスタ (ICR) の一部の機能は、図 7.3-1 と図 7.3-2 に示されているように、書込み時と読出し時では異なります。

割り込み制御レジスタ (ICR) へアクセスする際、リードモディファイライト命令は使用しないでください。リードモディファイライト命令を使用すると、正常に動作しません。

### 7.3.1 割り込み制御レジスタ (ICR00 ~ ICR15)

割り込み制御レジスタは、割り込み機能を持つすべてのリソースに対応しています。割り込み制御レジスタは、割り込み要求が発生すると割り込み処理を制御します。割り込み制御レジスタの一部の機能は、書き込み時と読み出し時では異なります。

#### ■ 割り込み制御レジスタ (ICR00 ~ ICR15)

図 7.3-1 書き込み時の割り込み制御レジスタ (ICR00 ~ ICR15)

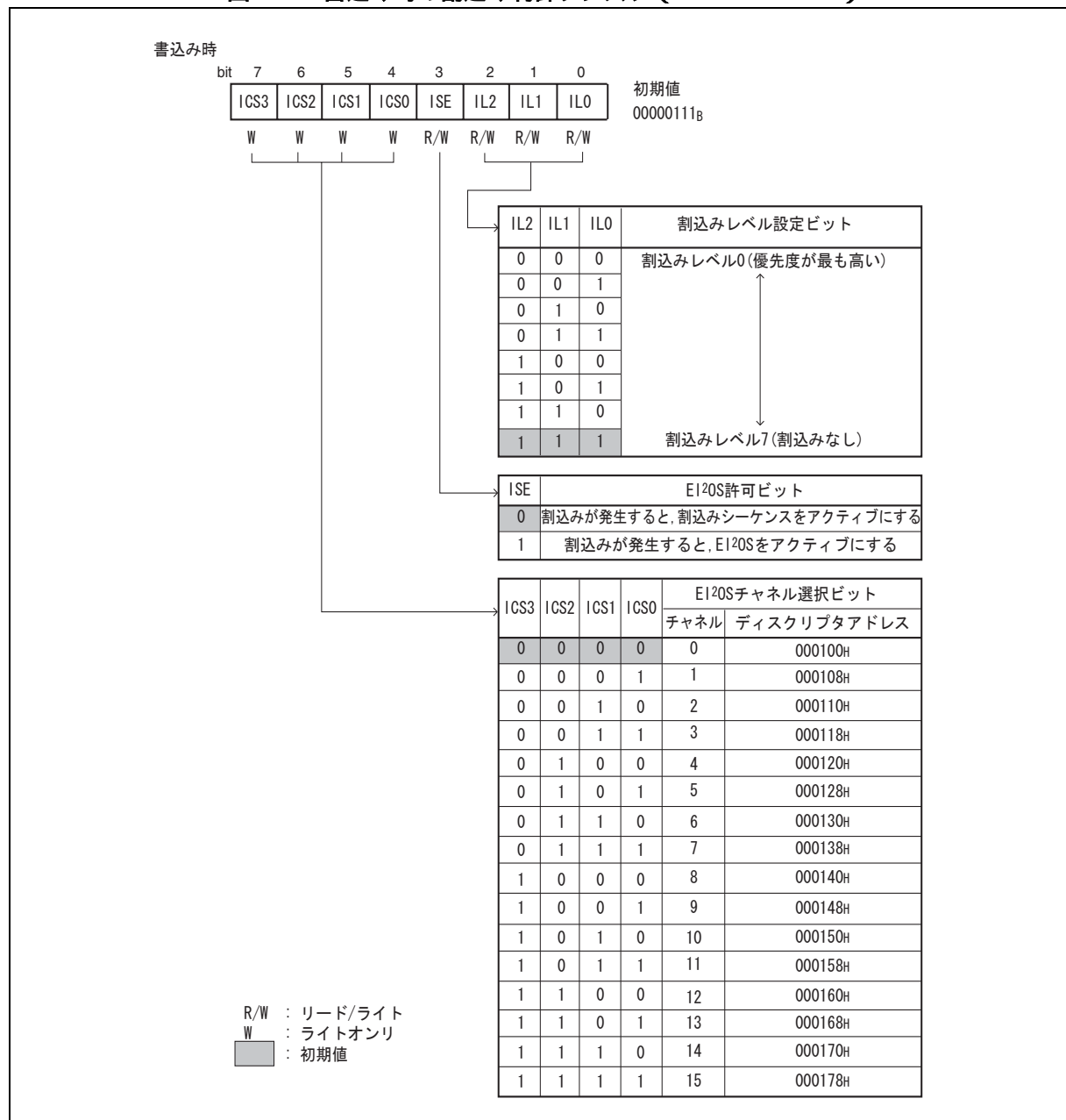
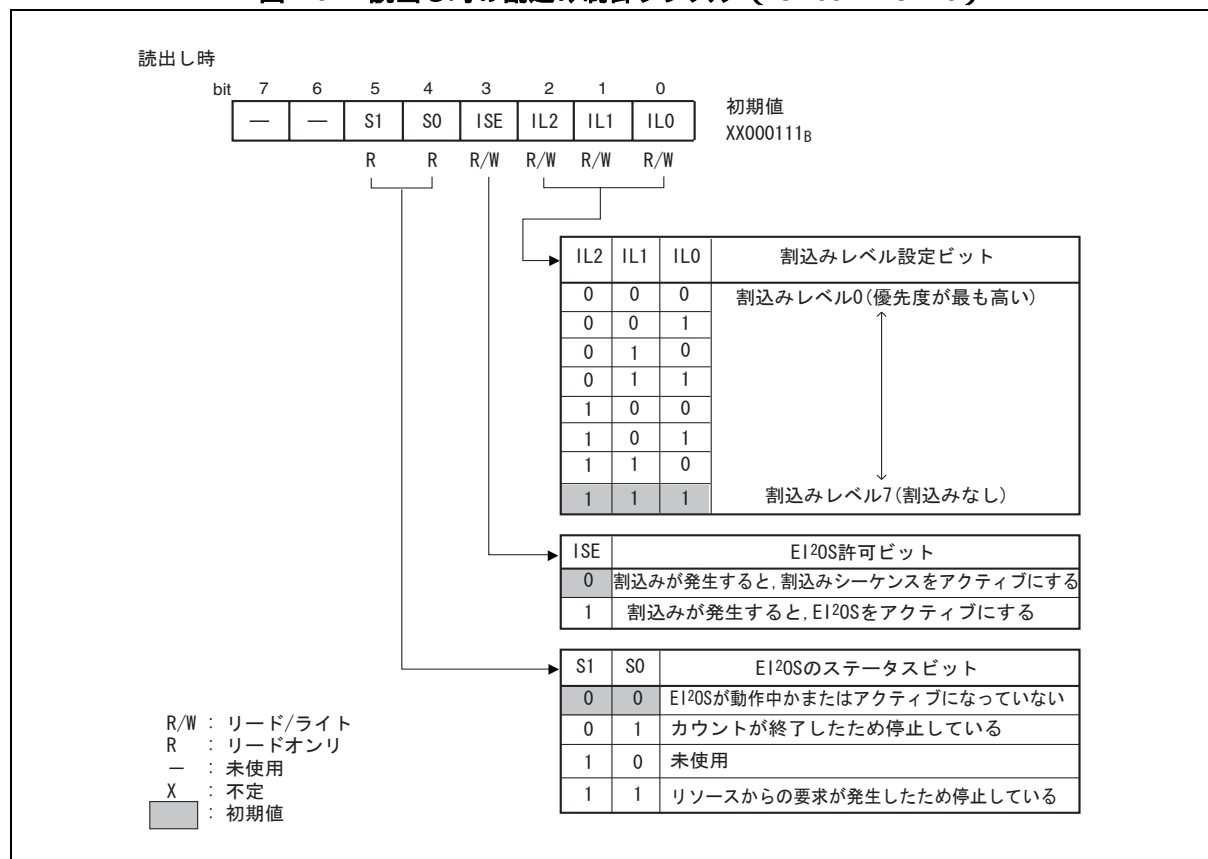


図 7.3-2 読出し時の割り込み制御レジスタ (ICR00 ~ ICR15)





## 7.3.2 割り込み制御レジスタの機能

割り込み制御レジスタ (ICR00 ~ ICR15) は、以下の4つの機能ビットから構成されています。

- 割り込みレベル設定ビット (IL2 ~ IL0)
- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) 許可ビット (ISE:bit3)
- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) チャンネル選択ビット (ICS3 ~ ICS0:bit7 ~ bit4)
- 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) ステータスビット (S1, S0:bit5, bit4)

### ■ 割り込み制御レジスタ (ICR)

割り込み制御レジスタ (ICR) のビット構成を図 7.3-3 に示します。

図 7.3-3 割り込み制御レジスタ (ICR) の構成



#### < 参考 >

- ICS3:bit7 から ICS0:bit4 までは、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) がアクティブになっている場合に限り有効です。EI<sup>2</sup>OS をアクティブにするには、ISE:bit3 に "1" を設定してください。EI<sup>2</sup>OS をアクティブにしないためには、ISE:bit3 に "0" を設定してください。EI<sup>2</sup>OS がアクティブでない場合は、ICS3 から ICS0 までのビット設定は不要です。
- ICS1, 0:bit5, bit4 は書込みのみ可能です。S1, 0:bit5, bit4 は、読出しのみ可能です。

### ■ 割り込み制御レジスタの機能

- 割り込みレベル設定ビット (IL2 ~ IL0: bit2 ~ bit0)

これらのビットは、対応するリソースの割り込みレベルを設定します。これらのビットは、リセットすることにより、レベル7 (割り込みなし) に初期化されます。

割り込みレベル設定ビットと割り込みレベルの対応関係を表 7.3-2 に示します。

表 7.3-2 割り込みレベル設定ビットと割り込みレベルの対応関係

IL2	IL1	IL0	割り込みレベル
0	0	0	0 (優先度が最も高い)
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	6 (優先度が最も低い)
1	1	1	7 (割り込みなし)

● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) 許可ビット (ISE:bit3)

割り込み要求生成時にこのビットが "1" の場合、EI<sup>2</sup>OS がアクティブになります。割り込み要求発生時にこのビットが "0" の場合、割り込みシーケンスがアクティブになります。EI<sup>2</sup>OS 終了条件が満たされる (S1, S0:bit5, bit4 が "00<sub>B</sub>" でない場合) と、ISE:bit3 がクリアされます。対応するリソースが EI<sup>2</sup>OS 機能を持っていない場合は、ソフトウェアにより "0" を設定しなければなりません。ISE:bit3 は、リセットにより "0" に初期化されます。

● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) チャンネル選択ビット (ICS3 ~ ICS0:bit7 ~ bit4)

これらのビットは書き込み専用ビットであり、EI<sup>2</sup>OS チャンネルを指定します。EI<sup>2</sup>OS ディスクリプタアドレスは、これらのビット値に基づいて決定されます。これらのビットは、リセットにより "0000<sub>B</sub>" に初期化されます。

EI<sup>2</sup>OS チャンネル選択ビットとディスクリプタアドレスの対応関係を表 7.3-3 に示します。

表 7.3-3 EI<sup>2</sup>OS チャンネル選択ビットとディスクリプタアドレスの対応関係

ICS3	ICS2	ICS1	ICS0	選択されるチャンネル	ディスクリプタアドレス
0	0	0	0	0	000100 <sub>H</sub>
0	0	0	1	1	000108 <sub>H</sub>
0	0	1	0	2	000110 <sub>H</sub>
0	0	1	1	3	000118 <sub>H</sub>
0	1	0	0	4	000120 <sub>H</sub>
0	1	0	1	5	000128 <sub>H</sub>
0	1	1	0	6	000130 <sub>H</sub>
0	1	1	1	7	000138 <sub>H</sub>
1	0	0	0	8	000140 <sub>H</sub>
1	0	0	1	9	000148 <sub>H</sub>
1	0	1	0	10	000150 <sub>H</sub>
1	0	1	1	11	000158 <sub>H</sub>
1	1	0	0	12	000160 <sub>H</sub>
1	1	0	1	13	000168 <sub>H</sub>
1	1	1	0	14	000170 <sub>H</sub>
1	1	1	1	15	000178 <sub>H</sub>

● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) ステータスビット (S1, S0:bit5, bit4)

これらのビットは、読出し専用ビットです。このビット値を EI<sup>2</sup>OS 終了時にチェックすると、動作状態と終了状態を区別できます。これらのビットは、リセットにより "00<sub>B</sub>" に初期化されます。

S1, S0:bit5, bit4 と EI<sup>2</sup>OS 状態の対応関係を表 7.3-4 に示します。

表 7.3-4 EI<sup>2</sup>OS ステータスビットと EI<sup>2</sup>OS 状態の対応関係

S1	S0	EI <sup>2</sup> OS の状態
0	0	EI <sup>2</sup> OS が動作中かまたはアクティブになっていない
0	1	カウントが終了したため停止している
1	0	未使用
1	1	リソースからの要求が発生したため停止している

## 7.4 ハードウェア割込み

---

ハードウェア割込み機能は、リソースからの割込み信号に応じて、CPU が実行中のプログラムを一時的に中断し、ユーザ定義の割込み処理プログラムへ制御を移します。

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) と外部割込みは、一種のハードウェア割込みとして実行されます。

---

### ■ ハードウェア割込み

#### ● ハードウェア割込み機能

ハードウェア割込み機能は、リソースから出力された割込み要求信号の割込みレベルを、CPU のプロセッサステータス (PS) の割込みレベルマスクレジスタ (ILM) と比較します。次に、ハードウェア割込み機能は、ハードウェアを介して CPU のプロセッサステータス (PS) の I フラグ値を参照し、割込みが受け付け可能か否かを決定します。ハードウェア割込みが受け付けられると、CPU 内部レジスタの値は自動的にシステムスタックへ退避されます。現在要求されている割込みレベルは、割込みレベルマスクレジスタ (ILM) に格納され、ハードウェア割込み機能は対応する割込みベクタへ分岐します。

#### ● 複数の割込み

複数のハードウェア割込みを起動できます。

#### ● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS)

EI<sup>2</sup>OS は、メモリと I/O 領域の間の自動転送機能です。指定された転送数が完了すると、ハードウェア割込みが起動します。複数の EI<sup>2</sup>OS の起動は発生しません。EI<sup>2</sup>OS 処理中は、ほかのすべての割込み要求と EI<sup>2</sup>OS 要求は保留されます。

#### ● 外部割込み

外部割込みは、ハードウェア割込みとしてリソース (割込み要求検出回路) から受け付けられます。

#### ● 割込みベクタ

割込み処理中に参照される割込みベクタテーブルは、"FFFC00<sub>H</sub>" から "FFFFFF<sub>H</sub>" までのメモリに割当てられます。これらのテーブルは、ソフトウェア割込みと共用されます。

割込み番号と割込みベクタの割当てに関する詳細は、「7.2 割込み要因と割込みベクタ」をご参照ください。

## ■ ハードウェア割り込みの構造

ハードウェア割り込みを実行する際に使用される 4 つの機能を表 7.4-1 に示します。これら 4 つの機能は、ハードウェア割り込みを起動する前に、プログラムに組み込まれていなければならない。

表 7.4-1 ハードウェア割り込みで使用されるメカニズム

	機能	動作
リソース	割り込み許可ビット， 割り込み要求ビット	リソースからの割り込み要求を制御する
割り込みコントローラ	割り込み制御レジスタ (ICR)	割り込みレベルを設定し，EI <sup>2</sup> OS を制御する
CPU	割り込み許可フラグ (I)	割り込み許可状態を識別する
	割り込みレベルマスクレジスタ (ILM)	要求割り込みレベルと現割り込みレベルを比較する
	マイクロコード	割り込み処理ルーチンを実行する
メモリの "FFFC00 <sub>H</sub> " から "FFFFFF <sub>H</sub> " までのアドレス	割り込みベクタテーブル	割り込み処理の分岐先アドレスを格納する

## ■ ハードウェア割り込みの抑制

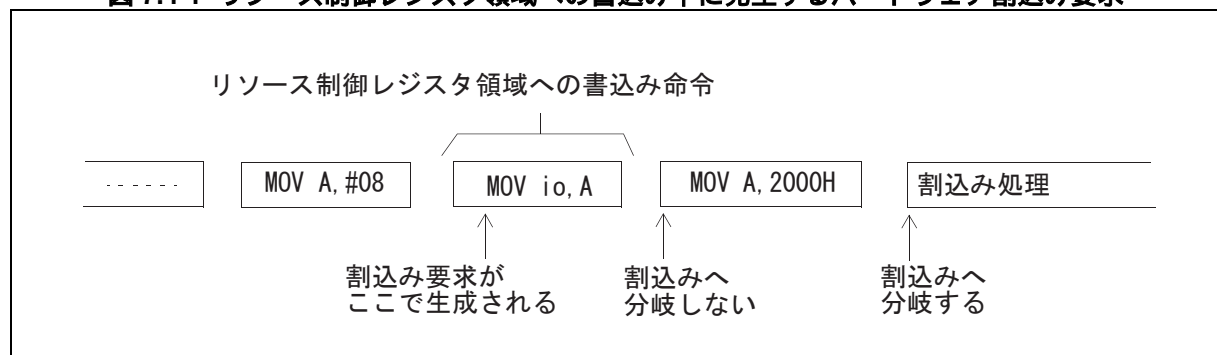
以下の場合，ハードウェア割り込み要求の受け付けは抑制されます。

### ● リソース制御レジスタへの書き込み中

リソース制御レジスタ領域へのデータ書き込み中，ハードウェア割り込み要求は受け付けられません。これにより，CPU の誤動作を防止できます。この誤動作は，リソースの割り込み制御レジスタへのデータ書き込み時に割り込み要求が生成された場合に発生する場合があります。リソース制御レジスタ領域は，"000000<sub>H</sub>" から "0000FF<sub>H</sub>" までの I/O アドレス指定領域ではなく，リソース制御レジスタとリソースデータレジスタに割当てられた領域です。

リソース領域への書き込み時におけるハードウェア割り込み動作を図 7.4-1 に示します。

図 7.4-1 リソース制御レジスタ領域への書き込み中に発生するハードウェア割り込み要求



● 割り込み抑制命令の実行中

表 7.4-2 に示されている 10 種類のハードウェア割り込み抑制命令の実行中にハードウェア割り込みが発生した場合、ハードウェア割り込み抑止命令の処理後、ほかの命令が実行された後に割り込みが処理されます。

表 7.4-2 ハードウェア割り込み抑制命令

	プリフィクスコード	割り込み抑止命令
割り込み要求を受付けない命令	PCB	MOV ILM, #imm8
	DTB	OR CCR, #imm8
	ADB	AND CCR, #imm8
	SPB	POPW PS
	CMR	
	NCC	

これらのハードウェア割り込み抑制命令のいずれかの実行中は、有効なハードウェア割り込み要求が生成されても、この割り込みは別の種類の命令が初めて実行されるまで処理されません。

● ソフトウェア割り込みの実行中

ソフトウェア割り込みがアクティブになると、I フラグは "0" にクリアされます。この場合、ほかの割り込み要求を受付けることはできません。

## 7.4.1 ハードウェア割込み動作

---

ハードウェア割込み要求発生から割込み処理完了までのハードウェア割込み動作について説明します。

---

### ■ ハードウェア割込みのアクティブ化

#### ● リソースの動作（割込み要求の発生）

ハードウェア割込み要求機能を持っているリソースは、割込み要求が存在することを示す割込み要求フラグと、CPU 割込み要求が許可か禁止かを判定する割込み許可フラグを持っています。割込み要求フラグは、リソース固有のイベントが発生した場合に設定されます。

#### ● 割込みコントローラの動作（割込み要求制御）

割込みコントローラは、同時に受取った割込み要求の割込みレベル（IL）を比較します。割込みコントローラは、最も高いレベル（IL 値は最も小さい）の要求を選択し、それを CPU へ通知します。複数の要求の割込みレベルが同じ場合は、割込み番号が小さい要求が最も高い優先度となります。

#### ● CPU 動作（割込み要求受け付けと割込み処理）

CPU は、受取った割込みレベル（ICR レジスタの IL2 ~ IL0:bit2 ~ bit0）と、割込みレベルマスクレジスタ（ILM）を比較します。"IL < ILM" であり、かつ割込みが許可（PS レジスタ中の CCR レジスタの I フラグ = 1）の場合は、現在実行中の命令が終了した後、CPU は割込み処理マイクロコードをアクティブにします。

CPU は、割込み処理マイクロコードの先頭で割込み制御レジスタ（ICR）の ISE:bit3 を参照します。ISE:bit3=0 の場合は、CPU は割込み処理の実行を継続します（ISE:bit3=1 の場合は、EI<sup>2</sup>OS はアクティブになります）。

割込み処理時は、専用レジスタ（A, DPR, ADB, DTB, PCB, PC, PS から構成される 12 バイト）の値がシステムスタック（SSB と SSP が示すシステムスタック空間）へ退避されます。

次に CPU は、割込みベクタプログラムカウンタ（PCB, PC）へデータをロードし、割込みレベルマスクレジスタ（ILM）を更新し、スタックフラグ（S）を設定します（コンディションコードレジスタ（CCR）の S ビットに "1" を設定し、システムスタックをアクティブにします）。

### ■ ハードウェア割込みからの復帰

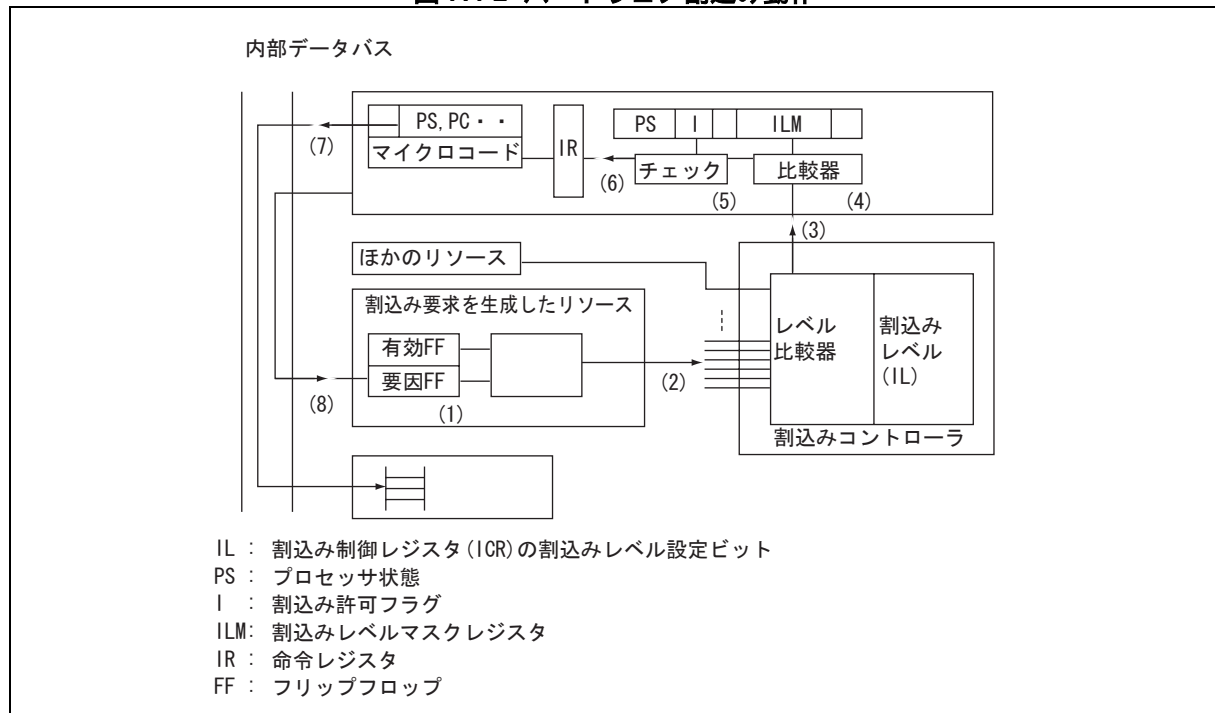
割込み処理プログラムでは、割込み要因が発生したリソースの割込み要求フラグがクリアされて RETI 命令が実行されると、システムスタックに退避されていた 12 バイトデータが専用レジスタへ復元され、割込みを分岐する前に実行していた処理が再開します。

割込み要求フラグがクリアされると、リソースが割込みコントローラへ出力した割込み要求は、自動的にキャンセルされます。

## ■ ハードウェア割り込み動作

ハードウェア割り込み発生から割り込み処理完了までのハードウェア割り込み動作を図 7.4-2 に示します。

図 7.4-2 ハードウェア割り込み動作



- (1) 割り込み要因がリソース内部で発生します。
- (2) リソースの割り込み許可ビットが参照されます。割り込みが許可の場合は、割り込み要求がリソースから割り込みコントローラへ出力されます。
- (3) 割り込み要求を受取る割り込みコントローラは、同時に受取った割り込み要求の優先度を決定し、対応する割り込み要求に一致する割り込みレベル (IL) を CPU へ転送します。
- (4) CPU は、割り込みコントローラへ要求された割り込みレベル (IL) を割り込みレベルマスクレジスタ (ILM) と比較します。
- (5) 比較の結果、当該割り込みレベルの優先度が現割り込み処理レベルよりも高いことを示している場合は、CPU はコンディションコードレジスタ (CCR) の I フラグ値をチェックします。
- (6) 上記ステップ (5) のチェックにより、I フラグが割り込みイネーブル (I=1) を示している場合、CPU は現在実行中の命令の実行が終了するまで処理を待ちます。当該命令が終了すると、CPU は要求されたレベル (IL) を割り込みレベルマスクレジスタ (ILM) に設定します。
- (7) レジスタが退避され、処理は割り込み処理ルーチンへ分岐します。
- (8) 上記ステップ (1) で発生した割り込み要因は、割り込み処理ルーチンによりソフトウェア的にクリアされます。RETI 命令を実行すると、割り込み処理は終了します。



### 7.4.2 割込み処理

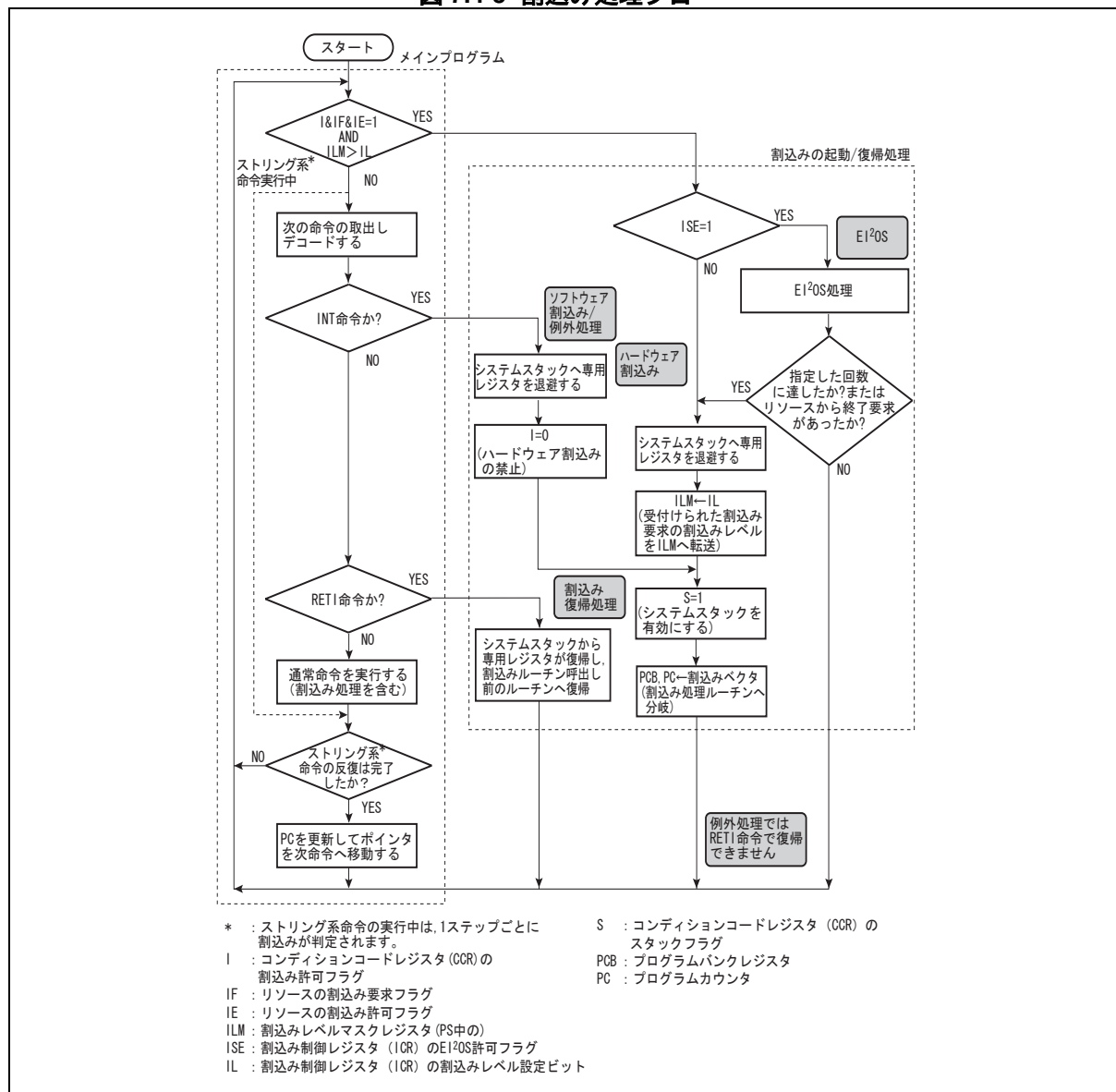
リソースにおいて割込み要求が発生すると、割込みコントローラは割込みレベルを CPU へ送信します。CPU が割込みを受付けることができる場合は、割込みコントローラは実行中の命令を一時的に中断します。次に割込みコントローラは、割込み処理ルーチンを実行するかまたは拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) をアクティブにします。

ソフトウェア割込みが INT 命令で生成された場合は、割込み処理ルーチンが CPU 状態とは無関係に実行されます。この場合、ハードウェア割込みは許可されません。

## ■ 割込み処理

割込み処理フローを図 7.4-3 に示します。

図 7.4-3 割込み処理フロー



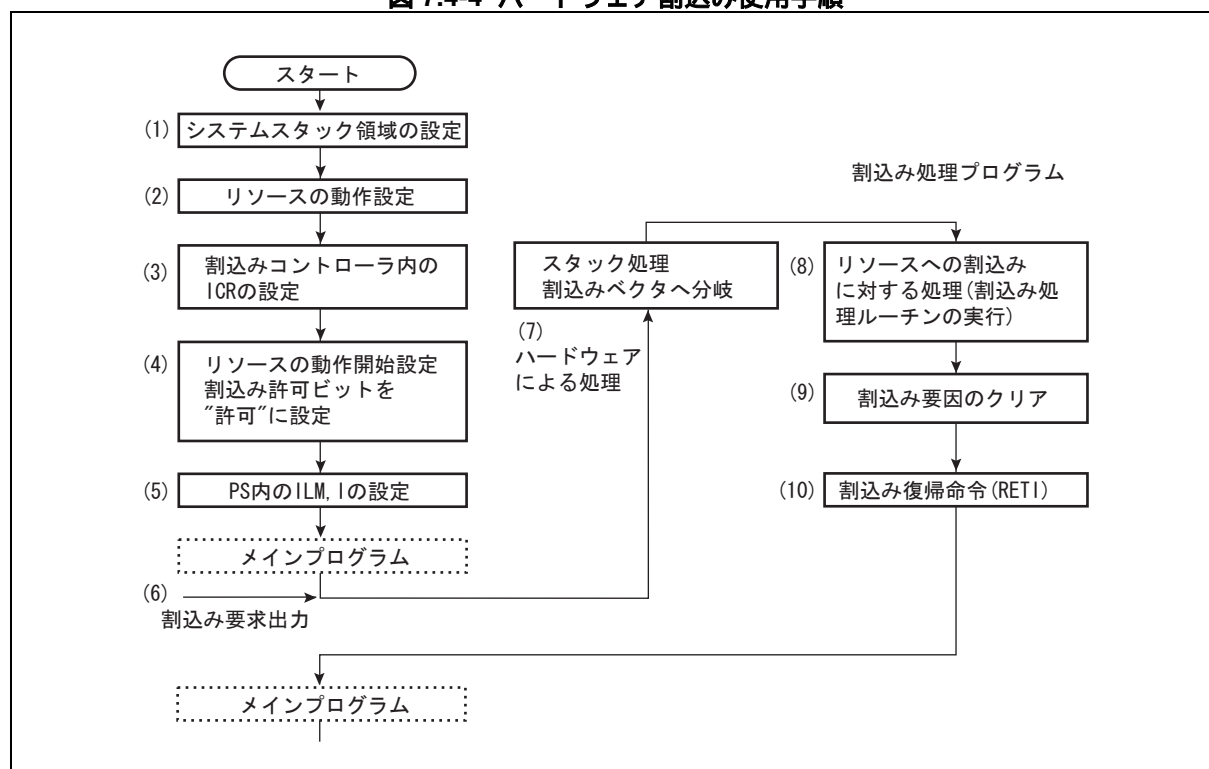
### 7.4.3 ハードウェア割り込み使用手順

ハードウェア割り込みを使用する場合は、事前にシステムスタック領域、リソースおよび割り込み制御レジスタ（ICR）を設定しておかなければなりません。

#### ■ ハードウェア割り込み使用手順

ハードウェア割り込み使用手順の例を図 7.4-4 に示します。

図 7.4-4 ハードウェア割り込み使用手順



- (1) システムスタック領域を設定してください。
- (2) 割り込み要求を発生できるリソースを初期化してください。
- (3) 割り込みコントローラの割り込み制御レジスタ（ICR）を設定してください。
- (4) リソースを動作開始状態に設定し、割り込み許可ビットを "許可" に設定してください。
- (5) 割り込みレベルマスクレジスタ（ILM）と割り込み許可フラグ（I）を "割り込み受け可能" に設定してください。
- (6) リソースにおいて発生した割り込みによりハードウェア割り込み要求が発生されます。
- (7) 割り込み処理ハードウェアがレジスタを退避し、割り込み処理プログラムへ分岐します。
- (8) 割り込みルーチンにてプログラムが実行されます。
- (9) リソースからの割り込み要求をクリアします。
- (10) 割り込み復帰命令を実行し、割り込みへ分岐する前に実行されていたプログラムへ制御を戻します。

## 7.4.4 複数の割込み

---

割込み制御レジスタ (ICR) の割込みレベル設定ビット (IL2 ~ IL0:bit2 ~ bit0) に複数の割込みレベルを設定すると、リソースから複数の割込み要求が発生した場合、複数のハードウェア割込みを起動できます。ただし、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) では複数の割込みを使用することはできません。

---

### ■ 複数の割込み

#### ● 複数の割込み動作

割込み処理ルーチンの実行中、優先度がより高い割込み要求が発生すると、現割込み処理が中断され、このより高い割込み優先度を持つ割込み要求が受け付けられます。この割込み要求が終了すると、CPU は前の割込み処理へ処理を戻します。

割込みレベルとしては "0" から "7" ままでを設定できます。レベル "7" を設定すると、CPU は割込み要求を受け付けません。

割込み処理実行中に割込み優先順位が同じかまたは低い割込み要求が生成されると、コンディションコードレジスタ (CCR) の I フラグまたは割込みレベルマスクレジスタ (ILM) が変更されない限り、現割込み処理が終了するまでこの新しい割込み要求は保留されます。

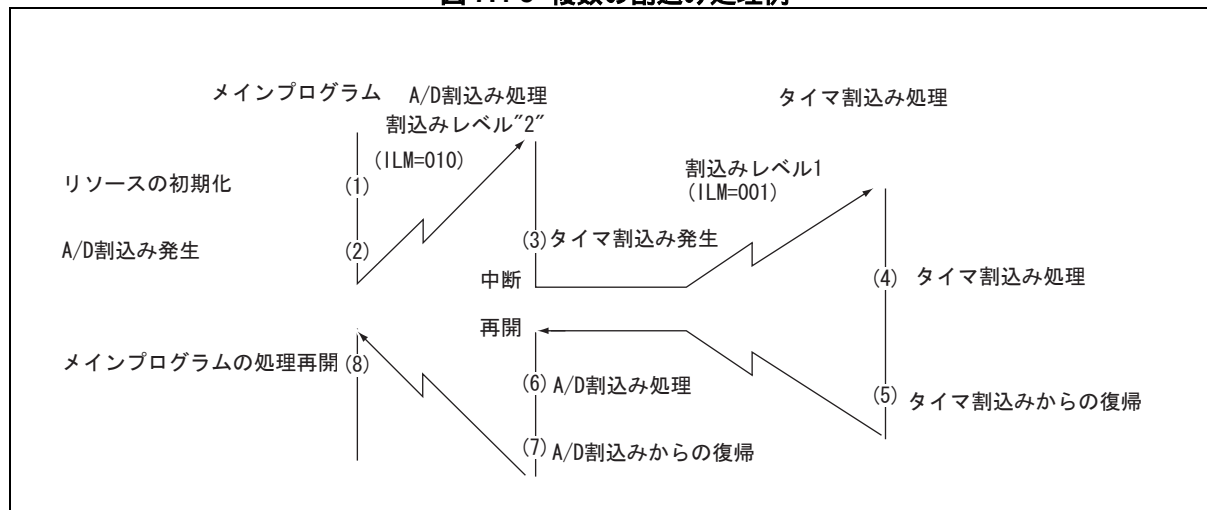
割込み中に起動予定であるほかの複数の割込みは、割込み処理ルーチンのコンディションコードレジスタ (CCR) の I フラグを割込み不可 (I=0<sub>H</sub>) に設定するかまたは割込みレベルマスクレジスタ (ILM) を割込み不可 (ILM=000<sub>B</sub>) に設定すると、一時的に禁止にすることができます。

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を使用して複数の割込みを起動することはできません。拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の処理中は、ほかのすべての割込み要求および拡張インテリジェント I/O サービス要求は保留されます。

#### ● 複数の割込み処理例

複数の割込み処理例では、タイマ割込みの優先度が A/D コンバータ割込みの優先度よりも高いものと仮定します。この例では、A/D コンバータの割込みレベルとして "2" が設定されており、タイマの割込みレベルとしては "1" が設定されています。A/D コンバータ割込み処理中にタイマ割込みが発生すると、図 7.4-5 のように処理します。

図 7.4-5 複数の割り込み処理例



- A/D 割り込み発生
  - A/D コンバータ割り込み処理が開始されると、割り込みレベルマスクレジスタ (ILM) には自動的に A/D コンバータ割り込みレベル (ICR レジスタの IL2 ~ IL0:bit2 ~ bit0) と同じ値 (例では "2") が設定されます。レベル "1" または "0" の割り込み要求が発生した場合は、これらの割り込み処理が優先されます。
- 割り込み処理の終了
  - 割り込み処理が終了し、復帰命令 (RETI) が実行されると、専用レジスタ (A, DPR, ADB, DTB, PCB, PC, PS) の値がスタックから戻され、割り込みレベルマスクレジスタ (ILM) には、割り込み発生前の値が設定されます。

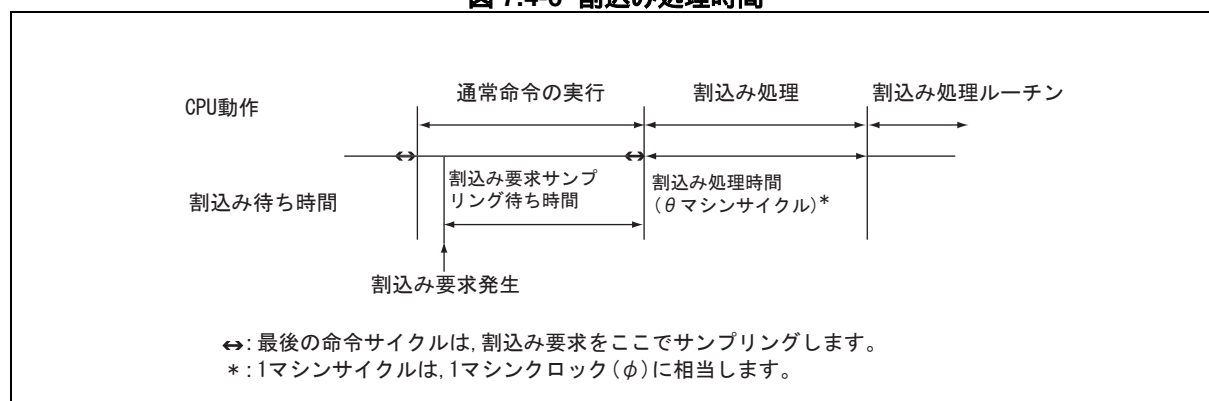
## 7.4.5 ハードウェア割り込みの処理に要する時間

ハードウェア割り込み要求が発生してから割り込み処理ルーチンを実行するまでには、現在実行中の命令を終了させるために要する時間と、割り込みを処理するために要する時間が必要です。

### ■ ハードウェア割り込みの処理に要する時間

ハードウェア割り込み要求が発生してから割り込みを受け、割り込み処理ルーチンを実行するまでには、割り込み要求をサンプリングするための待ち時間と、割り込みを処理するために要する時間（割り込み処理の準備に要する時間）が必要です。割り込みの処理に要する時間を図 7.4-6 に示します。

図 7.4-6 割り込み処理時間



#### ● 割り込み要求サンプリング待ち時間

割り込み要求サンプリング待ち時間は、割り込み要求が生成された時点から現在実行中の命令が終了した時点までの時間です。

割り込み要求が発生したか否かは、最後の命令サイクルの割り込み要求をサンプリングすることによって判定されます。各命令の実行中はCPUは割り込み要求を確認できず、待ち時間が発生します。

割り込み要求サンプリング待ち時間は、実行時間が最も長いPOPW RW0, ...RW7命令(45マシンサイクル)が開始された直後に割り込み要求が発生した時点で最大になります。

#### ● 割り込み処理時間 (θ マシンサイクル)

CPUは、割り込み要求を受け取ると、システムスタックに専用レジスタを退避し、割り込みベクタを読み出します。割り込み処理時間は、φ マシンサイクルです。割り込み処理時間は、以下に示す式で算出されます。

- 割り込みがアクティブになったとき :  $\theta = 24 + 6 + Z$  マシンサイクル
- 割り込みから制御が返されたとき :  $\theta = 11 + 6 + Z$  マシンサイクル (RETI 命令)

割り込み処理時間は、スタックポインタが指す各アドレスによって異なります。

割込み処理時間の補間値 (Z) を表 7.4-3 に示します。

**表 7.4-3 割込み処理時間の補間値 (Z)**

スタックポインタが指すアドレス	補間値 (Z)
外部 8 ビット	+4
偶数番号の外部アドレス	+1
奇数番号の外部アドレス	+4
偶数番号の内部アドレス	0
奇数番号の内部アドレス	+2

---

< 参考 >      1 マシンサイクルは, マシントック (φ) の 1 サイクルに相当します。

---

## 7.5 ソフトウェア割込み

---

ソフトウェア割込み命令 (INT 命令) を実行すると、ソフトウェア割込み機能は、CPU で実行中のプログラムからユーザ定義の割込み処理プログラムへ制御を移します。ハードウェア割込みは、ソフトウェア割込み実行中は禁止になります。

---

### ■ ソフトウェア割込みの起動

#### ● ソフトウェア割込みの起動

INT 命令は、ソフトウェア割込みを起動するために使用します。ソフトウェア割込み要求の場合、割込み要求フラグや割込み許可フラグは存在しません。INT 命令を実行すると、割込み要求が必ず発生します。

#### ● ハードウェア割込みの抑制

INT 命令は割込みレベルを持たないので、割込みレベルマスクレジスタ (ILM) は更新されません。INT 命令の実行中、コンディションコードレジスタ (CCR) の I フラグには "0" が設定され、ハードウェア割込みはマスクされます。

ソフトウェア割込み処理中にハードウェア割込みを許可するには、ソフトウェア割込み処理ルーチンにおいて I フラグに "1" を設定してください。

#### ● ソフトウェア割込み動作

CPU が INT 命令を読み出すと、ソフトウェア割込み処理マイクロコードがアクティブになります。このマイクロコードは、CPU 内部レジスタをシステムスタックに退避し、ハードウェア割込みをマスクし (CCR レジスタの I フラグ = 0)、対応する割込みベクタへ分岐します。

割込み番号と割込みベクタの割当てについての詳細は、「7.2 割込み要因と割込みベクタ」をご参照ください。

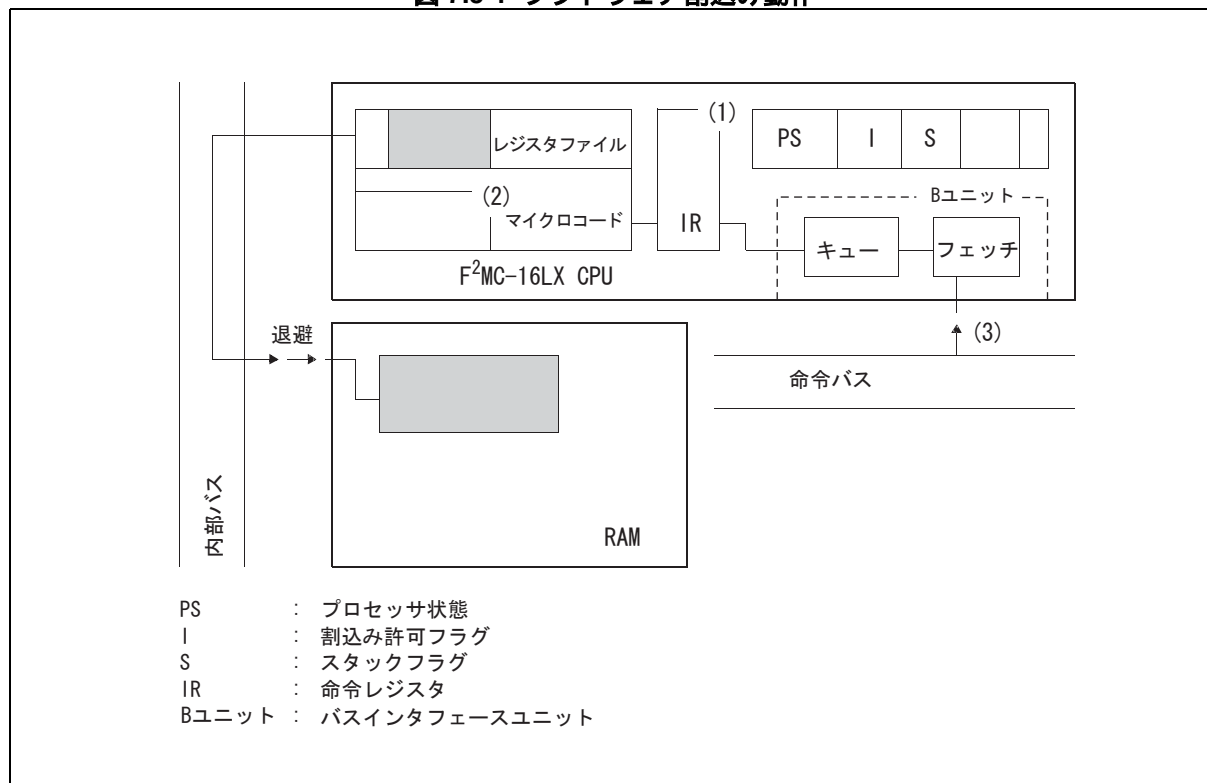
### ■ ソフトウェア割込みからの復帰

割込み処理プログラムでは、割込み復帰命令 (RETI 命令) を実行すると、システムスタックに退避されている 12 バイトデータが専用レジスタへ復元され、割込みへ分岐する前に実行されていた処理を再開します。

## ■ ソフトウェア割り込み動作

ソフトウェア割り込みが生成されてから割り込み処理が完了するまでのソフトウェア割り込み動作を図 7.5-1 に示します。

図 7.5-1 ソフトウェア割り込み動作



(1)ソフトウェア割り込み命令が実行されます。

(2)専用レジスタが、ソフトウェア割り込み命令に対応するマイクロコードに従って退避されます。またはほかの必要な処理が実行されます。次に、分岐処理が実行されます。

(3)ユーザの割り込み処理ルーチン中の RETI 命令が割り込み処理を終了させます。

プログラムバンクレジスタ (PCB) が "FF<sub>H</sub>" の場合、CALLV 命令のベクタ領域が INT #vct8 命令テーブルとオーバーラップします。ソフトウェアを作成する場合は、CALLV 命令と INT #vct8 命令の重複アドレスに注意してください。



## 7.6 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) からの割込み

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) は、自動的にリソース (I/O 領域) とメモリの間においてデータを転送します。データ転送が終了すると、ハードウェア割込みが発生します。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS)

拡張インテリジェント I/O サービスは、一種のハードウェア割込みです。拡張インテリジェント I/O サービスは、自動的にリソース (I/O 領域) とメモリの間においてデータを転送します。従来はリソース (I/O 領域) とメモリの間におけるデータ転送は、割込み処理プログラムが行っていました。EI<sup>2</sup>OS は、直接メモリアクセス (DMA) と同じ方法でこのデータを転送します。終了時に EI<sup>2</sup>OS は、終了コードを設定し、自動的に割込み処理ルーチンへ分岐します。ユーザは、EI<sup>2</sup>OS を起動および終了するためのプログラムのみを作成する必要があり、データ転送プログラムを作成する必要はありません。

#### ● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の利点

割込み処理ルーチンで実行されるデータ転送と比較すると、EI<sup>2</sup>OS の場合は以下の利点があります。

- データ転送プログラムをコード化する必要がなく、プログラムサイズを小さくできます。
- データ転送はリソース (I/O 領域) の状態によって停止できるので、不要なデータ転送を除外できます。
- バッファアドレスを更新するかまたは更新しないかを選択できます。
- I/O アドレスポインタを更新するかまたは更新しないかを選択できます。

#### ● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) からの終了割込み

EI<sup>2</sup>OS によるデータ転送が終了すると、終了状態は割込み制御レジスタ (ICR) の S1, S0: bit5, bit4 に設定されます。したがって、処理は自動的に割込み処理ルーチンへ分岐します。

EI<sup>2</sup>OS 終了要因は、割込み処理プログラムで EI<sup>2</sup>OS 状態 (ICR レジスタの S1, S0: bit5, bit4) をチェックすることで決定できます。

割込み番号と割込みベクタは、各リソースに対して固定されます。詳細は、「7.2 割込み要因と割込みベクタ」をご参照ください。

#### ● 割込み制御レジスタ (ICR)

割込みコントローラに存在するこのレジスタは、EI<sup>2</sup>OS をアクティブにし、EI<sup>2</sup>OS チャネルを指定し、EI<sup>2</sup>OS 終了状態を表示します。

### ● 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のディスクリプタ (ISD)

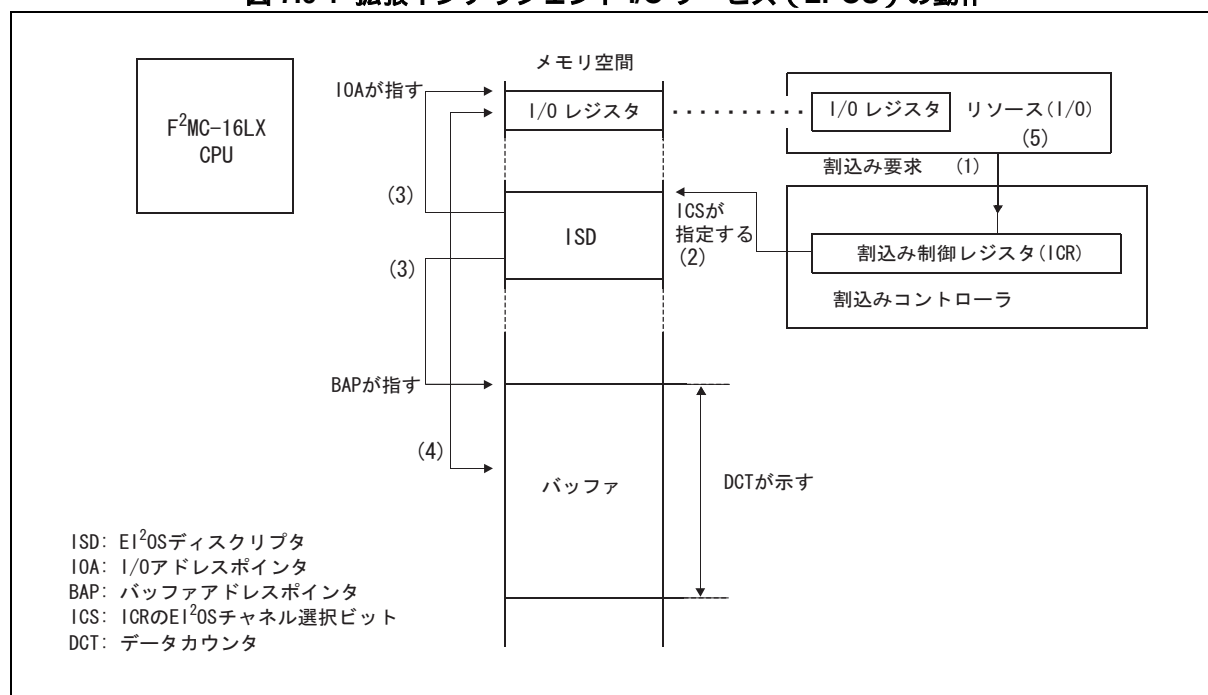
このディスクリプタは, "000100<sub>H</sub>" から "00017F<sub>H</sub>" までの内部 RAM に存在し, 転送モード, I/O アドレス, 転送数およびバッファアドレスを保持する "8 バイト × 16" チャンネルで構成されています。チャンネルは, 割り込み制御レジスタ (ICR) で指定します。

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作中, CPU プログラムの実行は停止します。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作

EI<sup>2</sup>OS の動作を図 7.6-1 に示します。

図 7.6-1 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作



- (1) 割り込み要求が発生し, EI<sup>2</sup>OS が起動されます。
- (2) 割り込みコントローラがディスクリプタを選択します。
- (3) ディスクリプタから転送元と転送先が読み出されます。
- (4) I/O 領域とメモリの間で転送が行われます。
- (5) 割り込み要因が自動的にクリアされます。

## 7.6.1 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のディスクリプタ (ISD)

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のディスクリプタ (ISD) は, "000100<sub>H</sub>" から "00017F<sub>H</sub>" までの内部 RAM に存在します。ISD は, "8 バイト × 16" チャンネルで構成されています。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のディスクリプタ (ISD) の構成

ISD は, "8 バイト × 16" チャンネルで構成されています。各 ISD は, 図 7.6-2 に示されている各ビットから構成されています。チャンネル番号と ISD アドレスの対応関係を表 7.6-1 に示します。

図 7.6-2 EI<sup>2</sup>OS のディスクリプタ (ISD) の構成

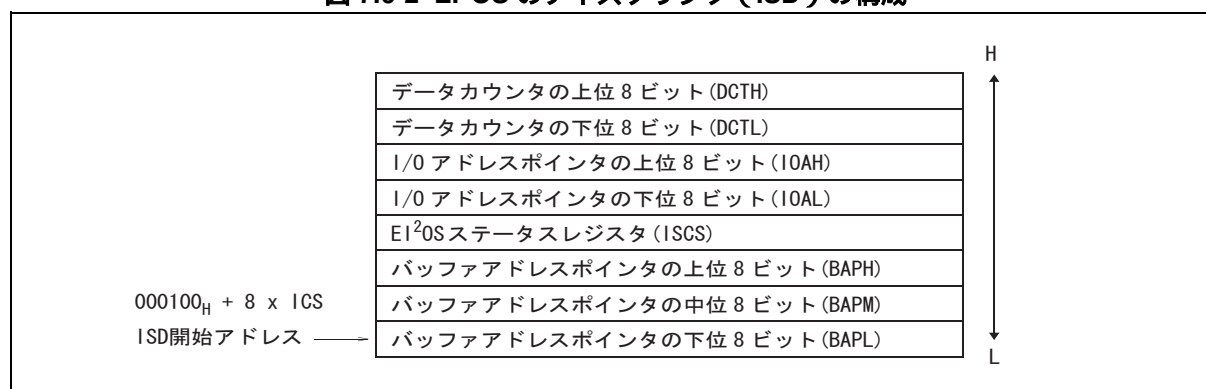


表 7.6-1 チャンネル番号とディスクリプタアドレスの対応関係

チャンネル	ディスクリプタアドレス
0	000100 <sub>H</sub>
1	000108 <sub>H</sub>
2	000110 <sub>H</sub>
3	000118 <sub>H</sub>
4	000120 <sub>H</sub>
5	000128 <sub>H</sub>
6	000130 <sub>H</sub>
7	000138 <sub>H</sub>
8	000140 <sub>H</sub>
9	000148 <sub>H</sub>
10	000150 <sub>H</sub>
11	000158 <sub>H</sub>
12	000160 <sub>H</sub>
13	000168 <sub>H</sub>
14	000170 <sub>H</sub>
15	000178 <sub>H</sub>

## 7.6.2 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のディスクリプタ (ISD) のレジスタ構成

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) ディスクリプタ (ISD) は、以下のレジスタで構成されています。

- データカウンタ (DCT)
- I/O レジスタアドレスポインタ (IOA)
- EI<sup>2</sup>OS ステータスレジスタ (ISCS)
- バッファアドレスポインタ (BAP)

リセット後の各レジスタの初期値は不定なので注意してください。

### ■ データカウンタ (DCT)

DCT は、データ転送数のカウンタとしての役割を果たす 16 ビットレジスタです。各データ転送が実行されると、カウンタは 1 ずつデクリメントします。カウンタが "0" に達すると、EI<sup>2</sup>OS は終了します。

DCT の構成を図 7.6-3 に示します。

図 7.6-3 DCT の構成

	bit	15	14	13	12	11	10	9	8	
データカウンタの上位バイト		B15	B14	B13	B12	B11	B10	B09	B08	DCTH
初期値 ⇒		X	X	X	X	X	X	X	X	
	bit	7	6	5	4	3	2	1	0	
データカウンタの下位バイト		B07	B06	B05	B04	B03	B02	B01	B00	DCTL
初期値 ⇒		X	X	X	X	X	X	X	X	

### ■ I/O レジスタアドレスポインタ (IOA)

IOA は、バッファへ (から) データを転送するために使用される、I/O レジスタの下位アドレス (A15 ~ A00) を示す 16 ビットレジスタです。上位アドレス (A23 ~ A16) は、すべて "00<sub>H</sub>" です。"000000<sub>H</sub>" から "00FFFF<sub>H</sub>" までの領域は、すべてアドレスで指定できます。

IOA の構成を図 7.6-4 に示します。

図 7.6-4 I/O レジスタアドレスポインタ (IOA) の構成

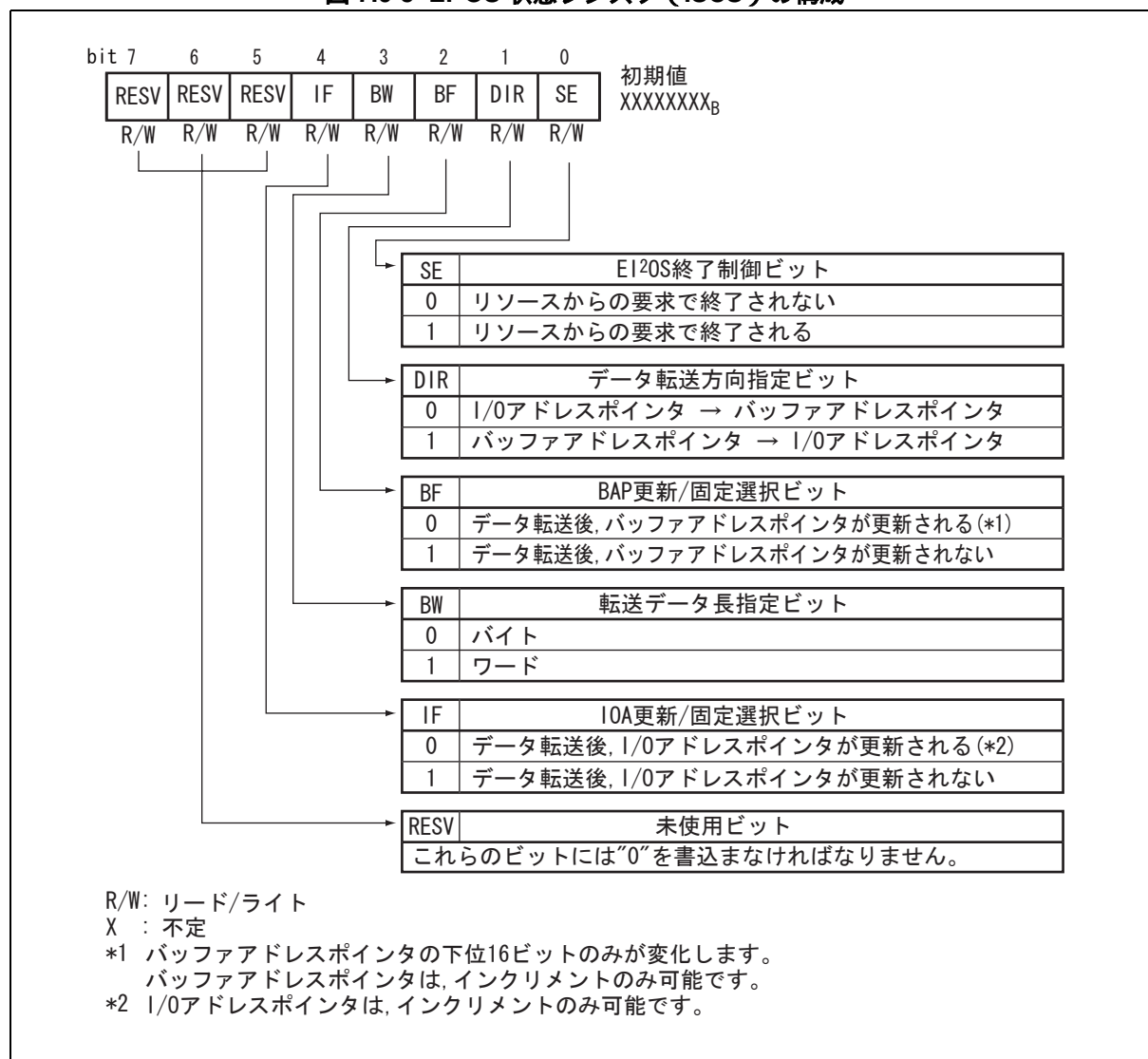
	bit	15	14	13	12	11	10	9	8	
上位アドレスポインタ		A15	A14	A13	A12	A11	A10	A09	A08	IOAH
初期値 ⇒		X	X	X	X	X	X	X	X	
	bit	7	6	5	4	3	2	1	0	
下位アドレスポインタ		A07	A06	A05	A04	A03	A02	A01	A00	IOAL
初期値 ⇒		X	X	X	X	X	X	X	X	

## ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のステータスレジスタ (ISCS)

ISCS は、8 ビットレジスタです。ISCS は、バッファアドレスポインタが更新されるか否か、I/O アドレスポインタが更新されるか否かおよび転送データフォーマット (バイトまたはワード) と転送方向を示します。

ISCS の構成を図 7.6-5 に示します。

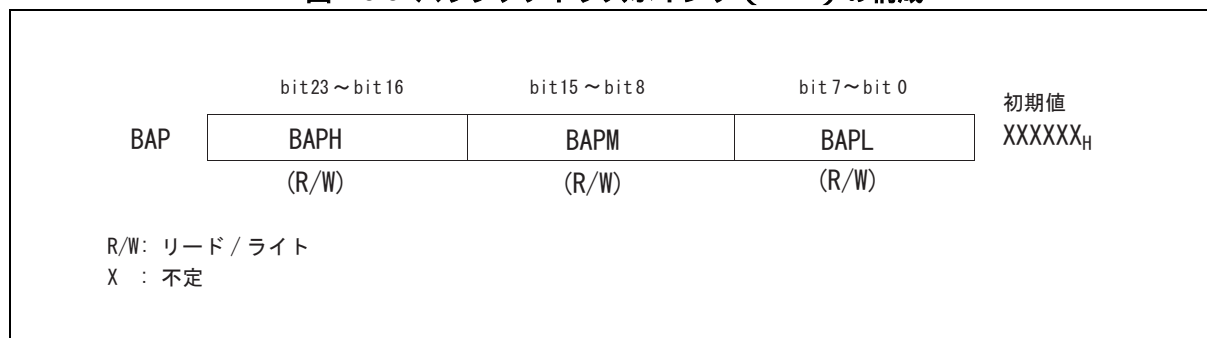
図 7.6-5 EI<sup>2</sup>OS 状態レジスタ (ISCS) の構成



## ■ バッファアドレスポインタ (BAP)

BAP は、その次の転送で EI<sup>2</sup>OS が使用するアドレスを保持する 24 ビットレジスタです。各 EI<sup>2</sup>OS チャンネルには 1 つの独立した BAP が存在するので、各 EI<sup>2</sup>OS チャンネルは 16 メガバイト空間内のアドレスと I/O 領域との間においてデータを転送できます。EI<sup>2</sup>OS 状態レジスタ (ISCS) の BF:bit2 (BAP 更新 / 固定選択ビット) に "0" を設定すると、BAP の下位 16 ビット (BAPM, BAPL) のみが変わり、上位 8 ビット (BAPH) は変化しません。BAP の構成を図 7.6-6 に示します。

図 7.6-6 バッファアドレスポインタ (BAP) の構成



- < 参考 >
- I/O アドレスポインタ (IOA) で指定できる領域は, "000000<sub>H</sub>" ~ "00FFFF<sub>H</sub>" までです。
  - バッファアドレスポインタ (BAP) で指定できる領域は, "000000<sub>H</sub>" ~ "FFFFFF<sub>H</sub>" までです。
  - データカウンタ (DCT) で指定できる最大転送数は, 65, 536 回 (64K バイト) です。

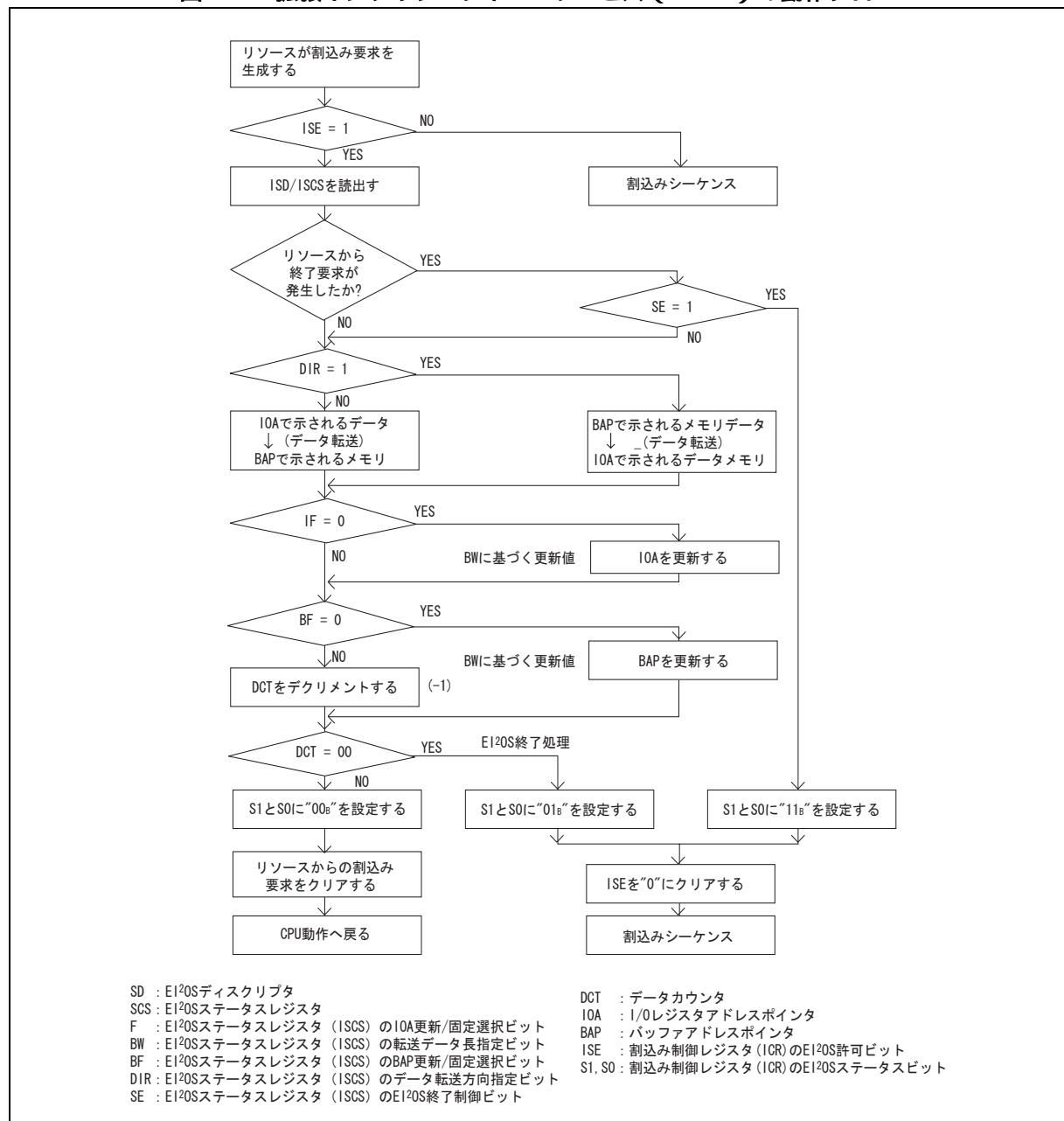
## 7.6.3 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作

割り込み要求がリソースにおいて発生すると、対応する割り込み制御レジスタ (ICR) に EI<sup>2</sup>OS の起動が設定され、CPU はデータを転送するために EI<sup>2</sup>OS を使用します。指定されたデータ転送数に達すると、ハードウェア割り込みが自動的に処理されます。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作フロー

CPU の内部マイクロコードに基づく EI<sup>2</sup>OS 動作フローを図 7.6-7 に示します。

図 7.6-7 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の動作フロー



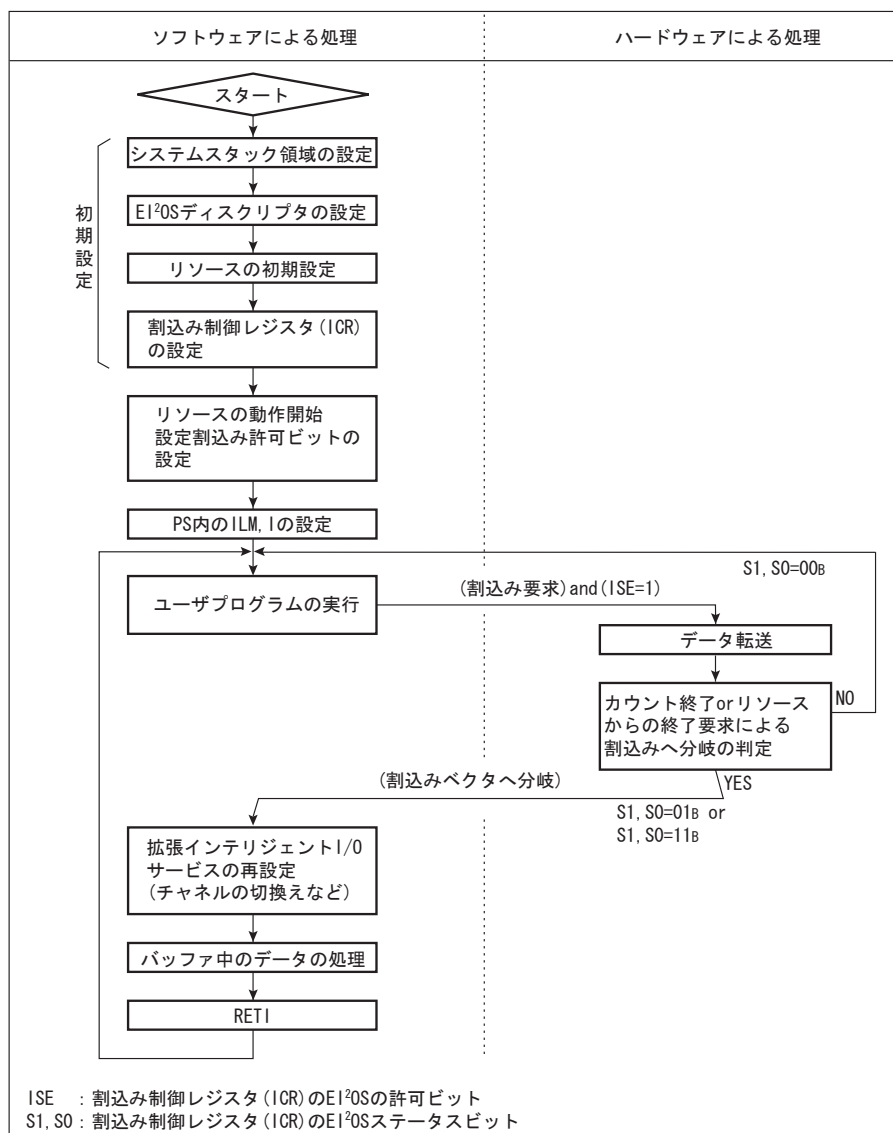
## 7.6.4 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の使用手順

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を使用する場合は、事前にシステムスタック領域、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) ディスクリプタ、割り込み機能および割り込み制御レジスタ (ICR) を設定しておかなければなりません。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の使用手順

EI<sup>2</sup>OS ソフトウェア / ハードウェア処理を図 7.6-8 に示します。

図 7.6-8 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の使用手順





## 7.6.5 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の処理時間

拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の処理に要する時間は、以下の要素によって変化します。

- EI<sup>2</sup>OS ステータスレジスタ (ISCS) の設定
- I/O レジスタアドレスポインタ (IOA) が指すアドレス (領域)
- バッファアドレスポインタ (BAP) が指すアドレス (領域)
- 外部アクセス用の外部データバス幅
- 転送データ長

EI<sup>2</sup>OS によるデータ転送が終了するとハードウェア割込みが起動するので、割込み処理時間が追加されます。

### ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) の処理時間 (1 回のデータ転送に要する時間)

#### ● データ転送を継続する場合

データ転送を継続する場合の EI<sup>2</sup>OS 処理時間を表 7.6-2 に示します(この表は EI<sup>2</sup>OS ステータスレジスタ (ISCS) の設定に基づいています)。

表 7.6-2 拡張インテリジェント I/O サービスの実行時間

EI <sup>2</sup> OS 終了制御ビット (SE:bit0) の設定		リソースからの終了要求で終了する		リソースからの終了要求を無視する	
IOA 更新 / 固定選択ビット (IF:bit4) の設定		固定	更新	固定	更新
BAP アドレス更新 / 固定選択ビット (BF:bit2) の設定	固定	32	34	33	35
	更新	34	36	35	37

単位: マシンサイクル (1 マシンサイクルはマシクロック ( ) の 1 サイクルに相当します。)

さらに表 7.6-3 に示しているように、EI<sup>2</sup>OS の実行状態によっては補間する必要があります。

表 7.6-3 EI<sup>2</sup>OS 実行時間のデータ転送補間値

I/O レジスタアドレスポインタ			内部アクセス		外部アクセス	
			B/ 偶数	奇数	B/ 偶数	8/ 奇数
バッファアドレスポインタ	内部アクセス	B/ 偶数	0	+2	+1	+4
		奇数	+2	+4	+3	+6
	外部アクセス	B/ 偶数	+1	+3	+2	+5
		8/ 奇数	+4	+6	+5	+8

B : バイトデータ転送

8 : 外部バス幅 8 ビットでワード転送

偶数 : 偶数番号アドレスのワード転送

奇数 : 奇数番号アドレスのワード転送

● データカウンタ (DCT) によるカウントを終了する場合 (最後のデータ転送)

EI<sup>2</sup>OSによるデータ転送が終了するとハードウェア割り込みが起動するので、割り込み処理時間が追加されます。カウントを終了させるために要する EI<sup>2</sup>OS 処理時間は、以下の式で算出されます。

**カウントを終了させるために要する EI<sup>2</sup>OS 処理時間 = データ転送に要する EI<sup>2</sup>OS 処理時間 + 割り込み処理時間 (21 + 6 × Z) マシンサイクル**

割り込み処理時間は、スタックポインタが指す各アドレスによって異なります。割り込み処理時間の補間値 (Z) を表 7.6-4 に示します。

**表 7.6-4 割り込み処理時間の補間値 (Z)**

スタックポインタが指すアドレス	補間値 (Z)
外部 8 ビット	+4
偶数番号の外部アドレス	+1
奇数番号の外部アドレス	+4
偶数番号の内部アドレス	0
奇数番号の内部アドレス	+2

● リソース (I/O) からの終了要求で終了する場合

EI<sup>2</sup>OS によるデータ転送をリソース (I/O 領域) からの終了要求で終了すると (ICR レジスタの S1, S0 : bit5, bit4=11<sub>B</sub>)、データは転送されず、ハードウェア割り込みが起動します。EI<sup>2</sup>OS 処理時間は、以下の式で算出されます。式中の "Z" は、割り込み処理時間の補間値を示します (表 7.6-4)。

**データ転送を終了させるために要する EI<sup>2</sup>OS 処理時間 = (36 + 6 × Z) マシンサイクル**

---

< 参考 >      1 マシンサイクルは、マシクロック (φ) の 1 サイクルに相当します。

---

## 7.7 例外処理割込み

---

F<sup>2</sup>MC-16LX ファミリでは、未定義命令を実行すると例外処理が発生します。

例外処理は、基本的には割込みと同じです。例外処理の発生を命令と命令の間で検出すると、通常処理が中断され、例外処理が実行されます。

一般に、例外処理は予期しない動作の結果として発生します。例外処理は、デバッグ時や緊急時に必要となる復帰ソフトウェアを起動する場合にのみ使用してください。

---

### ■ 例外処理

#### ● 例外処理の動作

F<sup>2</sup>MC-16LX ファミリは、命令マップにおいて未定義命令として定義されていない、すべてのコードを扱います。未定義命令を実行すると、INT #10 ソフトウェア割込み命令に相当する処理が実行されます。

例外処理が割込みルーチンへ分岐する前に以下の処理が実行されます。

- A, DPR, ADB, DTB, PCB, PC, PS レジスタがシステムスタックに退避されます。
- 状態コードレジスタ (CCR) の I フラグが "0" にクリアされ、ハードウェア割込みがマスクされます。
- コンディションコードレジスタ (CCR) の S フラグに "1" が設定され、システムスタックがアクティブになります。

システムスタックに退避されたプログラムカウンタ (PC) 値は、未定義命令が格納されるアドレスそのものです。2 バイト以上の命令コードの場合、"未定義" として識別されたコードはこのアドレスに格納されます。例外処理ルーチン内で例外要因の種類を決定しなければならない場合は、この PC 値を使用してください。

#### ● 例外処理からの復帰

RETI 命令で例外処理から制御が返されると、PC が未定義命令を指しているので、例外処理が再開されます。ソフトウェアリセットを行うなどで解決してください。

## 7.8 割り込み処理時のスタック動作

割り込みが一度受け付けられると、割り込み処理へ分岐する前に専用レジスタの値が自動的にシステムスタックへ退避されます。割り込み処理が終了すると、退避されていた専用レジスタは自動的にシステムスタックから復元します。

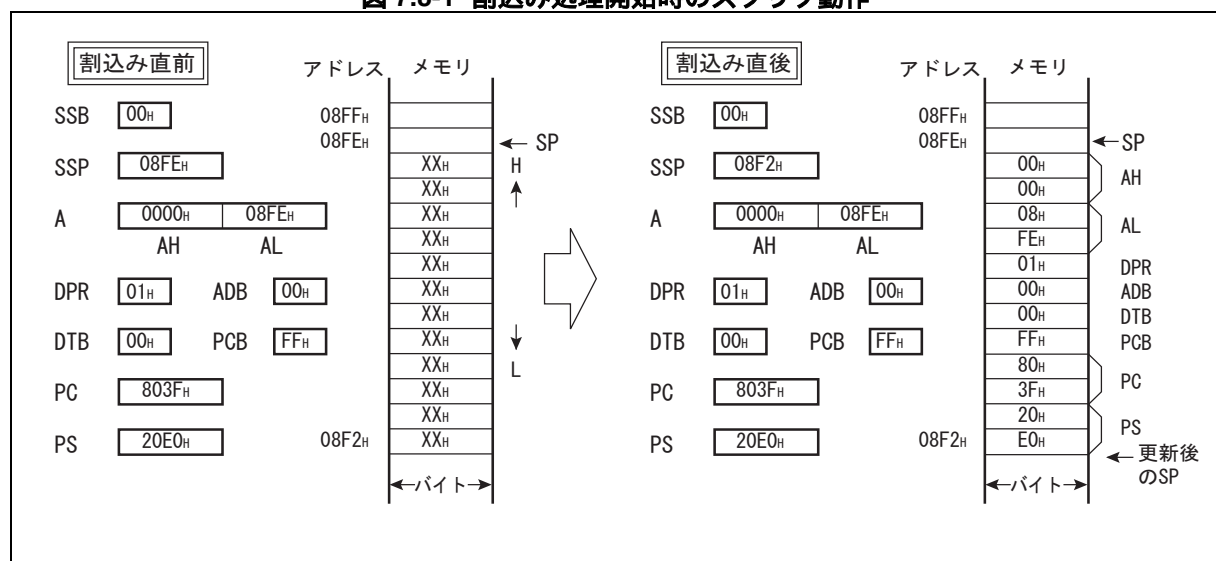
### ■ 割り込み処理開始時のスタック動作

割り込みが一度受け付けられると、CPU は自動的に現在の専用レジスタの値を、以下に示された順序でシステムスタックへ退避します。

- アキュムレータ (A)
- ダイレクトページレジスタ (DPR)
- アディショナルデータバンクレジスタ (ADB)
- データバンクレジスタ (DTB)
- プログラムバンクレジスタ (PCB)
- プログラムカウンタ (PC)
- プロセッサステータス (PS)

割り込み処理開始時のスタック動作を図 7.8-1 に示します。

図 7.8-1 割り込み処理開始時のスタック動作



### ■ 割り込み処理からの復帰時のスタック動作

割り込み処理終了時に割り込み復帰命令 (RETI) を実行すると、専用レジスタ値は退避時とは逆の順序 (PS, PC, PCB, DTB, ADB, DPR, A) でスタックから返されます。専用レジスタは、割り込み処理開始直前の状態に復元されます。

## ■ スタック領域

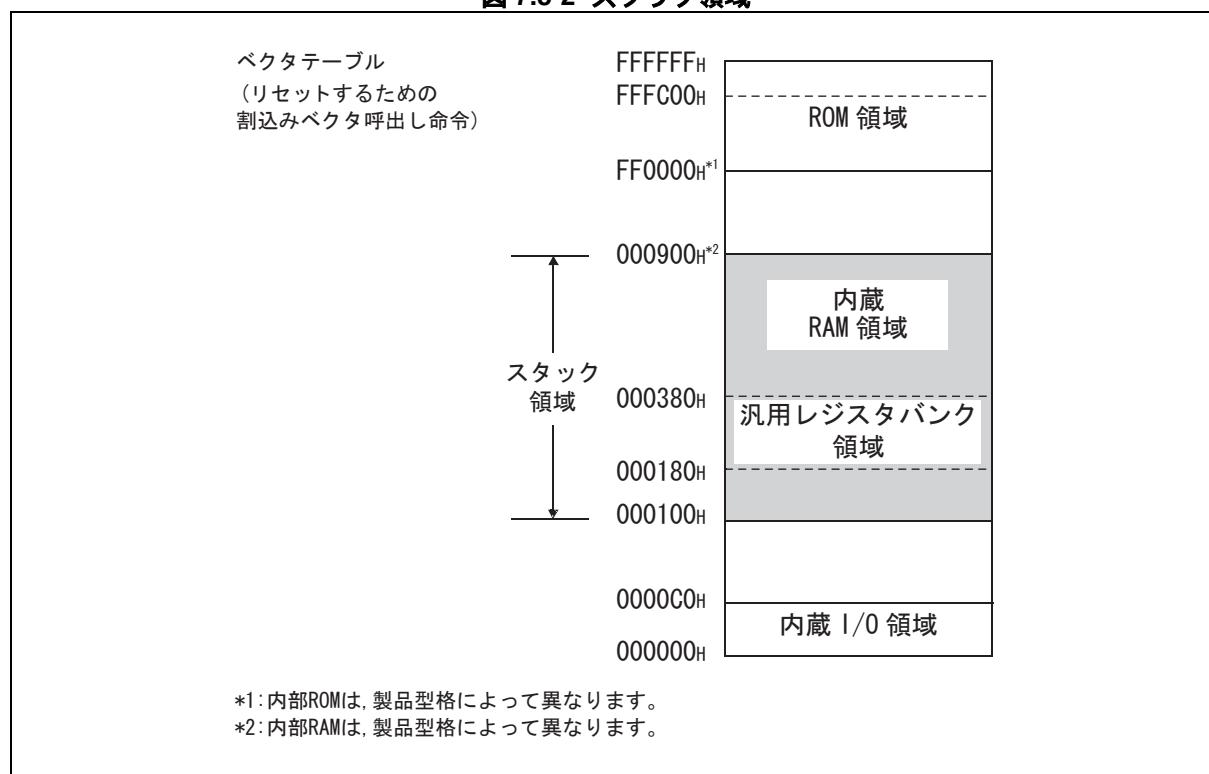
### ● スタック領域の割当て

スタック領域は、割込み処理時に加え、サブルーチン呼出し命令（CALL）やベクタ呼出し命令（CALLV）実行時にもプログラムカウンタ（PC）を退避および復元するために使用します。スタック領域は、PUSHW 命令や POPW 命令でレジスタを一時的に退避または復元するために使用します。

スタック領域は、RAM のデータ領域と一緒に割当てられます。

スタック領域を図 7.8-2 スタック領域に示します。

図 7.8-2 スタック領域



- 一般的に、偶数番号のアドレスはスタックポインタ( SSP, USP )へ設定してください。
- オーバラップが発生しないように注意して、システムスタック領域やユーザスタック領域、データ領域を割当ててください。

### ● システムスタックとユーザスタック

システムスタック領域は、割込み処理に使用されます。割込みが発生すると、使用中のユーザスタック領域は強制的にシステムスタックに切り換えられます。システムスタック領域は、主にユーザスタック領域を使用するシステムの場合でも正しく設定しなければなりません。

スタック空間を分割する必要が特でない場合は、システムスタックのみを使用してください。

## 7.9 割り込み処理のプログラム例

割り込み処理のプログラム例を示します。

### ■ 割り込み処理のプログラム例

#### ● 処理仕様

外部割り込み 0 (INT0) を使用する割り込み処理プログラム例を以下に示します。

#### ● コーディング例

```

DDR1    EQU    000011H    ; ポート 1 方向レジスタ
ENIR     EQU    030H      ; DTP/ 外部割り込み許可レジスタ
EIRR     EQU    031H      ; DTP/ 外部割り込み要因レジスタ
ELVR     EQU    032H      ; 要求レベル設定レジスタ
ICR04    EQU    0B4H      ; 割り込み制御レジスタ
STACK    SSEG            ; スタック
        RW      100
STACK_T  RW      1
STACK    ENDS
;----- メインプログラム -----
CODE     CSEG
START:
        MOV     RP, #0      ; 汎用レジスタは先頭バンクを使用
        MOV     ILM, #07H   ; PS の ILM をレベル 7 に設定
        MOV     A, #!STACK_T; システムスタックを設定
        MOV     SSB, A
        MOVW    A, #STACK_T ; スタックポインタを設定し、次に
        MOVW    SP, A       ; SSP を設定する (S フラグ = 1 なので)
        MOV     DDR1, #00000000B; P10/INT0 端子を " 入力 " に設定
        OR      CCR, #40H   ; PS の CCR の I フラグを設定して割り込みを許可
                        する
        MOV     I:ICR04, #00H ; 割り込みレベルを "0" (優先度が最も高い) に設定
        MOV     I:ELVR, #00000001B; INT0 に H レベルを入力するよう要求
        MOV     I:EIRR, #00H; INT0 割り込み要因をクリア
        MOV     I:EIRR, #01H; INT0 入力を許可する
        :
LOOP:    NOP                ; ダミーループ
        NOP
        NOP
        NOP
        BRA     LOOP        ; 無条件ジャンプ
;----- 割り込みプログラム -----
ED_INT1:
        MOV     I:EIRR, #00H; 新しい INT0 の受け付け不可
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        RETI                ; 割り込みからの復帰
CODE     ENDS

```

```

;----- ベクタ設定 -----
VECT    CSEG ABS=0FFH
        ORG    0FFACH      ; 割り込み #20 (14H) のベクタを設定
        DSL    ED_INT1
        ORG    0FFDCH      ; リセットベクタを設定
        DSL    START
        DB     00H          ; シングルチップモードを設定
VECT    ENDS
        END    START

```

## ■ 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のプログラム例

- このプログラムは、INT0 端子へ入力された H レベル信号を検出し、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を起動します。
- INT0 端子へ H レベルが入力されると、EI<sup>2</sup>OS が起動します。データは、ポート 0 からアドレス "003000<sub>H</sub>" のメモリへ転送されます。
- 転送データバイト数は、100 バイトです。100 バイトの転送が完了すると、EI<sup>2</sup>OS 転送が終了したので割り込みが発生します。

### ● コーディング例

```

DDR1    EQU    000011H      ; ポート 1 方向レジスタ
ENIR     EQU    000030H      ; DTP / 外部割り込み許可レジスタ
EIRR     EQU    000031H      ; DTP / 外部割り込み要因レジスタ
ELVR     EQU    000032H      ; 要求レベル設定レジスタ
ICR04    EQU    0000B4H      ; 割り込み制御レジスタ
BAPL     EQU    000100H      ; バッファアドレスポインタ下位
BAPM     EQU    000101H      ; バッファアドレスポインタ中位
BAPH     EQU    000102H      ; バッファアドレスポインタ上位
ISCS     EQU    000103H      ; EI2OS 状態
IOAL     EQU    000104H      ; I/O アドレスポインタ下位
IOAH     EQU    000105H      ; I/O アドレスポインタ上位
DCTL     EQU    000106H      ; データカウンタ下位
DCTH     EQU    000107H      ; データカウンタ上位
ER0      EQU    EIRR:0       ; 外部割り込み要求フラグビットの定義
STACK    SSEG
        RW     100
STACK_T  RW     1
STACK    ENDS

;----- メインプログラム -----
CODE     CSEG
START:
        AND    CCR,#0BFH      ; PS の CCR の I フラグをクリアして割り込みを
                                禁止
        MOV    RP,#00         ; レジスタバンクポインタを設定
        MOV    A,#STACK_T     ; システムスタックを設定
        MOV    SSB,A
        MOVW   A,#STACK_T     ; スタックポインタを設定し、次に
        MOVW   SP,A           ; SSP を設定する (S フラグ =1 なので)
        MOV    I:DDR1,#00000000B; P10/INT0 端子を " 入力 " に設定
        MOV    BAPL,#00H      ; バッファアドレス (003000H) を設定
        MOV    BAPM,#30H
        MOV    BAPH,#00H

```

```

MOV   ISCS, #00010001B    ; I/O アドレス更新, バイト転送, バッファアドレスを更新しない
                                ; I/O バッファへ転送, リソースで終了させる
MOV   IOAL, #00H           ; 転送元アドレス (ポート 0: 000000H) を設定
MOV   IOAH, #00H
MOV   DCTL, #64H           ; 転送バイト数 (100 バイト) を設定
MOV   DCTH, #00H
MOV   I:ICR04, #00001000B; EI2OS チャンネル 0, EI2OS 許可,
                                ; 割り込みレベル 0 (優先度が最も高い)
MOV   I:ELVR, #00000001B ; INT0 に H レベルを入力するよう要求
MOV   I:EIRR, #00H        ; INT0 割り込み要因をクリア
MOV   I:ENIR, #01H        ; INT0 割り込みを許可する
        MOV   ILM, #07H    ; PS の ILM をレベル 7 に設定
        OR    CCR, #40H    ; PS の CCR の I フラグを設定して割り込みを許可する
        :
LOOP   BRA    LOOP        ; 無限ループ
; ----- 割り込みプログラム -----
WARI   CLRB   ER0; DTP / 外部割り込み要求フラグをクリア
        :
        ユーザ処理        ; EI2OS の終了要因をチェックし,
        :                ; バッファのデータを処理し, EI2OS を再び設定
        RETI
CODE   ENDS
; ----- ベクタ設定 -----
VECT   CSEG   ABS=0FFH
        ORG    0FFACH    ; 割り込み #20 (14H) のベクタを設定
        DSL    WARI
        ORG    0FFDCH    ; リセットベクタを設定
        DSL    START
        DB     00H       ; シングルチップモードを設定
VECT   ENDS
        END    START

```





# 第8章

---

## モード設定

MB90460/465 シリーズの動作モードについて説明します。

8.1 モード設定

8.2 モード端子 (MD2 ~ MD0)

8.3 モードデータレジスタ

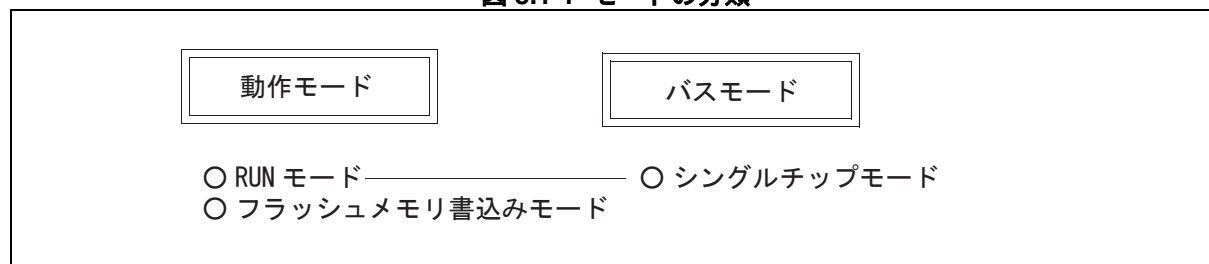
## 8.1 モード設定

リセット入力時のモード端子レベルの設定と、モードレジスタにモードデータを設定することにより動作モードを設定できます。

### ■ モード設定

モードの分類を図 8.1-1 に示します。

図 8.1-1 モードの分類



### ■ 動作モード

動作モードは、モード端子 (MD2 ~ MD0) とモードデータレジスタのバスモード設定ビット (M1, M0:bit7, bit6) で設定します。設定した動作モードによって起動します。

< 注意事項 > MD2 ~ MD0 端子に "011<sub>B</sub>" を、モードデータレジスタのバスモード設定ビット (M1, M0:bit7, bit6) は "00<sub>B</sub>" を設定してください。

### ■ バスモード

MB90460/465 シリーズはシングルチップモードのみ設定できます。

詳細は「8.2 モード端子 (MD2 ~ MD0)」と「8.3 モードデータレジスタ」をご参照ください。

### ■ RUN モード

RUN モードは、CPU が動作しているモードです。RUN モードとしては、メインクロックで動作するメインクロックモード、PLL クロックで動作する PLL クロックモードおよび低消費電力モードがあります。詳細は、「第 6 章 低消費電力モード」をご参照ください。

## 8.2 モード端子 (MD2 ~ MD0)

モード端子は、MD2 ~ MD0 の 3 本の外部端子であり、リセットベクタとモードデータの取込み方法を設定します。

### ■ モード端子 (MD2 ~ MD0)

モード端子によりモードを設定します。

フラッシュメモリ内蔵品の場合は、内蔵フラッシュメモリにプログラムを書き込むためのフラッシュメモリ書込みモードも、これらの端子により設定します。

表 8.2-1 モード端子の設定

MD2	MD1	MD0	モード名	リセットベクタ アクセス領域	外部データ バス幅	備考
0	0	0	設定禁止			
0	0	1				
0	1	0				
0	1	1	内部ベクタモード	内部メモリ	モードデータ レジスタで設定	リセットシーケンス 以降はモードデータ で制御
1	0	0	設定禁止			
1	0	1				
1	1	0	フラッシュメモリシ リアル書込みモード*	-	-	-
1	1	1	フラッシュメモリ モード	-	-	パラレルライタ 使用時のモード

MD2 ~ MD0: "0" =  $V_{SS}$ , "1" =  $V_{CC}$  としてください。

\*: フラッシュメモリのシリアル書込みは、モード端子を設定するだけでは実行できません。ほかの箇所も設定する必要があります。詳細は、「第 24 章 フラッシュシリアル書込み接続例」をご参照ください。

## 8.3 モードデータレジスタ

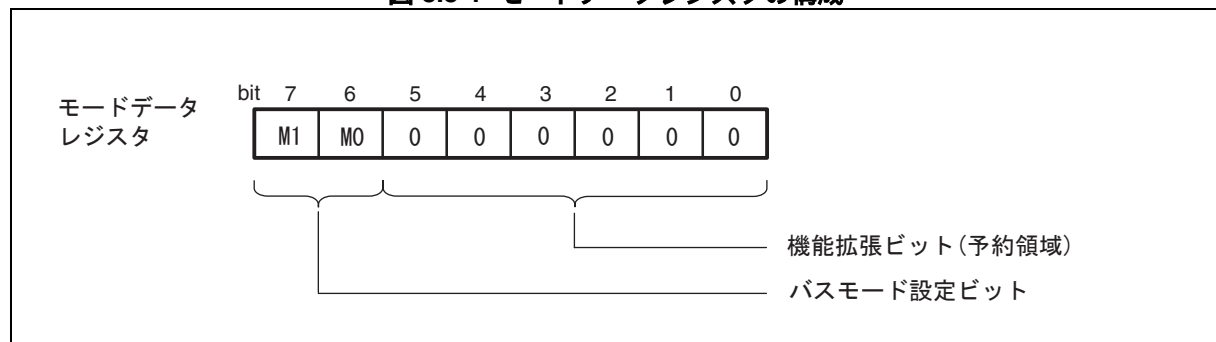
モードデータレジスタは, "FFFFDF<sub>H</sub>" 番地のメモリ上にあり, リセットシーケンス後のメモリアクセス動作を設定します。

### ■ モードデータレジスタ

リセットシーケンス実行中に, "FFFFDF<sub>H</sub>" 番地のモードデータをモードデータレジスタへ取込みます。モードデータレジスタのデータは, リセットシーケンス実行中に変更できます。命令では変更できません。モードデータの設定は, リセットシーケンス後に有効となります。

モードデータレジスタの構成を図 8.3-1 に示します。

図 8.3-1 モードデータレジスタの構成



### ■ バスモード設定ビット

リセットシーケンス終了後のバスモードを設定するビットです。

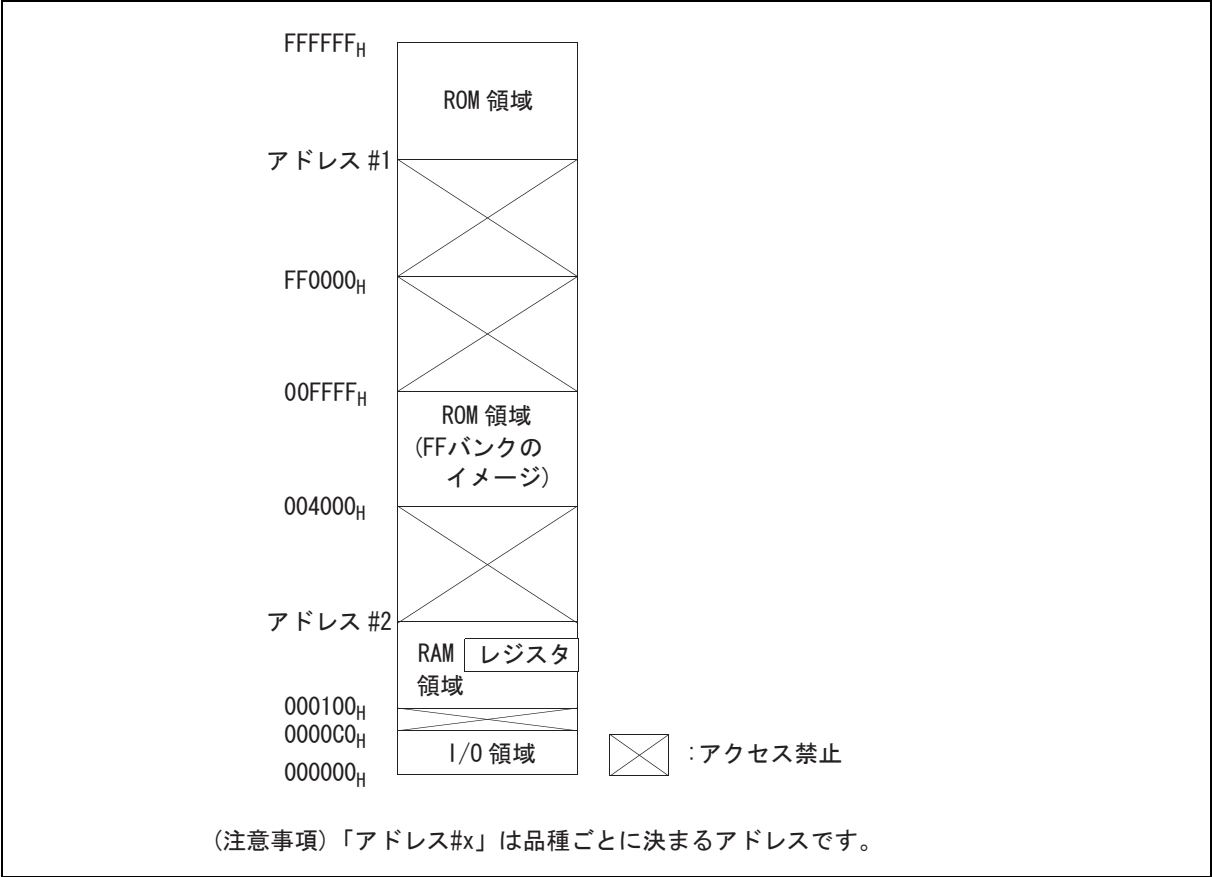
これらのビット機能を表 8.3-1 に示します。

表 8.3-1 バスモードのビット設定と機能

M1	M0	機能
0	0	シングルチップモード
0	1	(設定禁止)
1	0	
1	1	

MB90460/465 シリーズのアクセス領域と物理アドレスの関係を図 8.3-2 に示します。

図 8.3-2 MB90460/465 シリーズのアクセス領域と物理アドレスの関係



製品番号	アドレス #1	アドレス #2
MB90462/467	FF0000 <sub>H</sub>	000900 <sub>H</sub>
MB90F462	FF0000 <sub>H</sub>	000900 <sub>H</sub>
MB90F462A	FF0000 <sub>H</sub>	000900 <sub>H</sub>
MB90F463A	FE0000 <sub>H</sub>	000900 <sub>H</sub>
MB90V460*	FF0000 <sub>H</sub>	002100 <sub>H</sub>

\*:MB90V460 には , ROM は含まれていません。これらの領域は , 開発ツールから ROM デコード領域として使用するものと仮定します。

■ モード端子とモードデータの関係

モード端子とモードデータの関係を表 8.3-2 に示します。

表 8.3-2 モード端子とモードデータの関係

バスモード	モード端子の設定			モードデータレジスタの設定	
	MD2	MD1	MD0	M1	M0
シングルチップモード	0	1	1	0	0



# 第9章

---

## 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。

- 9.1 I/O ポートの概要
- 9.2 I/O ポートのレジスタ
- 9.3 ポート 0
- 9.4 ポート 1
- 9.5 ポート 2
- 9.6 ポート 3
- 9.7 ポート 4
- 9.8 ポート 5
- 9.9 ポート 6
- 9.10 I/O ポートのプログラム例



## 9.1 I/O ポートの概要

I/O ポートは、汎用入出力ポート（パラレル I/O ポート）として使用できます。MB90460/465 シリーズには、7 つのポート（51 本）があります。これらのポートは、リソース入出力端子としても使用します。

### ■ I/O ポートの機能

各 I/O ポートは、ポートデータレジスタ（PDR）のデータを端子へ出力したり、入力信号を端子から取込んだりします。各 I/O ポートは、ポート方向レジスタ（DDR）を使用して、ビット単位で入出力端子のデータ入出力方向を指定できます。各ポートの機能と、この機能を使用するリソースについて以下に説明します。

- ポート 0: 汎用入出力ポート / リソース（マルチパルスジェネレータ, PWC0）
- ポート 1: 汎用入出力ポート / リソース（外部割込み, マルチパルスジェネレータ, 多機能タイマ, 16 ビットリロードタイマ 0）
- ポート 2: 汎用入出力ポート / リソース（16 ビットリロードタイマ, PWC, インプットキャプチャ）
- ポート 3: 汎用入出力ポート / リソース（多機能タイマ, PPG1, PPG0）
- ポート 4: 汎用入出力ポート / リソース（UART0, マルチパルスジェネレータ, PPG2）
- ポート 5: 汎用入出力ポート / リソース（A/D）
- ポート 6: 汎用入出力ポート / リソース（UART1, 外部割込み）

各ポートの機能の概要を表 9.1-1 に示します。

表 9.1-1 各ポートの機能

ポート	端子	入力形式	出力形式	機能	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート 0	P00/OPT0 ~ P07/PW00	CMOS	CMOS プルアップ抵抗を設定可能	汎用入出力ポート	P07	P06	P05	P04	P03	P02	P01	P00
				リソース	PW00*	PW10*	OPT5*	OPT4*	OPT3*	OPT2*	OPT1*	OPT0*
ポート 1	P10/INT0/ DTTI0 ~ P17/FRCK	CMOS（ヒステリシス）		汎用入出力ポート	P17	P16	P15	P14	P13	P12	P11	P10
				リソース	FRCK	INT6/TO0	INT5/TIN0	INT4	INT3	INT2/DTTI1*	INT1	INT0/DTTI0
ポート 2	P20/TIN1 ~ P27/IN3	CMOS		汎用入出力ポート	P27	P26	P25	P24	P23	P22	P21	P20
				リソース	IN3	IN2	IN1	IN0	PW01	PW11	TO1	TIN1
ポート 3	P30/RTO0 ~ P37/PPG0	CMOS	CMOS	汎用入出力ポート	P37	P36	P35	P34	P33	P32	P31	P30
				リソース	PPG0	PPG1*	RTO5	RTO4	RTO3	RTO2	RTO1	RTO0
ポート 4	P40/SIN0 ~ P46/PPG2	CMOS（ヒステリシス）		汎用入出力ポート	-	P46	P45	P44	P43	P42	P41	P40
				リソース	-	PPG2	SNI2*	SNI1*	SNI0*	SCK0	SOT0	SIN0
ポート 5	P50/AN0 ~ P57/AN7	アナログ / CMOS		汎用入出力ポート	P57	P56	P55	P54	P53	P52	P51	P50
				アナログ入力	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
ポート 6	P60/SIN1 ~ P63/INT7	CMOS（ヒステリシス）		汎用入出力ポート	-	-	-	-	P63	P62	P61	P60
				リソース	-	-	-	-	INT7	SCK1	SOT1	SIN1

\*:MB90467 では、リソースが搭載されていないため、削除となります。

ポート5は、アナログ入力端子と兼用しています。ポート5を汎用ポートとして使用するためには、アナログデータ入力許可レジスタ（ADER）の対応ビットには必ず"0"を設定してください。リセットにより、ADERレジスタの各ビットには"1"が設定されます。

## 9.2 I/O ポートのレジスタ

I/O ポートの設定に関するレジスタの一覧を示します。

### ■ I/O ポート関連のレジスタ

各ポートに対応するレジスタの一覧を表 9.2-1 に示します。

表 9.2-1 レジスタと対応ポート

レジスタ	リード/ライト	アドレス	初期値
ポート 0 データレジスタ (PDR0)	R/W	000000 <sub>H</sub>	XXXXXXXX <sub>B</sub>
ポート 1 データレジスタ (PDR1)	R/W	000001 <sub>H</sub>	XXXXXXXX <sub>B</sub>
ポート 2 データレジスタ (PDR2)	R/W	000002 <sub>H</sub>	XXXXXXXX <sub>B</sub>
ポート 3 データレジスタ (PDR3)	R/W	000003 <sub>H</sub>	XXXXXXXX <sub>B</sub>
ポート 4 データレジスタ (PDR4)	R/W	000004 <sub>H</sub>	-XXXXXXXX <sub>B</sub>
ポート 5 データレジスタ (PDR5)	R/W	000005 <sub>H</sub>	XXXXXXXX <sub>B</sub>
ポート 6 データレジスタ (PDR6)	R/W	000006 <sub>H</sub>	----XXXX <sub>B</sub>
ポート 0 データ方向レジスタ (DDR0)	R/W	000010 <sub>H</sub>	00000000 <sub>B</sub>
ポート 1 データ方向レジスタ (DDR1)	R/W	000011 <sub>H</sub>	00000000 <sub>B</sub>
ポート 2 データ方向レジスタ (DDR2)	R/W	000012 <sub>H</sub>	00000000 <sub>B</sub>
ポート 3 データ方向レジスタ (DDR3)	R/W	000013 <sub>H</sub>	00000000 <sub>B</sub>
ポート 4 データ方向レジスタ (DDR4)	R/W	000014 <sub>H</sub>	-0000000 <sub>B</sub>
ポート 5 データ方向レジスタ (DDR5)	R/W	000015 <sub>H</sub>	00000000 <sub>B</sub>
ポート 6 データ方向レジスタ (DDR6)	R/W	000016 <sub>H</sub>	----0000 <sub>B</sub>
アナログデータ入力許可レジスタ (ADER)	R/W	000017 <sub>H</sub>	11111111 <sub>B</sub>
ポート 0 プルアップ抵抗設定レジスタ (RDR0)	R/W	00001C <sub>H</sub>	00000000 <sub>B</sub>
ポート 1 プルアップ抵抗設定レジスタ (RDR1)	R/W	00001D <sub>H</sub>	00000000 <sub>B</sub>

R/W : リード/ライト

X : 不定

- : 未使用

## 9.3 ポート 0

ポート 0 は汎用入出力ポートですが、リソース入出力と兼用しています。ポートの端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 0 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 0 の構成

ポート 0 は、以下から構成されています。

- ・ 汎用入出力ポート、マルチパルスジェネレータ出力、PWC 入出力( P00/OPT0 ~ P07/PWO0 )
- ・ ポート 0 データレジスタ ( PDR0 )
- ・ ポート 0 方向レジスタ ( DDR0 )
- ・ ポート 0 プルアップ抵抗設定レジスタ ( RDR0 )

### ■ ポート 0 の端子

ポート 0 は、リソースの入出力端子としても使用します。したがって、これらの端子は、リソース入出力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 0 の端子一覧を表 9.3-1 に示します。

表 9.3-1 ポート 0 の端子

ポート	端子	ポート機能		リソース機能		入出力形式		回路の種類
						入力	出力	
ポート 0	P00/OPT0*	P00	汎用入出力	OPT0*	マルチパルスジェネレータ出力	CMOS	CMOS	D
	P01/OPT1*	P01		OPT1*	マルチパルスジェネレータ出力			
	P02/OPT2*	P02		OPT2*	マルチパルスジェネレータ出力			
	P03/OPT3*	P03		OPT3*	マルチパルスジェネレータ出力			
	P04/OPT4*	P04		OPT4*	マルチパルスジェネレータ出力			
	P05/OPT5*	P05		OPT5*	マルチパルスジェネレータ出力			
	P06/PWI0*	P06		PWI0*	PWC0 入力			E
	P07/PWO0*	P07		PWO0*	PWC0 出力			

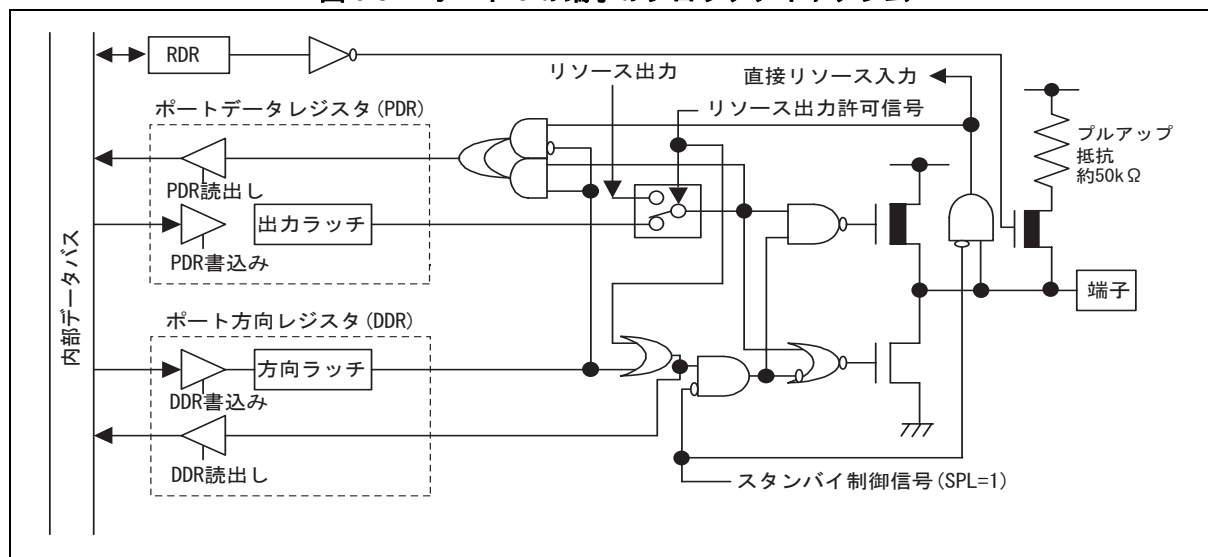
\*:MB90467 では、リソースが搭載されていないため、削除となります。

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

## ■ ポート0の端子のブロックダイアグラム

ポート0の端子のブロックダイアグラムを図9.3-1に示します。

図 9.3-1 ポート0の端子のブロックダイアグラム



リソース出力許可ビットを "許可" にすると、ポートは、DDR0 レジスタの値にかかわらずリソースの出力端子として機能します。

## ■ ポート0のレジスタ

ポート0のレジスタには、PDR0, DDR0, RDR0 があります。各レジスタを構成しているビットは、ポート0の端子に1対1で対応しています。ポート0の端子と、これらの端子に対応しているレジスタビットを表9.3-2に示します。

表 9.3-2 ポート0の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
ポート0	PDR0, DDR0, RDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	対応する端子	P07	P06	P05	P04	P03	P02	P01	P00

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

### 9.3.1 ポート 0 のレジスタ (PDR0, DDR0, RDR0)

---

ポート 0 のレジスタについて説明します。

---

#### ■ ポート 0 のレジスタの機能

- ポート 0 データレジスタ (PDR0)

PDR0 レジスタは、ポート 0 の各端子の状態を示します。

- ポート 0 方向レジスタ (DDR0)

DDR0 レジスタは、ポート 0 の各端子 (ビット) のデータ入出力方向を指定します。DDR0 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- ポート 0 プルアップ抵抗設定レジスタ (RDR0)

RDR0 レジスタは、ポート 0 の各端子 (ビット) に対するプルアップ抵抗を設定します。RDR0 レジスタのいずれかのビットが "1" の場合は、プルアップ抵抗がそのビットに対応するポート (端子) に対して有効になります。いずれかのビットが "0" の場合は、そのビットに対応するプルアップ抵抗は無効になります。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが "許可" に設定されていると、ポートはDDR0 レジスタの値とは無関係にリソースの出力端子として機能します。
- リソースの入力と兼用している端子をリソースの入力として使用する場合は、各入力端子に対応する DDR0 レジスタのビットに "0" を設定してポートを入力モードにしてください。

ポート 0 のレジスタの機能を表 9.3-3 に示します。

表 9.3-3 ポート 0 のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード/ ライト	アドレス	初期値
ポート 0 データ レジスタ (PDR0)	0	端子は low レベル	出力ラッチへ "0" がロード されます。端子が出力ポートと して機能すると、端子には low レベルが設定されます	R/W	000000 <sub>H</sub>	XXXXXXXX <sub>B</sub>
	1	端子は high レベル	出力ラッチへ "1" がロード されます。端子が出力ポートと して機能すると、端子には high レベルが設定されます			
ポート 0 方向 レジスタ (DDR0)	0	方向ラッチ は "0"	出力バッファが " オフ " にな り、ポートが入力モードにな ります	R/W	000010 <sub>H</sub>	00000000 <sub>B</sub>
	1	方向ラッチ は "1"	出力バッファが " オン " にな り、ポートが出力モードにな ります			
ポート 0 プルアップ 抵抗設定 レジスタ (RDR0)	0	設定ラッチ は "0"	入力モードにおいてプルアッ プ抵抗が遮断され、ポートは Hi-Z 状態になります	R/W	00001C <sub>H</sub>	00000000 <sub>B</sub>
	1	設定ラッチ は "1"	入力モードにおいてプルアッ プ抵抗が設定され、ポートは high レベルを保持します			

R/W : リード / ライト

X : 不定

## 9.3.2 ポート 0 の動作

---

ポート 0 の動作について説明します。

---

### ■ ポート 0 の動作

#### ● 出力モード時のポート動作

- DDR0 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
- 出力モード時において PDR0 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、そのままの端子へ出力されます。
- PDR0 レジスタを読み出すと PDR の出力ラッチの状態を読み出すことができます。

リードモディファイライト命令を PDR0 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR0 レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値は出力ラッチに書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR0 レジスタへ書き込み、次に DDR0 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR0 レジスタのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- ただし、RDR0 レジスタのいずれかのビットに "1" を設定してプルアップ抵抗を有効にすると、端子は high レベルを保持します。
- 入力モード時において PDR0 レジスタに書き込まれたデータは、PDR の出力ラッチが保持しますが、端子には出力されません。
- PDR0 レジスタを読み出すと端子のレベル ("0" または "1") を読み出すことができます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートはリソース出力として使用できます。"入力" と "出力" の切換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR0 レジスタのいずれかのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用されます。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リソース入力としてのポート動作

- ポートをリソース入力として使用する場合、端子の値がリソースの入力値となりますので、DDR0 レジスタに "0" を設定しポートを入力モードにしてください。



● リセット後のポート動作

- リセットにより、DDR0 レジスタは "0" に初期化されます。その結果、出力バッファが " オフ " になり（入出力モードは " 入力 " に変化する）、プルアップ抵抗が遮断され、端子はハイインピーダンス状態になります。
- PDR0 レジスタは、リセットでも初期化されません。したがって、ポートを出力モードで使用する場合は、出力データを PDR0 レジスタに設定した後、出力モードに DDR0 レジスタを設定しなければなりません。

● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット（SPL）がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが DDR0 レジスタの値とは無関係に " オフ " にさせられるからです。

また、プルアップ抵抗が " 有効 " のときは、SPL ビットに "1" を設定した場合、端子は high レベルを保持し、ハイインピーダンス状態にはならないので注意してください。ポート 0 の端子状態を表 9.3-4 に示します。

表 9.3-4 ポート 0 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード（SPL=0）	ストップモードまたはタイムベースタイマモード（SPL=1, RDR=0）	ストップモードまたはタイムベースタイマモード（SPL=1, RDR=1）
P00/OPT0 ～ P07/ PWO0	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力は遮断され、出力は Hi-Z 状態になる	入力は遮断され、H レベルに保持される

SPL: 低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット

Hi-Z: ハイインピーダンス

## 9.4 ポート 1

ポート 1 は汎用入出力ポートですが、リソース入出力と兼用しています。ポートの端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 1 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 1 の構成

ポート 1 は、以下から構成されています。

- ・ 汎用入出力ポート、外部割込み入力、16 ビットリロードタイマ入出力、フリーランタイマ外部クロック入力 (P10/INT0/DTT0 ~ P17/FRCK)
- ・ ポート 1 データレジスタ (PDR1)
- ・ ポート 1 方向レジスタ (DDR1)
- ・ ポート 1 プルアップ抵抗設定レジスタ (RDR1)

### ■ ポート 1 の端子

ポート 1 は、リソースの入力端子としても使用します。したがってこれらの端子は、リソースの入力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 1 の端子一覧を表 9.4-1 に示します。

表 9.4-1 ポート 1 の端子

ポート	端子	ポート機能		リソース機能		入出力形式		回路の種類
						入力	出力	
ポート 1	P10/INT0/ DTTI0	P10	汎用入出力	INT0/ DTTI0	外部割込み入力, 多機能 タイマ入力	CMOS (ヒステリシス)	CMOS	C
	P11/INT1	P11		INT1	外部割込み入力			
	P12/INT2/ DTTI1*	P12		INT2/ DTTI1*	外部割込み入力, マル チパルスジェネレータ 入力			
	P13/INT3	P13		INT3	外部割込み入力			
	P14/INT4	P14		INT4				
	P15/INT5/ TIN0	P15		INT5/ TIN0	外部割込み入力, リロー ドタイマクロック入力			
	P16/INT6/ TO0	P16		INT6/ TO0	外部割込み入力, リロー ドタイマイベント出力			
	P17/FRCK	P17		FRCK	フリーランタイマクロッ ク入力			

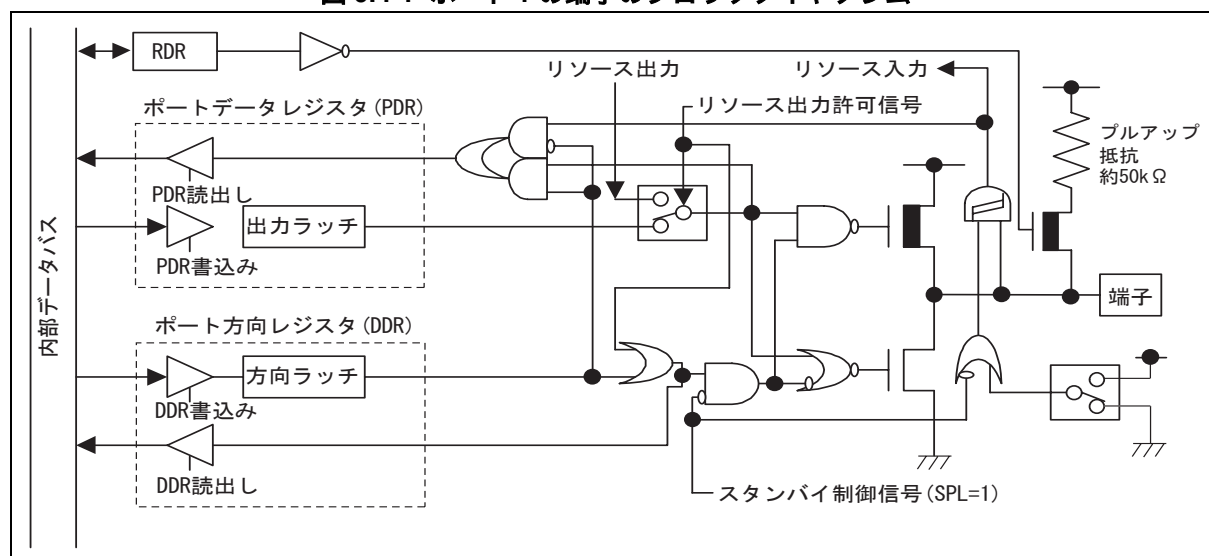
\*:MB90467 では、リソースが搭載されていないため、削除となります。

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

## ■ ポート 1 の端子のブロックダイアグラム

ポート 1 の端子のブロックダイアグラムを図 9.4-1 に示します。

図 9.4-1 ポート 1 の端子のブロックダイアグラム



リソース出力許可ビットを "許可" にすると、ポートは、DDR1 レジスタの値にかかわらずリソースの出力端子として機能します。

## ■ ポート 1 のレジスタ

ポート 1 のレジスタには、PDR1, DDR1, RDR1 があります。各レジスタを構成しているビットは、ポート 1 の端子に 1 対 1 で対応しています。ポート 1 の端子と、これらの端子に対応しているレジスタビットを表 9.4-2 に示します。

表 9.4-2 ポート 1 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
ポート 1	PDR1, DDR1, RDR1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	対応する端子	P17	P16	P15	P14	P13	P12	P11	P10

## 9.4.1 ポート 1 のレジスタ (PDR1, DDR1, RDR1)

---

ポート 1 のレジスタについて説明します。

---

### ■ ポート 1 のレジスタの機能

- ポート 1 データレジスタ (PDR1)

PDR1 レジスタは、ポート 1 の各端子の状態を示します。

- ポート 1 方向レジスタ (DDR1)

DDR1 レジスタは、ポート 1 の各端子 (ビット) のデータ入出力方向を指定します。DDR1 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- ポート 1 プルアップ抵抗設定レジスタ (RDR1)

RDR1 レジスタは、ポート 1 の各端子 (ビット) に対するプルアップ抵抗を設定します。RDR1 レジスタのいずれかのビットが "1" の場合は、プルアップ抵抗はそのビットに対応するポート (端子) に対して有効になります。いずれかのビットが "0" の場合は、そのビットに対応するプルアップ抵抗は無効になります。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが "許可" に設定されていると、ポートはDDR1 レジスタの値とは無関係にリソースの出力端子として機能します。
- リソースの入力と兼用している端子をリソースの入力として使用する場合は、各入力端子に対応する DDR1 レジスタのビットに "0" を設定してポートを入力モードにしてください。

ポート 1 のレジスタの機能を表 9.4-3 に示します。

表 9.4-3 ポート 1 のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード/ ライト	アドレス	初期値
ポート 1 データ レジスタ (PDR1)	0	端子は low レベル	出力ラッチへ "0" がロード されます。端子が出力ポートと して機能すると、端子には low レベルが設定されます	R/W	000001 <sub>H</sub>	XXXXXXXX <sub>B</sub>
	1	端子は high レベル	出力ラッチへ "1" がロード されます。端子が出力ポートと して機能すると、端子には high レベルが設定されます			
ポート 1 方向 レジスタ (DDR1)	0	方向ラッチ は "0"	出力バッファが " オフ " にな り、ポートが入力モードにな ります	R/W	000011 <sub>H</sub>	00000000 <sub>B</sub>
	1	方向ラッチ は "1"	出力バッファが " オン " にな り、ポートが出力モードにな ります			
ポート 1 プルアップ 抵抗設定 レジスタ (RDR1)	0	設定ラッチ は "0"	入力モードにおいてプルアッ プ抵抗が遮断され、ポートは Hi-Z 状態になります	R/W	00001D <sub>H</sub>	00000000 <sub>B</sub>
	1	設定ラッチ は "1"	入力モードにおいてプルアッ プ抵抗が設定され、ポートは high レベルを保持します			

R/W : リード / ライト

X : 不定

## 9.4.2 ポート 1 の動作

---

ポート 1 の動作について説明します。

---

### ■ ポート 1 の動作

#### ● 出力モード時のポート動作

- DDR1 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
- 出力モード時において PDR1 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、そのまま端子へ出力されます。
- PDR1 レジスタを読み出すと PDR の出力ラッチの状態を読み出すことができます。

リードモディファイライト命令を PDR1 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値は出力ラッチへ書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR1 レジスタに書き込み、次に DDR1 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR1 レジスタのいずれかのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- ただし、RDR1 レジスタのいずれかのビットに "1" を設定してプルアップ抵抗を "有効" にすると、端子は high レベルを保持します。
- 入力モード時において PDR1 レジスタに書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR1 レジスタを読み出すとポートの端子レベル ("0" または "1") を読み出すことができます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートはリソース出力として使用できます。"入力" と "出力" の切換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR1 レジスタのいずれかのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用されます。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リソース入力としてのポート動作

- ポートをリソース入力として使用する場合、端子の値がリソースの入力値となりますので、DDR1 レジスタに "0" を設定しポートを入力モードにしてください。

● リセット後のポート動作

- リセットにより、DDR1 レジスタは "0" に初期化されます。その結果、出力バッファが " オフ " になり（入出力モードは " 入力 " に変化する）、プルアップ抵抗が遮断され、端子はハイインピーダンス状態になります。
- PDR1 レジスタは、リセットでも初期化されません。したがって、ポートを出力モード時に使用する場合は、出力データを PDR1 レジスタに設定した後、出力モードに DDR1 レジスタを設定しなければなりません。

● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット（SPL）がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが DDR1 レジスタの値とは無関係に " オフ " にさせられるからです。

また、プルアップ抵抗が " 有効 " のときは、たとえ SPL ビットに "1" を設定した場合、端子は high レベルを保持し、ハイインピーダンス状態にはならないので注意してください。ポート 1 の端子状態を表 9.4-4 に示します。

表 9.4-4 ポート 1 の端子状態

端子	通常動作	スリープモード	ストップモード またはタイムベース タイマモード (SPL=0)	ストップモード またはタイムベース タイマモード (SPL=1, RDR=0)	ストップモード またはタイムベース タイマモード (SPL=1, RDR=1)
P10/ INT0/ DTT10 ~ P17/ FRCK	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力は有効になり*、 出力は Hi-Z 状態になります	入力は遮断され、H レベルを保持します

SPL : 低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット

Hi-Z : ハイインピーダンス

\* : 入力は、P10/INT0 ~ P16/INT6 を外部割込み端子として構成した場合のみ有効になり、それ以外の場合は遮断されます。

## 9.5 ポート 2

ポート 2 は汎用入出力ポートですが、リソース入出力と兼用しています。ポートの端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 2 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 2 の構成

ポート 2 は、以下から構成されています。

- 汎用入出力ポート、16 ビットリロードタイマ入出力、PWC 入出力、インプットキャプチャ入力（P20/TIN1 ~ P27/IN3）
- ポート 2 データレジスタ（PDR2）
- ポート 2 方向レジスタ（DDR2）

### ■ ポート 2 の端子

ポート 2 は、リソースの入出力端子としても使用します。したがってこれらの端子は、リソースの入出力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 2 の端子一覧を表 9.5-1 に示します。

表 9.5-1 ポート 2 の端子

ポート	端子	ポート機能	リソース機能	入出力形式		回路の種類
				入力	出力	
ポート 2	P20/TIN1	P20	TIN1	16 ビットリロードタイマ 1 イベント入力		CMOS (ヒステリシス)
	P21/TO1	P21	TO1	16 ビットリロードタイマ 1 タイマ出力		
	P22/PWI1	P22	PWI1	PWC1 入力		
	P23/PWO1	P23	PWO1	PWC1 出力		
	P24/IN0	P24	IN0	インプットキャプチャ チャンネル 0 入力		
	P25/IN1	P25	IN1	インプットキャプチャ チャンネル 1 入力		
	P26/IN2	P26	IN2	インプットキャプチャ チャンネル 2 入力		
	P27/IN3	P27	IN3	インプットキャプチャ チャンネル 3 入力		

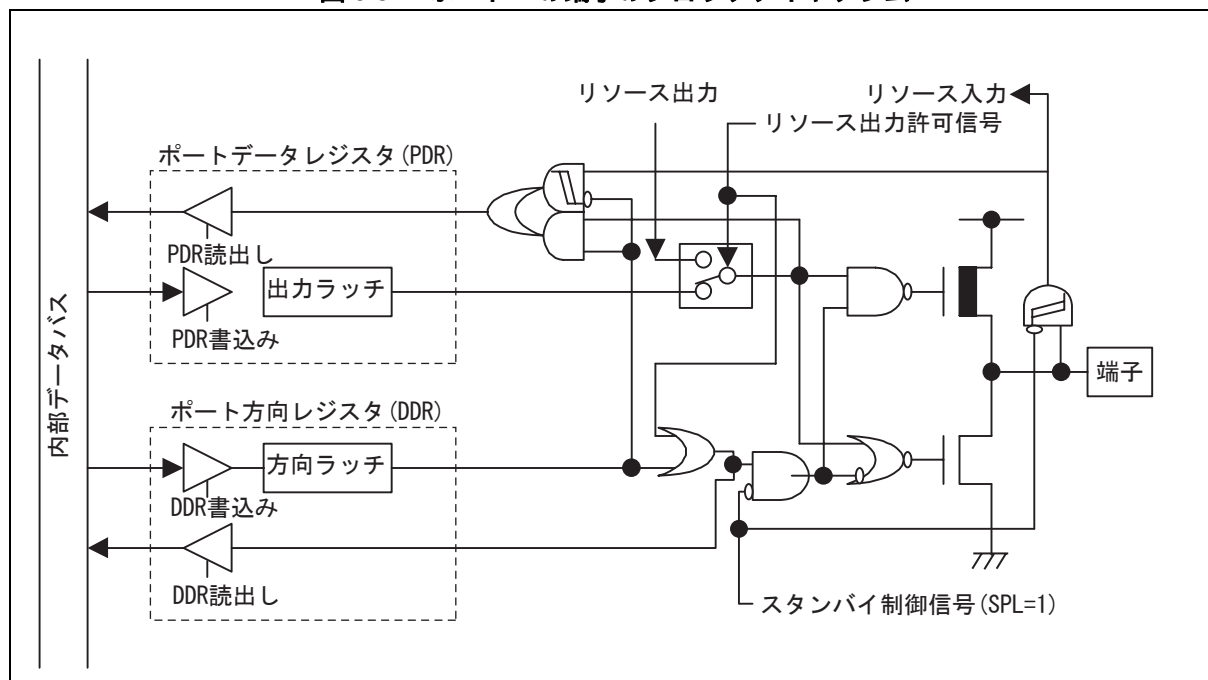
回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。



## ■ ポート 2 の端子のブロックダイアグラム

ポート 2 の端子のブロックダイアグラムを図 9.5-1 に示します。

図 9.5-1 ポート 2 の端子のブロックダイアグラム



リソース出力許可ビットを "許可" にすると、ポートは、DDR2 レジスタの値にかかわらずリソースの出力端子として機能します。

## ■ ポート 2 のレジスタ

ポート 2 のレジスタには、PDR2 と DDR2 があります。各レジスタを構成しているビットは、ポート 2 の端子に 1 対 1 で対応しています。ポート 2 の端子と、これらの端子に対応しているレジスタビットを表 9.5-2 に示します。

表 9.5-2 ポート 2 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
	PDR2, DDR2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート 2	対応する端子	P27	P26	P25	P24	P23	P22	P21	P20

## 9.5.1 ポート 2 のレジスタ (PDR2, DDR2)

ポート 2 のレジスタについて説明します。

### ■ ポート 2 のレジスタの機能

#### ● ポート 2 データレジスタ (PDR2)

PDR2 レジスタは、ポート 2 の各端子の状態を示します。

#### ● ポート 2 方向レジスタ (DDR2)

DDR2 レジスタは、ポート 2 の各端子 (ビット) のデータ入出力方向を指定します。DDR2 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが "許可" に設定されていると、ポートはDDR2 レジスタの値とは無関係にリソースの出力端子として機能します。
- リソースの入力と兼用している端子をリソースの入力として使用する場合は、各入力端子に対応するDDR2 レジスタのビットに "0" を設定してポートを入力モードにしてください。

ポート 2 のレジスタの機能を表 9.5-3 に示します。

表 9.5-3 ポート 2 のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード / ライト	アドレス	初期値
ポート 2 データレジスタ (PDR2)	0	端子は low レベル	出力ラッチへ "0" がロードされます。端子が出力ポートとして機能すると、端子には low レベルが設定されます	R/W	000002 <sub>H</sub>	XXXXXXXX <sub>B</sub>
	1	端子は high レベル	出力ラッチへ "1" がロードされます。端子が出力ポートとして機能すると、端子には high レベルが設定されます			
ポート 2 方向レジスタ (DDR2)	0	方向ラッチは "0"	出力バッファが "オフ" になり、ポートが入力モードになります	R/W	000012 <sub>H</sub>	00000000 <sub>B</sub>
	1	方向ラッチは "1"	出力バッファが "オン" になり、ポートが出力モードになります			

R/W: リード / ライト

X : 不定

## 9.5.2 ポート 2 の動作

---

ポート 2 の動作について説明します。

---

### ■ ポート 2 の動作

#### ● 出力モード時のポート動作

- DDR2 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
  - 出力モード時において PDR2 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、端子へ出力されます。
  - PDR2 レジスタを読み出すと PDR の出力ラッチの状態を読み出すことができます。
- リードモディファイライト命令を PDR2 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR2 レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値は出力ラッチに書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR2 レジスタに書き込み、次に DDR2 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR2 レジスタのいずれかのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- 入力モード時において PDR2 レジスタに書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR2 レジスタを読み出すと端子のレベル ("0" または "1") を読み出すことができます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートはリソース出力として使用できます。"入力" と "出力" の切り換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR0 レジスタのいずれかのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用されます。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リソース入力としてのポート動作

- ポートをリソース入力として使用する場合、端子の値がリソースの入力値となりますので、DDR2 レジスタに "0" を設定しポートを入力モードにしてください。

● リセット後のポート動作

- リセットにより, DDR2 レジスタは "0" に初期化されます。その結果, 出力バッファが " オフ " になり ( 入出力モードは " 入力 " に変化する ), 端子はハイインピーダンス状態になります。
- PDR2 レジスタは, リセットでも初期化されません。したがって, ポートを出力モードで使用する場合は, 出力データを PDR2 レジスタに設定した後, 出力モードに DDR2 レジスタを設定しなければなりません。

● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット (SPL) がすでに "1" の場合は, 端子はハイインピーダンス状態になります。これは, 出力バッファが DDR2 レジスタの値とは無関係に " オフ " にさせられるからです。ポート 2 の端子状態を表 9.5-4 に示します。

表 9.5-4 ポート 2 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード (SPL=0)	ストップモードまたはタイムベースタイマモード (SPL=1)
P20/TIN1 ~ P27/IN3	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力遮断され, 出力は Hi-Z 状態になります

SPL: 低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット  
Hi-Z: ハイインピーダンス

## 9.6 ポート 3

ポート 3 は汎用入出力ポートですが、リソース出力と兼用しています。ポート端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 3 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 3 の構成

ポート 3 は、以下から構成されています。

- 汎用入出力ポート、多機能タイマ出力、PPG 出力（P30/RTO0 ~ P37/PPG0）
- ポート 3 データレジスタ（PDR3）
- ポート 3 方向レジスタ（DDR3）

### ■ ポート 3 の端子

ポート 3 は、リソースの出力端子としても使用します。したがってこれらの端子は、リソースの入出力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 3 の端子一覧を表 9.6-1 に示します。

表 9.6-1 ポート 3 の端子

ポート	端子	ポート機能	リソース機能	入出力形式		回路の種類
				入力	出力	
ポート 3	P30/RTO0	P30	RTO0	CMOS	CMOS	G
	P31/RTO1	P31	RTO1			
	P32/RTO2	P32	RTO2			
	P33/RTO3	P33	RTO3			
	P34/RTO4	P34	RTO4			
	P35/RTO5	P35	RTO5			
	P36/PPG1*	P36	PPG1*			H
	P37/PPG0	P37	PPG0			

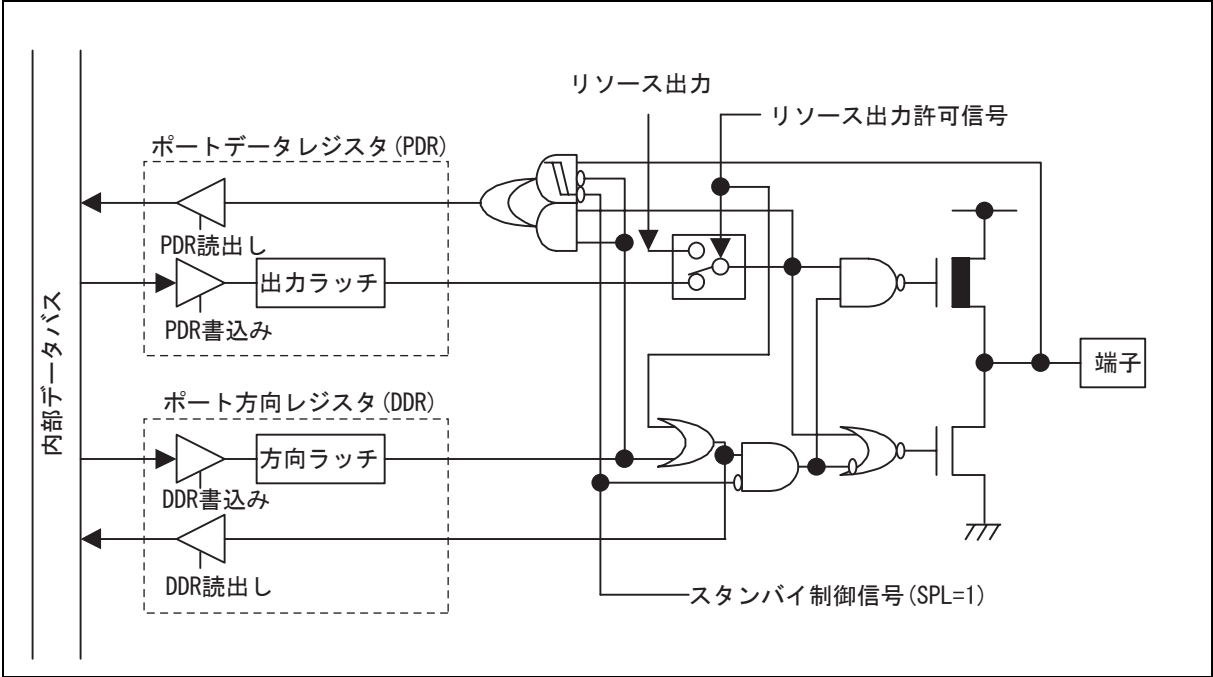
\*:MB90467 では、リソースが搭載されていないため、削除となります。

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

■ ポート 3 の端子のブロックダイヤグラム

ポート 3 の端子のブロックダイヤグラムを図 9.6-1 に示します。

図 9.6-1 ポート 3 の端子のブロックダイヤグラム



リソース出力許可ビットを " 許可 " にすると、ポートは、DDR3 レジスタの値にかかわらずリソースの出力端子として機能します。

■ ポート 3 のレジスタ

ポート 3 のレジスタには、PDR3 と DDR3 があります。各レジスタを構成しているビットは、ポート 3 の端子に 1 対 1 で対応しています。ポート 3 の端子と、これらの端子に対応しているレジスタビットを表 9.6-2 に示します。

表 9.6-2 ポート 3 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
	PDR3, DDR3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート 3	対応する端子	P37	P36	P35	P34	P33	P32	P31	P30

## 9.6.1 ポート 3 のレジスタ (PDR3, DDR3)

ポート 3 のレジスタについて説明します。

### ■ ポート 3 のレジスタの機能

#### ● ポート 3 データレジスタ (PDR3)

PDR3 レジスタは、ポート 3 の各端子の状態を示します。

#### ● ポート 3 方向レジスタ (DDR3)

DDR3 レジスタは、ポート 3 の各端子 (ビット) のデータ入出力方向を指定します。DDR3 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが "許可" に設定されていると、ポートはDDR3 レジスタの値とは無関係にリソースの出力端子として機能します。

ポート 3 のレジスタの機能を表 9.6-3 に示します。

表 9.6-3 ポートの 3 レジスタの機能

レジスタ	データ	読出し時	書込み時	リード / ライト	アドレス	初期値
ポート 3 データ レジスタ (PDR3)	0	端子は low レベル	出力ラッチへ "0" がロードされます。端子が出力ポートとして機能すると、端子には low レベルが設定されます	R/W	000003 <sub>H</sub>	XXXXXXXX <sub>B</sub>
	1	端子は high レベル	出力ラッチへ "1" がロードされます。端子が出力ポートとして機能すると、端子には high レベルが設定されます			
ポート 3 方向 レジスタ (DDR3)	0	方向ラッチ は "0"	出力バッファが "オフ" になり、ポートが入力モードになります	R/W	000013 <sub>H</sub>	00000000 <sub>B</sub>
	1	方向ラッチ は "1"	出力バッファが "オン" になり、ポートが出力モードになります			

R/W : リード / ライト

X : 不定

## 9.6.2 ポート 3 の動作

---

ポート 3 の動作について説明します。

---

### ■ ポート 3 の動作

#### ● 出力モード時のポート動作

- DDR3 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
  - 出力モード時において PDR3 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、端子へ出力されます。
  - PDR3 レジスタを読み出すと PDR の出力ラッチの状態を読み出すことができます。
- リードモディファイライト命令を PDR3 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR3 レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値は出力ラッチに書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR3 レジスタへ書き込み、次に DDR3 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR3 レジスタのいずれかのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- 入力モード時において PDR3 レジスタに書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR3 レジスタを読み出すと端子のレベル ("0" または "1") を読み出すことができます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートをリソース出力として使用できます。"入力" と "出力" の切り換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR3 レジスタのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用されます。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リセット後のポート動作

- リセットにより、DDR3 レジスタは "0" に初期化されます。その結果、出力バッファが "オフ" になり（入出力モードが "入力" に変化する）、端子がハイインピーダンス状態になります。
- PDR3 レジスタは、リセットでも初期化されません。したがって、ポートを出力モードで使用する場合は、出力データを PDR3 レジスタに設定した後、出力モードに DDR3 レジスタを設定しなければなりません。



● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット (SPL) がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが DDR3 レジスタの値とは無関係に "オフ" にさせられるからです。ポート 3 の端子状態を表 9.6-4 に示します。

表 9.6-4 ポート 3 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード (SPL=0)	ストップモードまたはタイムベースタイマモード (SPL=1)
P30/RT00 ~ P37/PPG0	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力は遮断され、出力は Hi-Z 状態になります

SPL: 低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット

Hi-Z: ハイインピーダンス

## 9.7 ポート 4

ポート 4 は汎用入出力ポートですが、リソース入出力と兼用しています。ポートの端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 4 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 4 の構成

ポート 4 は、以下から構成されています。

- 汎用入出力ポート、UART 入出力、マルチパルスジェネレータ入力、PPG 出力 (P40/SIN0 ~ P46/PPG2)
- ポート 4 データレジスタ (PDR4)
- ポート 4 方向レジスタ (DDR4)

### ■ ポート 4 の端子

ポート 4 は、リソースの入出力端子としても使用します。したがってこれらの端子は、リソースの入出力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 4 の端子一覧を表 9.7-1 に示します。

表 9.7-1 ポート 4 の端子

ポート	端子	ポート機能	リソース機能	入出力形式		回路の種類
				入力	出力	
ポート 4	P40/SIN0	P40	SIN0	UART0 データ入力	CMOS (ヒステリシス)	F
	P41/SOT0	P41	SOT0	UART0 データ出力		
	P42/SCK0	P42	SCK0	UART0 シリアルクロック I/O		
	P43/SNI0*	P43	SNI0*	マルチパルスジェネレータ入力 0		
	P44/SNI1*	P44	SNI1*	マルチパルスジェネレータ入力 1		
	P45/SNI2*	P45	SNI2*	マルチパルスジェネレータ入力		
	P46/PPG2	P46	PPG2	PPG2 出力		

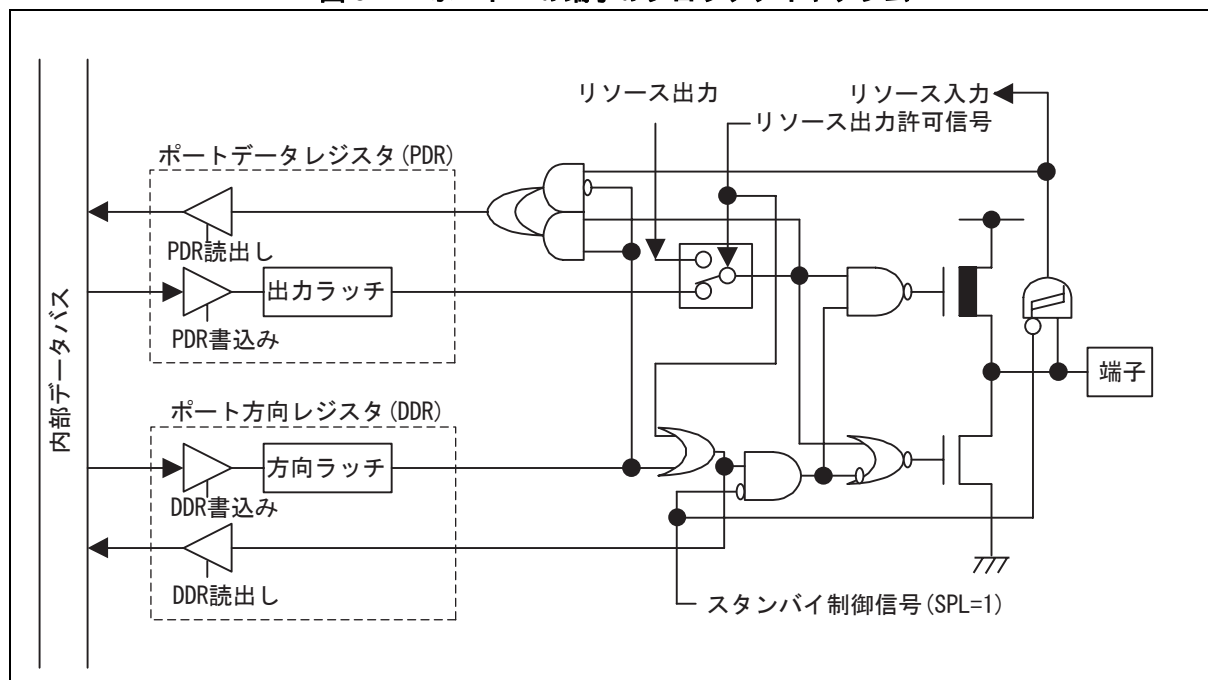
\*:MB90467 では、リソースが搭載されていないため、削除となります。

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

## ■ ポート 4 の端子のブロックダイアグラム

ポート 4 の端子のブロックダイアグラムを図 9.7-1 に示します。

図 9.7-1 ポート 4 の端子のブロックダイアグラム



リソース出力許可ビットを " 許可 " にすると、ポートは、DDR4 レジスタの値にかかわらずリソースの出力端子として機能します。

## ■ ポート 4 のレジスタ

ポート 4 のレジスタには、PDR4 と DDR4 があります。各レジスタを構成しているビットは、ポート 4 の端子に 1 対 1 で対応しています。ポート 4 の端子と、これらの端子に対応しているレジスタビットを表 9.7-2 に示します。

表 9.7-2 ポート 4 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
ポート 4	PDR4, DDR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	対応する端子	-	P46	P45	P44	P43	P42	P41	P40

## 9.7.1 ポート4のレジスタ (PDR4, DDR4)

ポート4のレジスタについて説明します。

### ■ ポート4のレジスタの機能

#### ● ポート4 データレジスタ (PDR4)

PDR4 レジスタは、ポート4の各端子の状態を示します。

#### ● ポート4 方向レジスタ (DDR4)

DDR4 レジスタは、ポート4の各端子(ビット)のデータ入出力方向を指定します。DDR4 レジスタのいずれかのビットが"1"の場合は、そのビットに対応するポート(端子)は出力ポートとして設定されます。いずれかのビットが"0"の場合は、対応するポート(端子)は入力ポートとして設定されます。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが"許可"に設定されていると、ポートはDDR4 レジスタの値とは無関係にリソースの出力端子として機能します。
- リソースの入力と兼用している端子をリソースの入力として使用する場合は、各入力端子に対応するDDR4 レジスタビットに"0"を設定してポートを入力モードにしてください。

ポート4のレジスタの機能を表9.7-3に示します。

表 9.7-3 ポート4のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード/ ライト	アドレス	初期値
ポート4 データ レジスタ (PDR4)	0	端子は low レベル	DDR5 ビットに "0" を設定すると、端子はハイインピーダンス状態になります。DDR5 ビットに "1" を設定すると、端子は出力ポートとして機能し、low レベルが設定されます	R/W	000004 <sub>H</sub>	-XXXXXXX <sub>B</sub>
	1	端子は high レベル	DDR5 ビットに "0" を設定すると、端子はハイインピーダンス状態になります。DDR5 ビットに "1" を設定すると、端子は出力ポートとして機能し、high レベルが設定されます			
ポート4 方向 レジスタ (DDR4)	0	方向ラッチ は "0"	出力バッファが "オフ" になり、ポートが入力モードになります	R/W	000014 <sub>H</sub>	-0000000 <sub>B</sub>
	1	方向ラッチ は "1"	出力バッファが "オン" になり、ポートが出力モードになります			

R/W : リード / ライト

X : 不定

- : 未定義

## 9.7.2 ポート 4 の動作

---

ポート 4 の動作について説明します。

---

### ■ ポート 4 の動作

#### ● 出力モード時のポート動作

- DDR4 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
- 出力モード時において PDR4 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、端子へ出力されます。
- PDR4 レジスタを読み出すと PDR の出力ラッチ状態を読み出すことができます。

リードモディファイライト命令を PDR4 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR4 レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値は出力ラッチへ書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR4 レジスタへ書き込み、次に DDR4 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR4 レジスタのいずれかのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- 入力モード時において PDR4 レジスタに書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR4 レジスタを読み出すと端子のレベル ("0" または "1") が読み出されます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートはリソース出力として使用できます。"入力" と "出力" の切換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR4 レジスタのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用されます。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リソース入力としてのポート動作

- ポートをリソース入力として使用する場合、端子の値がリソースの入力の値となりますので、DDR4 レジスタに "0" を設定することでポートを入力モードにしてください。

● リセット後のポート動作

- リセットにより、DDR4 レジスタは "0" に初期化されます。その結果、出力バッファが " オフ " になり（入出力モードが " 入力 " に変化する）、端子がハイインピーダンス状態になります。
- PDR4 レジスタは、リセットでも初期化されません。したがって、ポートを出力モードで使用する場合は、出力データを PDR4 レジスタに設定した後、出力モードに DDR4 レジスタを設定しなければなりません。

● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力のモード制御レジスタ（LPMCR）の端子状態指定ビット（SPL）がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが DDR4 レジスタの値とは無関係に " オフ " にさせられるからです。ポート 4 の端子状態を表 9.7-4 に示します。

表 9.7-4 ポート 4 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード（SPL=0）	ストップモードまたはタイムベースタイマモード（SPL=1）
P40/SIN0 ~ P46/PPG2	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力は遮断され、出力は Hi-Z 状態になります

SPL: 低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット  
Hi-Z: ハイインピーダンス

## 9.8 ポート 5

ポート 5 は汎用入出力ポートですが、A/D コンバータアナログ入力と兼用しています。ポートの端子は、ビット単位でポート機能とアナログ入力機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 5 の構成、端子一覧、端子のブロックダイヤグラムおよび対応するレジスタについて説明します。

### ■ ポート 5 の構成

ポート 5 は、以下から構成されています。

- 汎用入出力ポート、アナログ入力端子（P50/AN0 ~ P57/AN7）
- ポート 5 データレジスタ（PDR5）
- ポート 5 方向レジスタ（DDR5）
- アナログ入力許可レジスタ（ADER）

### ■ ポート 5 の端子

ポート 5 は、アナログの入力端子としても使用します。したがってこれらの端子は、アナログ入力として使用した場合、汎用入出力ポートとしては使用できません。同様に、ポート 5 は、汎用入出力ポートとして使用した場合、アナログ入力としては使用できません。ポート 5 の端子一覧を表 9.8-1 に示します。

表 9.8-1 ポート 5 の端子

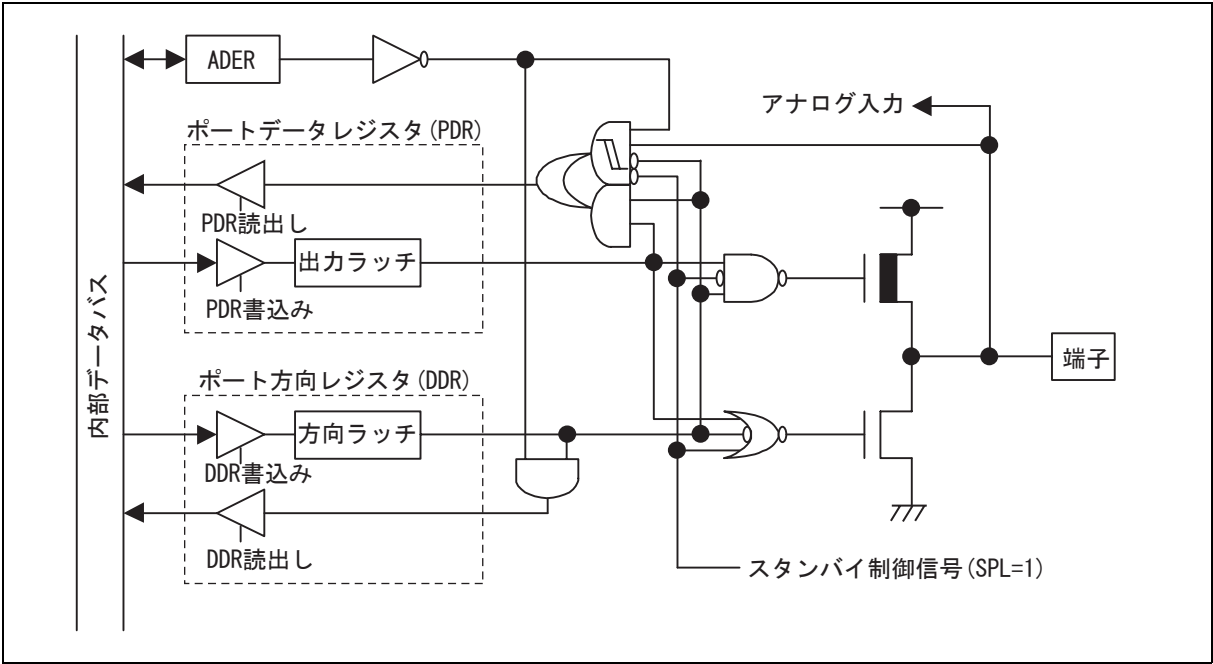
ポート	端子	ポート機能		リソース機能		入出力形式		回路の種類
						入力	出力	
ポート 5	P50/AN0	P50	汎用入出力	AN0	アナログ入力 0	アナログ / CMOS	CMOS	I
	P51/AN1	P51		AN1	アナログ入力 1			
	P52/AN2	P52		AN2	アナログ入力 2			
	P53/AN3	P53		AN3	アナログ入力 3			
	P54/AN4	P54		AN4	アナログ入力 4			
	P55/AN5	P55		AN5	アナログ入力 5			
	P56/AN6	P56		AN6	アナログ入力 6			
	P57/AN7	P57		AN7	アナログ入力 7			

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

■ ポート 5 の端子のブロックダイヤグラム

ポート 5 の端子のブロックダイヤグラムを図 9.8-1 に示します。

図 9.8-1 ポート 5 の端子のブロックダイヤグラム



入力ポートとして使用される端子の場合は、その端子に対応する DDR5 レジスタの各ビットに "0" が設定され、またその端子に対応する ADER レジスタの各ビットにも "0" が設定されます。

アナログ入力端子として使用される端子の場合は、その端子に対応する DDR5 レジスタビットに "0" が設定され、その端子に対応する ADER レジスタビットには "1" が設定されます。この場合、PDR5 レジスタから読み出される値は "0" です。

■ ポート 5 のレジスタ

ポート 5 のレジスタには、PDR5, DDR5, ADER があります。各レジスタを構成しているビットは、ポート 5 の端子に 1 対 1 に対応しています。ポート 5 の端子と、これら端子に対応しているレジスタビットを表 9.8-2 に示します。

表 9.8-2 ポート 5 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
	PDR5, DDR5, ADER	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート 5	対応する端子	P57	P56	P55	P54	P53	P52	P51	P50



## 9.8.1 ポート 5 のレジスタ (PDR5, DDR5, ADER)

---

ポート 5 のレジスタについて説明します。

---

### ■ ポート 5 のレジスタの機能

- ポート 5 データレジスタ (PDR5)

PDR5 レジスタは、ポート 5 の各端子の状態を示します。

- ポート 5 方向レジスタ (DDR5)

DDR5 レジスタは、ポート 5 の各端子 (ビット) のデータ入出力方向を指定します。DDR5 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- アナログ入力許可レジスタ (ADER)

ADER レジスタの各ビットは、対応するポート 5 端子を "汎用入出力ポート" として使用するか "アナログ入力端子" として使用するかを指定します。ADER ビットに "1" を設定すると、対応する端子をアナログ入力として使用します。ADER ビットに "0" を設定すると、対応する端子を汎用入出力として使用します。

- ポートを入力モードに設定して中間レベルの信号が入力されると、入力リーク電流が発生します。したがって、端子をアナログ入力として使用する場合は、対応する ADER ビットには必ず "1" を設定してください。

---

< 参考 >      リセットにより、DDR5 レジスタは "0" にリセットされ、ADER レジスタには "1" (端子をアナログ入力として使用する) が設定されます。

---

ポート 5 のレジスタの機能を表 9.8-3 に示します。

表 9.8-3 ポート 5 のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード/ ライト	アドレス	初期値
ポート 5 データ レジスタ ( PDR5 )	0	端子は low レベル		R/W	000005 <sub>H</sub>	XXXXXXXX <sub>B</sub>
	1	端子は high レベル				
ポート 5 方向 レジスタ ( DDR5 )	0	方向ラッチ は "0"	出力バッファが " オフ " になり、 ポートが入力モードになります	R/W	000015 <sub>H</sub>	00000000 <sub>B</sub>
	1	方向ラッチ は "1"	出力バッファが " オン " になり、 ポートが出力モードになります			
アナログ 入力許可 レジスタ ( ADER )	0	ポート入出力モード		R/W	000017 <sub>H</sub>	11111111 <sub>B</sub>
	1	アナログ入力モード				

R/W : リード / ライト

X : 不定

## 9.8.2 ポート 5 の動作

---

ポート 5 の動作について説明します。

---

### ■ ポート 5 の動作

#### ● 出力モード時のポート動作

- DDR5 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
- 出力モード時において PDR5 レジスタに書き込まれたデータは、PDR の出力ラッチが保持し、端子へ出力されます。
- PDR5 レジスタを読み出すと PDR の出力ラッチ状態を読み出すことができます。リードモディファイライト命令を PDR5 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR5 レジスタにより "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値が出力ラッチへ書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR5 レジスタへ書き込み、次に DDR5 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR5 レジスタと ADER レジスタのいずれかのビットに "0" を設定すると、対応する端子が入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- 入力モード時において PDR5 レジスタへ書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR5 レジスタを読み出すと端子のレベル ("0" または "1") が読み出されます。

#### ● アナログ入力としてのポート動作

- ポート端子を "アナログ入力" として使用する場合は、その端子に対応する ADER ビットに "1" を設定してください。すると、端子は汎用ポート端子としては動作せず、アナログ入力端子として動作します。この状態で PDR5 レジスタを読み出すと "0" が読み出されます。

#### ● リセット後のポート動作

- リセットにより、DDR5 レジスタは "0" に、ADER レジスタは "1" に初期化され、ポートがアナログ入力モードになります。ポートを汎用入出力ポートとして使用する場合は、前もって ADER レジスタに "0" 設定し、ポートを入出力モードにしてください。

● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット (SPL) がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが "オフ" にさせられるからです。ポート 5 の端子状態を表 9.8-4 に示します。

表 9.8-4 ポート 5 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード (SPL=0)	ストップモードまたはタイムベースタイマモード (SPL=1)
P50/AN0 ~ P57/AN7	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力遮断され、出力は Hi-Z 状態になります

SPL: 低消費電力モード制御レジスタ (LPMCR) の端子状態指定ビット  
Hi-Z: ハイインピーダンス

## 9.9 ポート 6

ポート 6 は汎用入出力ポートですが、リソース入出力と兼用しています。ポートの端子は、ビット単位でポート機能とリソース機能を切り換えることができます。ここでは、汎用入出力ポート機能を中心に説明します。また、ポート 6 の構成、端子一覧、端子のブロックダイアグラムおよび対応するレジスタについて説明します。

### ■ ポート 6 の構成

ポート 6 は、以下から構成されています。

- 汎用入出力ポート、UART 入出力、外部割込み入力（P60/SIN1 ~ P63/INT7）
- ポート 6 データレジスタ（PDR6）
- ポート 6 方向レジスタ（DDR6）

### ■ ポート 6 の端子

ポート 6 は、リソースの入出力端子としても使用します。したがってこれらの端子は、リソースの入出力端子として使用した場合、汎用入出力ポートとしては使用できません。ポート 6 の端子一覧を表 9.9-1 に示します。

表 9.9-1 ポート 6 の端子

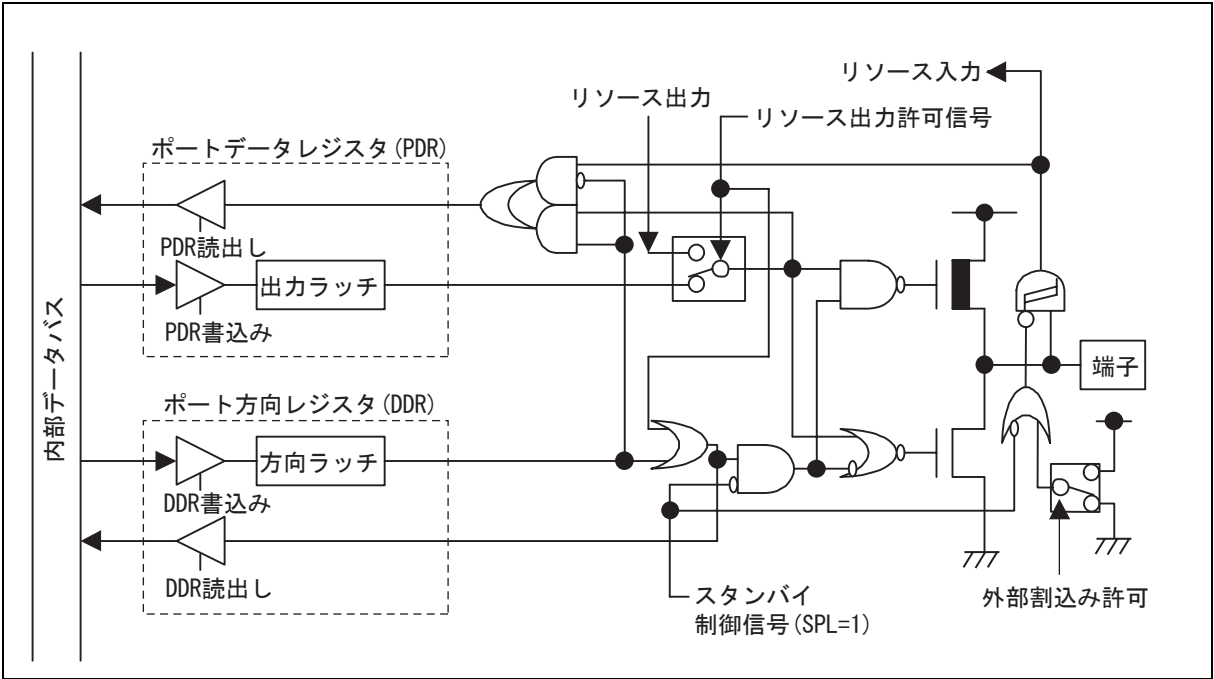
ポート	端子	ポート機能		リソース機能		入出力形式		回路の種類
						入力	出力	
ポート 6	P60/SIN1	P60	汎用 I/O	SIN1	UART1 データ入力	CMOS (ヒステリシス)	CMOS	F
	P61/SOT1	P61		SOT1	UART1 データ出力			
	P62/SCK1	P62		SCK1	UART1 シリアルクロック入出力			
	P63/INT7	P63		INT7	外部割込み入力			

回路の種類についての詳細は、「1.7 I/O 回路の種類」をご参照ください。

■ ポート 6 の端子のブロックダイヤグラム

ポート 6 の端子のブロックダイヤグラムを図 9.9-1 に示します。

図 9.9-1 ポート 6 の端子のブロックダイヤグラム



■ ポート 6 のレジスタ

ポート 6 のレジスタには、PDR6 と DDR6 があります。各レジスタを構成しているビットは、ポート 6 の端子に 1 対 1 で対応しています。ポート 6 の端子と、これらの端子に対応しているレジスタビットを表 9.9-2 に示します。

表 9.9-2 ポート 6 の端子と対応レジスタビット

ポート	レジスタビットと対応ポート端子								
ポート 6	PDR6, DDR6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	対応する端子	-	-	-	-	P63	P62	P61	P60

## 9.9.1 ポート 6 のレジスタ (PDR6, DDR6)

ポート 6 のレジスタについて説明します。

### ■ ポート 6 のレジスタの機能

#### ● ポート 6 データレジスタ (PDR6)

PDR6 レジスタは、ポート 6 の各端子の状態を示します。

#### ● ポート 6 方向レジスタ (DDR6)

DDR6 レジスタは、ポート 6 の各端子 (ビット) のデータ入出力方向を指定します。DDR6 レジスタのいずれかのビットが "1" の場合は、そのビットに対応するポート (端子) は出力ポートとして設定されます。いずれかのビットが "0" の場合は、対応するポート (端子) は入力ポートとして設定されます。

- リソースの出力と兼用している端子は、その端子に対応するリソース出力許可ビットが "許可" 設定されていると、ポートは DDR6 レジスタの値とは無関係にリソースの出力端子として機能します。
- リソースの入力と兼用している端子をリソースの入力として使用する場合は、各入力端子に対応する DDR6 レジスタビットに "0" を設定してポートを入力モードにします。

ポート 6 のレジスタの機能を表 9.9-3 に示します。

表 9.9-3 ポート 6 のレジスタの機能

レジスタ	データ	読出し時	書込み時	リード / ライト	アドレス	初期値
ポート 6 データレジスタ (PDR6)	0	端子は low レベル	出力ラッチへ "0" がロードされます。端子が出力ポートとして機能すると、端子には low レベルが設定されます	R/W	000006 <sub>H</sub>	----XXXX <sub>B</sub>
	1	端子は high レベル	出力ラッチへ "1" がロードされます。端子が出力ポートとして機能すると、端子には high レベルが設定されます			
ポート 6 方向レジスタ (DDR6)	0	方向ラッチは "0"	出力バッファが "オフ" になり、ポートが入力モードになります	R/W	000016 <sub>H</sub>	----0000 <sub>B</sub>
	1	方向ラッチは "1"	出力バッファが "オン" になり、ポートが出力モードになります			

R/W : リード / ライト

X : 不定

- : 未定義

## 9.9.2 ポート 6 の動作

---

ポート 6 の動作について説明します。

---

### ■ ポート 6 の動作

#### ● 出力モード時のポート動作

- DDR6 レジスタのいずれかのビットに "1" を設定すると、そのビットに対応するポートの端子は出力モードになります。
  - 出力モード時において PDR6 レジスタへ書き込まれたデータは、PDR の出力ラッチが保持し、端子へ出力されます。
  - PDR6 レジスタを読み出すと PDR の出力ラッチの状態を読み出すことができます。
- リードモディファイライト命令を PDR6 レジスタに対して使用すると、このレジスタのターゲットビットには書き込んだ値が設定されます。DDR6 レジスタによって "出力" として指定されているビットは影響を受けませんが、"入力" として指定されているビットの場合は、端子からの入力値が出力ラッチへ書き込まれ、そのまま出力されます。したがって、ビットのモードを "入力" から "出力" に切り換える前に、出力データを PDR6 レジスタへ書き込み、次に DDR6 レジスタを出力モードに設定してください。

#### ● 入力モード時のポート動作

- DDR6 レジスタのビットに "0" を設定すると、そのビットに対応する端子は入力モードになります。
- 入力モード時は、出力バッファは "オフ" になり、端子はハイインピーダンス状態になります。
- 入力モード時において PDR6 レジスタへ書き込まれたデータは、PDR の出力ラッチが保持しますが、端子へは出力されません。
- PDR6 レジスタは、読み出すと端子のレベル ("0" または "1") が読み出されます。

#### ● リソース出力としてのポート動作

- リソース出力許可ビットを "許可" にすると、ポートをリソース出力として使用できます。"入力" と "出力" の切換えを指定する場合、リソース許可ビットの状態が優先されます。すなわち、たとえ DDR6 レジスタのいずれかのビットが "0" であっても、リソース出力が "許可" になっている場合は、そのビットに対応する端子はリソース出力として使用します。しかしリソース出力が "許可" であっても端子の値を読み出すことはできるので、リソースの出力値を読み出すことができます。

#### ● リソース入力としてのポート動作

- ポートをリソース入力として使用する場合、端子の値がリソースの入力の値となりますので、DDR6 レジスタに "0" を設定しポートを入力モードにしてください。



## ● リセット後のポート動作

- リセットにより、DDR6 レジスタは "0" に初期化されます。その結果、出力バッファが " オフ " になり（入出力モードが " 入力 " に変化する）、端子がハイインピーダンス状態になります。
- PDR6 レジスタは、リセットでも初期化されません。ポートを出力モード時に使用する場合は、出力データを PDR6 レジスタへ設定した後、出力モードに DDR6 レジスタを設定しなければなりません。

## ● ストップモードまたはタイムベースタイマモード時のポート動作

ストップモードまたはタイムベースタイマモードに切り換えられたときに低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット（SPL）がすでに "1" の場合は、端子はハイインピーダンス状態になります。これは、出力バッファが DDR6 レジスタの値とは無関係に " オフ " にさせられるからです。ポート 6 の端子状態を表 9.9-4 に示します。

表 9.9-4 ポート 6 の端子状態

端子	通常動作	スリープモード	ストップモードまたはタイムベースタイマモード（SPL=0）	ストップモードまたはタイムベースタイマモード（SPL=1）
P60/SIN1 ～ P63/INT7	汎用入出力ポート	汎用入出力ポート	汎用入出力ポート	入力は遮断され、出力は Hi-Z 状態になります

SPL: 低消費電力モード制御レジスタ（LPMCR）の端子状態指定ビット

Hi-Z: ハイインピーダンス

## 9.10 I/O ポートのプログラム例

I/O ポート端子を使用したプログラム例を示します。

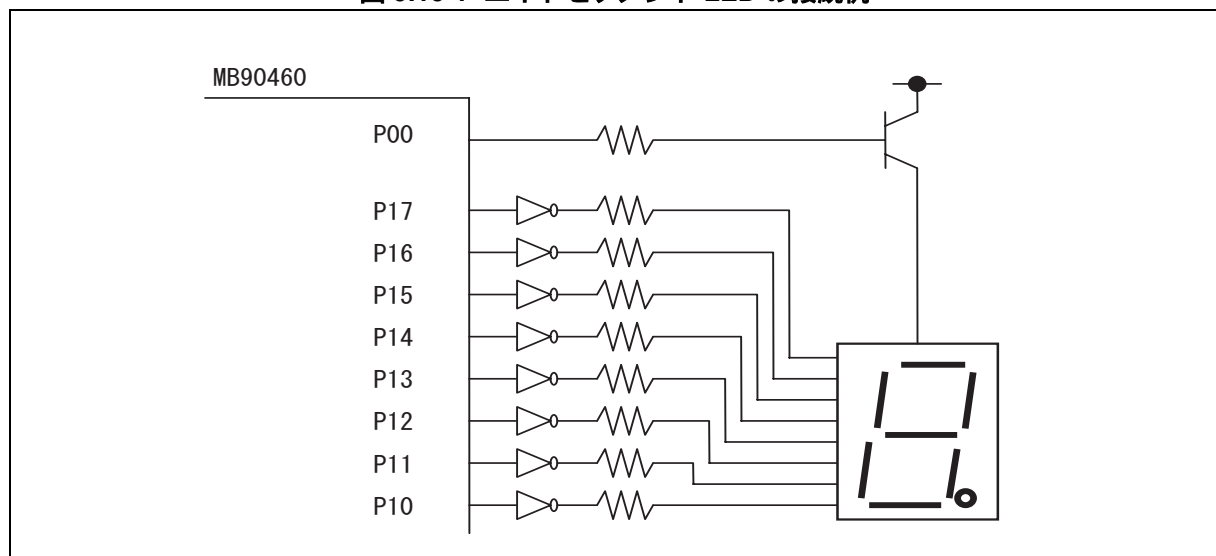
### ■ I/O ポートのプログラム例

#### ● 処理仕様

- ポート0とポート1を使用してセブンセグメント（小数点を含めるとエイトセグメント）LEDのすべての線分を"オン"にします。
- 端子P00はセブンセグメントLEDの陽極共通端子に対応しており、端子P10～P17は各セグメント端子に対応しています。

MB90460/465シリーズのポートにエイトセグメントLEDを接続する例を図9.10-1に示します。

図 9.10-1 エイトセグメント LED の接続例



#### ● コーディング例

```

PDR0    EQU    0000000H
PDR1    EQU    000001H
DDR0    EQU    000010H
DDR1    EQU    000011H
;----- メインプログラム -----
CODE          CSEG
START:
    ; 初期値
    MOV    I:PDR0,#00000000B    ; P00 に low レベル ( #XXXXXXX0B ) を設定
    MOV    I:DDR0,#11111111B    ; ポート 0 のすべてのビットを出力モードにする
    MOV    I:PDR1,#11111111B    ; ポート 1 のすべてのビットに "1" を設定する
    MOV    I:DDR1,#11111111B    ; ポート 1 のすべてのビットを出力モードにする
CODE          ENDS
;-----
END    START

```



# 第10章

---

## タイムベースタイマ

タイムベースタイマの機能と動作について説明します。

- 10.1 タイムベースタイマの概要
- 10.2 タイムベースタイマの構成
- 10.3 タイムベースタイマ制御レジスタ（TBTC）
- 10.4 タイムベースタイマ割込み
- 10.5 タイムベースタイマの動作
- 10.6 タイムベースタイマ使用上の注意
- 10.7 タイムベースタイマのプログラム例

## 10.1 タイムベースタイマの概要

タイムベースタイマは、メインクロック（原発振を 2 分周したもの）に同期してカウントアップする 18 ビットフリーランカウンタです。

タイムベースタイマは、4 つのインターバル時間を選択できるインターバルタイマ機能を備えています。タイムベースタイマはまた、発振安定待ち時間用のタイマおよびウォッチドッグタイマにクロックへ供給するための機能も備えています。

### ■ インターバルタイマ機能

インターバルタイマ機能は、割込み要求をある一定のインターバルで繰返し生成します。

- 割込み要求は、タイムベースカウンタのインターバルタイマビットのオーバフローが発生すると生成されます。
- インターバル時間は、4 種類の中から選択できます。タイムベースタイマのインターバル時間を表 10.1-1 に示します。

表 10.1-1 タイムベースタイマのインターバル時間

メインクロック周期	インターバル時間
2/HCLK (0.5 μs)	$2^{12}/\text{HCLK}$ (約 1.0ms)
	$2^{14}/\text{HCLK}$ (約 4.1ms)
	$2^{16}/\text{HCLK}$ (約 16.4ms)
	$2^{19}/\text{HCLK}$ (約 131.1ms)

HCLK: 発振クロック  
括弧内の数値は、発振クロック周波数 4MHz で動作させた場合のインターバル時間です。

## ■ クロック供給機能

クロック供給機能は、発振安定待ち時間用のタイマおよび一部のリソースへクロックを供給します。

タイムベースタイマから各リソースへ供給されるクロック周期を表 10.1-2 に示します。

表 10.1-2 タイムベースタイマから供給されるクロック周期

クロック供給先	クロック周期	備 考
発振安定待ち時間タイマ	$2^{13}/\text{HCLK}$ (約 2.0ms)	セラミック振動子の発振安定待ち時間
	$2^{15}/\text{HCLK}$ (約 8.2ms)	水晶振動子の発振安定待ち時間
	$2^{17}/\text{HCLK}$ (約 32.8ms)	
ウォッチドッグタイマ	$2^{12}/\text{HCLK}$ (約 1.0ms)	ウォッチドッグタイマの カウントアップクロック
	$2^{14}/\text{HCLK}$ (約 4.1ms)	
	$2^{16}/\text{HCLK}$ (約 16.4ms)	
	$2^{19}/\text{HCLK}$ (約 131.1ms)	

HCLK: 発振クロック

括弧内の数値は、発振クロック周波数 4MHz で動作させた場合のインターバル時間です。

---

< 参考 >      発振開始時の発振周期は不安定なので、上記に示されている発振安定待ち時間は目安として使用してください。

---

## 10.2 タイムベースタイマの構成

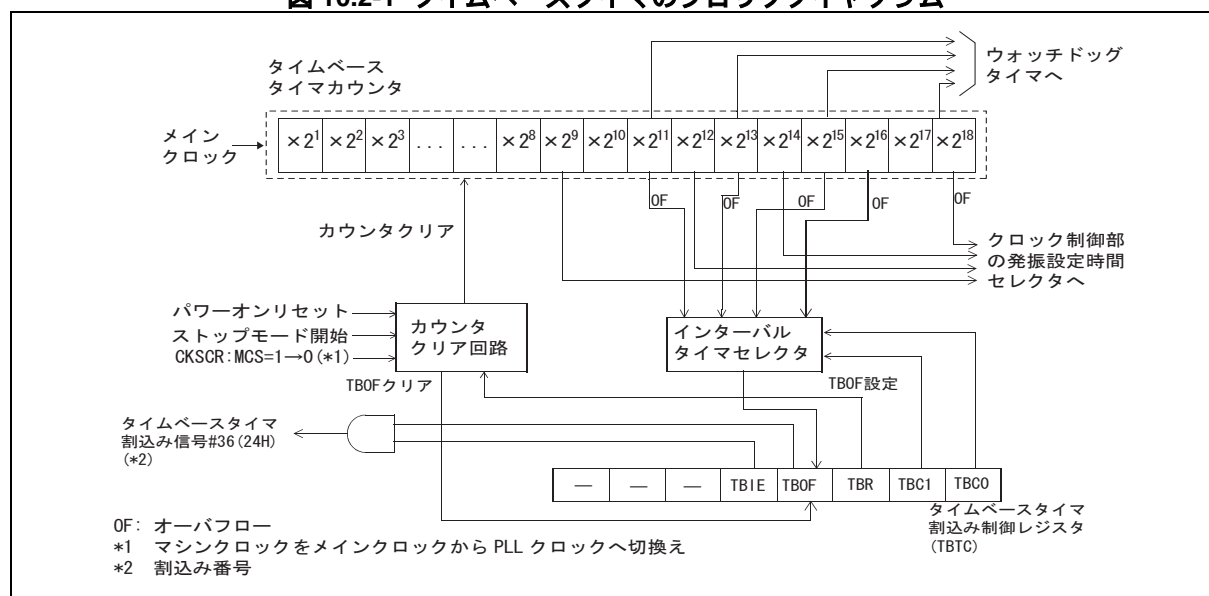
タイムベースタイマは、以下の 4 つのブロックから構成されています。

- タイムベースタイマカウンタ
- カウンタクリア回路
- インターバルタイマセクタ
- タイムベースタイマ制御レジスタ (TBTC)

### ■ タイムベースタイマのブロックダイアグラム

タイムベースタイマのブロックダイアグラムを図 10.2-1 に示します。

図 10.2-1 タイムベースタイマのブロックダイアグラム



#### ● タイムベースタイマカウンタ

この 18 ビットアップカウンタは、メインクロックをカウントクロックとして使用します。

#### ● カウンタクリア回路

カウンタをクリアするために使用されます。カウンタは、パワーオンリセット時またはストップモードへの遷移時 (LPMCR:STP=1) に TBTC レジスタの TBR:bit10 に "0" を書き込むことによってクリアされます。

#### ● インターバルタイマセクタ

タイムベースタイマカウンタの 4 種類の出力から選択します。選択されたビットのオーバーフローは、割込み要因となります。

#### ● タイムベースタイマ制御レジスタ (TBTC)

インターバル時間の選択、カウンタのクリア、割込み要求の制御および状態の確認を行います。

## 10.3 タイムベースタイマ制御レジスタ (TBTC)

タイムベースタイマ制御レジスタ (TBTC) は、インターバル時間の設定、カウンタのクリア、割り込み要求の制御および状態の確認を行います。

### ■ タイムベースタイマ制御レジスタ (TBTC)

図 10.3-1 タイムベースタイマ制御レジスタ (TBTC)

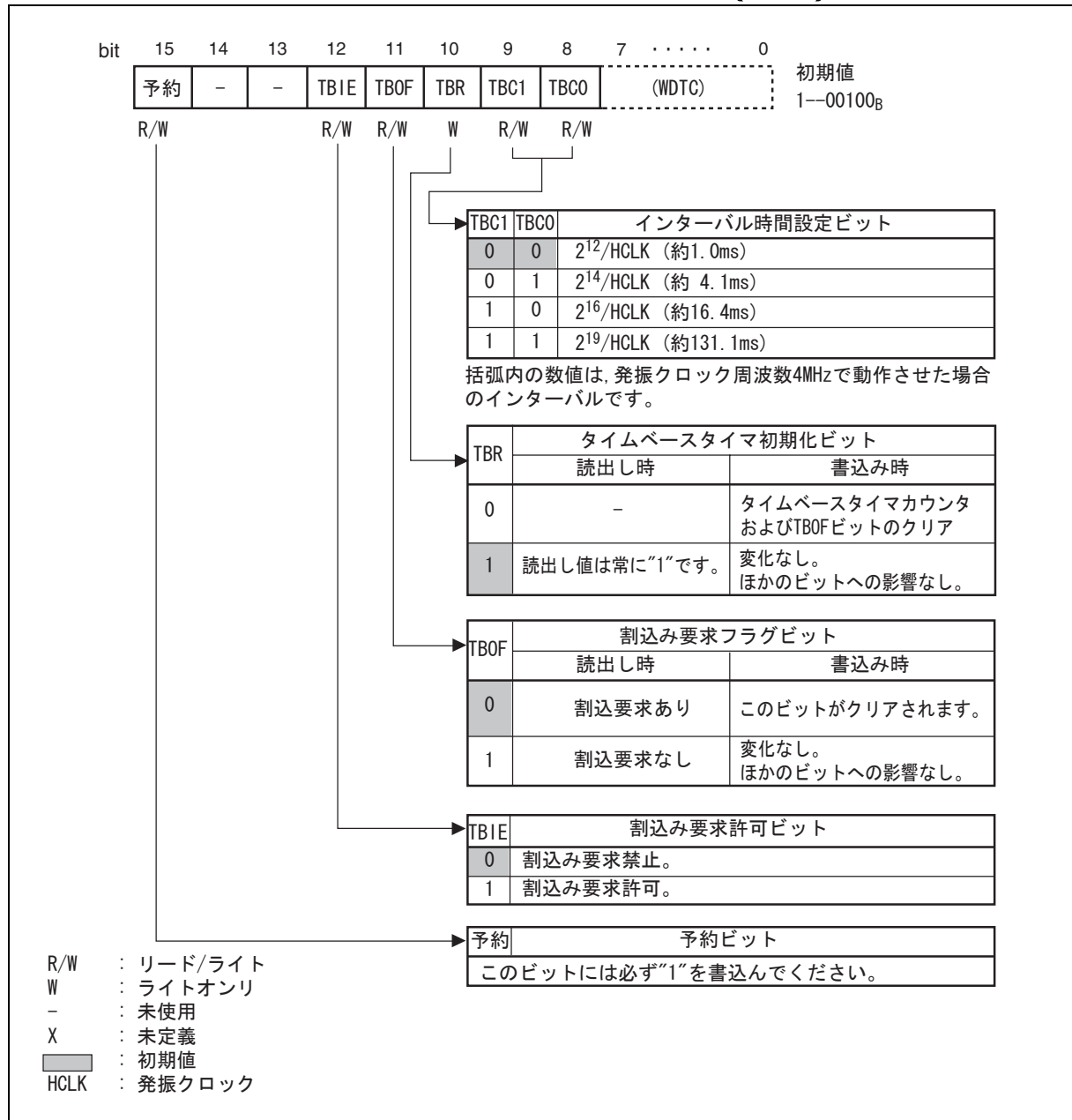




表 10.3-1 タイムベースタイマ制御レジスタ (TBTC) の各ビットの機能説明

ビット名		機 能
bit15	予約: 予約ビット	(注意事項) このビットには必ず "1" を書き込んでください。
bit14, bit13	未使用	<ul style="list-style-type: none"> <li>読出し値は不定です。</li> <li>書込みを行っても動作への影響はありません。</li> </ul>
bit12	TBIE: 割込み要求許可 ビット	<ul style="list-style-type: none"> <li>割込み要求を許可または禁止します。</li> <li>このビットが "1" のときに割込み要求フラグビット (TBOF) が "1" になると、割込み要求が発生します。</li> </ul>
BIT11	TBOF: 割込み要求フラグ ビット	<ul style="list-style-type: none"> <li>このビットは、タイムベースタイマカウンタの指定するビットよりオーバフローが発生すると、"1" が設定されます。</li> <li>このビットが "1" になったときに割込み要求許可 (TBIE) が "1" の場合、割込要求が発生します。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは変化せず、動作へ影響しません。</li> </ul> <p>(注意事項)</p> <ol style="list-style-type: none"> <li>このビットをクリアするには、TBIE ビットまたはプロセッサステータス (PS) ILM ビットを設定することでタイムベースタイマ割込みを禁止してください。</li> <li>このビットは、"0" の書込み、ストップモードへの遷移、TBR ビットによるタイムベースタイマのクリアまたはリセットよりクリアされます。</li> </ol>
bit10	TBR: タイムベースタイ マ初期化ビット	<ul style="list-style-type: none"> <li>タイムベースタイマカウンタをクリアします。</li> <li>このビットに "0" を設定した場合: カウンタがクリアされ、TBOF ビットがクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは変化せず、動作へ影響しません。</li> </ul> <p>&lt; 参考 &gt; 読出し値は常に "1" です。</p>
bit9, bit8	TBC1, TBC0: インターバル時間 選択ビット	<ul style="list-style-type: none"> <li>インターバル時間を選択します。</li> <li>タイムベースタイマカウンタのインターバルタイマビットが設定されます。</li> <li>4 つの異なるインターバル時間を選択できます。</li> </ul>

## 10.4 タイムベースタイマ割込み

タイムベースタイマは、タイムベースタイマカウンタの指定するビットよりオーバーフローが発生すると、割込み要求が生成されます。

### ■ タイムベースタイマ割込み

割込み要求フラグビット (TBTC レジスタの TBOF:bit11) に "1" が設定されるのは、タイムベースタイマカウンタが内部カウントクロックでカウントアップし、選択されたインターバルタイマビットのオーバーフローが発生した場合です。割込み要求フラグビットに "1" が設定されたとき、割込み要求許可ビットが "許可" になっている (TBTC レジスタの TBIE:12=1) 場合は、割込み要求 (#36) が生成されます。割込み処理ルーチンで TBOF ビットに "0" を書き込むと、割込み要求はクリアされます。指定されたビットのオーバーフローが発生すると、TBIE ビット値とは無関係に TBOF ビットが "1" に設定されます。

割込み要求フラグビット (TBTC レジスタの TBOF:bit11) のクリアは、TBIE ビットまたはプロセッサステータス (PS) ILM ビットの設定によりタイムベースタイマ割込みを禁止している間に行ってください。

< 参考 > TBOF ビットが "1" であり、TBIE ビットの状態が禁止から許可に遷移する ("0" "1") と、割込み要求が直ちに発生します。

### ■ タイムベースタイマ割込みと EI<sup>2</sup>OS

タイムベースタイマ割込みと EI<sup>2</sup>OS を表 10.4-1 に示します。

表 10.4-1 タイムベースタイマ割込みと EI<sup>2</sup>OS

割込み番号	割込みレベル設定レジスタ		ベクタテーブルのアドレス			EI <sup>2</sup> OS
	レジスタ名	アドレス	下位	上位	バンク	
#36 (24H)	ICR12	0000BC <sub>H</sub>	FFFF6C <sub>H</sub>	FFFF6D <sub>H</sub>	FFFF6E <sub>H</sub>	

: ICR を共用する割込み要因を使用していない場合に使用できます。

ICR12 は、タイムベースタイマ割込みとインプットキャプチャチャネル 2/3 割込みに共用されます。割込みは 2 つのアプリケーションに対して使用できますが、これらの割込みレベルは同一です。

## 10.5 タイムベースタイマの動作

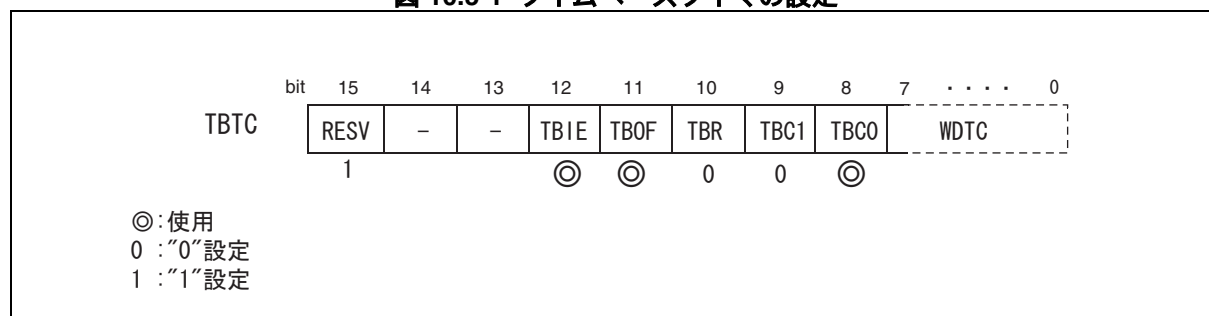
タイムベースタイマは、インターバルタイマ機能とクロック供給機能（各リソースへクロックを供給する）を備えています。

### ■ インターバルタイマ機能（タイムベースタイマ）の動作

インターバルタイマ機能は、割込み要求を各インターバルで生成します。

すべてのタイマをインターバルタイマとして動作させるには、図 10.5-1 のように設定しなければなりません。

図 10.5-1 タイムベースタイマの設定



- タイムベースタイマカウンタは、クロックが発振されている限りメインクロック（発振クロックを 2 分周したもの）と同期してカウントアップを継続します。
- カウンタがクリアされる（TBR=0）と、カウンタは "0" からカウントアップを開始します。インターバルタイマビットのオーバーフローが発生すると、割込み要求フラグビット（TBOF）に "1" が設定されます。このとき割込み要求出力がすでに許可になっている場合は、クリアされた時間を基準に選択された各インターバルで割込みを生成します。
- タイムベースタイマがクリアされるために、インターバル時間は設定された時間より長くなることがあります。

### ■ 発振安定待ち時間用タイマ機能

タイムベースタイマは、発振クロックおよび PLL クロックの発振安定待ち時間用タイマとしても使用されます。

発振安定待ち時間は、カウンタが "0"（カウントクリア）からカウントアップを開始してから発振安定待ち時間ビットのオーバフローが発生するまでの時間をインターバルとして設定されます。タイムベースタイマモードから PLL クロックモードに制御が戻ると、タイムベースタイマカウンタはクリアされていないので、発振安定待ち時間はカウントの途中から開始します。タイムベースタイマカウンタのクリアと、発振安定待ち時間を表 10.5-1 に示します。

**表 10.5-1 タイムベースタイマカウンタのクリアおよび発振安定待ち時間**

動作	カウンタ のクリア	TBOF のクリア	発振安定待ち時間
TBTC レジスタの TBR:bit10 への "0" 書込み	0	0	
パワーオンリセット	0	0	発振クロック発振安定待ち時間
ウォッチドッグリセット			
ストップモードの解除	0	0	発振クロック発振安定待ち時間 (メインクロックモードへの復帰時)
発振クロックモードから PLL クロックモードへの遷移 (MCS=1 0)	0	0	PLL クロック発振安定待ち時間
タイムベースタイマモードの解除	X	X	PLL クロック発振安定待ち時間 (PLL クロックモードへの復帰時)
スリープモードの解除	X	X	

O: 使用可能

X: 使用不可

### ■ クロック供給機能

タイムベースタイマは、ウォッチドッグタイマへクロックを供給します。タイムベースカウンタのクリアは、ウォッチドッグタイマの動作に影響を与えます。

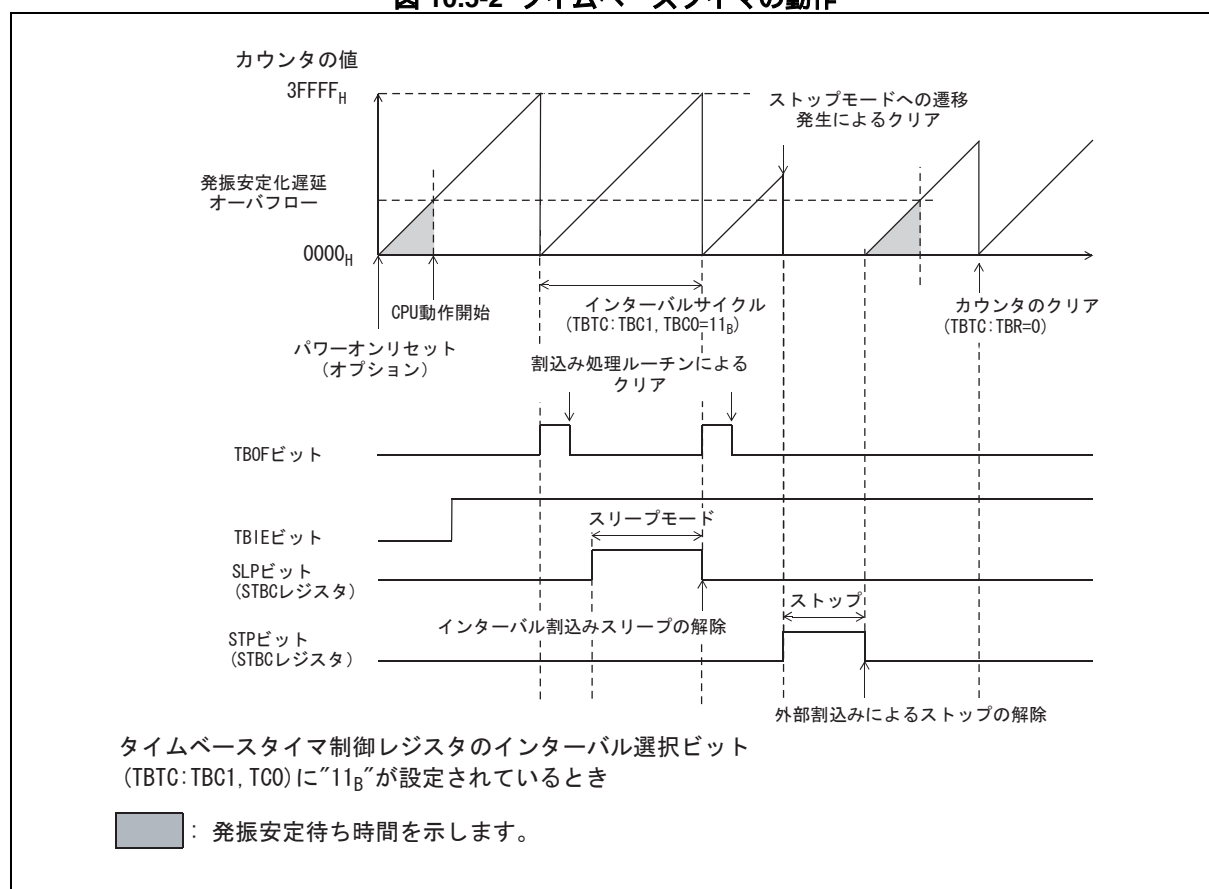
## ■ タイムベースタイマの動作

以下の動作を図 10.5-2 に示します。

- パワーオンリセットの発生
- インターバルタイマ機能動作時のスリープモードへの遷移
- ストップモードへ遷移した場合
- カウンタクリア要求の送出

ストップモードに入ると、タイムベースタイマがクリアされ、タイムベースタイマの動作が停止します。ストップモードから復帰すると、タイムベースタイマが直ちに発振安定待ち時間のカウンタを開始します。

図 10.5-2 タイムベースタイマの動作



## 10.6 タイムベースタイマ使用上の注意

割込み要求のクリアおよびタイムベースタイマカウンタのクリアによる機能への影響などの注意点を示します。

### ■ タイムベースタイマ使用上の注意

#### ● 割込み要求のクリア

タイムベースタイマ制御レジスタの TBOF ビットは、タイムベースタイマ割込みが TBIE ビットまたはプロセッサステータス (PS) 割込みレベルマスケジスタ (ILM) でマスクされている間にクリアしなければなりません。

#### ● タイムベースタイマのクリアによる影響

タイムベースタイマカウンタをクリアすると、以下が影響を受けます。

- タイムベースタイマで使用中のインターバルタイマ機能
- 使用中のウォッチドッグタイマ

#### ● 発振安定待ち時間用タイマとしてのタイムベースタイマの使用

メインストップモード時およびパワーオン時には、メインクロックの原発振は停止します。発振子の動作開始後、タイムベースタイマから供給される動作クロックは、メインクロックの発振安定待ち時間を設けるために使用されます。適切な発振安定待ち時間を、メインクロック発振子 (クロック生成部) に接続されている発振子のタイプに基づいて選択しなければなりません。詳細については、「5.5 発振安定待ち時間」をご参照ください。

#### ● タイムベースタイマからクロックが供給されるリソースについての注意

メインクロックソース発振が停止しているモード時は、タイムベースタイマカウンタがクリアされ、タイムベースタイマ動作が停止します。タイムベースタイマカウンタがクリアされると、タイムベースタイマから供給されるクロックはタイムベースタイマの初期状態から供給されます。その結果、「H」レベルが 1/2 サイクル短くなり、「L」レベルが 1/2 サイクル長くなります。ウォッチドッグタイマへのクロックもタイムベースタイマの初期状態から供給されますが、ウォッチドッグタイマは正常サイクルで動作します (ウォッチドッグタイマカウンタのクリアは、タイムベースタイマカウンタのクリアと同時にされるからです)。

## 10.7 タイムベースタイマのプログラム例

タイムベースタイマのプログラム例を示します。

### ■ タイムベースタイマのプログラム例

#### ● 処理

割込みが  $2^{12}/\text{HCLK}$  (HCLK: 発振クロック) のインターバルで繰返し生成されます。インターバル時間は、約 1.0ms (4MHz で動作させた場合) です。

#### ● コーディング例

```
ICR12 EQU 0000BCH ; タイムベースタイマ割込み制御レジスタ
TBTC EQU 0000A9H ; タイムベースタイマ制御レジスタ
TBOF EQU TBTC:3 ; 割込み要求フラグビット
;----- メインプログラム -----
CODE CSEG
START:
; : ; スタックポインタ (SP) はすでに初期化され
; : ; ているものと仮定します
AND CCR, #0BFH ; 割込みを禁止します
MOV I:ICR12, #00H ; 割込みレベル 0 (優先順位が最も高い)
MOV I:TBTC, #10010000B ; 上位 3 ビットを固定します
; : ; 割込みを許可し, TBOF をクリアします
; : ; カウンタをクリアします
; : ; インターバル  $2^{12}/\text{HCLK}$  を選択します
MOV ILM, #07H ; PS の ILM をレベル 7 に設定します
OR CCR, #40H ; 割込みを許可にします
LOOP: MOV A, #00H ; 無限ブール
MOV A, #01H
BRA LOOP
;----- 割込みプログラム -----
WARI:
CLRB I:TBOF ; 割込み要求フラグをクリアします
; :
; ユーザ処理
; :
RETI ; 割込み処理から復帰します
CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
ORG 0FF6CH ; 割込み #36 ( $24_H$ ) のベクタを設定します
DSL WARI
ORG 0FFDCH ; リセットベクタを設定します
DSL START
DB 00H ;
VECT ENDS
END START
```

# 第11章

---

## ウォッチドッグタイマ

ウォッチドッグタイマ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の概要
- 11.2 ウォッチドッグタイマの構成
- 11.3 ウォッチドッグタイマ制御レジスタ (WDTC)
- 11.4 ウォッチドッグタイマの動作
- 11.5 ウォッチドッグタイマ使用上の注意
- 11.6 ウォッチドッグタイマのプログラム例



## 11.1 ウォッチドッグタイマの概要

ウォッチドッグタイマは、タイムベースタイマから供給されるクロックをカウントクロックとして使用する 2 ビットカウンタです。ウォッチドッグタイマが起動されてからある一定時間内にウォッチドッグタイマがクリアされない場合は、CPU がリセットされます。

### ■ ウォッチドッグタイマ機能

ウォッチドッグタイマは、プログラムの暴走に対処するためのカウンタです。ウォッチドッグタイマは、一度起動されると、ある一定時間ごとに定期的にクリアする必要があります。プログラムが無限ループに入ってしまい、ウォッチドッグタイマがある一定時間経過後もクリアされない場合は、ウォッチドッグリセットが生成されます。

ウォッチドッグタイマのインターバル時間を表 11.1-1 に示します。ウォッチドッグタイマがクリアされない場合は、最小時間と最大時間の範囲内でウォッチドッグリセットが発生します。カウンタは、この表に示されている最小時間内でクリアしてください。

表 11.1-1 ウォッチドッグタイマのインターバル時間

インターバル時間		
最大 *	最小 *	発振クロックサイクル数
約 3.58ms	約 4.61ms	$2^{14} \pm 2^{11}$ サイクル
約 14.33ms	約 18.3ms	$2^{16} \pm 2^{13}$ サイクル
約 57.23ms	約 73.73ms	$2^{18} \pm 2^{15}$ サイクル
約 458.75ms	約 589.82ms	$2^{21} \pm 2^{18}$ サイクル

\*: 発振クロック周波数 4MHz で動作させた場合のインターバル時間です。

ウォッチドッグタイマのインターバル時間は、カウントクロック(タイムベースタイマから供給されるクロック)のサイクルの 3.5 倍 ~ 4.5 倍です。

詳細については、「11.4 ウォッチドッグタイマの動作」をご参照ください。

ウォッチドッグカウンタは、タイムベースタイマから供給される信号をカウントクロックとして使用する 2 ビットカウンタです。したがって、タイムベースタイマがクリアされるとウォッチドッグリセットのインターバル時間は設定された時間より長くなる場合があります。

< 参考 >	ウォッチドッグタイマは、起動されるとパワーオンリセットまたはウォッチドッグリセットで初期化され、その後、停止状態になります。ウォッチドッグタイマは、外部端子リセット、ソフトリセット、WTE ビット(ウォッチドッグタイマ制御レジスタ)への書込み、スリープモードへの遷移またはストップモードへの遷移によりクリアされます。ただし、ウォッチドッグタイマは、クリアされても停止状態にはなりません。
--------	---

## 11.2 ウォッチドッグタイマの構成

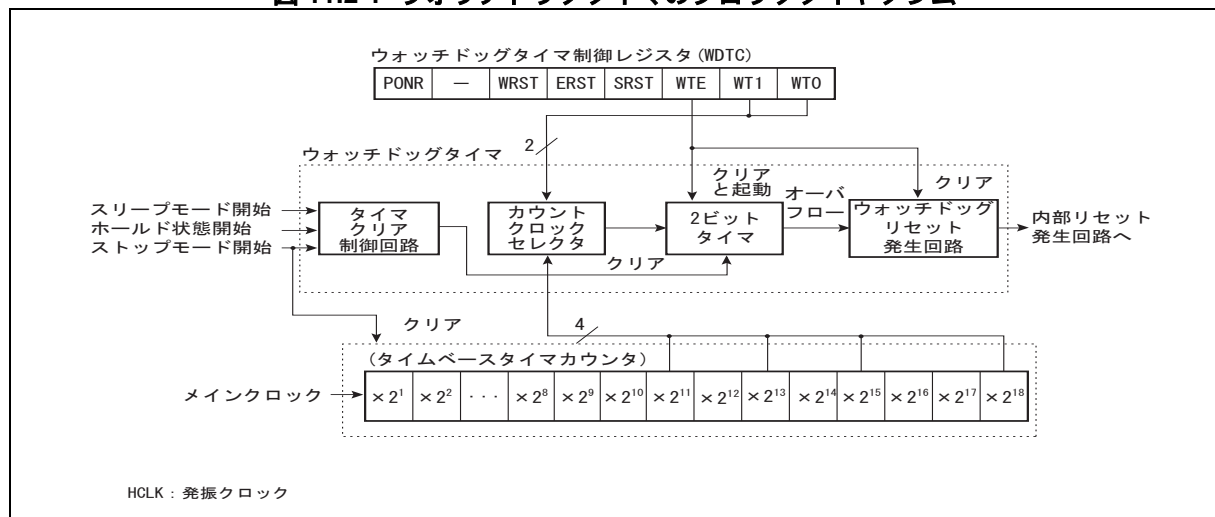
ウォッチドッグタイマは、以下の 5 ブロックから構成されています。

- カウントクロックセクタ
- ウォッチドッグカウンタ (2 ビットカウンタ)
- ウォッチドッグリセットジェネレータ
- カウンタクリア制御回路
- ウォッチドッグタイマ制御レジスタ (WDTC)

### ■ ウォッチドッグタイマのブロックダイアグラム

ウォッチドッグタイマのブロックダイアグラムを図 11.2-1 に示します。

図 11.2-1 ウォッチドッグタイマのブロックダイアグラム



#### ● カウントクロックセクタ

この回路は、4つの異なるタイムベースタイマ出力からウォッチドッグタイマのカウントクロックを選択するために使用されます。ウォッチドッグリセットのインターバル時間は、選択されたカウントクロックに基づいて決まります。

#### ● ウォッチドッグカウンタ (2 ビットカウンタ)

この2ビットアップカウンタは、タイムベースタイマ出力をカウントクロックとして使用します。

#### ● ウォッチドッグリセット発生回路

ウォッチドッグカウンタオーバーフローでリセット信号を発生します。

#### ● カウンタクリア制御回路

ウォッチドッグカウンタのクリアとウォッチドッグカウンタの動作または停止を制御します。

#### ● ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグタイマを起動またはクリアします。リセット発生要因を示します。

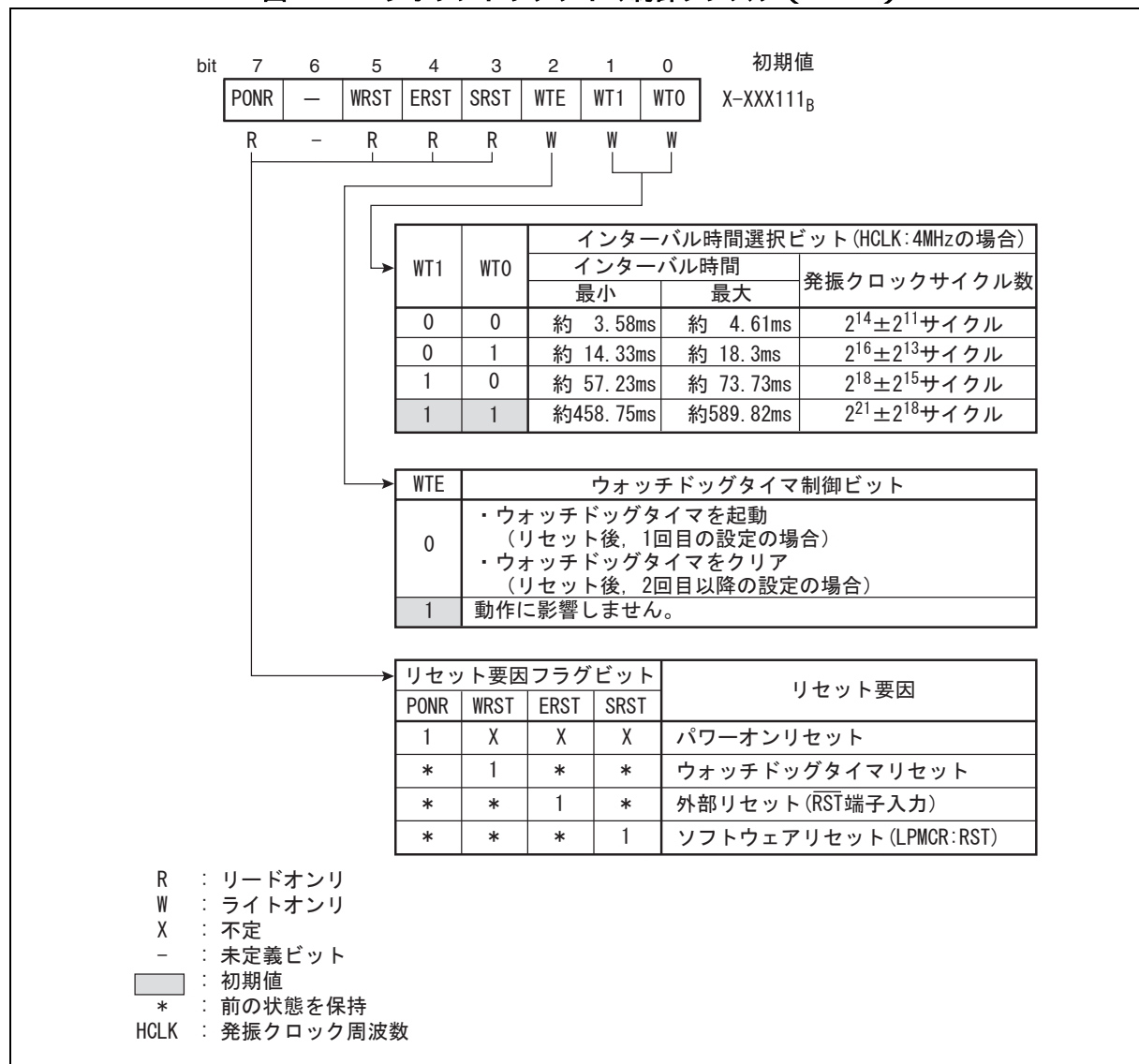
## 11.3 ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグタイマ制御レジスタ (WDTC) は、ウォッチドッグタイマの起動とクリアおよびリセット要因の表示を行います。

### ■ ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグタイマ制御レジスタ (WDTC) を図 11.3-1 に示します。また、ウォッチドッグタイマ制御レジスタ (WDTC) の各ビットの機能説明を表 11.3-1 に示します。

図 11.3-1 ウォッチドッグタイマ制御レジスタ (WDTC)



ウォッチドッグタイマのインターバル時間は、カウントクロック(タイムベースタイマ出力値)のサイクルの 3.5 倍～ 4.5 倍です。

詳細については、「11.4 ウォッチドッグタイマの動作」をご参照ください。

表 11.3-1 ウォッチドッグタイマ制御レジスタ (WDTC) の各ビットの機能説明

ビット名		機 能
bit7 ～ bit3	PONR, WRST, ERST, SRST: リセット要因フラ グビット	<ul style="list-style-type: none"> <li>リセット要因を示すための読出し専用ビットです。複数のリセット要因が発生した場合は、各リセット要因に対応するビットに "1" が設定されます。</li> <li>これらのビットは、ウォッチドッグタイマ制御レジスタ (WDTC) の読出し後、すべて "0" にクリアされます。</li> <li>パワーオン時、PONR ビット以外のビット値は不定となります。したがって、PONR ビットが "1" のとき、PONR ビット以外のビット値は無視してください。</li> </ul>
bit2	WTE: ウォッチドッグタ イマ制御ビット	<ul style="list-style-type: none"> <li>このビットに "0" を設定した場合：ウォッチドッグタイマが起動される (リセット後の最初の書込み) かまたは 2 ビットカウンタがクリアされます (リセット後の 2 番目以降の書込み)。</li> <li>このビットに "1" を設定した場合：動作へ影響しません。</li> </ul>
bit1, bit0	WT1, WT0: インターバル時間 選択ビット	<ul style="list-style-type: none"> <li>ウォッチドッグタイマのインターバル時間を選択します。</li> <li>ウォッチドッグタイマが起動されたときのデータのみが有効です。ウォッチドッグタイマが起動された後に書き込まれたデータは、無視されます。</li> <li>これらのビットは、書込み専用ビットです。</li> </ul>

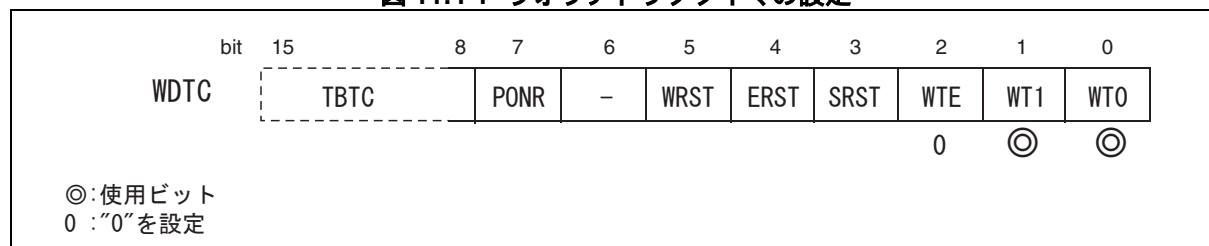
## 11.4 ウォッチドッグタイマの動作

ウォッチドッグタイマは、ウォッチドッグカウンタのオーバフローによりウォッチドッグリセットを生成します。

### ■ ウォッチドッグタイマの動作

ウォッチドッグタイマを動作させるには、図 11.4-1 のような設定が必要です。

図 11.4-1 ウォッチドッグタイマの設定



#### ● ウォッチドッグタイマの起動

- ウォッチドッグタイマは、リセット後の最初の "0" 書込みをウォッチドッグタイマ制御レジスタ (WDTC) の WTE ビットに行うと起動されます。インターバル時間は、ウォッチドッグタイマ制御レジスタの WT1 ビットと WT0 ビットを同時に設定してください。
- ウォッチドッグタイマが一度起動すると、パワーオンリセットまたはウォッチドッグリセットでしかウォッチドッグタイマを停止できません。

#### ● ウォッチドッグタイマのクリア

- 2 番目以降の "0" 書込みを WTE ビットに行うと、ウォッチドッグタイマの 2 ビットカウンタはクリアされます。このカウンタが指定インターバル時間内にクリアされない場合は、オーバフローが発生し、ウォッチドッグリセットが発生します。
- ウォッチドッグカウンタは、リセット生成、スリープモードへの遷移、ストップモードへの遷移またはクロックモードへの遷移によりリセットされます。

#### ● ウォッチドッグタイマのインターバル

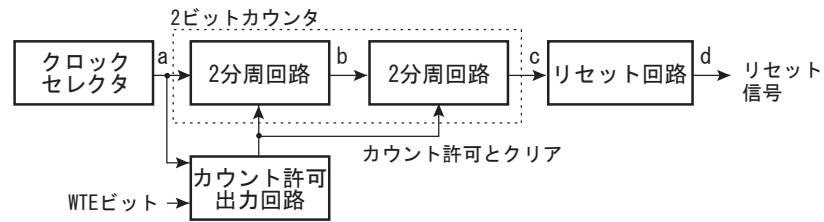
ウォッチドッグタイマのクリアタイミングとインターバル時間の関係を図 11.4-2 に示します。ウォッチドッグタイマのインターバル時間は、ウォッチドッグタイマのクリアタイミングに従って変化し、またカウントクロックサイクルの 3.5 倍 ~ 4.5 倍の長さが必要とします。

#### ● リセット要因の確認

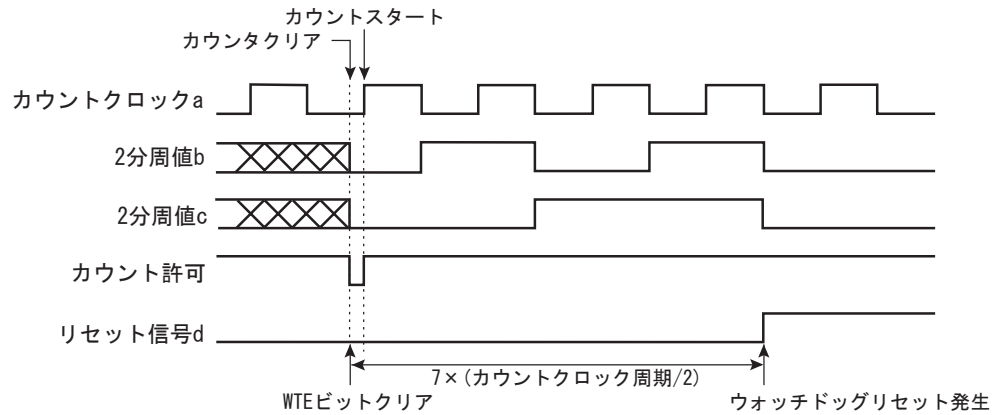
リセット要因は、リセット後のウォッチドッグタイマ制御レジスタ (WDTC) の PONR, WRST, ERST, SRST ビットを確認することで判断できます。

図 11.4-2 ウォッチドッグタイマのクリアタイミングとインターバル時間の関係

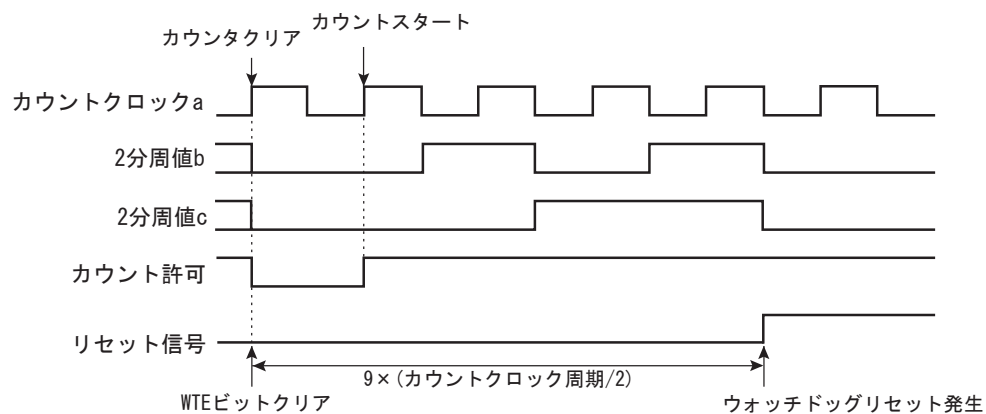
[ウォッチドッグタイマのブロック図]



[最小インターバル時間] カウントクロックの立上り直前にウォッチドッグ制御ビット(WTE)をクリアした場合



[最大インターバル時間] カウントクロックの立上り直後にウォッチドッグ制御ビット(WTE)をクリアした場合



## 11.5 ウォッチドッグタイマ使用上の注意

---

ウォッチドッグタイマ使用上の注意を以下に示します。

---

### ■ ウォッチドッグタイマ使用上の注意

#### ● ウォッチドッグタイマの停止

ウォッチドッグタイマは、一度起動されると、パワーオンリセットまたはウォッチドッグリセットが発生するまで停止できません。ウォッチドッグタイマカウンタは、外部リセットまたはソフトリセットでクリアされます。ただし、カウンタがクリアされてもウォッチドッグタイマは停止状態にはなりません。

#### ● インターバル時間

タイムベースタイマから供給される信号をインターバルカウントクロックとして使用するので、ウォッチドッグタイマのインターバル時間は、タイムベースタイマがクリアされると、設定された時間より長くなる場合があります。

タイムベースタイマは、タイムベースタイマ制御レジスタ (TBTC) のタイムベースタイマ初期化ビット (TBR) への "0" 書き込みのほか、メインクロックモードから PLL クロックモードへの遷移時にもクリアされますので、ご注意ください。

#### ● インターバル時間の選択

インターバルは、ウォッチドッグタイマが起動されたとき選択できます。起動以外の動作時に書き込まれたデータは無視されます。

#### ● プログラム作成上の注意

メインループ内でウォッチドッグタイマを繰返しクリアするプログラムを作成する場合、メインループの処理時間（割込み処理を含む）は、ウォッチドッグタイマの最小インターバル時間に等しいかまたはそれ未満でなければなりません。

#### ● タイムベースタイマモード時のウォッチドッグタイマの動作

タイムベースタイマは、タイムベースタイマモードが設定されている間動作します。ただし、ウォッチドッグタイマは一時的に停止状態になります。

## 11.6 ウォッチドッグタイマのプログラム例

ウォッチドッグタイマのプログラム例を示します。

### ■ ウォッチドッグタイマのプログラム例

#### ● 処理

- ウォッチドッグタイマは、毎回メインプログラムループ内でクリアされます。
- メインループは、ウォッチドッグタイマの最小インターバル時間内に 1 回、繰返さなければなりません。

#### ● コーディング例

```

WDTC    EQU    0000A8H    ; ウォッチドッグタイマ制御レジスタ
WTE     EQU    WDTC:2     ; ウォッチドッグ制御ビット
;----- メインプログラム -----
CODE    CSEG
START:
;   :                               ; スタックポインタ (SP) はすでに初期化されているもの
;   :                               ; と仮定します
WDG_SRART:
    MOV    WDTC, #00000011B; ウォッチドッグタイマを起動します
                                ; インターバル  $2^{21}+2^{18}$  サイクルを選択します
;----- メインループ -----
MAIN:   CLRB   I:WTE        ; ウォッチドッグタイマをクリアします
;   :                               ; このビットを定期的にクリアします
;   ユーザ処理
;   :
    JMP    MAIN              ; ウォッチドッグタイマのインターバル時間未満の時間で
                                ; ループします
CODE    ENDS
;----- ベクタ設定 -----
VECT    CSEG    ABS=0FFH
        ORG     0FFDCH      ; リセットベクタを設定します
        DSL     START
        DB      00H         ;
VECT    ENDS
        END     START

```





# 第12章

---

## 16 ビットリロードタイマ

16 ビットリロードタイマの機能と動作について説明します。

- 12.1 16 ビットリロードタイマの概要
- 12.2 16 ビットリロードタイマのブロックダイアグラム
- 12.3 16 ビットリロードタイマの端子
- 12.4 16 ビットリロードタイマのレジスタ
- 12.5 16 ビットリロードタイマの割込み
- 12.6 16 ビットリロードタイマの動作
- 12.7 16 ビットリロードタイマ使用上の注意
- 12.8 16 ビットリロードタイマのプログラム例

## 12.1 16 ビットリロードタイマの概要

16 ビットリロードタイマにおいては、以下の 2 つのクロックモードと 2 つのカウント動作モードを選択できます。

- 内部クロックモード : 3 種類の内部クロックから選択した 1 つのクロックに同期してカウントダウンします。
- イベントカウントモード : 外部端子における動作エッジの検出によってカウントダウンタイマは、カウント値が "0000<sub>H</sub>" "FFFF<sub>H</sub>" へ変更されたときにアンダフロー条件結果を定義します。すなわち、アンダフローは "リロードレジスタ設定値 +1" の後、発生することになります。
- リロードモード : カウント設定値を再リロードしてカウントを繰り返します。
- ワンショットモード : アンダフローによってカウントを停止します。カウンタアンダフローは、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) に対応しています。

MB90460/465 シリーズの 16 ビットリロードタイマは、2 つのチャンネルを内蔵しています。

### ■ 16 ビットリロードタイマの動作モード

16 ビットリロードタイマの動作モードを表 12.1-1 に示します。

表 12.1-1 16 ビットリロードタイマの動作モード

クロックモード	カウンタ動作モード	動作モード
内部クロックモード	リロードモード	ソフトウェアトリガ動作 外部トリガ入力動作 外部ゲート入力動作
	ワンショットモード	
イベントカウントモード (外部クロックモード)	リロードモード	ソフトウェアトリガ動作
	ワンショットモード	

### ■ 内部クロックモード

タイマ制御状態レジスタ上位 (TMCSR<sub>H</sub>) のカウントロック設定ビット (CSL1, CSL0:bit11, bit10) に "00<sub>B</sub>" や "01<sub>B</sub>", "10<sub>B</sub>" を設定した場合は、内部クロックモードになります。

内部クロックモード時は、以下の 3 種類の動作モードを選択できます。

#### ● ソフトウェアトリガ動作

タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のカウント許可ビット (CNTE:bit1) に "1" が設定されソフトウェアトリガビット (TRG:bit0) に "1" が設定されるとカウントを開始します。TRG ビットによるトリガ入力は、外部ゲート入力と同様に常に外部トリガ入力に対し有効です。

● 外部トリガ入力動作

タイマ制御状態レジスタ下位( TMCSRL0, TMCSRL1 )のカウンタ許可ビット( CNTE:bit1 )に "1" が設定されている場合に、動作モード設定ビット( MOD2 ~ MOD0:bit9 ~ bit7 )によって設定されているトリガ入力の有効ビット( 立上り, 立下り, 両エッジから設定可能 )が TIN0, TIN1 端子へ入力されると、カウンタを開始します。

● 外部ゲート入力動作

タイマ制御状態レジスタ下位( TMCSRL0, TMCSRL1 )のカウンタ許可ビット( CNTE:bit1 )に "1" が設定されている場合に、動作モード設定ビット( MOD2 ~ MOD0:bit9 ~ bit7 )によって設定されているゲート入力の有効レベル( "L" または "H" を設定可能 )が TIN0, TIN1 端子へ入力されると、カウンタを開始します。

## ■ イベントカウンタモード (外部クロックモード)

タイマ制御状態レジスタ上位 ( TMCSRH0, TMCSRH1 ) のカウンタブロック設定ビット ( CSL1, CSL0:bit11, bit10 )に "11<sub>B</sub>" を設定した場合に、動作設定ビット( MOD2 ~ MOD0:bit9 ~ bit7 )によって設定されているトリガ入力の有効エッジ ( 立上り, 立下り, 両エッジから設定可能 )が TIN0, TIN1 端子へ入力されると、カウンタを開始します。一定周期の外部クロックを入力する場合は、インターバルタイマとしても使用できます。

## ■ カウンタ動作

● リロードモード

16 ビットダウンカウンタのアンダフロー( "0000<sub>H</sub>" ~ "FFFF<sub>H</sub>" )により 16 ビットリロードレジスタ上位 ( TMRDH0, TMRDH1 ) の値を 16 ビットダウンカウンタへロードし、カウンタを開始します。また、アンダフローにより割り込み要求を出力しますので、インターバルタイマとして使用できます。

16 ビットリロードタイマ用インターバル時間を表 12.1-2 に示します。

表 12.1-2 16 ビットリロードタイマのインターバル時間

カウンタクロック	カウンタクロック周期	インターバル時間
内部カウンタクロック	$2^1\phi$ ( 0.125 $\mu$ s )	0.125 $\mu$ s ~ 8.192ms
	$2^3/\phi$ ( 0.5 $\mu$ s )	0.5 $\mu$ s ~ 32.768ms
	$2^5/\phi$ ( 2.0 $\mu$ s )	2.0 $\mu$ s ~ 131.1ms
外部カウンタクロック	$2^3/\phi$ ( 0.5 $\mu$ s ) 以上	0.5 $\mu$ s 以上

$\phi$ : マシクロック

( ) 内はマシクロックが 16MHz の場合です。

● ワンショットモード

カウンタダウンやカウンタ停止の間にアンダフロー( "0000<sub>H</sub>" ~ "FFFF<sub>H</sub>" )が発生した場合、アンダフローの発生のために割り込みが発生します。

カウンタ動作中、カウンタが進行中であることを示す矩形波を TO0 と TO1 端子から出力できます。

- 
- < 参考 >
- 16 ビットリロードタイマ 0 は, UART0 のボーレート生成に使用できます。
  - 16 ビットリロードタイマ 1 は, UART1 のボーレート生成や, A/D コンバータの起動トリガに使用できます。
- 

## ■ 16 ビットリロードタイマの割込みと EI<sup>2</sup>OS

16 ビットダウンカウンタのアンダフロー ( "0000<sub>H</sub>" "FFFF<sub>H</sub>" ) により, 割込み要求を出力します。

16 ビットリロードタイマの割込みと EI<sup>2</sup>OS を表 12.1-3 に示します。

表 12.1-3 16 ビットリロードタイマの割込みと EI<sup>2</sup>OS

チャンネル	割込み番号	割込み制御レジスタ		ベクタテーブルのアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	上位	バンク	
16 ビットリロードタイマ 0 <sup>*1</sup>	#30 ( 1E <sub>H</sub> )	ICR09	0000B9 <sub>H</sub>	FFFF84 <sub>H</sub>	FFFF85 <sub>H</sub>	FFFF86 <sub>H</sub>	
16 ビットリロードタイマ 1 <sup>*2</sup>	#18 ( 12 <sub>H</sub> )	ICR03	0000BA <sub>H</sub>	FFFFB4 <sub>H</sub>	FFFFB5 <sub>H</sub>	FFFFB6 <sub>H</sub>	

: 使用可能

\*1 :16 ビットリロードタイマ "0" のアンダフローと同じ割込み番号が 16 ビットタイマ 0, 1, 2 カウンタポローに割当てられます。

\*2 :16 ビットリロードタイマ "1" のアンダフローと同じ割込み番号が 16 ビットアウトプウトコンペアのチャンネル 2 に割当てられます。

## 12.2 16ビットリロードタイマのブロックダイアグラム

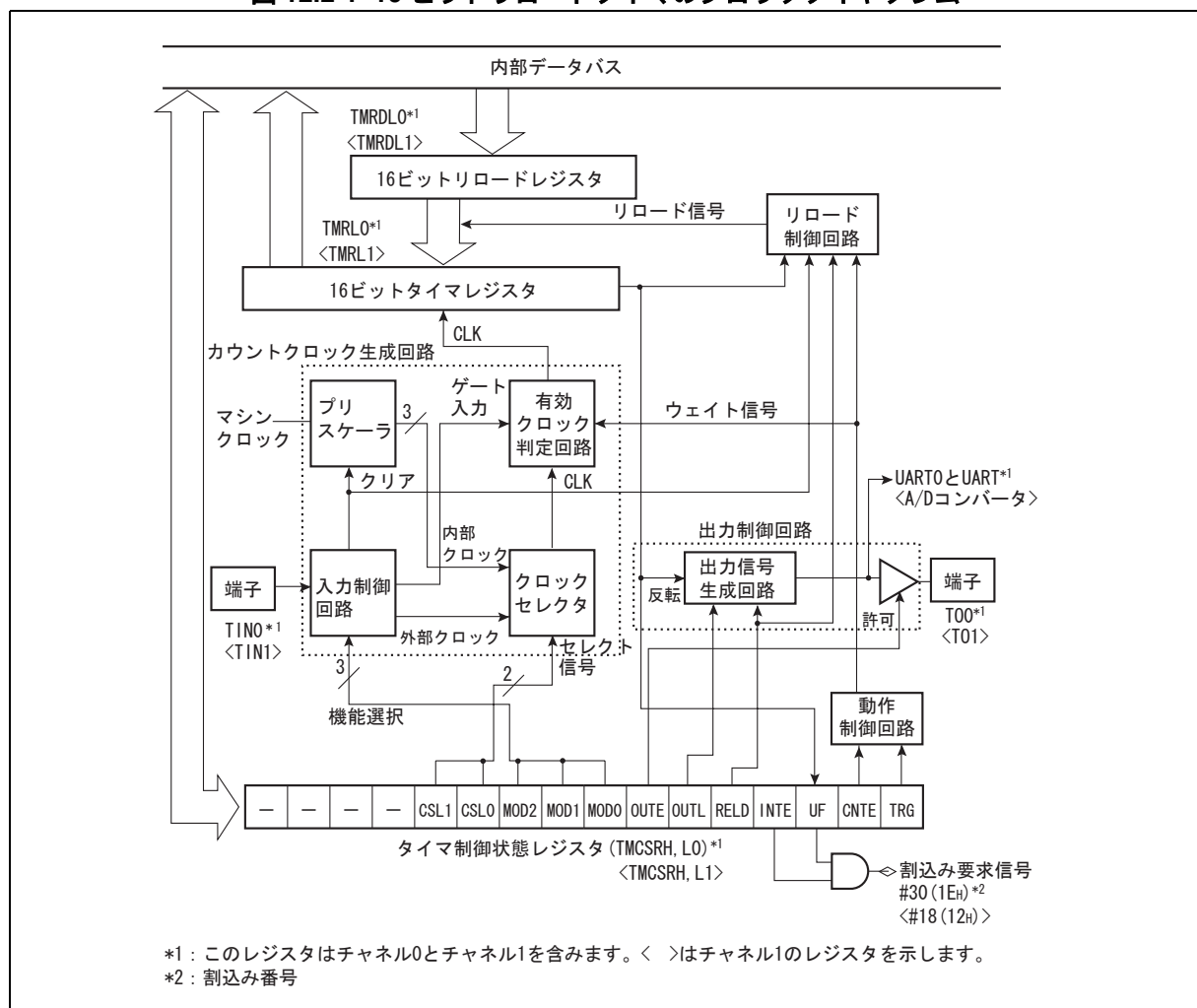
16 ビットリロードタイマ 0, 1 は、それぞれ以下の 7 種類のブロックで構成されています。

- カウントクロック生成回路
- リロード制御回路
- 出力制御回路
- 動作制御回路
- 16 ビットタイマレジスタ (TMRH0, TMRH1, TMRL0, TMRL1)
- 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1)
- タイマ制御状態レジスタ (TMCSRH0, TMCSRH1, TMCSRL0, TMCSRL1)

## ■ 16 ビットリロードタイマのブロックダイアグラム

16ビットリロードタイマのブロックダイアグラムを図 12.2-1 に示します。

**図 12.2-1 16 ビットリロードタイマのブロックダイヤグラム**



- カウントクロック生成回路

マシナクロックまたは外部入力クロックから 16 ビットリロードタイマ用のカウントクロックを生成します。

- リロード制御回路

タイマ起動とアンダフロー発生時にリロード動作を制御します。

- 出力制御回路

16 ビットダウンカウンタのアンダフローによる TO 端子出力の反転制御と、TO 端子出力の許可 / 禁止を制御します。

- 動作制御回路

16 ビットダウンカウンタの起動 / 停止を制御します。

- 16 ビットタイマレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1)

16 ビットダウンカウンタです。読出しの間は、現在のカウンタ値が読み出されます。

- 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1)

16 ビットダウンカウンタへのロード値を設定するレジスタです。16 ビットリロードレジスタの設定値を 16 ビットダウンカウンタにロードし、ダウンカウントします。

- タイマ制御状態レジスタ (TMCSRH0, TMCSRH1, TMCSRL0, TMCSRL1)

このレジスタは、16 ビットリロードタイマのカウントクロックと動作モードを選択した後、動作条件を設定し、ソフトウェアにより起動しカウントを許可 / 禁止にして、割込みと状態を制御します。リロードまたはワンショットモードおよび端子出力レベルを選択した後、タイマ出力を許可 / 禁止にして、割込みと状態を制御します。

## 12.3 16 ビットリロードタイマの端子

16 ビットリロードタイマの端子について説明します。また端子のブロックダイアグラムも示します。

### ■ 16 ビットリロードタイマの端子

16 ビットリロードタイマの端子は、汎用入出力ポートと兼用になっています。

端子状態や I/O 形式、16 ビットリロードタイマを使用するにあたって必要とする設定を表 12.3-1 に示します。

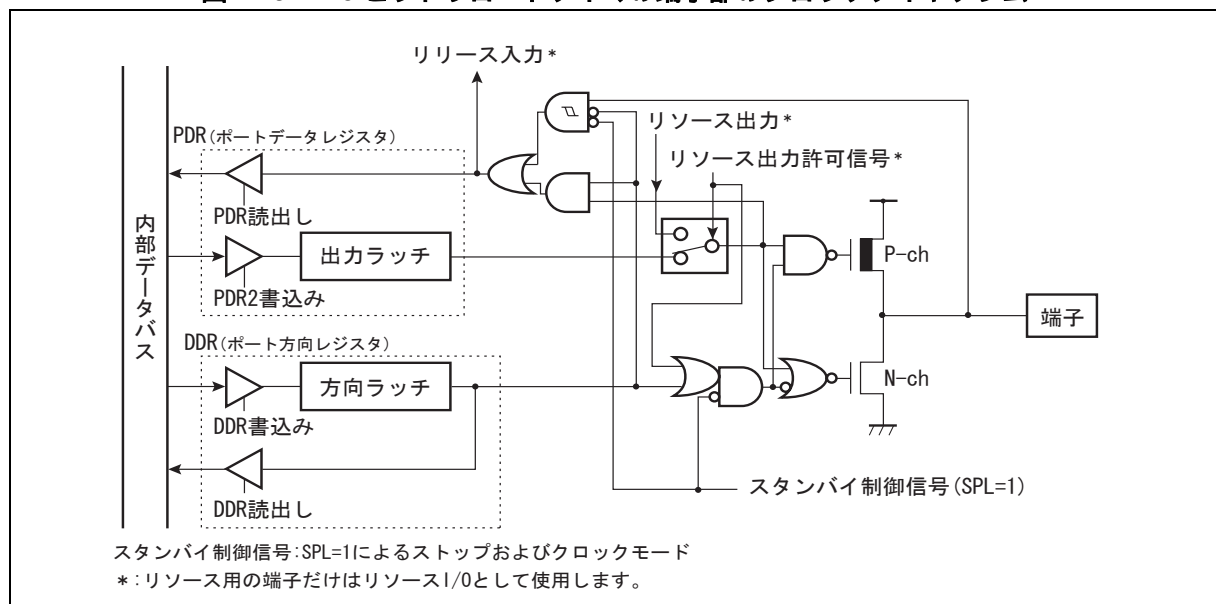
表 12.3-1 16 ビットリロードタイマの端子

端子名	端子機能	入力形式	プルアップ オプション	スタンバイ 制御	端子設定
P15/INT5/ TIN0	ポート 1 入出力 / 外部割込み入力 / タイマ入力	CMOS 出力 / CMOS ヒス テリシス入力	選択可	あり	入力ポートに設定 ( DDR1:bit7=0 )
P16/INT6/ TO0	ポート 1 入出力 / 外部割込み入力 / タイマ入力				タイマ出力を " 許可 " に設定 ( TMCSRL0:OUTE =1 )
P20/TIN1	ポート 2 入出力 / タイマ出力		なし		入力ポートに設定 ( DDR2:bit0 =0 )
P21/TO1	ポート 2 入出力 / タイマ出力				タイマ出力を " 許可 " に設定 ( TMCSRL1:OUTE =1 )

### ■ 16 ビットリロードタイマ端子のブロックダイアグラム

16 ビットリロードタイマ端子のブロックダイアグラムを図 12.3-1 に示します。

図 12.3-1 16 ビットリロードタイマの端子部のブロックダイアグラム





## 12.4 16 ビットリロードタイマのレジスタ

16 ビットリロードタイマのレジスタ一覧を示します。

### ■ 16 ビットリロードタイマのレジスタ一覧

16 ビットリロードタイマのレジスタを図 12.4-1 に示します。

図 12.4-1 16 ビットリロードタイマのレジスタ

タイマ制御状態レジスタ (上位)									
	bit 15	14	13	12	11	10	9	8	
	—	—	—	—	CSL1	CSL0	MOD2	MOD1	TMC SRH0, 1
リード/ライト ⇒	—	—	—	—	R/W	R/W	R/W	R/W	
初期値 ⇒	—	—	—	—	0	0	0	0	
タイマ制御状態レジスタ (下位)									
	bit 7	6	5	4	3	2	1	0	
	MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	TMC SRL0, 1
リード/ライト ⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇒	0	0	0	0	0	0	0	0	
16ビットタイマレジスタ (上位)									
	bit 15	14	13	12	11	10	9	8	
	D15	D14	D13	D12	D11	D10	D9	D8	TMR H0, 1
リード/ライト ⇒	R	R	R	R	R	R	R	R	
初期値 ⇒	X	X	X	X	X	X	X	X	
16ビットタイマレジスタ (下位)									
	bit 7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	TMR L0, 1
リード/ライト ⇒	R	R	R	R	R	R	R	R	
初期値 ⇒	X	X	X	X	X	X	X	X	
16ビットリロードレジスタ (上位)									
	bit 15	14	13	12	11	10	9	8	
	D15	D14	D13	D12	D11	D10	D9	D8	TMR DH0, 1
リード/ライト ⇒	W	W	W	W	W	W	W	W	
初期値 ⇒	X	X	X	X	X	X	X	X	
16ビットリロードレジスタ (下位)									
	bit 7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	TMR DL0, 1
リード/ライト ⇒	W	W	W	W	W	W	W	W	
初期値 ⇒	X	X	X	X	X	X	X	X	

### 12.4.1 タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1)

タイマ制御状態レジスタ (TMCSRH0, TMCSRH1) の bit11 ~ bit7 により, 16 ビットリロードタイマの動作モードと動作条件を設定します。

#### ■ タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1)

図 12.4-2 タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1)

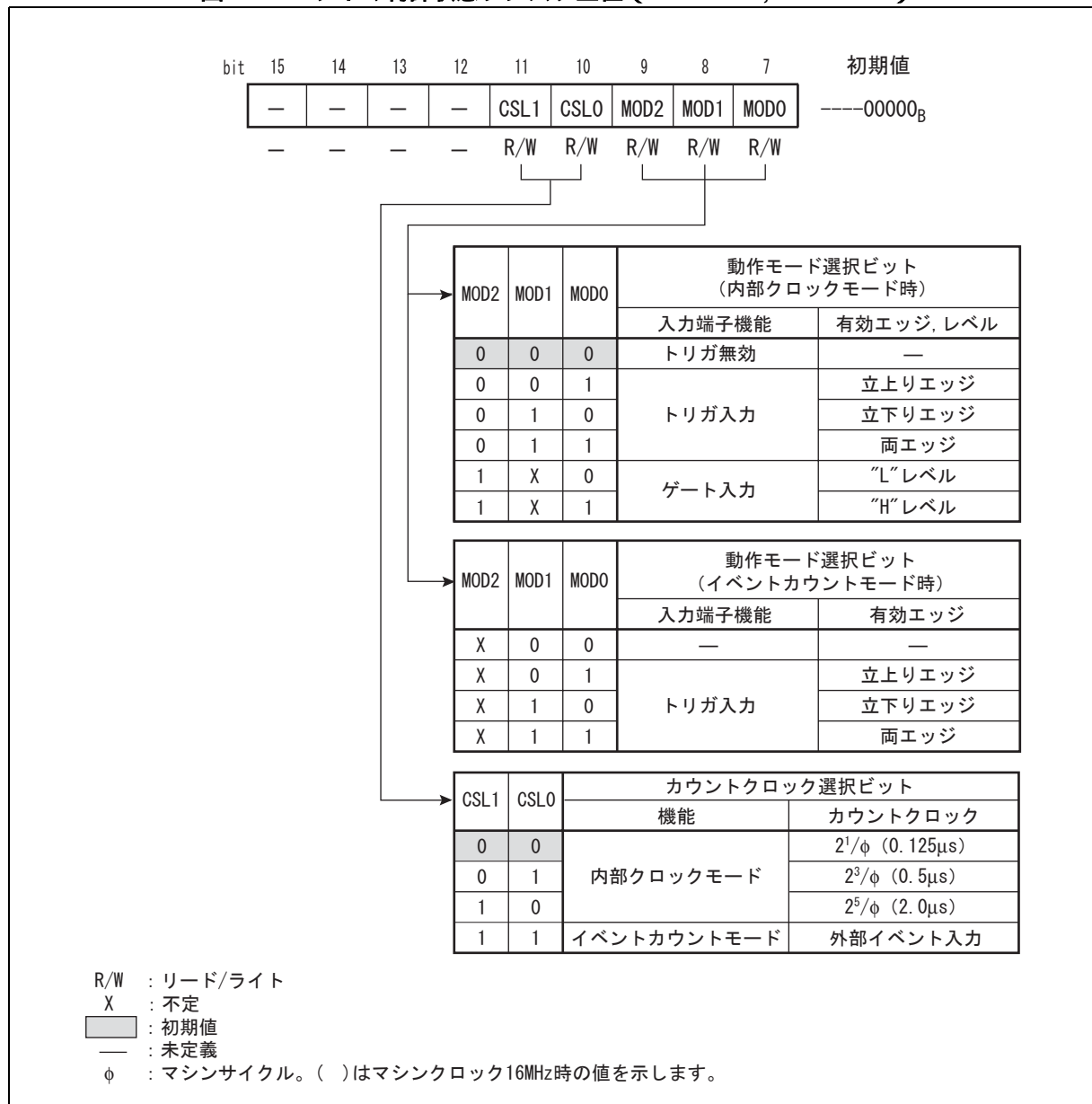


表 12.4-1 タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1)

ビット名		機 能
bit15 ~ bit12	- : 未定義ビット	<ul style="list-style-type: none"> <li>読出し値は不定値です。</li> <li>このビットへの書込みは、動作へ影響しません。</li> </ul>
bit11, bit10	CSL1, CSL0: カウントク ロック選択 ビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットリロードタイマのカウントクロックを設定します。</li> <li>これらのビットに "11<sub>B</sub>", "10<sub>B</sub>", "01<sub>B</sub>", "00<sub>B</sub>" を設定した場合： 内部クロックモードが設定されます。内部クロックはこのモード時においてカウントします。</li> <li>これらのビットに "11<sub>B</sub>" を設定した場合： イベントカウントモードが設定されます。外部クロックのエッジはこのモード時においてカウントされます。</li> </ul>
bit9 ~ bit7	MOD2, MOD1, MOD0: 動作モード選 択ビット	<p>&lt; 内部クロックモード時 &gt;</p> <ul style="list-style-type: none"> <li>MOD2:bit9 は、入力端子の機能を設定します。また、bit9 が "0" の場合は、入力端子はトリガ入力端子となり、設定した有効エッジが入力されると、16 ビットリロードレジスタ (TMRDH0, 1, TMRDL0, 1) の値を 16 ビットカウンタへロードし、カウント動作を継続します。</li> <li>MOD1, MOD0:bit8, bit7 は、有効エッジを設定します。</li> <li>MOD2:bit9 に "1" を設定した場合は、入力端子はゲート入力となり、MOD0:bit7 で設定した有効レベル信号が入力されている間、カウントを継続します。</li> <li>MOD1:bit8 に設定した値は、動作に影響しません。</li> <li>MOD0:bit7 は有効レベルを設定します。</li> </ul> <p>&lt; イベントカウントモード時 &gt;</p> <ul style="list-style-type: none"> <li>MOD2:bit9 に設定した値は、動作に影響しません。任意の値 ("0" か "1") を設定してください。</li> <li>入力端子はイベント入力用のトリガ入力端子となり、有効エッジは MOD1 と MOD0 ビットで設定されます。</li> </ul>

## 12.4.2 タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1)

タイマ制御状態レジスタ (TMCSRL0, TMCSRL1) の下位 8 ビットであり, 16 ビットリロードタイマの動作条件の設定, カウント動作の許可 / 禁止の設定, 割込み制御および割込み要求の状態を確認する機能があります。

### ■ タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1)

図 12.4-3 タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1)

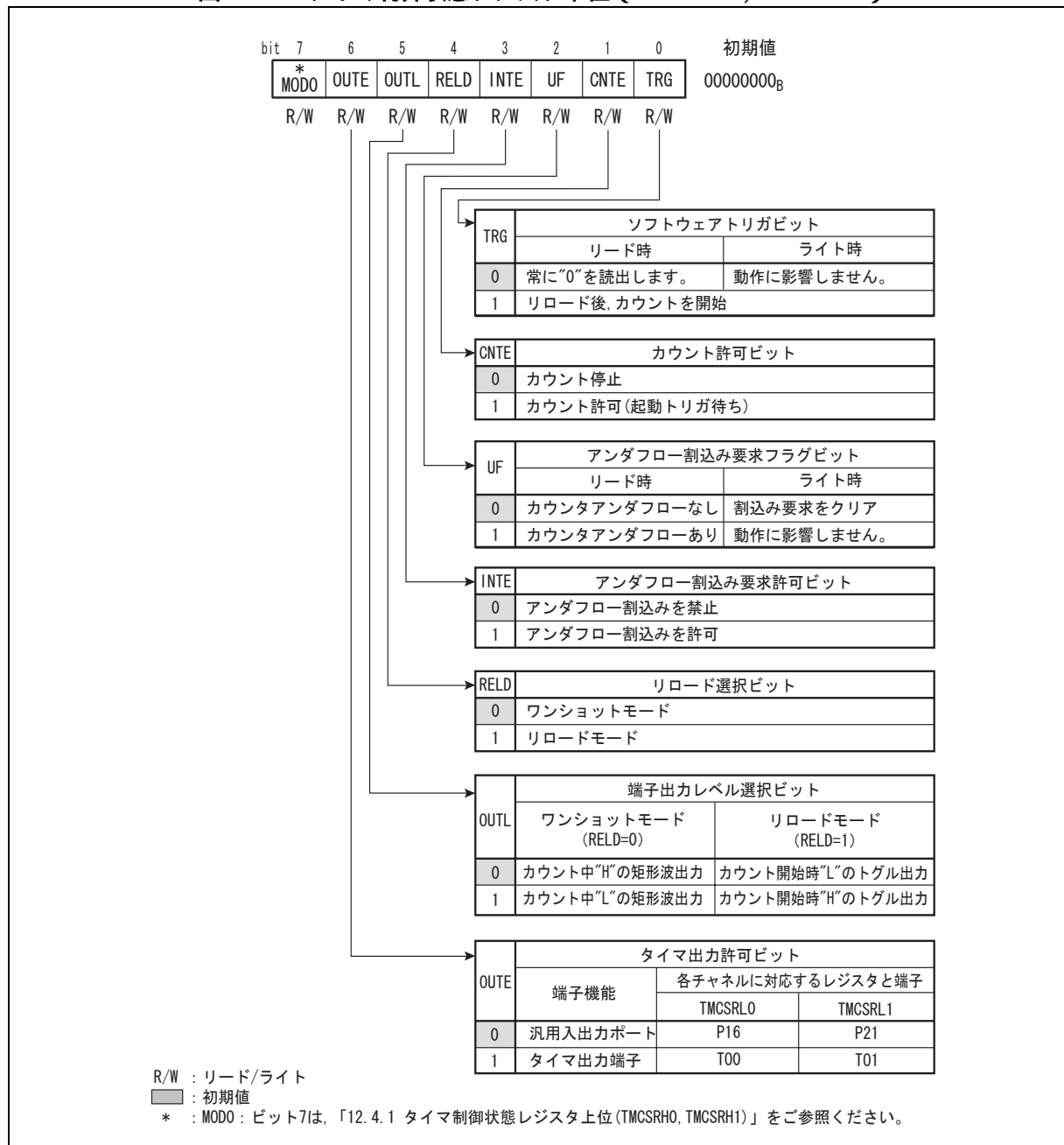


表 12.4-2 タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1)

ビット名		機 能
bit6	OUTE: タイマ出力許可ビット	<ul style="list-style-type: none"> <li>このビットは、タイマ出力端子への出力を許可 / 禁止します。</li> <li>このビットに "0" を設定した場合：端子は汎用入出力ポートとなります。</li> <li>このビットに "1" を設定した場合：端子はタイマ出力端子となります。</li> <li>タイマ出力端子の出力波形は、リロードモードを設定した場合トグル波形を出力し、ワンショットモードを設定した場合はカウント動作中を示す矩形波を出力します。</li> </ul>
bit5	OUTL: 端子出力レベル選択ビット	<ul style="list-style-type: none"> <li>タイマ出力端子への出力レベルを設定するビットです。</li> <li>このビットに "0" を設定した場合と "1" を設定した場合は、端子出力レベルが逆になります。</li> </ul>
bit4	RELD: リロード選択ビット	<ul style="list-style-type: none"> <li>このビットは、リロード動作を許可します。</li> <li>このビットに "0" を設定した場合： ワンショットモードとなり、16 ビットカウンタのアンダフローによりカウントを停止します。</li> <li>このビットに "1" を設定した場合： リロードモードとなり、16 ビットカウンタのアンダフローの発生と同時に、16 ビットリロードレジスタに設定した値が 16 ビットカウンタにロードされ、カウントを継続します。</li> </ul>
bit3	INTE: アンダフロー割込み要求許可ビット	<ul style="list-style-type: none"> <li>このビットは割込み要求を許可 / 禁止します。</li> <li>このビットに "1" を設定した場合： アンダフロー割込み要求フラグビット (UF) に "1" がセットされると、タイマは割込み要求を出力します。</li> </ul>
bit2	UF: アンダフロー割込み要求フラグビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットカウンタのアンダフローにより "1" がセットされます。</li> <li>アンダフロー割込み要求許可ビット (INTE) に "1" を設定している場合にこのビットに "1" がセットされると割込み要求を出力します。</li> <li>このビットに "0" を設定した場合：割込み要求がクリアされます。</li> <li>このビットに "1" を設定した場合：動作へ影響しません。</li> <li>このビットは、EI<sup>2</sup>OS の起動により、"0" にクリアされます。</li> </ul>
bit1	CNTE: カウント許可ビット	<ul style="list-style-type: none"> <li>このビットは、カウント動作を許可 / 禁止します。</li> <li>このビットに "1" を設定した場合： カウンタは起動トリガ待ち状態となります。トリガの発生により、実カウントを開始します。</li> </ul>
bit0	TRG: ソフトウェアトリガビット	<ul style="list-style-type: none"> <li>このビットは、インタバルタイマ機能またはカウンタ機能をソフトウェアで起動します。</li> <li>このビットに "0" 設定した場合：動作へ影響しません。</li> <li>このビットに "1" 設定した場合： ソフトウェアトリガとなり、16 ビットリロードレジスタに設定した値が 16 ビットダウンカウンタにロードされ、カウントを開始します。</li> <li>このビットの読出し値は "0" です。</li> <li>動作モードにかかわらず CNTE ビットに "1" が設定されたとき、トリガ入力は常に "許可" です。</li> </ul>



## 12.4.4 16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1)

16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1) は, 16 ビットダウンカウンタへのリロード値を設定するレジスタです。16 ビットリロードレジスタに設定された値が, 16 ビットダウンカウンタにリロードされ, ダウンカウントを実行します。

### ■ 16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1)

16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1) を図 12.4-5 に示します。

図 12.4-5 16 ビットリロードレジスタ上位, 下位 (TMRDH0, TMRDH1, TMRDL0, TMRDL1)

bit	15	14	13	12	11	10	9	8	
	D15	D14	D13	D12	D11	D10	D9	D8	TMRDH0, TMRDH1
リード/ライト⇒	W	W	W	W	W	W	W	W	
初期値⇒	X	X	X	X	X	X	X	X	
bit	7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	TMRDL0, TMRDL1
リード/ライト⇒	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

16 ビットリロードタイマのモードにかかわらず, カウントが禁止 (TMCSRL0, TMCSRL1 レジスタの CNTE:bit1=0) のときは, カウンタの初期値がこのレジスタに設定されます。カウントが許可 (TMCSRL0, TMCSRL1 レジスタの CNTE: レジスタ =1) であり, かつカウントが開始したとき, このレジスタに書き込まれた値からカウントダウンを開始します。

リロードモード時において, アンダフローが発生しダウンカウントが継続しているときには, 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) に設定された値はカウンタへリロードされます。

- < 注意事項 >
- 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) へ値を設定する場合は, カウント動作を停止 (TMCSRL0, TMCSRL1 の CNTE:bit1=0) してください。
  - 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) への設定は, ワード転送命令 (MOVW 003AH, A) をご使用ください。
  - 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) は, 書込み専用のレジスタであり, 読み専用レジスタである 16 ビットタイマレジスタ (TMRH0, TMRH1, TMRL0, TMRL1) と同一アドレスに配置されているため, 書込み値と読み出し値は異なります。リードモディファイライト命令は使用できません。

## 12.5 16 ビットリロードタイマの割込み

16 ビットリロードタイマは、16 ビットダウンカウンタのアンダフロー時に割込み要求を出力します。拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) に対応しています。

### ■ 16 ビットリロードタイマの割込み

16 ビットリロードタイマの割込み制御ビットと割込み要因を表 12.5-1 に示します。

表 12.5-1 16 ビットリロードタイマの割込み制御ビットと割込み要因

	16 ビットリロードタイマ 0	16 ビットリロードタイマ 1
アンダフロー割込み要求フラグビット	TMCSRL0 の UF:bit2	TMCSRL1 の UF:bit2
割込み要求許可ビット	TMCSRL0 の INTE:bit3	TMCSRL1 の INTE:bit3
割込み要因	16 ビットダウンカウンタ (TMRH0, TMRL0) のアンダフロー	16 ビットダウンカウンタ (TMRH1, TMRL1) のアンダフロー

16 ビットリロードタイマでは、16 ビットダウンカウンタのアンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") により、タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のアンダフロー割込み要求フラグビット (UF:bit2) に "1" がセットされます。アンダフロー割込み要求許可ビットが割込み要求を許可 (TMCSRL0, TMCSRL1 の INTE:bit3=1) にしている場合、割込み要求を割込みコントローラへ出力します。

### ■ 16 ビットリロードタイマの割込みと EI<sup>2</sup>OS 機能

16 ビットリロードタイマの割込みと EI<sup>2</sup>OS 機能を表 12.5-2 に示します。

表 12.5-2 16 ビットリロードタイマの割込みと EI<sup>2</sup>OS 機能

チャンネル	割込み番号	割込み制御レジスタ		ベクタテーブルのアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	上位	バンク	
16 ビットリロードタイマ 0 <sup>*1</sup>	#30 (1E <sub>H</sub> )	ICR09	0000B9 <sub>H</sub>	FFFF84 <sub>H</sub>	FFFF85 <sub>H</sub>	FFFF86 <sub>H</sub>	
16 ビットリロードタイマ 1 <sup>*2</sup>	#18 (12 <sub>H</sub> )	ICR03	0000B3 <sub>H</sub>	FFFFB4 <sub>H</sub>	FFFFB5 <sub>H</sub>	FFFFB6 <sub>H</sub>	

: 使用可能

\*1 : 波形シーケンサ 16 ビットタイマカウンタボローと同じ割込み番号が、16 ビットリロードタイマ 0 に割付けられています。

\*2 : アウトプットコンペア ch.2 一致と同じ割込み番号が、16 ビットリロードタイマ 1 に割付けられています。



## 12.6 16 ビットリロードタイマの動作

16 ビットリロードタイマの設定と、カウンタの動作状態について説明します。

### ■ 16 ビットリロードタイマの設定

#### ● 内部クロックモードの設定

インターバルタイマとして動作させるには、下位のレジスタを図 12.6-1 のように設定してください。

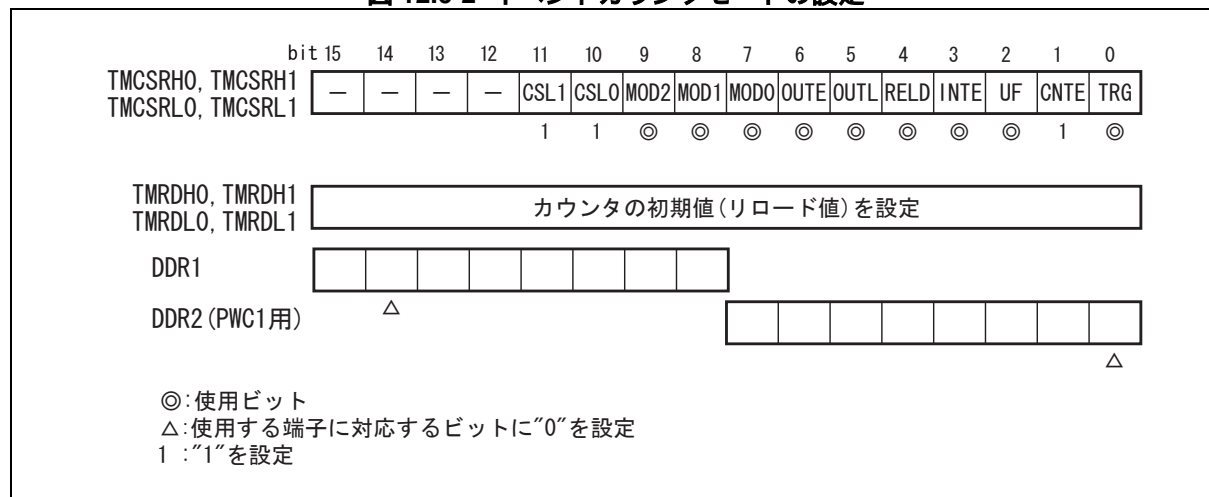
図 12.6-1 内部クロックモードの設定



#### ● イベントカウントモードの設定

イベントカウンタとして動作させるには、下位のレジスタを図 12.6-2 のように設定してください。

図 12.6-2 イベントカウンタモードの設定

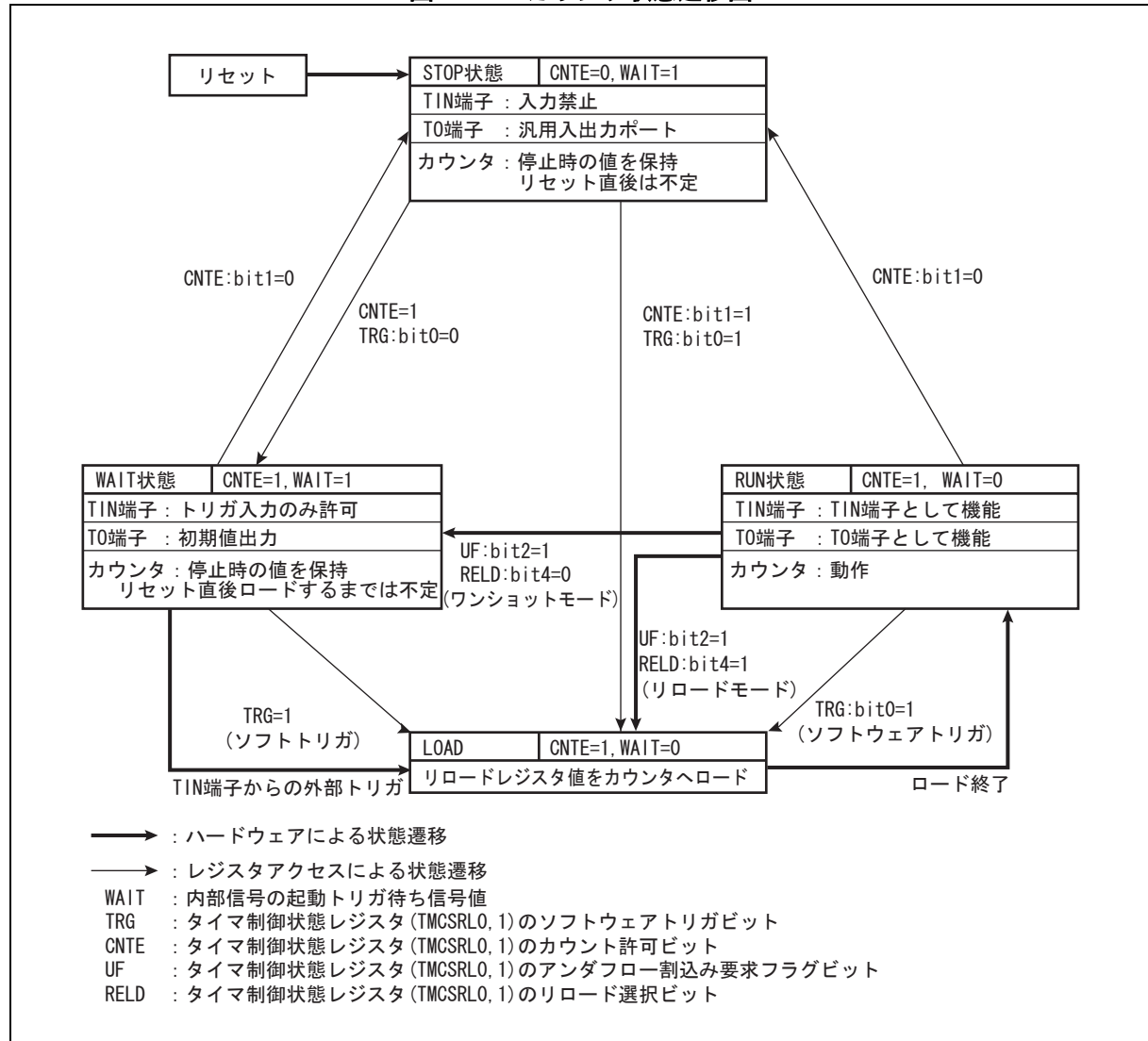


## ■ カウンタの動作状態

カウンタの状態は、タイマ制御状態レジスタ (TMCSR0, TMCSR1) のカウント許可ビット (CNTE:bit1) 値と内部信号の起動トリガ待ち信号値 (WAIT) で決まります。STOP 状態 (停止状態)、WAIT 状態 (起動トリガ待ち状態) および RUN 状態 (動作状態) が設定可能です。

これらのカウンタの状態遷移を図 12.6-3 に示します。

図 12.6-3 カウンタ状態遷移図



## 12.6.1 内部クロックモード（リロードモード）

内部カウントクロックに同期して、16 ビットダウンカウンタをダウンカウントし、アンダフロー（"0000<sub>H</sub>" "FFFF<sub>H</sub>"）のために、CPU への割り込み要求を出力します。また、タイマ出力端子からトグル波形を出力できます。

### ■ 内部クロックモード（リロードモード）の動作

タイマ制御状態レジスタ（TMCSRL0, TMCSRL1）のカウント許可ビット（CNTE:bit1）に "1" を設定してカウントが許可の場合に、ソフトウェアトリガビット（TRG:bit0）に "1" を設定あるいは外部トリガによりタイマを開始すると、16 ビットリロードレジスタ（TMRDH0, TMRDH1, TMRDL0, TMRDL1）に設定した値が 16 ビットダウンカウンタにリロードされ、ダウンカウントを開始します。カウント許可ビット（CNTE:bit1）とソフトウェアトリガビット（TRG:bit0）同時に "1" を設定したときにカウント動作を許可すると、同時にカウントを開始します。

16 ビットカウンタのアンダフロー（"0000<sub>H</sub>" "FFFF<sub>H</sub>"）により、16 ビットリロードレジスタ（TMRDH0, TMRDH1, TMRDL0, TMRDL1）に設定した値が、16 ビットカウンタにリロードされ、カウントを継続します。アンダフロー割り込み要求許可ビット（INTE:bit3）に "1" を設定しているときに、タイマ制御状態レジスタ下位（TMCSRL0, TMCSRL1）のアンダフロー割り込み要求フラグビット（UF:bit2）に "1" が設定されると、割り込み要求を出力します。

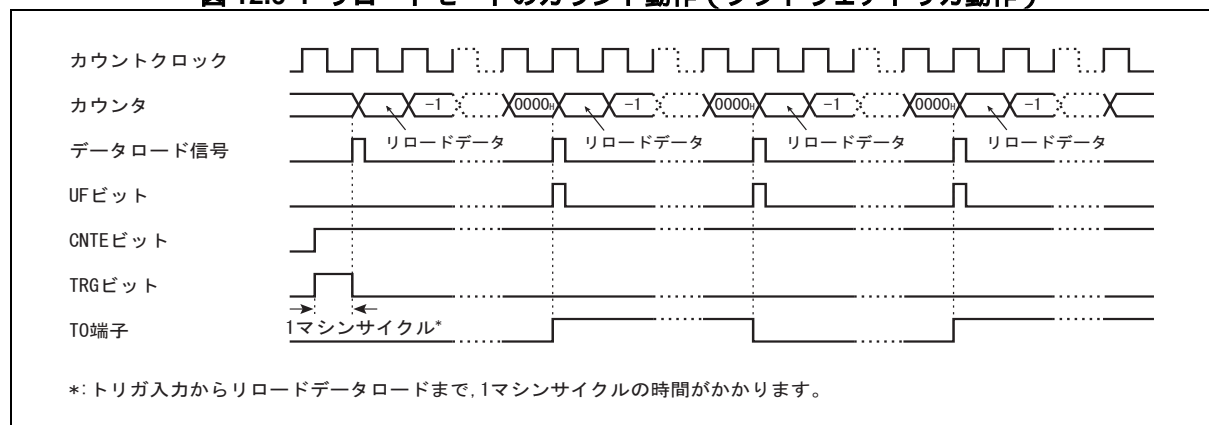
TO0, TO1 端子からは、アンダフローごとに反転するトグル波形が出力されます。

#### ● ソフトウェアトリガ動作

タイマ制御状態レジスタ下位（TMCSRL0, TMCSRL1）のカウント許可ビット（CNTE:bit1）に "1" を設定している場合に、ソフトウェアトリガビット（TRG:bit0）に "1" を設定すると、カウントを開始します。

リロードモード時のソフトウェアトリガ動作を図 12.6-4 に示します。

図 12.6-4 リロードモードのカウント動作（ソフトウェアトリガ動作）

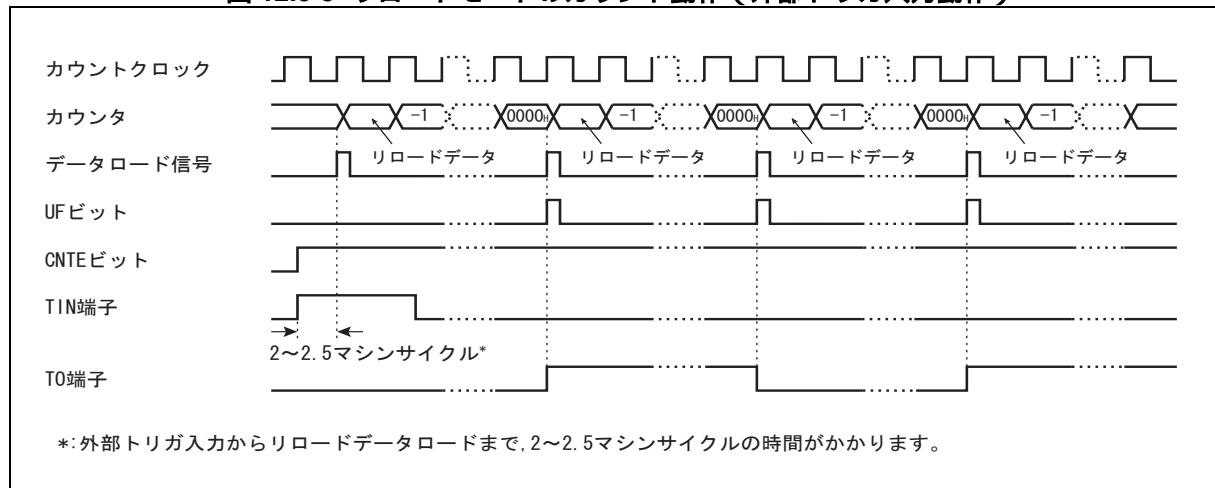


● 外部トリガ入力動作

タイマ制御状態レジスタ (TMCSRL0, TMCSRL1) のカウント許可ビット (CNTE:bit1) に "1" を設定している場合に、動作モード選択ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) が TIN 端子に入力されると、カウントを開始します。

リロードモード時における外部トリガ入力動作を図 12.6-5 に示します。

図 12.6-5 リロードモードのカウント動作 (外部トリガ入力動作)



< 注意事項 > TIN0, TIN1 端子へ入力するトリガパルス幅は,  $2/\phi$  ( $\phi$ : マシンクロック) 以上としてください。

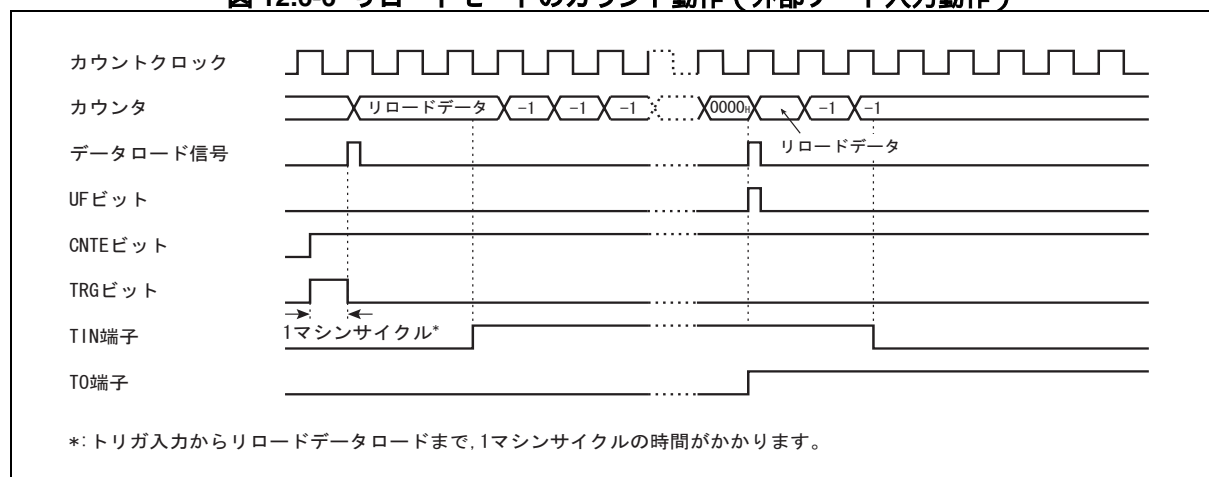
### ● ゲート入力動作

タイマ制御状態レジスタ下位( TMCSRL0, TMCSRL1 )のカウンタ許可ビット( CNTE:bit1 )に "1" を設定している場合に, ソフトウェアトリガビット ( TRG:bit0 ) に "1" を設定すると, カウントを開始します。

動作モード選択ビット ( MOD2, MOD1, MOD0 ) で設定しているゲート入力の許可レベル ( "L" または "H" を設定可能 ) が TIN 端子に入力されている間, カウントします。

リロードモード時におけるゲート入力動作を図 12.6-6 に示します。

図 12.6-6 リロードモードのカウンタ動作 (外部ゲート入力動作)



< 注意事項 > TIN0, TIN1 端子へ入力するゲート入力のパルス幅は,  $2/\phi$  ( $\phi$ : マシンクロック) 以上としてください。

## 12.6.2 内部クロックモード (ワンショットモード)

内部カウントクロックに同期して、16 ビットカウンタのダウンカウント用 16 ビットリロードタイマを使用し、アンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") のために、CPU に対する割込み要求を出力します。また、TO 端子から、カウント中を示す矩形波を出力できます。

### ■ 内部クロックモード (ワンショットモード)

タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のカウント許可ビット (CNTE:bit1) に "1" を設定している場合に、ソフトウェアトリガビット (TRG:bit0) に "1" を設定または動作モード選択ビット (MOD2, MOD1, MOD0) で選択されたトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) が TIN 端子に入力されると、16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) に設定した値が 16 ビットカウンタにリロードされ、ダウンカウントを開始します。カウント許可ビット (CNT:bit1) とソフトウェアトリガビット (TRG:bit0) 同時に "1" を設定した場合は、カウントを許可すると同時にカウントを開始します。

16 ビットカウンタのアンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") により、16 ビットカウンタは "FFFF<sub>H</sub>" の状態でカウントを停止します。

タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のアンダフロー割込み要求フラグビット (UF:bit2) に "1" がセットされ、アンダフロー割込み要求許可ビット (INTE:bit3) に "1" を設定している場合は、割込み要求を出力します。

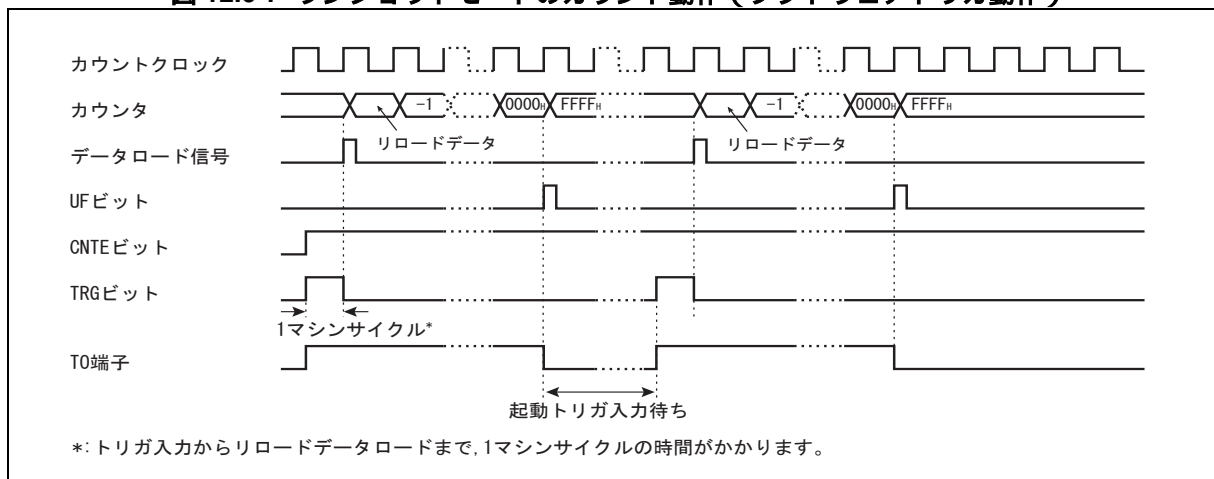
TO0, TO1 端子からは、カウント中を示す矩形波を出力できます。

#### ● ソフトウェアトリガ動作

タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のカウント許可ビット (CNTE:bit1) に "1" を設定している場合に、ソフトウェアトリガビット (TRG:bit0) に "1" を設定すると、カウントを開始します。

ワンショットモード時におけるソフトウェアトリガ動作を図 12.6-7 に示します。

図 12.6-7 ワンショットモードのカウント動作 (ソフトウェアトリガ動作)

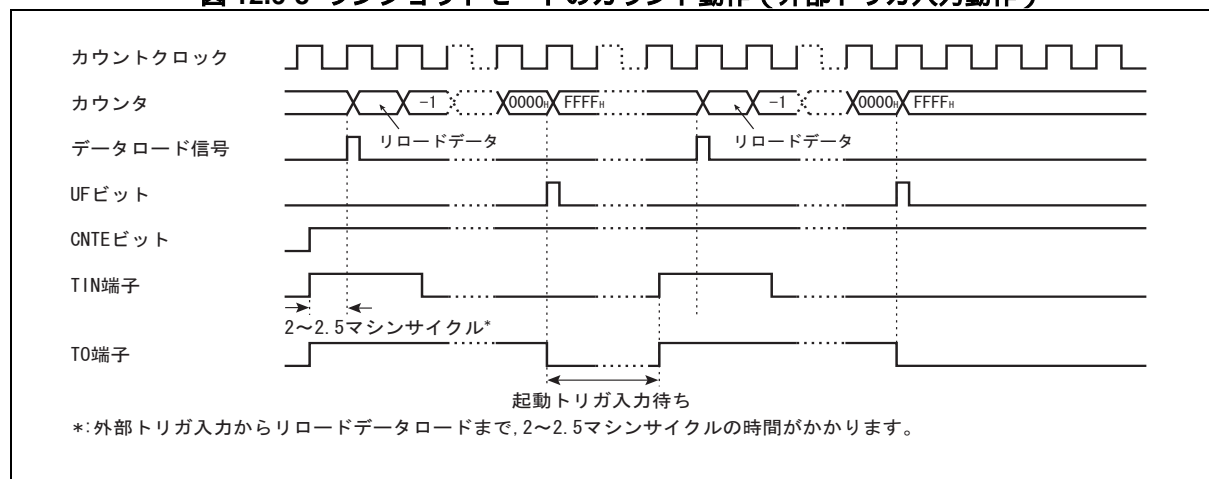


## ● 外部トリガ入力動作

タイマ制御状態レジスタ下位( TMC SRL0, TMC SRL1 )のカウンタ許可ビット( CNTE:bit1 )に "1" を設定している場合に、動作モード選択ビット( MOD2, MOD1, MOD0 )で設定しているトリガ入力の有効エッジ( 立上り, 立下り, 両エッジから設定可能 )が TIN 端子に入力されると、カウンタを開始します。

ワンショットモード時における外部トリガ入力動作を図 12.6-8 に示します。

図 12.6-8 ワンショットモードのカウンタ動作 (外部トリガ入力動作)



< 注意事項 > TIN0, TIN1 端子へ入力するトリガパルス幅は、 $2/\phi$  ( $\phi$ : マシンクロック) 以上としてください。

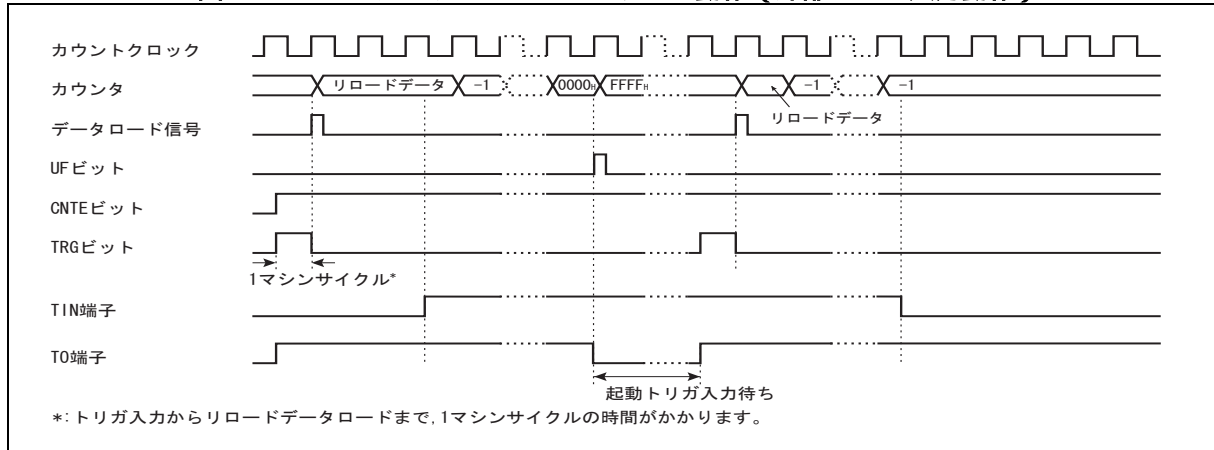
### ● ゲート入力動作

タイマ制御状態レジスタ下位( TMCSRL0, TMCSRL1 )のカウンタ許可ビット( CNTE:bit1 )に "1" を設定している場合に, ソフトウェアトリガビット ( TRG:bit0 ) に "1" を設定すると, カウントを開始します。

動作モード選択ビット ( MOD2, MOD1, MOD0 ) で設定しているトリガ入力の許可レベル ( "L" または "H" を設定可能 ) が TIN0, TIN1 端子に入力されている間, カウントします。

ワンショットモード時における外部ゲート入力動作を図 12.6-9 に示します。

図 12.6-9 ワンショットモードのカウンタ動作 (外部ゲート入力動作)



< 注意事項 > TIN0, TIN1 端子へ入力するゲート入力のパルス幅は,  $2\phi$  ( $\phi$ : マシンクロック) 以上としてください。



## 12.6.3 イベントカウントモード

TIN 端子に入力されたパルスの有効エッジを検出することにより、16 ビットダウンカウンタをダウンカウントし、アンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") により、CPU への割込み要求を出力します。また、TO0, TO1 端子から、トグル波形または矩形波を出力できます。

### ■ イベントカウントモード

タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のカウント許可ビット (CNTE:bit1) に "1" を設定している場合に、ソフトウェアトリガビット (TRG:bit0) に "1" を設定すると、16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) に設定した値が、16 ビットカウンタにリロードされます。TIN 端子に入力されたパルス (外部カウントクロック) の有効エッジ (立上り, 立下り, 両エッジから設定可能) を検出することにより、カウントします。

カウント許可ビット (CNTE:bit1) とソフトウェアトリガビット (TRG:bit0) 同時に "1" を設定した場合、カウントを許可にすると同時に、カウントを開始します。

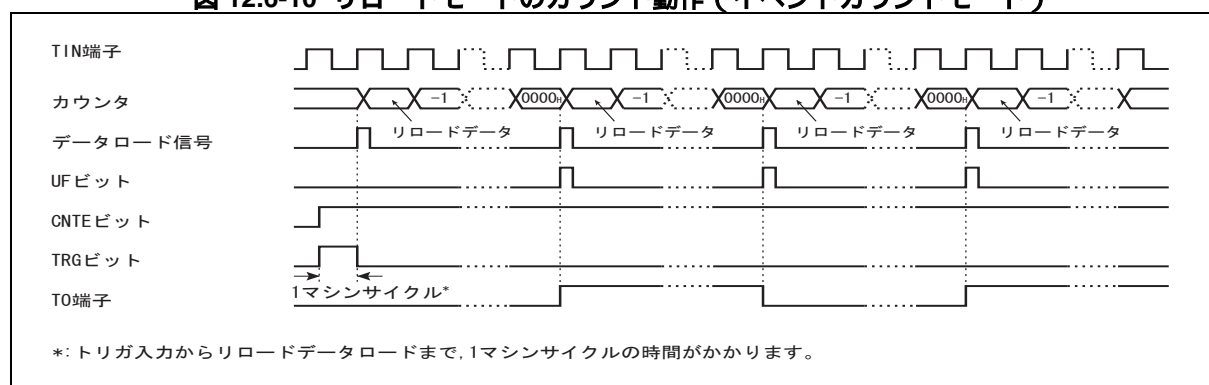
#### ● リロードモードの動作

16 ビットカウンタのアンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") 時は、16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) に設定した値が 16 ビットカウンタにリロードされ、カウントを継続します。

16 ビットカウンタのアンダフロー ("0000<sub>H</sub>" "FFFF<sub>H</sub>") 時は、タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のアンダフロー割込み要求フラグビット (UF:bit0) に "1" が設定され、アンダフロー割込み要求許可ビット (INTE:bit3) に "1" を設定している場合は、割込み要求を出力します。

TO0, TO1 端子からは、アンダフローごとに反転するトグル波形を出力できます。リロード時におけるカウント動作を図 12.6-10 に示します。

図 12.6-10 リロードモードのカウント動作 (イベントカウントモード)



< 注意事項 > TIN0, TIN1 端子へ入力されるパルスの "H" 幅および "L" 幅は、 $4/\phi$  ( $\phi$ : マシンクロック) 以上としてください。

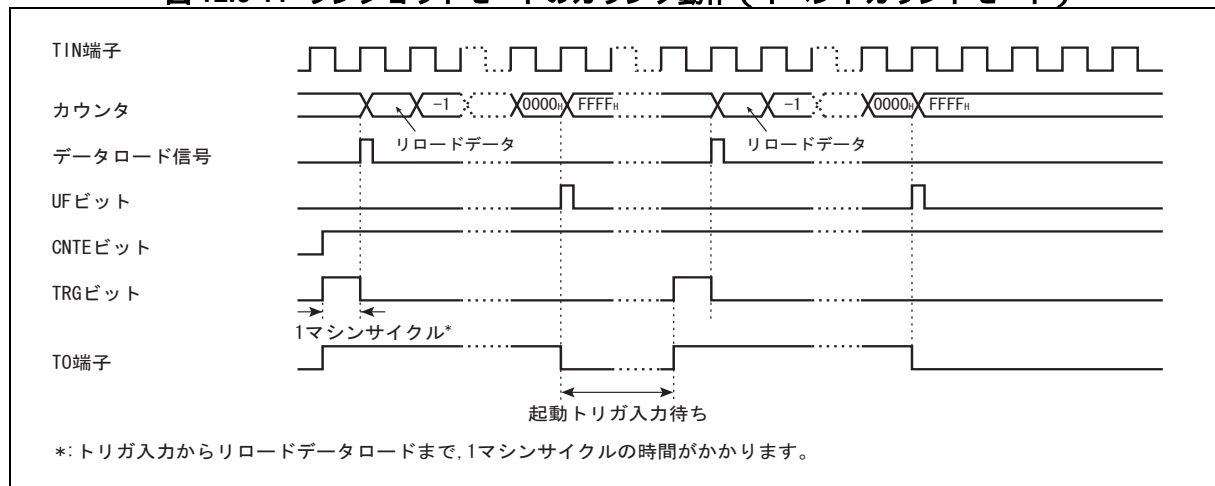
### ● ワンショットモードの動作

16 ビットカウンタのアンダフロー( "0000<sub>H</sub>" "FFFF<sub>H</sub>" )時は, 16 ビットカウンタの値は, "FFFF<sub>H</sub>" の状態でカウントを停止します。

タイマ制御状態レジスタ下位 ( TMCSRL0, TMCSRL1 ) のアンダフロー要求フラグビット ( UF:bit0 ) に "1" がセットされ, アンダフロー割込み要求許可ビット ( INTE:bit3 ) に "1" を設定している場合は, 割込み要求を出力します。

TO0, TO1 端子からは, カウント中を示す矩形波を出力できます。ワンショットモード時におけるカウント動作を図 12.6-11 に示します。

図 12.6-11 ワンショットモードのカウント動作 ( イベントカウントモード )



< 注意事項 > TIN0, TIN1 端子へ入力されるパルスの "H" 幅および "L" 幅は,  $4/\phi$  (  $\phi$ : マシンクロック ) 以上としてください。

## 12.7 16 ビットリロードタイマ使用上の注意

---

16 ビットリロードタイマ使用上の注意を以下に示します。

---

### ■ 16 ビットリロードタイマ使用上の注意

#### ● プログラム設定上の注意

- 16 ビットリロードレジスタ (TMRDH0, TMRDH1, TMRDL0, TMRDL1) に値を設定する場合は、カウントを停止 (TMCSRL0, TMCSRL1 レジスタの CNTE:bit1=0) させてください。16 ビットタイマレジスタ (TMRH0, TMRH1, TMRL0, TMRL1) を読み出す場合は、必ずワード転送命令 (MOVW A, dir) をご使用ください。
- タイマ制御状態レジスタ上位 (TMCSRH0, TMCSRH1) のカウントクロック選択ビット (CSL1, CSL0) を変更する場合は、カウントを停止 (TMCSRL0, TMCSRL1 レジスタの CNTE:bit1=0) してください。

#### ● 割込みに関する注意

- タイマ制御状態レジスタ下位 (TMCSRL0, TMCSRL1) のアンダフロー割込み要求フラグビット (UF:bit2) に "1" が設定され、アンダフロー割込み要求許可ビット (INTE:bit3) に "1" を設定している場合は、割込み処理から復帰できません。アンダフロー割込み要求フラグビット (UF:bit2) は、必ず "0" にクリアしてください。
- 16 ビットリロードタイマは、ほかのリソースと割込みベクタを共用しているため、同時に同じレベルの割込みを出力した場合は、割込み番号が小さい方 (優先度が高い) を優先します。EI<sup>2</sup>OS が 16 ビットリロードタイマによって使用される場合、共用するリソースの割込みは禁止でなければなりません。

## 12.8 16 ビットリロードタイマのプログラム例

16 ビットリロードタイマの内部クロックモードとイベントカウントモードのプログラム例を示します。

### ■ 内部クロックモードのプログラム例

#### ● 処理仕様

- 16 ビットリロードタイマ0により, 25msのインターバルタイマ割込みを生成します。
- タイマはリロードモード時において, 割込みの繰返し生成のために使用します。
- 外部トリガ入力を使用せず, ソフトウェアトリガにより 16 ビットリロードタイマを起動します。
- EI<sup>2</sup>OS は使用しません。
- 16MHz はマシクロック用, 2 $\mu$ s はカウントクロック用です。

#### ● コーディング例

```
ICR09 EQU 0000B9H ; 16 ビットリロードタイマ用割込み制御レジスタ
TMCSRL EQU 000082H ; タイマ制御状態レジスタ下位
TMRL EQU 000084H ; 16 ビットタイマレジスタ下位
TMRDL EQU 000084H ; 16 ビットリロードレジスタ下位
UF EQU TMCSSR:2 ; アンダフロー割込み要求フラグビット
CNTE EQU TMCSSR:1 ; カウント許可ビット
TRG EQU TMCSSR:0 ; ソフトウェアトリガビット

;----- メインプログラム -----
CODE CSEG
START:
; ; ; スタックポインタ (SP) は既に初期化されているもの
; ; ; と仮定します

AND CCR, #0BFH ; 割込み禁止
MOV I:ICR09, #00H ; 割込みレベル 0 (最強)
CLRB I:CNTE ; カウンタを一時停止
MOVW I:TMRD, #30D3H ; 25ms タイマのデータを設定
MOVW I:TMCSSR, #0000100000011011B
; ; ; インターバルタイマ動作, クロック 2 $\mu$ s
; ; ; 外部トリガ禁止, 外部出力禁止,
; ; ; リロードモード選択, 割込み許可
; ; ; 割込みフラグクリア, カウンタ開始

MOV ILM, #07H ; PS 内の ILM レジスタをレベル 7 に設定
OR CCR, #40H ; 割込み許可
LOOP: MOV A, #00H ; 無限プール
MOV A, #01H ;
BRA LOOP ;

;----- 割込みプログラム -----
WARI:
CLRB I:UF ; アンダフロー割込み要求フラグをクリア
; ; ;
```

```

; ユーザ処理
; :
; RETI ; 割り込みから復帰

CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
      ORG 0FF84H ; 割り込み #30 (1EH) のベクタを設定
      DSL WARI
      ORG 0FFDCH ; リセットベクタを設定
      DSL START
      DB 00H ; シングルチップモードを設定
VECT ENDS
      END START

```

## ■ イベントカウントモードのプログラム例

### ● 処理仕様

- 外部イベント入力端子に入力されるパルスの立上がりエッジを 16 ビットリロードタイマ / カウンタ 0 で 10,000 回カウントすると、割り込みを生成します。
- ワンショットモードで動作します。
- 外部トリガ入力は、立上がりエッジを設定します。
- EI<sup>2</sup>OS は使用しません。

### ● コーディング例

```

ICR09 EQU 0000B9H ; 16 ビットリロードタイマ用割り込み制御レジスタ
TMCSR EQU 000082H ; タイマ制御状態レジスタ下位
TMR EQU 000084H ; 16 ビットタイマレジスタ下位
TMRD EQU 000084H ; 16 ビットリロードレジスタ下位
DDR1 EQU 000011H ; ポート方向レジスタ
UF EQU TMCSR:2 ; アンダフロー割り込み要求フラグビット
CNTE EQU TMCSR:1 ; カウント許可ビット
TRG EQU TMCSR:0 ; ソフトウェアトリガビット
;----- メインプログラム -----
CODE CSEG
START:
; : ; スタックポインタ (SP) は既に初期化されているものと
; 仮定します
AND CCR, #0BFH ; 割り込み禁止
MOV I:ICR09, #00H ; 割り込みレベル 0 (最強)
MOV I:DDR1, #00H ; INT5 と TIN0 端子を入力に設定
CLRB I:CNTE ; カウンタを一時停止
MOVW I:TMRD, #2710H ; リロード値 10,000 回の設定
MOVW I:TMCSR, #0000110010001011B
; カウンタ動作, 外部イベント入力, 立上りエッジ,
; 外部出力禁止
; ワンショットモード設定, 割り込み許可
; 割り込みフラグクリア, カウンタ開始
MOV ILM, #07H ; PS 内の ILM レジスタをレベル 7 に設定
OR CCR, #40H ; 割り込み許可
LOOP: MOV A, #00H ; 無限ループ
      MOV A, #01H ;
      BRA LOOP

```

```

;----- 割込みプログラム -----
WARI:
    CLRB  I:UF          ; アンダフロー割込み要求フラグをクリア
;   :
;   ユーザ処理
;
    RETI                ; 割込みから復帰

CODE  ENDS

;----- ベクタ設定 -----
VECT  CSEG  ABS=0FFH
      ORG   0FF84H      ; 割込み #30 (1EH) のベクタを設定
      DSL   WARI
      ORG   0FFDCH      ; リセットベクタを設定
      DSL   START
      DB    00H         ; シングルチップモードを設定
VECT  ENDS
      END    START

```



# 第13章

---

## 16 ビット PPG タイマ

16 ビット PPG タイマの機能と動作について説明します (MB90467 では, PPG1 タイマ ch.1 未搭載となります)。

- 13.1 16 ビット PPG タイマの概要
- 13.2 16 ビット PPG タイマのブロックダイアグラム
- 13.3 16 ビット PPG タイマの端子
- 13.4 16 ビット PPG タイマのレジスタ
- 13.5 16 ビット PPG タイマ割込み
- 13.6 16 ビット PPG タイマの動作
- 13.7 16 ビット PPG タイマ使用上の注意
- 13.8 16 ビット PPG タイマのプログラム例



## 13.1 16 ビット PPG タイマの概要

---

16 ビット PPG タイマは、16 ビットダウンカウンタ、プリスケアラ、16 ビット周期設定バッファレジスタ、16 ビットデューティ設定バッファレジスタ、16 ビット制御レジスタおよび PPG 出力端子から構成されています。

---

### ■ 16 ビット PPG タイマ ( × 3 )

16 ビット PPG タイマは、16 ビットダウンカウンタ、プリスケアラ、16 ビット周期設定バッファレジスタ、16 ビットデューティ設定バッファレジスタ、16 ビット制御レジスタおよび PPG 出力端子から構成されています。このモジュールを使用すると、多機能タイマからのソフトウェアトリガまたは GATE 信号で同期化されたパルスを出力できます。多機能タイマについての詳細は、「第 14 章 多機能タイマ」をご参照ください。

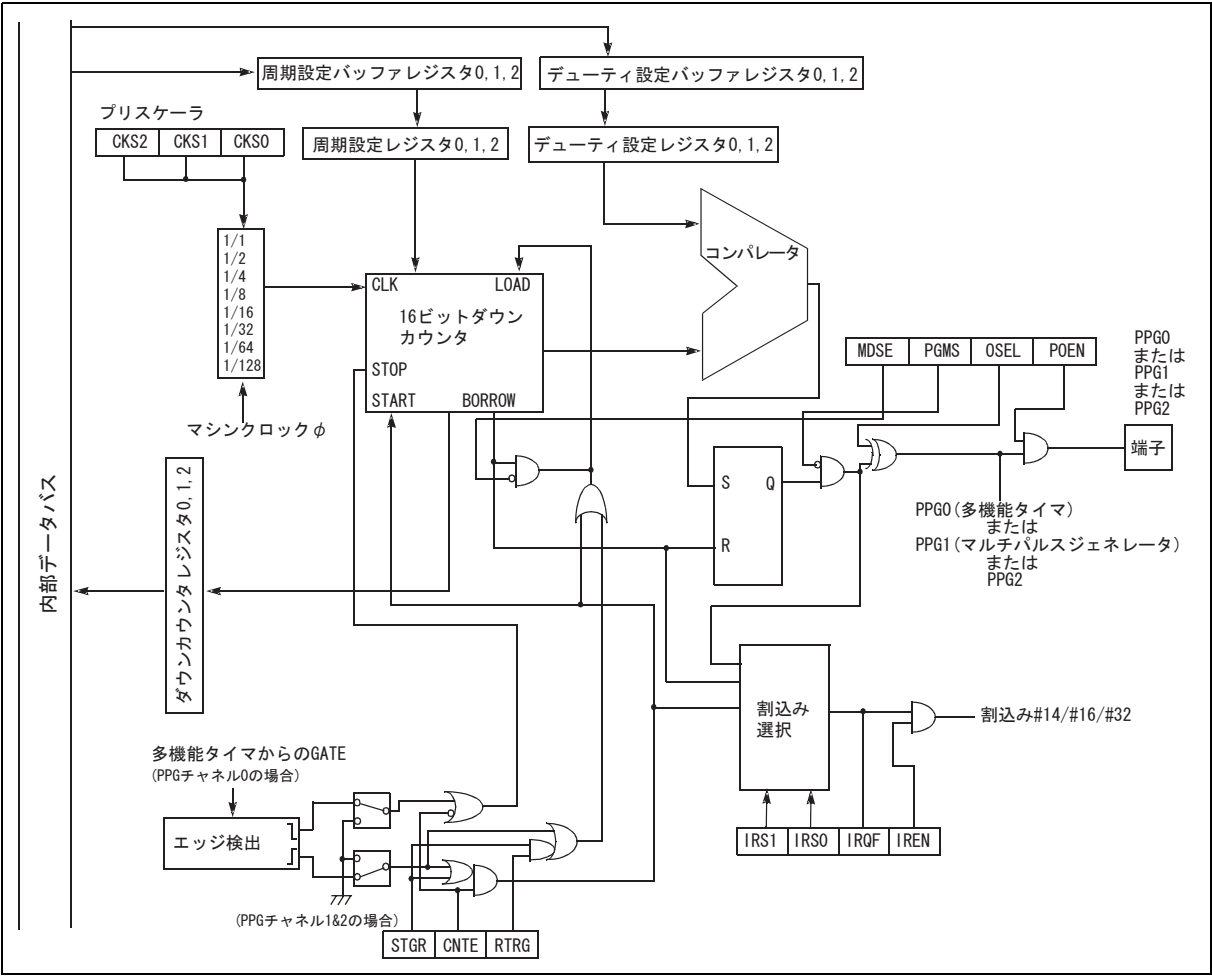
- カウンタ動作クロックは、8 種類 (  $\phi$ ,  $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ ,  $\phi/32$ ,  $\phi/64$ ,  $\phi/128$  ) の中から選択できます (  $\phi$  はマシクロック )。
- 割込みは、トリガまたはカウンタボローが発生した場合または PPG 立上り ( 通常極性 ) が発生した場合または PPG 立下り ( 反転極性 ) が発生した場合に生成されます。
- PPG 出力動作 : 16 ビット PPG タイマは、周期およびデューティ比が可変のパルス波形を出力できます。また、16 ビット PPG タイマは、外付け回路により D/A コンバータとしても使用できます。

## 13.2 16 ビット PPG タイマのブロックダイアグラム

16 ビット PPG タイマのブロックダイアグラムを示します。

### ■ 16 ビット PPG タイマのブロックダイアグラム

図 13.2-1 16 ビット PPG タイマのブロックダイアグラム



## 13.3 16 ビット PPG タイマの端子

16 ビット PPG タイマの端子について説明します。また端子のブロックダイアグラムも示します。

### ■ 16 ビット PPG タイマの端子

16 ビット PPG タイマの端子は、汎用入出力ポートと共用されます。端子の機能や I/O 形式、端子の設定（16 ビット PPG タイマを使用するために必要な設定）を表 13.3-1 に示します。

表 13.3-1 16 ビット PPG タイマの端子

端子名	端子の機能	I/O 形式	スタンバイ制御	端子設定
P37/PPG0	ポート 3 入出力, PPG0 出力	CMOS 出力または CMOS 入力	使用可能	PPG0 タイマ出力設定 (PNCTL0:POEN=1)
P36/ PPG1*	ポート 3 入出力, PPG1 出力			PPG1 タイマ出力許可設定許可 (PNCTL1:POEN=1)
P46/PPG2	ポート 4 入出力, PPG2 出力	CMOS 出力, CMOS ヒ ステリシス入力		PPG2 タイマ出力許可設定許可 (PNCTL2:POEN=1)

\*:MB90467 では、リソースが搭載されていないため、削除となります。

■ 16 ビット PPG タイマ端子のブロックダイヤグラム

図 13.3-1 16 ビット PPG0, PPG1 タイマ端子のブロックダイヤグラム

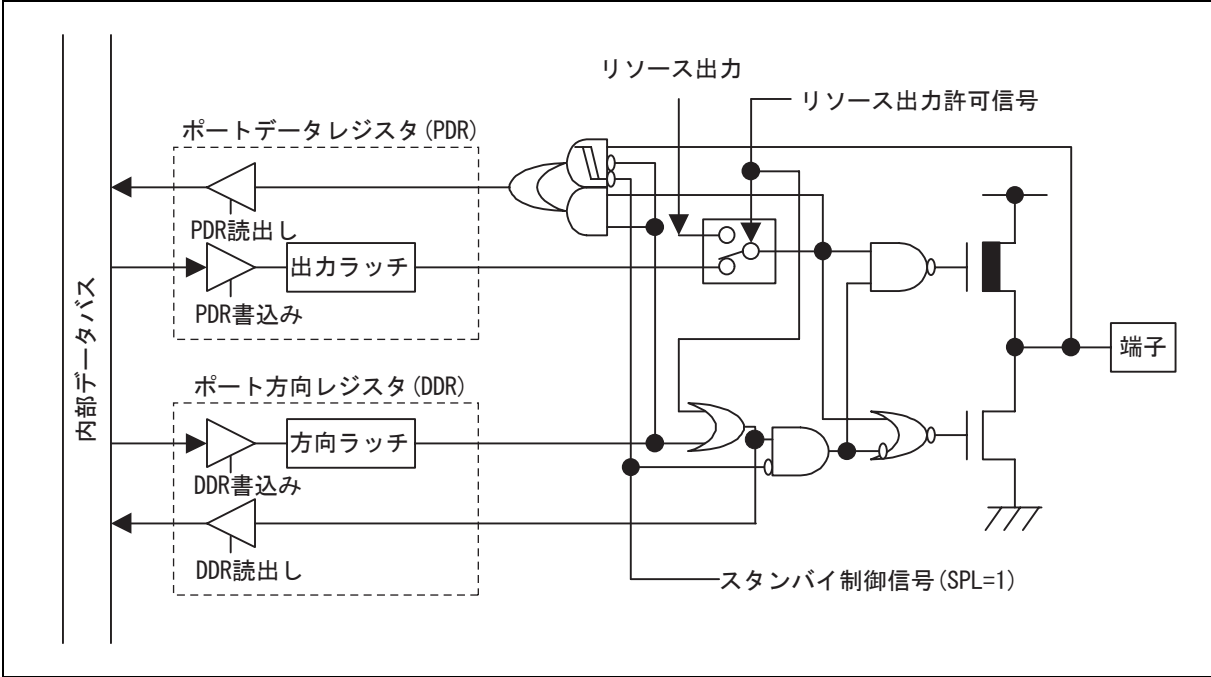
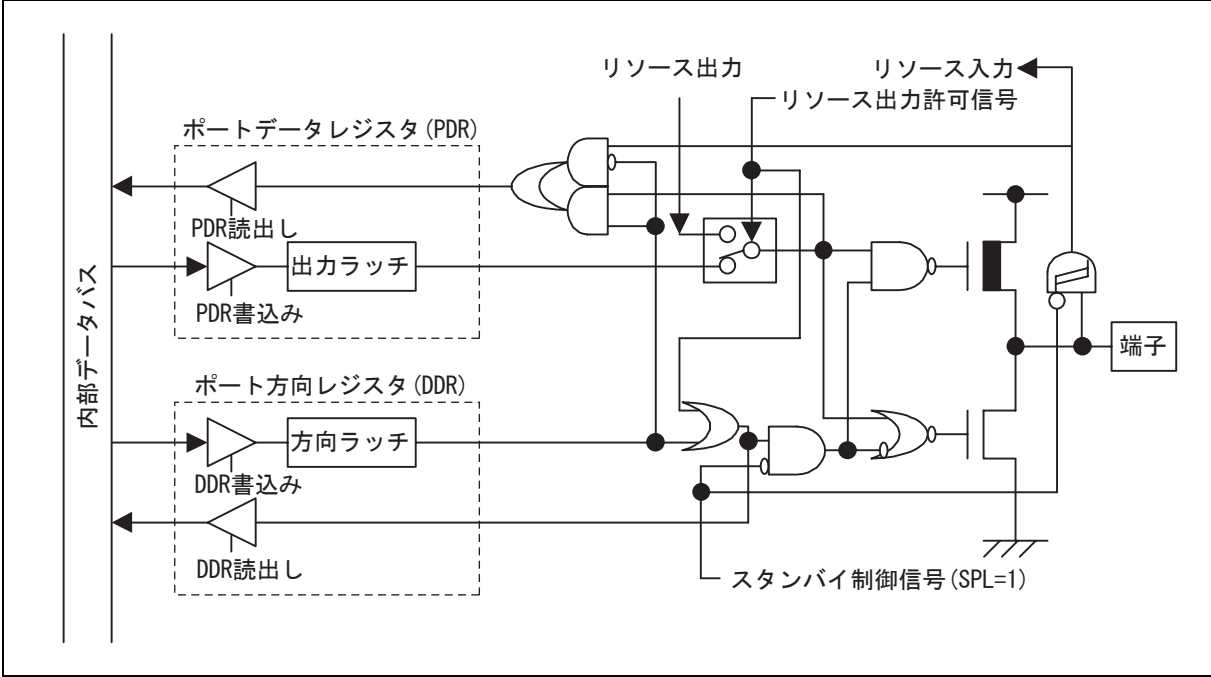


図 13.3-2 16 ビット PPG2 タイマ端子のブロックダイヤグラム



## 13.4 16 ビット PPG タイマのレジスタ

16 ビット PPG タイマのレジスタ一覧を示します。

### ■ 16 ビット PPG タイマのレジスタ

図 13.4-1 16 ビット PPG タイマのレジスタ

PPG ダウンカウンタレジスタ(上位)									
	bit	15	14	13	12	11	10	9	8
		DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08
		PDCRH0～PDCRH2							
リード/ライト	⇒	R	R	R	R	R	R	R	R
初期値	⇒	1	1	1	1	1	1	1	1
PPG ダウンカウンタレジスタ(下位)									
	bit	7	6	5	4	3	2	1	0
		DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00
		PDCRL0～PDCRL2							
リード/ライト	⇒	R	R	R	R	R	R	R	R
初期値	⇒	1	1	1	1	1	1	1	1
PPG 周期設定バッファレジスタ(上位)									
	bit	15	14	13	12	11	10	9	8
		CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08
		PCSRH0～PCSRH2							
リード/ライト	⇒	W	W	W	W	W	W	W	W
初期値	⇒	X	X	X	X	X	X	X	X
PPG 周期設定バッファレジスタ(下位)									
	bit	7	6	5	4	3	2	1	0
		CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00
		PC SRL0～PC SRL2							
リード/ライト	⇒	W	W	W	W	W	W	W	W
初期値	⇒	X	X	X	X	X	X	X	X
PPG デューティ設定バッファレジスタ(上位)									
	bit	15	14	13	12	11	10	9	8
		DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08
		PDUTH0～PDUTH2							
リード/ライト	⇒	W	W	W	W	W	W	W	W
初期値	⇒	X	X	X	X	X	X	X	X
PPG デューティ設定バッファレジスタ(下位)									
	bit	7	6	5	4	3	2	1	0
		DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00
		PDUTL0～PDUTL2							
リード/ライト	⇒	W	W	W	W	W	W	W	W
初期値	⇒	X	X	X	X	X	X	X	X

( 続く )

( 続き )

PPG 状態制御レジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	CNTE	STGR	MDSE	RTRG	CKS2	CKS1	CKS0	PGMS	PCNTH0~PCNTH2
リード/ライト	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

PPG 状態制御レジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	—	—	IREN	IRQF	IRS1	IRS0	POEN	OSEL	PCNTL0~PCNTL2
リード/ライト	⇒ -	⇒ -	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ -	⇒ -	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

### 13.4.1 PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2)

PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2) は, 16 ビット PPG ダウンカウンタのカウンタ値を読み出すために使用する 16 ビットレジスタです。

#### ■ PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2)

図 13.4-2 PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2)

図 10-12 PPG ダウンカウンタレジスタ (PDCRH, PDCRL0 ~ PDCRL2)

PPG ダウンカウンタレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	DC15	DC14	DC13	DC12	DC11	DC10	DC09	DC08	PDCRH0~PDCRH2
リード/ライト	⇒ R	R	R	R	R	R	R	R	
初期値	⇒ 1	1	1	1	1	1	1	1	

PPG ダウンカウンタレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	DC07	DC06	DC05	DC04	DC03	DC02	DC01	DC00	PDCRL0~PDCRL2
リード/ライト	⇒ R	R	R	R	R	R	R	R	
初期値	⇒ 1	1	1	1	1	1	1	1	

これらのレジスタは, 16 ビットダウンカウンタの値を格納するために使用する 16 ビットレジスタです。これらのレジスタの初期値は, すべて "1" です。これらのレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。これらのレジスタは, 読み出し専用レジスタです。

### 13.4.2 PPG 周期設定バッファレジスタ (PCSRH, PCSRL0 ~ PCSRL2)

PPG 周期設定バッファレジスタは、PPG で生成される出力パルスの周期を設定するために使用します。

■ PPG 周期設定バッファレジスタ (PCSRH, PCSRL0 ~ PCSRL2)

図 13.4-3 PPG 周期設定バッファレジスタ (PCSRH, CSRL0 ~ CSRL2)

PPG 周波数設定バッファレジスタ(上位)									
	bit 15	14	13	12	11	10	9	8	
	CS15	CS14	CS13	CS12	CS11	CS10	CS09	CS08	PCSRH0~PCSRH2
リード/ライト	⇒ W	W	W	W	W	W	W	W	
初期値	⇒ X	X	X	X	X	X	X	X	
PPG 周波数設定バッファレジスタ(下位)									
	bit 7	6	5	4	3	2	1	0	
	CS07	CS06	CS05	CS04	CS03	CS02	CS01	CS00	PCSRL0~PCSRL2
リード/ライト	⇒ W	W	W	W	W	W	W	W	
初期値	⇒ X	X	X	X	X	X	X	X	

これらのレジスタは、PPG で生成される出力パルスの周期を設定するために使用する 16 ビットレジスタです。これらのレジスタの初期値は不定なので、動作を開始する前にレジスタに値を設定しなければなりません。これらのレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。これらのレジスタは、書込み専用レジスタです。

PPG 周期設定バッファレジスタから周期設定レジスタへのデータ転送は、カウンタロー、トリガまたは再トリガが発生した場合に実行されます。

PPG 周期設定バッファレジスタを更新する場合は、この PPG 周期設定バッファレジスタを更新した後に、この更新値をデューティ設定バッファレジスタに書き込まなければなりません。すなわち、PPG 周期設定バッファレジスタのみの更新は禁止されています。



### 13.4.3 PPG デューティ設定バッファレジスタ (PDUTH, PDUTL0 ~ PDUTL2)

PPG デューティ設定バッファレジスタは、PPG で生成される出力パルスのデューティ比を制御するために使用します。

#### ■ PPG デューティ設定バッファレジスタ (PDUTH, PDUTL0 ~ PDUTL2)

図 13.4-4 PPG デューティ設定バッファレジスタ (PDUTH, PDUTL0 ~ PDUTL2)

図 10-7 PPG デューティ設定バッファレジスタ (PDUTH0, PDUTH1, PDUTH2)

PPG デューティ設定バッファレジスタ(上位)

bit	15	14	13	12	11	10	9	8	
	DU15	DU14	DU13	DU12	DU11	DU10	DU09	DU08	PDUTH0~PDUTH2
リード/ライト	⇒ W	W	W	W	W	W	W	W	
初期値	⇒ X	X	X	X	X	X	X	X	

PPG デューティ設定バッファレジスタ(下位)

bit	7	6	5	4	3	2	1	0	
	DU07	DU06	DU05	DU04	DU03	DU02	DU01	DU00	PDUTL0~PDUTL2
リード/ライト	⇒ W	W	W	W	W	W	W	W	
初期値	⇒ X	X	X	X	X	X	X	X	

これらのレジスタは、PPG で生成される出力パルスのデューティ比を制御するために使用する 16 ビットレジスタです。これらのレジスタの初期値は不定なので、動作を開始する前にレジスタに値を設定しなければなりません。これらのレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。これらのレジスタは、書込み専用レジスタです。

PPG デューティ設定バッファレジスタからデューティ設定レジスタへのデータ転送は、カウンタボロー、トリガまたは再トリガが発生した場合に実行されます。

PPG 周期設定レジスタとデューティ設定レジスタに同じ値を設定すると、通常極性の場合はすべて "H" を出力し、反転極性の場合は全て "L" を出力します。

PPG の出力は、"PCSR < PDUT" の場合は不定です。

PPG 周期設定バッファレジスタを更新しない場合は、PPG デューティ設定バッファレジスタへの書込みは可能です。

13.4.4 PPG 状態制御レジスタ (PCNTH, PCNTL0 ~ PCNTL2)

PPG 状態制御レジスタは, 16 ビット PPG タイマの許可/禁止, ソフトウェアトリガ, 再トリガ制御割込みおよび出力極性に関する動作状態を設定するために使用します。また状態チェックのためにも使用します。

■ PPG 状態制御レジスタ, 上位バイト (PCNTH0 ~ PCNTH2)

図 13.4-5 PPG 状態制御レジスタ, 上位 (PCNTH0 ~ PCNTH2)

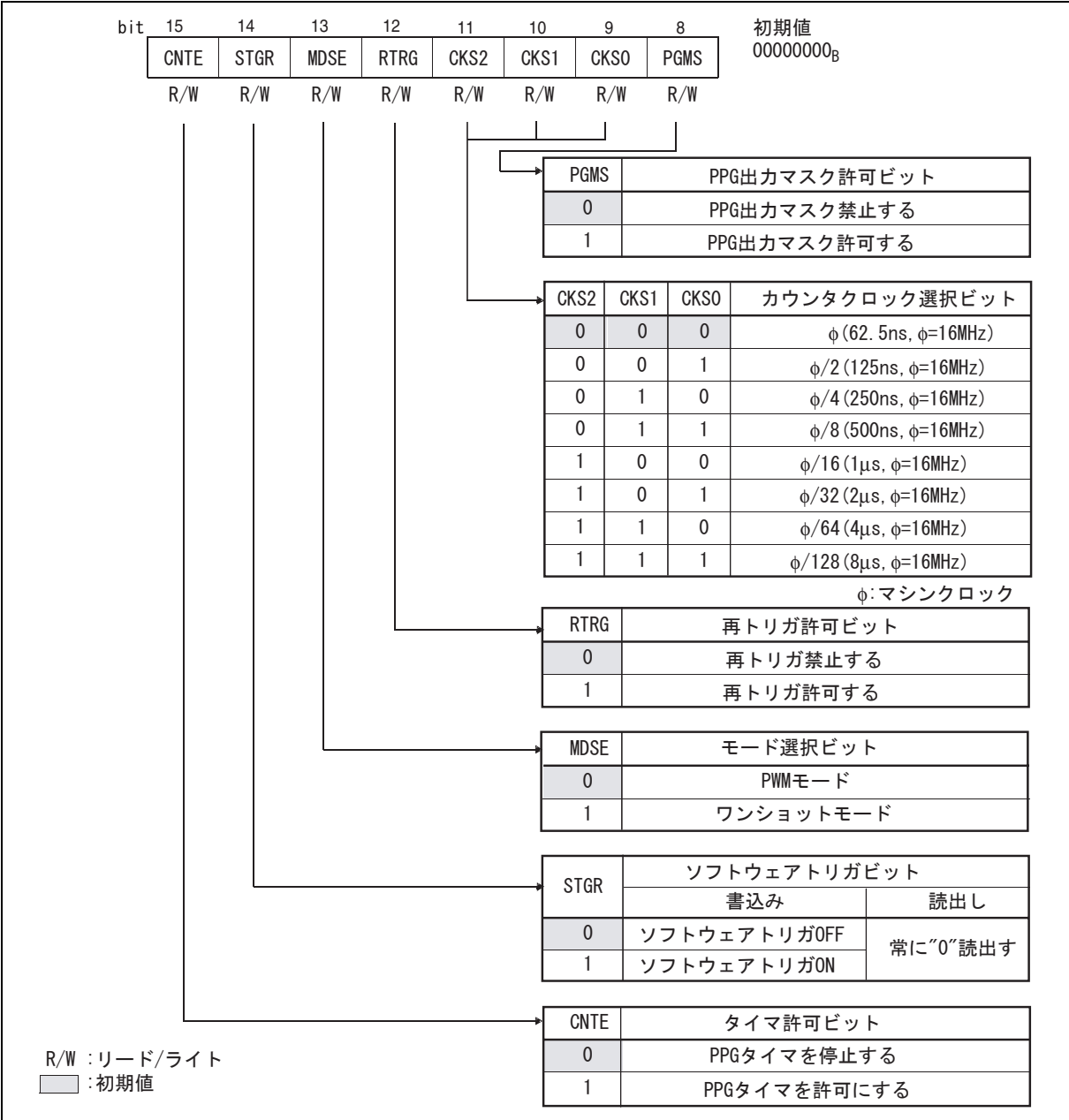


表 13.4-1 PPG 状態制御レジスタ, 上位 (PCNTH0 ~ PCNTH2)

ビット名		機 能
bit15	CNTE: タイマ許可 ビット	<ul style="list-style-type: none"> <li>このビットは, PPG タイマ動作を許可するために使用します。</li> <li>このビットに "0" を設定した場合: 動作が停止します。</li> <li>このビットに "1" を設定した場合: PPG 動作が "許可" になり, PPG 動作がトリガによって開始されますのを待つ状態になります。</li> </ul>
bit14	STGR: ソフトウェア トリガビット	<ul style="list-style-type: none"> <li>このビットは, PPG をソフトウェアで起動するために使用します。</li> <li>このビットに "1" を設定した場合: PPG がソフトウェアで起動します。</li> <li>このビットからは必ず "0" が読み出されます。</li> </ul>
bit13	MDSE: モード選択 ビット	<ul style="list-style-type: none"> <li>このビットに "0" を設定した場合: PPG は PWM モードで動作します。</li> <li>このビットに "1" を設定した場合: PPG はワンショットモードで動作します。</li> </ul>
bit12	RTRG: 再トリガ許可 ビット	<ul style="list-style-type: none"> <li>このビットは, 動作中に PPG の再トリガ機能を許可するために使用します。</li> <li>このビットに "0" を設定した場合: 再トリガ機能は "禁止" になります。</li> <li>このビットに "1" を設定した場合: 再トリガ機能は "許可" になります。</li> </ul>
bit11 ~ bit9	CKS2, CKS1, CKS0: カウンタク ロック選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, 16 ビット PPG タイマの動作クロックを選択するために使用します。</li> </ul>
bit8	PGMS: PPG 出力マス ク許可ビット	<ul style="list-style-type: none"> <li>このビットは, モード設定 (PCNTH0 ~ PCNTH2 レジスタの MDSE:bit13), 周期設定 (PCSRH, L) またはデューティ設定 (PDUTH, L) とは無関係に特定レベルへの PPG 出力をマスクするために使用します。</li> <li>このビットに "0" を設定した場合: PPG 出力マスク機能は "禁止" になります。</li> <li>このビットに "1" を設定した場合: PPG 出力は, 極性設定が "通常" (PCNTL レジスタ OSEL:bit=0) の場合は必ず "L" がマスクされます。</li> <li>このビットに "1" を設定した場合: PPG 出力は, 極性設定が "反転" (PCNTL レジスタ OSEL:bit=0) の場合は必ず "H" がマスクされます。</li> </ul> <p>(注意事項)</p> <p>PPG 周期設定バッファレジスタ (PCSRH, L) と PPG デューティ設定バッファレジスタ (PDUTH, L) に同じ値を設定すると, このビットが "1" 場合で, かつ通常極性の場合はオール "H" を出力でき, 反転極性の場合はオール "L" を出力できます。</p>

■ PPG 状態制御レジスタ , 下位バイト ( PCNTL0 ~ PCNTL2 )

図 13.4-6 PPG 状態制御レジスタ , 下位 ( PCNTL0 ~ PCNTL2 )

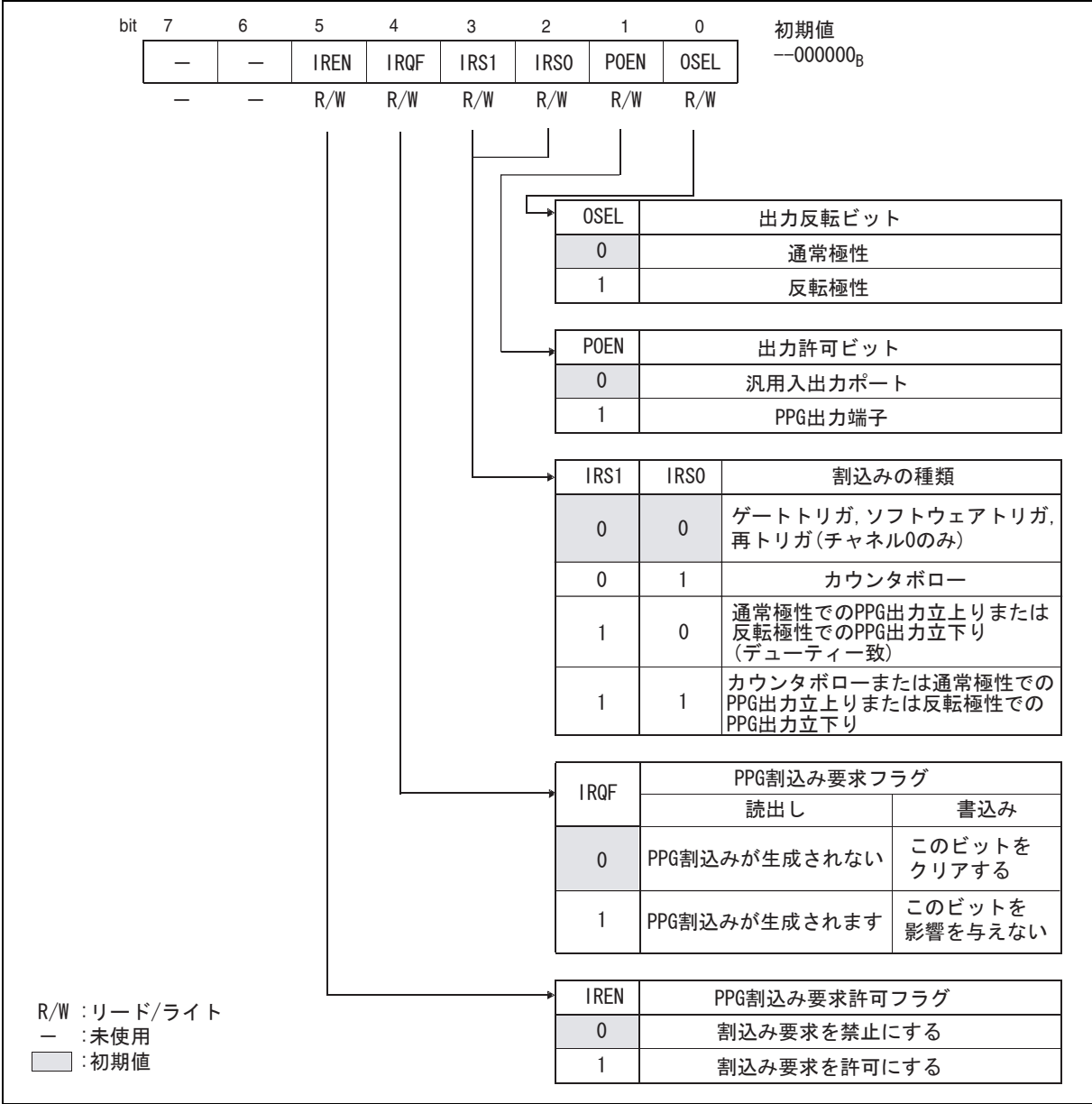


表 13.4-2 PPG 状態制御レジスタ, 下位 (PCNTL0 ~ PCNTL2)

ビット名		機 能
bit7, bit6	未使用ビット	<ul style="list-style-type: none"> <li>読出し値は不定です。</li> <li>このビットへの書き込みは動作へ影響しません。</li> </ul>
bit5	IREN: PPG 割込み要求許可ビット	<ul style="list-style-type: none"> <li>このビットは, CPU への PPG 割込み要求を許可または禁止します。</li> <li>このビットと割込みフラグ (IRQF) ビットが "1" の場合, PPG は割込み要求を出力します。</li> </ul>
bit4	IRQF: PPG 割込みフラグビット	<ul style="list-style-type: none"> <li>このビットは, PPG 割込みが発生すると "1" に設定されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト動作時は必ず "1" が読み出されます。</li> <li>このビットは, EI<sup>2</sup>OS がアクティブになった場合にもクリアされます。</li> </ul>
bit3, bit2	IRS1, IRS0: 割込み選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, PPG タイマの割込み動作を選択するために使用します。</li> </ul>
bit1	POEN: 出力許可ビット	<ul style="list-style-type: none"> <li>このビットは, PPG 出力端子からの出力を許可または禁止します。</li> <li>このビットに "0" を設定した場合: 端子は汎用ポートとして機能します。</li> <li>このビットに "1" を設定した場合: 端子は PPG タイマ出力端子として機能します。</li> </ul>
bit0	OSEL: 出力反転ビット	<ul style="list-style-type: none"> <li>このビットは, PPG 出力端子の極性を選択します。</li> <li>このビットに "0" を設定した場合: 通常極性が選択されます。PPG は, 16 ビットダウンカウンタ値が PDUTH, L レジスタよりも大きい場合は "L" を出力し, 16 ビットダウンカウンタ値が PDUTH, L レジスタ以下の場合は "H" を出力します。</li> <li>このビットに "1" を設定した場合: PPG 出力は反転します。</li> </ul>

## 13.5 16 ビット PPG タイマ割込み

16 ビット PPG タイマは、以下の場合に割込み要求を生成できます。

- トリガまたはカウンタポローが発生した場合
- 通常極性で PPG 立上りが発生した場合
- 反転極性で PPG 立下りが発生した場合

これらの割込み動作は PCNTL0 ~ PCNTL2 レジスタの IRS1:bit3 と、IRS0:bit2 で設定されます。また 16 ビット PPG タイマは、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) とも関係しています。

### ■ 16 ビット PPG タイマ割込み

16 ビット PPG タイマの割込み制御ビットと割込み要因を表 13.5-1 に示します。

表 13.5-1 16 ビット PPG タイマの割込み制御ビットと割込み要因

	16 ビット PPG タイマ 0	16 ビット PPG タイマ 1	16 ビット PPG タイマ 2
割込みフラグビット	PCNTL0:IRQF ( bit4 )	PCNTL1:IRQF ( bit4 )	PCNTL2:IRQF ( bit4 )
割込み要求許可ビット	PCNTL0:IREN ( bit5 )	PCNTL1:IREN ( bit5 )	PCNTL2:IREN ( bit5 )
割込み種類選択ビット	PCNTL0:IRS1, IRS0 ( bit3, bit2 )	PCNTL1:IRS1, IRS0 ( bit3, bit2 )	PCNTL2:IRS1, IRS0 ( bit3, bit2 )
割込み要因	PCNTL0:IRS1, IRS0=00 16 ビットダウンカウンタ ( ch.0 ) のゲートトリガ、 ソフトウェアトリガ、再 トリガ	PCNTL1:IRS1, IRS0=00 16 ビットダウンカウンタ ( ch.1 ) のソフトウェアト リガ、再トリガ	PCNTL2:IRS1, IRS0=00 16 ビットダウンカウンタ ( ch.2 ) のソフトウェアト リガ、再トリガ
	PCNTL0:IRS1, IRS0=00 16 ビットダウンカウンタ ( ch.0 ) のカウンタポロー	PCNTL1:IRS1, IRS0=01 16 ビットダウンカウンタ ( ch.1 ) のカウンタポロー	PCNTL2:IRS1, IRS0=00 16 ビットダウンカウンタ ( ch.2 ) のカウンタポロー
	PCNTL0:IRS1, IRS0=00 通常極性での PPG0 出力 立上りまたは反転極性で の PPG0 出力立下り	PCNTL1:IRS1, IRS0=00 通常極性での PPG1 出力 立上りまたは反転極性で の PPG1 出力立下り	PCNTL2:IRS1, IRS0=00 通常極性での PPG2 出力 立上りまたは反転極性で の PPG2 出力立下り
	16 ビットダウンカウンタ ( ch.0 ) のカウンタポロー または通常極性での PPG0 出力立上りまたは反転極 性での PPG0 出力立下り	16 ビットダウンカウンタ ( ch.1 ) のカウンタポロー または通常極性での PPG1 出力立上りまたは反転極 性での PPG1 出力立下り	16 ビットダウンカウンタ ( ch.2 ) のカウンタポロー または通常極性での PPG2 出力立上りまたは反転極 性での PPG2 出力立下り

16 ビット PPG タイマでは、PPG 状態制御レジスタ ( PCNTL ) の IRQF:bit4 に "1" が設定され、割込み要求が許可されると ( PCNTL レジスタの IREN:bit5=1 )、割込み要求が割込みコントローラへ出力されます。

## ■ 16 ビット PPG タイマ割込みと EI<sup>2</sup>OS

16 ビット PPG タイマ割込みと EI<sup>2</sup>OS を表 13.5-2 に示します。

表 13.5-2 16 ビット PPG タイマ割込みと EI<sup>2</sup>OS

チャンネル	割込み 番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ 名	アドレス	下位	中位	上位	
16 ビット PPG タイマ 0 <sup>*1</sup>	#14 (0E <sub>H</sub> )	ICR01	0000B1 <sub>H</sub>	FFFFC4 <sub>H</sub>	FFFFC5 <sub>H</sub>	FFFFC6 <sub>H</sub>	
16 ビット PPG タイマ 1 <sup>*2</sup>	#16 (10 <sub>H</sub> )	ICR02	0000B2 <sub>H</sub>	FFFFBC <sub>H</sub>	FFFFBD <sub>H</sub>	FFFFBE <sub>H</sub>	
16 ビット PPG タイマ 2 <sup>*3</sup>	#32 (20 <sub>H</sub> )	ICR10	0000BA <sub>H</sub>	FFFF7C <sub>H</sub>	FFFF7D <sub>H</sub>	FFFF7E <sub>H</sub>	

\*1 :16 ビット PPG タイマ 0 に割当てられた割込み制御レジスタと同じレジスタが PWC タイマ 0 に割当てられます。

\*2 :16 ビット PPG タイマ 1 に割当てられた割込み制御レジスタと同じレジスタが 16 ビットアウトプットコンペアチャンネル 1 一致に割当てられます。

\*3 :16 ビット PPG タイマ 2 に割当てられた割込み制御レジスタと同じレジスタが 16 ビットフリーランタイム 0 検出に割当てられます。

## ■ 16 ビット PPG タイマの EI<sup>2</sup>OS 機能

16 ビット PPG タイマは EI<sup>2</sup>OS と関係する回路を持っているので、カウンタは、PPG 割込みが発生すると EI<sup>2</sup>OS を起動できます。

ただし、EI<sup>2</sup>OS は、割込み制御レジスタ (ICR) を共用するほかのリソースが割込みを使用しない場合に限り使用可能です。たとえば、16 ビット PPG タイマ 0 が EI<sup>2</sup>OS を使用している場合、アウトプットコンペアチャンネル 0 一致の割込みは、禁止にしなければなりません。

## 13.6 16 ビット PPG タイマの動作

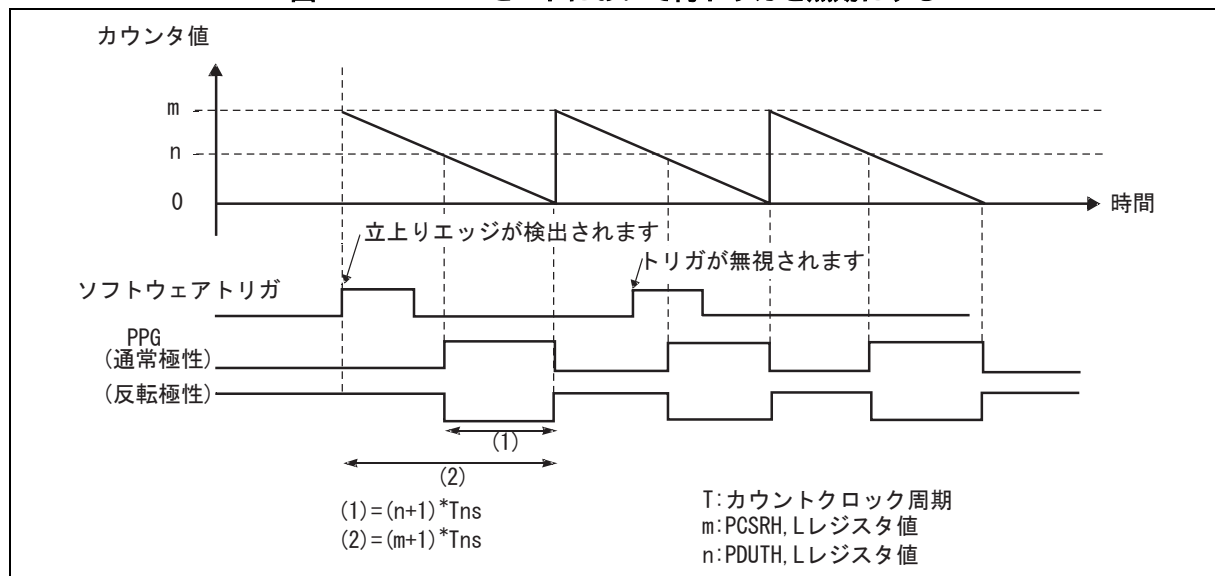
16 ビット PPG タイマは、PWM モードまたはワンショットモードで動作します。また、16 ビット PPG タイマでは再トリガを使用することもできます。

### ■ PWM モード (PCNTH レジスタの MDSE:bit13=0)

PWM 動作の場合、16 ビットダウンカウンタは、PCSR 値がロードされ、有効なトリガが検出された後、カウントを開始します。16 ビットダウンカウンタは、“0” に達したら、PCSR 値がリロードされ、カウントを反復します。PPG 出力は、16 ビットダウンカウンタに PCSR 値がリロードされると、切り換わります。出力パルスの周期は PCSRH, PCSRL レジスタを設定することで制御でき、デューティ比は PDUTH, PDUTL レジスタを設定することで制御できます。

#### ● 再トリガを無効にする (PCNTH レジスタの RTRG:bit12=0)

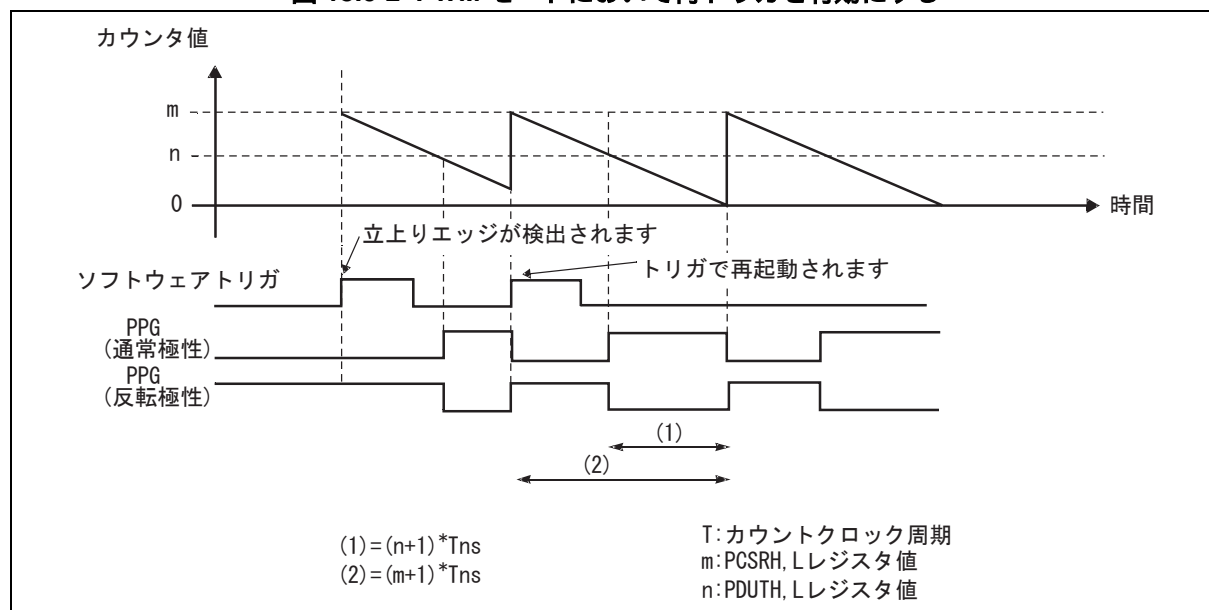
図 13.6-1 PWM モードにおいて再トリガを無効にする





● 再トリガを有効にする (PCNTH レジスタの RTRG:bit12=1)

図 13.6-2 PWM モードにおいて再トリガを有効にする

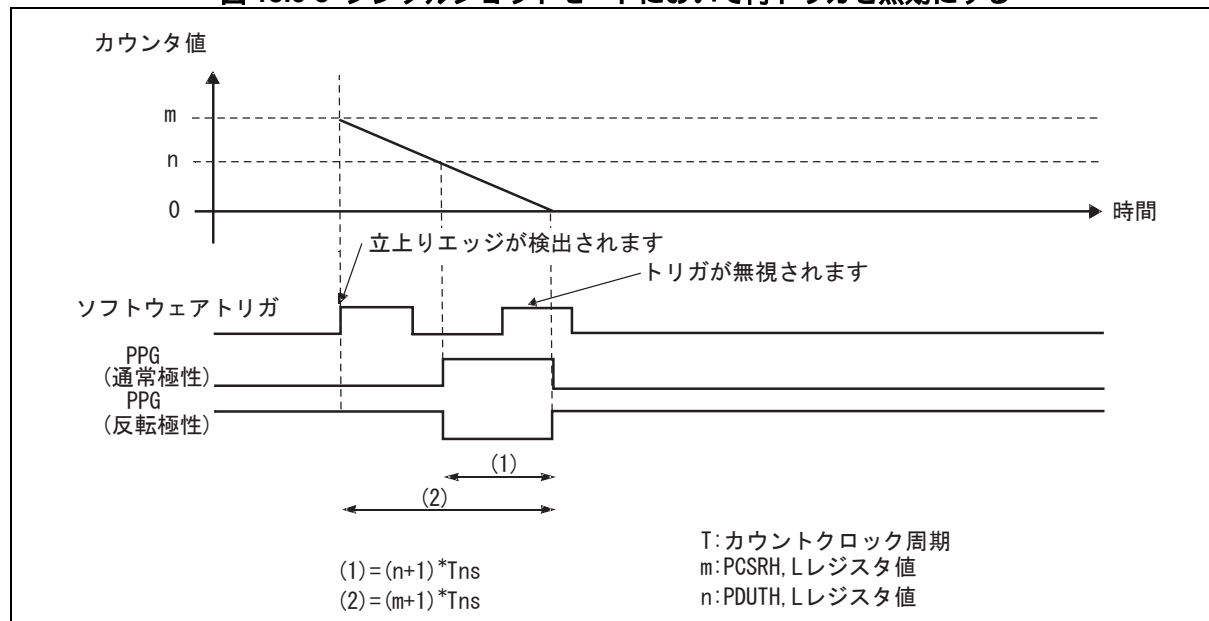


■ ワンショットモード (PCNTH レジスタの MDSE:bit13=1)

ワンショット動作の場合, 指定された幅の単一パルスを有効なトリガで出力できます。再トリガを有効にすると, 動作中にエッジが検出された場合, カウンタに値がリロードされます。

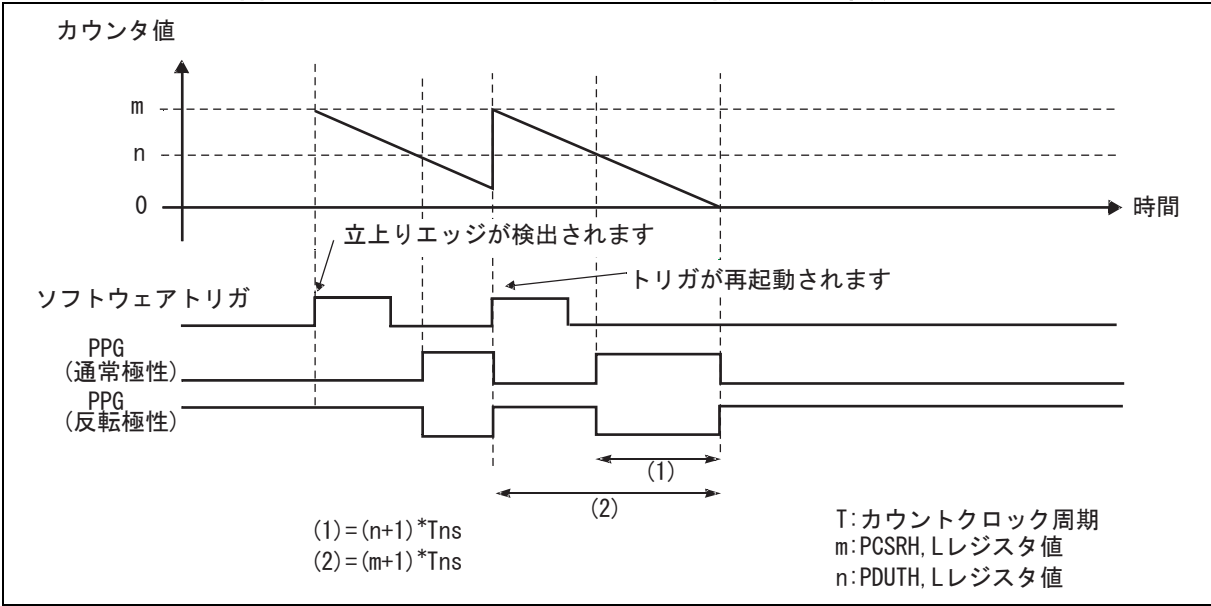
● 再トリガを無効にする (PCNTH レジスタの RTRG:bit12=0)

図 13.6-3 シングルショットモードにおいて再トリガを無効にする



● 再トリガを有効にする (PCNTH レジスタの RTRG:bit12=1)

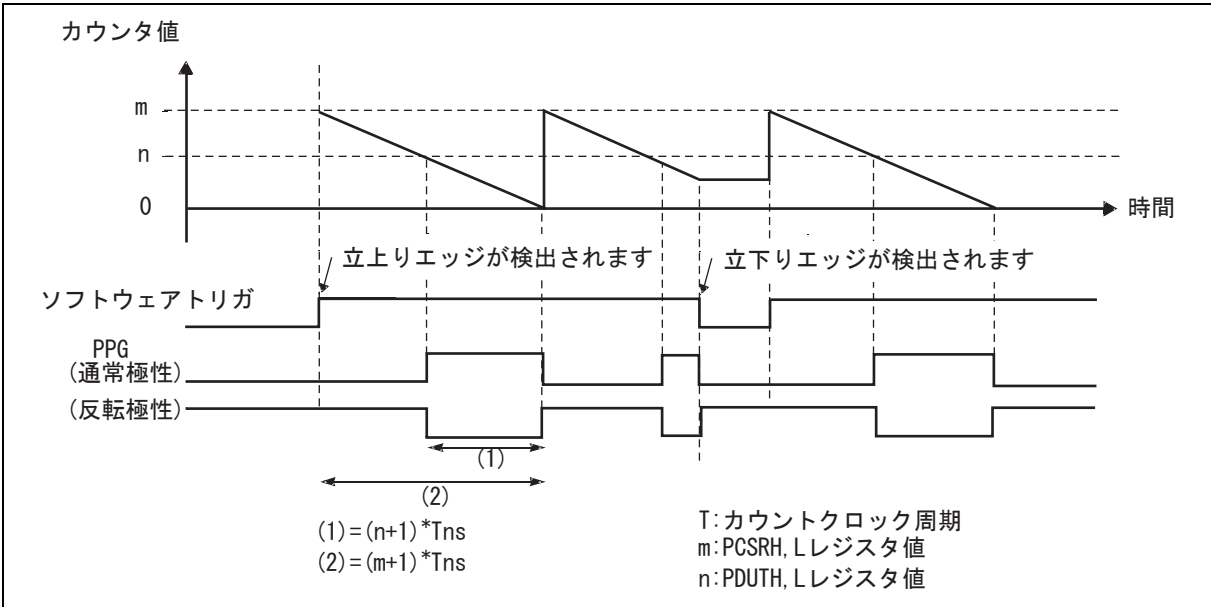
図 13.6-4 ワンショットモードにおいて再トリガを有効にする



■ ゲートトリガ (PPG チャネル 0 の場合のみ)

ゲートトリガを使用した場合、PPG はゲートトリガの立上がりエッジが検出されると動作を開始し、ゲートトリガの立下りエッジが検出されると動作を停止します。PPG は、その次の立上がりエッジで動作を再開します。

図 13.6-5 再トリガを有効にした場合の PWM モードにおけるゲートトリガ

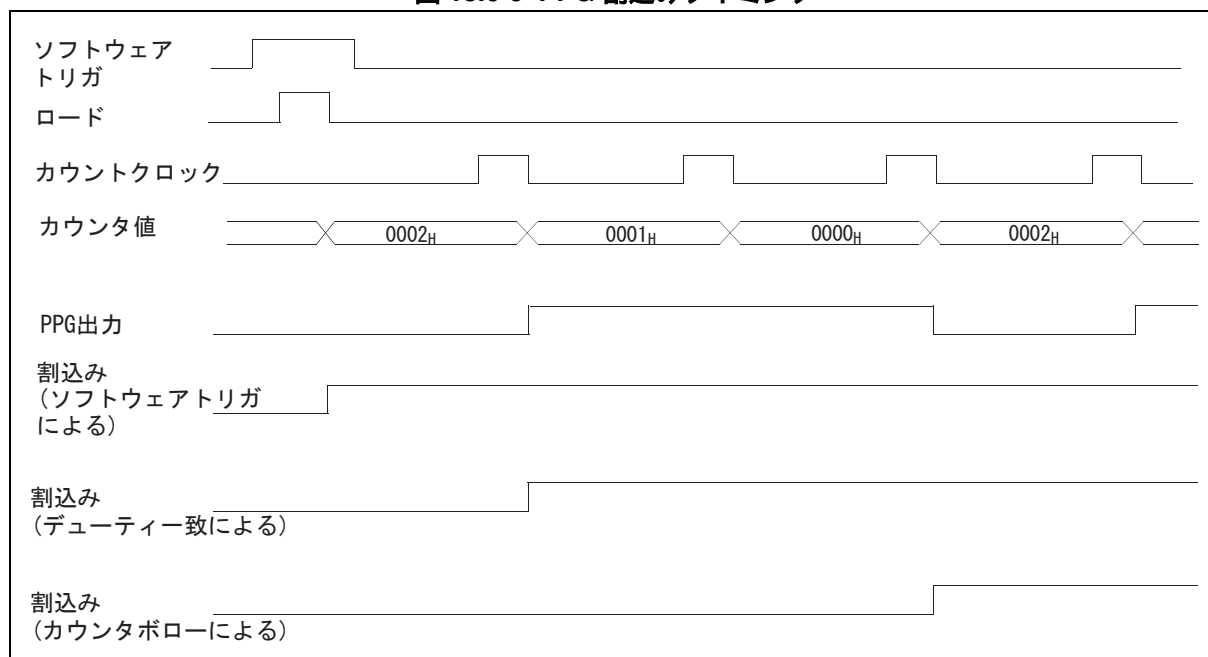


## ■ PPG 割込み

割込み種類選択ビット (PCNTL レジスタの IRS1, IRS0:bit3, bit2) で選択された 1 つの割込みフラグ (PCNTL レジスタの IRQF:bit4) を共用する割込みとしては、以下の 4 種類があります。

- ゲートトリガ (PPG チャンネル 0 の場合のみ)、ソフトウェアトリガまたは再トリガ
- カウンタボロー
- デューティー致 (通常極性において PPG 出力立上りが発生した場合または反転極性において PPG 出力立下りが発生した場合)
- カウンタボローまたはデューティー致

図 13.6-6 PPG 割込みタイミング



## 13.7 16 ビット PPG タイマ使用上の注意

---

16 ビット PPG タイマ使用上の注意を以下に示します。

---

### ■ 16 ビット PPG タイマ使用上の注意

#### ● プログラム設定上の注意

- PPG 周期設定バッファレジスタ (PCSRH, PCSRL) を更新する場合は、この PPG 周期設定バッファレジスタを更新した後、この更新値を PPG デューティ設定バッファレジスタ (PDUTH, PDUTL) に書き込まなければなりません。すなわち、PCSRH, PCSRL レジスタのみの更新は禁止されています。
- PPG デューティ設定バッファレジスタ (PDUTH, PDUTL) へは、必ず PPG 周期設定バッファレジスタ (PCSRH, PCSRL) 以下の値を設定してください。PCSRH, PCSRL の値よりも大きい値を設定すると、PPG 出力は不定になります。
- PPG が停止したとき (PCNTH レジスタの CNTE:bit15=0) は、PPG 状態制御レジスタ (PCNTH) の CKS2 ~ CKS0:bit11 ~ bit9 を変更してください。

#### ● 割込みについての注意

- PPG 状態制御レジスタ (PCNTL) の IRQF:bit4 に "1" を設定し、次に割込み要求を有効にすると (PCNTL レジスタの IREN:bit5=1)、制御は割込み処理から戻ることができません。したがって、IRQF:bit4 は必ずクリアしてください。
- 16 ビット PPG タイマは割込みベクトルをほかのリソースと共用するので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、16 ビット PPG タイマで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みを無効にしなければなりません。

## 13.8 16 ビット PPG タイマのプログラム例

16 ビット PPG タイマのプログラム例を示します。

### ■ 16 ビット PPG タイマのプログラム例

#### ● 処理

- 66% デューティ比の 160kHz 時における出力は、16 ビット PPG タイマ 0 によって生成します。
- 本タイマは PWM モード時において割込みの繰返し生成に使用します。
- 本タイマはソフトウェアトリガによって開始します。
- EI<sup>2</sup>OS を使用しません。
- 16 MHz はマシナクロック用、62.5ns はカウントクロック用です。

#### ● コーディング例

```
ICR01 EQU 0000B1H ; 16 ビット PPG タイマ用割込み制御レジスタ
PCSRLO EQU 00003AH ; PPG 周期設定バッファレジスタ
PDUTLO EQU 00003CH ; PPG デューティ設定バッファレジスタ
PCNTLO EQU 00003EH ; PPG 制御状態レジスタ
IRQF EQU PCNTLO:4 ; 割込み要求フラグビット
;----- メインプログラム -----
CODE CSEG
START:
; : ; スタックポインタ (SP) が既に初期化されているものと仮定

AND CCR, #0BFH ; 割込み禁止
MOV I:ICR01, #00H ; 割込みレベル 0 (最強)
MOVW I:PCSRLO, #0063H; PPG 出力の周期を設定
MOVW I:PDUTLO, #003BH; PPG 出力のデューティ比の設定
MOVW I:PCNTLO, #00110001B
; 通常極性時における PPG 出力を許可
; 16 ビット PPG タイマと 62.5ns クロックを有効
; ソフトウェアが PPG をトリガ
; PWM モードを選択, 割込みを許可
; 割込みフラグをクリア, カウンタ開始

MOV ILM, #07H ; PS の ILM をレベル 7 に設定
OR CCR, #40H ; 割込み許可
LOOP: MOV A, #00H ; 無限ループ
MOV A, #01H ;
BRA LOOP ;
;----- 割込みプログラム -----
WARI:
CLRBI:IRQF ; 割込み要求フラグをクリア
; :
; ユーザ処理
; :
RETI ; 割込みから復帰

CODE ENDS
```

```
;----- ベクタ設定 -----  
VECT      CSEG  ABS=0FFH  
           ORG   0FFC4H      ; 割込み #14 (0EH) 用ベクタを設定  
           DSL   WARI  
           ORG   0FFDCH      ; リセットベクタを設定  
           DSL   START  
           DB     00H        ; シングルチップモードを設定  
VECT      ENDS  
           END   START
```



# 第14章

---

## 多機能タイマ

多機能タイマの機能と動作について説明します。

- 14.1 多機能タイマの概要
- 14.2 多機能タイマのブロックダイアグラム
- 14.3 多機能タイマの端子
- 14.4 多機能タイマのレジスタ
- 14.5 多機能タイマ割込み
- 14.6 多機能タイマの動作
- 14.7 多機能タイマ使用上の注意
- 14.8 多機能タイマのプログラム例



## 14.1 多機能タイマの概要

---

多機能タイマは、1 つの 16 ビットフリーランタイマ、6 つの 16 ビットアウトプットコンペア、4 つの 16 ビットインプットコンペア、1 チャンネルの 16 ビット PPG タイマおよび 1 つの波形ジェネレータから構成されています。この波形ジェネレータを使用すると、12 個の別々の波形を 16 ビットフリーランタイマから出力でき、また入力パルス幅と外部クロックサイクルを測定することもできます。

---

### ■ 16 ビットフリーランタイマ (× 1)

- 16 ビットフリーランタイマは、16 ビットアップ / アップダウンカウンタ、制御レジスタ、16 ビットコンペアクリアレジスタ (バッファレジスタを持っています) およびプリスケアラから構成されています。
- 8 種類のカウンタ動作クロック ( $\phi$ ,  $\phi/2$ ,  $\phi/4$ ,  $\phi/8$ ,  $\phi/16$ ,  $\phi/32$ ,  $\phi/64$ ,  $\phi/128$ ) を選択できます ( $\phi$  はマシンのクロック)。
- コンペアクリア割込みは、コンペアクリアレジスタと 16 ビットフリーランタイマが比較され、一致した場合に生成されます。ゼロ検出割込みは、16 ビットフリーランタイマがカウント値 "0" を検出している間に生成されます。
- コンペアクリアレジスタは、選択可能なバッファレジスタを持っています (このバッファレジスタに書き込まれたデータはコンペアクリアレジスタへ転送されます)。16 ビットフリーランタイマが停止し、バッファにデータが書き込まれると、転送は直ちに実行されます。16 ビットフリーランタイマの動作中にタイマ値 "0" が検出されると、バッファからデータが転送されます。
- アップカウントモードにおいてリセットやソフトウェアクリア、あるいはコンペアクリアレジスタとのコンペア一致が発生すると、カウンタ値は "0000<sub>H</sub>" にリセットされます。
- このカウンタの出力値は、多機能タイマのアウトプットコンペアとインプットキャプチャのクロックカウントとして使用できます。

### ■ 16 ビットアウトプットコンペア (× 6)

- 16 ビットアウトプットコンペアは、6 つの 16 ビットコンペアレジスタ (選択可能なバッファレジスタを持っています)、コンペア出力ラッチ、コンペア制御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とコンペアレジスタが一致すると、割込みが生成され、出力レベルが反転します。
- 6 つのコンペアレジスタは、別々に動作させることができます。  
出力端子と割込みフラグは、各コンペアレジスタに対応しています。
- 2 つのコンペアレジスタを対 (ペア) にして出力端子を制御できます。  
2 つのコンペアレジスタを一緒に使用することによって出力端子を反転させます。
- 各出力端子の初期値を設定できます。
- 割込みは、アウトプットコンペアレジスタが 16 ビットフリーランタイマと一致した場合に生成されます。

## ■ 16 ビットインプットキャプチャ (× 4)

インプットキャプチャは、4 つの独立した外部入力端子と、この端子に対応するキャプチャレジスタおよびキャプチャ制御レジスタから構成されています。外部端子において入力信号のエッジを検出すると、16 ビットフリーランタイムの値をキャプチャレジスタへ格納でき、また割込みも同時に生成されます。

- 外部入力信号の 3 種類のトリガエッジ (立上りエッジ、立下りエッジおよびその両方のエッジ) を選択でき、またトリガエッジが立上りエッジであるか立下りエッジであるかを示すビットを持っています。
- 4 つの入力キャプチャを別々に動作させることができます。
- 割込みは、外部入力からの有効なエッジが検出されると生成されます。
- チャンネル 0 とチャンネル 1 は、割込み #33 を共用します。
- チャンネル 2 とチャンネル 3 は、割込み #35 を共用します。

## ■ 16 ビット PPG タイマ (× 1)

16 ビット PPG タイマ 0 は、波形ジェネレータへ PPG 信号を供給するために使用します。16 ビット PPG タイマ 0 の詳細については、「第 13 章 16 ビット PPG タイマ」をご参照ください。

## ■ 波形ジェネレータ

波形ジェネレータは、3 つの 16 ビットデッドタイムレジスタ、3 つのタイマ制御レジスタおよび 1 つの 16 ビット波形制御レジスタから構成されています。

波形ジェネレータは、リアルタイム出力、16 ビット PPG 波形出力、ノンオーバーラップ 3 相波形出力 (インバータ制御用) および DC チョッパ波形出力を生成できます。

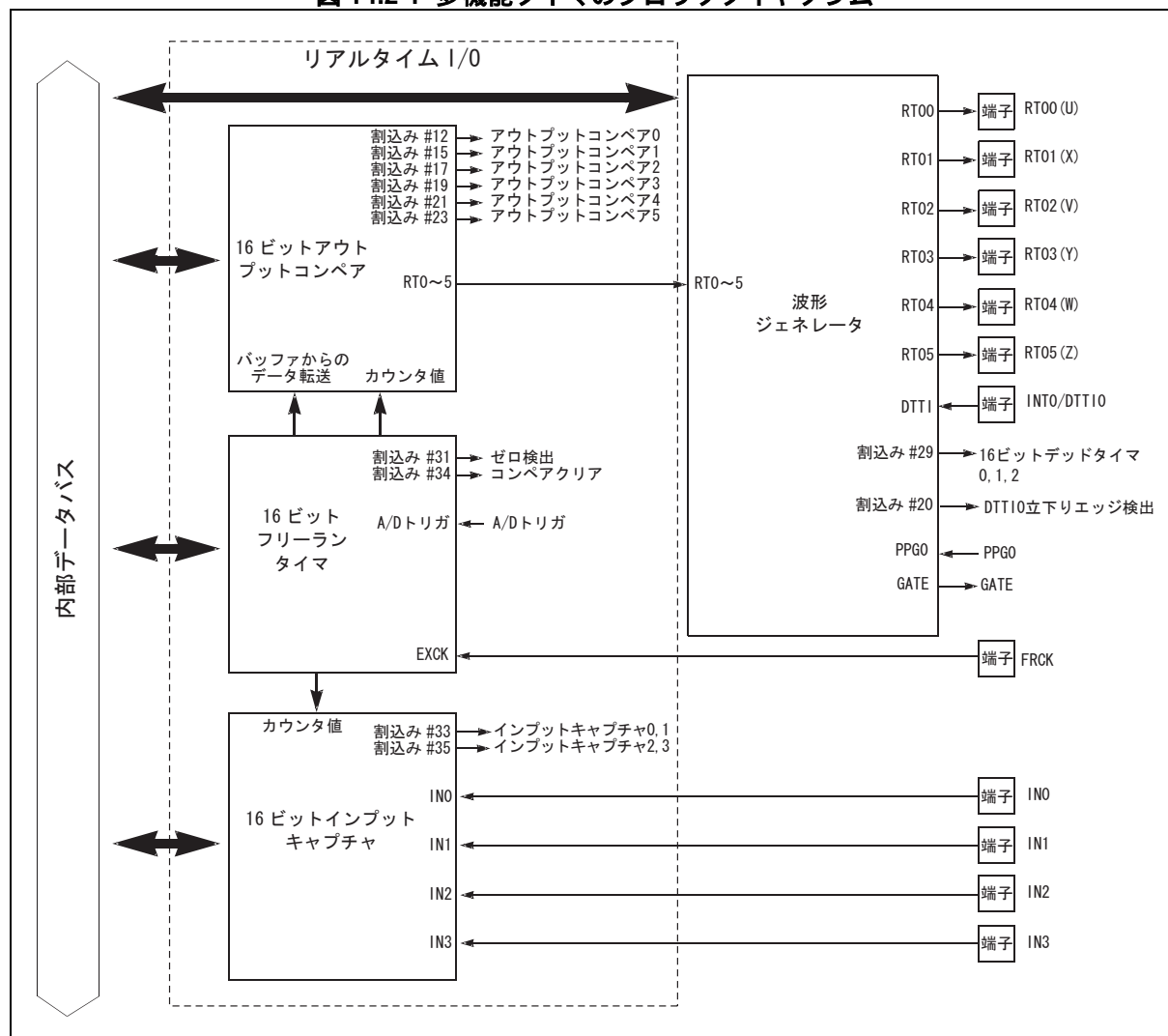
- 16 ビットデッドタイムのデッドタイムに基づいてノンオーバーラップ波形出力を生成できます (デッドタイムタイマ機能)。
- 2 チャンネルモード時にリアルタイムアウトプットを動作させることにより、ノンオーバーラップ波形出力を生成できます (デッドタイムタイマ機能)。
- リアルタイムアウトプットコンペア一致を検出すると、GATE 信号が生成され、この信号により PPG タイマの動作が開始または停止します (GATE 機能)。
- リアルタイムアウトプットコンペア一致が検出されると、16 ビットデッドタイムがアクティブになり、PPG 動作の制御用 GATE 信号を生成することによって、PPG タイマを容易に開始または停止させることができます (GATE 機能)。
- DTTI 端子を使用することによって強制的に停止を制御できます。

## 14.2 多機能タイマのブロックダイアグラム

多機能タイマのブロックダイアグラムを示します。

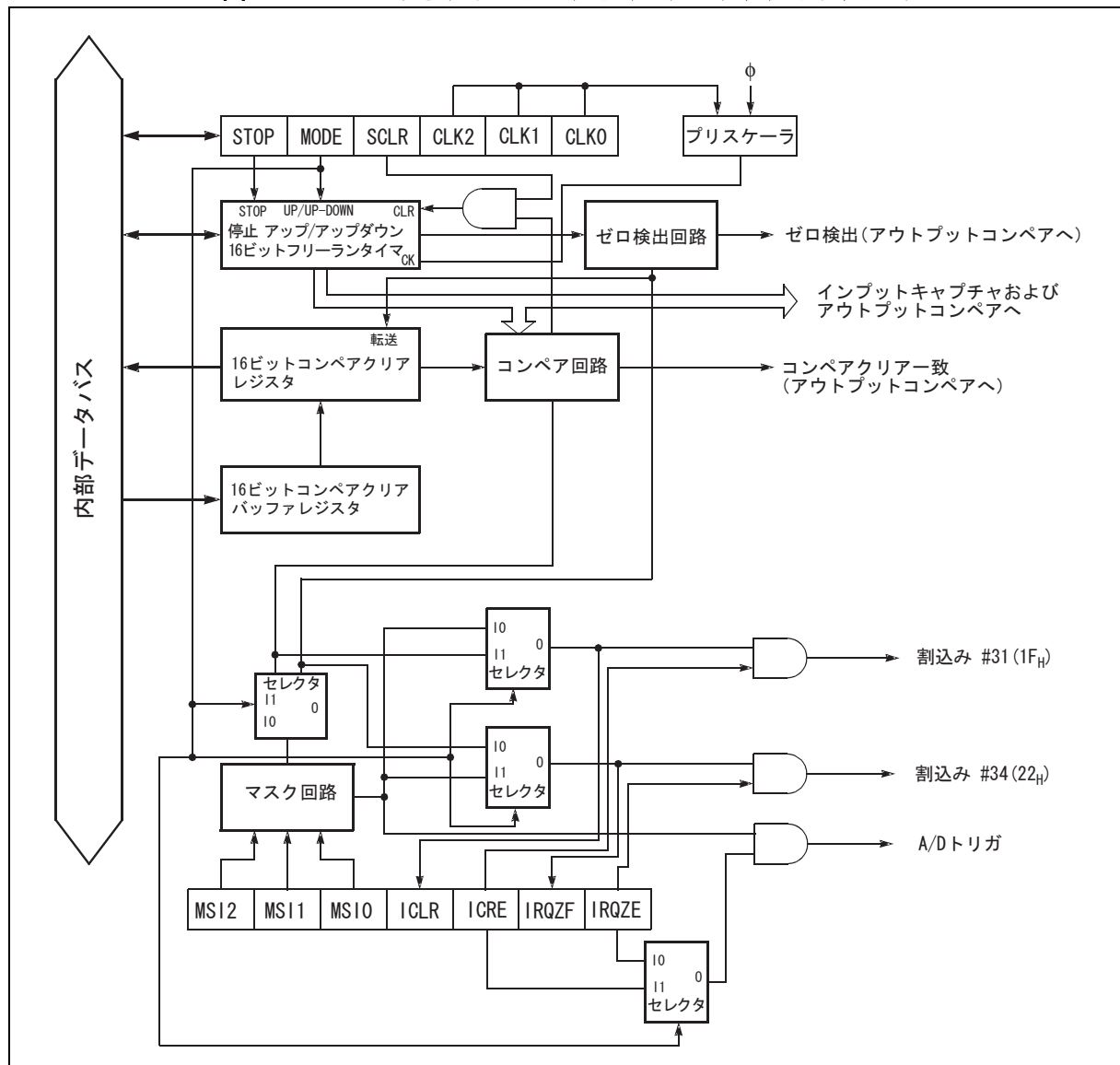
### ■ 多機能タイマのブロックダイアグラム

図 14.2-1 多機能タイマのブロックダイアグラム



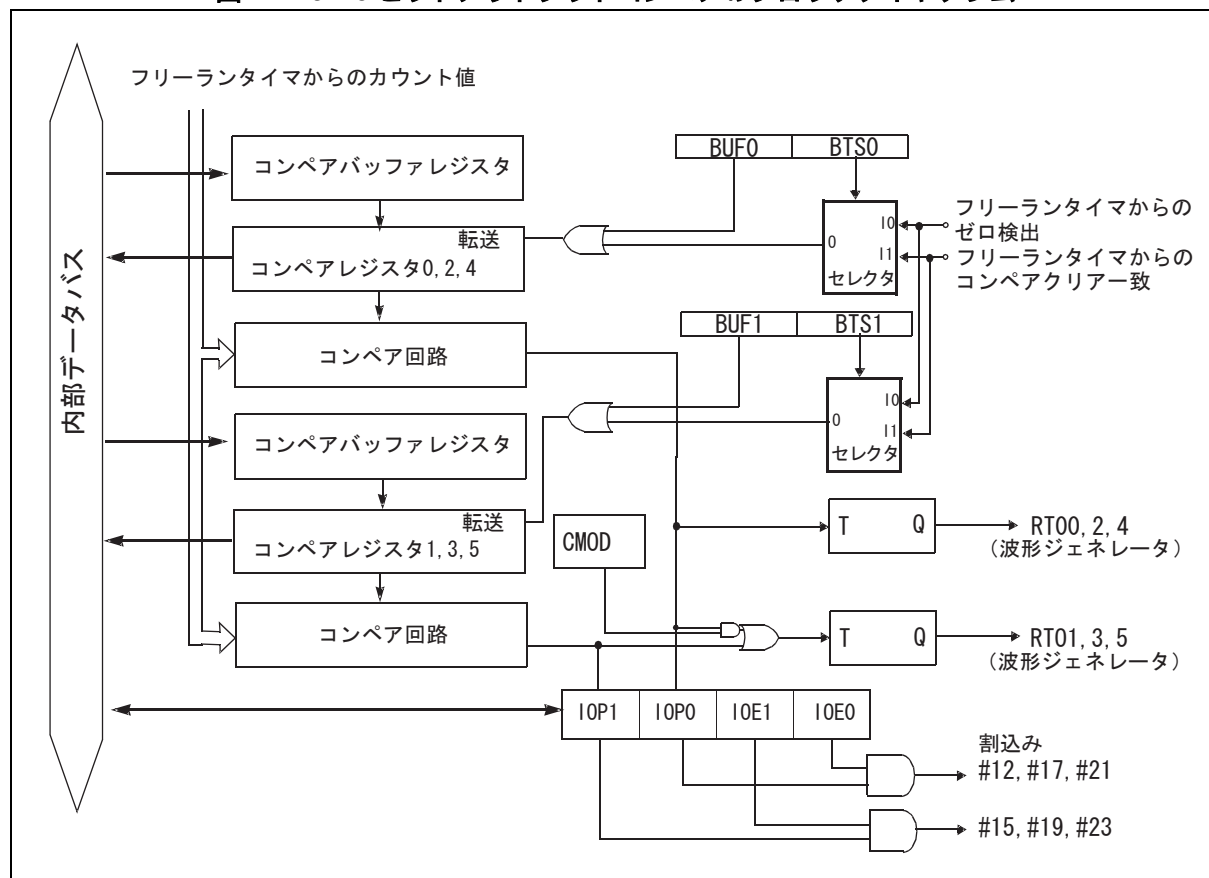
## ■ 16 ビットフリーランタイムのブロックダイアグラム

図 14.2-2 16 ビットフリーランタイムのブロックダイヤグラム



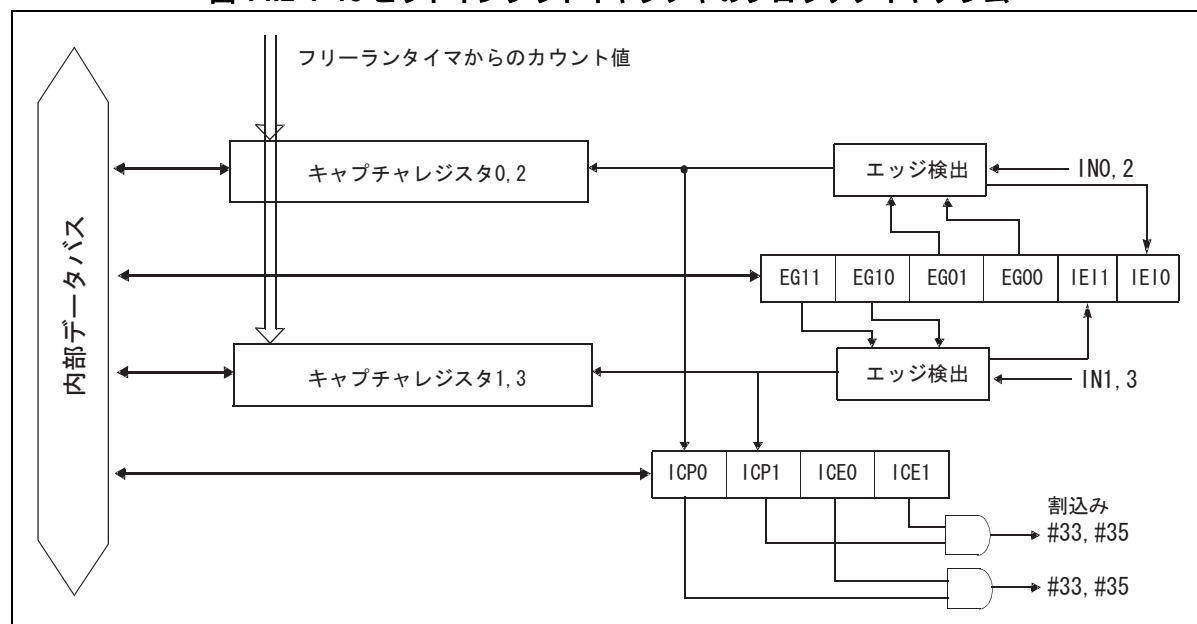
## ■ 16 ビットアウトプットコンペアのブロックダイアグラム

図 14.2-3 16 ビットアウトプットコンペアのブロックダイアグラム



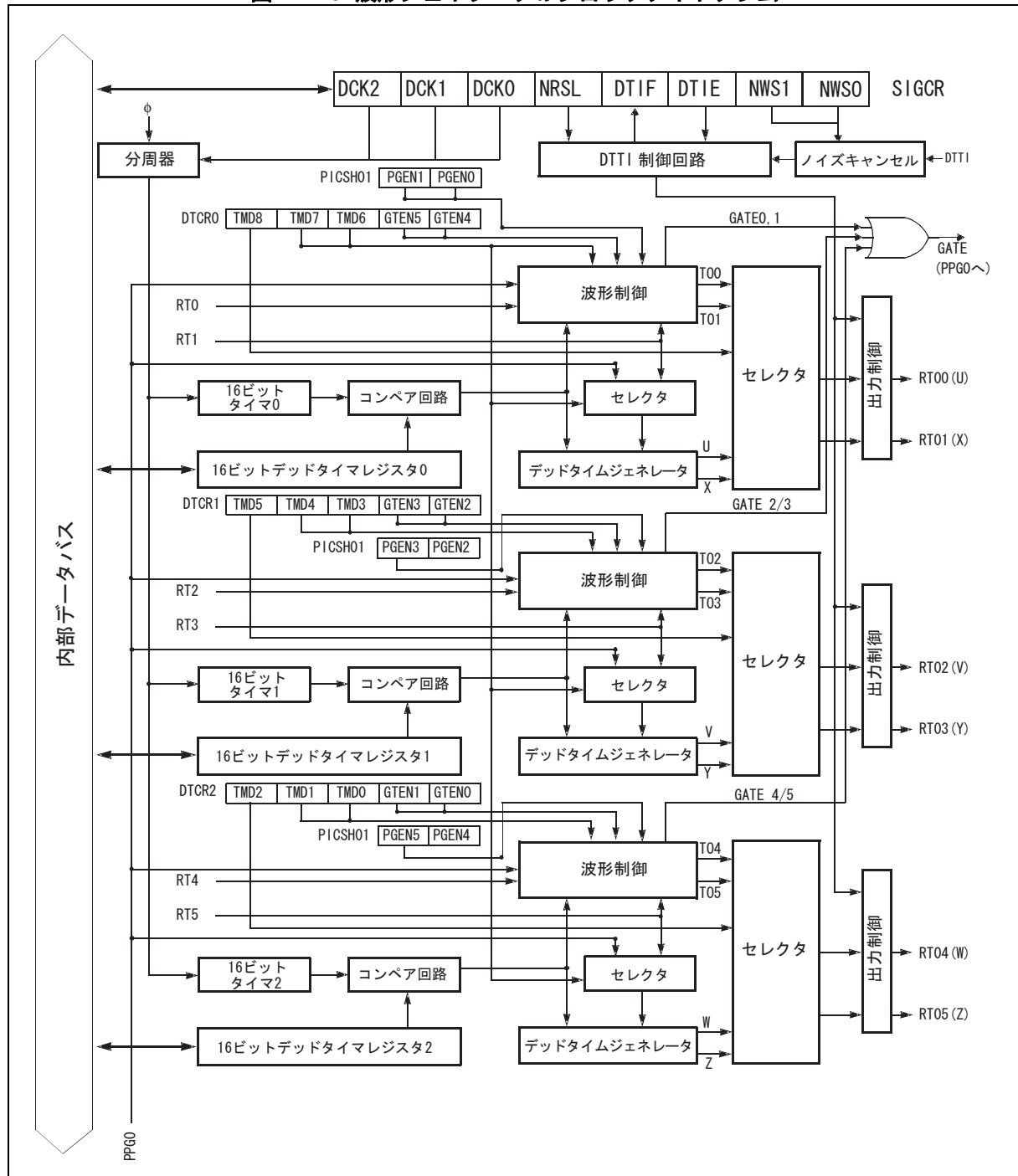
## ■ 16 ビットインプットキャプチャのブロックダイアグラム

図 14.2-4 16 ビットインプットキャプチャのブロックダイアグラム



## ■ 波形ジェネレータのブロックダイアグラム

図 14.2-5 波形ジェネレータのブロックダイアグラム



## 14.3 多機能タイマの端子

多機能タイマの端子について説明します。また端子のブロックダイアグラムも示します。

### ■ 多機能タイマの端子

表 14.3-1 多機能タイマの端子

端子名	端子機能	I/O 形式	ブルアップ オプション	スタンバイ 制御	端子設定
P10/INT0/ DTTI0	ポート 1 入出力, 外部割込み入力, DTTI0	CMOS 出力, CMOS ヒステ リシス入力	選択可能	あり	端子を入力ポートとして設 定する (DDR1:bit0=0)
P17/FRCK	ポート 1 入出力, 外部クロック				端子を入力ポートとして設 定する (DDR1:bit7=0)
P24/IN0	ポート 2 入出力, インプットキャ プチャ 0		端子を入力ポートとして設 定する (DDR2:bit4=0)		
P25/IN1	ポート 2 入出力, インプットキャ プチャ 1		端子を入力ポートとして設 定する (DDR2:bit5=0)		
P26/IN2	ポート 2 入出力, インプットキャ プチャ 2		端子を入力ポートとして設 定する (DDR2:bit6 =0)		
P27/IN3	ポート 2 入出力, インプットキャ プチャ 3		端子を入力ポートとして設 定する (DDR2:bit7=0)		
P30/RTO0 ( U )	ポート 3 入出力, RTO0	CMOS 出力, CMOS 入力	なし		RTO0 出力を設定する ( OCS1:OTE0=1 )
P31/RTO1 ( X )	ポート 3 入出力, RTO1				RTO1 出力を設定する ( OCS1:OTE1=1 )
P32/RTO2 ( V )	ポート 3 入出力, RTO2				RTO2 出力を設定する ( OCS3:OTE0=1 )
P33/RTO3 ( Y )	ポート 3 入出力, RTO3				RTO3 出力を設定する ( OCS3:OTE1=1 )
P34/RTO4 ( W )	ポート 3 入出力, RTO4				RTO4 出力を設定する ( OCS5:OTE0=1 )
P35/RTO5 ( Z )	ポート 3 入出力, RTO5				RTO5 出力を設定する ( OCS5:OTE1=1 )

DDR<sub>x</sub>: ポート方向レジスタ

OCS<sub>x</sub>: コンペア制御レジスタ





## 14.4 多機能タイマのレジスタ

多機能タイマのレジスタについて説明します。

### ■ 16 ビットフリーランタイマのレジスタ

図 14.4-1 16 ビットフリーランタイマのレジスタ

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	CPCLR <sub>BH</sub> /CPCLR <sub>H</sub>
(CPCLR <sub>BH</sub> ) リード/ライト ⇨	W	W	W	W	W	W	W	W	
(CPCLR <sub>H</sub> ) ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	1	1	1	1	1	1	1	1	

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (下位)

bit	7	6	5	4	3	2	1	0	⇨ ビット番号
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	CPCLR <sub>B</sub> L/CPCLR <sub>L</sub>
(CPCLR <sub>B</sub> L) リード/ライト ⇨	W	W	W	W	W	W	W	W	
(CPCLR <sub>L</sub> ) ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	1	1	1	1	1	1	1	1	

タイマデータレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	T15	T14	T13	T12	T11	T10	T09	T08	TCDTH
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

タイマデータレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	T07	T06	T05	T04	T03	T02	T01	T00	TCDTL
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

タイマ状態制御レジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	ECKE	IRQZF	IRQZE	MS12	MS11	MS10	ICLR	ICRE	TCCSH
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

タイマ状態制御レジスタ (下位)

bit	7	6	5	4	3	2	1	0	⇨ ビット番号
	—	BFE	STOP	MODE	SCLR	CLK2	CLK1	CLK0	TCCSL
リード/ライト ⇨	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	X	0	1	0	0	0	0	0	

## ■ 16 ビットアウトプットコンペアのレジスタ

図 14.4-2 16 ビットアウトプットコンペアのレジスタ

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	OCCPBH0~OCCPBH5/ OCCPH0~OCCPH5
リード/ライト ⇨	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	

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	OCCPBL0~OCCPBL5/ OCCPL0~OCCPL5
リード/ライト ⇨	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	

コンペア制御レジスタ1, 3, 5 (上位)

bit	15	14	13	12	11	10	9	8	
	—	BTS1	BTS0	CMOD	OTE1	OTE0	OTD1	OTD0	OCSH1, OCSH3, OCSH5
リード/ライト ⇨	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	X	1	1	0	0	0	0	0	

コンペア制御レジスタ0, 2, 4 (下位)

bit	7	6	5	4	3	2	1	0	
	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0	OCSL0, OCSL2, OCSL4
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	1	1	0	0	

## ■ 16 ビットインプットキャプチャのレジスタ

図 14.4-3 16 ビットインプットキャプチャのレジスタ

インプットキャプチャデータレジスタ(上位)

bit	15	14	13	12	11	10	9	8	
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	IPCPH0~IPCPL3
リード/ライト ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	X	X	X	X	X	X	X	X	

インプットキャプチャデータレジスタ(下位)

bit	7	6	5	4	3	2	1	0	
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	IPCPL0~IPCPL3
リード/ライト ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	X	X	X	X	X	X	X	X	

インプットキャプチャ状態制御レジスタ(ch2, 3) (上位)

bit	15	14	13	12	11	10	9	8	
	—	—	—	—	—	—	IE13	IE12	ICSH23
リード/ライト ⇨	—	—	—	—	—	—	R	R	
初期値 ⇨	X	X	X	X	X	X	0	0	

インプットキャプチャ状態制御レジスタ(ch2, 3) (下位)

bit	7	6	5	4	3	2	1	0	
	ICP3	ICP2	ICE3	ICE2	EG31	EG31	EG21	EG20	ICSL23
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

PPG 出力制御 / インプットキャプチャ状態制御レジスタ(ch0, 1) (上位)

bit	15	14	13	12	11	10	9	8	
	PGEN5	PGEN4	PGEN3	PGEN2	PGEN1	PGEN0	IE11	IE10	PICSH01
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R	R	
初期値 ⇨	0	0	0	0	0	0	0	0	

インプットキャプチャ状態制御レジスタ(ch0, 1) (下位)

bit	7	6	5	4	3	2	1	0	
	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	PICSL01
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

## ■ 波形ジェネレータのレジスタ

図 14.4-4 波形ジェネレータのレジスタ

## 16 ビットデッドタイムレジスタ(上位)

bit	15	14	13	12	11	10	9	8	
	TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	TMRRH0~TMRRH2
リード/ライト⇒	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	

## 16 ビットデッドタイムレジスタ(下位)

bit	7	6	5	4	3	2	1	0	
	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	TMRRLO~TMRR2
リード/ライト⇒	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	

## 16 ビットデッドタイム制御レジスタ 0

bit	7	6	5	4	3	2	1	0	
	DMOD0	GTEN1	GTEN0	TMIF0	TMIE0	TMD2	TMD1	TMD0	DTCR0
リード/ライト⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値⇒	0	0	0	0	0	0	0	0	

## 16 ビットデッドタイム制御レジスタ 1

bit	15	14	13	12	11	10	9	8	
	DMOD1	GTEN3	GTEN2	TMIF1	TMIE1	TMD5	TMD4	TMD3	DTCR1
リード/ライト⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値⇒	0	0	0	0	0	0	0	0	

## 16 ビットデッドタイム制御レジスタ 2

bit	7	6	5	4	3	2	1	0	
	DMOD2	GTEN5	GTEN4	TMIF2	TMIE2	TMD8	TMD7	TMD6	DTCR2
リード/ライト⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値⇒	0	0	0	0	0	0	0	0	

## 波形制御レジスタ

bit	15	14	13	12	11	10	9	8	
	DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0	SIGCR
リード/ライト⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値⇒	0	0	0	0	0	0	0	0	

### 14.4.1 コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL) / コンペアクリアレジスタ (CPCLRH, CPCLRL)

コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL) は、コンペアクリアレジスタ (CPCLRH, CPCLRL) に存在する 16 ビットバッファレジスタです。CPCLRBH, CPCLRBL レジスタと CPCLRH, CPCLRL レジスタは、両方とも同じアドレスに存在します。

#### ■ コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL)

図 14.4-5 コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL)

コンペアクリアバッファレジスタ (上位)										
bit		15	14	13	12	11	10	9	8	
		CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	CPCLRBH
リード/ライト	⇒	W	W	W	W	W	W	W	W	
初期値	⇒	1	1	1	1	1	1	1	1	
コンペアクリアバッファレジスタ (下位)										
bit		7	6	5	4	3	2	1	0	
		CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	CPCLRBL
リード/ライト	⇒	W	W	W	W	W	W	W	W	
初期値	⇒	1	1	1	1	1	1	1	1	

コンペアクリアバッファレジスタは、コンペアクリアレジスタ (CPCLRH, CPCLRL) と 同じアドレス に存在するバッファレジスタです。バッファ機能が無効になるか (タイマ状態制御レジスタ下位 (TCCSL) の BFE:bit6=0) またはフリーランタイマが停止すると、コンペアクリアバッファレジスタの値が直ちにコンペアクリアレジスタへ転送されます。バッファ機能が有効になると、16 ビットフリーランタイマのカウント値 "0" が検出されたときに値がコンペアクリアレジスタへ転送されます。

このレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。

#### ■ コンペアクリアレジスタ (CPCLRH, CPCLRL)

図 14.4-6 コンペアクリアレジスタ (CPCLRH, CPCLRL)

コンペアクリアレジスタ (上位)												
		bit 15		14	13	12	11	10	9	8		
				CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	CPCLR <sub>H</sub>
リード/ライト ⇒		R		R		R		R		R		
初期値 ⇒		1		1		1		1		1		
コンペアクリアレジスタ (下位)												
		bit 7		6	5	4	3	2	1	0		
				CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	CPCLR <sub>L</sub>
リード/ライト ⇒		R		R		R		R		R		
初期値 ⇒		1		1		1		1		1		

コンペアクリアレジスタは、16 ビットフリーランタイマのカウント値と比較されるために使用します。アップカウントモード時は、このレジスタが 16 ビットフリーランタイマのカウント値と一致すると、16 ビットフリーランタイマは "0000<sub>H</sub>" にリセットされます。アップダウンカウントモード時は、このレジスタが 16 ビットフリーランタイマのカウント値と一致すると、16 ビットフリーランタイマは "0" 検出時にアップカウントからダウンカウントに変わるかまたはダウンカウントからアップカウントに変わります。

このレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。

## 14.4.2 タイマデータレジスタ (TCDTH, TCDTL)

タイマデータレジスタ (TCDTH, TCDTL) は, 16 ビットフリーランタイムのカウンタ値を読み出すために使用します。

### ■ タイマデータレジスタ (TCDTH, TCDTL)

図 14.4-7 タイマデータレジスタ (TCDTH, TCDTL)

タイマデータレジスタ(上位)									
	bit 15	14	13	12	11	10	9	8	
	T15	T14	T13	T12	T11	T10	T09	T08	TCDTH
リード/ライト ⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇒	0	0	0	0	0	0	0	0	

タイマデータレジスタ(下位)									
	bit 7	6	5	4	3	2	1	0	
	T07	T06	T05	T04	T03	T02	T01	T00	TCDTL
リード/ライト ⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇒	0	0	0	0	0	0	0	0	

タイマデータレジスタは, 16 ビットフリーランタイムのカウンタ値を読み出すために使用します。カウンタ値は, リセットが発生すると直ちに "000<sub>H</sub>" にクリアされます。タイマ値は, このレジスタへ値を書き込むことで設定できます。ただし, 値の書き込みはタイマの停止中 (タイマ状態制御レジスタ下位 (TCCSL) の STOP:bit5=1) でなければなりません。タイマデータレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。

16 ビットフリーランタイムは, 以下の要因が発生すると直ちに初期化されます。

- リセット
- タイマ状態制御レジスタ (TCCSL) のクリアビット (SCLR:bit3) =1
- アップカウントモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE:bit4=0) 時におけるコンペアクリアレジスタとタイマカウンタ値の一致

### 14.4.3 タイマ状態制御レジスタ (TCCSH, TCCSL)

タイマ状態制御レジスタ (TCCSH, TCCSL) は, 16 ビットフリーランタイムの動作を制御するために使用する 16 ビットレジスタです。

#### ■ タイマ状態制御レジスタ, 上位バイト (TCCSH)

図 14.4-8 タイマ状態制御レジスタ, 上位バイト (TCCSH)

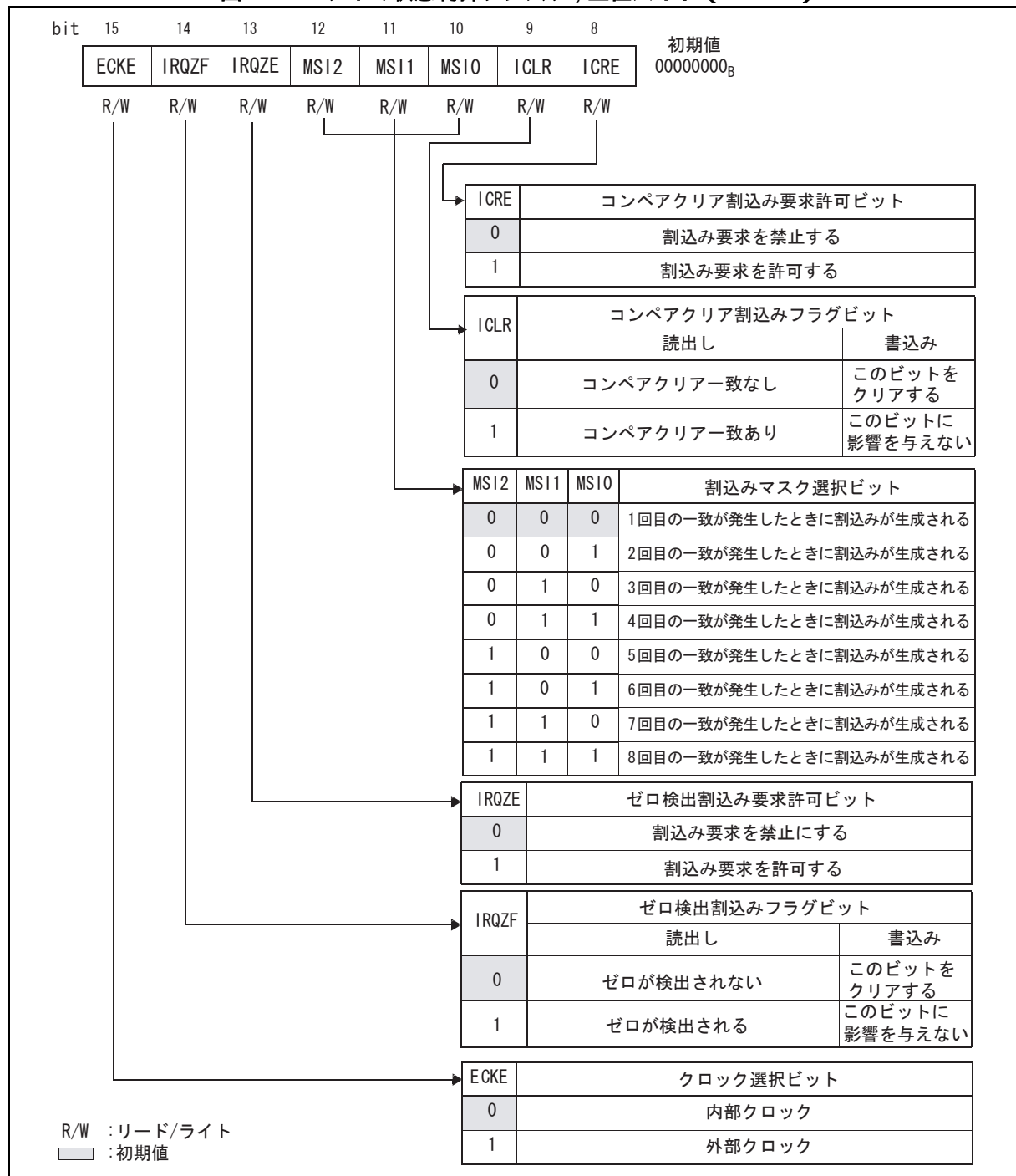


表 14.4-1 タイマ状態制御レジスタ, 上位バイト (TCCSH) ( 1 / 2 )

ビット名		機 能
bit15	ECKE: クロック選 択ビット	<ul style="list-style-type: none"> <li>このビットは、内部クロックまたは外部クロックを 16 ビットフリーランタイムのカウンタクロックとして選択するために使用します。</li> <li>このビットに "0" を設定した場合：内部クロックが選択されます。カウンタクロック周波数を選択するためには、TCCSL レジスタのクロック周波数選択ビット (CLK2 ~ CLK0:bit2 ~ bit0) も設定しなければなりません。</li> <li>このビットに "1" を設定した場合：外部クロックが選択されます。外部クロックは、"FRCK" 端子から入力されます。したがって、ポート方向レジスタ (DDR1) の bit7 へ "0" を書き込んで外部クロック入力を有効にしなければなりません。</li> </ul> <p>(注意事項)</p> <p>カウンタクロックは、このビットが設定されると直ちに変更されます。したがって、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間でなければなりません。</p>
bit14	IRQZF: ゼロ検出割 込みフラグ ビット	<ul style="list-style-type: none"> <li>16 ビットフリーランタイムのカウンタ値が "0000<sub>H</sub>" のとき、このビットには "1" がセットされます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul> <p>(注意事項)</p> <p>ソフトウェアクリア (タイマ状態制御レジスタ下位 (TCCSL) の SCLR:bit3 への "1" 書込み) では、このビットは設定されません。</p> <p>アップダウンカウンタモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE:bit4=1) 時は、割込みマスク選択ビット (タイマ状態制御レジスタ上位 (TCCSH) の MSI2 ~ MSI0:bit12 ~ bit10 が "000<sub>B</sub>" 以外) で設定した割込みが発生したときにこのビットに "1" が設定されます。割込みが発生しないときは、このビットに "1" は設定されません。</p> <p>アップカウンタモード (MODE: ビット 4 =0) 時は、MSI2 ~ MSI0:bit12 ~ bit10 の値とは無関係に、このビットはゼロ検出が発生するたびに設定されます。</p>
bit13	IRQZE: ゼロ検出割 込み要求許 可ビット	<ul style="list-style-type: none"> <li>このビットと割込みフラグビット (IRQZF:bit14) に "1" が設定されると、CPU に対する割込み要求が生成されます。</li> </ul>
bit12 ~ bit10	MSI2 ~ MSI0: 割込みマス ク選択ビッ ト	<ul style="list-style-type: none"> <li>これらのビットは、アップカウンタモード (MODE=0) 時はコンペアクリア割込みのマスク回数を設定するために使用します。アップダウンカウンタモード (MODE=1) 時は、ゼロ検出割込みのマスク回数を設定するために使用します。</li> <li>このビットに "0" を設定した場合：割込み要因はマスクされません。</li> </ul> <p>(注意事項)</p> <p>割込み要因を 2 回マスクし、3 回目の割込みを処理する際には、これらのビットに "010<sub>B</sub>" を設定しなければなりません。</p>



表 14.4-1 タイマ状態制御レジスタ, 上位バイト (TCCSH)( 2 / 2 )

ビット名		機 能
bit9	ICLR: コンペアク リア割込み フラグビット	<ul style="list-style-type: none"> <li>コンペアクリア値と 16 ビットフリーランタイム値が一致すると、このビットには "1" が設定されます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul> <p>(注意事項)</p> <p>アップカウントモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE: bit4=1) 時は、割込みマスク選択ビット (タイマ状態制御レジスタ上位 (TCCSH) の MSI2 ~ MSI0:bit12 ~ bit10 が "000<sub>B</sub>" 以外) で設定した割込みが発生したときにこのビットに "1" が設定されます。割込みが発生しないときは、このビットに "1" は設定されません。</p> <p>アップダウンカウントモード (MODE=1) 時は、MSI2 ~ MSI0 ビットの値とは無関係に、このビットはコンペアクリアが発生するたびに設定されます。</p>
bit8	ICRE: コンペアク リア割込み 要求許可 ビット	<ul style="list-style-type: none"> <li>このビットとコンペアクリア割込みフラグビット (ICLR:bit9) に "1" が設定されると、CPU に対する割込み要求が生成されます。</li> </ul>

## ■ タイマ状態制御レジスタ, 下位バイト (TCCSL)

図 14.4-9 タイマ状態制御レジスタ, 下位バイト (TCCSL)

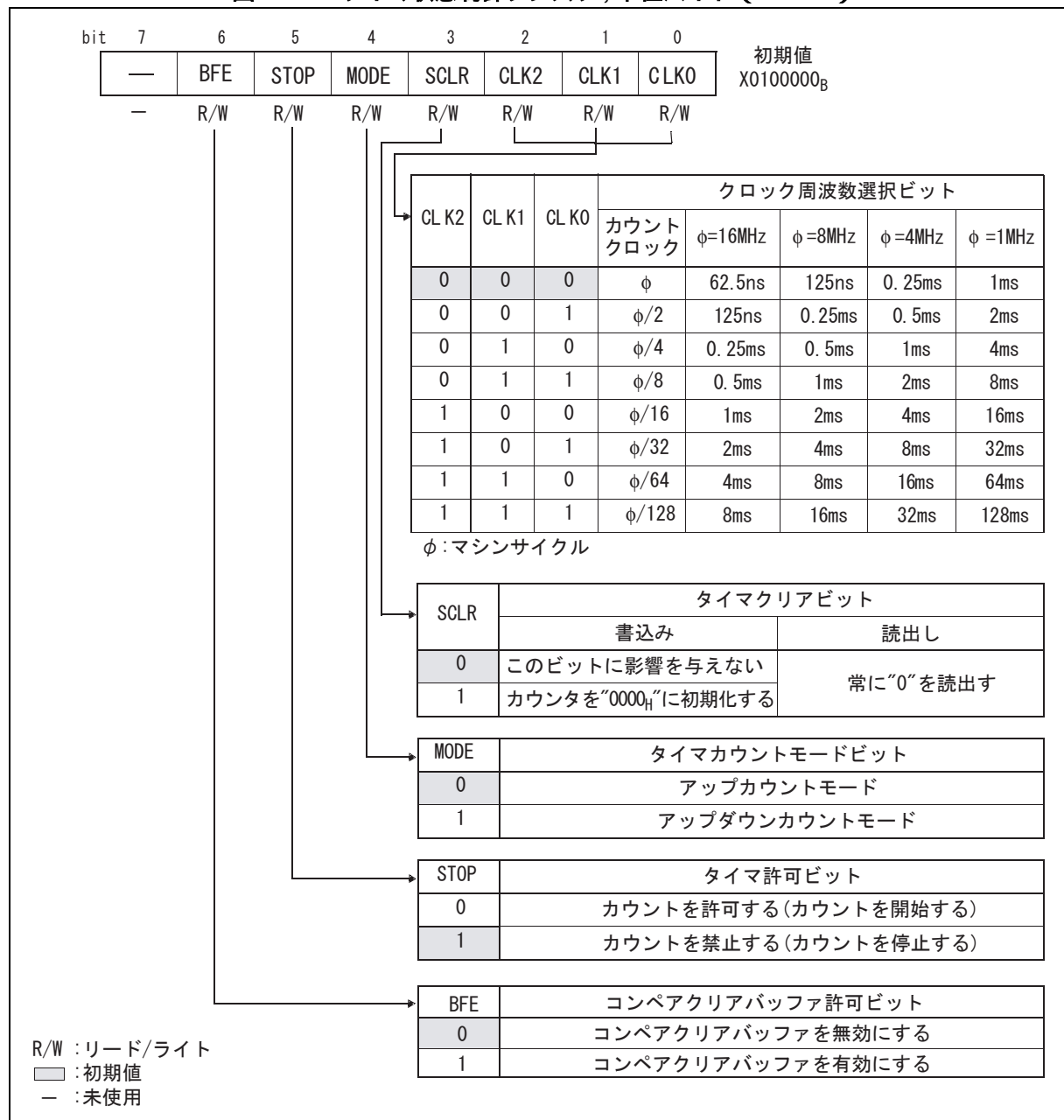


表 14.4-2 タイマ状態制御レジスタ, 下位バイト (TCCSL)( 1 / 2 )

ビット名		機 能
bit7	未使用ビット	<ul style="list-style-type: none"> <li>読出し値は不定です。</li> <li>このビットへの書込みは動作に影響しません。</li> </ul>
bit6	BFE: コンペアクリアバッファ有効ビット	<ul style="list-style-type: none"> <li>このビットは、コンペアクリアバッファを有効にするために使用します。</li> <li>このビットに "0" を設定した場合: コンペアクリアバッファは無効になります。したがってコンペアクリアレジスタ (CPCLR<sub>H</sub>, CPCLR<sub>L</sub>) に直接書き込むことが可能です。</li> <li>このビットに "1" を設定した場合: コンペアクリアバッファは有効になります。コンペアクリアバッファに書き込まれ保持されていたデータは、16 ビットフリーランタイムのカウント値 "0" が検出されると、コンペアクリアレジスタへ転送されます。</li> </ul>
bit5	STOP: タイマ有効ビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットフリーランタイムのカウントを停止 / 開始するために使用します。</li> <li>このビットに "0" を設定した場合: 16 ビットフリーランタイムのカウントは開始します。</li> <li>このビットに "1" を設定した場合: 16 ビットフリーランタイムのカウントは停止します。</li> </ul> <p>(注意事項)</p> <p>16 ビットフリーランタイムが停止すると、アウトプットコンペアの動作も停止します。</p>
bit4	MODE: タイマカウントモードビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットフリーランタイムのカウントモードを選択するために使用します。</li> <li>このビットに "0" を設定した場合: アップカウントモードが選択されます。タイマは、カウンタ値がコンペアクリアレジスタと一致して "0000<sub>H</sub>" にリセットされるまでカウントアップし、その後、再びカウントアップします。</li> <li>このビットに "1" を設定した場合: アップダウンカウントモードが選択されます。タイマは、カウンタ値がコンペアクリアレジスタと一致するまでカウントアップし、その後ダウンカウントに変わります。その後、カウンタ値が "0000<sub>H</sub>" に達すると再びアップカウントに変わります。</li> <li>このビットは、タイマが動作中であっても停止されていても書込みが可能です。タイマが動作中の場合は、このビットに書き込まれた値はバッファに入れられ、その後、タイマ値が "0000<sub>H</sub>" になるとバッファの値によりカウントモードが変わります。</li> </ul>
bit3	SCLR: タイマクリアビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットフリーランタイムを "0000<sub>H</sub>" に初期化するために使用します。</li> <li>このビットに "0" を設定した場合: 意味はありません。</li> <li>このビットに "1" を設定した場合: 16 ビットフリーランタイムは、その次のカウントクロックで "0000<sub>H</sub>" に初期化されます。</li> <li>読出し値は、必ず "0" です。</li> </ul> <p>(注意事項)</p> <ul style="list-style-type: none"> <li>このビットは、タイマが停止しているとき (STOP=1) にタイマを初期化するためには使用できません。タイマデータレジスタ (TCDTH, TCDTL) に "0000<sub>H</sub>" を書き込むと、タイマを初期化できます。</li> <li>このビットに "1" を書き込んでも、ゼロ検出割込みは生成されません。</li> <li>"1" を書き込んだ後、次のカウントクロックまでにこのビットへの "0" 書込みを行うと、カウンタ値の初期化は行われません。</li> </ul>

表 14.4-2 タイマ状態制御レジスタ, 下位バイト (TCCSL)( 2 / 2 )

ビット名		機 能
bit2 ~ bit0	CLK2 ~ 0: クロック周波 数選択ビット	<ul style="list-style-type: none"><li>• このビットは, 16 ビットフリーランタイマのカウントクロック周波数を選択するために使用します。</li><li>• カウントクロックは, これらのビットが設定されると直ちに変更されます。したがって, これらのビットの変更は, 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間でなければなりません。</li></ul>

#### 14.4.4 アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL0 ~ OCCPBL5) / アウトプットコンペアレジスタ (OCCPH, OCCPL0 ~ OCCPL5)

アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL) は, アウトプットコンペアレジスタ (OCCPH, OCCPL) 用の 16 ビットバッファレジスタです。OCCPBH, OCCPBL レジスタと OCCPH, OCCPL レジスタは, 両方とも同じアドレスに存在します。

##### ■ アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL0 ~ OCCPBL5)

図 14.4-10 アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL0 ~ OCCPBL5)

アウトプットコンペアバッファレジスタ (上位)									
	bit 15	14	13	12	11	10	9	8	
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	OCCPBH0~OCCPBH5
リード/ライト ⇨	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	
アウトプットコンペアバッファレジスタ (下位)									
	bit 7	6	5	4	3	2	1	0	
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	OCCPBL0~OCCPBL 5
リード/ライト ⇨	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	

アウトプットコンペアバッファレジスタは, アウトプットコンペアレジスタ (OCCPH, OCCPL) 用のバッファレジスタです。バッファ機能が無効になるか (コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の BUF1, BUF 0:bit3, bit2=11<sub>B</sub>) またはフリーランタイマが停止すると, アウトプットコンペアバッファレジスタの値は, 直ちにアウトプットコンペアレジスタへ転送されます。バッファ機能が有効になると (コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の BUF1, 0:bit3, bit2=00<sub>B</sub>), 値はコンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の転送選択ビット (BTS1, 0:bit14, bit13) に従ってコンペアクリア一致時またはゼロ検出時に転送されます。

このレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。

## ■ アウトプットコンペアレジスタ (OCCPH, OCCPL0 ~ OCCPL5)

図 14.4-11 アウトプットコンペアレジスタ (OCCPH, OCCPL0 ~ OCCPL5)

# アウトプットコンペアレジスタ (上位)

	bit 15	14	13	12	11	10	9	8	
	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	OCCPH0~OCCPH5
リード/ライト ⇨	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	

# アウトプットコンペアレジスタ (下位)

	bit 7	6	5	4	3	2	1	0	
	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	OCCPL0~OCCPL5
リード/ライト ⇨	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	

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ムのカウント値と比較するために使用する 16 ビットレジスタです。アウトプットコンペアレジスタの初期値は不定なので、タイマの動作を有効にする前にアウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL) に値を設定しなければなりません。

アウトプットコンペアレジスタの値が 16 ビットフリーランタイムのカウント値と一致すると、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ビット (コンペア制御レジスタ下位 OCSL0, OCSL2, OCSL4 の IOP1, IOP0:bit7, bit6) が設定されます。出力レベルが設定されると (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の OTD1, OTD0:bit9, bit8), アウトプットコンペアレジスタ (OCCPH, OCCPL0 ~ OCCPL5) に対応する出力レベル波形ジェネレータ RTO0 ~ RTO5 を反転させることができます。

このレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。

## 14.4.5 コンペア制御レジスタ (OCSH, OCSL0 ~ OCSL5)

コンペア制御レジスタは, RT0 ~ RT5 の出力レベル, 出力許可, 出力レベル反転モード, コンペア動作許可, コンペアー一致割込み許可およびコンペアー一致割込みフラグを制御するために使用します。

### ■ コンペア制御レジスタ, 上位バイト (OCSH1, OCSH3, OCSH5)

図 14.4-12 コンペア制御レジスタ, 上位バイト (OCSH1, OCSH3, OCSH5)

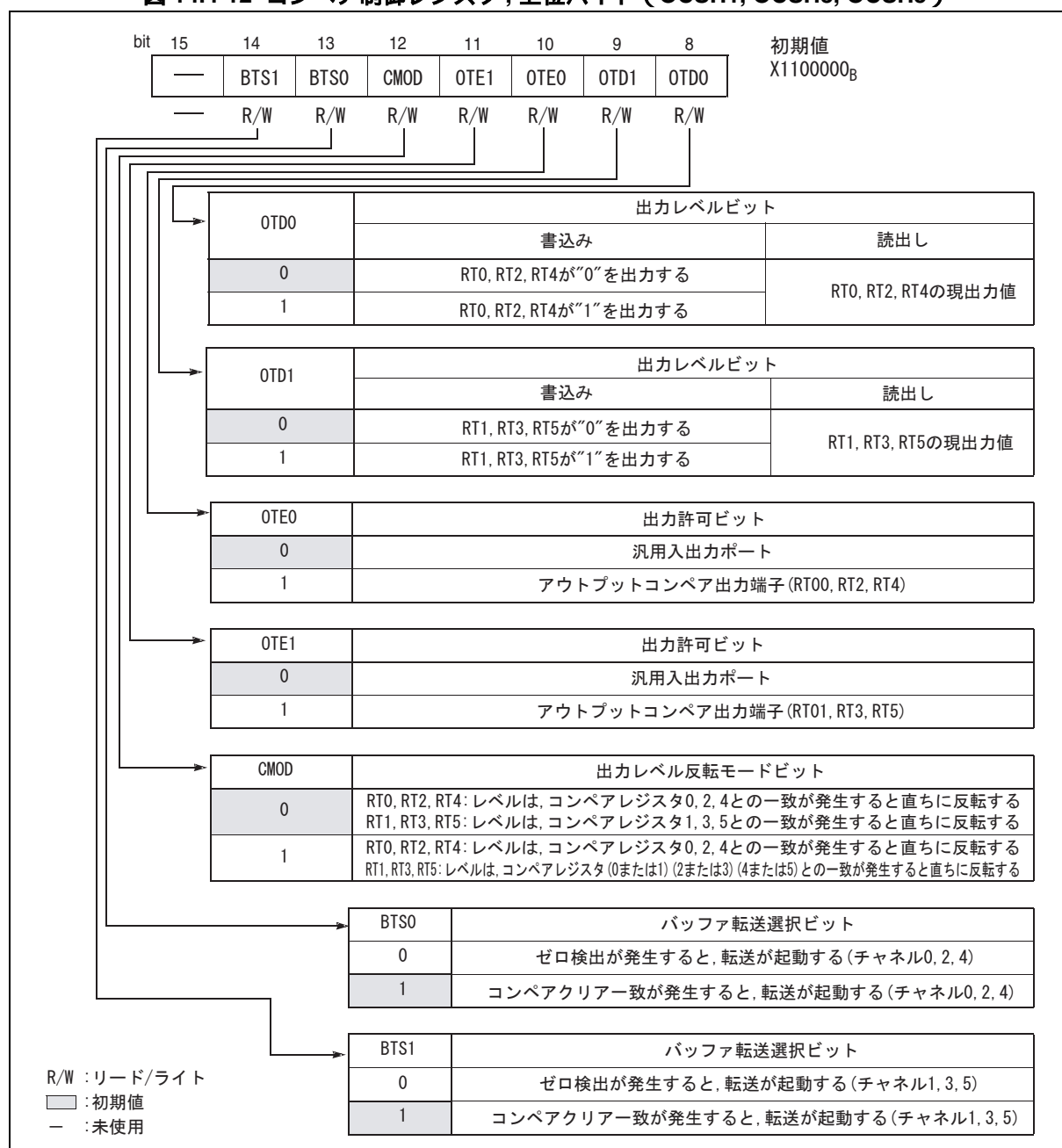


表 14.4-3 コンペア制御レジスタ, 上位バイト (OCSH1, OCSH3, OCSH5)( 1 / 2 )

ビット名		機能
bit15	未使用ビット	<ul style="list-style-type: none"> <li>読出し値は不定です。</li> <li>このビットへの書込みは動作に影響しません。</li> </ul>
bit14	BTS1: バッファ転送 選択ビット	<ul style="list-style-type: none"> <li>このビットは, アウトプットコンペアバッファレジスタ ( OCCPBH, OCCPBL1, OCCPBL3, OCCPBL5 ) からアウトプットコンペアレジスタ ( OCCPH, OCCPL1, OCCPL3, OCCPL5 ) へのデータ転送時期を選択するために使用します。</li> <li>このビットに "0" を設定した場合: データ転送は, 16 ビットフリーランタイムのカウント値 "0" が検出されると起動します。</li> <li>このビットに "1" を設定した場合: データ転送は, 16 ビットフリーランタイムでコンペアクリア一致が発生すると起動します。</li> </ul>
bit13	BTS0: バッファ転送 選択ビット	<ul style="list-style-type: none"> <li>このビットは, アウトプットコンペアバッファレジスタ ( OCCPBH, OCCPBL0, OCCPBL2, OCCPBL4 ) からアウトプットコンペアレジスタ ( OCCPH, OCCPL0, OCCPL2, OCCPL4 ) へのデータ転送時期を選択するために使用します。</li> <li>このビットに "0" を設定した場合: データ転送は, 16 ビットフリーランタイムのカウント値 "0" が検出されると起動します。</li> <li>このビットに "1" を設定した場合: データ転送は, 16 ビットフリーランタイムでコンペアクリア一致が発生すると起動します。</li> </ul>
bit12	CMOD: 出力レベル 反転モード ビット	<ul style="list-style-type: none"> <li>このビットは, 端子出力が有効の間 ( OTE1=1 または OTE0=1 ) に一致が発生した場合に端子出力レベル反転モードを直ちに切り換えるために使用します。</li> <li>このビットに "0" を設定した場合: 端子の出力レベルは, 対応するコンペアレジスタとの一致が発生すると直ちに反転します。 <ul style="list-style-type: none"> <li>RT0, RT2, RT4: レベルは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 が一致すると直ちに反転します。</li> <li>RT1, RT3, RT5: レベルは, 16 ビットフリーランタイムとコンペアレジスタ 1, 3, 5 が一致すると直ちに反転します。</li> </ul> </li> <li>このビットに "1" を設定した場合: コンペアレジスタに対応する RT0, RT2, RT4 の出力レベルは, 本ビットが "0" の場合と同様に反転します。ただし, コンペアレジスタ 1, 3, 5 に対応する RT1, RT3, RT5 の出力レベルは, コンペアレジスタ 0, 2, 4 または 1, 3, 5 で一致が発生すると反転します。コンペアレジスタ 0, 2, 4 と 1, 3, 5 が同じ値の場合は, ただ 1 つのコンペアレジスタが使用される場合と同じ動作になります。 <ul style="list-style-type: none"> <li>RT0, RT2, RT4: レベルは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 が一致すると直ちに反転します。</li> <li>RT1, RT3, RT5: レベルは, 16 ビットフリーランタイムとコンペアレジスタ (0 または 1) (2 または 3) (4 または 5) が一致すると直ちに反転します。</li> </ul> </li> </ul>



表 14.4-3 コンペア制御レジスタ, 上位バイト (OCSH1, OCSH3, OCSH5)( 2 / 2 )

ビット名		機能
bit11	OTE1: 出力許可 ビット	<ul style="list-style-type: none"> <li>このビットは, ポートへの波形ジェネレータ出力 (RTO1, RTO3, RTO5) を許可するために使用します。</li> <li>このビットの初期値は "0" です。</li> </ul> <p>(注意事項)</p> <p>波形ジェネレータが無効 (16 ビットデッドタイム制御レジスタ下位 (DTCRL0, DTCRL1, DTCRL2) の TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6:bit2, bit1=000<sub>B</sub>) の場合は, RTO1, RTO3, RTO5 はアウトプットコンペアと同じ値を出力します。</p>
bit10	OTE0: 出力許可 ビット	<ul style="list-style-type: none"> <li>このビットは, ポートへの波形ジェネレータ出力 (RTO0, RTO2, RTO4) を許可するために使用します。</li> <li>このビットの初期値は "0" です。</li> </ul> <p>(注意事項)</p> <p>波形ジェネレータが無効 (16 ビットデッドタイム制御レジスタ下位 (DTCRL0, DTCRL1, DTCRL2) の TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6:bit2, bit1=000<sub>B</sub>) の場合は, RTO0, RTO2, RTO4 はアウトプットコンペアと同じ値を出力します。</p>
bit9	OTD1: 出力レベル ビット	<ul style="list-style-type: none"> <li>このビットは, アウトプットコンペア 1, 3, 5 (RT1, RT3, RT5) の端子出力レベルを変更するために使用します。</li> <li>コンペア端子出力の初期値は "0" です。</li> <li>値を書き込む場合は, 必ず前もってコンペア動作を停止させてください。このビットの読出し値は, RT1, RT3, RT5 におけるアウトプットコンペア値を示します。</li> </ul>
bit8	OTD0: 出力レベル ビット	<ul style="list-style-type: none"> <li>このビットは, アウトプットコンペア 0, 2, 4 (RT0, RT2, RT4) の端子出力レベルを変更するために使用します。</li> <li>コンペア端子出力の初期値は "0" です。</li> <li>値を書き込む場合は, 必ず前もってコンペア動作を停止させてください。このビットの読出し値は, RT0, RT2, RT4 におけるアウトプットコンペア値を示します。</li> </ul>

## ■ コンペア制御レジスタ , 下位バイト (OCSL0, OCSL2, OCSL4)

図 14.4-13 コンペア制御レジスタ , 下位バイト (OCSL0, OCSL2, OCSL4)

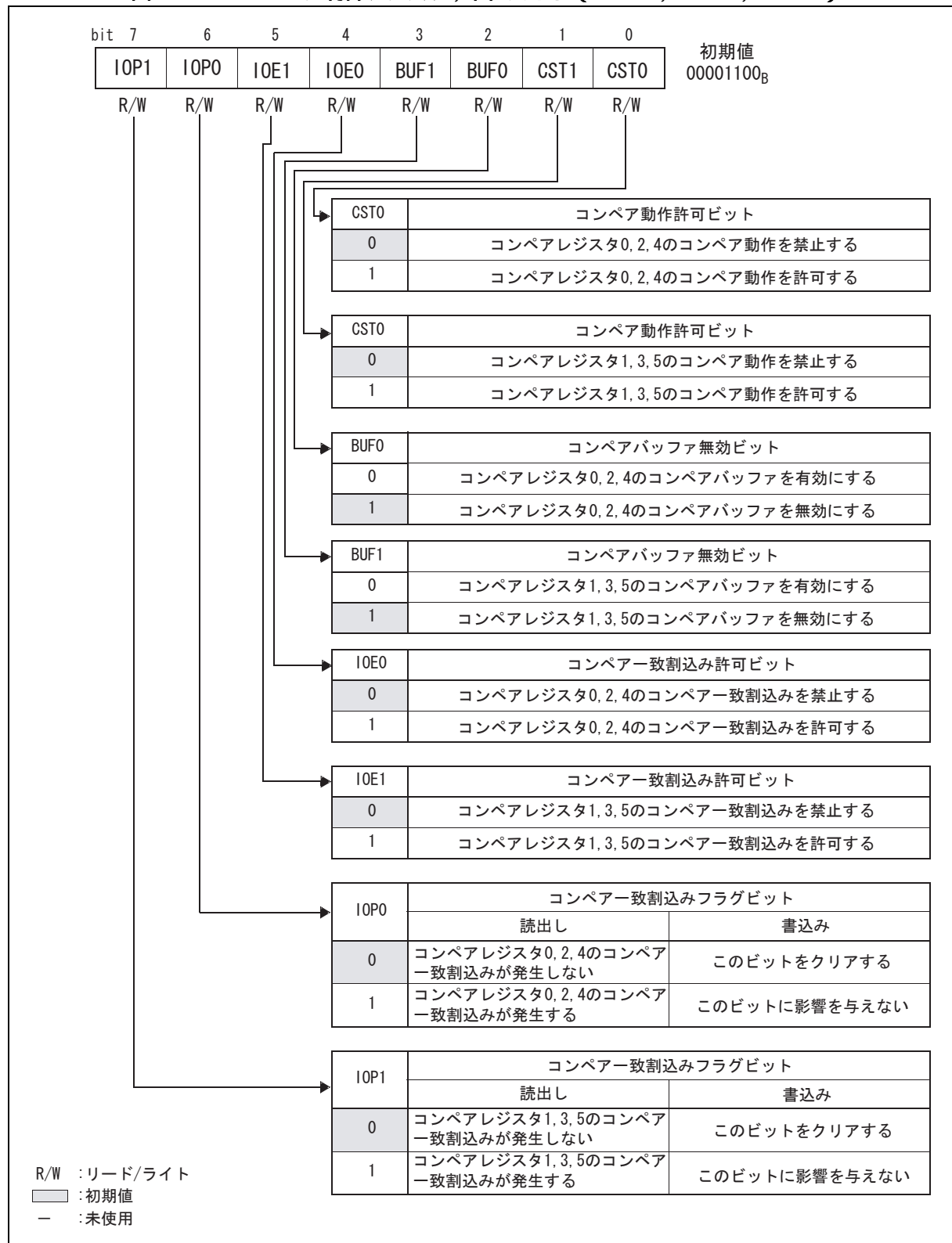


表 14.4-4 コンペア制御レジスタ, 下位バイト (OCSL0, OCSL2, OCSL4) ( 1 / 2 )

ビット名		機能
bit7	IOP1: コンペア一致 割込みフラグ ビット	<ul style="list-style-type: none"> <li>このビットは、コンペアレジスタ 1, 3, 5 が 16 ビットフリーランタイムの値と一致したことを示す割込みフラグです。</li> <li>このビットは、コンペアレジスタ値が 16 ビットフリーランタイム値に一致した場合に "1" が設定されます。</li> <li>コンペア一致割込み許可ビット (IOE1:bit5) が "許可" になっている間にこのビットが設定されると、アウトプットコンペア割込みが発生します。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit6	IOP0: コンペア一致 割込みフラグ ビット	<ul style="list-style-type: none"> <li>このビットは、コンペアレジスタ 0, 2, 4 が 16 ビットフリーランタイムの値と一致したことを示す割込みフラグです。</li> <li>このビットは、コンペアレジスタ値が 16 ビットフリーランタイム値に一致した場合に "1" が設定されます。</li> <li>コンペア一致割込み許可ビット (IOE0:bit4) が "許可" になっている間にこのビットが設定されると、アウトプットコンペア割込みが発生します。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit5	IOE1: コンペア一致 割込み許可 ビット	<ul style="list-style-type: none"> <li>このビットは、コンペアレジスタ 1, 3, 5 のアウトプットコンペア割込みを "許可" にするために使用します。</li> <li>このビットに "1" が書き込まれている間にコンペア一致割込みフラグビット (IOP1:bit7) が設定されると、アウトプットコンペア割込みが発生します。</li> </ul>
bit4	IOE0: コンペア一致 割込み許可 ビット	<ul style="list-style-type: none"> <li>このビットは、コンペアレジスタ 0, 2, 4 のアウトプットコンペア割込みを "許可" にするために使用します。</li> <li>このビットに "1" が書き込まれている間にコンペア一致割込みフラグビット (IOP0:bit7) が設定されると、アウトプットコンペア割込みが発生します。</li> </ul>
bit3	BUF1: コンペアバッ ファ無効 ビット	<ul style="list-style-type: none"> <li>このビットは、アウトプットコンペアレジスタ 1, 3, 5 のバッファ機能を無効にするために使用します。</li> <li>このビットに "0" を設定した場合: バッファ機能が有効になります。</li> </ul>
bit2	BUF0: コンペアバッ ファ無効 ビット	<ul style="list-style-type: none"> <li>このビットは、アウトプットコンペアレジスタ 0, 2, 4 のバッファ機能を無効にするために使用します。</li> <li>このビットに "0" を設定した場合: バッファ機能が有効になります。</li> </ul>
bit1	CST1: コンペア動作 許可ビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットフリーランタイムとコンペアレジスタ 1, 3, 5 の間のコンペア動作を許可するために使用します。</li> <li>コンペア動作を許可にする場合は、必ず前もってコンペアレジスタ 1, 3, 5 とタイマデータレジスタ (TCDTH, L) に値を書き込んでください。</li> </ul> <p>(注意事項) アウトプットコンペアは 16 ビットフリーランタイムクロックと同期化されるので、16 ビットフリーランタイムを停止すると、コンペア動作も停止します。</p>

表 14.4-4 コンペア制御レジスタ, 下位バイト (OCSL0, OCSL2, OCSL4)( 2 / 2 )

ビット名		機能
bit0	CST0: コンペア動作 許可ビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 の間のコンペア動作を許可にするために使用します。</li> <li>コンペア動作を許可にする場合は, 必ず前もってコンペアレジスタ 0, 2, 4 とタイマデータレジスタ (TCDTH, L) に値を書き込んでください。</li> </ul> <p>(注意事項)</p> <p>アウトプットコンペアは 16 ビットフリーランタイムクロックと同期化されるので, 16 ビットフリーランタイムを停止するとゼロ検出, コンペア動作も停止します。</p>

## 14.4.6 インพุットキャプチャデータレジスタ (IPCPH, IPCPL0 ~ IPCPL3)

インพุットキャプチャデータレジスタは、入力波形の有効エッジが検出されたときの 16 ビットデッドタイムのカウント値を保持するために使用します。

### ■ インพุットキャプチャデータレジスタ (IPCPH, IPCPL0 ~ IPCPL3)

図 14.4-14 インพุットキャプチャデータレジスタ (IPCPH, IPCPL0 ~ IPCPL3)

# インプットキャプチャデータレジスタ(上位)

bit	15	14	13	12	11	10	9	8	
	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	IPCPH0~IPCPH3
リード/ライト⇒	R	R	R	R	R	R	R	R	
初期値⇒	X	X	X	X	X	X	X	X	

# インプットキャプチャデータレジスタ(下位)

bit	7	6	5	4	3	2	1	0	
	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	IPCPL0~IPCPL3
リード/ライト⇒	R	R	R	R	R	R	R	R	
初期値⇒	X	X	X	X	X	X	X	X	

このレジスタは、対応する外部端子入力波形の有効エッジが検出されたときの 16 ビットデッドタイム値を格納するために使用します(このレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。このレジスタにデータを書き込むことはできません)。

## 14.4.7 インพุットキャプチャ状態制御 /PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01)

インพุットキャプチャ状態制御 /PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01) は, エッジ選択, 割込み要求許可, 割込み要求フラグおよび PPG 出力を制御するために使用します。またインพุットキャプチャ 0 ~ 3 において検出された有効なエッジを示すためにも使用します。

### ■ インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 上位バイト (ICSH23)

図 14.4-15 インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 上位バイト (ICSH23)

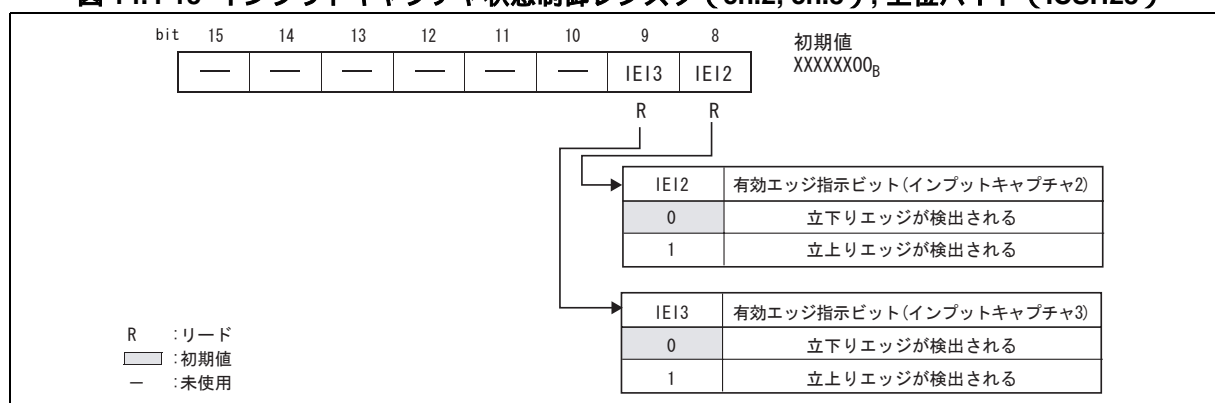


表 14.4-5 インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 上位バイト (ICSH23)

ビット名		機能
bit15 ~ bit10	未使用ビット	<ul style="list-style-type: none"> <li>読み出し値は不定です。</li> <li>このビットへの書き込みは動作に影響しません。</li> </ul>
bit9	IEI3: 有効エッジ 指示ビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> <li>このビットは, キャプチャレジスタ 3 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。</li> <li>立下りエッジが検出されると, このビットに "0" が書き込まれます。</li> <li>立上りエッジが検出されると, このビットに "1" が書き込まれます。</li> <li>このビットは読み出し専用ビットです。</li> </ul> <p>(注意事項)</p> <p>インพุットキャプチャ状態制御レジスタ下位 (ICSL23) の EG31, EG30:bit3, 2=00<sub>B</sub> の場合, 読み出し値は意味がありません。</p>
bit8	IEI2: 有効エッジ 指示ビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> <li>このビットは, キャプチャレジスタ 2 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。</li> <li>立下りエッジが検出されると, このビットに "0" が書き込まれます。</li> <li>立上りエッジが検出されると, このビットに "1" が書き込まれます。</li> <li>このビットは読み出し専用ビットです。</li> </ul> <p>(注意事項)</p> <p>インพุットキャプチャ状態制御レジスタ下位 (ICSL23) の EG21, EG20:bit1, bit0=00<sub>B</sub> の場合, 読み出し値は意味がありません。</p>

## ■ インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 下位バイト (ICSL23)

図 14.4-16 インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 下位バイト (ICSL23)

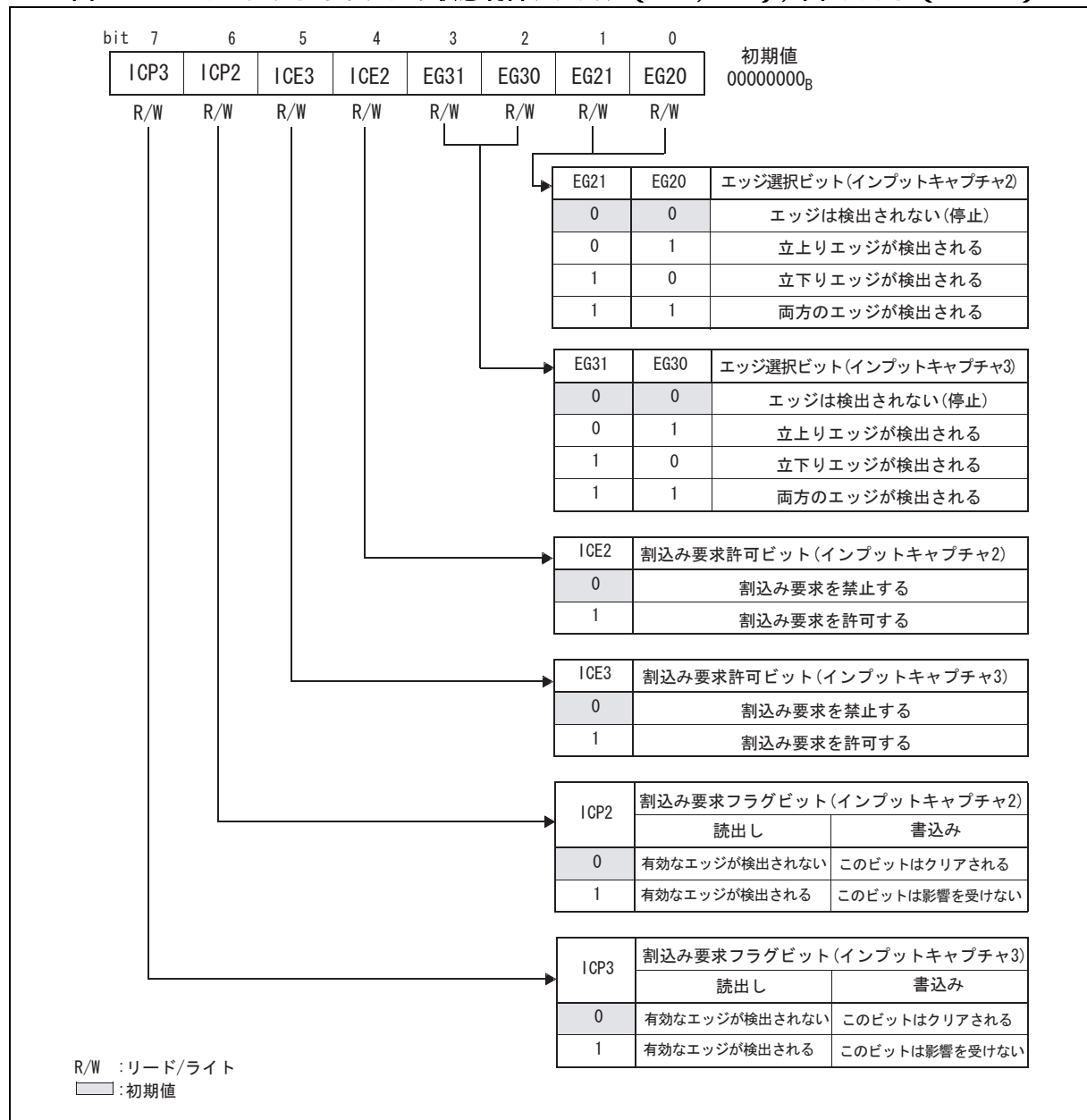


表 14.4-6 インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 下位バイト (ICSL23)

ビット名		機能
bit7	ICP3: 割込み要求フ ラグビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 3 の割込み要求フラグとして使用します。</li> <li>このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。</li> <li>割込み許可ビット (ICE3:bit5) が設定されている間に有効エッジが検出されると、直ちに割込みを生成できます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit6	ICP2: 割込み要求フ ラグビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 2 の割込み要求フラグとして使用します。</li> <li>このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。</li> <li>割込み許可ビット (ICE2:bit4) が設定されている間に有効エッジが検出されると、直ちに割込みを生成できます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit5	ICE3: 割込み要求許 可ビット (イン พุットキャ プチャ 3)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 3 のインพุ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。</li> <li>このビットに "1" が設定されている間に割込みフラグ (ICP3:bit7) が設定されると、インพุットキャプチャ割込みが生成されます。</li> </ul>
bit4	ICE2: 割込み要求許 可ビット (イン พุットキャ プチャ 2)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 2 のインพุ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。</li> <li>このビットに "1" が設定されている間に割込みフラグ (ICP2:bit6) が設定されると、インพุットキャプチャ割込みが生成されます。</li> </ul>
bit3, bit2	EG31, EG30: エッジ選択 ビット (イン พุットキャ プチャ 3)	<ul style="list-style-type: none"> <li>これらのビットは、インพุットキャプチャ 3 の外部入力の有効エッジ極性を指定するために使用します。</li> <li>これらのビットは、インพุットキャプチャ動作を有効にするためにも使用します。</li> </ul>
bit1, bit0	EG21, EG20: エッジ選択 ビット (イン พุットキャ プチャ 2)	<ul style="list-style-type: none"> <li>これらのビットは、インพุットキャプチャ 2 の外部入力の有効エッジ極性を指定するために使用します。</li> <li>これらのビットは、インพุットキャプチャ動作を有効にするためにも使用します。</li> </ul>



## ■ PPG 出力制御 / インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 上位バイト (PICS01)

図 14.4-17 PPG 出力制御 / インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 上位バイト (PICS01)

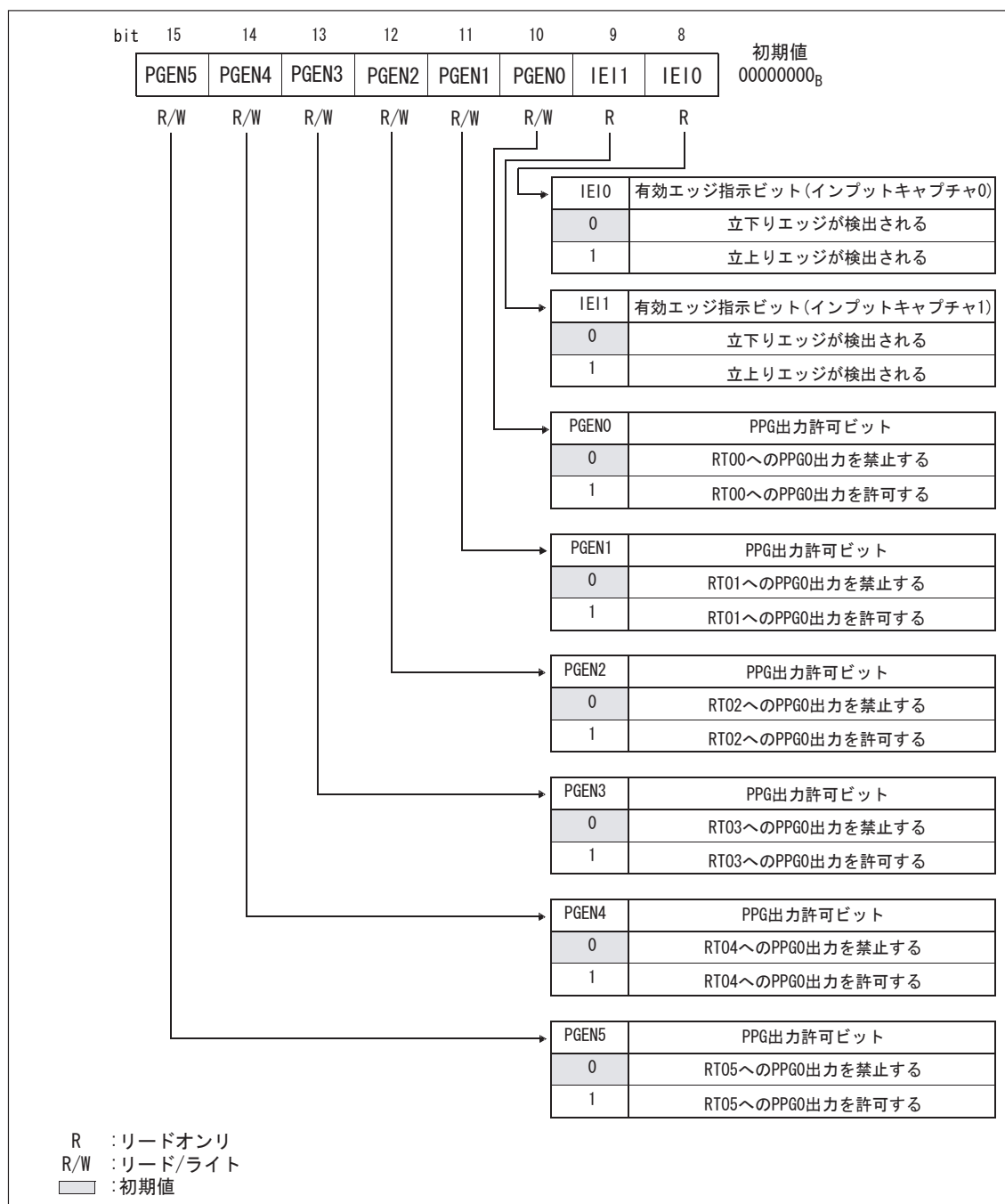


表 14.4-7 PPG 出力制御 / インプットキャプチャ状態制御レジスタ (ch.0, ch.1), 上位バイト (PICSH01)

ビット名		機能
bit15 ~ bit10	PGEN5 ~ PGEN0: PPG 出力許可 ビット	<ul style="list-style-type: none"> <li>これらのビットは, RTO0 ~ RTO5 への PPG0 出力を選択するために使用します。</li> </ul>
bit9	IEI1: 有効エッジ指 示ビット (イ ンプットキャ プチャ 1)	<ul style="list-style-type: none"> <li>このビットは, キャプチャレジスタ 1 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。</li> <li>立下りエッジが検出されると, このビットに "0" が書き込まれます。</li> <li>立上りエッジが検出されると, このビットに "1" が書き込まれます。</li> <li>このビットは読出し専用ビットです。</li> </ul> <p>(注意事項)</p> <p>インプットキャプチャ状態制御レジスタ下位 (PICSL01) の EG11, EG10:bit3, bit2=00<sub>B</sub> の場合, 読出し値は意味がありません。</p>
bit8	IEI0: 有効エッジ指 示ビット (イ ンプットキャ プチャ 0)	<ul style="list-style-type: none"> <li>このビットは, キャプチャレジスタ 0 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。</li> <li>立下りエッジが検出されると, このビットに "0" が書き込まれます。</li> <li>立上りエッジが検出されると, このビットに "1" が書き込まれます。</li> <li>このビットは読出し専用ビットです。</li> </ul> <p>(注意事項)</p> <p>インプットキャプチャ状態制御レジスタ下位 (PICSL01) の EG01, EG00:bit1, bit0=00<sub>B</sub> の場合, 読出し値は意味がありません。</p>

## ■ インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 下位バイト (PICSL01)

図 14.4-18 インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 下位バイト (PICSL01)

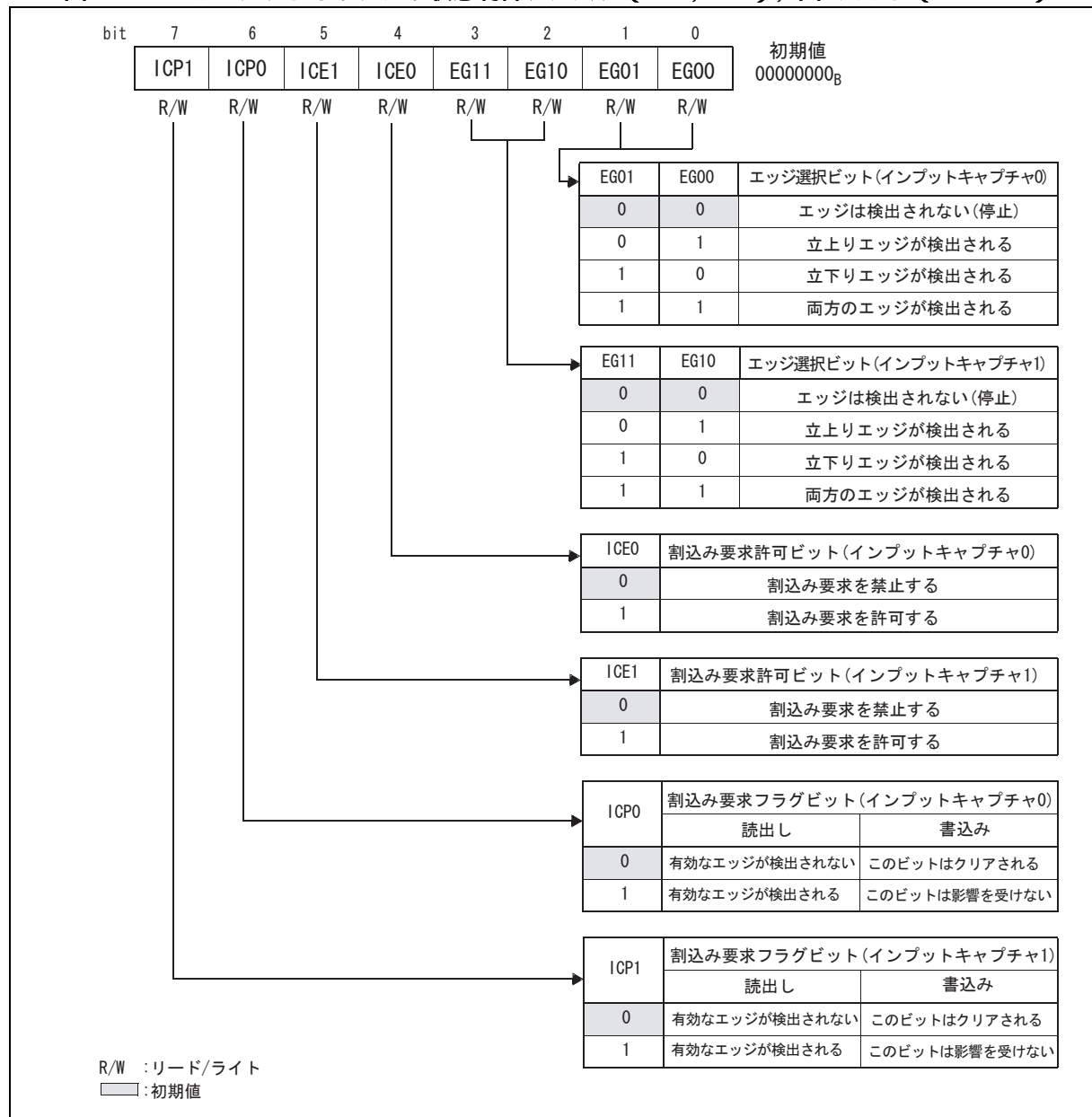


表 14.4-8 インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 下位バイト (PICSL01)

ビット名		機能
bit7	ICP1: 割込み要求フ ラグビット (インพุット キャプチャ 1)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 1 の割込み要求フラグとして使用します。</li> <li>このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。</li> <li>割込み許可ビット (ICE1:bit5) が設定されている間に有効エッジが検出されると、直ちに割込みが生成されます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit6	ICP0: 割込み要求フ ラグビット (インพุット キャプチャ 0)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 0 の割込み要求フラグとして使用します。</li> <li>このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。</li> <li>割込み許可ビット (ICE0:bit4) が設定されている間に有効エッジが検出されると、直ちに割込みが生成されます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト時は、必ず "1" が読み出されます。</li> </ul>
bit5	ICE1: 割込み要求許 可ビット (イン พุットキャ プチャ 1)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 1 のインพุ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。</li> <li>このビットに "1" が設定されている間に割込みフラグ (ICP1:bit7) が設定されると、インพุットキャプチャ 1 割込みが生成されます。</li> </ul>
bit4	ICE0: 割込み要求許 可ビット (イン พุットキャ プチャ 0)	<ul style="list-style-type: none"> <li>このビットは、インพุットキャプチャ 0 のインพุ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。</li> <li>このビットに "1" が設定されている間に割込みフラグ (ICP0:bit6) が設定されると、インพุットキャプチャ 0 割込みが生成されます。</li> </ul>
bit3, bit2	EG11, EG10: エッジ選択 ビット (イン พุットキャ プチャ 1)	<ul style="list-style-type: none"> <li>これらのビットは、インพุットキャプチャ 1 の外部入力の有効エッジ極性を指定するために使用します。</li> <li>これらのビットは、インพุットキャプチャ 1 の動作を有効にするためにも使用します。</li> </ul>
bit1, bit0	EG01, EG00: エッジ選択 ビット (イン พุットキャ プチャ 0)	<ul style="list-style-type: none"> <li>これらのビットは、インพุットキャプチャ 0 の外部入力の有効エッジ極性を指定するために使用します。</li> <li>これらのビットは、インพุットキャプチャ 0 の動作を有効にするためにも使用します。</li> </ul>



### 14.4.9 16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2)

16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) は、波形ジェネレータの動作モード、割込み要求許可、割込み要求フラグ、GATE 信号許可および出力レベル極性を制御するため使用します。

#### ■ 16 ビットデッドタイム制御レジスタ、下位バイト (DTCR0)

図 14.4-20 16 ビットデッドタイム制御レジスタ、下位バイト (DTCR0)

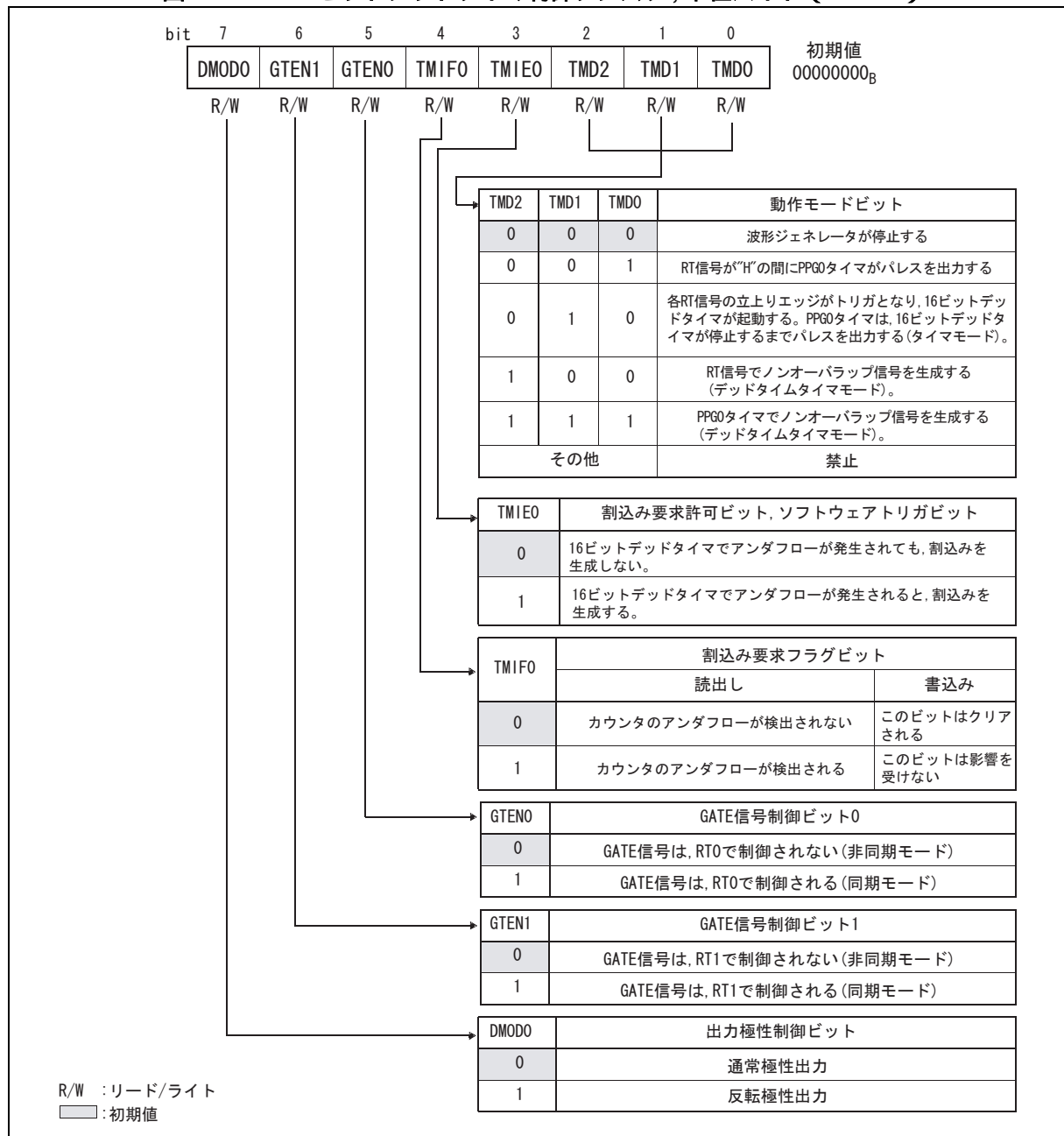


表 14.4-9 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR0) ( 1 / 2 )

ビット名		機能
bit7	DMOD0: 出力極性制御 ビット	<ul style="list-style-type: none"> <li>このビットは, デッドタイムタイマモードにおいて U/V/W の出力極性を設定するために使用します。</li> <li>このビットを設定すると, U/V/W の出力極性は反転します。</li> </ul> <p>( 注意事項 )</p> <p>このビットは, デッドタイムタイマモードが選択されていない場合 ( TMD2: bit2=0 ) は意味がありません。</p>
bit6	GTEN1: GATE 信号制 御ビット 1	<ul style="list-style-type: none"> <li>このビットは, RT1 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit5	GTEN0: GATE 信号制 御ビット 0	<ul style="list-style-type: none"> <li>このビットは, RT0 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit4	TMIF0: 割込み要求フ ラグビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイムの割込み要求フラグとして使用します。</li> <li>このビットは, 16 ビットデッドタイム 0, 2 でアンダフローが発生すると "1" が設定されます。</li> <li>このビットに "0" を書き込むと, このビットはクリアされます。"1" を書き込んでもこのビットは影響を受けません。</li> <li>リードモディファイライト時は, 必ず "1" が読み出されます。</li> </ul> <p>( 注意事項 )</p> <p>このビットは, TMD2 ~ TMD0:bit2 ~ bit0 ) が "000<sub>B</sub>" または "001<sub>B</sub>" の場合のみ機能し, ほかの値の場合は必ず "0" になります。</p> <ul style="list-style-type: none"> <li>ソフトウェアクリア ( "0" 書き込み ) とハードウェアセット ( 16 ビットデッドタイム 0, 2 でアンダフローが発生する ) が同時に発生した場合は, ソフトウェアクリアがハードウェアセットよりも優先され, このビットはクリアされます。</li> </ul>
bit3	TMIE0: 割込み要求許 可ビット, ソ フトウェアト リガビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用されます。</li> <li>TMD2 ~ TMD0:bit2 ~ bit0 が "000<sub>B</sub>" または "001<sub>B</sub>" の場合, このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると, 16 ビットデッドタイムがトリガとなり, 値がリロードされ, ダウンカウントが開始します。</li> <li>このビットが "1" であり, 割込み要求フラグビット ( TMIF0:bit4 ) が "1" の場合, 割込み要求が CPU へ送られます。</li> </ul> <p>( 注意事項 )</p> <p>16 ビットデッドタイムを再度トリガとする場合には, このビットに "1" を書き込む前に必ず "0" を書き込んでください。</p>

表 14.4-9 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR0)( 2 / 2 )

ビット名		機能
bit2 ~ bit0	TMD2 ~ TMD0: 動作モード ビット	<ul style="list-style-type: none"> <li>これらのビットは、波形ジェネレータの動作モードを選択するために使用します。</li> <li>TMD2 ~ TMD0:bit2 ~ bit0 が "000<sub>B</sub>" の場合、アウトプットコンペアの RT0 と RT1 の信号は、RTO0 と RTO1 のそれぞれから出力されます。また、16 ビットデッドタイムはリロードタイマとしても使用できます。</li> <li>TMD2 ~ TMD0:bit2 ~ bit0 が "001<sub>B</sub>" の場合、アウトプットコンペアの RT0 と RT1 の信号は、PPG0 出力が禁止（PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位（PICSH01）の PGEN0:bit10=0, PGEN1:bit11=0）になると、RTO0 と RTO1 のそれぞれから出力されます。また、16 ビットデッドタイムはリロードタイマとしても使用できます。</li> </ul> <p>（注意事項）</p> <p>デッドタイムタイマモードで波形ジェネレータを動作させるには、必ず RT1 に対して 2 チャンネルモード（コンペア制御レジスタ上位（OCSH1）の CMOD:bit12=1）を選択してください。</p> <p>TMD2 ~ TMD0:bit2 ~ bit0 が "111<sub>B</sub>" の場合、RT0 出力と RT1 出力は、PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位（PICSH01）の PGEN0:bit10=0, PGEN1:bit11=0 の設定に依存しません。</p>



## ■ 16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)

図 14.4-21 16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)

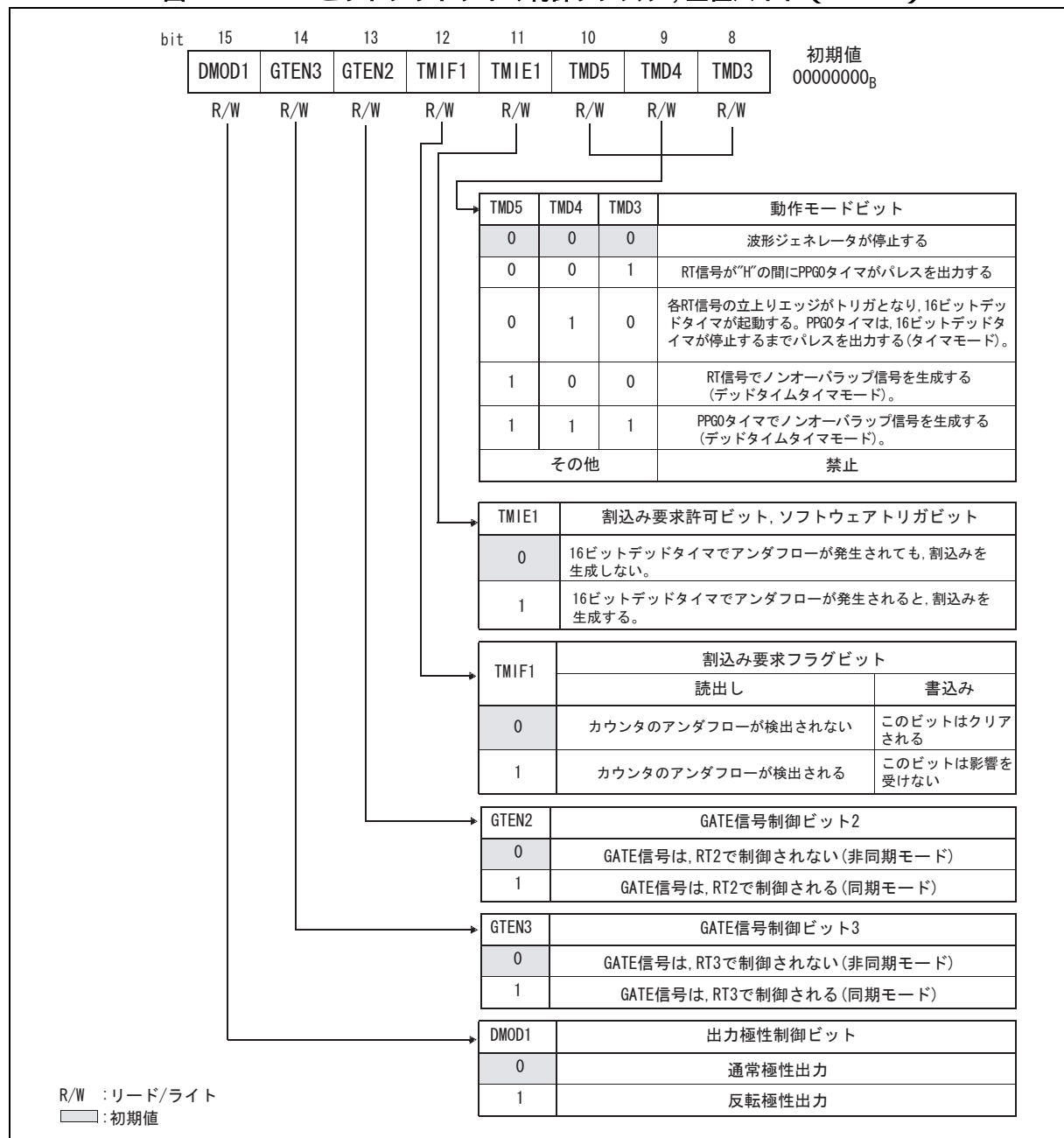


表 14.4-10 16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)( 1 / 2 )

ビット名		機能
bit15	DMOD1: 出力極性制御 ビット	<ul style="list-style-type: none"> <li>このビットは, デッドタイムタイマモードにおいて U/V/W の出力極性を設定するために使用します。</li> <li>このビットを設定すると, U/V/W の出力極性は反転します。</li> <li>このビットは, デッドタイムタイマモードが選択されていない場合 (TMD5: bit10=0) は意味がありません。</li> </ul>
bit14	GTEN3: GATE 信号 制御ビット 3	<ul style="list-style-type: none"> <li>このビットは, RT3 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit13	GTEN2: GATE 信号 制御ビット 2	<ul style="list-style-type: none"> <li>このビットは, RT2 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit12	TMIF1: 割込み要求 フラグビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイムの割込み要求フラグとして使用します。</li> <li>このビットは, 16 ビットデッドタイム 1 でアンダフローが発生すると "1" が設定されます。</li> <li>このビットに "0" を書き込むと, このビットはクリアされます。"1" を書き込んでもこのビットは影響を受けません。</li> <li>リードモディファイライト時は必ず "1" が読み出されます。</li> </ul> <p>(注意事項)</p> <p>このビットは, TMD5 ~ TMD3: ビット 10 ~ 8 が "000<sub>B</sub>" または "001<sub>B</sub>" の場合のみ機能し, ほかの値の場合は必ず "0" になります。</p> <p>ソフトウェアクリア ("0" 書き込み) とハードウェアセット (16 ビットデッドタイム 1 でアンダフローが発生する) が同時に発生した場合は, ソフトウェアクリアがハードウェアセットよりも優先され, このビットをクリアします。</p>
bit11	TMIE1: 割込み要求許 可ビット, ソフトウェア トリガビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用します。</li> <li>TMD5 ~ TMD3: bit10 ~ bit8 が "000<sub>B</sub>" または "001<sub>B</sub>" の場合, このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると, 16 ビットデッドタイムがトリガとなり, 値がリロードされ, ダウンカウントが開始します。</li> <li>このビットが "1" であり, 割込みフラグビット (TMIF1: bit12) が "1" の場合, 割込み要求が CPU へ送られます。</li> </ul> <p>(注意事項)</p> <p>16 ビットデッドタイムを再度トリガとする場合には, このビットに "1" を書き込む前に必ず "0" を書き込んでください。</p>

表 14.4-10 16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)( 2 / 2 )

ビット名		機能
bit10 ~ bit8	TMD5 ~ TMD3: 動作モード ビット	<ul style="list-style-type: none"> <li>これらのビットは、波形ジェネレータの動作モードを選択するために使用します。</li> <li>TMD5 ~ TMD3:bit10 ~ bit8 が "000<sub>B</sub>" の場合、アウトプットコンペアの RT2 と RT3 の信号は、RTO2 と RTO3 のそれぞれから出力されます。また、16 ビットデッドタイムはリロードタイマとしても使用できます。</li> <li>TMD5 ~ TMD3:bit10 ~ bit8 が "001<sub>B</sub>" の場合、アウトプットコンペアの RT2 と RT3 は、PPG0 出力が禁止（PPG 出力制御 / インพุットキャプチャ状態制御レジスタ上位（PICSH01）の PGEN2:bit12=0, PGEN3:bit13=0 になると、RTO2 と RTO3 のそれぞれから出力されます。また、16 ビットデッドタイムはリロードタイマとしても使用できます。</li> </ul> <p>（注意事項）</p> <p>デッドタイムタイマモードで波形ジェネレータを動作させるには、必ず RT3 に対して 2 チャンネルモード（コンペア制御レジスタ上位（OCSH3）の CMOD:bit12=1 を選択してください。</p> <p>TMD5 ~ TMD3:bit10 ~ bit8 が "111<sub>B</sub>" の場合、RTO2 の出力と RTO3 の出力は、PPG 出力制御インพุットキャプチャ状態制御レジスタ上位（PICSH01）の PGEN2:bit12=0, PGEN3:bit13=0 の設定に依存しません。</p>

## ■ 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)

図 14.4-22 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)

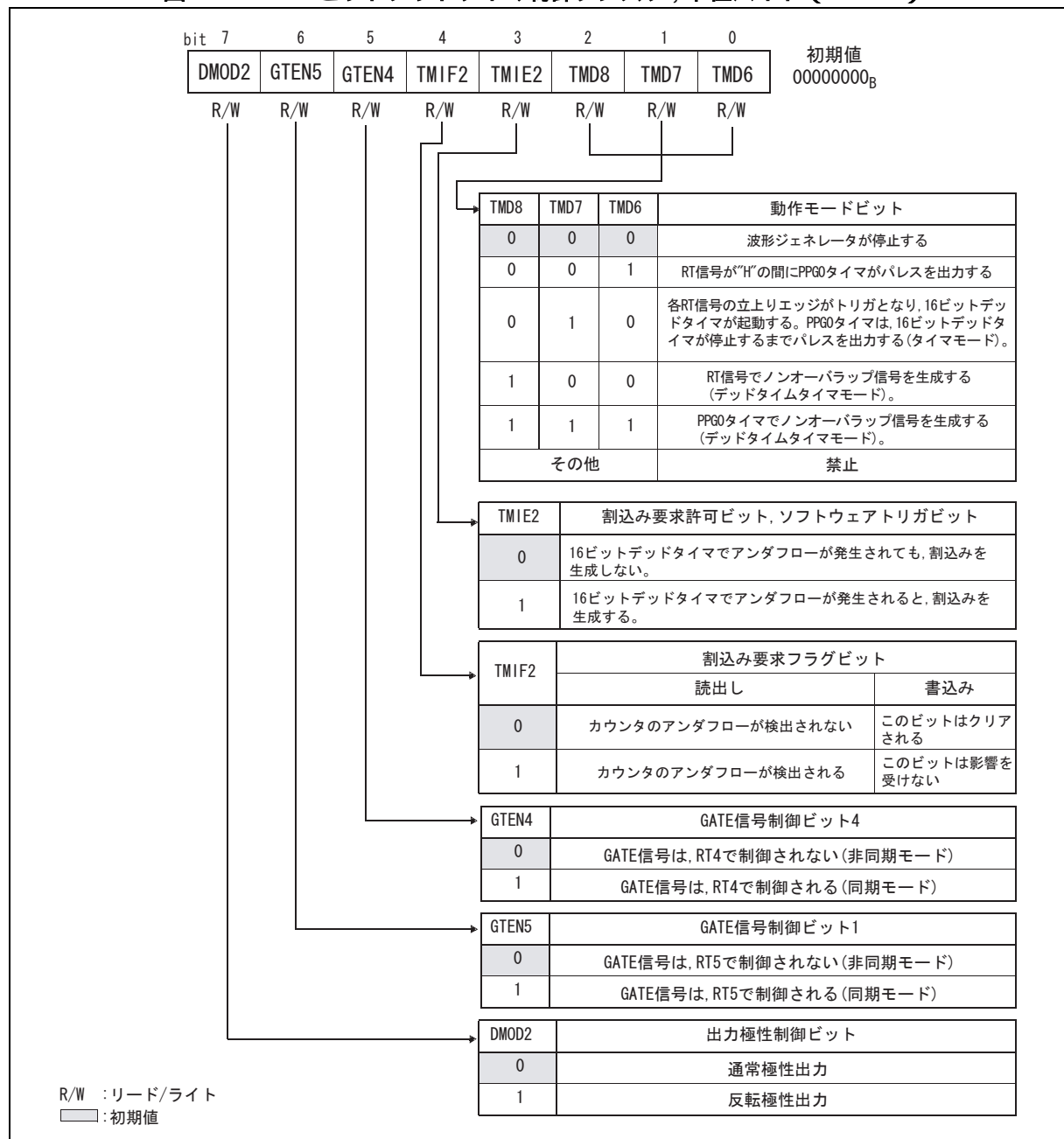


表 14.4-11 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)( 1 / 2 )

ビット名		機能
bit7	DMOD2: 出力極性制御 ビット	<ul style="list-style-type: none"> <li>このビットは, デッドタイムタイマモードにおいて U/V/W の出力極性を設定するために使用します。</li> <li>このビットを設定すると, U/V/W の出力極性は反転します。</li> </ul> (注意事項) このビットは, デッドタイムタイマモードが選択されていない場合 (TMD8: bit2=0) は意味がありません。
bit6	GTEN5: GATE 信号制 御ビット 5	<ul style="list-style-type: none"> <li>このビットは, RT5 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit5	GTEN4: GATE 信号制 御ビット 4	<ul style="list-style-type: none"> <li>このビットは, RT4 で PPG0 タイマの GATE 信号出力を制御するために使用します。</li> </ul>
bit4	TMIF2: 割込み要求フ ラグビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイムの割込み要求フラグとして使用します。</li> <li>このビットは, 16 ビットデッドタイム 1 でアンダフローが発生すると "1" が設定されます。</li> <li>このビットに "0" を書き込むと, このビットはクリアされます。"1" を書き込んでもこのビットは影響を受けません。</li> <li>リードモディファイライト時は必ず "1" が読み出されます。</li> </ul> (注意事項) このビットは, TMD8 ~ TMD6:bit2 ~ bit0 が "000 <sub>B</sub> " または "001 <sub>B</sub> " の場合のみ機能し, ほかの値の場合は必ず "0" になります。 ソフトウェアクリア ("0" 書き込み) とハードウェアセット (16 ビットデッドタイム 1 でアンダフローが発生する) が同時に発生した場合は, ソフトウェアクリアがハードウェアセットよりも優先され, このビットをクリアします。
bit3	TMIE2: 割込み要求許 可ビット, ソ フトウェアト リガビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットデッドタイム 16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用します。</li> <li>TMD8 ~ TMD6:bit2 ~ bit0 が "000<sub>B</sub>" または "001<sub>B</sub>" の場合, このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると, 16 ビットデッドタイムがトリガとなり, 値がリロードされ, ダウンカウントが開始します。</li> <li>このビットが "1" であり, 割込みフラグビット (TMIF2:bit4) が "1" の場合, 割込み要求が CPU へ送られます。</li> </ul> (注意事項) 16 ビットデッドタイムを再度トリガとする場合には, このビットに "1" を書き込む前に必ず "0" を書き込んでください。

表 14.4-11 16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)( 2 / 2 )

ビット名		機能
bit2 ~ bit0	TMD8 ~ TMD6: 動作モード ビット	<ul style="list-style-type: none"> <li>これらのビットは, 波形ジェネレータの動作モードを選択するために使用します。</li> <li>TMD8 ~ TMD6:bit2 ~ bit0 が "000<sub>B</sub>" の場合, アウトプットコンペアの RT4 と RT5 の信号は, RTO4 と RTO5 のそれぞれから出力されます。また, 16 ビットデッドタイムはリロードタイマとしても使用できます。</li> <li>TMD8 ~ TMD6:bit2 ~ bit0 が "001<sub>B</sub>" の場合, アウトプットコンペアの RT4 と RT5 は, PPG0 出力が禁止 (PPG 出力制御 / インพุットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN4:bit14=0, PGEN5:bit15=0 になると, RTO4 と RTO5 のそれぞれから出力されます。また, 16 ビットデッドタイムはリロードタイマとしても使用できます。</li> </ul> <p>(注意事項)</p> <p>デッドタイムタイマモードで波形ジェネレータを動作させるには, 必ず RT5 に対して 2 チャンネルモード (コンペア制御レジスタ上位 (OCSH5) の CMOD:bit12=1 を選択してください。</p> <p>TMD8 ~ TMD6:bit2 ~ bit0 が "111<sub>B</sub>" の場合, RTO4 の出力と RTO5 の出力は, PPG 出力制御インพุットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN4:bit14=0, PGEN5:bit15=0 の設定に依存しません。</p>

## 14.4.10 波形制御レジスタ (SIGCR)

波形制御レジスタは、動作クロック周波数、ノイズキャンセル機能有効、DTTI 入力有効および DTTI 割込みを制御するために使用します。

### ■ 波形制御レジスタ (SIGCR)

図 14.4-23 波形制御レジスタ (SIGCR)

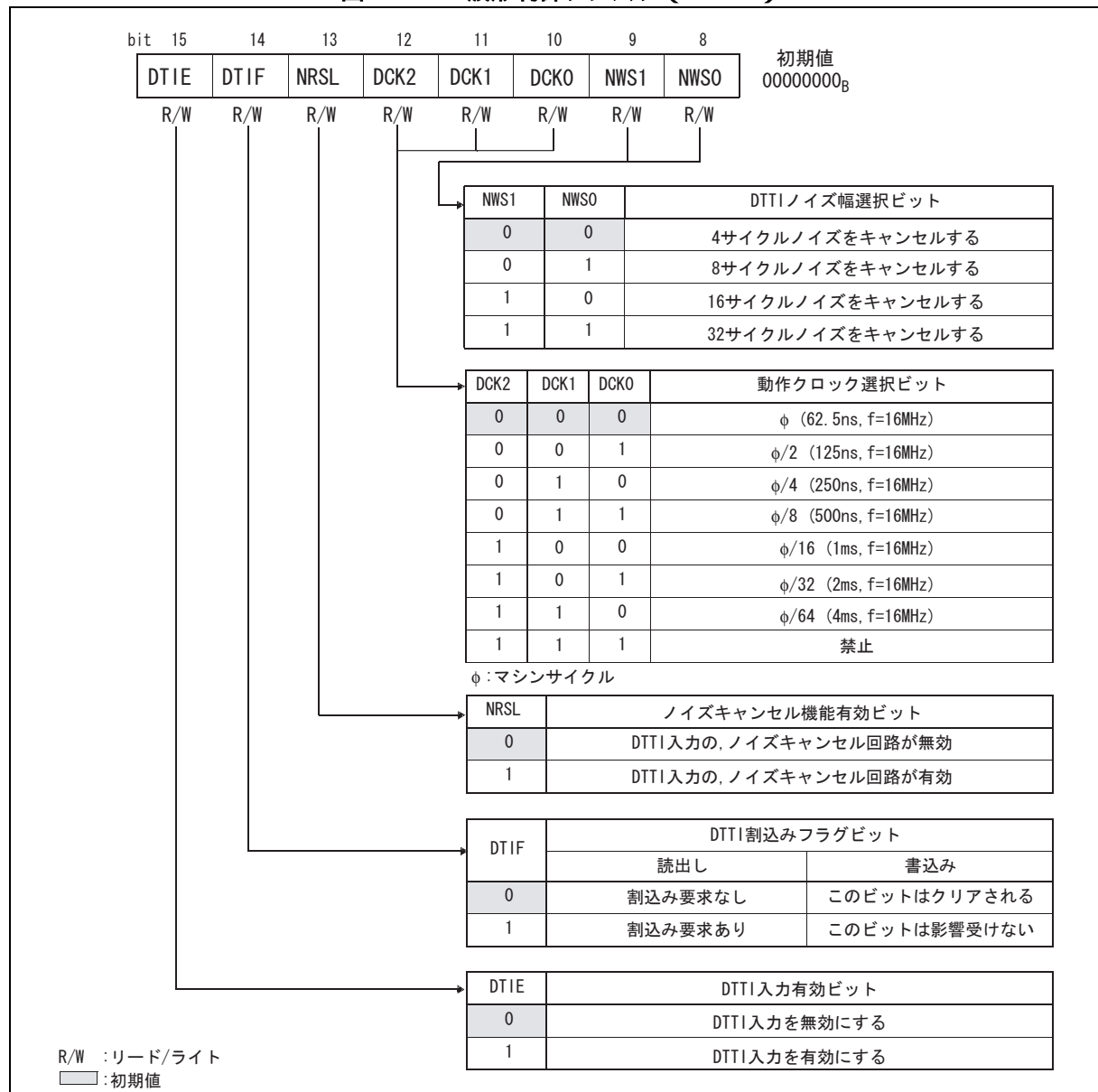


表 14.4-12 波形制御レジスタ (SIGCR)

ビット名		機 能
bit15	DTIE: DTTI 入力有効 ビット	<ul style="list-style-type: none"> <li>このビットは, RTO0 ~ RTO5 端子の出力レベル制御用 DTTI 端子を有効にするために使用します。</li> </ul>
bit14	DTIF: DTTI 割込み フラグビット	<ul style="list-style-type: none"> <li>このビットは, DTTI の割込みフラグです。</li> <li>DTTI 入力が無効になり (DTIE:bit5=1), DTTI の low レベルが検出されると, このビットが設定され, 割込み要求が CPU へ送られます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト時では, 必ず "1" が読み出されます。</li> </ul> <p>(注意事項)</p> <p>ノイズキャンセル機能が有効になった場合 (NRSL:bit13=1), ノイズパルス幅が渡されると, このビットには "1" が設定されます。</p> <p>ソフトウェアクリア ("0" 書込み) とハードウェアセット (DTTI の low レベル検出) が同時に発生した場合は, ソフトウェアクリアがハードウェアリセットよりも優先され, このビットはクリアされます。</p>
bit13	NRSL: ノイズキャン セル機能有効 ビット	<ul style="list-style-type: none"> <li>このビットは, ノイズキャンセル機能を有効にするために使用します。</li> <li>ノイズキャンセル回路は, カウンタでオーバフローが発生するまで low レベルが保持されると, DTTI 入力信号を受取ります。カウンタは, low レベル入力で作られる n ビットカウンタです。n は, NWS1, NWS0 ビット:9, 8 の設定に基づいて 2, 3, 4 または 5 のいずれかの値になります。</li> </ul> <p>(注意事項)</p> <p>ノイズパルス幅をキャンセルするには, 約 <math>2^n</math> マシンサイクルが必要になります。</p> <p>ノイズキャンセル回路を選択すると, 内部クロックが停止するモード (停止モードなど) 時は入力が無効になります。</p>
bit12 ~ bit10	DCK2 ~ DCK0: 動作クロック 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, 16 ビットデッドタイムの動作クロックを選択するために使用します。</li> </ul>
bit9, bit8	NWS1, NWS 0: DTTI ノイズ幅 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, 除去する DTTI 端子ノイズパルス幅を選択するために使用します。</li> </ul>



## 14.5 多機能タイマ割込み

多機能タイマは、16 ビットフリーランタイム割込み、16 ビットアウトプットコンペア割込み、16 ビットインプットキャプチャ割込みおよび波形ジェネレータ割込みを生成できます。

### ■ 16 ビットフリーランタイム割込み

16 ビットフリーランタイムの割込み制御ビットと割込み要因を表 14.5-1 に示します。

表 14.5-1 16 ビットフリーランタイムの割込み制御ビットと割込み要因

	16 ビットフリーランタイム	
	コンペアクリア	ゼロ検出
割込み要求フラグビット	タイマ状態レジスタ上位 (TCCSH) の ICLR:bit9	タイマ状態レジスタ上位 (TCCSH) の IRQZF:bit14
割込み要求許可ビット	タイマ状態レジスタ上位 (TCCSH) の ICRE:bit8	タイマ状態レジスタ上位 (TCCSH) の IRQZE:bit13
割込み要因	16 ビットフリーランタイム値がコンペアクリアレジスタ (CPCLRH, L) と一致する	16 ビットフリーランタイム値が "0" になる

16 ビットフリーランタイムの値がコンペアクリアレジスタ (CPCLRH, L) と一致すると、タイマ状態制御レジスタ (TCCSH) の ICLR:bit9 に "1" が設定されます。この状態において割込み要求が許可 (TCCSH レジスタの ICRE:bit8=1) になると、割込み要求が割込みコントローラへ出力されます。

タイマ値が "0000<sub>H</sub>" になると、タイマ状態制御レジスタ (TCCSH) の IRQZF:bit14 に "1" が設定されます。この状態において割込み要求が許可 (TCCSH レジスタの IRQZE:bit13=1) になると、割込み要求が割込みコントローラへ出力されます。

### ■ 16 ビットフリーランタイム割込みと EI<sup>2</sup>OS

16 ビットフリーランタイム割込みと EI<sup>2</sup>OS を表 14.5-2 に示します。

表 14.5-2 16 ビットフリーランタイム割込みと EI<sup>2</sup>OS

チャンネル	割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
コンペアクリア *1	#34 (22 <sub>H</sub> )	ICR11	0000BB <sub>H</sub>	FFFF74 <sub>H</sub>	FFFF75 <sub>H</sub>	FFFF76 <sub>H</sub>	
ゼロ検出 *2	#31 (1F <sub>H</sub> )	ICR10	0000BA <sub>H</sub>	FFFF80 <sub>H</sub>	FFFF81 <sub>H</sub>	FFFF82 <sub>H</sub>	

\*1:16 ビットフリーランタイムコンペアクリアの割込み制御レジスタと同じ値が 16 ビット入力キャプチャチャンネル 0, 1 に割当てられます。

\*2:16 ビットフリーランタイム "0" 検出の割込み制御レジスタと同じ値が 16 ビット PPG タイマ 2 に割当てられます。

## ■ 16 ビットアウトプットコンペア割込み

16 ビットアウトプットコンペアの割込み制御ビットと割込み要因を表 14.5-3 に示します。

表 14.5-3 16 ビットアウトプットコンペア 0 ~ 5 の割込み制御ビットと割込み要因

	16 ビットアウトプット コンペア 0, 1	16 ビットアウトプット コンペア 2, 3	16 ビットアウトプット コンペア 4, 5
割込み要求フラグビット	コンペア制御レジスタ下位 (OCSL0) の IOP1, IOP0 (bit7, bit6)	コンペア制御レジスタ下位 (OCSL2) の IOP1, IOP0 (bit7, bit6)	コンペア制御レジスタ下位 (OCSL4) の IOP1, IOP0 (bit7, bit6)
割込み要求許可ビット	コンペア制御レジスタ下位 (OCSL0) の IOE1, IOE0 (bit5, bit4)	コンペア制御レジスタ下位 (OCSL2) の IOE1, IOE0 (bit5, bit4)	コンペア制御レジスタ下位 (OCSL4) の IOE1, IOE0 (bit5, bit4)
割込み原因	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH, L0, L1) と一致する	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH, L2, L3) と一致する	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH, L4, L5) と一致する

16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH, L0 ~ L5) と一致すると、コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の IOP1, IOP0:bit7, bit6 に "1" に設定されます。この状態において割込み要求が許可 (OCSL0, OCSL2, OCSL4 レジスタの IOE1, IOE0:bit5, bit4=1<sub>B</sub>) になると、割込み要求が割込みコントローラへ出力されます。

## ■ 16 ビットアウトプットコンペア割込みと EI<sup>2</sup>OS

16 ビットアウトプットコンペア割込みと EI<sup>2</sup>OS を表 14.5-4 に示します。

表 14.5-4 16 ビットアウトプットコンペア割込みと EI<sup>2</sup>OS

チャンネル	割込み 番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
出力コンペア 0 一致 <sup>*1</sup>	#12(0C <sub>H</sub> )	ICR00	0000B0 <sub>H</sub>	FFFFCC <sub>H</sub>	FFFFCD <sub>H</sub>	FFFFCE <sub>H</sub>	
出力コンペア 1 一致 <sup>*2</sup>	#15(0F <sub>H</sub> )	ICR02	0000B2 <sub>H</sub>	FFFFC0 <sub>H</sub>	FFFFC1 <sub>H</sub>	FFFFC2 <sub>H</sub>	
出力コンペア 2 一致 <sup>*3</sup>	#17(11 <sub>H</sub> )	ICR03	0000B3 <sub>H</sub>	FFFFB8 <sub>H</sub>	FFFFB9 <sub>H</sub>	FFFFBA <sub>H</sub>	
出力コンペア 3 一致 <sup>*4</sup>	#19(13 <sub>H</sub> )	ICR04	0000B4 <sub>H</sub>	FFFFB0 <sub>H</sub>	FFFFB1 <sub>H</sub>	FFFFB2 <sub>H</sub>	
出力コンペア 4 一致 <sup>*5</sup>	#21(15 <sub>H</sub> )	ICR05	0000B5 <sub>H</sub>	FFFA8 <sub>H</sub>	FFFA9 <sub>H</sub>	FFFAA <sub>H</sub>	
出力コンペア 5 一致 <sup>*6</sup>	#23(17 <sub>H</sub> )	ICR06	0000B6 <sub>H</sub>	FFFA0 <sub>H</sub>	FFFA1 <sub>H</sub>	FFFA2 <sub>H</sub>	

\*1:16 ビットアウトプットコンペア 0 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が A/D 変換終了に割当てられています。

\*2:16 ビットアウトプットコンペア 1 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が 16 ビット PPG タイマ 1 に割当てられています。

\*3:16 ビットアウトプットコンペア 2 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が 16 ビット リロードタイマ 1 アンダフローに割当てられています。

\*4:16 ビットアウトプットコンペア 3 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が DTP, 外部割込みチャンネル 0, 1 検出, DTI に割当てられています。

\*5:16 ビットアウトプットコンペア 4 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が DTP, 外部割込みチャンネル 2, 3 検出, DTI に割当てられています。

\*6:16 ビットアウトプットコンペア 5 の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が PWC タイマ 1 に割当てられています。

## ■ 16 ビットインプットキャプチャ割り込み

16 ビットインプットキャプチャの割り込み制御ビットと割り込み要因を表 14.5-5 に示します。

表 14.5-5 16 ビットインプットキャプチャ 0 ～ 3 の割り込み制御ビットと割り込み要因

	16 ビットインプットキャプチャ 0, 1	16 ビットインプットキャプチャ 2, 3
割り込み要求フラグビット	インプットキャプチャ状態制御レジスタ下位 (PICSL01) の ICP1, ICP0 ( bit7, bit 6 )	インプットキャプチャ状態制御レジスタ下位 (ICSL23) の ICP3, ICP2 ( bit7, bit 6 )
割り込み要求許可ビット	インプットキャプチャ状態制御レジスタ下位 (PICSL01) の ICE1, ICE0 ( bit5, bit 4 )	インプットキャプチャ状態制御レジスタ下位 (ICSL23) の ICP3, ICP2 ( bit5, bit4 )
割り込み要因	有効なエッジが IN0, IN1 端子で検出される	有効なエッジが IN2, IN3 端子で検出される

16 ビットインプットキャプチャでは、有効なエッジが IN0 ～ IN3 端子で検出されると、インプットキャプチャ状態制御レジスタ (PICSL01, ICSL23) の ICP3 ～ ICP0: 共に bit7, bit6 に "11<sub>B</sub>" が設定されます。この状態において割り込み要求が許可 (PICSL01, ICSL23 レジスタの ICE3 ～ ICE0: 共に bit5, bit4=11<sub>B</sub>) になると、割り込み要求は割り込みコントローラへ出力されます。

## ■ 16 ビットインプットキャプチャ割り込みと EI<sup>2</sup>OS

16 ビットインプットキャプチャ割り込みと EI<sup>2</sup>OS を表 14.5-6 に示します。

表 14.5-6 16 ビットインプットキャプチャ割り込みと EI<sup>2</sup>OS

チャンネル	割り込み番号	割り込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
インプットキャプチャ 0, 1 <sup>*1</sup>	#33 ( 21 <sub>H</sub> )	ICR11	0000BB <sub>H</sub>	FFFF78 <sub>H</sub>	FFFF79 <sub>H</sub>	FFFF7A <sub>H</sub>	
インプットキャプチャ 2, 3 <sup>*2</sup>	#35 ( 23 <sub>H</sub> )	ICR12	0000BC <sub>H</sub>	FFFF70 <sub>H</sub>	FFFF71 <sub>H</sub>	FFFF72 <sub>H</sub>	

\*1:16 ビットインプットキャプチャ 0, 1 の割り込み制御レジスタと同じ値 (レジスタ番号とアドレス) が 16 ビットフリーランタイムコンペアクリアに割り当てられています。

\*2:16 ビットインプットキャプチャ 2, 3 の割り込み制御レジスタと同じ値 (レジスタ番号とアドレス) がタイムベースタイマに割り当てられています。

## ■ 波形ジェネレータ割込み

波形ジェネレータの割込み制御ビットと割込み要因を表 14.5-7 に示します。

表 14.5-7 波形ジェネレータの割込み制御ビットと割込み要因

	波形ジェネレータ	
	16 ビットデッドタイム 0, 1, 2	DTTI0
割込み要求フラグビット	16 ビットデッドタイム制御レジスタ上位, 下位 (DTCR0 ~ DTCR2) の TMIF0 ~ TMIF2 (上位は bit12, 下位は bit4)	波形制御レジスタ (SIGCR) の DTIF (bit14)
割込み要求許可ビット	16 ビットデッドタイム制御レジスタ上位, 下位 (DTCR0 ~ DTCR2) の TMIE0 ~ TMIE2 (上位は bit11, 下位は bit3)	-
割込み要因	16 ビットデッドタイム 0, 1, 2 アンダフロー	DTTI で low レベルが検出される

波形ジェネレータでは, 16 ビットデッドタイムのアンダフローが発生し, かつ DTCR0 ~ DTCR2 レジスタの TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "000<sub>B</sub>" または "001<sub>B</sub>" のとき, 16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMIF0 ~ TMIF2 (上位は bit12, 下位は bit4) には "1" が設定されます。この状態において割込み要求が許可 (DTCR0 ~ DTCR2 レジスタの TMIE0 ~ TMIE2 (上位は bit11, 下位は bit3)=1) になると, 割込み要求は割込みコントローラへ出力されます。

## ■ 波形ジェネレータ割込みと EI<sup>2</sup>OS

波形ジェネレータ割込みと EI<sup>2</sup>OS を表 14.5-8 に示します。

表 14.5-8 波形ジェネレータ割込みと EI<sup>2</sup>OS

チャネル	割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
16 ビットデッドタイム 0, 1, 2 アンダフロー *1	#29(1D <sub>H</sub> )	ICR09	0000B9 <sub>H</sub>	FFFF88 <sub>H</sub>	FFFF89 <sub>H</sub>	FFFF8A <sub>H</sub>	
DTTI *2	#20(14 <sub>H</sub> )	ICR04	0000B4 <sub>H</sub>	FFFFAC <sub>H</sub>	FFFFAD <sub>H</sub>	FFFFAE <sub>H</sub>	

\*1:16 ビットデッドタイム 0, 1, 2 アンダフローの割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が 16 ビットリロードタイム 0 アンダフローに割当てられています。

\*2:DTTI の割込み制御レジスタと同じ値 (レジスタ番号とアドレス) が多機能タイマの DTP, 外部割込みチャネル 0, 1 検出, 16 ビットアウトプットコンペア 3 に割当てられています。

## ■ 多機能タイマの EI<sup>2</sup>OS 機能

多機能タイマは EI<sup>2</sup>OS と関係する回路を持っているので, 割込みが発生すると EI<sup>2</sup>OS を起動できます。

ただし, EI<sup>2</sup>OS は, 割込み制御レジスタ (ICR) を共用するほかのリソースが割込みを使用しない場合に限り使用可能です。たとえば, 16 ビットフリーランタイムのコンペアクリアが EI<sup>2</sup>OS を使用している場合, 16 ビットインプットキャプチャチャネル 0, 1 割込みを禁止しなければなりません。

## 14.6 多機能タイマの動作

---

多機能タイマの動作について説明します。

---

### ■ 多機能タイマの動作

#### ● 16 ビットフリーランタイム

16 ビットフリーランタイムは、リセット解除後、タイマデータレジスタ (TCDTH, TCDTL) に設定されている値からカウントアップを開始します。カウンタ値は、16 ビットアウトブットコンペアと16ビットインプットキャプチャの基準時間として使用されます。

#### ● 16 ビットアウトブットコンペア

16 ビットアウトブットコンペアは、"指定されたアウトブットコンペアレジスタに設定されている値" と "16 ビットフリーランタイム値" の比較に使用します。一致が検出された場合は、割込みフラグが設定され、出力レベルは反転します。

#### ● 16 ビットインプットキャプチャ

16 ビットインプットキャプチャは、指定された有効なエッジを検出するために使用します。有効なエッジが検出されると、割込みフラグが設定され、16 ビットフリーランタイム値が取出され、インプットキャプチャデータレジスタへ格納されます。

#### ● 波形ジェネレータ

波形ジェネレータは、リアルタイム出力 (RTO0 ~ RTO5)、16 ビット PPG タイマ 0、16 ビットデッドタイマを使用してさまざまな波形 (デッドタイムを含む) を生成します。

### 14.6.1 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、リセット完了後、タイマデータレジスタ (TCDTH, TCDTL) に設定されている値からカウントアップを開始します。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基準時間として使用されます。

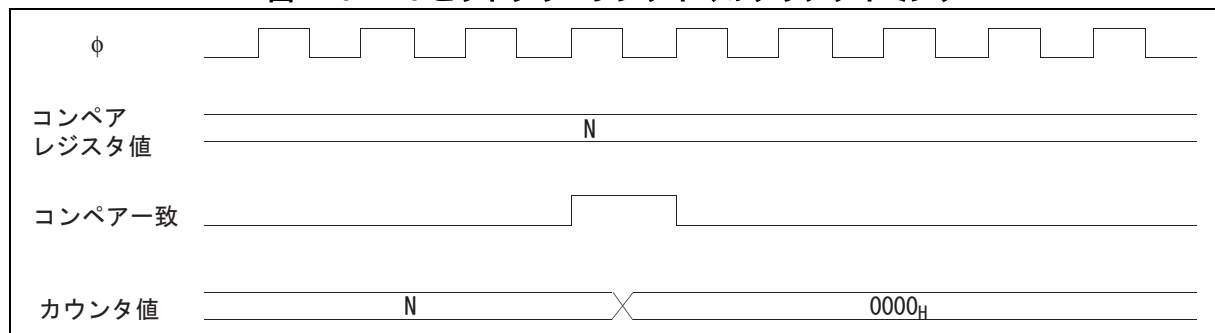
#### ■ タイマクリア

16 ビットフリーランタイマのカウンタ値は、下記のいずれかの場合にクリアされます。

- アップカウントモード (TCCSL レジスタの MODE:bit4=0) によってコンペアクリアレジスタとの一致が検出された場合
- 動作中に TCCSL レジスタの SCLR:bit3 に "1" が書き込まれた場合
- 停止中に TCDTH, TCDTL レジスタに "0000<sub>H</sub>" が書き込まれた場合
- リセットされた場合

リセットされると、カウンタは直ちにクリアされます。ソフトウェアクリアされた場合またはコンペアクリアレジスタとの一致が発生した場合は、カウンタはカウントタイミングと同期してクリアされます。

図 14.6-1 16 ビットフリーランタイマのクリアタイミング



#### ■ タイマモード

16 ビットフリーランタイマでは、以下のどちらかのモードを選択できます。

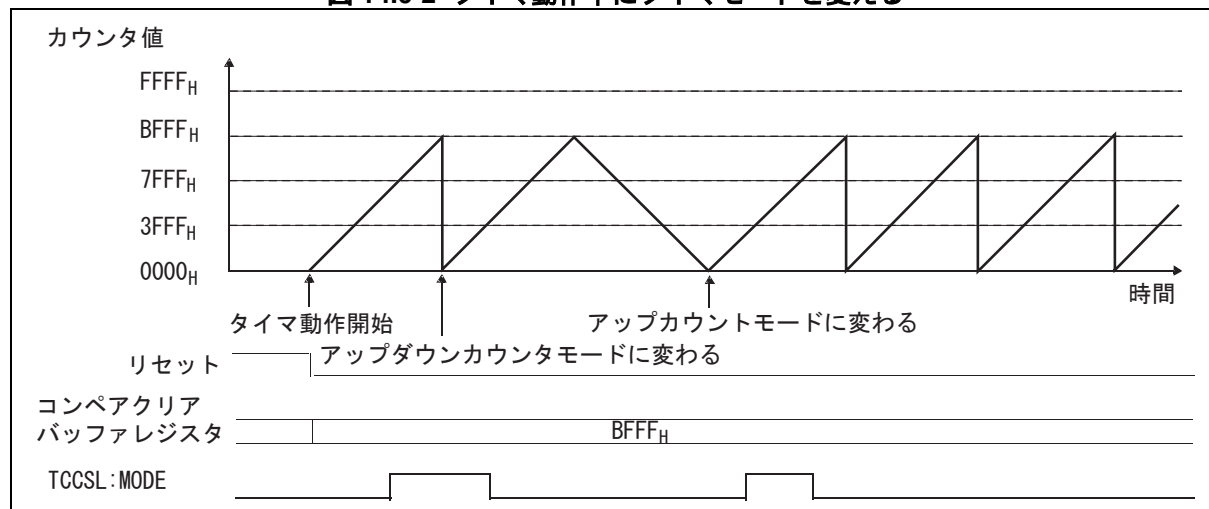
- アップカウントモード (TCCSL レジスタの MODE:bit4=0)
- アップダウンカウントモード (TCCSL レジスタの MODE:bit4=1)

アップカウントモード時は、カウンタは事前に設定されているタイマデータレジスタ (TCDTH, TCDTL) からカウントを開始し、カウンタ値がコンペアクリアレジスタ (CPCLR<sub>H</sub>, CPCLR<sub>L</sub>) の値と一致するまでカウントアップし、次にカウンタは "0000<sub>H</sub>" にクリアされ、次に再びカウントアップします。

アップダウンカウントモード時は、カウンタは事前に設定されているタイマデータレジスタ (TCDTH, TCDTL) からカウントを開始し、カウンタ値がコンペアクリアレジスタ (CPCLR<sub>H</sub>, CPCLR<sub>L</sub>) の値と一致するまでカウントアップし、次にカウンタがアップカウントからダウンカウントに変わり、カウンタ値が "0000<sub>H</sub>" に達するまでカウントダウンし、次に再びカウントアップします。

モードビット (TCCSL レジスタの MODE:bit4) には, タイマが動作中であろうと停止していようといつでも値を書き込むことができます。タイマ動作中にこのビットに書き込まれた値はバッファに入れられ, カウントモードはタイマ値が "0000<sub>H</sub>" になると変わります。

図 14.6-2 タイマ動作中にタイマモードを変える



## ■ コンペアクリアバッファ

コンペアクリアレジスタ (CPCLR<sub>H</sub>, CPCLR<sub>L</sub>) には, 有効または無効にできるバッファ機能が存在します。バッファ機能が有効 (TCCSL レジスタの BFE:bit6=1) の場合は, コンペアクリアバッファレジスタ (CPCLR<sub>BH</sub>, CPCLR<sub>BL</sub>) に書き込まれたデータは, 16 ビットフリーランタイム値 "0" が検出されると CPCLR<sub>H</sub>, CPCLR<sub>L</sub> レジスタに転送されます。バッファ機能が無効 (TCCSL ビットの BFE:bit6=0) の場合は, CPCLR<sub>BH</sub>, CPCLR<sub>BL</sub> レジスタは透過であり, データは CPCLR<sub>H</sub>, CPCLR<sub>L</sub> レジスタに直接書き込むことができます。

図 14.6-3 コンペアクリアバッファが無効 (TCCSL レジスタの BFE:bit6=0) 時のアップカウントモードによる動作

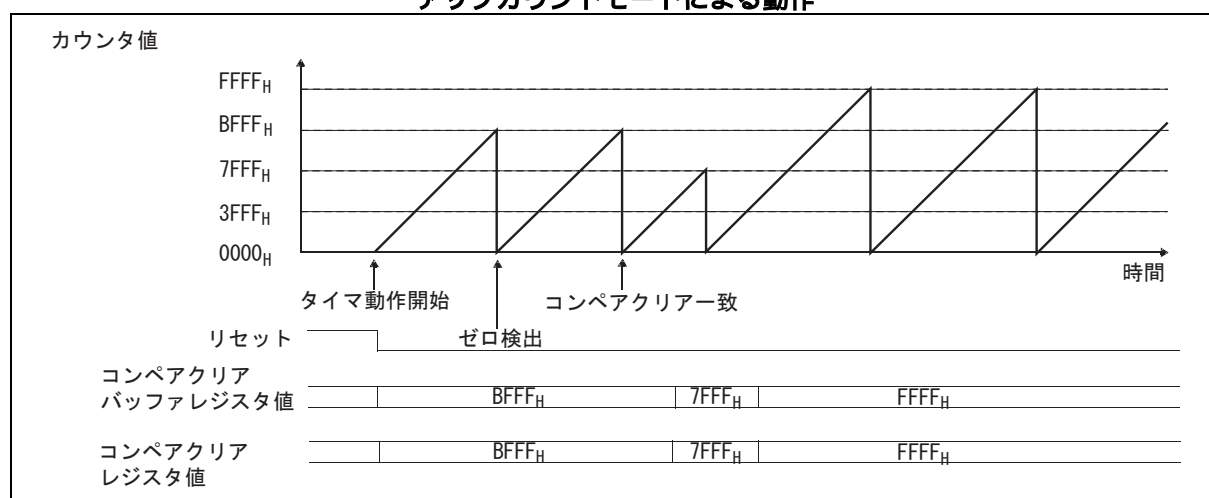


図 14.6-4 コンペアクリアバッファが有効 (TCCSL レジスタの BFE:bit6=1) 時の  
アップカウントモードによる動作

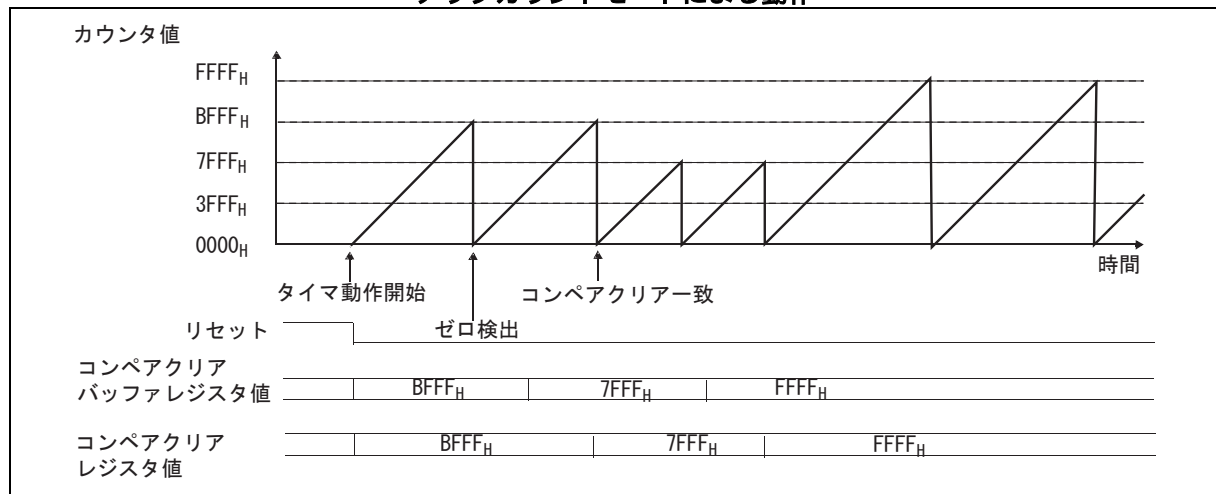
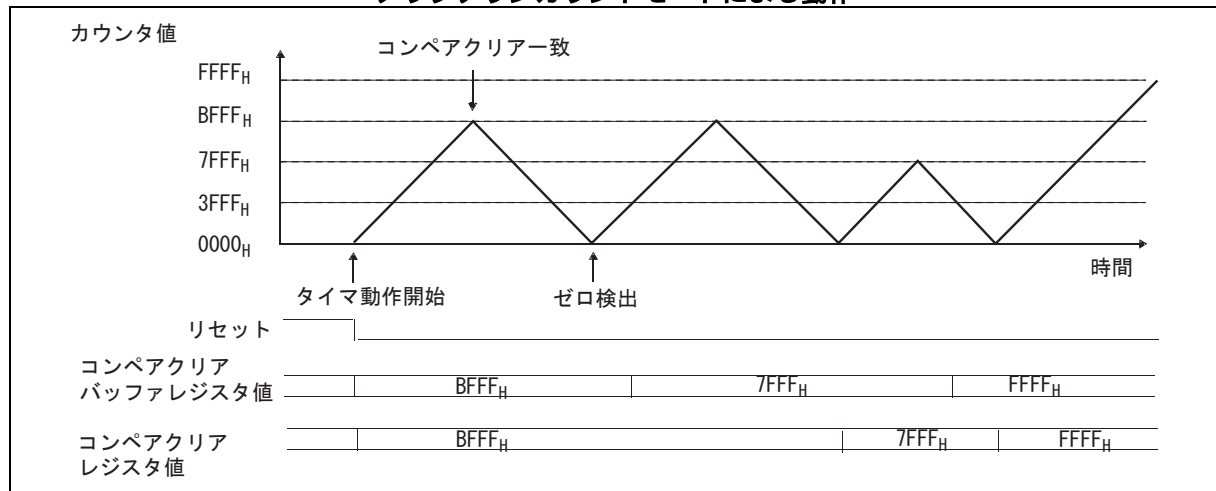


図 14.6-5 コンペアクリアバッファが有効 (TCCSL レジスタの BFE:bit6=1) 時の  
アップダウンカウントモードによる動作





## ■ タイマ割込み

16 ビットフリーランタイマでは、以下の 2 つの割込みを生成できます。

- コンペアクリア割込み
- ゼロ検出割込み

コンペアクリア割込みは、タイマ値がコンペアクリアレジスタ (CPCLR<sub>H</sub>, CPCLR<sub>L</sub>) の値と一致すると生成されます。

ゼロ検出割込みは、タイマ値が "0000<sub>H</sub>" に達すると生成されます。

---

< 注意事項 > ソフトウェアクリア (TCCSL レジスタの SCLR:bit3=1) は、ゼロ検出割込みを生成しません。

---

図 14.6-6 アップカウントモード (TCCSL レジスタの MODE:bit4=0) で生成された割込み

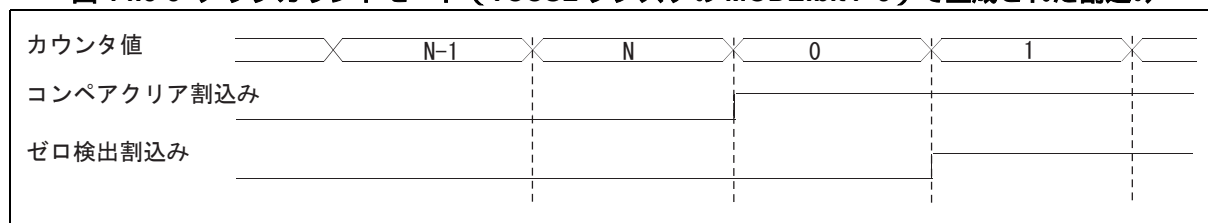
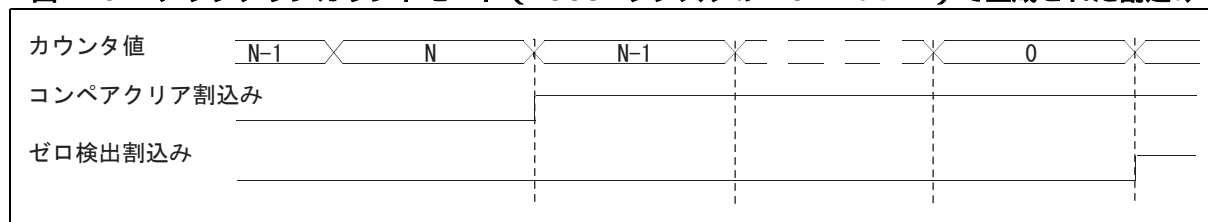


図 14.6-7 アップダウンカウントモード (TCCSL レジスタの MODE:bit4=1) で生成された割込み



## ■ 割込みマスク機能

TCCSH レジスタの MSI2 ~ MSI0:bit12 ~ bit10 を設定すると、割込み要求をマスクできます。MSI2 ~ MSI0 ビットは、カウンタ値が "000<sub>B</sub>" に達すると値をリロードする 3 ビットリロードダウンカウンタです。カウンタ値は、MSI2 ~ MSI0 ビットに直接書き込むことによってロードできます。マスクカウンタは、MSI2 ~ MSI0 ビットに設定された値です。MSI2 ~ MSI0 ビットが "000<sub>B</sub>" になると、割込み要因はマスクされません。

割込み要因は、カウントモード (TCCSL レジスタの MODE:bit4) によって異なります。アップカウントモード時は、コンペアクリア割込みのみをマスクでき、ゼロ検出割込みは "0" が検出されるたびに生成されます。アップダウンカウントモード時は、ゼロ検出割込みのみをマスクでき、コンペアクリア割込みはコンペアクリアが検出されるたびに生成されます。

---

< 注意事項 > ソフトウェアクリア (TCCSL レジスタの SCLR:bit3=1) は、ゼロ検出割込みを生成しません。

---

図 14.6-8 アップカウントモードでマスクされるコンペアクリア割込み

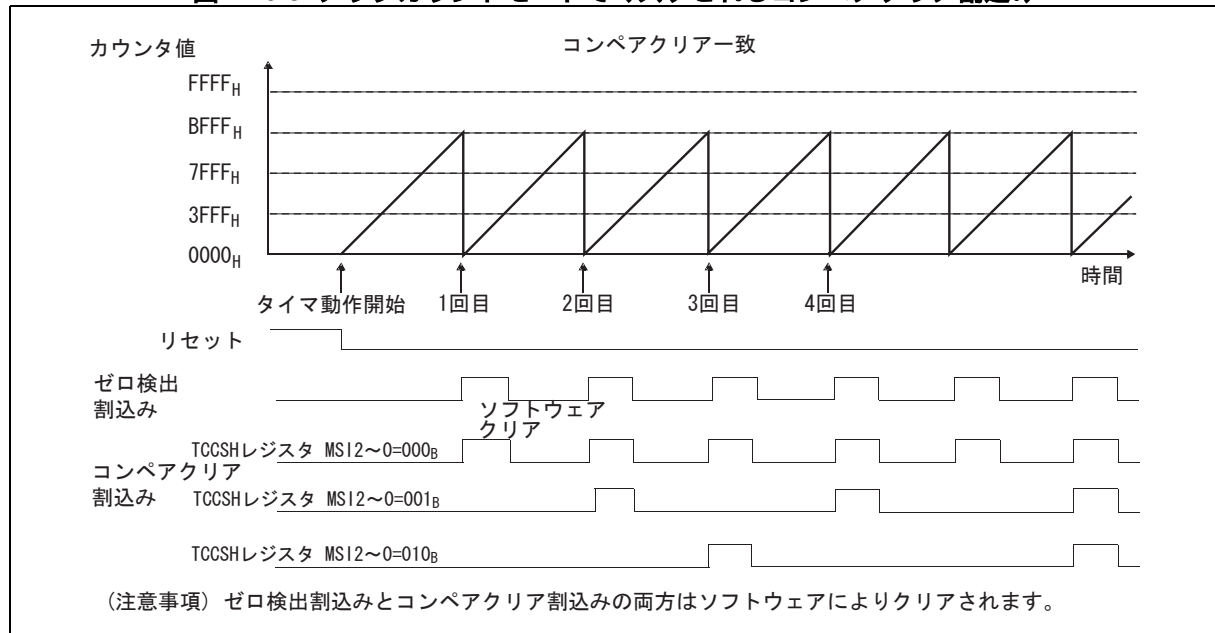
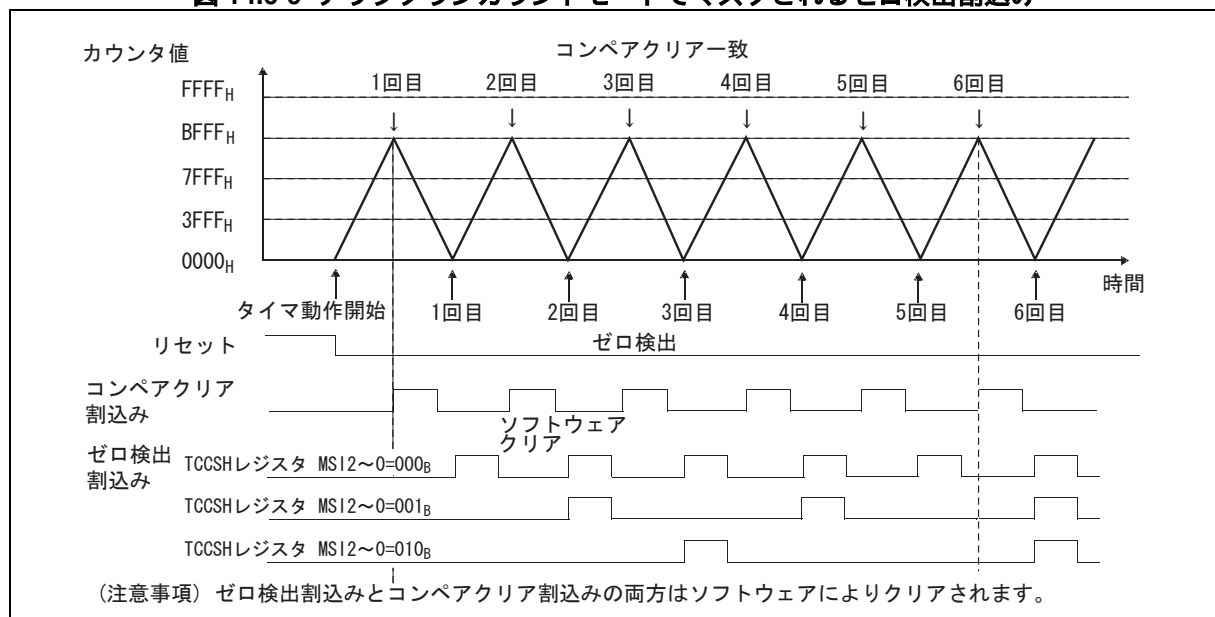


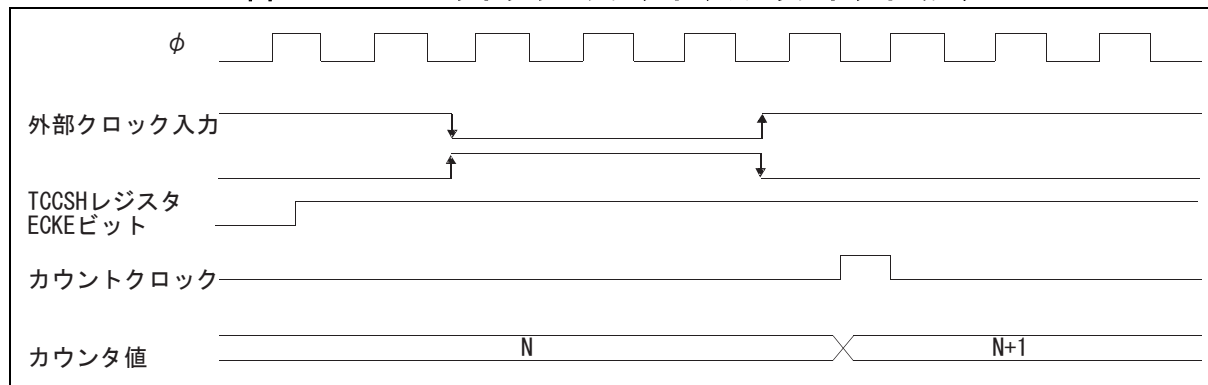
図 14.6-9 アップダウンカウントモードでマスクされるゼロ検出割込み



## ■ 選択された外部カウントクロック

16 ビットフリーランタイムは、入力クロック（内部クロックまたは外部クロック）に基づいてインクリメントされます。外部クロックが選択されると、外部クロックモード（TCCSH レジスタの ECKE:bit15=1）が選択された後、16 ビットフリーランタイムは外部入力の初期値が "1" のとき立上りエッジでカウントアップするかまたは外部入力の初期値が "0" のとき立下りエッジでカウントアップします。

図 14.6-10 16 ビットフリーランタイムのカウントタイミング



## 14.6.2 16 ビットアウトプットコンペアの動作

アウトプットコンペアは、" 指定されたコンペアレジスタに設定されている値 " と "16 ビットフリーランタイムの値 " の比較に使用します。一致が検出された場合は、割込みフラグが設定され、出力レベルが反転します。

### ■ 16 ビットアウトプットコンペアの動作

- コンペア動作は、各チャネル ( コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の CMOD:bit12=0) において実行できます。

図 14.6-11 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を別々に使用した際の出力波形例 (フリーランタイムはアップカウントモード)

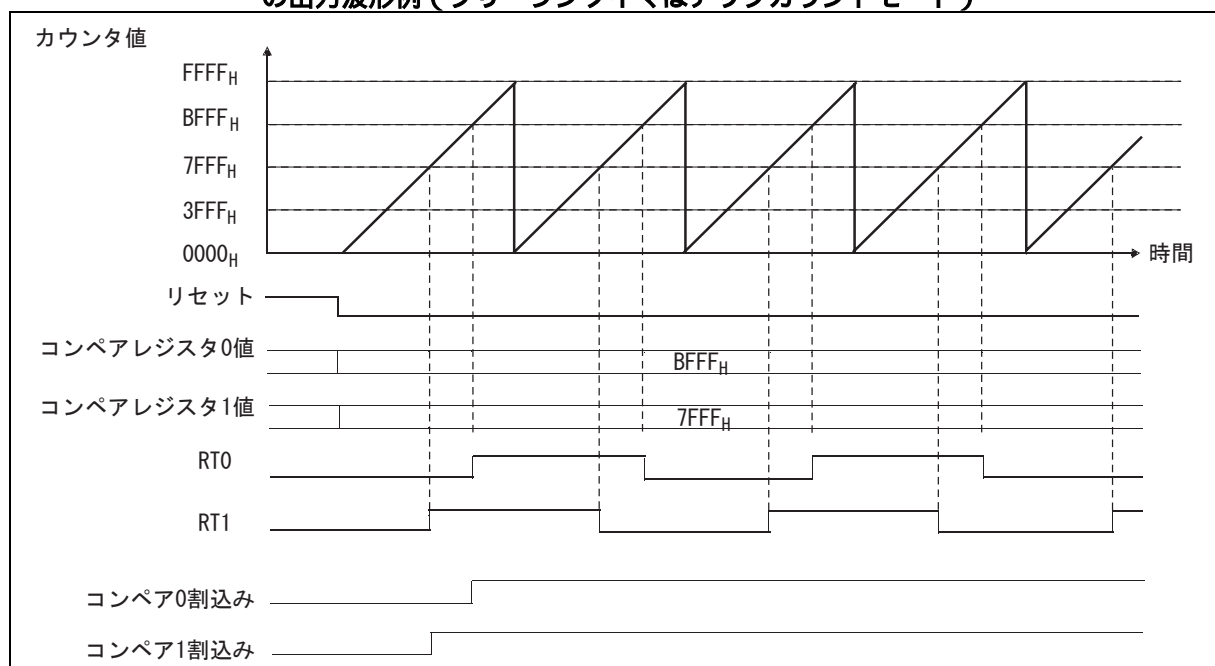
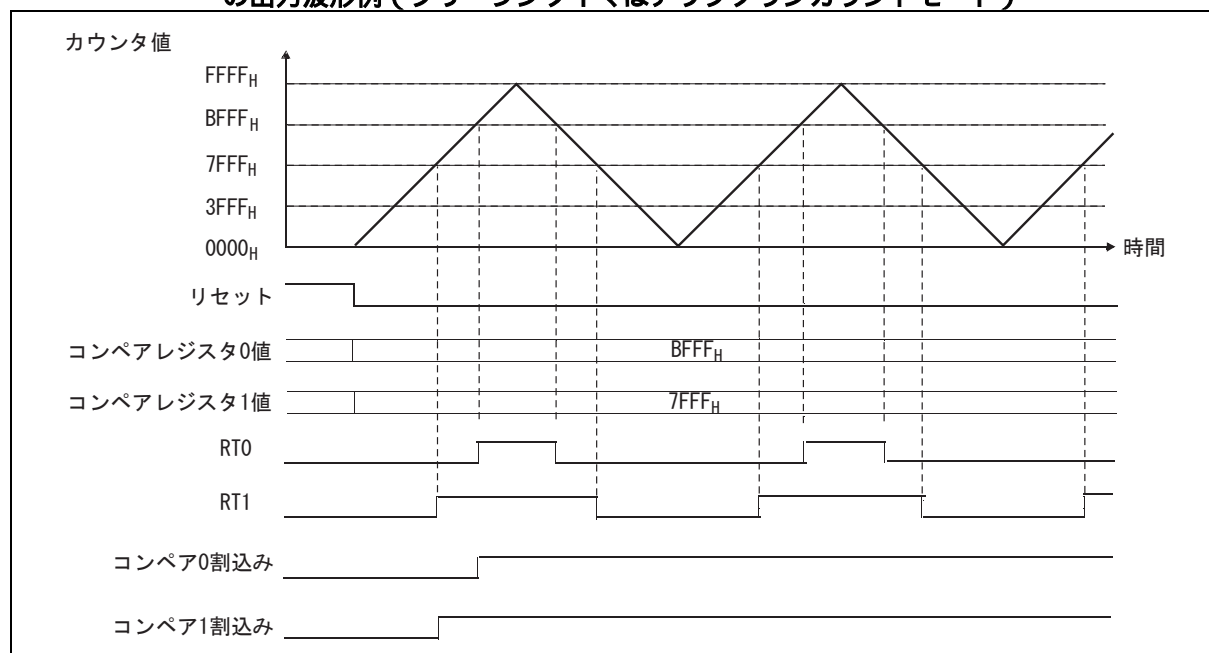


図 14.6-12 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を別々に使用した際の出力波形例（フリーランタイムはアップダウンカウントモード）



- 出力レベルは、一対のコンペアレジスタ（コンペア制御レジスタ上位（OCSH1, OCSH3, OCSH5）の CMOD:bit12=1）を使用して変更できます。

図 14.6-13 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 をペアで使用した際の出力波形例（フリーランタイムはアップカウントモード）

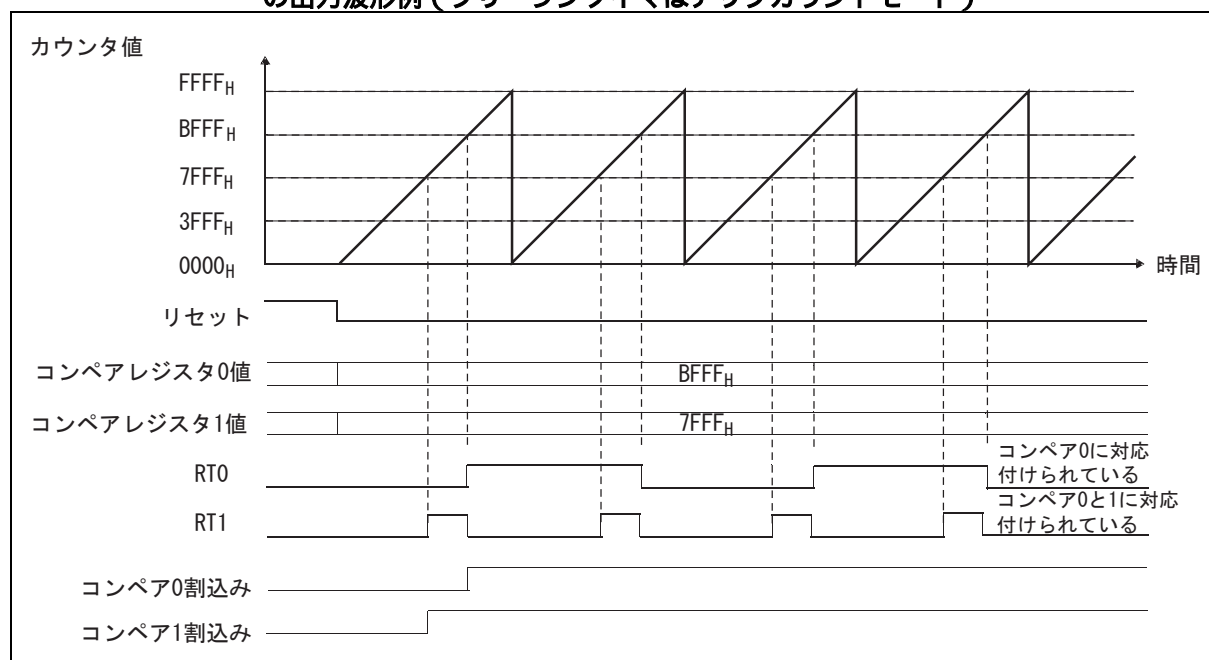
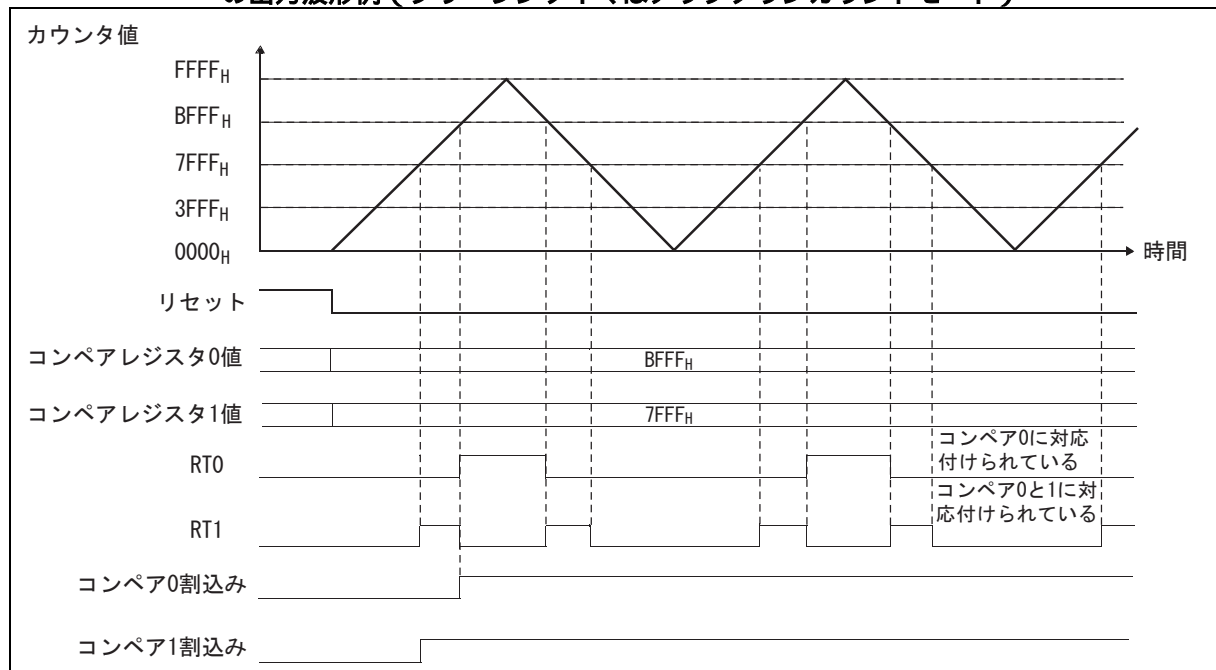
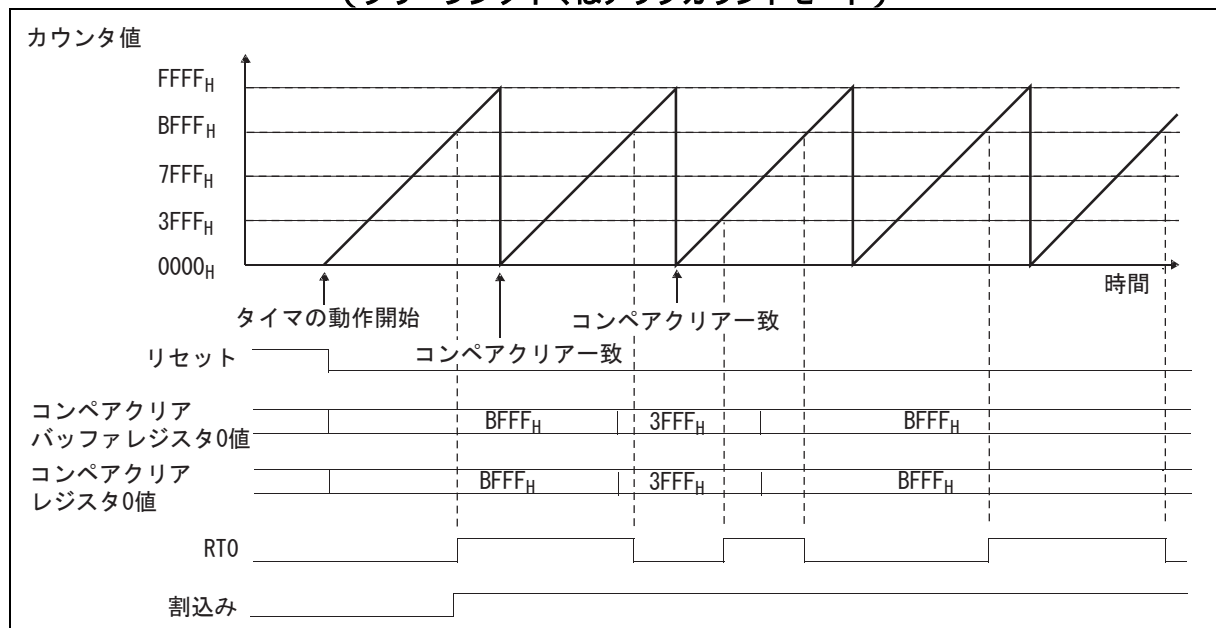


図 14.6-14 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を一緒に使用した際の出力波形例（フリーランタイムはアップダウンカウントモード）



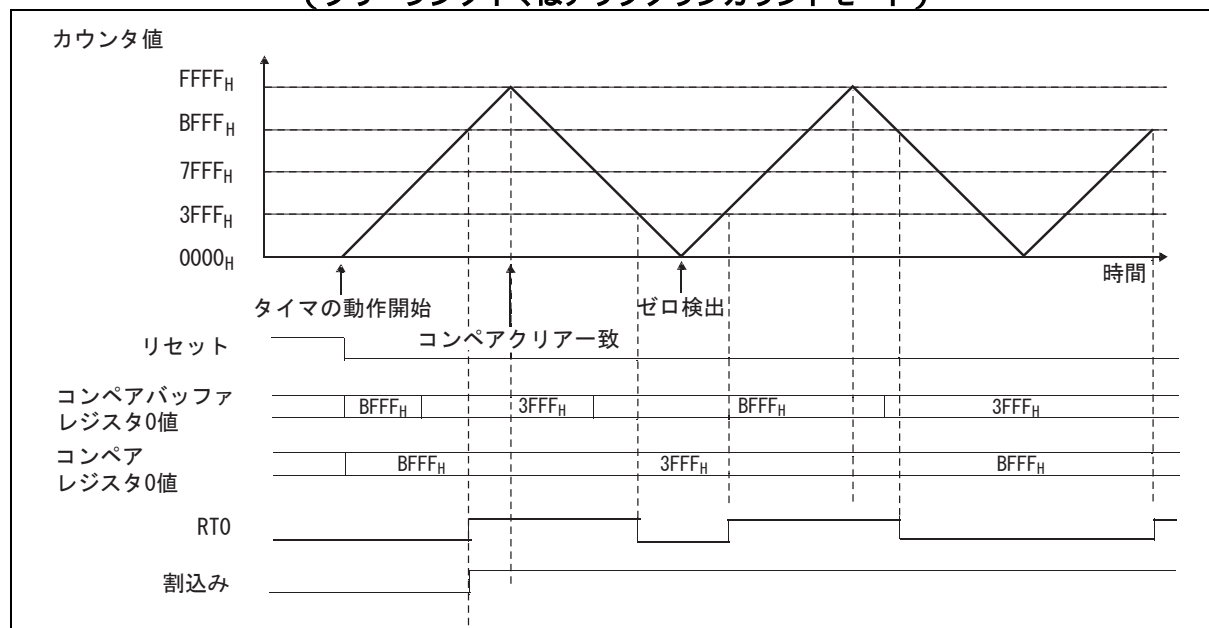
● コンペアバッファが無効になったときの出力レベル

図 14.6-15 コンペアバッファが無効になっているときの出力波形例（フリーランタイムはアップカウントモード）



- コンペアクリアー発生時にコンペアバッファが選択された際の出力レベル

図 14.6-16 コンペアバッファが有効になったときの出力波形例  
(フリーランタイムはアップダウンカウントモード)



## ■ 16 ビットアウトプットコンペアタイミング

フリーランタイマがコンペアレジスタ値と一致すると、アウトプットコンペアはコンペア一致信号を生成して出力を反転し、割込みを生成します。コンペア一致が発生すると、出力はカウンタのカウンタタイミングと同期して反転します。

< 注意事項 > コンペアレジスタが更新されると、カウンタ値とは比較されません。

図 14.6-17 コンペアレジスタが更新されたときのコンペア動作

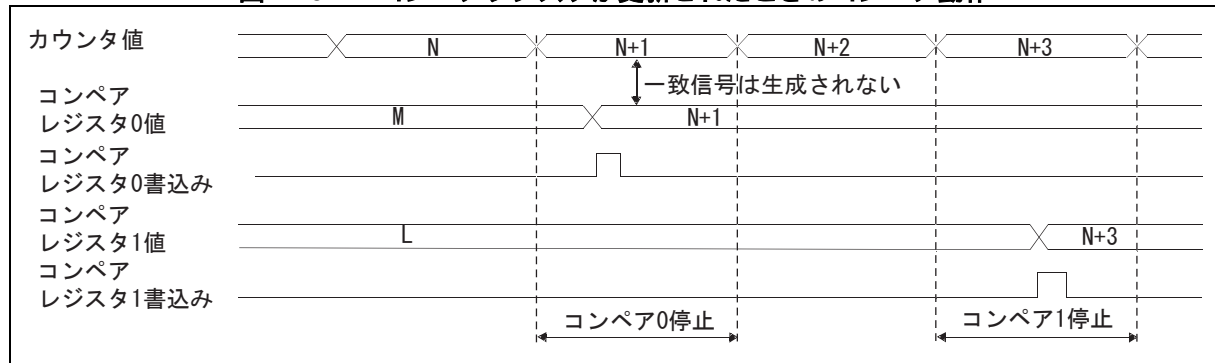


図 14.6-18 コンペア割込みタイミング

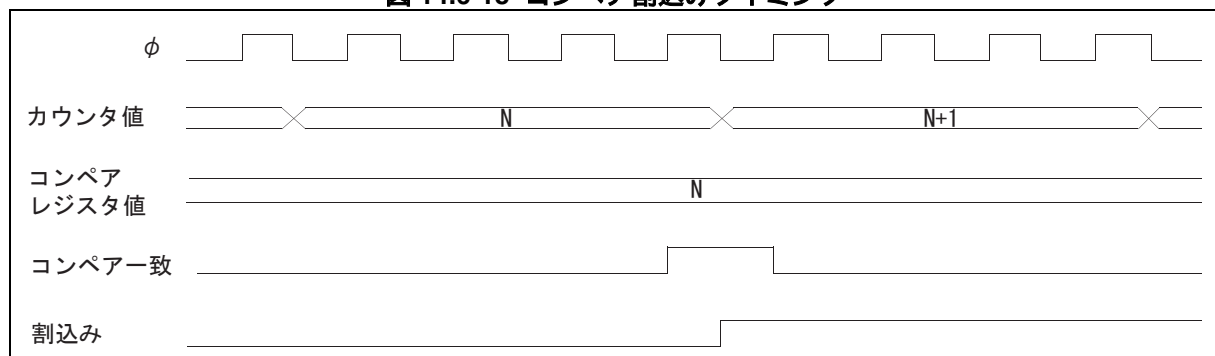
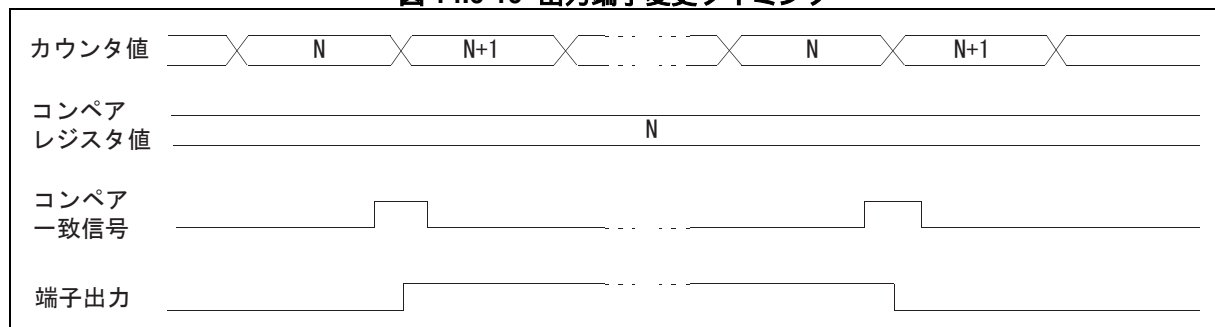


図 14.6-19 出力端子変更タイミング



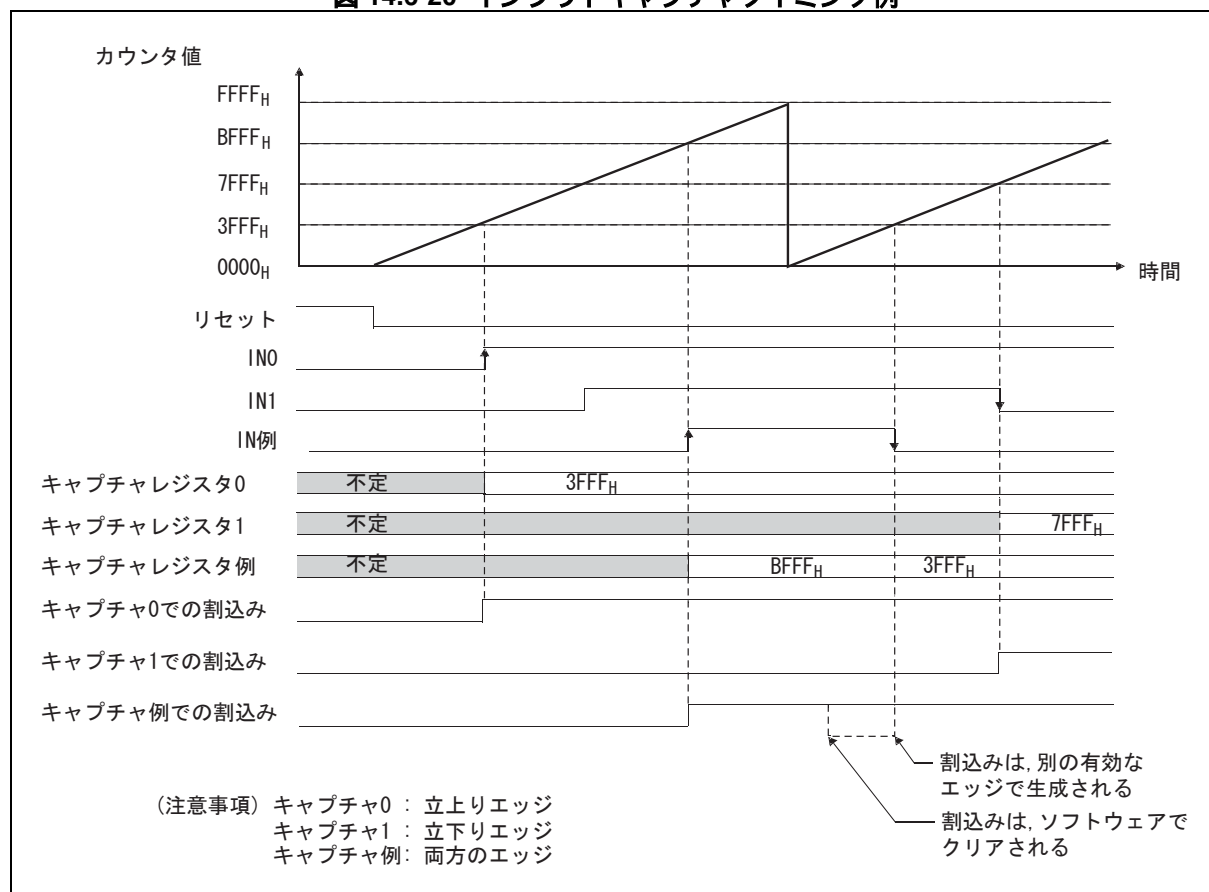


### 14.6.3 16 ビットインプットキャプチャの動作

インプットキャプチャは、指定された有効なエッジを検出するために使用します。有効なエッジが検出されると、割込みフラグが設定され、16 ビットフリーランタイムの値がキャプチャレジスタへロードされます。

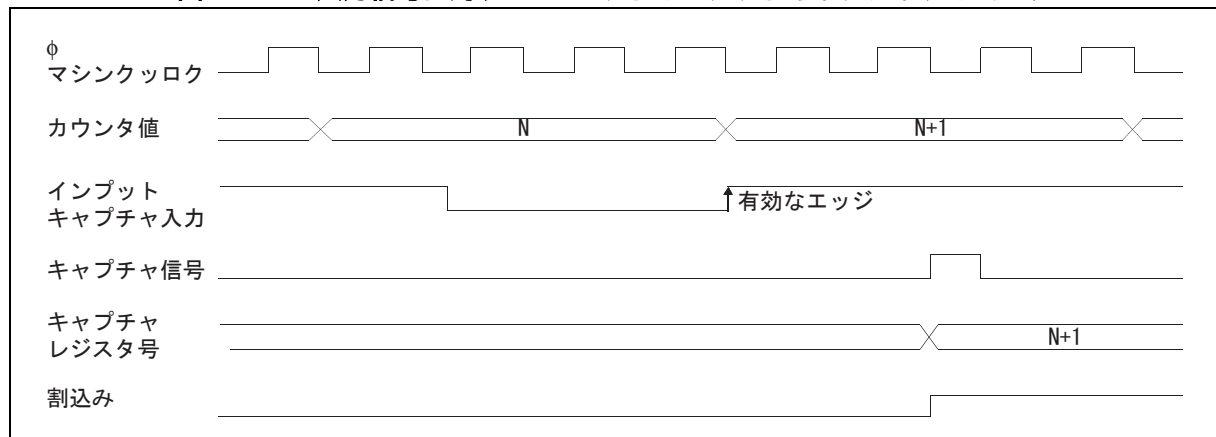
#### ■ 16 ビットインプットキャプチャの動作

図 14.6-20 インプットキャプチャタイミング例



## ■ 16 ビットインプットキャプチャ入力タイミング

図 14.6-21 入力信号に対する 16 ビットインプットキャプチャタイミング



## 14.6.4 波形ジェネレータの動作

波形ジェネレータは、リアルタイム出力（RTO0 ~ RTO5）、16 ビット PPG タイマ 0、16 ビットデッドタイマ 0、1、2 を使用してさまざまな波形（デッドタイムを含む）を生成できます。

### ■ RTO0 ~ RTO5 と GATE の出力状態

表 14.6-1 RTO0 ~ RTO5/GATE 出力状態とビット設定 (1 / 2)

TMD2	TMD1	TMD0	GTENx	PGENx	RTOx	GATE
0	0	0	X	X	リアルタイム出力 RTx (16 ビットアウト プットコンペア出力)	常に "0"
0	0	1	X	0	リアルタイム出力 RTx (16 ビットアウト プットコンペア出力)	(RTx & GTENx) *3
0	0	1	0	1	RTx が "H" の期間に PPG0 のパルスを出力 *1	常に "0"
0	0	1	1	1	RTx が "H" の期間に GATE 信号により起動さ れた PPG0 のパルスを出力	(RTO0RT 1IRT2IRT3I RT4IRT5)
0	1	0	X	0	RT0, 1 の立上りエッジにより 16 ビットデッ ドタイマ 0 を起動し、16 ビットデッドタイ マ 0 がアンダフローするまで "H" を出力	タイマ動 作期間中 は "H" を 出力 *4
					RT2, 3 の立上りエッジにより 16 ビットデッ ドタイマ 1 を起動し、16 ビットデッドタイ マ 1 がアンダフローするまで "H" を出力	
					RT4, 5 の立上りエッジにより 16 ビットデッ ドタイマ 2 を起動し、16 ビットデッドタイ マ 2 がアンダフローするまで "H" を出力	
0	1	0	0	1	RT0, 1 の立上りエッジにより 16 ビットデッ ドタイマ 0 を起動し、16 ビットデッドタイ マ 0 がアンダフローするまで PPG0 のパル スを出力 *1	常に "0"
					RT2, 3 の立上りエッジにより 16 ビットデッ ドタイマ 1 を起動し、16 ビットデッドタイ マ 1 がアンダフローするまで PPG0 のパル スを出力 *1	
					RT4, 5 の立上りエッジにより 16 ビットデッ ドタイマ 2 を起動し、16 ビットデッドタイ マ 2 がアンダフローするまで PPG0 のパル スを出力 *1	

表 14.6-1 RTO0 ~ RTO5/GATE 出力状態とビット設定 (2 / 2)

TMD2	TMD1	TMD0	GTENx	PGENx	RTOx	GATE
0	1	0	1	1	RT0, 1 の立上りエッジにより 16 ビットデッドタイム 0 を起動し, 16 ビットデッドタイム 0 がアンダフローするまで GATE 信号により起動された PPG0 のパルスを出力	タイマ動作期間中は "H" を出力 *4
					RT2, 3 の立上りエッジにより 16 ビットデッドタイム 1 を起動し, 16 ビットデッドタイム 1 がアンダフローするまで GATE 信号により起動された PPG0 のパルスを出力	
					RT4, 5 の立上りエッジにより 16 ビットデッドタイム 2 を起動し, 16 ビットデッドタイム 2 がアンダフローするまで GATE 信号により起動された PPG0 のパルスを出力	
1	0	0	X	X	RT1 でノンオーバーラップ信号を生成 *2	常に "0"
					RT3 でノンオーバーラップ信号を生成 *2	
					RT5 でノンオーバーラップ信号を生成 *2	
1	1	1	0	X	PPG0 でノンオーバーラップ信号を生成	常に "0"
1	1	1	1	X	GATE 信号により起動された PPG0 でノンオーバーラップ信号を生成	( RT0 RT1 RT2 RT3 RT4 RT5 )
その他					常に "0"	常に "0"

\*1: あらかじめ PPG0 を起動しておく必要があります。

\*2: ノンオーバーラップ信号を生成するには, 必ず RT1, RT3, RT5 に対して 2 チャネルモード (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の CMOD:bit12=1) を選択してください。

\*3: GTENx ビットに "1" を設定した RTx から GATE 信号が生成されます。

\*4: GTENx ビットに "1" を設定した RTx によって起動されるタイマの動作期間中に, GATE 信号が生成されます。複数の GATEx ビットに "1" を設定した場合, GATE 信号は各々のタイマ動作期間中の信号を OR した信号となります。

(注意事項)

RTO0, RTO1 は 16 ビットデッドタイム制御レジスタ下位 (DTCR0) の TMD2 ~ TMD0:bit2 ~ bit0 により, RTO2, RTO3 は (DTCR1) 上位レジスタの TMD5 ~ TMD3:bit10 ~ bit8 により, RTO4, RTO5 は下位レジスタ (DTCR2) の TMD8 ~ TMD6:bit2 ~ bit0 により制御されます。

## ■ PPG0 出力制御

RTO0 ~ RTO5 端子への PPG0 出力は、PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN5 ~ PGEN0:bit15 ~ bit10 で許可にできます。

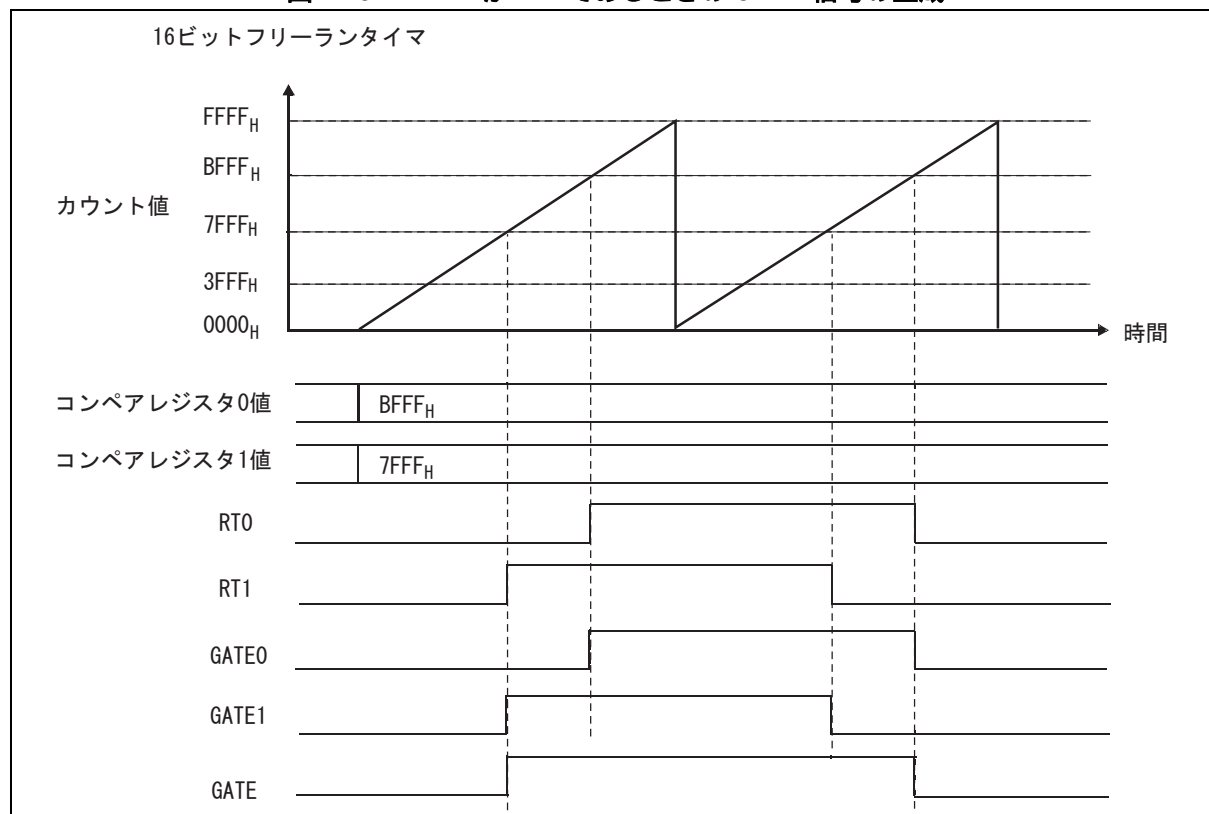
## ■ ゲートトリガされた PPG0 出力

波形ジェネレータではリアルタイム出力 RTO0 ~ RTO5 により GATE 信号を生成でき、16 ビットデッドタイム 0, 1, 2 では PPG0 カウントをトリガとして動作できます。1 つの 16 ビットデッドタイム 0, 1, 2 で 2 つのリアルタイム出力 (RTO0/2/4, RTO1/3/5) が操作され、6 つの別々のゲート信号が生成されます。これら 6 つのゲート信号は論理和がとられて GATE 信号を生成し、PPG0 カウントのトリガとなります。

また、PGEN0 ~ PGEN5 信号を使用すると、PPG0 のみを使用することで RTO0 ~ RTO5 端子に 6 つの異なる波形を出力できます。

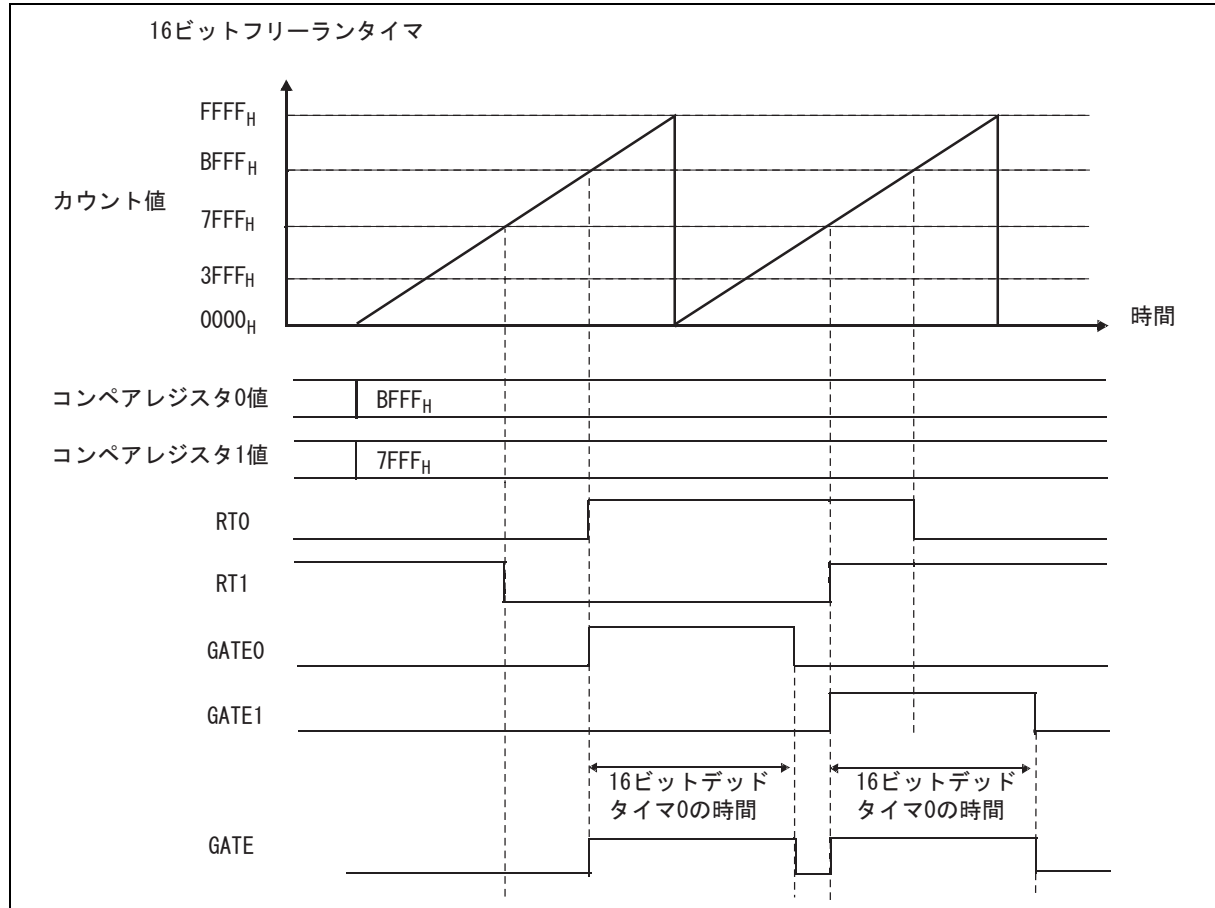
- GTENx がアクティブであり各 RTx が "H" であるとき (16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "001<sub>B</sub>" または "111<sub>B</sub>") の GATE 信号生成

図 14.6-22 RTx が "H" であるときの GATE 信号の生成



- GTENx がアクティブ( DTCR0 ~ DTCR2 レジスタの TMD8 ~ TMD0=010<sub>B</sub>)であるときの RTx 立上りエッジから 16 ビットデッドタイム 0, 1, 2 アンダフローまでにおける GATE 信号の生成

図 14.6-23 RTx 立上りエッジから 16 ビットデッドタイムアンダフローまでにおける GATE 信号の生成



< 注意事項 > 各 16 ビットデッドタイムは、2 つの RT に対して使用されます。すなわち、16 ビットデッドタイム 0 は RT0 と RT1 に対して使用され、16 ビットデッドタイム 1 は RT2 と RT3 に対して使用され、16 ビットデッドタイム 2 は RT4 と RT5 に対して使用されます。したがって、RT を使用して、すでに動作中のタイマの起動を試みてはなりません。このような試みを行った場合は、GATE 信号出力が拡張され、その結果、誤動作が発生する場合があります。

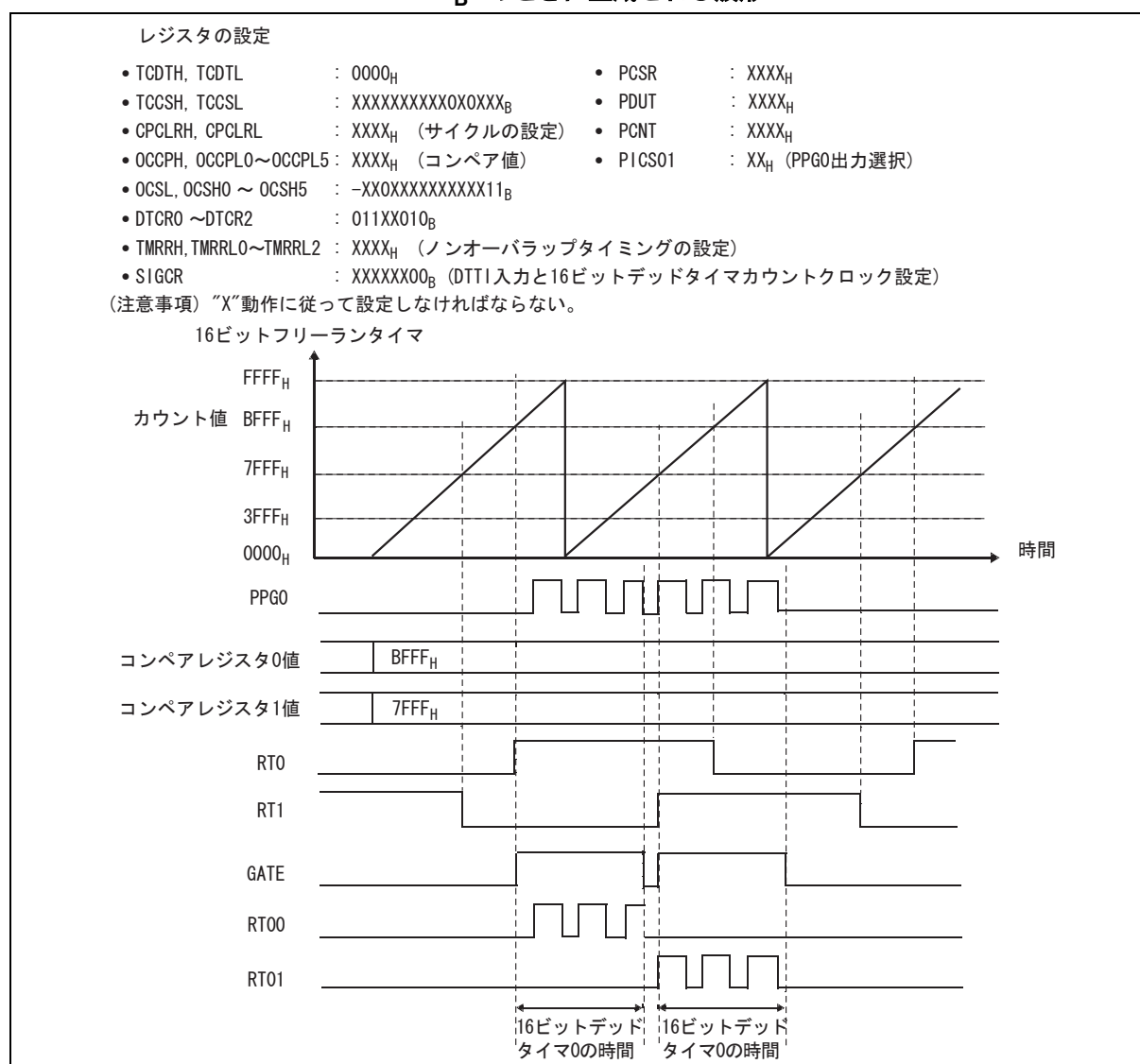
### 14.6.4.1 タイマモードの動作

RT0 ~ RT5 端子の立上がりエッジが検出されると, 16 ビットデッドタイマに値がリロードされて 16 ビットデッドタイマがダウンカウントを開始します。PPG タイマ 0 は 16 ビットデッドタイマでアンダフローが発生するまで RT00 ~ RT05 端子へ出力し続けます。

#### ■ タイマモードの動作

- RT立上がりエッジから 16ビットデッドタイマアンダフローまでにおけるPPG0出力パルス生成 (DTCR0 ~ DTCR2 レジスタの TMD8 ~ TMD0(上位は bit10 ~ bit8, 下位は bit2 ~ bit0)=010<sub>B</sub>)

図 14.6-24 TMD2 ~ TMD0(上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "010<sub>B</sub>" のときに生成される波形



---

< 注意事項 > 各 16 ビットデッドタイムは、2 つの RT に対して使用されます。すなわち、16 ビットデッドタイム 0 は RT0 と RT1 に対して使用され、16 ビットデッドタイム 1 は RT2 と RT3 に対して使用され、16 ビットデッドタイム 2 は RT4 と RT5 に対して使用されます。したがって、RT を使用して、すでに動作中の PPG0 の起動を試みてはなりません。このような試みを行った場合は、GATE 信号出力が拡張され、その結果、誤動作が発生する場合があります。

---



## 14.6.4.2 デッドタイムタイマモード時の動作

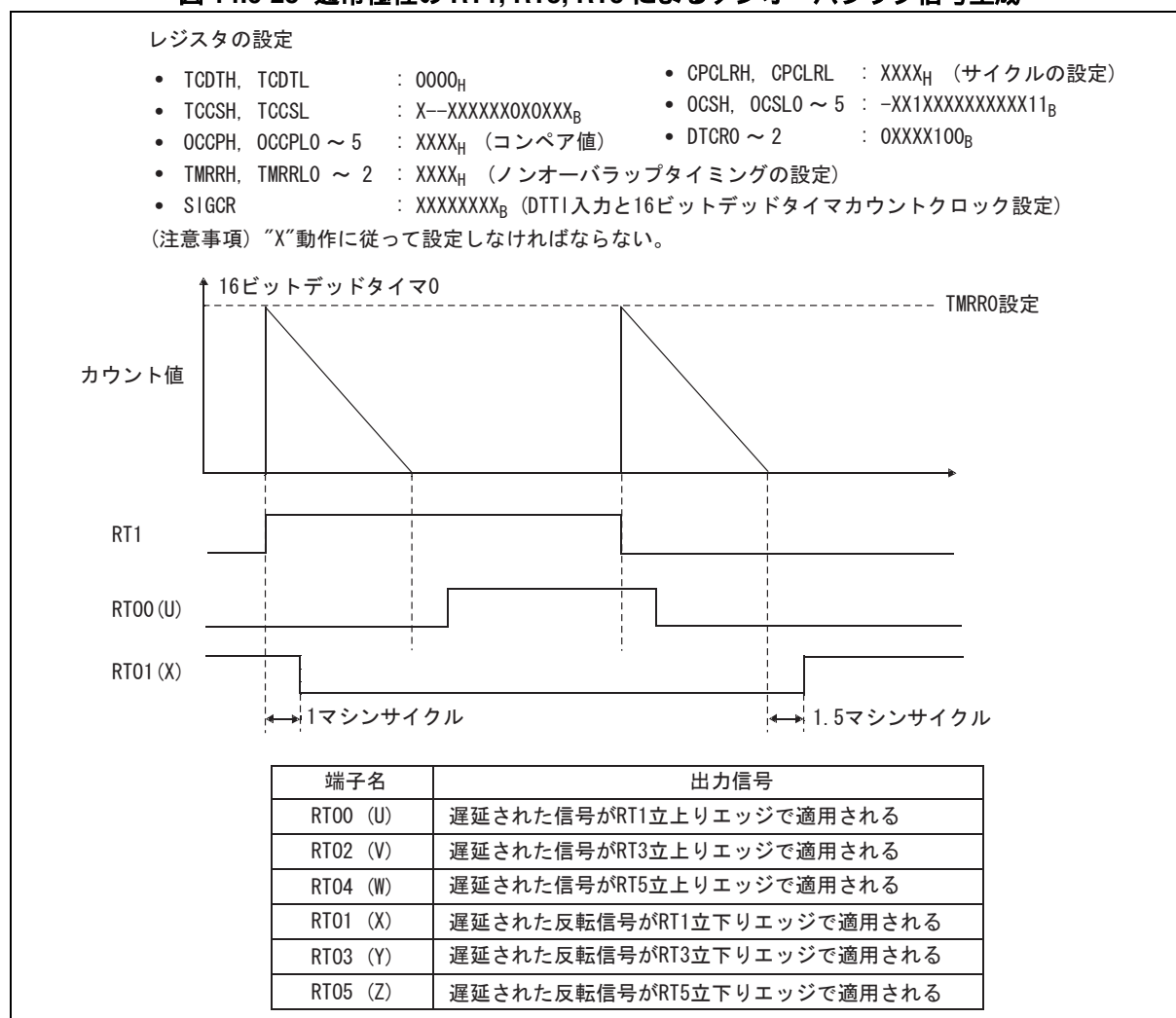
デッドタイムジェネレータは、リアルタイム出力 (RT1, RT3, RT5), もしくは PPG0 タイマパルス出力を入力し, 外部端子 (RT00 ~ RT05) へノンオーバーラップ信号 (反転信号) を出力します。

### ■ デッドタイムタイマモード時の動作

- 通常極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMD8 ~ TMD0 (上位は 10 ~ 8, 下位は 2 ~ 0) =100<sub>B</sub>)

DTCR0 ~ DTCR2 レジスタの DMOD が "0" (通常極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH, L0 ~ L2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, RT1, RT3, RT5 端子の立上りエッジまたは立下りエッジで適用されます。RT1, RT3, RT5 パルス幅が, 設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムはその次の RT エッジの TMRRH, L0 ~ L2 レジスタ値からダウンカウントを再開します。

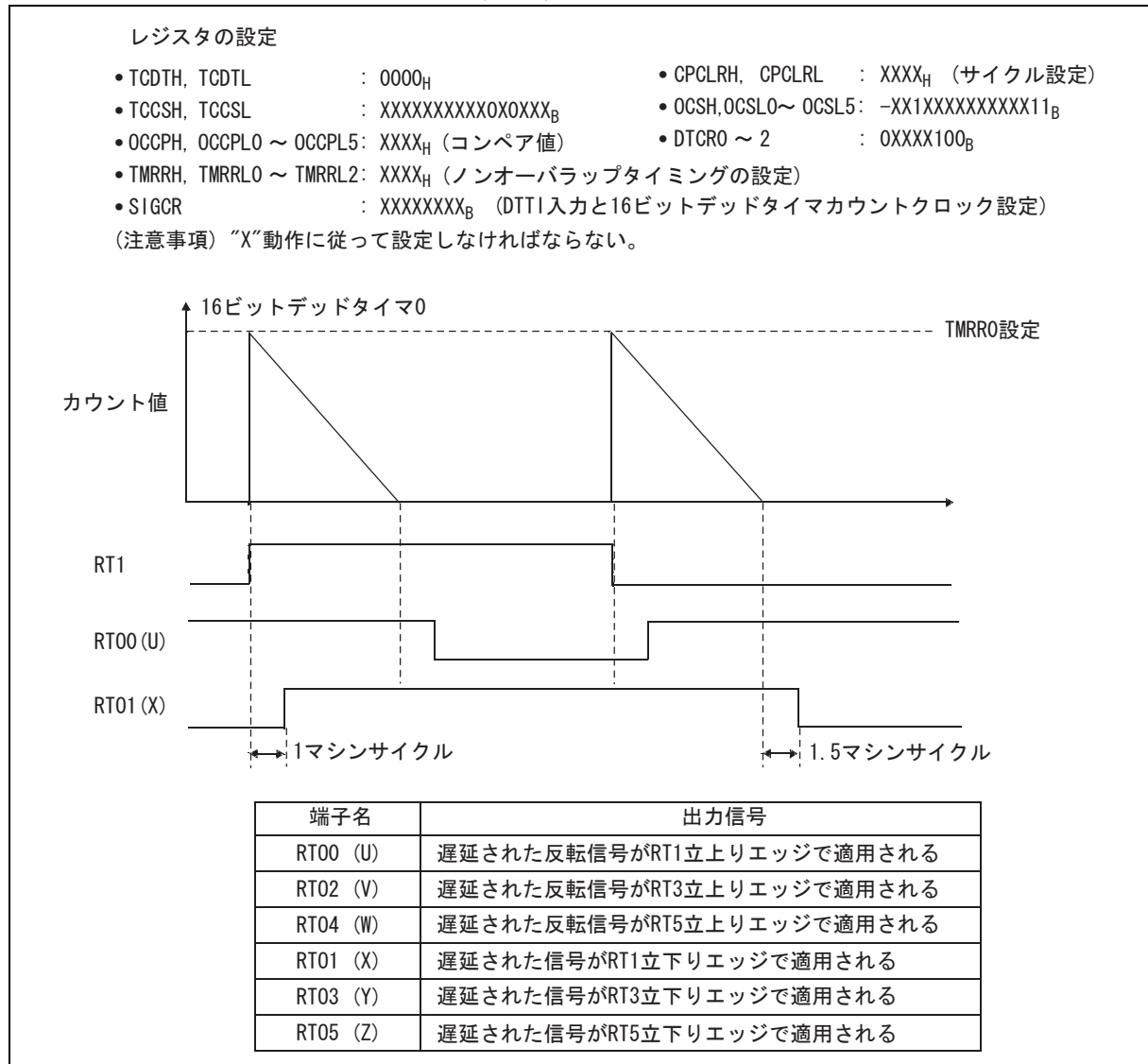
図 14.6-25 通常極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成



- 反転極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は 10 ~ 8, 下位は 2 ~ 0) = 100<sub>B</sub>)

DTCR0 ~ DTCR2 レジスタの DMOD (上位は bit15, 下位は bit7) が "1" (反転極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH, L0 ~ L2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, RT1, RT3, RT5 の立上りエッジまたは立下りエッジで適用されます。RT1, RT3, RT5 パルス幅が, 設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムはその次の RT エッジの TMRRH, L0 ~ L2 値からダウンカウントを再開します。

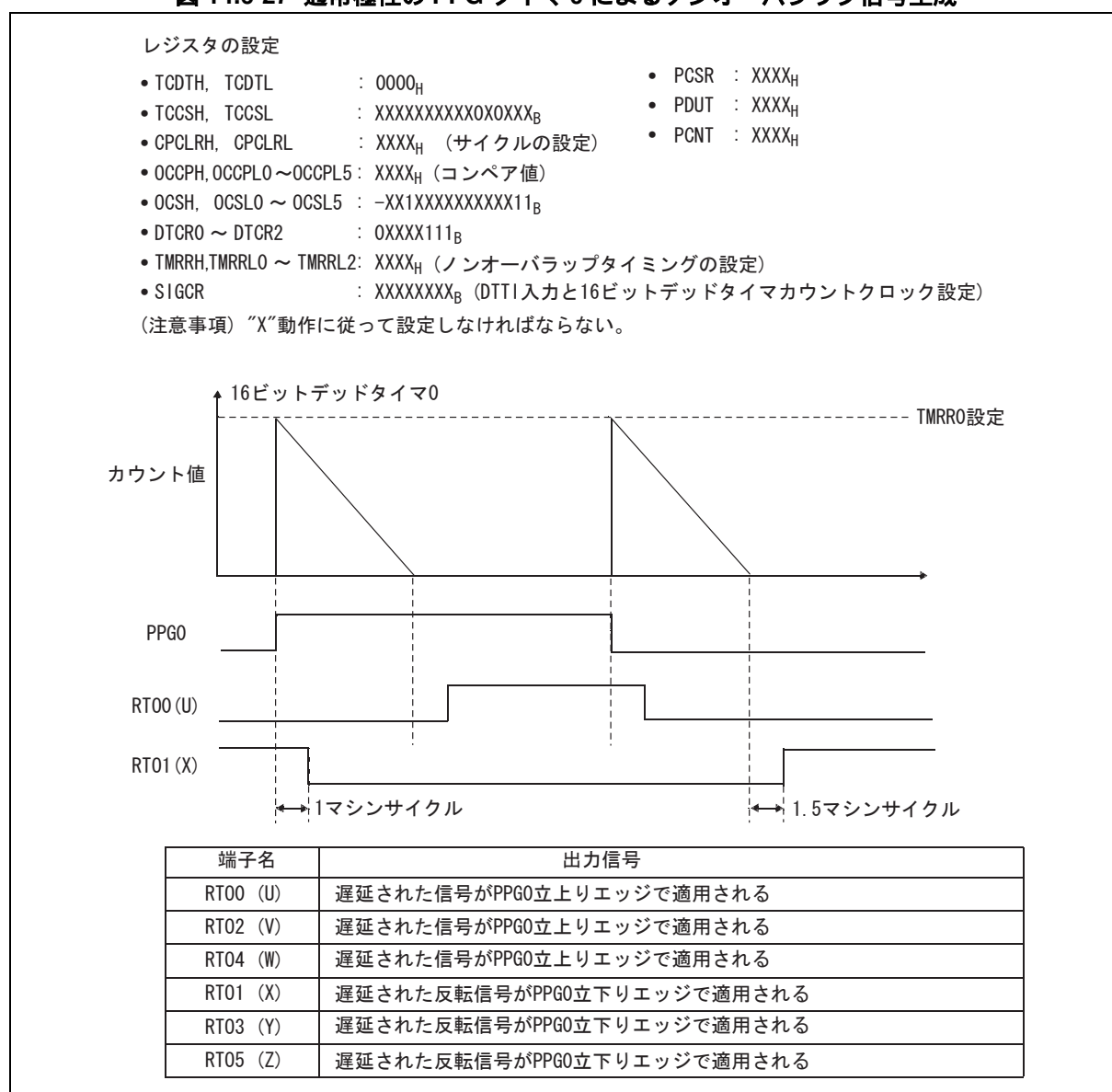
図 14.6-26 反転極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成



- 通常極性の PPG によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 111<sub>B</sub>)

DTCR0 ~ DTCR2 レジスタの DMOD2 ~ DMOD0 (上位は bit15, 下位は bit7) が "0" (通常極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, PPG0 タイマパルス信号または反転信号の立上りエッジで適用されます。PPG タイマパルス幅が, 設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムは PPG0 パルスのその次のエッジの TMRRH, L0 ~ L2 値からダウンカウントを再開します。

図 14.6-27 通常極性の PPG タイマ 0 によるノンオーバーラップ信号生成



● 反転極性の PPG によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMD8 ~ TMD0 (上位は 10 ~ 8, 下位は 2 ~ 0) =111<sub>B</sub>)

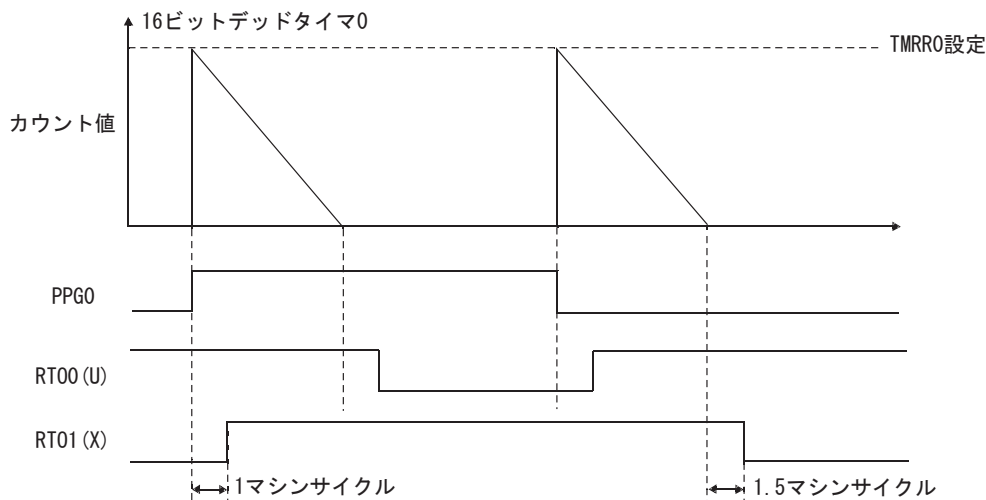
DTCR0 ~ DTCR2 ビットの DMOD2 ~ DMOD0 (上位は bit15, 下位は bit7) が "1" (反転極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, PPG0 タイマパルス信号または反転信号の立上りエッジで適用されます。PPG0 タイマパルス幅が, 設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムは PPG0 パルスのその次のエッジの TMRRH, TMRRL0 ~ TMRRL2 値からダウンカウントを再開します。

図 14.6-28 反転極性の PPG タイマ 0 によるノンオーバーラップ信号生成

レジスタの設定

- TCDTH, TCDTL : 0000<sub>H</sub>
- TCCSH, TCCSL : XXXXXXXXX0X0XX<sub>B</sub>
- CPCLRH, CPCLRL : XXX<sub>H</sub> (サイクルの設定)
- OCCPH, OCCPL0 ~ 5 : XXX<sub>H</sub> (コンペア値)
- OCSH, OCSL0 ~ 5 : -XX1XXXXXXXXXX11<sub>B</sub>
- DTCR0 ~ 2 : 1XXXX111<sub>B</sub>
- TMRRH, TMRRL0 ~ 2 : XXX<sub>H</sub> (ノンオーバーラップタイミングの設定)
- SIGCR : XXXXXXX<sub>B</sub> (DTTI入力と16ビットデッドタイムカウントクロック設定)
- PCSR : XXX<sub>H</sub>
- PDUT : XXX<sub>H</sub>
- PCNT : XXX<sub>H</sub>

(注意事項) "X"動作に従って設定しなければならない。



端子名	出力信号
RT00 (U)	遅延された反転信号がPPG0立上りエッジで適用される
RT02 (V)	遅延された反転信号がPPG0立上りエッジで適用される
RT04 (W)	遅延された反転信号がPPG0立上りエッジで適用される
RT01 (X)	遅延された信号がPPG0立下りエッジで適用される
RT03 (Y)	遅延された信号がPPG0立下りエッジで適用される
RT05 (Z)	遅延された信号がPPG0立下りエッジで適用される

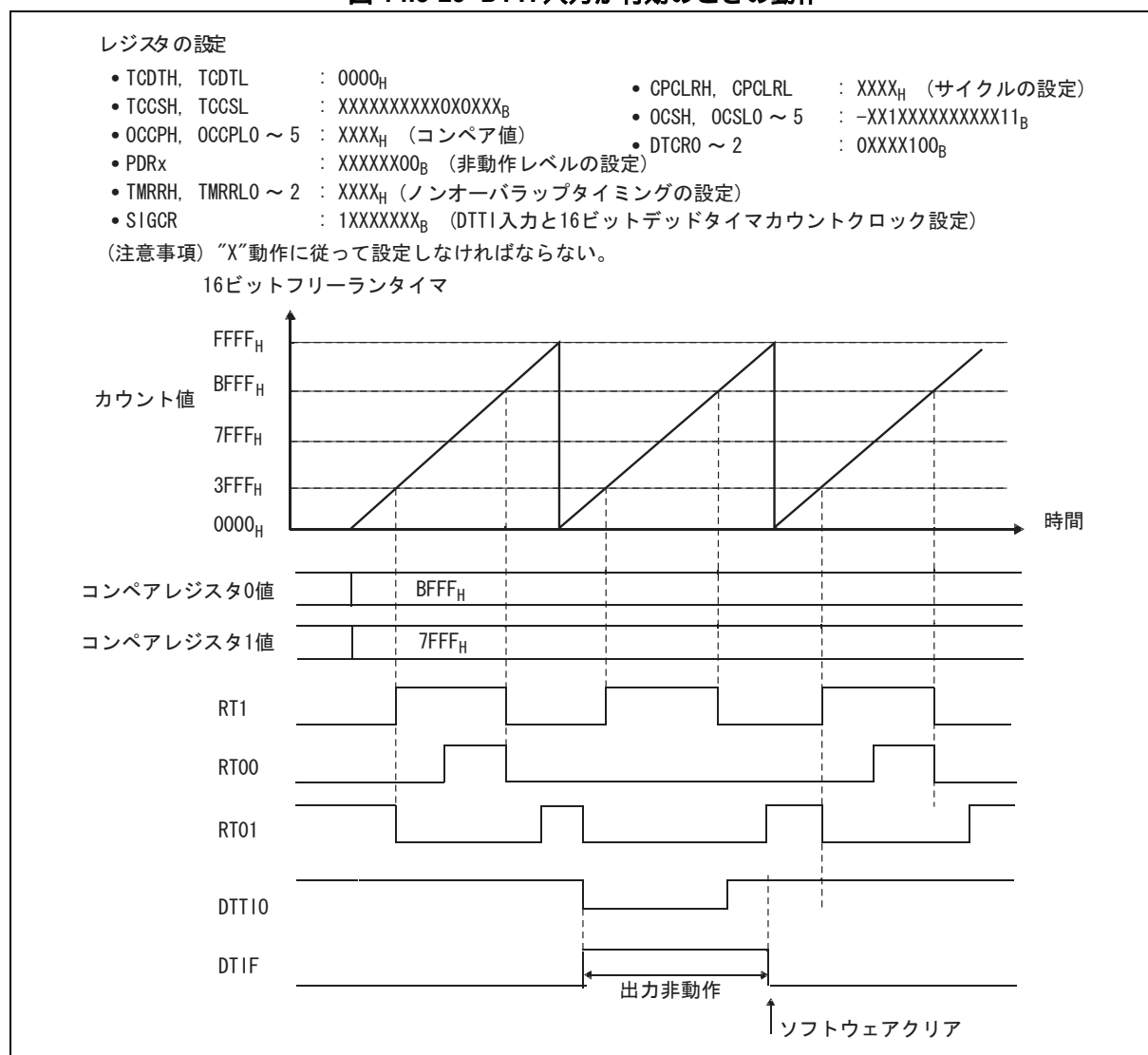
### 14.6.4.3 DTTI 端子制御の動作

波形制御レジスタ (SIGCR) の DTIE:bit15 に "1" を設定すると, RTO0 ~ RTO5 出力を DTTI 端子で制御できます。DTTI 端子の "L" レベルが検出されると, RTO0 ~ RTO5 出力は, 割込みフラグ (SIGCR レジスタの DTIF:bit14) がクリアされるまで非動作レベルに固定されます。RTO0 ~ RTO5 の非動作レベルは, これらの端子を共用しているポートデータレジスタ (PDR) を使ってソフトウェア的に設定できます。

#### ■ DTTI 端子入力動作

DTTI 端子入力の "L" が検出された場合でも, タイマは波形ジェネレータが動作している間は動作を継続しますが, 波形は外部端子 RTO0 ~ RTO5 へは出力されません。

図 14.6-29 DTTI 入力有効のときの動作



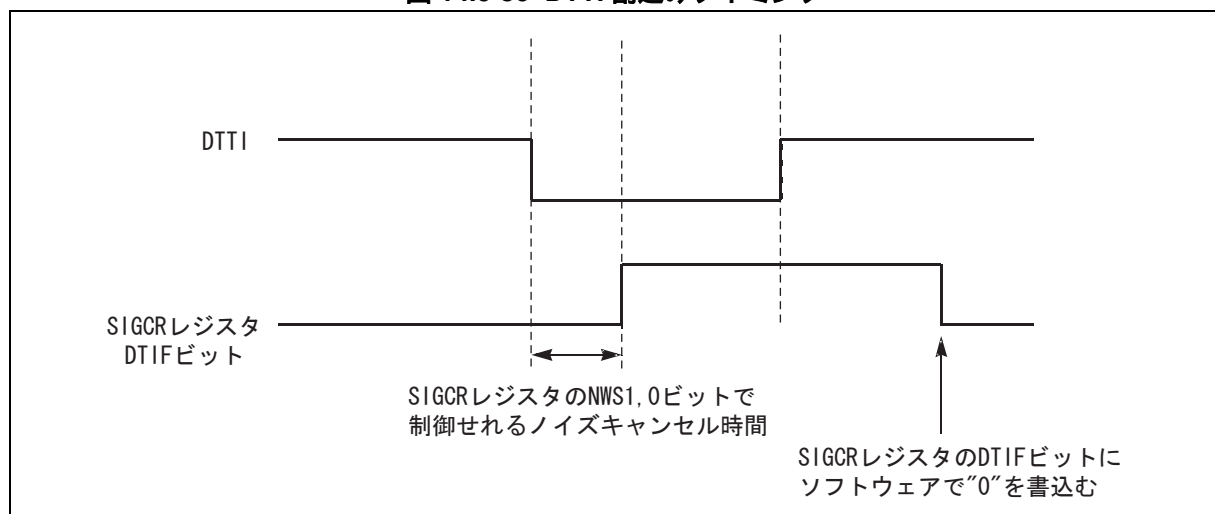
## ■ DTTI 端子ノイズキャンセル機能

波形制御レジスタ (SIGCR) の NRSL:bit13 に "1" を設定すると、DTTI 端子入力のノイズキャンセル機能が有効になります。ノイズキャンセル機能が有効になると、出力端子 (RTO0 ~ RTO5) を非動作レベルに固定するために要する時間が 4, 8, 16 または 32 マシンサイクル (SIGCR レジスタの NWS1, NWS0:bit9, bit8 で選択) だけ遅延します。ノイズキャンセル回路はリソースを使用するので、発振が停止するモード (停止モードなど) 時において DTTI 入力がある場合でも入力は無効になります。

## ■ DTTI 割込み

DTTI の "L" レベルが検出されると、ノイズキャンセル時間が経過した後で DTTI 割込みフラグ (SIGCR レジスタの DTIF:bit14) に "1" が設定され、割込み要求は割込みコントローラへ送信されます。

図 14.6-30 DTTI 割込みタイミング



- 
- < 注意事項 >
- ノイズキャンセル時間内に SIGCR レジスタの NWS1, NWS0 ビットの値が変化した場合は、さらに大きな (NWS1, NWS0) ノイズサイクル値が有効になります。
  - SIGCR レジスタの DTIF:bit14 は、ソフトウェアでのみクリアできます。
-

## 14.7 多機能タイマ使用上の注意

多機能タイマ使用上の注意を以下に示します。

### ■ 16 ビットフリーランタイム使用上の注意

#### ● プログラムによる設定上の注意

- リセットを実行すると、タイマが有効 (TCCSL レジスタの STOP:bit5=0) になった後、その次のカウントクロックでタイマ値が "0000<sub>H</sub>" になり、ゼロ検出割込みフラグに "1" が設定されます。
- タイマモードビット (TCCSL レジスタの MODE:bit4) はバッファを持っているので、その次のカウントサイクルでタイマモードの変更は有効になります。ゼロ検出割込みは、タイマモードがアップカウントモードからアップダウンカウントモードに変わると必ず生成されます。
- ソフトウェアクリア (TCCSL レジスタの SCLR:bit3=1) はタイマを初期化しますが、ゼロ検出割込みを生成しません。

#### ● 割込みの注意

- タイマ状態制御レジスタ上位 (TCCSH) の IRQZF:bit14 に "1" を設定し、次に割込み要求を許可にすると (TCCSH レジスタの IRQZE:bit13=1)、制御は割込み処理から戻ることができません。ZIRQF:bit14 は、必ずクリアしてください。
- タイマ状態制御レジスタ上位 (TCCSH) の ICLRbit9 に "1" を設定し、次に割込み要求を許可にすると (TCCSH レジスタの ICRE:bit8=1)、制御は割込み処理から戻ることができません。ICLR:bit9 は、必ずクリアしてください。
- 16 ビットフリーランタイムは割込みベクトルをほかのリソースと共用しますので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、16 ビットフリーランタイムで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みを禁止にしなければなりません。

### ■ 16 ビットアウトプットコンペア使用上の注意

#### ● 割込みの注意

- コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の IOP1, IOP0:bit7, bit6 に "11<sub>B</sub>" を設定し、次に割込み要求を許可にすると (OCSL レジスタの IOE1, IOE0:bit6, bit5=11<sub>B</sub>)、制御は割込み処理から戻ることができません。IOP0, IOP1 ビットは、必ずクリアしてください。
- 16 ビットアウトプットコンペアは、割込みベクトルをほかのリソースと共用するので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、16 ビットアウトプットコンペアで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みを禁止にしなければなりません。

## ■ 16 ビットインプットキャプチャ使用上の注意

### ● 割込みの注意

- インプットキャプチャ状態制御レジスタ下位 (PICSL01, ICSL23) の ICP3 ~ ICP0 (共に bit7, bit6) に "1" を設定し、次に割込み要求を許可にすると (PICSL01, ICSL23 レジスタの ICE3 ~ ICE0 (共に bit5, bit4) = 11<sub>B</sub>), 制御は割込み処理から戻ることができません。ICP3 ~ ICP0 (共に bit7, bit6) は、必ずクリアしてください。
- インプットキャプチャ端子 (IN) レベルが、ICP3 ~ ICP0 ビットが設定されてから割込みルーチンが処理されるまでの間に切り換わると、有効エッジ指示ビット (ICSH23 レジスタの IEI3, IEI2:bit9, bit8 または PICS01 レジスタの IEI1, IEI0:bit9, bit8) は、検出された最新のエッジを示します。
- 16 ビットインプットキャプチャは割込みベクトルをほかのリソースと共用するので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、16 ビットインプットキャプチャで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みを禁止にしなければなりません。

## ■ 波形ジェネレータ使用上の注意

### ● プログラムによる設定上の注意

- 波形ジェネレータが動作中 (DTCR0 ~ DTCR2 レジスタの TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6 が "001<sub>B</sub>", "010<sub>B</sub>", "100<sub>B</sub>" または "111<sub>B</sub>") に、16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMD8, TMD5, TMD2 (上位は bit10, 下位は bit2), TMD7, TMD4, TMD1 (上位は bit9, 下位は bit1), TMD6, TMD3, TMD0 (上位は bit8, 下位は bit0) ビット値を変更する場合は、トリガソースおよび 16 ビットデッドタイムがカウント中でないことを必ず確認してください。この操作を行わない場合は、以前のトリガでスケジュールされた出力が原因となり予期しない波形が RTO 端子から出力されます。ただし、RTO 出力は、タイマでアンダフローが発生したり、新しいトリガソースで再トリガされたりすると、正常動作に戻ります。
  - トリガソースとは、DTCR0, ~ DTCR2 レジスタの TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "001<sub>B</sub>" の場合は "RT の H レベル" であり、TMD8 ~ TMD0 ビットが "010<sub>B</sub>" の場合は "RT の立上りエッジ" であり、TMD8 ~ TMD0 ビットが "100<sub>B</sub>" の場合は "RT の立上りエッジまたは立下りエッジ" であり、TMD8 ~ TMD0 ビットが "111<sub>B</sub>" の場合は "PPG0 の立上りエッジまたは立下りエッジ" です。  
たとえば、TMD8 ~ TMD0 ビットが "100<sub>B</sub>" から "111<sub>B</sub>" へ変更すると、下記の手順を実行できます。
1. 16 ビットデッドタイムレジスタ (TMRRH, L0 ~ L2) を "0001<sub>H</sub>" のような非常に小さな値を設定する。
  2. RT1, RT3, RT5 の出力を "L" または "H" に設定し、タイマ 0, 1, 2 でアンダフローが発生するまで待つ。
  3. モードビット (TMD8 ~ TMD0) および対応する設定を変更する。
  4. 修正された出力波形が 1 マシンサイクル後 RTO 端子に現れる。



- タイマがカウント中に 16 ビットデッドタイムレジスタ (TMRRH, L0 ~ L2) に値が書き込まれると、この新しい値は次のタイマトリガ時に有効になります。タイマレジスタへアクセスする場合は、必ずワード転送命令 (MOVW A, dir など) をご使用ください。
- タイマがカウントしていない場合のみ、波形制御レジスタ (SIGCR) の DCK2 ~ DCK0:bit12 ~ bit10 を変更してください。
- ノイズキャンセル機能が無効になった場合のみ、波形制御レジスタ (SIGCR) の NWS1, NWS0:bit9, bit8 を変更してください。

### ● 割込みの注意

- 16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMIF2 ~ TMIF0 (上位は bit12, 下位は bit4) に "1" を設定し、次に割込み要求を許可にすると (DTCR0 ~ DTCR 2 レジスタの TMIE2 ~ TMIE0 (上位は bit11, 下位は bit3=1), 制御は割込み処理から戻ることができません。TMIF ビットは、必ずクリアしてください。
- 波形制御レジスタ (SIGCR) の DTIF:bit14 に "1" を設定すると、制御は割込み処理から戻ることができません。DTIF ビットは、必ずクリアしてください。
- 波形ジェネレータは割込みベクトルをほかのリソースと共用するので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、波形ジェネレータで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みをディスエーブルにしなければなりません。

## 14.8 多機能タイマのプログラム例

多機能タイマのプログラム例を示します。

### ■ 16 ビットフリーランタイマのプログラム例

#### ● 処理

- 4ms コンペアクリア割込みは 16 ビットフリーランタイマ 0 により生成します。
- 本タイマはアップダウンモード時にコンペアクリアタイマの再生成のために使用します。
- EI<sup>2</sup>OS は使用しません。
- 16MHz はマシクロック用, 62.5ms はカウントクロック用です。

#### ● コーディング例

```

ICR11 EQU 0000BBH ;16 ビットフリーランタイマ用割込み制御レジスタ
TCCSL EQU 00005EH ; タイマ制御状態レジスタ
CPCLRBH EQU 00005BH ; コンペアクリアバッファレジスタ
ICLR EQU TCCSL:9 ; 割込み要求フラグビット
;----- メインプログラム -----
CODE CSEG
START:
;      : ; スタックポインタ (SP) が既に初期化されているものと仮
定
      AND CCR,#0BFH; 割込み禁止
      MOV I:ICR11,#00H; 割込みレベル 0 (最強)
      MOVW I:CPCLRBH,#0FFFFH;16 ビットフリーランタイマをアップカウントか
          らダウ
          ; ンカウントへ変更するためにコンペアクリア値を設定
      MOVW I:TCCSL,#0110H; アップダウンモード, 62.5ms カウントクロックを設定
          ; コンペアクリア割込みを許可
          ; 割込みマスクを禁止
          ; 割込みフラグと有効タイマをクリア
      MOV ILM,#07H ;PS 中の ILM をレベル 7 に設定
      OR CCR,#40H ; 割込み許可
LOOP:  MOV A,#00H ; 無限ループ
      MOV A,#01H ;
      BRA LOOP ;
;----- 割込みプログラム -----
WARI:
      CLRB I:ICLR ; 割込み要求フラグをクリア
;      :
;      ユーザ処理
;      :
      RETI ; 割込みから復帰

CODE ENDS

```

```

;----- ベクタ設定 -----
VECT      CSEG  ABS=0FFH
          ORG   0FF74H      ; 割込み #34 (22H) 用ベクタを設定
          DSL   WARI
          ORG   0FFDCH      ; リセットベクタを設定
          DSL   START
          DB    00H          ; シングルチップモードの設定
VECT      ENDS
          END    START

```

## ■ 16 ビットアウトプットコンペアのプログラム例

### ● 処理

- 16 ビットフリーランタイマのカウンタ値がアウトプットコンペア用と一致したとき、アウトプットコンペア一致が生成します。
- 16 ビットフリーランタイマがアップカウントモード時に使用します。
- EI<sup>2</sup>OS は使用しません。
- 16MHz はマシクロック用、62.5ms は 16 ビットフリーランタイマのカウンタクロック用です。

### ● コーディング例

```

ICR00     EQU    0000B0H      ; アウトプットコンペア 0 用割込みカウンタレジスタ
TCCSL     EQU    00005EH      ; タイマ制御状態レジスタ
CPCLRBH   EQU    00005BH      ; コンペアクリアバッファレジスタ
OCCPL0    EQU    000070H      ; アウトプットコンペアレジスタ 0
OCCPL1    EQU    000072H      ; アウトプットコンペアレジスタ 1
OCSL01    EQU    00007CH      ; コンペア制御レジスタ
IOP       EQU    OCS01:6      ; 割込み要求フラグビット
;----- メインプログラム -----
CODE      CSEG
START:
; :                               ; スタックポインタ (SP) が既に初期化されているものと仮定
AND       CCR, #0BFH          ; 割込み禁止
MOV       I:ICR00, #00H        ; 割込みレベル 0 (最強)
MOVW     I:TCCSL, #0000H       ; 16 ビットフリーランタイマを有効
                                   ; アップカウントモードを設定
MOVW     I:OCCPL0, #0BFFFH; アウトプットコンペアレジスタ 0 を設定
MOVW     I:OCCPL1, #07FFFH; アウトプットコンペアレジスタ 1 を設定
MOVW     I:OCSL01, #0C1FH; アウトプットコンペア出力を有効
                                   ; コンペア一致割込み 0 を有効
                                   ; 割込みフラグと有効アウトプットコンペアをクリア
MOV       ILM, #07H           ; PS 中の ILM をレベル 7 に設定
OR        CCR, #40H           ; 割込み許可
LOOP:    MOV     A, #00H        ; 無限ループ
MOV       A, #01H              ;
BRA       LOOP                 ;
;----- 割込みプログラム -----
WARI:
        CLRB    I:IOP          ; 割込みレジスタフラグをクリア
;
; ユーザ処理
;

```

```

                                RETI                                ; 割込みから復帰

CODE    ENDS
;----- ベクタ設定 -----
VECT    CSEG    ABS=0FFH
        ORG     0FFCCH      ; 割込み #12 (0CH) 用ベクタの設定
        DSL     WARI
        ORG     0FFDCH      ; リセットベクタの設定
        DSL     START
        DB      00H         ; シングルチップモードの設定
VECT    ENDS
        END     START

```



# 第15章

---

## マルチパルスジェネレータ

マルチパルスジェネレータの機能と動作について説明します (MB90467 では、未搭載となります)。

- 15.1 マルチパルスジェネレータの概要
- 15.2 マルチパルスジェネレータのブロックダイアグラム
- 15.3 マルチパルスジェネレータの端子
- 15.4 マルチパルスジェネレータのレジスタ
- 15.5 マルチパルスジェネレータの割込み
- 15.6 マルチパルスジェネレータの動作
- 15.7 マルチパルスジェネレータ使用上の注意
- 15.8 マルチパルスジェネレータのプログラム例

## 15.1 マルチパルスジェネレータの概要

マルチパルスジェネレータは、16 ビット PPG タイマ、16 ビットリロードタイマ、波形シーケンサから構成されています。波形シーケンサを使用すると、16 ビット PPG タイマ出力信号をマルチパルスジェネレータの入力信号 (SNI2 ~ SNI0) に従ってマルチパルスジェネレータ出力 (OPT5 ~ OPT0) へ送出できます。また、OPT5 ~ OPT0 出力信号は、緊急の場合は DTTI 入力 (DTTI1) を使用してハードウェア的に終了させることができます。OPT5 ~ OPT0 出力信号は、望ましくないグリッチを除去するために PPG 信号と同期します。

なお、16 ビットリロードタイマと 16 ビット PPG タイマについての詳細は、「第 12 章 16 ビットリロードタイマ」と「第 13 章 16 ビット PPG タイマ」のそれぞれをご参照ください。

### ■ 波形シーケンサの機能

#### ● 出力信号制御

波形シーケンサを使用すると、マルチパルスジェネレータ出力 (OPT5 ~ OPT0) において 16 ビット PPG 波形出力と DC チョップパルス波形出力を生成できます。

- マルチパルスジェネレータの位置検出入力 (SNI2 ~ SNI0) により有効エッジが検出された場合または 16 ビットリロードタイマでアンダフローが発生した場合または OPDRH, OPDRL レジスタへ書き込みされた場合に、出力データバッファレジスタ (OPDBR0 ~ OPDBRB) のいずれかのデータが出力データレジスタ (OPDRH, OPDRL) へロードされます。
- 出力データレジスタ (OPDRH, OPDRL) は、OPT 出力端子 (OPT5 ~ OPT0) へ出力される 16 ビット PPG タイマ出力を決定します。異なる複数の出力データバッファレジスタ (OPDBR0 ~ OPDBRB) のデータを出力データレジスタ (OPDRH, OPDRL) へロードすると、OPT 出力端子 (OPT5 ~ OPT0) からさまざまな組合わせの波形を出力することが可能になります。  
したがって、16 ビット PPG タイマ出力は、出力データレジスタ (OPDRH, OPDRL) および 12 個の出力データバッファレジスタ (OPDBR0 ~ OPDBRB) に設定されているシーケンスに従って、マルチパルスジェネレータ出力への信号送出を制御したり、出力先を任意の OPT 出力から別の OPT 出力に切り換えたりすることができます。また、16 ビットリロードタイマは、OPT 出力切換え時に遅延を挿入することもできます。
- OPDBR0 ~ OPDBRB レジスタから OPDRH, OPDRL レジスタへのデータ転送の組合わせを、表 15.1-1 に示します。

表 15.1-1 OPDBR0 ~ OPDBRB レジスタから OPDRH, OPDRL レジスタへのデータ転送

組合せ	OPDBR0 ~ OPDBRB から OPDRH, OPDRL へのデータ転送
1	OPDBR0 への書込みがソフトウェアで行われた後の OPDBR0 から OPDRH, OPDRL へのデータ転送
2	16 ビットリロードタイマ 0 のアンダフローにより起動する
3	位置検出入力端子 (SNI2 ~ SNI0) により起動する
4	16 ビットリロードタイマ 0 のアンダフローにより起動する 16 ビットタイマは、位置検出比較回路で開始する
5	16 ビットリロードタイマ 0 のアンダフローまたは位置検出入力により起動する

- 波形シーケンサは、位置を検出できなかった場合にモータの速度を測定したり、OPT 出力を禁止にしたりするために使用できる 16 ビットタイマを内蔵しています。
- DTTI 端子入力 (DTTI1) による強制的停止制御
  - 外部端子は、発振が停止している場合でもクロックレスで入力可能な DTTI 端子入力で制御できます (端子レベルは、各端子またはソフトウェアで設定できます)。DTTI 入力には、選択可能なノイズフィルタが存在します。DTTI1 端子のノイズフィルタのノイズ幅を表 15.1-2 に示します。

表 15.1-2 ノイズフィルタのノイズ幅

選択	DTTI 端子と SNI2 ~ SNI0 端子のノイズ幅
1	4 マシンサイクルノイズを除去する
2	8 マシンサイクルノイズを除去する
3	16 マシンサイクルノイズを除去する
4	32 マシンサイクルノイズを除去する

● 出力信号と PPG との間の同期化

シーケンサ状態遷移時点におけるショートパルス (またはグリッチ) の発生を回避するには、書込みタイミング (WTO) を遅延させ、PPG 出力波形のその次のエッジと同期化させる必要があります。詳細は、図 15.1-1 と図 15.1-2 をご参照ください。この機能は、ソフトウェアで有効または無効にできます。入力制御レジスタ (IPCUR) の WTS1:bit15 と WTS0:bit14 は、この機能が無効にしたり同期化対象の PPG エッジの極性を選択したりするために使用します。



図 15.1-1 PPG 立上がりエッジの同期化

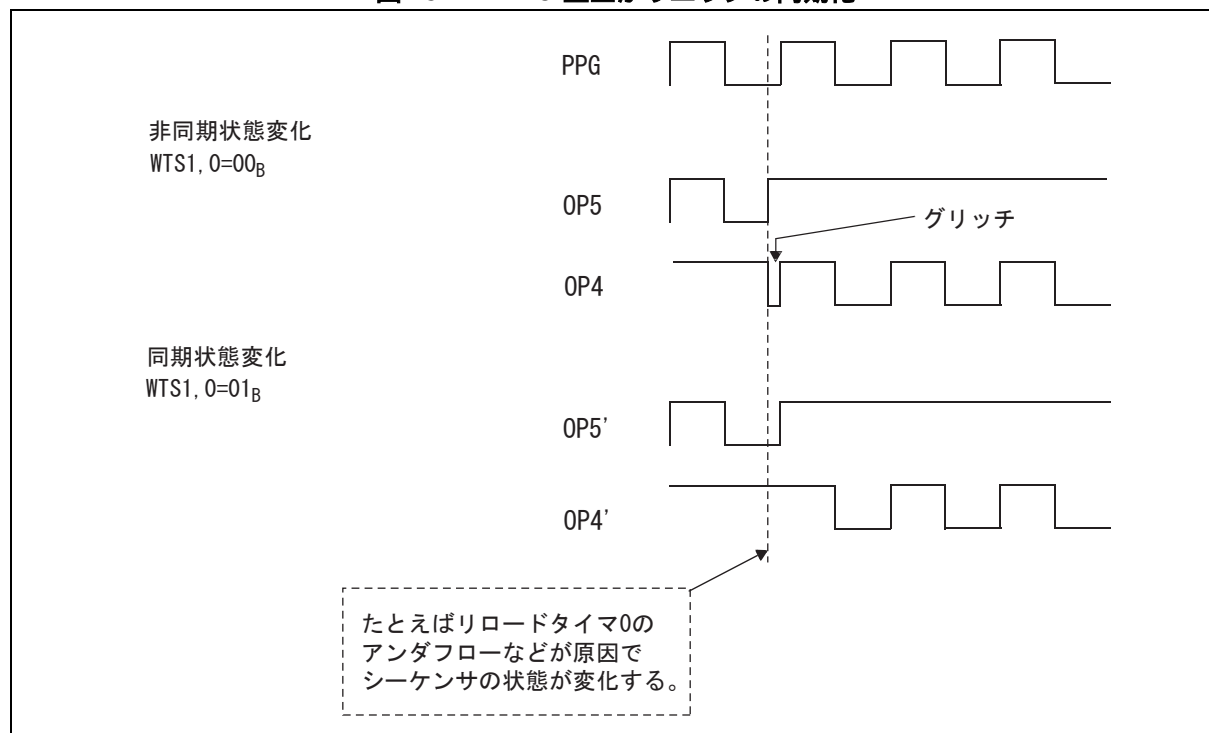
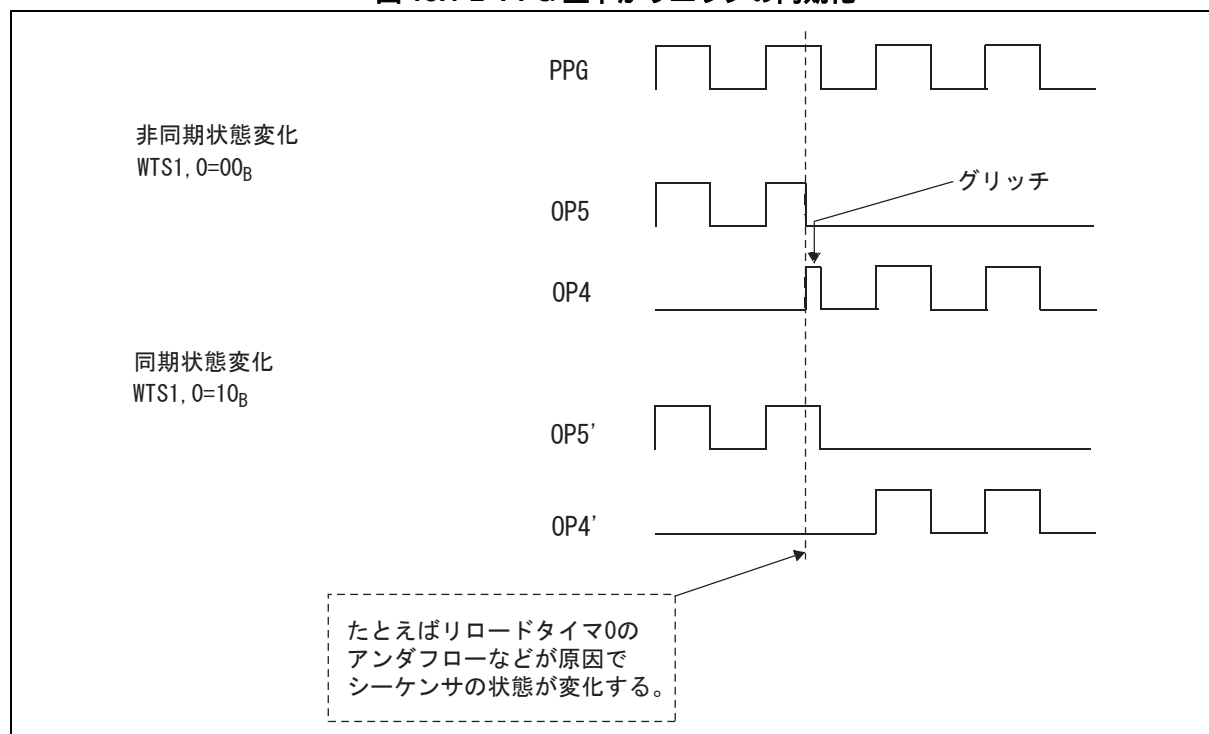


図 15.1-2 PPG 立下がりエッジの同期化



< 注意事項 > PPG 同期モードからほかの PPG 同期モードへの変化（例：立上がりエッジ同期から立下り同期，あるいはその逆）は，禁止です。非同期モードはこのような変化に対して遷移しなければなりません。

## ● 入力位置検出制御

マルチパルスジェネレータ入力端子 (SNI2 ~ SNI0) の入力信号は、直流モータのロータ位置を検出するために使用します。SNI2 ~ SNI0 の各入力端子には、ノイズフィルタが存在します。このノイズフィルタのノイズ幅を表 15.1-2 に示します。入力位置検出回路の条件を以下に示します。

- SNI2 ~ SNI0 の各入力では、3 種類のエッジのいずれかを選択できます。すなわち、立上りエッジ、立下りエッジまたは両エッジを選択できます。
- SNI2 ~ SNI0 入力のレベルは、出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 と比較されます。

上記の条件が満たされると、OPDBR0 ~ OPDBRB レジスタと OPDRH, OPDRL レジスタの間におけるデータ転送用書込みタイミング信号が生成されます。

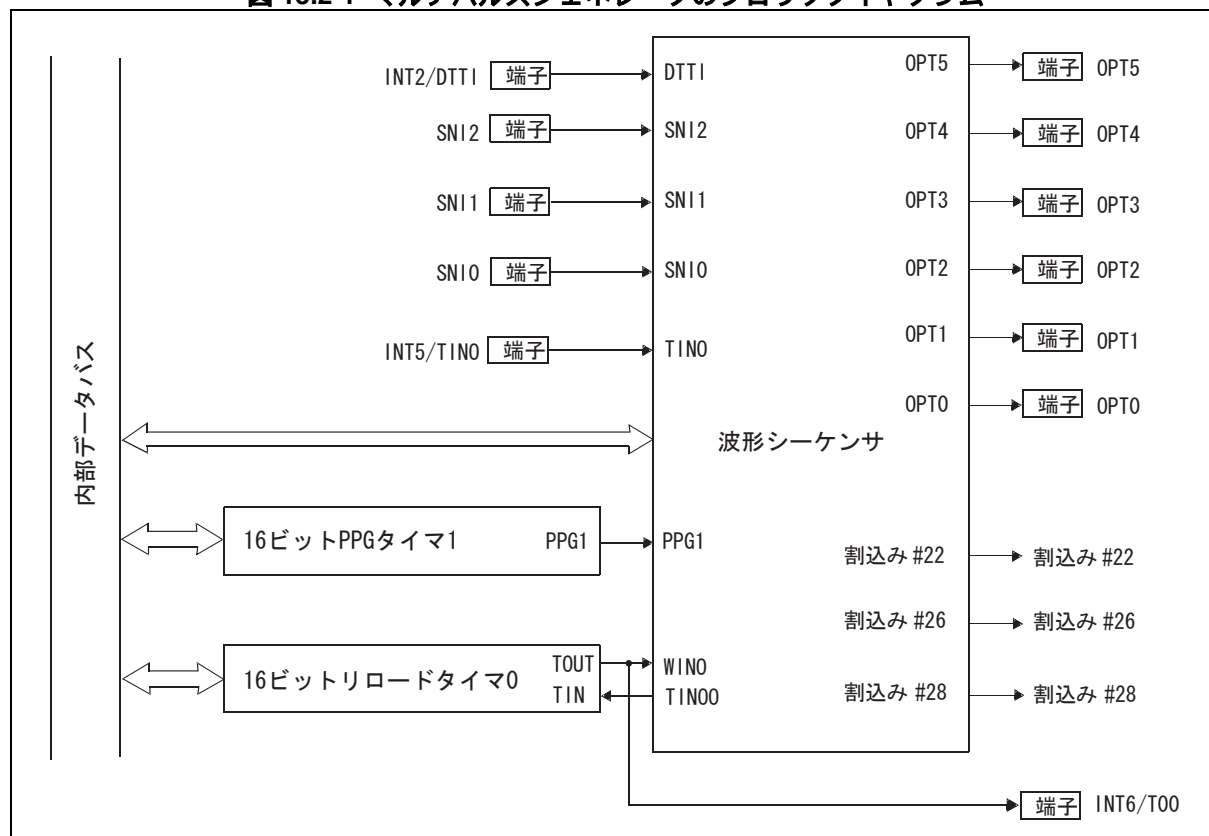
また、各入力 (SNI2 ~ SNI0) のエッジ検出を無効または有効にできます。

## 15.2 マルチパルスジェネレータのブロックダイアグラム

マルチパルスジェネレータのブロックダイアグラムを図 15.2-1 に示し、波形シーケンサのブロックダイアグラムを図 15.2-2 に示します。

### ■ マルチパルスジェネレータのブロックダイアグラム

図 15.2-1 マルチパルスジェネレータのブロックダイアグラム



#### ● 16 ビット PPG タイマ 1

16 ビット PPG タイマ 1 は、波形シーケンサの PPG 信号を供給するために使用します。16 ビット PPG タイマ 1 についての詳細は、「第 13 章 16 ビット PPG タイマ」をご参照ください。

#### ● 16 ビットリロードタイマ 0

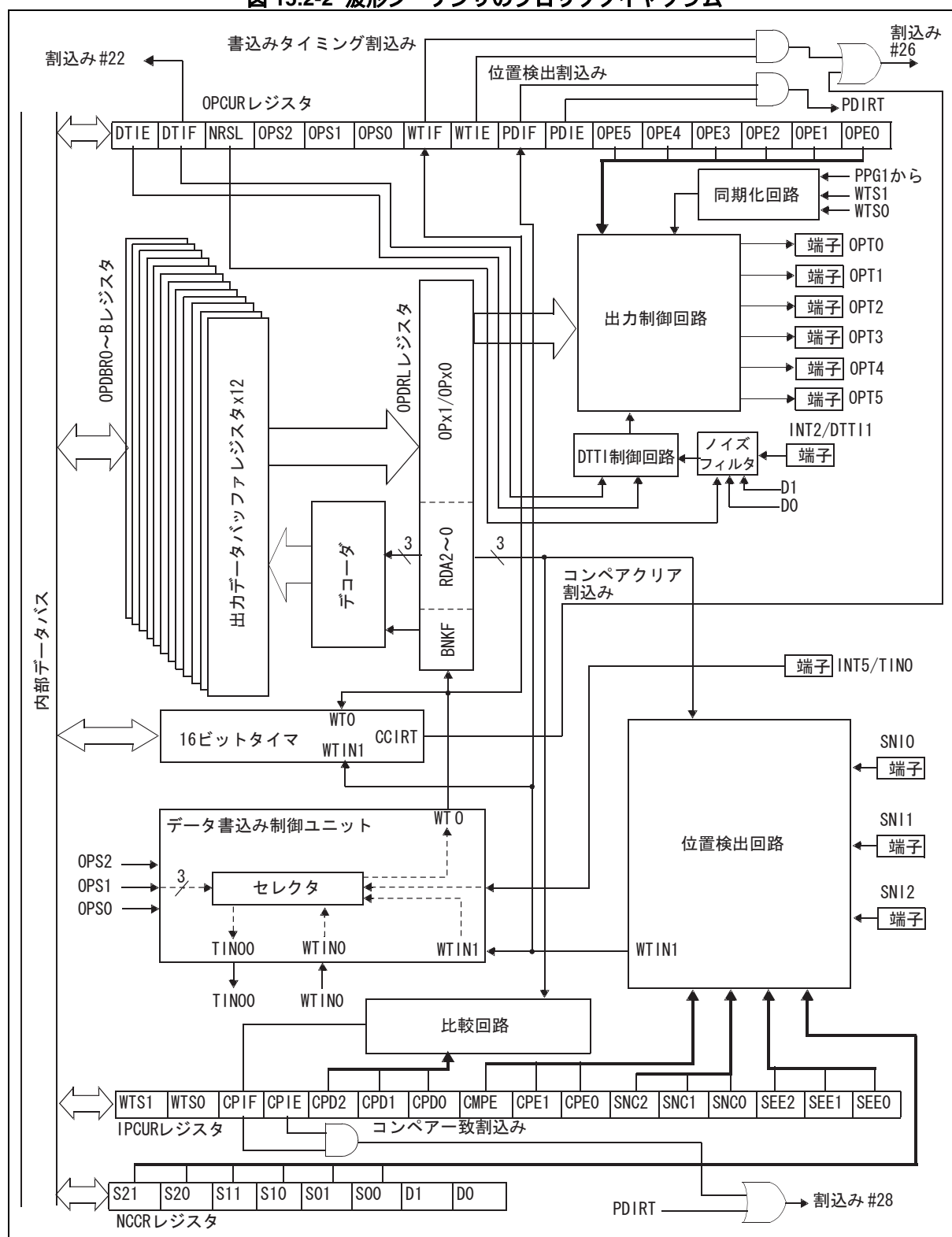
16 ビットリロードタイマ 0 は、波形シーケンサのインタバルタイマとして使用します。16 ビットリロードタイマ 0 についての詳細は、「第 12 章 16 ビットリロードタイマ」をご参照ください。

#### ● 波形シーケンサ

波形シーケンサは、さまざまな波形を生成できるマルチパルスジェネレータの心臓部です。波形シーケンサのブロックダイアグラムを図 15.2-2 に示します。

## ■ 波形シーケンサのブロックダイアグラム

図 15.2-2 波形シーケンサのブロックダイアグラム



● 16 ビットタイマ

16 ビットタイマは、モータ速度を検査するインターバルタイマとして使用します。また直流センサレスモータを制御するときの異常検出タイマとしても使用します。16 ビットタイマの詳細を図 15.2-3 に示します。

● 比較回路

比較回路は、モータ方向を変更する際に出力データレジスタ上位(OPDRH)の RDA2 ~ RDA0:bit14 ~ bit12 を入力制御レジスタ上位(IPCUR)の CPD2 ~ CPD0:bit11 ~ bit9 と比較するために使用します。比較の結果、一致が検出されると、コンペアー一致割込みが生成されます。

● データ書込み制御ユニット

データ書込み制御ユニットは、出力データバッファレジスタ(OPDBR0 ~ OPDBRB)から出力データレジスタ(OPDRH, OPDRL)へのデータ転送用書込み信号(WTO)を生成するために使用します。データ書込み制御ユニットの詳細を図 15.2-4 に示します。

● デコーダ

デコーダは、出力データレジスタにロードされる出力データバッファレジスタ(OPDBR0 ~ OPDBRB)を選択するために使用する出力データレジスタ(OPDRH, OPDRL)をデコードするために使用します。

● DTTI 制御

DTTI 制御は、緊急の場合に DTTI 入力レベル "0" がトリガとなりマルチパルスジェネレータ出力を停止するために使用します。

● ノイズフィルタ

ノイズフィルタは、入力信号(4 種類のサンプリングクロックの中から選択される)のノイズを除去するために使用します。

● 出力制御ユニット

出力制御ユニットは、マルチパルスジェネレータ出力端子(OPT5 ~ OPT0)への PPG 信号を有効または無効にするために使用します。

● 位置検出回路

位置検出回路は、位置入力端子(SNI2 ~ SNI0)のエッジ/レベルを検出するために使用します。位置検出回路の詳細を図 15.2-5 に示します。

● 同期回路

同期回路は、OPT5 ~ OPT0 出力を PPG 信号と同期化するために使用します。

● ノイズキャンセル制御レジスタ(NCCR)

ノイズキャンセル制御レジスタ(NCCR)は、4 種類のサンプリングクロックの中からいずれかを選択するために使用します(選択されたサンプリングクロックは、ノイズフィルタでノイズの除去に使用されます)。

● 出力制御レジスタ (OPCUR)

出力制御レジスタ (OPCUR) は、書込みタイミング割込みフラグを有効にし、位置検出割込みフラグを有効にし、データ転送方式を設定し、OPT5 ~ OPT0 端子と DTTI1 端子の制御を設定します。

● 出力データバッファレジスタ (OPDBR0 ~ OPDBRB)

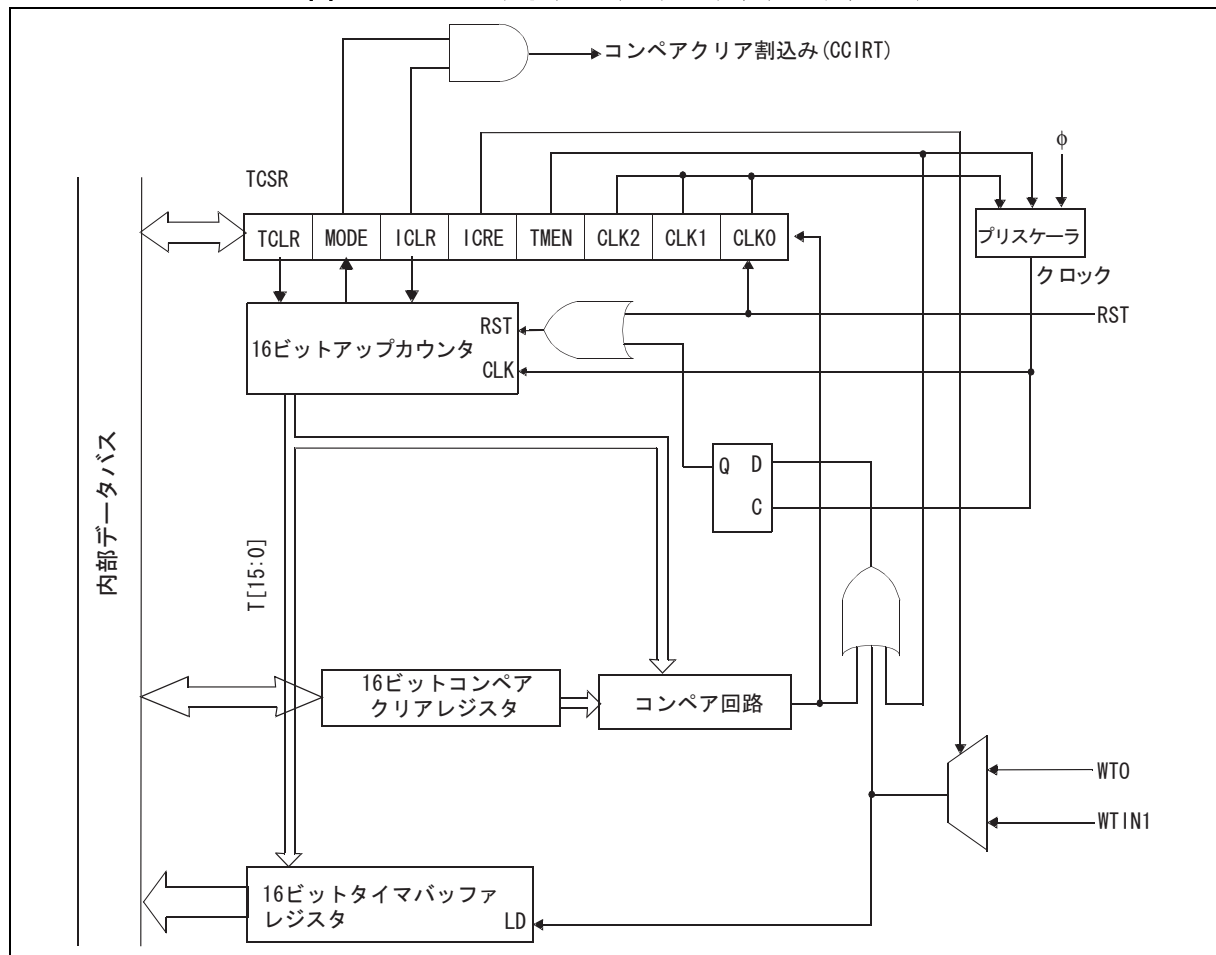
出力データバッファレジスタは、12 個のレジスタ (OPDBR0 ~ OPDBRB) から構成されています。BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 で指定される OPDBRx レジスタ値は、データ書込み制御ユニットで生成される書込み信号の立上りエッジで OPDRH, OPDRL レジスタへロードされます。

● 出力データレジスタ (OPDRH, L)

出力データレジスタ (OPDRH, L) は、OPT5 ~ OPT0 端子への出力データを格納するために使用します。

## ■ 16 ビットタイマのブロックダイアグラム

図 15.2-3 16 ビットタイマのブロックダイアグラム



● 16 ビットアップカウンタ

16 ビットアップカウンタは、16 ビットアップカウンタのカウント値とコンペアクリアレジスタの間で一致が検出されると、クリアされます。

● コンペア回路

コンペア回路は、16 ビットアップカウンタのカウンタ値とコンペアクリアレジスタのデータを比較するために使用します。

● コンペアクリアレジスタ (CPCR<sub>H</sub>, CPCRL)

コンペアクリアレジスタ (CPCR<sub>H</sub>, CPCRL) は、16 ビットアップカウンタのカウンタ値と比較する 16 ビットのデータを格納するために使用します。

● タイマバッファレジスタ (TMBR<sub>H</sub>, TMBRL)

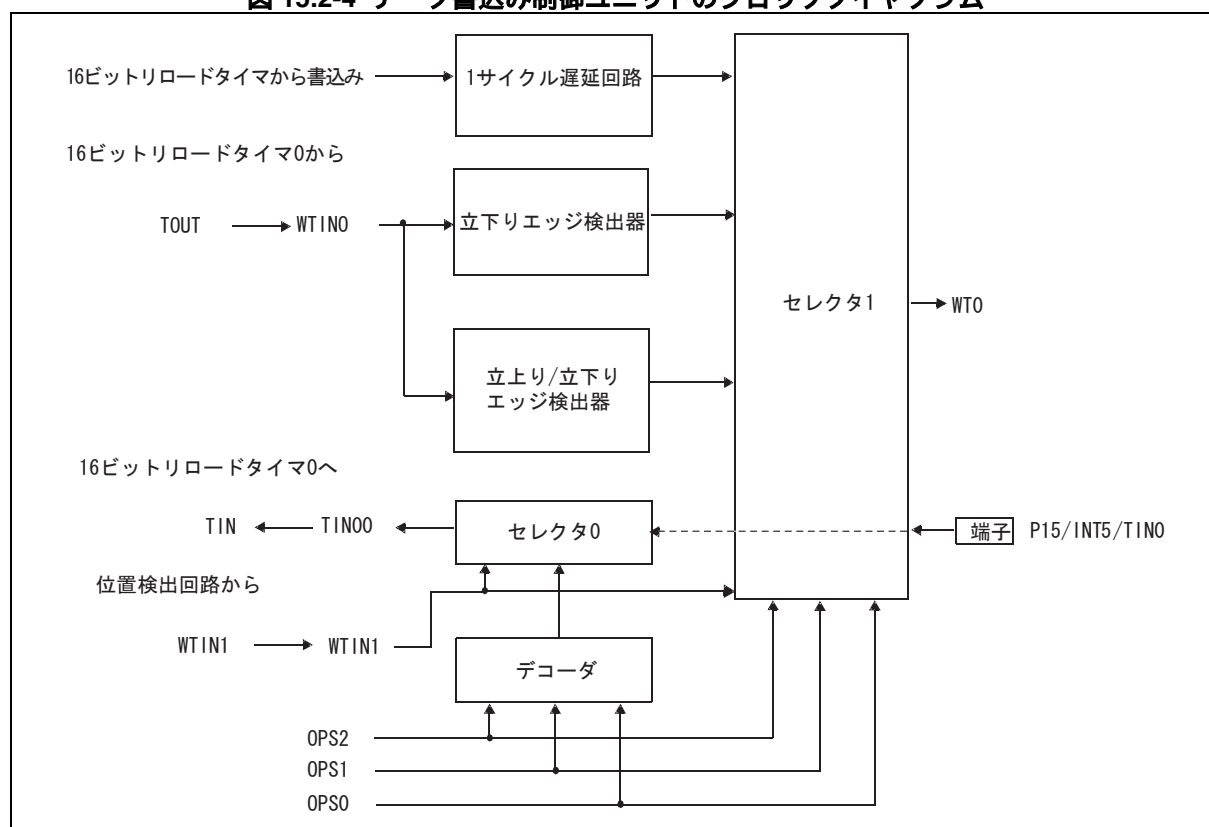
タイマバッファレジスタ (TMBR<sub>H</sub>, TMBRL) は、書込みタイミング割込みまたは位置検出割込みが発生したときの 16 ビットアップカウンタのカウンタ値を格納するために使用します。

● タイマ状態制御レジスタ (TCSR)

タイマ状態制御レジスタ (TCSR) は、クロック周波数や割込み有効 / 禁止などの、16 ビットタイマの動作を制御するために使用します。

■ データ書込み制御ユニットのブロックダイアグラム

図 15.2-4 データ書込み制御ユニットのブロックダイアグラム



● 1 サイクル遅延回路

1 サイクル遅延回路は、出力データバッファレジスタ 0 (OPDBR0) へ書込みされたときにトリガ信号を 1 マシンサイクル遅延させるために使用します。

● 立下りエッジ検出器

立下りエッジ検出器は、16 ビットリロードタイマ 0 が出力 (TOUT) する立下りエッジを検出するために使用します。

● 立上り / 立下りエッジ検出器

立上り / 立下りエッジ検出器は、16 ビットリロードタイマ 0 が出力 (TOUT) する立上り / 立下りエッジを検出するために使用します。

● セレクタ 0

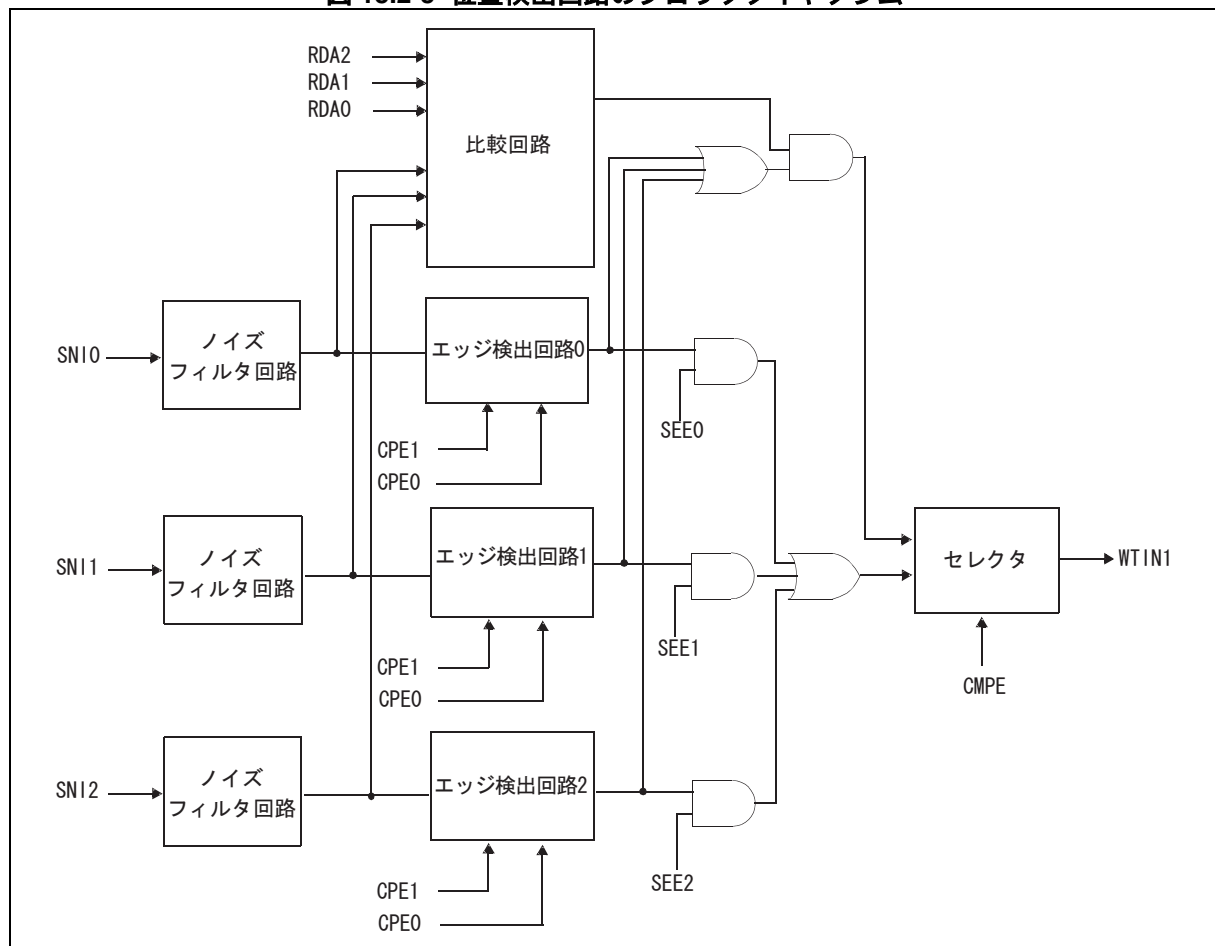
セレクタ 0 は、16 ビットリロードタイマ 0 のカウントを有効にするために位置検出回路の WTIN1 か、あるいは外部端子 (INT5/TIN0) を選択するために使用します。

● セレクタ 1

セレクタ 1 は、書込みタイミング信号 (WTO) を生成するために 16 ビットリロードタイマ 0 からの OPDBR0 書込みまたは TOUT または位置検出回路の WTIN1 を選択するために使用します。

■ 位置検出回路のブロックダイアグラム

図 15.2-5 位置検出回路のブロックダイアグラム



● 比較回路

比較回路は、位置検出入力 (SN12 ~ SN10) のレベルを出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 と比較するために使用します。セレクタが選択された場合は、一致が検出されるとデータ書込みタイミング出力信号が生成されます。



- エッジ検出回路 0, 1, 2

エッジ検出回路 0, 1, 2 は同じ機能回路です。

エッジ検出回路は、位置入力 (SNI2 ~ SNI0) のエッジを 3 種類のエッジ設定と比較するために使用します。セレクトが選択された場合は、SNI2 ~ SNI0 入力のいずれかから有効エッジが検出されると、データ書込みタイミング出力信号が生成されます。

- ノイズフィルタ

ノイズフィルタは、入力信号 (4 種類のサンプリングクロックの中から選択される) のノイズを除去するために使用します。

- セレクト

セレクトは、データ書込み制御ユニットへのデータ書込みタイミング出力信号を生成するために、エッジ検出回路または比較回路を選択します。

## 15.3 マルチパルスジェネレータの端子

マルチパルスジェネレータの端子について説明します。また端子のブロックダイアグラムも示します。

### ■ マルチパルスジェネレータの端子

マルチパルスジェネレータは、P00/OPT0 ~ P05/OPT5, P43/SNI0 ~ P45/SNI2, P12/INT2/DTTI1 および P15/INT5/TIN0 を使用します。

#### ● P00/OPT0 ~ P05/OPT5 端子

P00/OPT0 ~ P05/OPT5 端子は、汎用入出力ポート (P00 ~ P05) またはマルチパルスジェネレータの波形出力として機能できます。

波形出力ビットを許可 (OPCLR レジスタの OPE5 ~ OPE0:bit5 ~ bit0=11111<sub>B</sub>) にすると、P00/OPT0 ~ P05/OPT5 端子は、ポート方向レジスタ (DDR0 レジスタの bit5 ~ bit0) の値とは無関係に自動的に出力端子として設定され、OPT5 ~ OPT0 端子として機能します。

#### ● P43/SNI0 ~ P45/SNI2 端子

P43/SNI0 ~ P45/SNI2 端子は、汎用入出力ポート (P43 ~ P45) またはマルチパルスジェネレータの位置検出入力として機能できます。

P43/SNI0 ~ P45/SNI2 端子は、SNI2 ~ 0 端子として使用する場合は、ポート方向レジスタで入力ポートとして設定 (DDR4 レジスタの bit5 ~ bit3=000<sub>B</sub>) してください。

#### ● P12/INT2/DTTI1 端子

P12/INT2/DTTI1 端子は、汎用入出力ポート (P12)、外部割込み (INT2) またはマルチパルスジェネレータの DTTI1 入力として機能できます。

P12/INT2/DTTI1 端子は、DTTI1 端子として使用する場合は、ポート方向レジスタで入力ポートとして設定 (DDR1 レジスタの bit1=0) してください。

#### ● P15/INT5/TIN0 端子

P15/INT5/TIN0 端子は、汎用入出力ポート (P15)、外部割込み (INT5) またはマルチパルスジェネレータの 16 ビットリロードタイマ 0 入力として機能できます。

P15/INT5/TIN0 端子は、TIN0 端子として使用する場合は、ポート方向レジスタで入力ポートとして設定 (DDR1 レジスタの bit5=0) してください。

## ■ マルチパルスジェネレータ端子のブロックダイアグラム

図 15.3-1 P00/OPT0 ~ P05/OPT5 端子のブロックダイアグラム

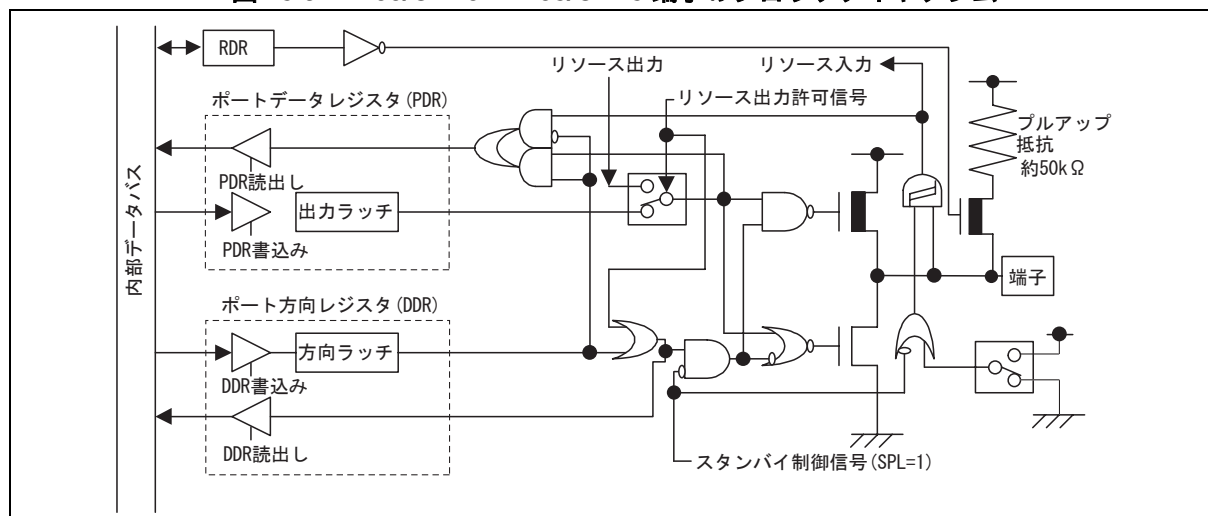


図 15.3-2 P12/INT2/DTTI1 端子および P15/INT5/TIN0 端子のブロックダイアグラム

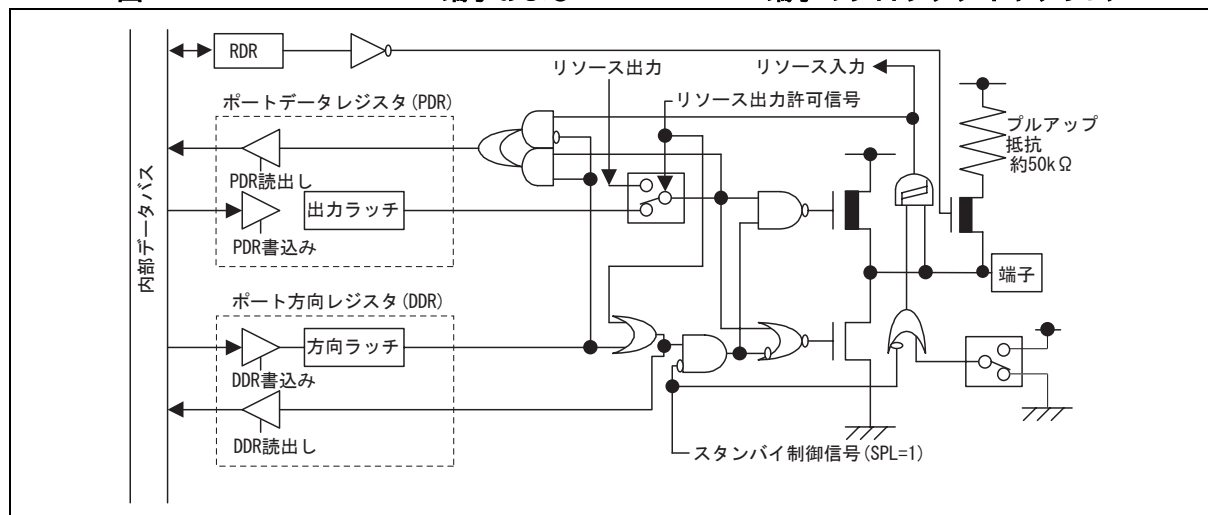
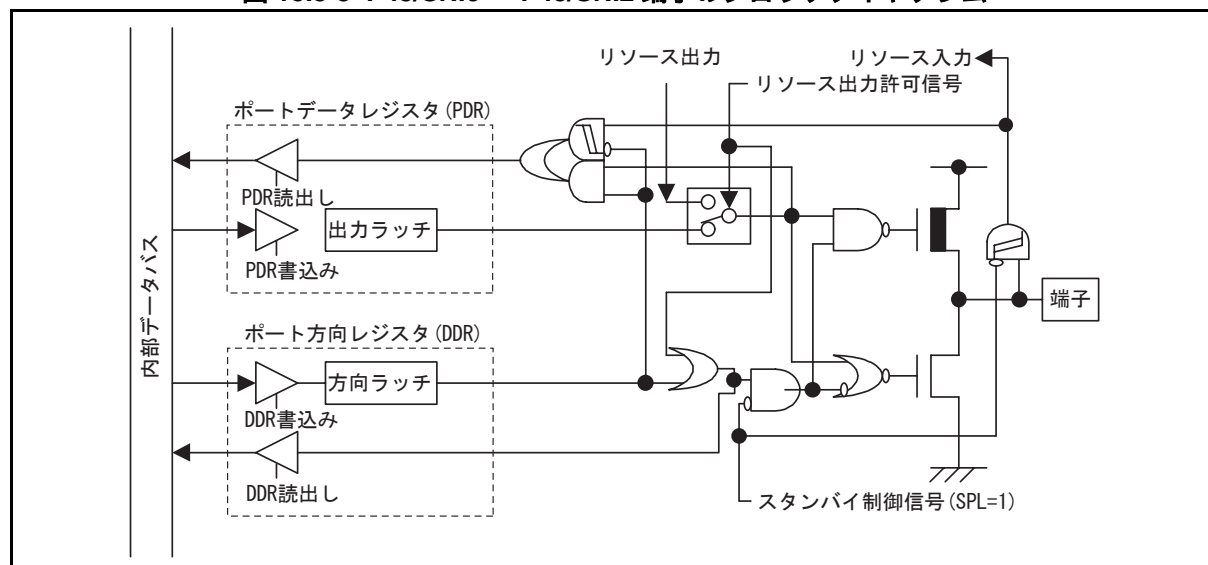


図 15.3-3 P43/SNI0 ~ P45/SNI2 端子のブロックダイアグラム



## 15.4 マルチパルスジェネレータのレジスタ

マルチパルスジェネレータのレジスタについて説明します。

### ■ マルチパルスジェネレータのレジスタ

図 15.4-1 マルチパルスジェネレータのレジスタ

出力制御レジスタ (上位)								
bit	15	14	13	12	11	10	9	8
	DTIE	DTIF	NRSL	OPS2	OPS1	OPS0	WTIF	WTIE
	OPCUR							
リード/ライト	⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値	⇒	0	0	0	0	0	0	0
出力制御レジスタ (下位)								
bit	7	6	5	4	3	2	1	0
	PDIF	PDIE	OPE5	OPE4	OPE3	OPE2	OPE1	OPE0
	OPCLR							
リード/ライト	⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値	⇒	0	0	0	0	0	0	0
出力データレジスタ (上位)								
bit	15	14	13	12	11	10	9	8
	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40
	OPDRH							
リード/ライト	⇒	R	R	R	R	R	R	R
初期値	⇒	0	0	0	0	X	X	X
出力データレジスタ (下位)								
bit	7	6	5	4	3	2	1	0
	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00
	OPDRL							
リード/ライト	⇒	R	R	R	R	R	R	R
初期値	⇒	X	X	X	X	X	X	X

( 続く )

出力データバッファレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	OPDBR0~B
リード/ライト	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

出力データバッファレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	OPDBR0~B
リード/ライト	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

入力制御レジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	WTS1	WTS0	CP1F	CP1E	CPD2	CPD1	CPD0	CMPE	IPCUR
リード/ライト	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

入力制御レジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	CPE1	CPE0	SNC2	SNC1	SNC0	SEE2	SEE1	SEE0	IPCLR
リード/ライト	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	⇒ R/W	
初期値	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	⇒ 0	

( 続く )

( 続き )

## コンペアクリアレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	CPCR <sub>H</sub>
リード/ライト ⇨	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	

## コンペアクリアレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	CPCR <sub>L</sub>
リード/ライト ⇨	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	

## タイマバッファレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	T15	T14	T13	T12	T11	T10	T09	T08	TMBR <sub>H</sub>
リード/ライト ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	0	0	0	0	0	0	0	0	

## タイマバッファレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	T07	T06	T05	T04	T03	T02	T01	T00	TMBR <sub>L</sub>
リード/ライト ⇨	R	R	R	R	R	R	R	R	
初期値 ⇨	0	0	0	0	0	0	0	0	

## ノイズキャンセル制御レジスタ

bit	15	14	13	12	11	10	9	8	
	S21	S20	S11	S10	S01	S00	D1	D0	NCCR
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

## タイマ状態制御レジスタ

bit	7	6	5	4	3	2	1	0	
	TCLR	MODE	ICLR	ICRE	TMEN	CLK2	CLK1	CLK0	TCSR
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値 ⇨	0	0	0	0	0	0	0	0	

## 15.4.1 出力制御レジスタ上位, 下位 (OPCUR, OPCLR)

出力制御レジスタ (OPCUR, OPCLR) は, 書込みタイミング割込みフラグを有効にし, 位置検出割込みフラグを有効にし, データ転送方式を設定し, OPT5 ~ OPT0 端子と DTTI1 端子の制御を設定します。

### ■ 出力制御レジスタ, 上位 (OPCUR)

図 15.4-2 出力制御レジスタ, 上位 (OPCUR)

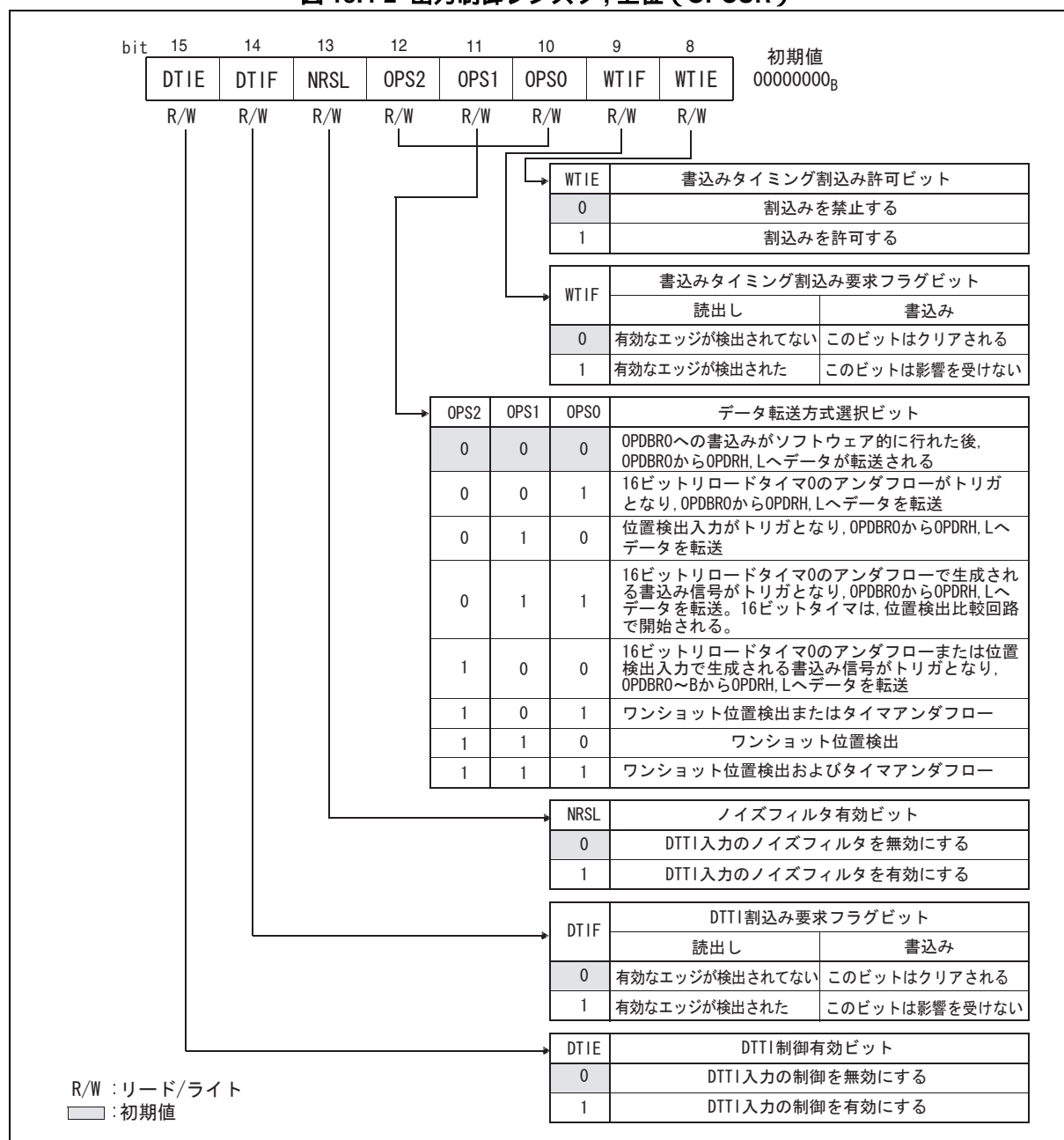


表 15.4-1 出力制御レジスタ, 上位 (OPCUR)

ビット名		機能
bit15	DTIE: DTTI 制御有効 ビット	<ul style="list-style-type: none"> <li>このビットは, DTTI 端子による OPT5 ~ OPT0 端子の出力レベル制御を可能にします。ソフトウェアは, PORTx の PDRx の各 OPTx 端子に対してインアクティブレベルを設定できます。</li> </ul>
bit14	DTIF: DTTI 割込み要 求フラグビッ ト	<ul style="list-style-type: none"> <li>このビットは, DTTI 入力割込み要求フラグです。このビットは, DTTI の立下がりエッジが検出され, DTTI 制御有効ビット (DTIE) に "1" が設定されていると必ず設定されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: 割込みが生成されます。このビットは影響を受けません。</li> <li>リードモディファイライト動作時は必ず "1" が読み出されます。</li> </ul>
bit13	NRSL: ノイズフィル タ有効ビット	<ul style="list-style-type: none"> <li>このビットは, DTTI 端子入力があるときにノイズ除去機能を選択します。</li> <li>ノイズ除去回路は, アクティブレベルが入力されたとき内部 n- ビットカウンタを開始します ("n" の値は, ノイズキャンセルレジスタ (NCCR) の D1, D0:bit10, bit9 の設定に基づいて 2, 3, 4 または 5 をとることができます)。カウンタでオーバフローが発生するまでアクティブレベルが保持されると, 回路は DTTI 端子からの入力を受け付けます。したがって, 除去可能なノイズのパルス幅は約 2n マシンサイクルです。</li> <li>ノイズキャンセル回路が有効の場合でも, 入力は内部クロックが停止しているモード (ストップモードなど) 時は無効になります。</li> </ul>
bit12 ~ bit10	OPS2 ~ OPS0: データ転送方 式選択ビット	<ul style="list-style-type: none"> <li>OPTx 端子出力タイミング制御選択ビット。</li> <li>これらのビットは, OPDRH, OPDRL レジスタ書込みタイミング制御動作モードを選択します。データは, 選択された動作モード時に制御される書込みタイミングで出力データバッファレジスタから出力データレジスタへ転送されます。</li> </ul>
bit9	WTIF: 書込みタイミ ング割込み要 求フラグビッ ト	<ul style="list-style-type: none"> <li>このビットは, 書込み信号により設定される出力タイミングスイッチ割込み要求フラグです。出力データレジスタ上位 (OPDRH) の BNKF:bit15 と RDA2 ~ RDA0:bit14 ~ bit12 で指定される OPDBRx レジスタのデータは, 書込み信号の立上りエッジで OPDRH, OPDRL へ転送されます。このとき, 本ビットは "1" に設定されます。</li> <li>このビットに "1" が設定され, かつ書込みタイミング割込み許可ビット (WTIE) にも "1" が設定していると, 割込みが生成されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト動作時は, 必ず "1" が読み出されます。</li> </ul>
bit8	WTIE: 書込みタイミ ング割込み許 可ビット	<ul style="list-style-type: none"> <li>このビットに "1" を設定し, かつ書込みタイミング割込み要求フラグビット (WTIF) に "1" が設定されると, 割込みが生成されます。</li> </ul>



■ 出力制御レジスタ, 下位 (OPCLR)

図 15.4-3 出力制御レジスタ, 下位 (OPCLR)

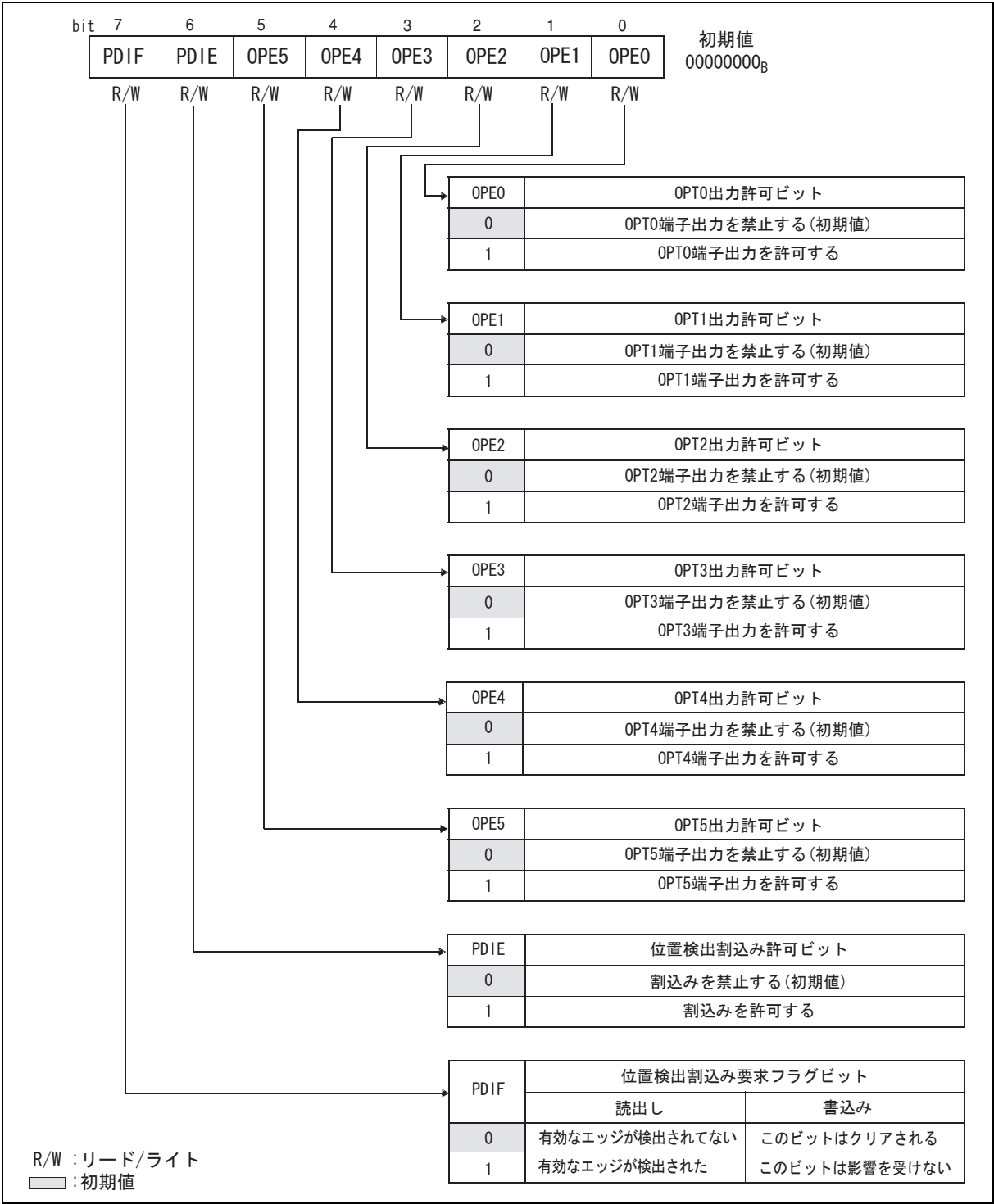


表 15.4-2 出力制御レジスタ, 下位 (OPCLR)

ビット名		機能
bit7	PDIF: 位置検出割込み要求フラグビット	<ul style="list-style-type: none"> <li>このビットは、入力制御レジスタ上位 (IPCUR) の CMPE:bit8 に "0" を設定し、SNI2 ~ SNI0 端子が出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 と比較され一致するかまたは CMPE ビットに "1" を設定していて、有効なエッジが SNIx 端子で検出されると、"1" が設定されます。</li> <li>このビットに "1" が設定され、かつ位置検出割込み許可ビット (PDIE) に "1" を設定していると、割込みが生成されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト動作時は、必ず "1" が読み出されます。</li> </ul>
bit6	PDIE: 位置検出割込み許可ビット	<ul style="list-style-type: none"> <li>このビットに "1" を設定し、かつ位置検出割込み要求フラグビット (PDIF) に "1" が設定されると、割込みが生成されます。</li> </ul>
bit5 ~ bit0	OPE5 ~ OPE0: OPT5 ~ OPE0 出力許可ビット	<ul style="list-style-type: none"> <li>これらのビットが設定されると、OPT5 ~ OPT0 端子への出力は許可になります。</li> </ul>

## 15.4.2 出力データレジスタ上位, 下位 (OPDRH, OPDRL)

出力データレジスタは, OPT5 ~ OPT0 端子への出力データを格納します。

### ■ 出力データレジスタ, 上位 (OPDRH)

図 15.4-4 出力データレジスタ, 上位 (OPDRH)

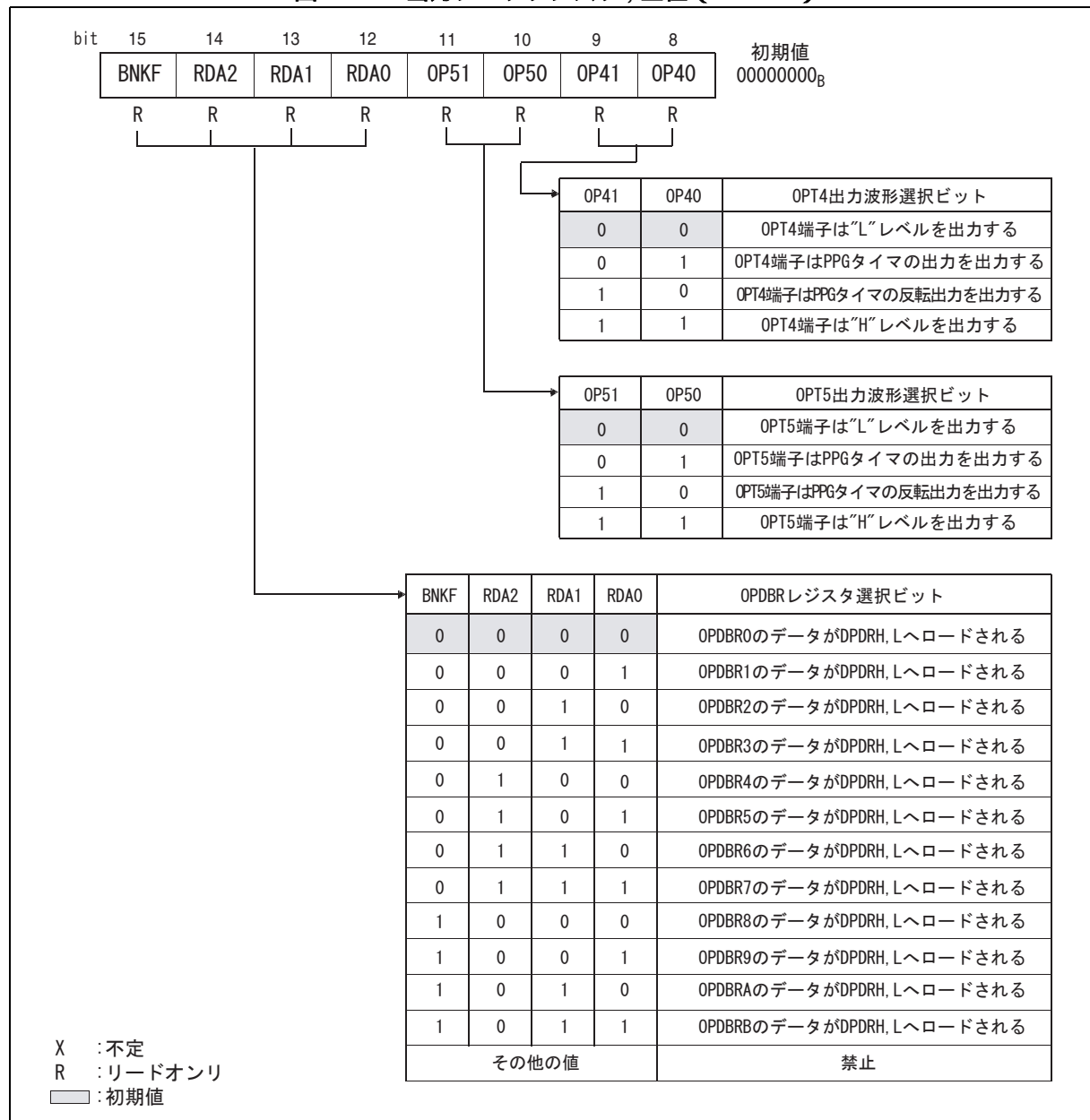


表 15.4-3 出力データレジスタ, 上位 (OPDRH)

ビット名		機能
bit15 ~ bit12	BNKF, RDA2 ~ RDA0: OPDBR0 ~ OPDBRB レジ スタ選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPDBR0 ~ OPDBRB レジスタのアドレスを示します。また OPDRH レジスタにロードされる出力データバッファレジスタ値を決定します。</li> </ul>
bit11, bit10	OP51, OP50: OPT5 出力波形 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPT5 端子に出力される出力波形の種類を選択します。</li> </ul>
bit9, bit8	OP41, OP40: OPT4 出力波形 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPT4 端子に出力される出力波形の種類を選択します。</li> </ul>

■ 出力データレジスタ , 下位 ( OPDRL )

図 15.4-5 出力データレジスタ , 下位 ( OPDRL )

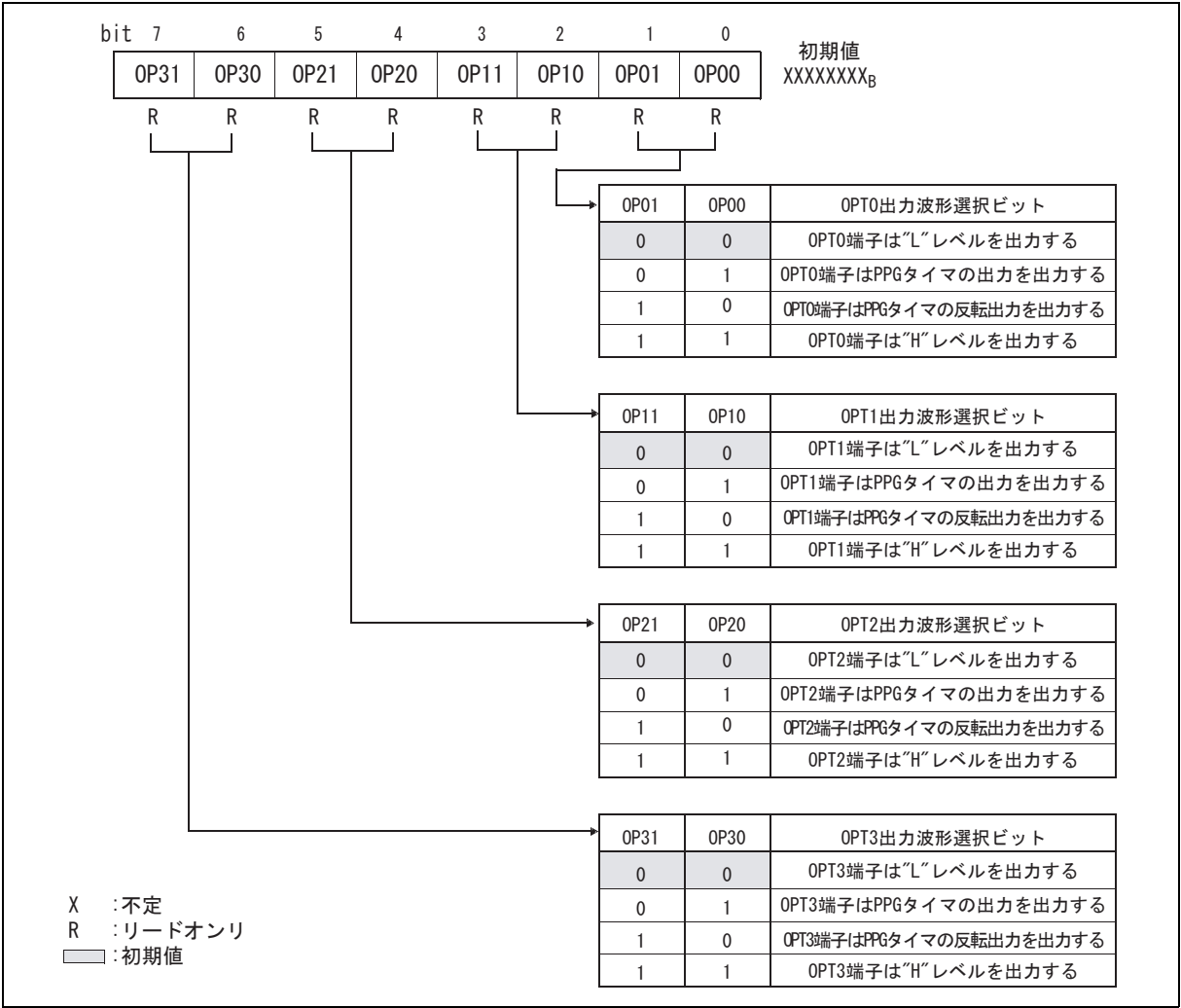


表 15.4-4 出力データレジスタ , 下位 ( OPDRL )

ビット名		機能
bit7, bit6	OP31, OP30: OPT3 出力波 形選択ビット	• これらのビットは , OPT3 端子に出力される出力波形の種類を選択します。
bit5, bit4	OP21, OP20: OPT2 出力波 形選択ビット	• これらのビットは , OPT2 端子に出力される出力波形の種類を選択します。
bit3, bit2	OP11, OP10: OPT1 出力波 形選択ビット	• これらのビットは , OPT1 端子に出力される出力波形の種類を選択します。
bit1, bit0	OP01, OP00: OPT0 出力波 形選択ビット	• これらのビットは , OPT0 端子に出力される出力波形の種類を選択します。

### 15.4.3 出力データバッファレジスタ (OPDBR0 ~ OPDBRB)

出力データバッファレジスタは、12 個のレジスタ (OPDBR0 ~ OPDBRB) から構成されています。本レジスタ上位の BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 で指定される OPDBRx レジスタ値は、データ書込み制御ユニットで生成される書込み信号の立上りエッジで OPDRH, OPDRL レジスタへロードされます。

#### ■ 出力データバッファレジスタ, 上位 (OPDBR0 ~ OPDBRB)

図 15.4-6 出力データバッファレジスタ, 上位 (OPDBR0 ~ OPDBRB)

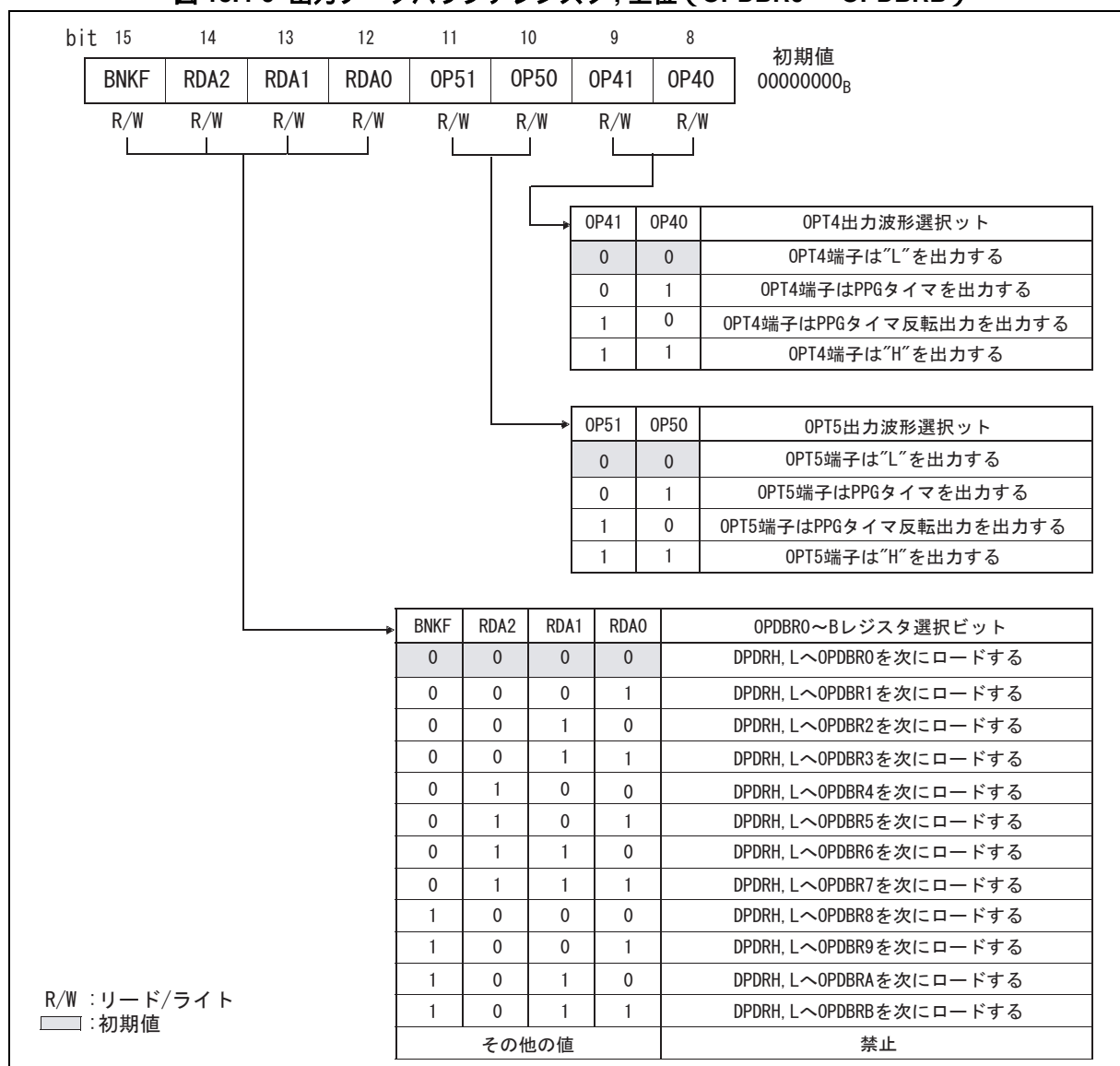


表 15.4-5 出力データバッファレジスタ, 上位 (OPDBR0 ~ OPDBRB)

ビット名		機能
bit15 ~ bit12	BNKF, RDA2 ~ RDA0: OPDBR0 ~ OPDBRB レジ スタ選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPDBR0 ~ OPDBRB レジスタのアドレスを示します。また OPDRH レジスタへロードされる出力データバッファレジスタ値を決定します。</li> </ul>
bit11, bit10	OP51, OP50: OPT5 出力波形 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPDRH レジスタへロードされた後に OPT5 端子へ出力される出力波形の種類を選択します。</li> </ul>
bit9, bit8	OP41, OP40: OPT4 出力波形 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは, OPDRH レジスタへロードされた後に OPT4 端子へ出力される出力波形の種類を選択します。</li> </ul>

## ■ 出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBRB)

図 15.4-7 出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBRB)

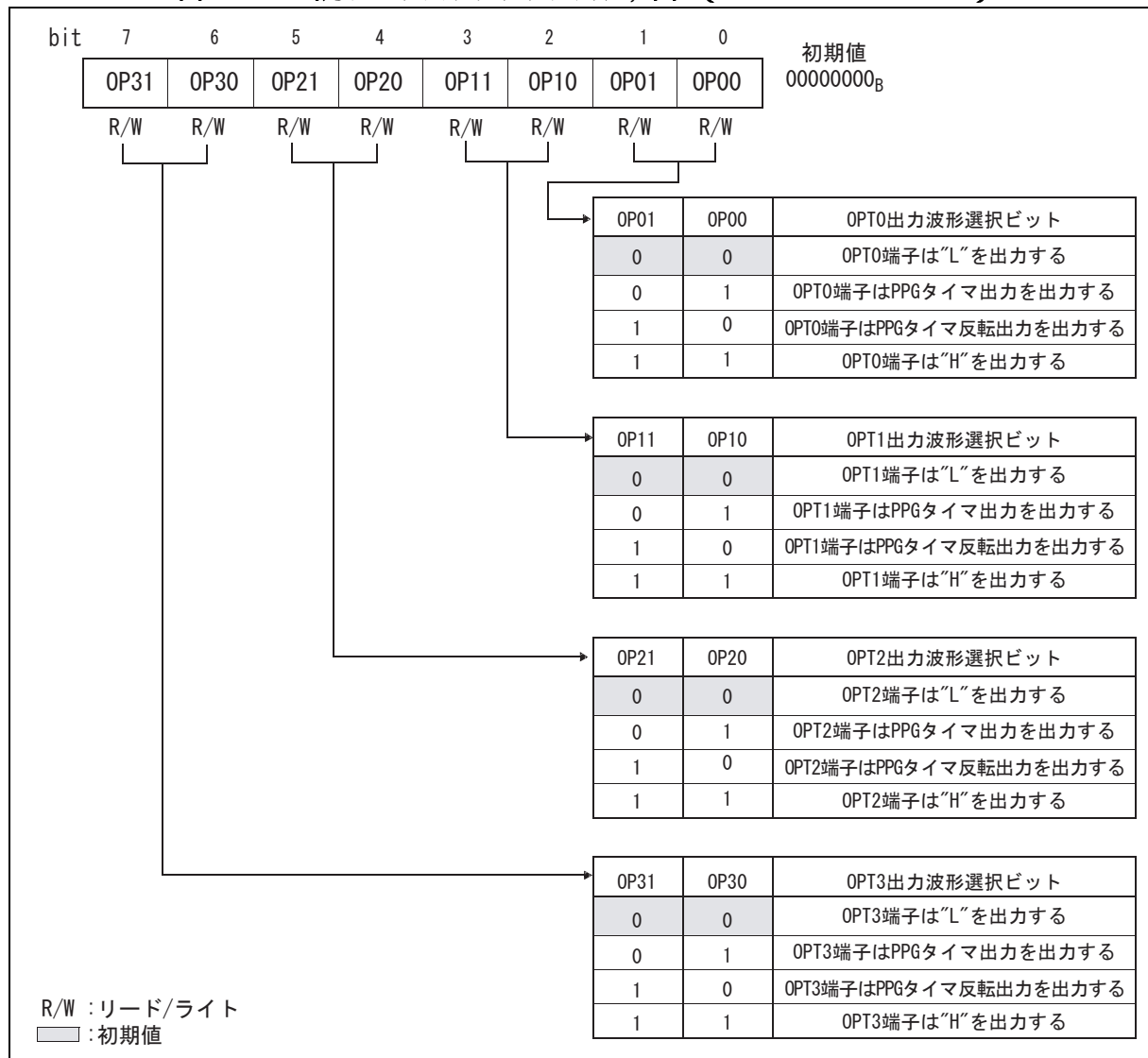


表 15.4-6 出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBRB)

ビット名		機能
bit7, bit6	OP31, OP30: OPT3 出力波 形選択ビット	• これらのビットは, OPDRL レジスタへロードされた後に OPT3 端子へ出力される出力波形の種類を選択します。
bit5, bit4	OP21, OP20: OPT2 出力波 形選択ビット	• これらのビットは, OPDRL レジスタへロードされた後に OPT2 端子へ出力される出力波形の種類を選択します。
bit3, bit2	OP11, OP10: OPT1 出力波 形選択ビット	• これらのビットは, OPDRL レジスタへロードされた後に OPT1 端子へ出力される出力波形の種類を選択します。
bit1, bit0	OP01, OP00: OPT0 出力波 形選択ビット	• これらのビットは, OPDRL レジスタへロードされた後に OPT0 端子へ出力される出力波形の種類を選択します。



## 15.4.4 入力制御レジスタ上位，下位（IPCUR, IPCLR）

入力制御レジスタ上位，下位（IPCUR, IPCLR）は，位置検出入力を制御するレジスタです。

### ■ 入力制御レジスタ，上位（IPCUR）

図 15.4-8 入力制御レジスタ，上位（IPCUR）

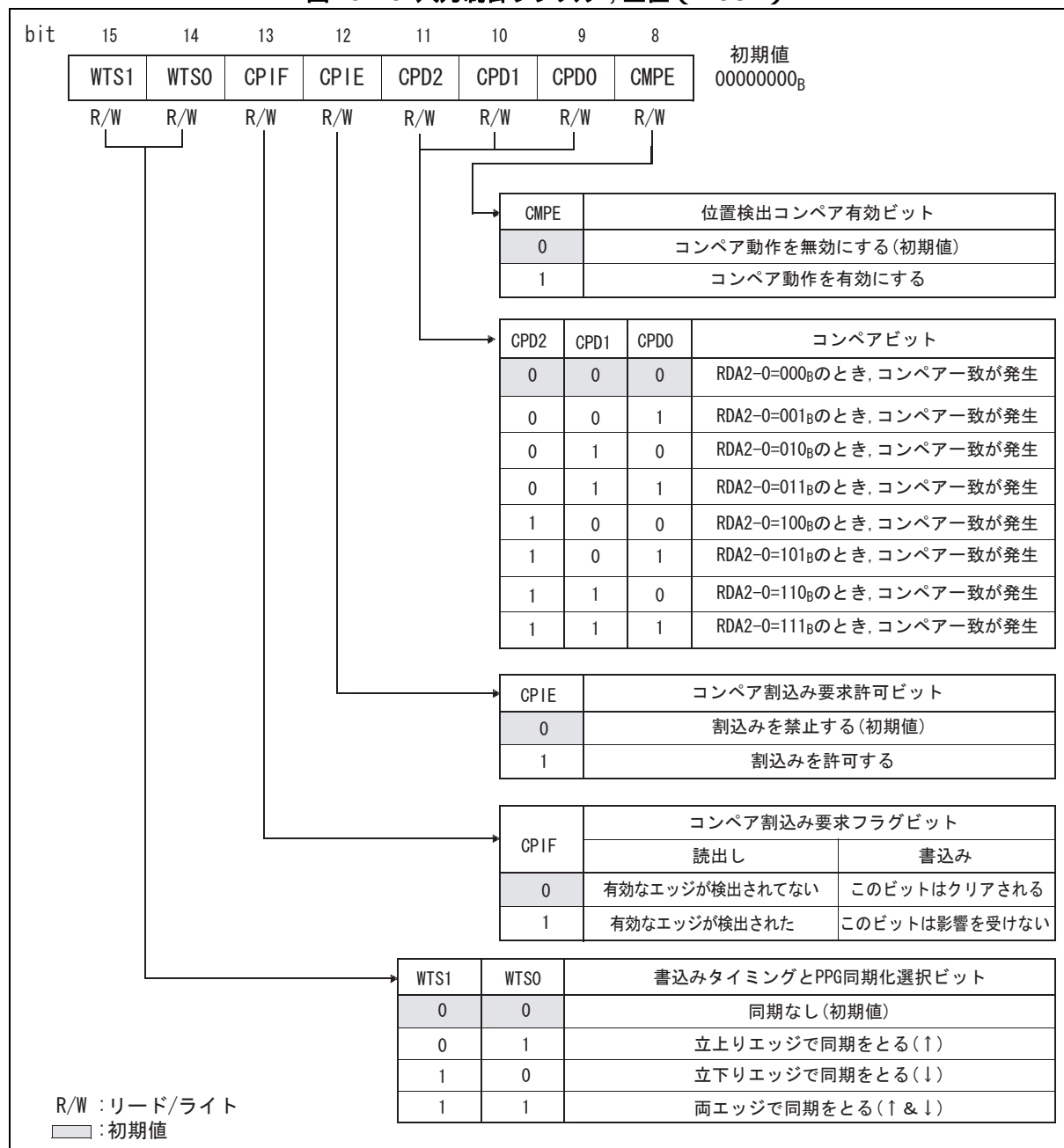


表 15.4-7 入力制御レジスタ，上位（IPCUR）

ビット名		機能
bit15, bit14	WTS1, WTS0: 書込みタイミングと PPG エッジ同期化 選択ビット	<ul style="list-style-type: none"> <li>これらのビットは，書込みタイミングと同期化させるための " 次の PPG 信号エッジ " を選択します。</li> </ul>
bit13	CPIF: コンペア割込み 要求フラグ ビット	<ul style="list-style-type: none"> <li>SNI2 ~ SNI0 端子が CPD2 ~ CPD0 と比較され一致すると，このビットに "1" が設定されます。</li> <li>コンペア割込み要求許可ビット（CPIE）に "1" を設定していると，割込みが生成されます。</li> <li>このビットに "0" を設定した場合：このビットはクリアされます。</li> <li>このビットに "1" を設定した場合：このビットは影響を受けません。</li> <li>リードモディファイライト動作時は，必ず "1" が読み出されます。</li> </ul>
bit12	CPIE: コンペア割込み 要求許可 ビット	<ul style="list-style-type: none"> <li>このビットに "1" を設定し，かつコンペア割込み要求フラグ（CPIF）に "1" が設定されると，割込みが生成されます。</li> </ul>
bit11 ~ bit9	CPD2 ~ CPD0: コンペアビッ ト	<ul style="list-style-type: none"> <li>これらのビットは，出力データレジスタ上位（OPDRH）の RDA2 ~ RDA0:bit14 ~ bit12 と比較されます。これらのビットの値が RDA2 ~ RDA0 ビットの値と一致すると，コンペア割込み要求フラグビット（CPIF）に "1" が設定されます。</li> </ul>
bit8	CMPE: 位置検出コン ペア有効ビッ ト	<ul style="list-style-type: none"> <li>このビットは，位置検出の比較動作を有効にします。</li> </ul>

■ 入力制御レジスタ, 下位 (IPCLR)

図 15.4-9 入力制御レジスタ, 下位 (IPCLR)

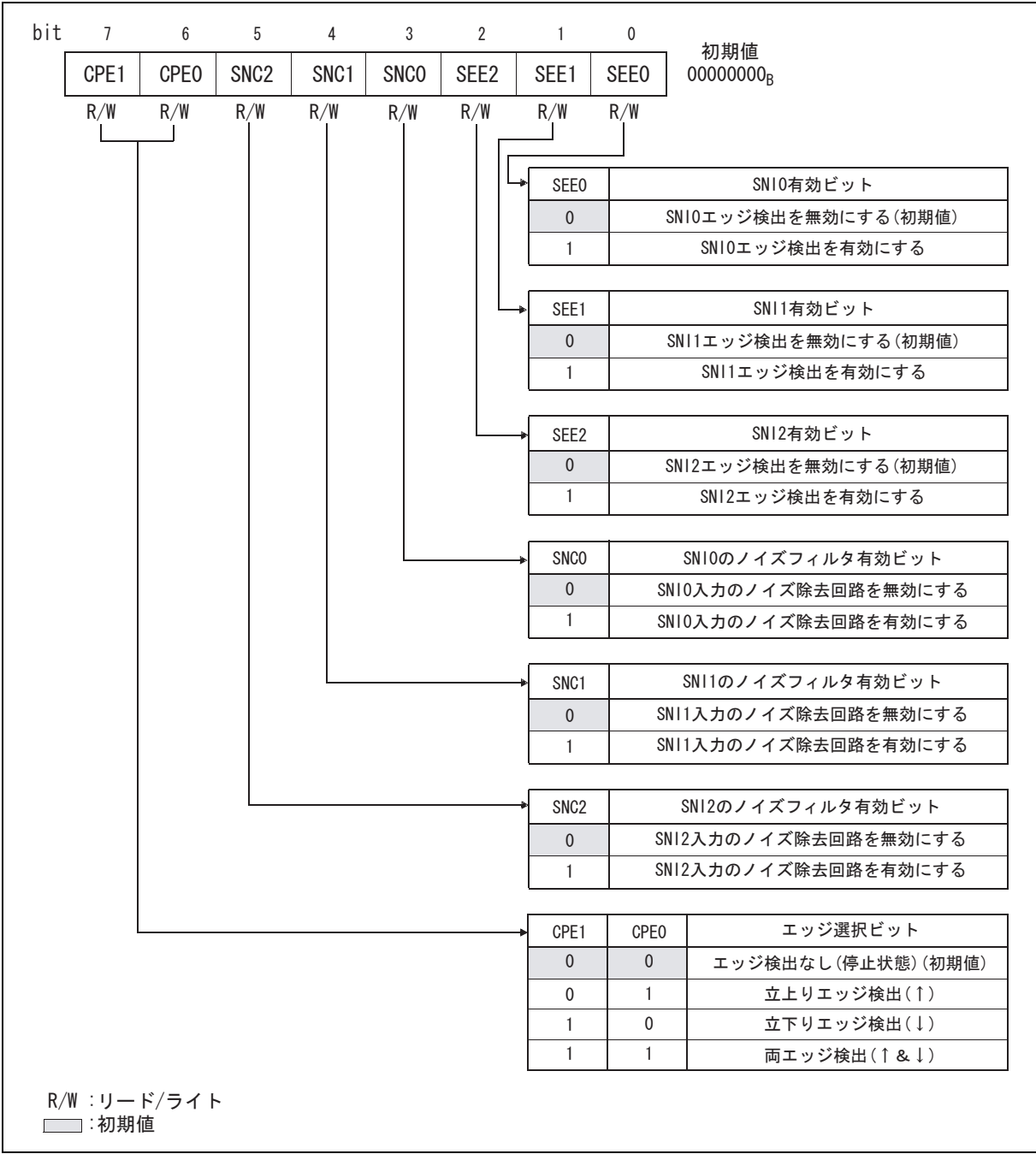


表 15.4-8 入力制御レジスタ, 下位 (IPCLR)

ビット名		機能
bit7, bit6	CPE1, CPE0: エッジ選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, 位置検出の入力エッジ極性を選択します。位置検出は, これらのビットに設定された入力エッジ極性に従って動作します。</li> </ul>
bit5 ~ bit3	SNC2 ~ SNC0: SNI2 ~ SNI0 のノイズフィ ルタ有効ピッ ト	<ul style="list-style-type: none"> <li>これらのビットは, SNI2 ~ SNI0 端子入力があるときにノイズ除去機能を選択します。</li> <li>ノイズ除去回路は, アクティブレベルが入力されたとき内部 n- ビットカウントを開始します ("n" の値はノイズキャンセルレジスタ (NCCR) の S21/S20, S11/S10, S01/S00 ビットの設定に基づいて 2, 3, 4 または 5 をとることができます)。カウンタでオーバフローが発生するまでアクティブレベルが保持されると, 回路は SNI2 ~ SNI0 端子からの入力を受付けます。したがって, 除去可能なノイズのパルス幅は約 2n マシンサイクルです。</li> </ul> <p>(注意事項)</p> <p>ノイズ除去回路が有効の場合でも, 入力は内部クロックが停止しているモード (ストップモードなど) 時は無効です。</p>
bit2 ~ bit0	SEE2 ~ SEE0: SNI2 ~ SNI0 の有効ビット	<ul style="list-style-type: none"> <li><b>これらのビットに "1" を設定した場合:</b> SNI2 ~ SNI0 端子のエッジ検出は有効になります。</li> <li>これらのビットは, 入力制御レジスタ上位 (IPCUR) の CMPE:bit8 に "1" を設定する前に設定してください。</li> </ul>

### 15.4.5 コンペアクリアレジスタ上位, 下位 (CPCR<sub>H</sub>, CPCR<sub>L</sub>)

コンペアクリアレジスタ上位, 下位 (CPCR<sub>H</sub>, CPCR<sub>L</sub>) は, 16 ビットのレジスタです。このレジスタが 16 ビットタイマのカウント値と一致すると, 16 ビットタイマは "0000<sub>H</sub>" にリセットされます。

#### ■ コンペアクリアレジスタ上位, 下位 (CPCR<sub>H</sub>, CPCR<sub>L</sub>)

コンペアクリアレジスタは 16 ビットのレジスタです。16 ビットタイマのカウント値と比較されます。このレジスタの初期値は不定なので, 動作を開始する前にこのレジスタに値を設定しなければなりません。

< 注意事項 > このレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。

このレジスタが 16 ビットタイマのカウント値と一致すると, 16 ビットタイマは "0000<sub>H</sub>" にリセットされ, コンペアクリア割込み要求フラグ (ICLR) が設定されます。また, 割込み動作が許可されているときは, 割込み要求が生成されます。

< 注意事項 > コンペアクリアレジスタ上位, 下位 (CPCR<sub>H</sub>, CPCR<sub>L</sub>) にタイマカウンタ値と同じ値がロードされると, 比較動作は次回この状態が発生するまで行われません。

図 15.4-10 コンペアクリアレジスタ上位, 下位 (CPCR<sub>H</sub>, CPCR<sub>L</sub>)

コンペアクリアレジスタ (上位)										
	bit	15	14	13	12	11	10	9	8	
		CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	CPCR <sub>H</sub>
リード/ライト	⇒	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	

コンペアクリアレジスタ (下位)										
	bit	7	6	5	4	3	2	1	0	
		CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	CPCR <sub>L</sub>
リード/ライト	⇒	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	

### 15.4.6 タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL)

タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL) は, 16 ビットタイマのカウント値を読み出すために使用します。

#### ■ タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL)

タイマバッファレジスタは, 書込みタイミングまたは位置検出トリガが生成された時点の 16 ビットタイマのカウント値を格納します。この後, カウンタは "0000<sub>H</sub>" にクリアされます。

< 注意事項 > タイマバッファレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。

図 15.4-11 タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL)

タイマバッファレジスタ (上位)								
	bit 15	14	13	12	11	10	9	8
	T15	T14	T13	T12	T11	T10	T09	T08
リード/ライト ⇨	R	R	R	R	R	R	R	R
初期値 ⇨	0	0	0	0	0	0	0	0
タイマバッファレジスタ (下位)								
	bit 7	6	5	4	3	2	1	0
	T07	T06	T05	T04	T03	T02	T01	T00
リード/ライト ⇨	R	R	R	R	R	R	R	R
初期値 ⇨	0	0	0	0	0	0	0	0

## 15.4.7 ノイズキャンセル制御レジスタ (NCCR)

ノイズキャンセル制御レジスタ (NCCR) は, DTTI 端子および SNIx 端子の除去するノイズパルス幅を制御するために使用します。

### ■ ノイズキャンセル制御レジスタ (NCCR)

図 15.4-12 ノイズキャンセル制御レジスタ (NCCR)

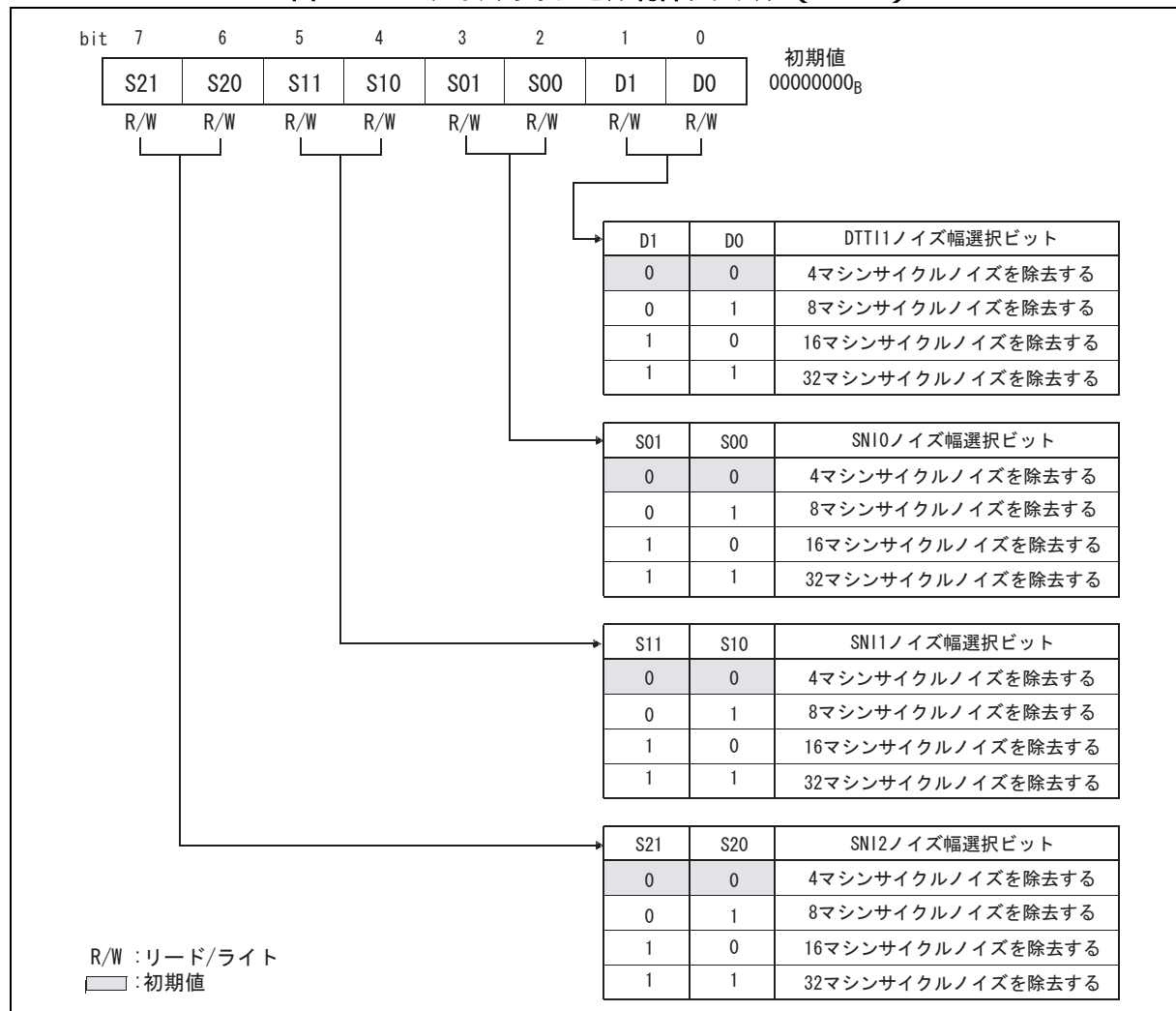


表 15.4-9 ノイズキャンセル制御レジスタ (NCCR)

ビット名		機 能
bit7, bit6	S21, S20: ノイズ幅選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, SNI2 端子の除去対象ノイズパルス幅を指定します。</li> </ul>
bit5, bit4	S11, S10: ノイズ幅選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, SNI1 端子の除去対象ノイズパルス幅を指定します。</li> </ul>
bit3, bit2	S01, S00: ノイズ幅選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, SNI0 端子の除去対象ノイズパルス幅を指定します。</li> </ul>
bit1, bit0	D1, D0: ノイズ幅選択 ビット	<ul style="list-style-type: none"> <li>これらのビットは, DTTI1 端子の除去対象ノイズパルス幅を指定します。</li> </ul>



## 15.4.8 タイマ状態制御レジスタ (TCSR)

タイマ状態制御レジスタ (TCSR) は, 16 ビットタイマの動作を制御するために使用します。

### ■ タイマ状態制御レジスタ (TCSR)

図 15.4-13 タイマ状態制御レジスタ (TCSR)

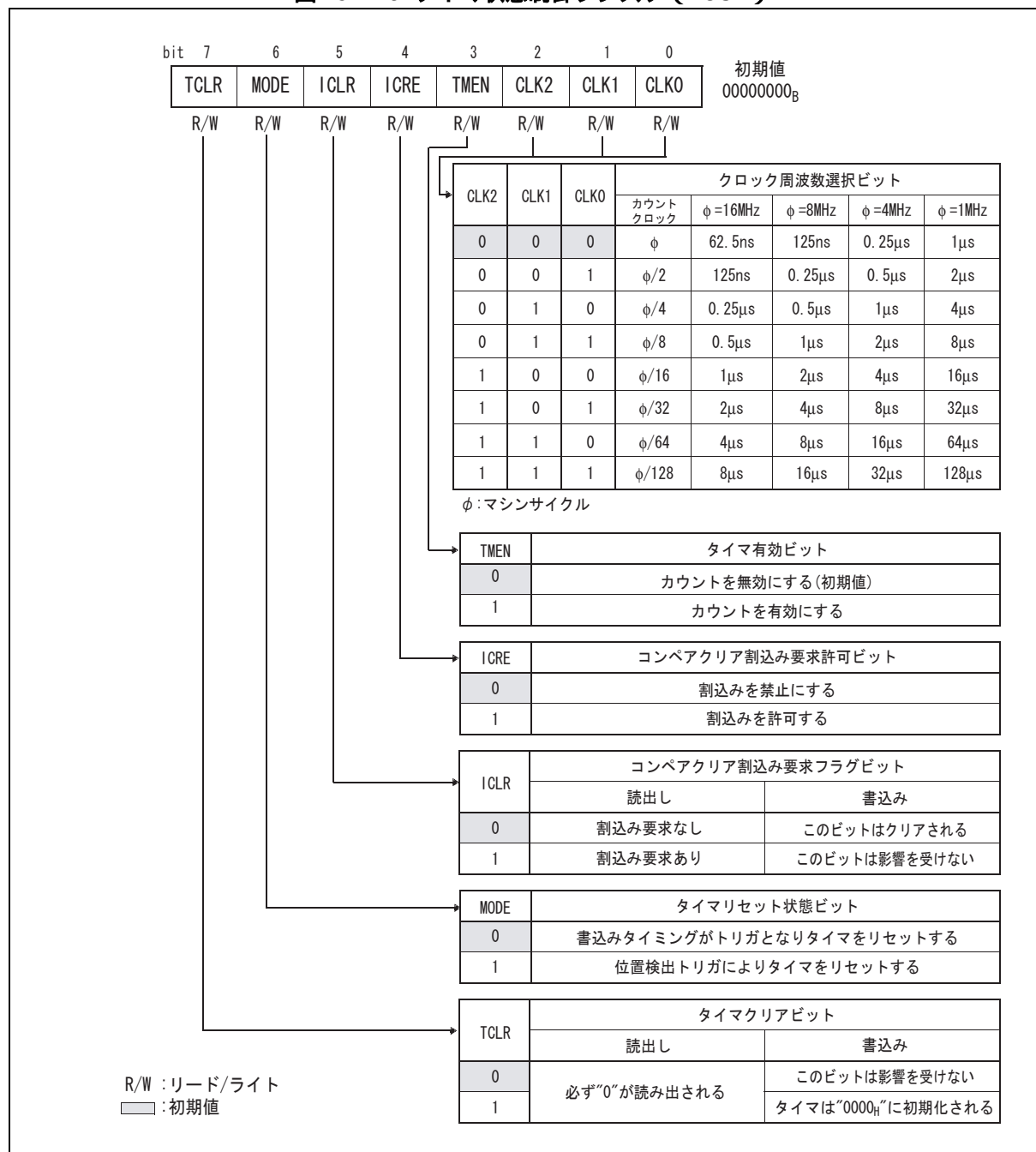


表 15.4-10 タイマ状態制御レジスタ (TCSR)

ビット名		機 能
bit7	TCLR: タイマクリア ビット	<ul style="list-style-type: none"> <li>読出し値は必ず "0" です。</li> <li>このビットに "0" を設定した場合: このビットは影響を受けません。</li> <li>このビットに "1" を設定した場合: タイマは "0000<sub>H</sub>" へ初期化されます。</li> </ul>
bit6	MODE: タイマリセッ ト状態ビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットタイマのリセット状態を設定します。</li> <li>このビットに "0" を設定した場合: 16 ビットタイマは書込みタイミング信号でリセットされます。</li> <li>このビットに "1" を設定した場合: 16 ビットタイマは位置検出信号でリセットされます。</li> </ul> <p>(注意事項) タイマ値のリセットは、タイマ値が変更された時点で行われます。</p>
bit5	ICLR: コンペアクリ ア割込み要求 フラグビット	<ul style="list-style-type: none"> <li>コンペアクリアレジスタと 16 ビットフリーランタイム値が一致すると、タイマがクリアされ、このビットには "1" が設定されます。</li> <li>コンペアクリア割込み要求許可ビット (ICRE) に "1" が設定されると、割込みが生成されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト動作時は、必ず "1" が読み出されます。</li> </ul>
bit4	ICRE: コンペアクリ ア割込み要求 許可ビット	<ul style="list-style-type: none"> <li>このビットに "1" を設定した場合: コンペアクリア割込み要求フラグビット (ICLR) に "1" が設定されると、割込みが生成されます。</li> </ul>
bit3	TMEN: タイマ有効 ビット	<ul style="list-style-type: none"> <li>このビットは、16 ビットタイマのカウントを有効または無効にします。</li> <li>このビットに "0" を設定した場合: 16 ビットタイマのカウントは無効になります。</li> <li>このビットに "1" を設定した場合: 16 ビットタイマのカウントは有効になります。</li> </ul> <p>(注意事項) 16 ビットタイマが無効の場合は、出力コンペア動作も無効になります。</p>
bit2 ~ bit0	CLK2 ~ CLK0: クロック周波 数選択ビット	<ul style="list-style-type: none"> <li>これらのビットは、16 ビットフリーランタイムのカウントクロックを選択します。</li> </ul> <p>(注意事項) これらのビットが更新されるとクロックが直ちに变更されるので、これらのビットはタイマ停止状態の間に変更することを推奨します。</p>

## 15.5 マルチパルスジェネレータの割込み

---

マルチパルスジェネレータは、以下の場合に割込み要求を生成できます。

- 書込みタイミング出力がデータ書込み制御ユニットで生成されます。
  - 有効な位置検出入力が検出されます。
  - 入力制御レジスタ上位 (IPCUR) の CPD2 ~ CPD0:bit11 ~ bit9 と出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 の間でコンペア一致が検出されます。
  - 16 ビットタイマでコンペアクリアが生成されます。
  - DTTI が信号レベル low に変化します。
- 

### ■ マルチパルス割込み

マルチパルスジェネレータは、以下の 5 つの割込み要因を持っています。

- 書込みタイミング割込み
- コンペアクリア割込み
- 位置検出割込み
- コンペア一致割込み
- DTTI 割込み

書込みタイミング割込みはコンペアクリア割込みと共有し、位置検出割込みはコンペア一致割込みと共有しています。

#### ● 書込みタイミング割込み

出力制御レジスタ上位 (OPCUR) の WTIE:bit8 に "1" を設定し、かつ書込みタイミングがデータ書込み制御回路で出力されると、この書込みタイミング割込みが生成され、12 個の出力データバッファレジスタ (OPDBR0 ~ OPDBRB) のいずれかから出力データレジスタ (OPDRH, OPDRL) へデータが転送されます。

この割込みが生成されると、出力制御レジスタ上位 (OPCUR) の書込みタイミング割込み要求フラグビット (WTIF:bit9) に "1" が設定されます。

#### ● コンペアクリア割込み

タイマ状態制御レジスタ (TCSR) の ICRE:bit4 に "1" を設定し、かつ 16 ビットタイマでアンダフローが発生すると、コンペアクリア割込みが生成されます。

この割込みが生成されると、タイマ状態制御レジスタ (TCSR) のコンペアクリア割込み要求フラグビット (ICLR:bit5) に "1" が設定されます。

### ● 位置検出タイミング割込み

出力制御レジスタ下位 (OPCLR) の PDIE:bit6 に "1" を設定し、かつ書込みタイミングが位置検出回路で出力されると、この位置検出割込みが生成され、12 個の出力データバッファレジスタ(OPDBR0 ~ OPDBRB)のいずれかから出力データレジスタ(OPDRH, OPDRL)へデータが転送されます。この書込みタイミングは、位置入力(SNI2 ~ SNI0)のレベルと出力データレジスタ上位(OPDRH)の RDA2 ~ RDA0:bit14 ~ bit12 の間でコンペアー一致が検出された場合または 3 つの異なるエッジ設定のいずれかを持っている位置入力(SNI2 ~ SNI0)でエッジが検出された場合、出力できます。

この割込みが生成されると、出力制御レジスタ下位(OPCLR)の位置検出割込み要求フラグビット(PDIF:bit7)に "1" が設定されます。

### ● コンペアー一致割込み

入力制御レジスタ上位(IPCUR)の CPIE:bit12 に "1" を設定し、かつ出力データレジスタ上位(OPDRH)の RDA2 ~ RDA0:bit14 ~ bit12 と入力制御レジスタ上位(IPCUR)の CPD2 ~ CPD0:bit11 ~ bit9 の間で一致が検出されると、このコンペアー一致割込みが生成されます。

この割込みが生成されると、入力制御レジスタ上位(IPCUR)のコンペアー一致割込み要求フラグビット(CPIF:bit13)に "1" が設定されます。

### ● DTTI 割込み

出力制御レジスタ上位(OPCUR)の DTIE:bit15 に "1" を設定し、かつ DTTI1 端子で low 入力が検出されると、この DTTI 割込みが生成されます。

この割込みが生成されると、出力制御レジスタ上位(OPCUR)の DTTI 割込み要求フラグビット(DTIF:bit14)に "1" が設定されます。

## ■ マルチパルスジェネレータの割込み要因

割込み #22: この割込みは、DTTI 割込みが発生すると生成されます。

DTTI 割込みは、DTTI1 端子において low レベル入力が検出され、かつ出力制御レジスタ上位(OPCUR)の DTIE:bit15 に "1" を設定していると生成されます。

割込み #26: この割込みは、位置検出割込みまたはコンペアー一致割込みが発生すると生成されます。

書込みタイミング信号がデータ書込み制御回路から生成されたとき、出力制御レジスタ上位(OPCUR)の WTIE:bit8 に "1" を設定していれば、書込みタイミング割込みは生成されます。

割込み #28: この割込みは、位置検出割込みまたはコンペアー一致割込みが発生すると生成されます。

SNI2 ~ SNI0 における有効エッジが検出されたとき、出力制御レジスタ下位(OPCLR)の PDIE:bit6 に "1" を設定していれば、位置検出割込みは生成されます。

SNI2 ~ SNI0 入力が出力データレジスタ上位(OPDRH)の RDA2 ~ RDA0:bit14 ~ bit12 と一致したとき、入力制御レジスタ上位(IPCUR)の CPIE:bit12 に "1" を設定していれば、コンペアー一致割込みが生成されます。

## ■ マルチパルスジェネレータの割込みと EI<sup>2</sup>OS

表 15.5-1 マルチパルスジェネレータの割込みと EI<sup>2</sup>OS

割込み要因	割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	上位	バンク	
DTTI1	#22 (16 <sub>H</sub> )	ICR05	0000B5 <sub>H</sub>	FFFFAC <sub>H</sub>	FFFFAD <sub>H</sub>	FFFFAE <sub>H</sub>	×
書込みタイミングまたはコンペアクリア	#26 (1A <sub>H</sub> )	ICR07	0000B7 <sub>H</sub>	FFFF94 <sub>H</sub>	FFFF95 <sub>H</sub>	FFFF96 <sub>H</sub>	
位置検出またはコンペア一致	#28 (1C <sub>H</sub> )	ICR08	0000B8 <sub>H</sub>	FFFF8C <sub>H</sub>	FFFF8D <sub>H</sub>	FFFF8E <sub>H</sub>	

:EI<sup>2</sup>OS をサポートしています

## ■ マルチパルスジェネレータの EI<sup>2</sup>OS 機能

マルチパルスジェネレータは、書込みタイミング割込み、コンペアクリア割込み、位置検出割込みまたはコンペア一致割込みが生成されると起動可能な EI<sup>2</sup>OS 操作回路を持っています。

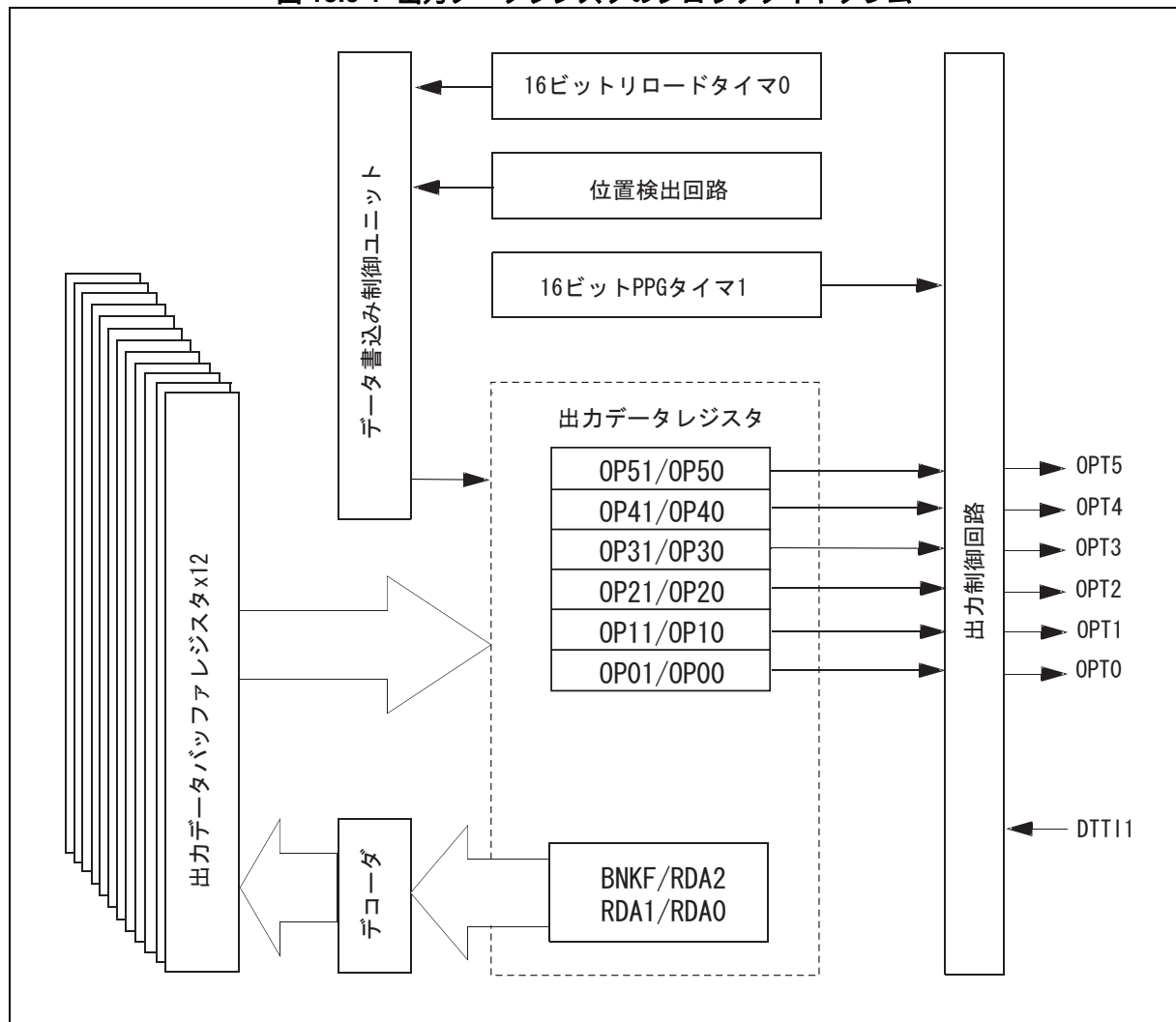
ただし、EI<sup>2</sup>OS は、割込み制御レジスタ (ICR) を共用するほかのリリースが割込みを使用しない場合に限り使用可能です。たとえば、EI<sup>2</sup>OS を起動するために書込みタイミング割込みまたはコンペアクリア割込みを使用している場合、DTP/ 外部割込みチャンネル 4, 5 の検出は無効にしなければなりません。

## 15.6 マルチパルスジェネレータの動作

マルチパルスジェネレータの動作について説明します。OPTx 端子は、出力データレジスタ上位、下位 (OPDRH, OPDRL) の OPx1, OPx0:bit11 ~ bit0 の設定に従って、対応する種類の波形 ("H", "L" または PPG 出力) を出力します。表 15.6-1 をご参照ください。

### ■ 出力データレジスタのブロックダイアグラム

図 15.6-1 出力データレジスタのブロックダイアグラム



## ■ 出力データレジスタ上位, 下位 (OPDRH, OPDRL)

出力データレジスタ (OPDRH, OPDRL) の値は, データ書込み制御ユニットで生成された書込みタイミング信号 (WTO) に従って出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から受取られます。また, OPTx 出力波形は更新されます。さらに, 出力レベルは DTTI 端子入力により強制的に固定できます。

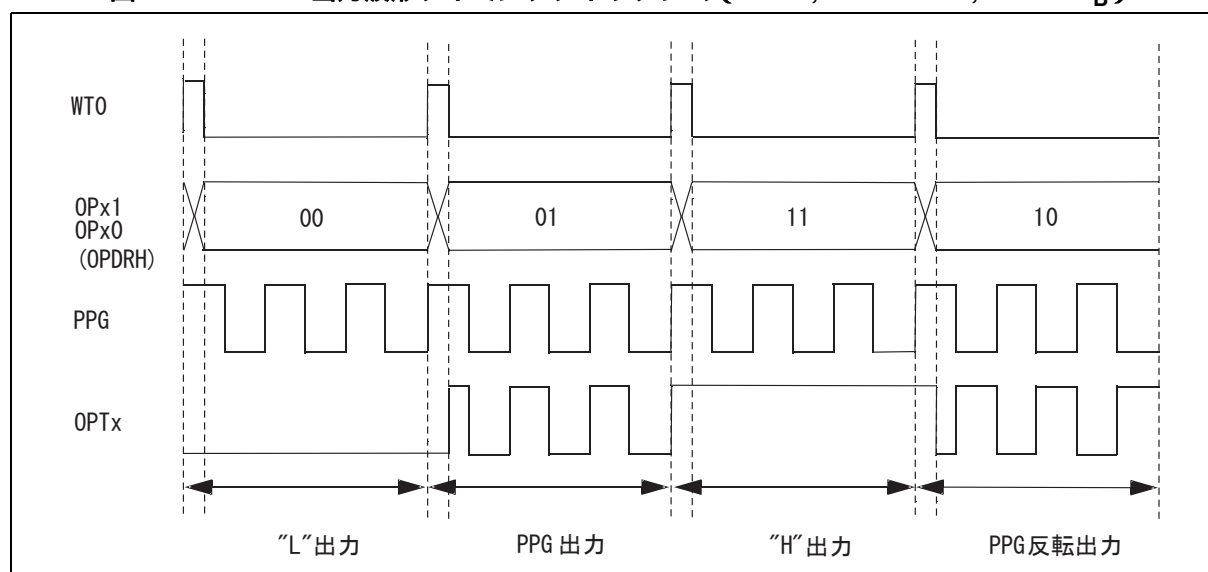
表 15.6-1 出力データレジスタ上位, 下位 (OPDRH, OPDRL)

OPx1/OPx0 設定	OPTx 出力
OPx1, OPx0 = 0, 0	low レベル
OPx1, OPx0 = 0, 1	16 ビット PPG タイマ出力
OPx1, OPx0 = 1, 0	16 ビット PPG タイマ反転出力
OPx1, OPx0 = 1, 1	high レベル

OPTx 出力波形タイミングダイアグラムを図 15.6-2 に示します。動作についての説明は次項をご参照ください。

### ● OPTx 出力波形タイミングダイアグラム (WTS1, WTS0:bit15, bit14=00<sub>B</sub>)

図 15.6-2 OPTx 出力波形タイミングダイアグラム (WTS1, WTS0:bit15, bit14=00<sub>B</sub>)



## 15.6.1 位置検出の動作

位置検出回路の動作について説明します。有効な位置が検出されており、かつ出力制御レジスタ下位 (OPCLR) の PDIE:bit7 に "1" を設定している場合、データ書込みタイミング出力 (WTIN1) はデータ書込み制御ユニットに対して生成され、位置検出割込みが生成されます。

### ■ 位置検出の動作

WTIN1 信号は、以下の条件が満たされると位置検出回路で生成されます。

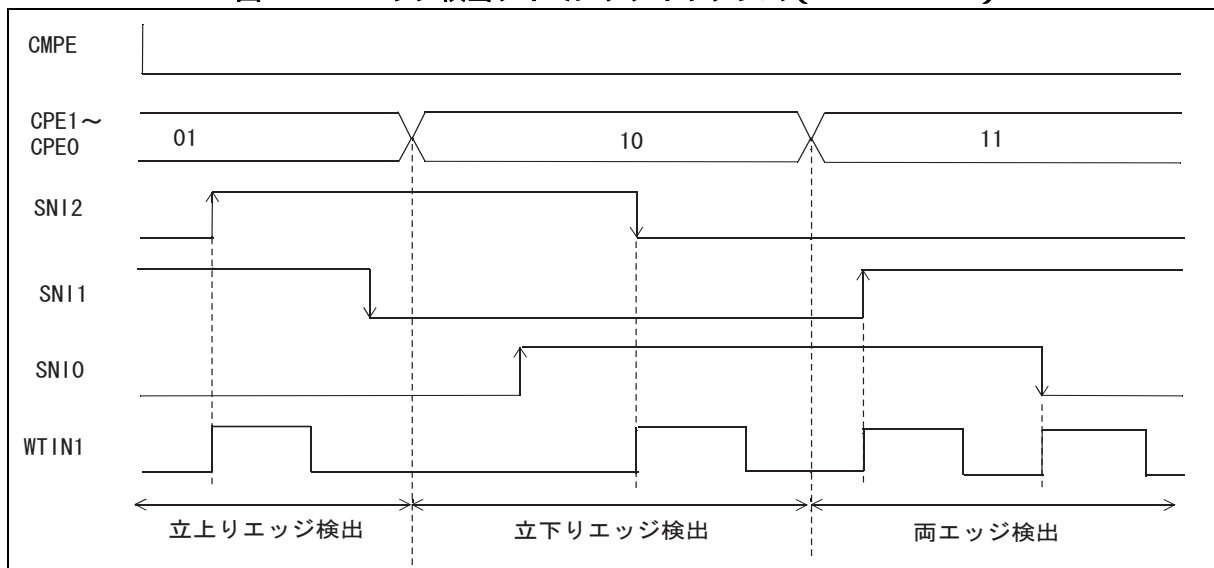
- SNI2 ~ SNI0 端子と出力データバッファレジスタ上位 (OPDBR0 ~ OPDBRB) の RDA2 ~ RDA0:bit14 ~ bit12 の間でコンペア一致が検出されたとき (このコンペア一致のトリガは、SNI2 ~ SNI0 端子の有効エッジ)。
- 対応する入力制御レジスタ下位 (IPCLR) の SEEx:bit2, bit1 または 0 により有効にされる SNIx 端子において有効エッジが検出されたとき。

入力制御レジスタ上位 (IPCUR) の CMPE:bit8 に "0" を設定すると、SEE2 ~ SEE0:bit2 ~ bit0 により有効にされた SNIx 端子のエッジ検出のみが位置検出のエッジ検出動作となります。たとえば、SEE0:bit0 のみに "1" が設定され、SNI0 端子への入力エッジが有効な場合、有効エッジが SNI0 端子で検出された場合のみデータ書込み出力信号が生成されます。CMPE:bit8=0 のときのエッジ検出タイミングダイアグラムを図 15.6-3 に示します。

入力制御レジスタ上位 (IPCUR) の CMPE:bit8 に "1" を設定すると、SNI2 ~ SNI0 端子は出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 と比較されます。この比較は、SNI2 ~ SNI0 端子のエッジ変化がトリガとなります。CMPE:bit8=1 のときのエッジ検出タイミングダイアグラムを図 15.6-4 に示します。

#### ● エッジ検出タイミングダイアグラム (CMPE:bit8=0)

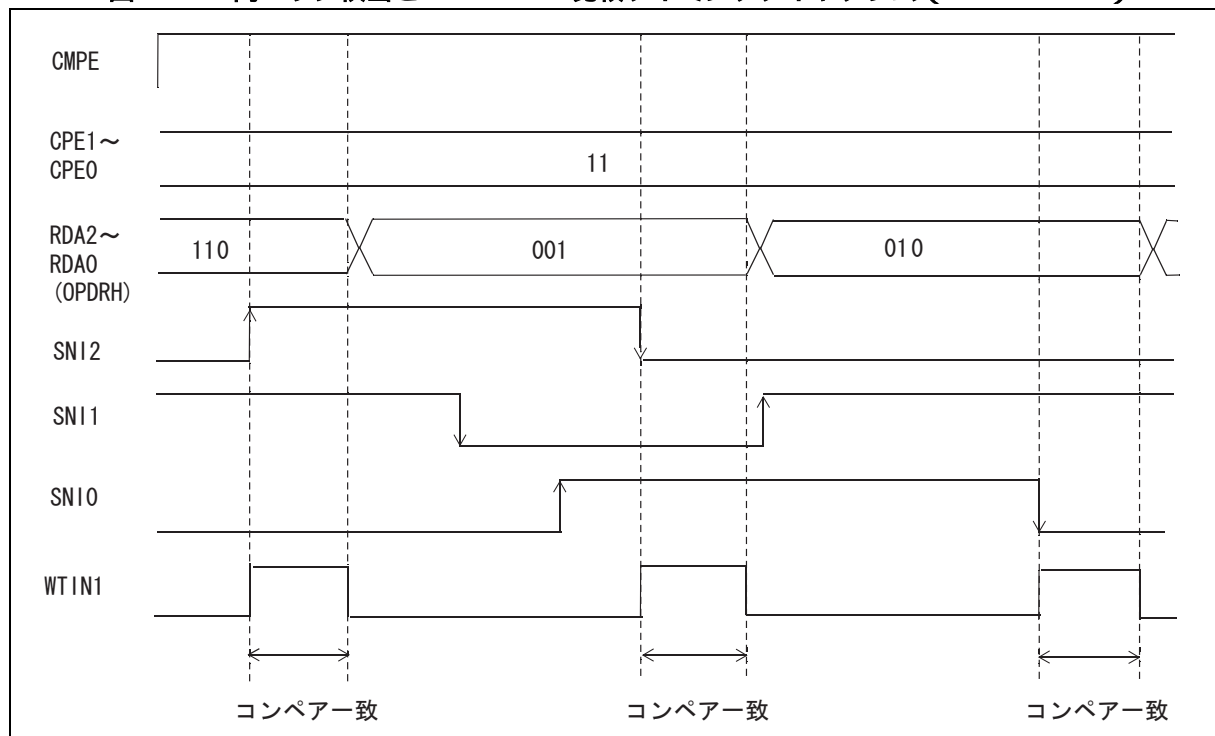
図 15.6-3 エッジ検出タイミングダイアグラム (CMPE:bit8=0)





## ● 両エッジ検出と SNIx/RDAx 比較タイミングダイアグラム (CMPE:bit8=1)

図 15.6-4 両エッジ検出と SNIx/RDAx 比較タイミングダイアグラム (CMPE:bit8=1)



## ■ WTIN1 出力状態とレジスタ設定

表 15.6-2 WTIN1 出力状態とレジスタ設定

CMPE	CPE1	CPE0	SEEx	WTIN1 出力状態
0	0	0	0	出力なし (初期値)
0	X	X	0	出力なし
0	0	0	1	出力なし
0	0	1	1	SNIx 立上りエッジを検出する
0	1	0	1	SNIx 立上りエッジを検出する
0	1	1	1	SNIx 両エッジを検出する
1	0	0	X	設定禁止
1	0	1	X	SNIx 立上りエッジと SNIx/RDAx コンペアー一致を検出する
1	1	0	X	SNIx 立下りエッジと SNIx/RDAx コンペアー一致を検出する
1	1	1	X	SNIx 両エッジと SNIx/RDAx コンペアー一致を検出する

(注意事項)

CMPE:bit8=1 のとき, SEEx には "0" を設定してください (SEEx に "1" を設定することは推奨しません)。

## 15.6.2 データ書込み制御ユニットの動作

データ書込み制御ユニットは、出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から出力データレジスタ (OPDRH, OPDRL) へデータを転送するために必要となる書込みタイミング出力 (WTO) を生成します。

### ■ データ書込み制御ユニットの動作

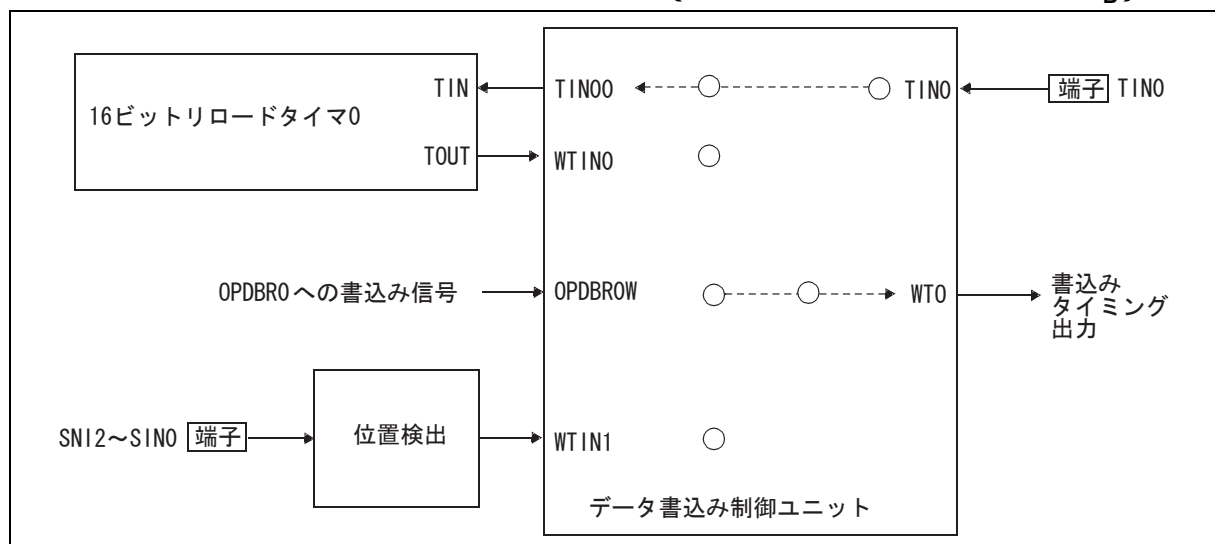
書込みタイミング出力 (WTO) は、以下の場合に生成できます。

- 出力データバッファレジスタ OPDBR0 への書込みがソフトウェア的に行われる。
- 16 ビットリロードタイマ 0 のアンダフローがトリガとなる。
- 16 ビットリロードタイマ 0 のアンダフローがトリガとなる (16 ビットタイマは位置検出比較回路で起動される)。
- 位置検出入力 (SNI2 ~ SNI0) がトリガとなる (16 ビットリロードタイマ 0 は遅延を発生させる)。
- 16 ビットリロードタイマ 0 のアンダフローがトリガとなるかまたは位置検出入力 がトリガとなる。

WTO の生成要因は、出力制御レジスタ上位 (OPCUR) の OPS2 ~ OPS0:bit12 ~ bit10 の値を設定することで定義されます。

- OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub> の場合における出力データバッファレジスタ (OPDBR0) の信号フローダイアグラム

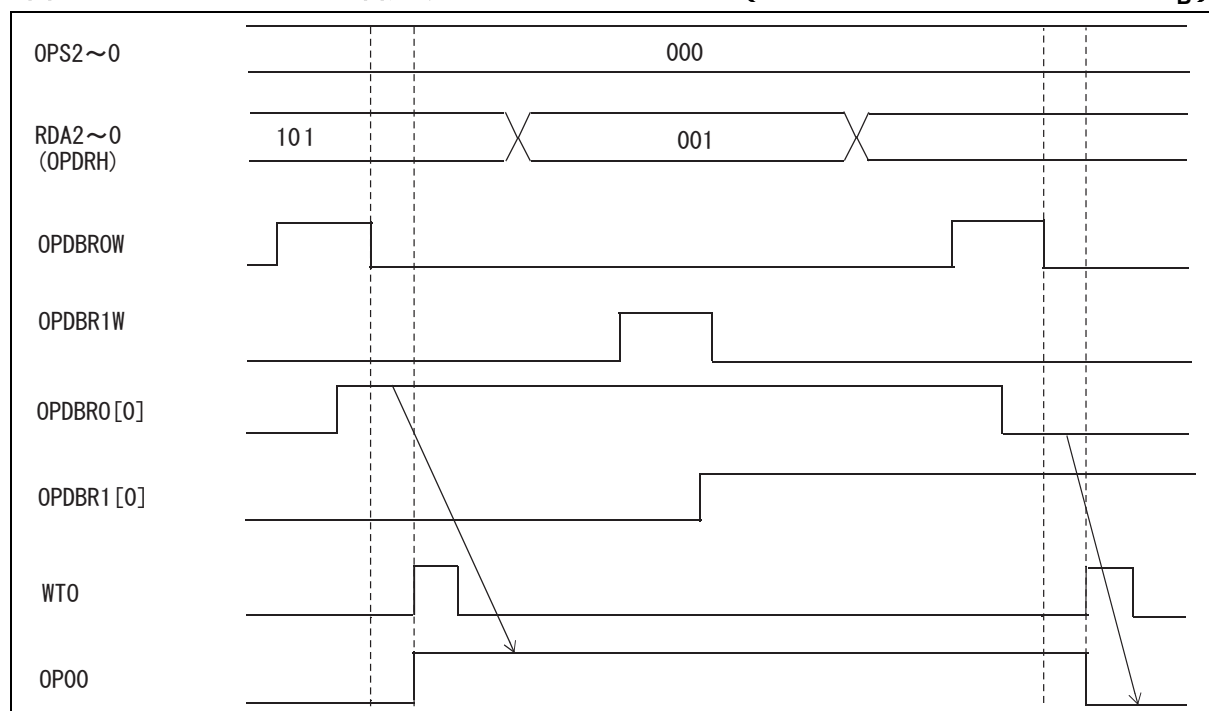
図 15.6-5 OPDBR0 の信号フローダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>)



書込みタイミング出力信号は、値が OPDBR0 レジスタに書き込まれるたびにデータ書込み制御ユニットから生成され、また OPDBR0 のデータは 1 サイクル後に出力データレジスタ (OPDRH, OPDRL) へ転送されます。

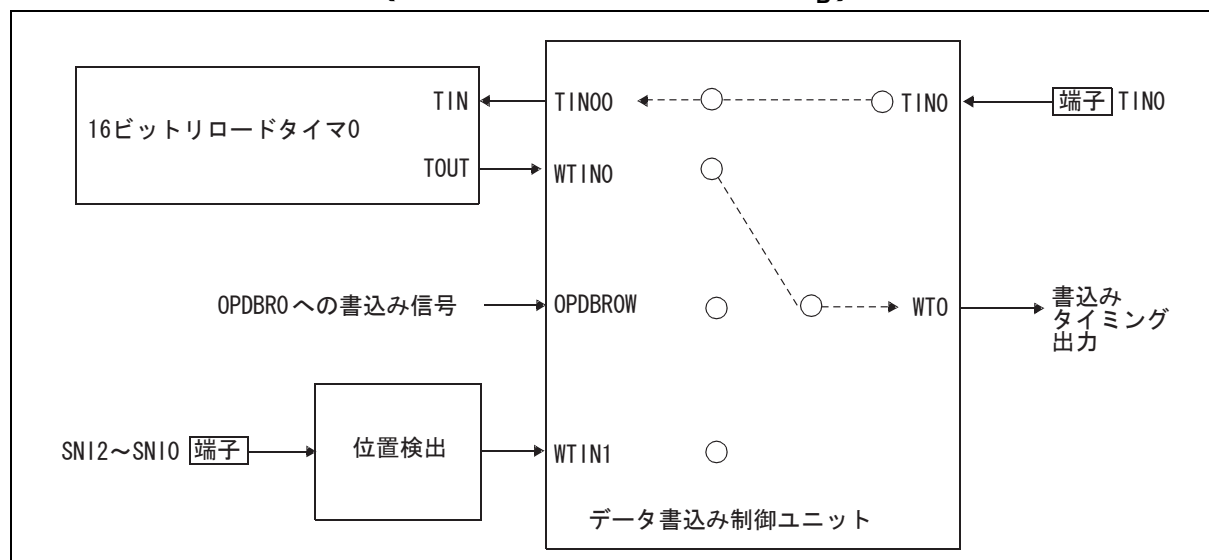
● OPDRH レジスタ書き込みタイミングダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>)

図 15.6-6 OPDRH レジスタ書き込みタイミングダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>)



● リロードタイマ 0 のアンダフローの信号フローダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>)

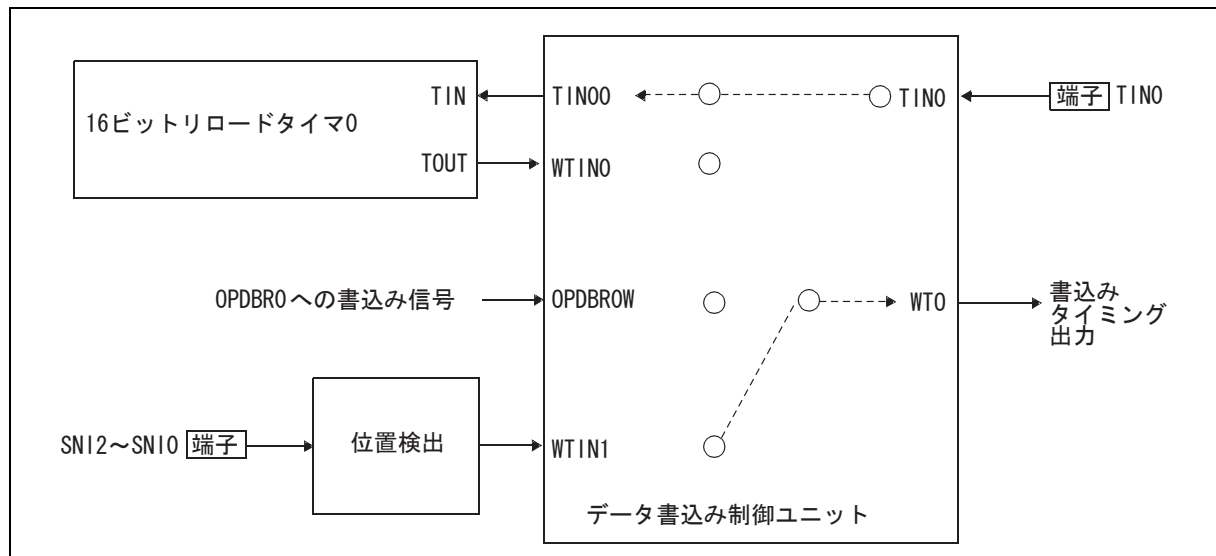
図 15.6-7 リロードタイマ 0 のアンダフローの信号フローダイアグラム  
(OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>)



16 ビットリロードタイマ 0 は、この設定で書き込み信号を生成するために TIN 入力とソフトウェアの両方で起動できます。書き込み信号は、16 ビットリロードタイマ 0 のアンダフローで制御されます。

- 位置検出の信号フローダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=010<sub>B</sub> または 110<sub>B</sub>)

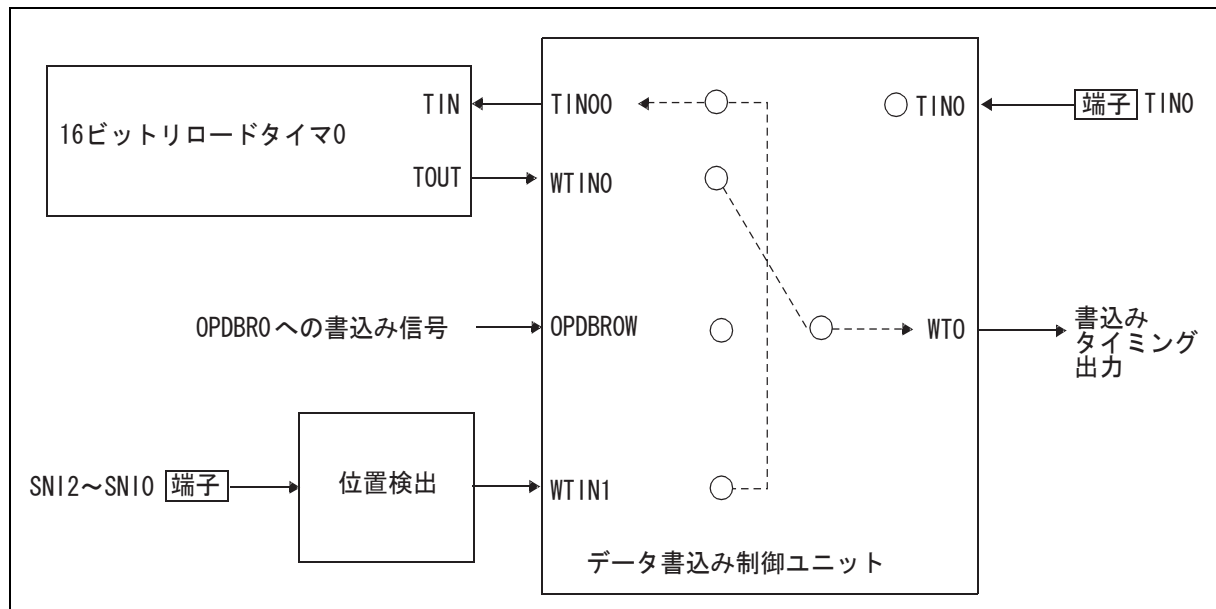
図 15.6-8 位置検出の信号フローダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=010<sub>B</sub> または 110<sub>B</sub>)



書き込み信号は、コンペア一致で生成されるかまたは位置検出の有効エッジ入力で生成されます。

- リロードタイマ0と位置検出の信号フローダイアグラム(OPS2 ~ OPS0:bit12 ~ bit10=011<sub>B</sub> または 111<sub>B</sub>)

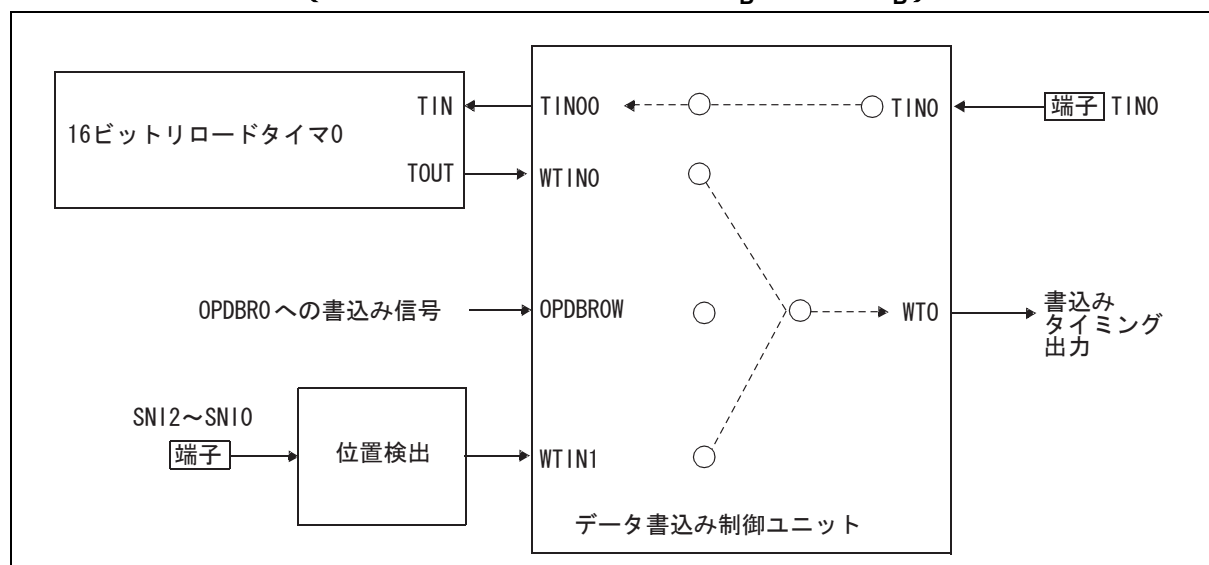
図 15.6-9 リロードタイマ0と位置検出の信号フローダイアグラム  
(OPS2 ~ OPS0:bit12 ~ bit10=011<sub>B</sub> または 111<sub>B</sub>)



この設定の場合、16ビットリロードタイマ0はコンペア一致で起動されるかまたは位置検出回路の有効エッジ入力で起動されます。その後、16ビットリロードタイマ0でアンダフローが発生するたびに書き込み信号が生成されます。コンペア一致は、SN1<sub>x</sub>端子の有効エッジの変化がトリガとなります。

- リロードタイマ 0 または位置検出の信号フローダイアグラム (OPS2 ~ OPS0:bit12 ~ bit10=100<sub>B</sub> または 101<sub>B</sub>)

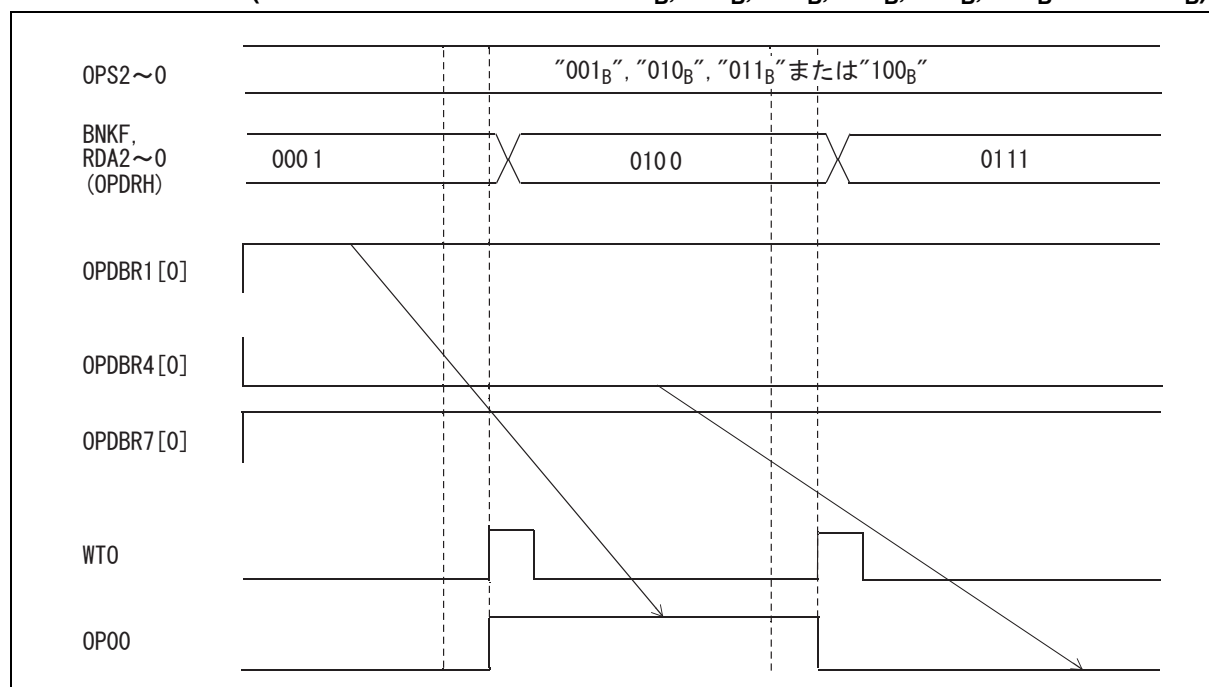
図 15.6-10 リロードタイマ 0 または位置検出の信号フローダイアグラム  
(OPS2 ~ OPS0:bit12 ~ bit10=100<sub>B</sub> または 101<sub>B</sub>)



この設定の場合，書き込み信号はコンペアー一致で生成されるかまたは位置検出の有効エッジ入力で生成されるかまたは 16 ビットリロードタイマ 0 でアンダフローが発生するたびに生成されます。コンペアー一致は，SN<sub>ix</sub> 端子の有効エッジの変化がトリガとなります。

- 出力データレジスタ (OPDRH) 書き込みタイミングダイアグラム  
(OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>, 010<sub>B</sub>, 011<sub>B</sub>, 100<sub>B</sub>, 101<sub>B</sub>, 110<sub>B</sub> または 111<sub>B</sub>)

図 15.6-11 出力データレジスタ (OPDRH) 書き込みタイミングダイアグラム  
(OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>, 010<sub>B</sub>, 011<sub>B</sub>, 100<sub>B</sub>, 101<sub>B</sub>, 110<sub>B</sub> または 111<sub>B</sub>)



### 15.6.3 出力データバッファレジスタの動作

出力データバッファレジスタ (OPDBR0 ~ OPDBRB) は, 12 個のレジスタから構成されています。出力データレジスタ (OPDRH, OPDRL) へさまざまな OPDBR0 ~ OPDBRB レジスタの値をロードすると, マルチパルスジェネレータ出力 (OPT5 ~ OPT0) からさまざまな種類の波形が出力されます。

#### ■ 出力データバッファレジスタの動作

出力データレジスタ上位 (OPDRH) の BNKF:bit15 および RDA2 ~ 0:bit14 ~ bit12 によりアドレスが指定される出力データバッファレジスタ (OPDBR0 ~ OPDBRB) のデータは, データ書込み制御ユニットで生成される書込みタイミングで出力データレジスタ (OPDRH, OPDRL) へ転送されます。

出力データバッファレジスタ (OPDBR0 ~ OPDBRB) の BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 は出力データレジスタ (OPDRH, OPDRL) へのデータ転送順序を決定し, OPx1/OPx0: bit11 ~ bit0 は出力波形の形状を決定します。出力波形は, 書込みタイミング (WTO) が生成されるごとに自動的に更新されます。

出力データバッファレジスタ (OPDBR0 ~ OPDBRB) の設定例を表 15.6-3 に示します。

表 15.6-3 出力データバッファレジスタ (OPDBR0 ~ OPDBRB) ( 1 / 2 )

No.	0	1	2	3	4	5	6	7	8	9	A	B
BNKF	0	0	0	0	0	1	0	X	X	0	1	1
RDA2	1	1	0	0	1	0	0	X	X	1	0	0
RDA1	0	0	1	0	0	1	1	X	X	0	1	0
RDA0	0	1	1	1	0	0	0	X	X	0	1	1
OP51	0	0	0	1	0	0	0	X	X	0	0	1
OP50	0	0	1	1	0	0	0	X	X	0	1	1
OP41	1	0	0	0	0	1	0	X	X	0	0	0
OP40	1	1	0	0	0	1	0	X	X	1	0	0
OP31	0	0	0	0	0	0	1	X	X	0	0	0
OP30	0	0	0	0	1	0	1	X	X	0	0	0
OP21	0	0	0	0	1	0	0	X	X	0	0	0
OP20	1	0	0	0	1	1	0	X	X	0	0	0
OP11	0	0	1	0	0	0	0	X	X	0	1	0
OP10	0	0	1	0	0	0	1	X	X	0	1	0
OP01	0	1	0	0	0	0	0	X	X	1	0	0
OP00	0	1	0	1	0	0	0	X	X	1	0	1
OPBDR 番号の シーケンス	4	5	3	1	6	A	2	X	X	4	B	9

表 15.6-3 出力データバッファレジスタ (OPDBR0 ~ OPDBRB)( 2 / 2 )

No.	0	1	2	3	4	5	6	7	8	9	A	B
OPT5 出力	L	L	PPG	H	L	L	L	X	X	L	PPG	H
OPT4 出力	H	PPG	L	L	L	H	L	X	X	PPG	L	L
OPT3 出力	L	L	L	L	PPG	L	H	X	X	L	L	L
OPT2 出力	PPG	L	L	L	H	PPG	L	X	X	L	L	L
OPT1 出力	L	L	H	L	L	L	PPG	X	X	L	H	L
OPT0 出力	L	H	L	PPG	L	L	L	X	X	H	L	PPG

出力データバッファレジスタ (OPDBR0) を、表 15.6-3 のように設定すると、出力データレジスタ (OPDRH, L) の値は初期化されます。以下のシーケンスは、生成される書き込みタイミングに従って動作を開始します。

**No.4 No.6 No.2 No.3 No.1 No.5 No.A No.B No.9 (No.4 に戻ってリサイクル)**

データは、出力データレジスタ (OPDRH, OPDRL) へ順次転送されます。出力データバッファレジスタ (OPDBR0 ~ OPDBRB) は、なにも設定されていない場合 (たとえば、表 15.6-3 の No.7 および No.8) は使用されません。

## 15.6.4 出力データレジスタへのデータ転送動作

出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から出力データレジスタ (OPDRH, OPDRL) へ自動でデータを転送をする際には, 8 つの方式 (以降の小項で説明) を使用できます。各方式は, 出力制御レジスタ (OPCUR) の OPS2 ~ OPS0:bit12 ~ bit10 を設定することによって選択します。

### ■ 出力データレジスタへのデータ転送動作

出力データバッファレジスタ(OPDBR0 ~ OPDBRB)から出力データレジスタ(OPDRH, OPDRL)へのデータ転送には, 以下に示す 8 つの方式を使用できます。

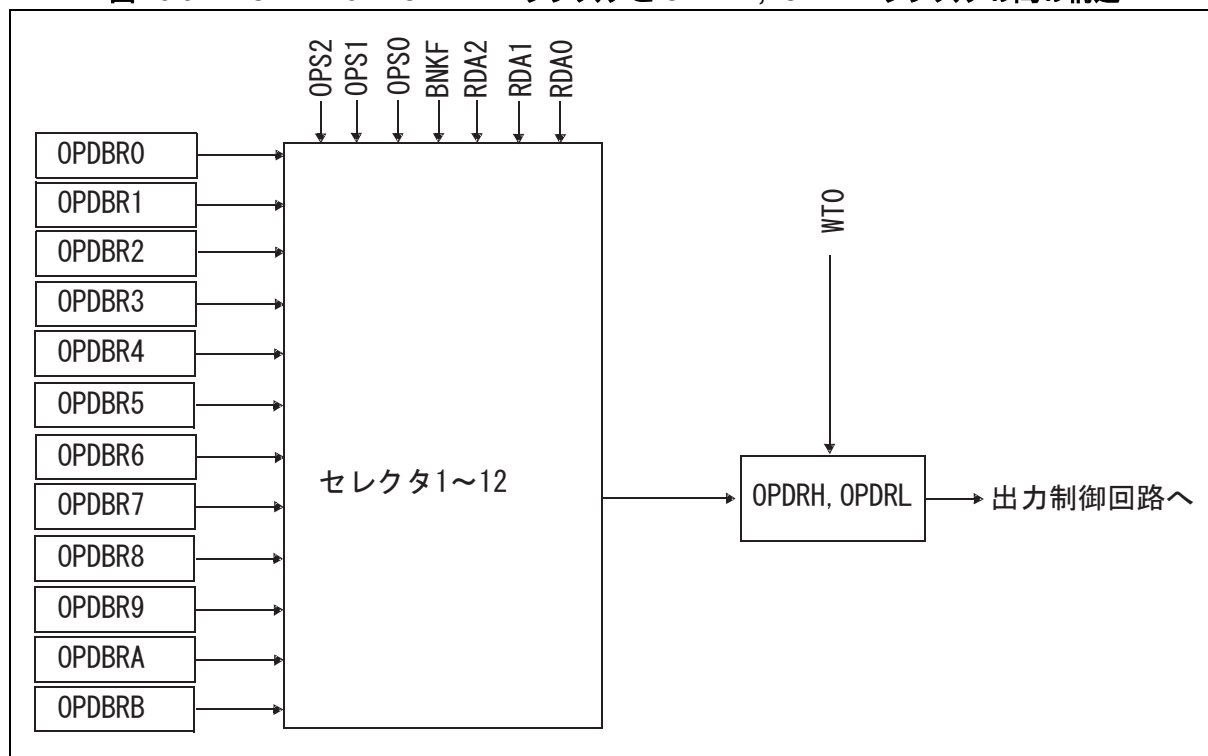
- OPDBR0 書込み
- 16 ビットリロードタイマアンダフロー
- 位置検出
- 位置検出と 16 ビットリロードタイマアンダフロー
- 位置検出または 16 ビットリロードタイマアンダフロー
- ワンショット位置検出
- ワンショット位置検出と 16 ビットリロードタイマアンダフロー
- ワンショット位置検出または 16 ビットリロードタイマアンダフロー

出力データレジスタ上位(OPDRH)の BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 により選択された出力データバッファレジスタ(OPDBR0 ~ OPDBRB)値は, データ書込み制御回路において書込み信号が生成されると, 出力データレジスタ(OPDRH, OPDRL)へ転送されます。ただし, OPS2 ~ OPS0=000<sub>B</sub> の場合は, BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 の値とは無関係に, 必ず OPDBR0 の値が出力データレジスタ(OPDRH, OPDRL)へ転送されます。OPDBR0 ~ OPDBRB レジスタと OPDRH, OPDRL レジスタの間の構造を図 15.6-12 に示します。

< 注意事項 > データ転送方式を変更すると, 次に選択されるデータバッファレジスタは, 必ずデータ出力レジスタ上位(OPDRH)の BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 で指定されます。これは, BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 が無視される "OPDBR0 書込み" 方式には適用されません。出力データレジスタへアクセスする場合は, ワードアクセス命令をご使用ください。



図 15.6-12 OPDBR0 ~ OPDBRB レジスタと OPDRH, OPDRL レジスタの間の構造



### 15.6.4.1 "OPDBR0 書込み" 方式

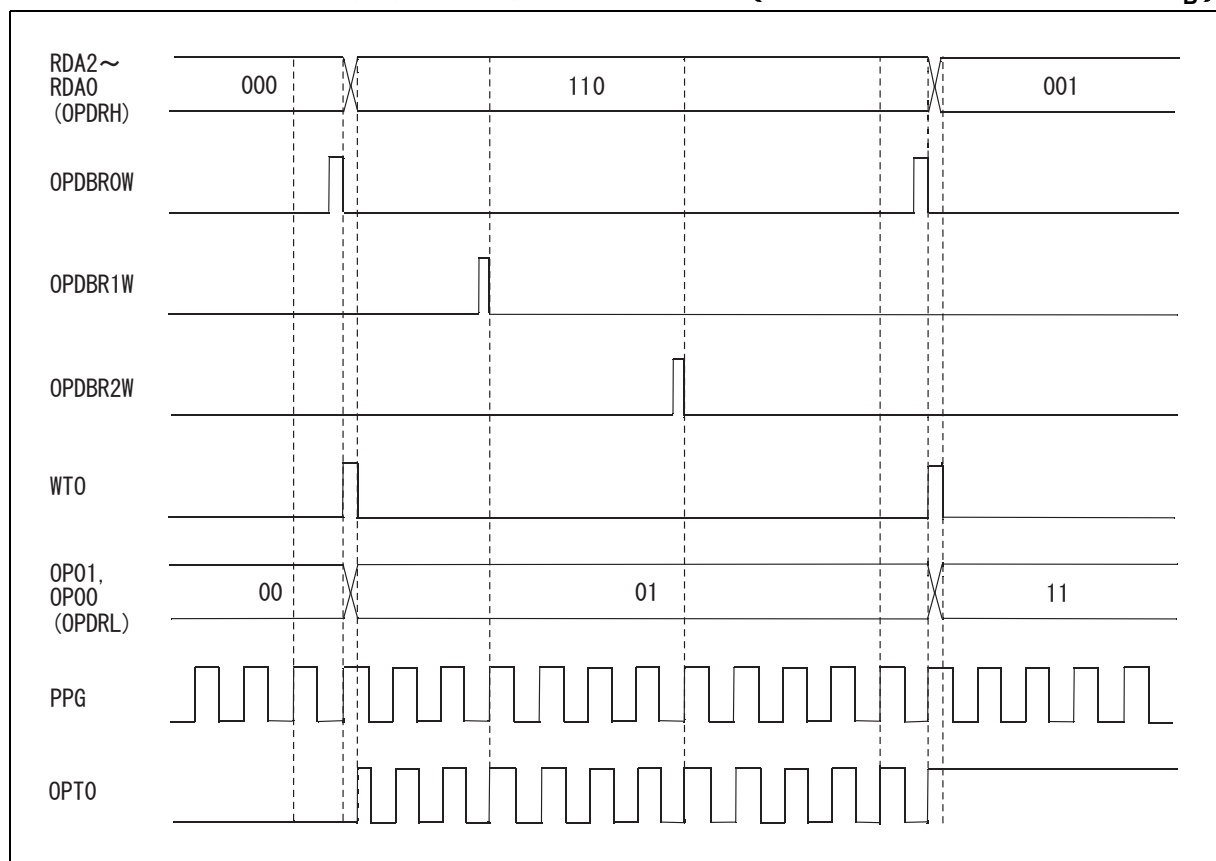
"OPDBR0 書込み" 方式の転送時における出力端子 (OPTx) のタイミング更新を図 15.6-13 に示します。

#### ■ "OPDBR0 書込み" 方式

< 注意事項 > この動作時は、出力データバッファレジスタ 0 に対してワードアクセス命令をご使用ください。下位レジスタまたは上位レジスタへバイトアクセスしても転送動作は開始しません。この方式は、リロードタイマ 0 を自由に使用できます。

● "OPDBR0 書込み" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>)

図 15.6-13 "OPDBR0 書込み" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>)



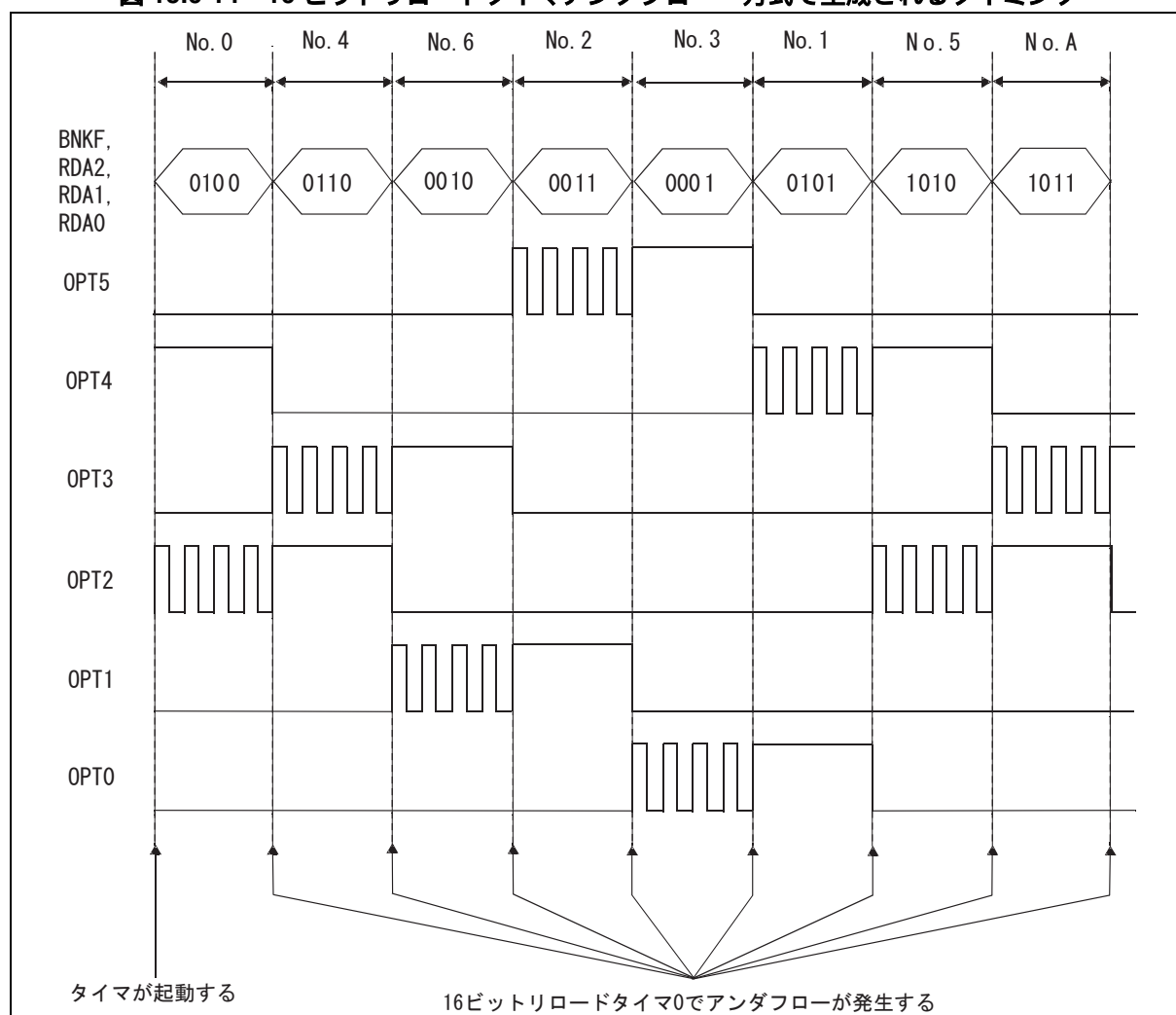
## 15.6.4.2 "16 ビットリロードタイマアンダフロー" 方式

"16 ビットリロードタイマアンダフロー" 方式の転送時における出力端子 (OPTx) のタイミング更新を図 15.6-14 および図 15.6-15 に示します。

### ■ "16 ビットリロードタイマアンダフロー" 方式

- "16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング

図 15.6-14 "16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング

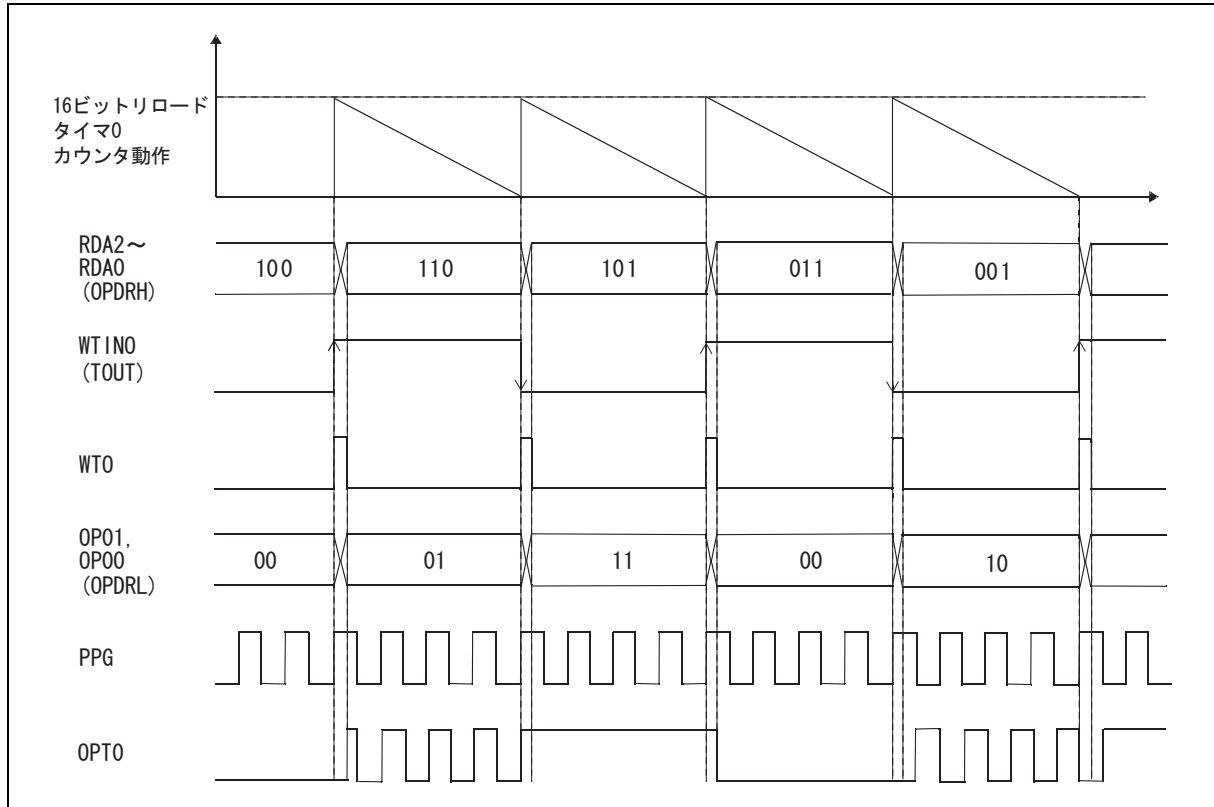


出力データレジスタ上位 (OPDRH) の BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 により指定される出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から出力データレジスタ (OPDRH, OPDRL) へのデータ転送は, 図 15.6-15 に示されているように 16 ビットリロードタイマ 0 のアンダフローが生成されるたびに自動的に更新されます。

この方式を使用するには, リロードタイマを "リロードモード" で使用しなければなりません。リロードタイマを起動するには, ソフトウェアトリガを使用しなければなりません。前もって更新時間を設定し, 連続的制御動作を実行するには, 16 ビットリロードタイマ 0 を使用しなければなりません。

- "16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>)

図 15.6-15 "16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=001<sub>B</sub>)



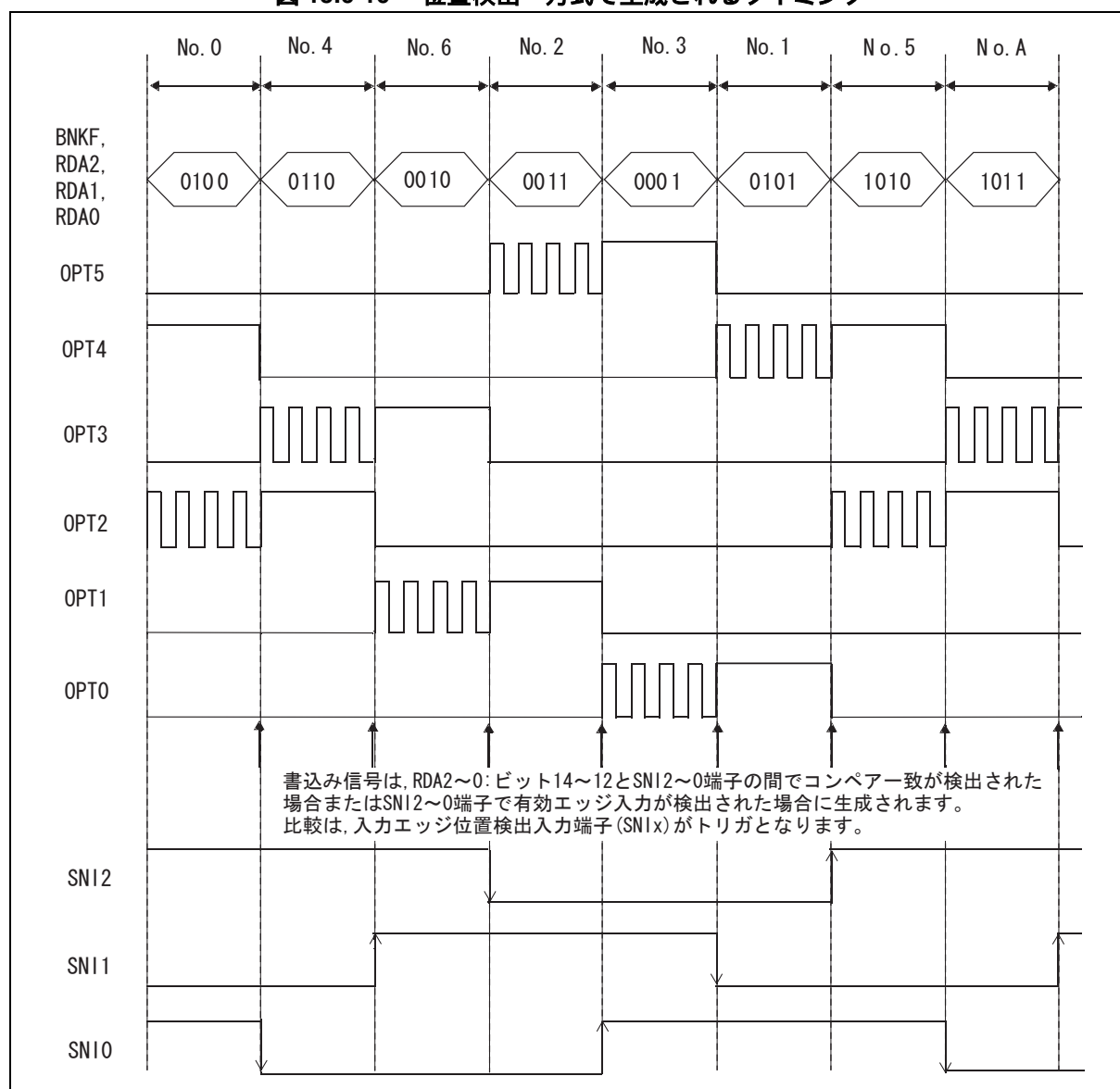
### 15.6.4.3 "位置検出" 方式

"位置検出"方式の転送時における入力端子 (SNIx) がトリガとなる出力タイミング更新を図 15.6-16 および図 15.6-17 に示します。

#### ■ "位置検出" 方式

- "位置検出"方式で生成されるタイミング

図 15.6-16 "位置検出"方式で生成されるタイミング



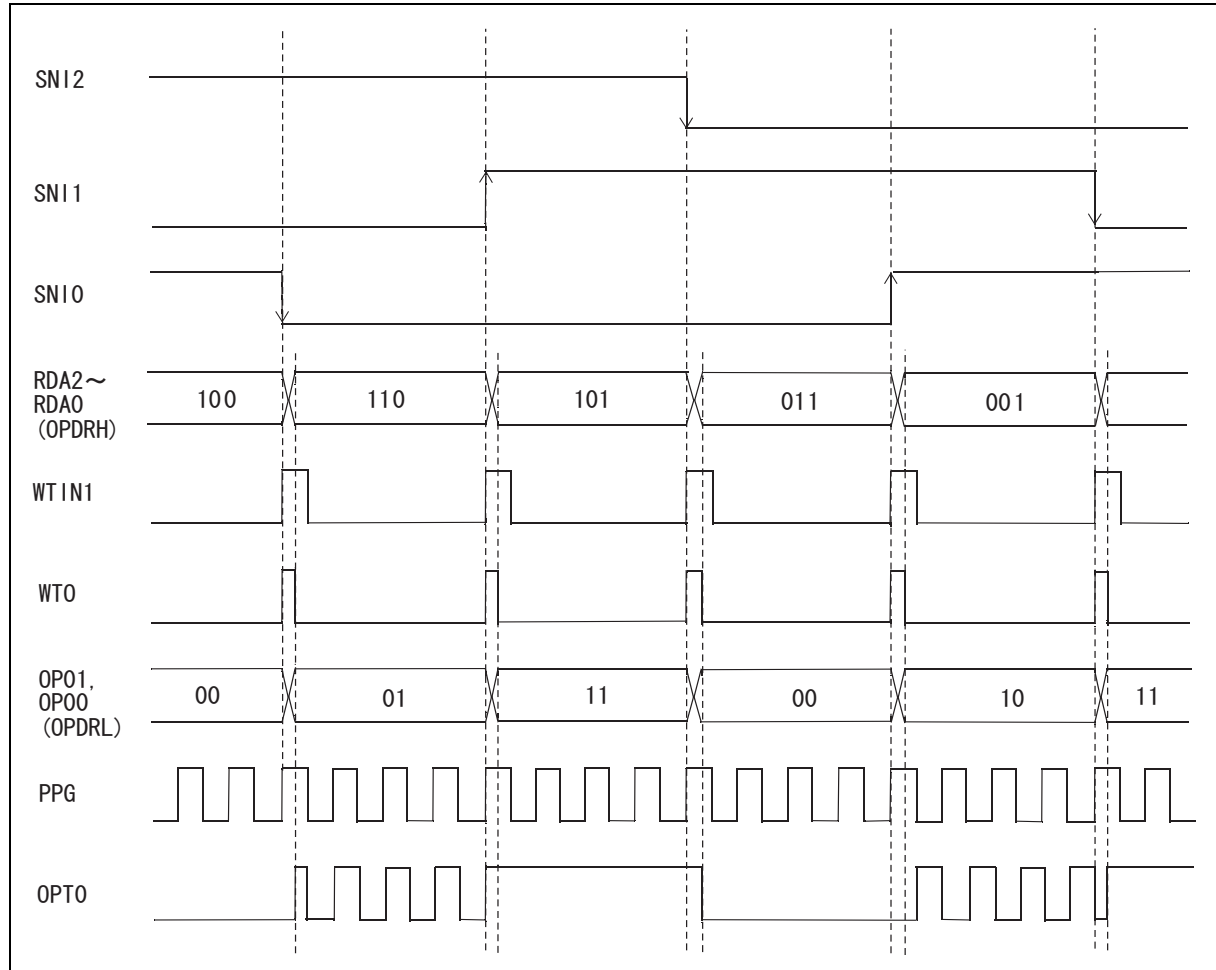
比較は, 各位置検出において SNI2 端子と出力データレジスタ上位 (OPDRH) の RDA2:bit14 の間, SNI1 端子と RDA1:bit13 の間および SNI0 端子と RDA0:bit12 の間で行われます。OPTx 出力波形は, 図 15.6-17 のように SNIx 端子の有効エッジ入力に従って更新されます。BNKF:bit15 および RDA2 ~ RDA0:bit14 ~ bit12 で指定された出力データバッファレジスタ(OPDBR0 ~ OPDBRB)データは出力データレジスタ(OPDRH,

OPDRL) へ転送され, 出力データは SNI2 ~ SNI0 端子と RDA2 ~ RDA0:bit14 ~ bit12 値の間でコンペア一致が検出されると自動的に更新されます。

この方式時は, リロードタイマ 0 を自由に使用できます。

- "位置検出" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=010<sub>B</sub>)

図 15.6-17 "位置検出" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=010<sub>B</sub>)



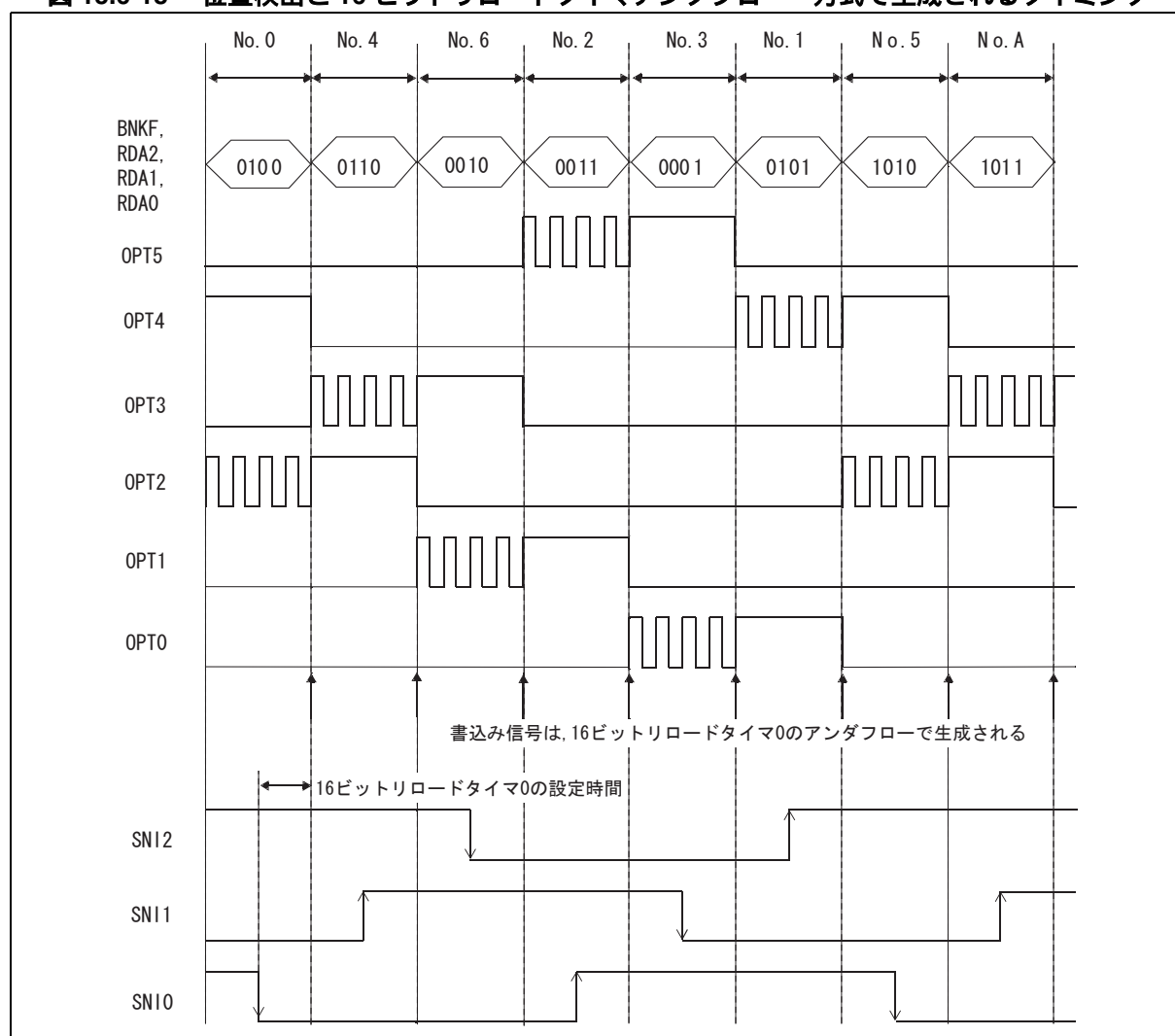
### 15.6.4.4 "位置検出と 16 ビットリロードタイマアンダフロー" 方式

"位置検出と 16 ビットリロードタイマアンダフロー" 方式の転送時における出力タイミング更新を図 15.6-18 および図 15.6-19 に示します。

#### ■ "位置検出と 16 ビットリロードタイマアンダフロー" 方式

- "位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング

図 15.6-18 "位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング



位置検出での比較は、 $\text{SNI}_x$  端子と出力データレジスタ上位 (OPDRH) の  $\text{RDA}_x$ :bit14, bit13 または bit12 の間で行われます (すなわち、 $\text{SNI}_2$  端子と  $\text{RDA}_2$ :bit14 の間、 $\text{SNI}_1$  端子と  $\text{RDA}_1$ :bit13 の間および  $\text{SNI}_0$  端子と  $\text{RDA}_0$ :bit12 の間で行われます)。コンパレー一致が検出されると、16 ビットリロードタイマ 0 が起動します。書込み信号は、16 ビットリロードタイマ 0 のアンダフローで生成されます。

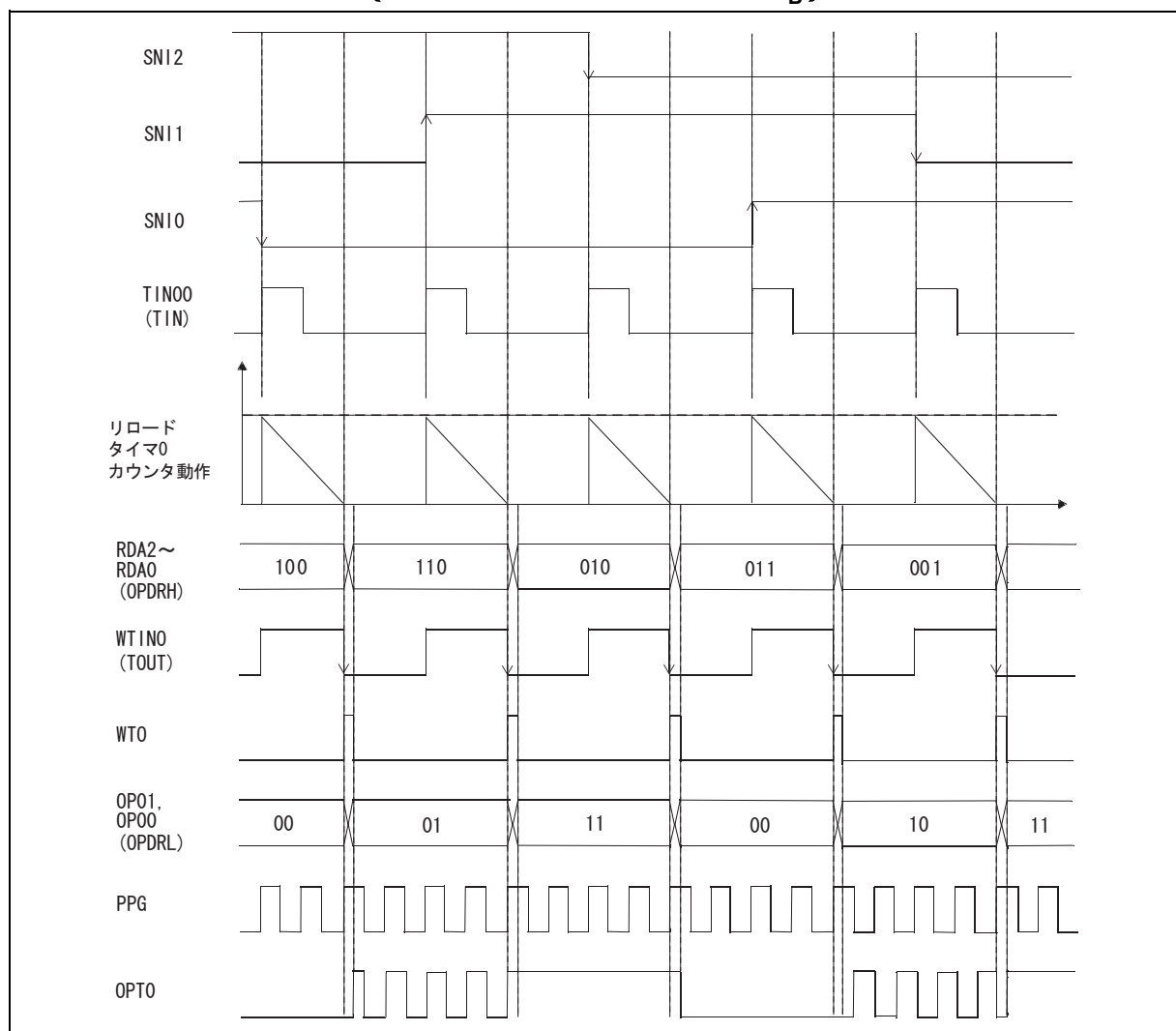
$\text{SNI}_x$  端子の有効エッジ入力に従って出力される  $\text{OPT}_x$  端子出力波形は、図 15.6-19 に示

されているとおりです。16 ビットリロードタイマ 0 は、SNI2 ~ SNI0 端子と出力データレジスタ上位 (OPDRH) の RDA2 ~ RDA0:bit14 ~ bit12 値の間でコンペア一致が検出されると起動します。RDA2 ~ RDA0:bit14 ~ bit12 で指定された出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から出力データレジスタ (OPDRH, OPDRL) へのデータ転送は、16 ビットリロードタイマ 0 のアンダフローがトリガとなります。出力データの動作は自動的に更新されます。

この方式を使用するには、リロードタイマは "ワンショットモード" で使用しなければなりません。TIN00 は、2 マシンサイクルより長くなければなりません。

- "位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=011<sub>B</sub>)

図 15.6-19 "位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=011<sub>B</sub>)





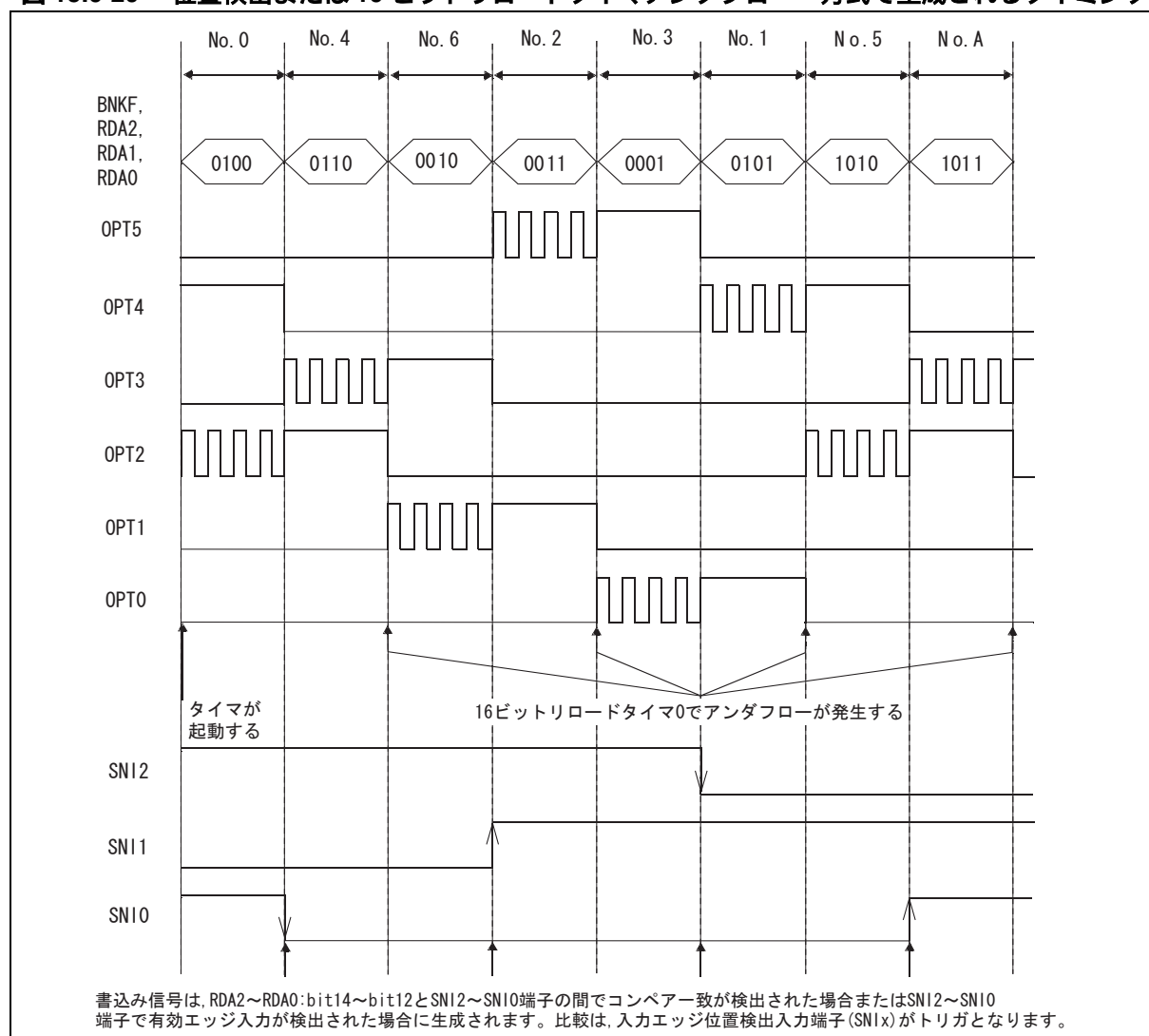
### 15.6.4.5 "位置検出または 16 ビットリロードタイマアンダフロー" 方式

"位置検出または 16 ビットリロードタイマアンダフロー" 方式の転送時における出力タイミング更新を図 15.6-20 および図 15.6-21 に示します。この方式は, 出力制御レジスタ上位 (OPCUR) の OPS2 ~ OPS0:bit12 ~ bit10=100<sub>B</sub> を設定することにより選択されます。

#### ■ "位置検出または 16 ビットリロードタイマアンダフロー" 方式

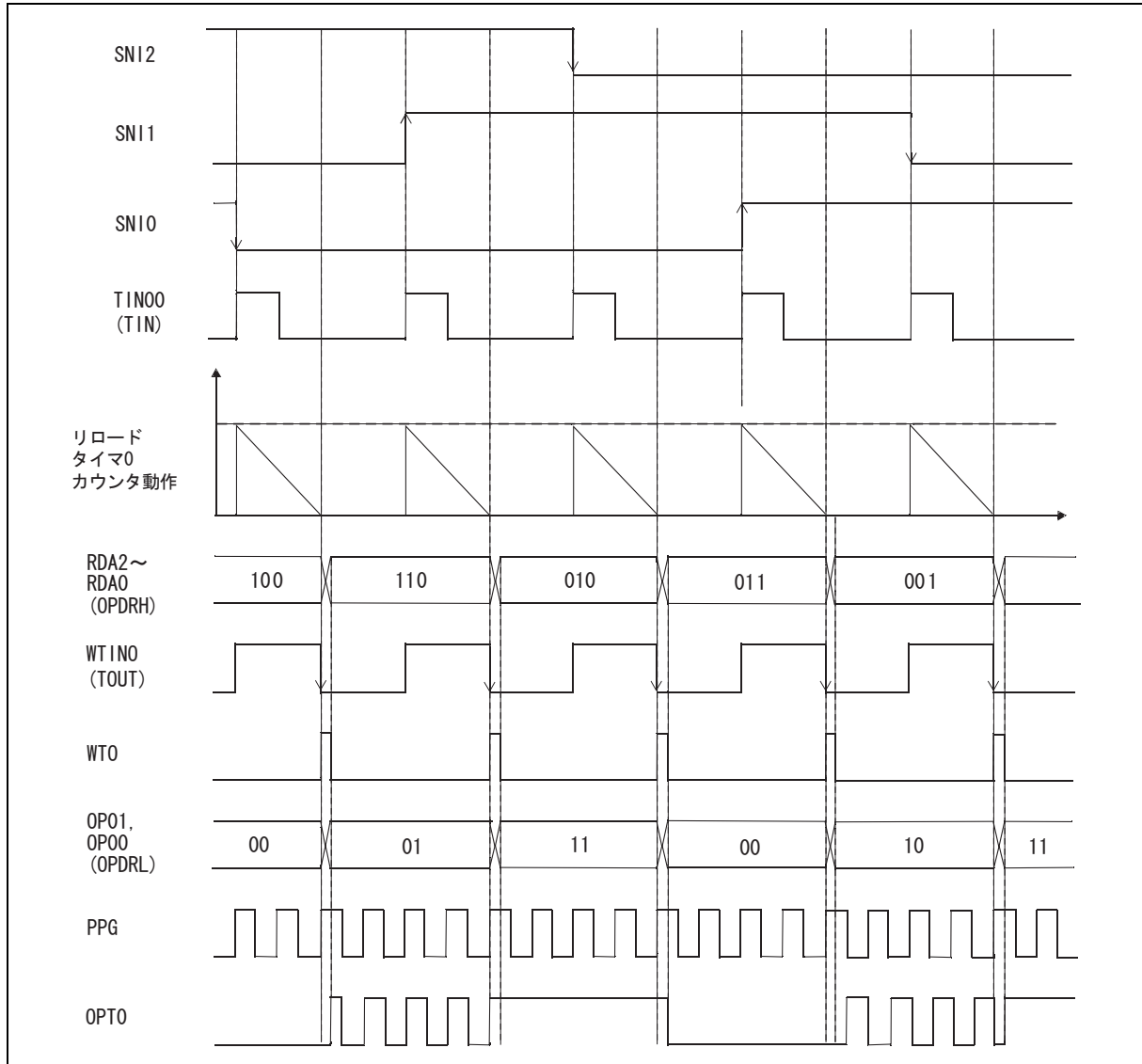
- "位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング

図 15.6-20 "位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング



- "位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=100<sub>B</sub>)

図 15.6-21 "位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=100<sub>B</sub>)



### 15.6.4.6 "ワンショット位置検出" 方式

"ワンショット位置検出" 方式の転送時における入力端子 (SN<sub>i</sub>x) がトリガとなる出力タイミング更新を図 15.6-22 に示します。

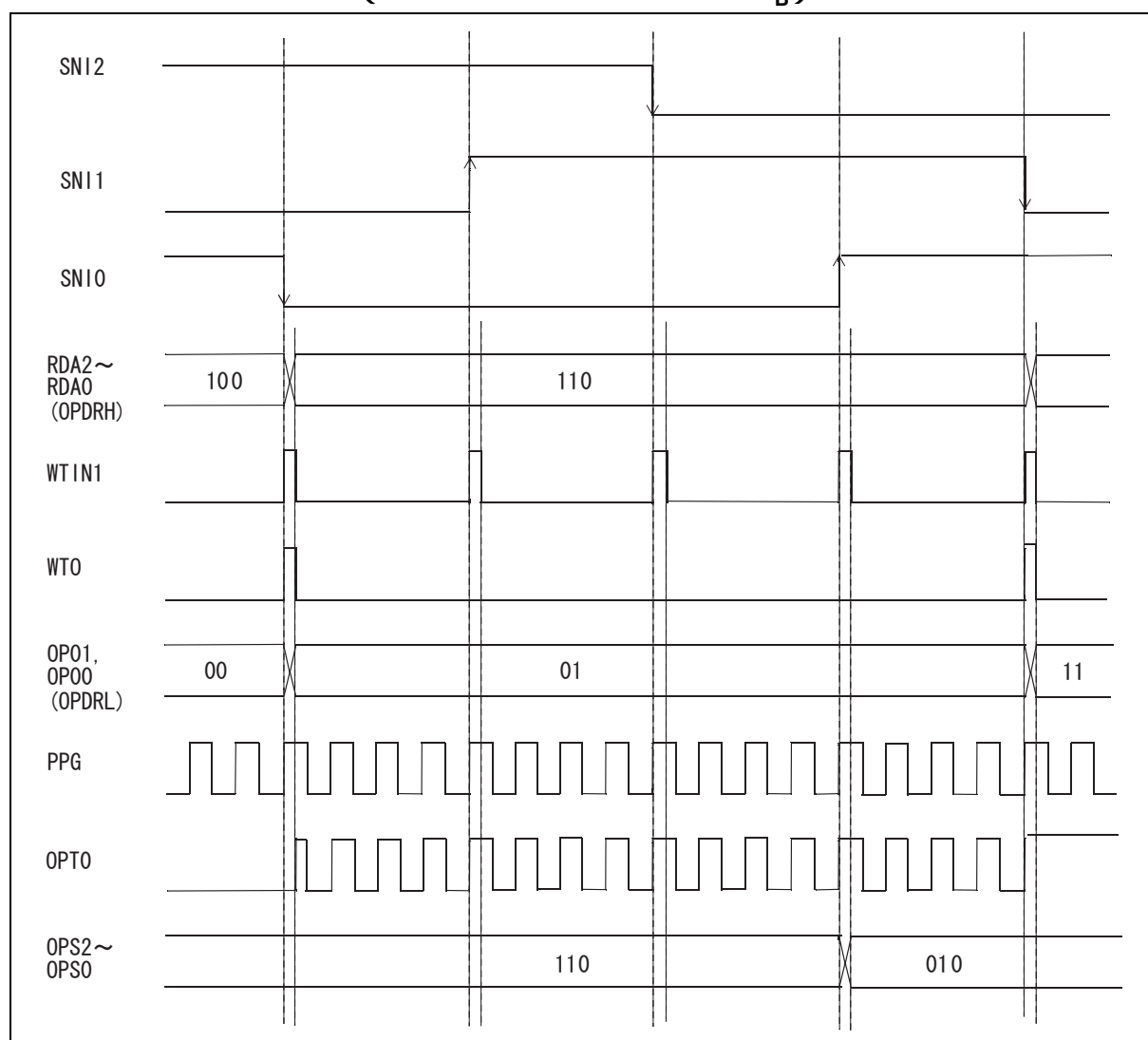
#### ■ "ワンショット位置検出" 方式

最初の有効位置が検出されてから動作モードが変更されるまでに位置検出が認識されない点を除き, "位置検出" 方式の動作と同じです。OPT<sub>x</sub> 出力波形をに示します。

この方式時は, リロードタイマ 0 を自由に使用できます。

- "ワンショット位置検出" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=110<sub>B</sub>)

図 15.6-22 "ワンショット位置検出" 方式で生成されるタイミング  
(OPS2 ~ OPS0:bit12 ~ bit10=110<sub>B</sub>)



### 15.6.4.7 "ワンショット位置検出と 16 ビットリロードタイマアンダフロー" 方式

"ワンショット位置検出と 16 ビットリロードタイマアンダフロー" 方式の転送時における出力タイミング更新を図 15.6-23 に示します。

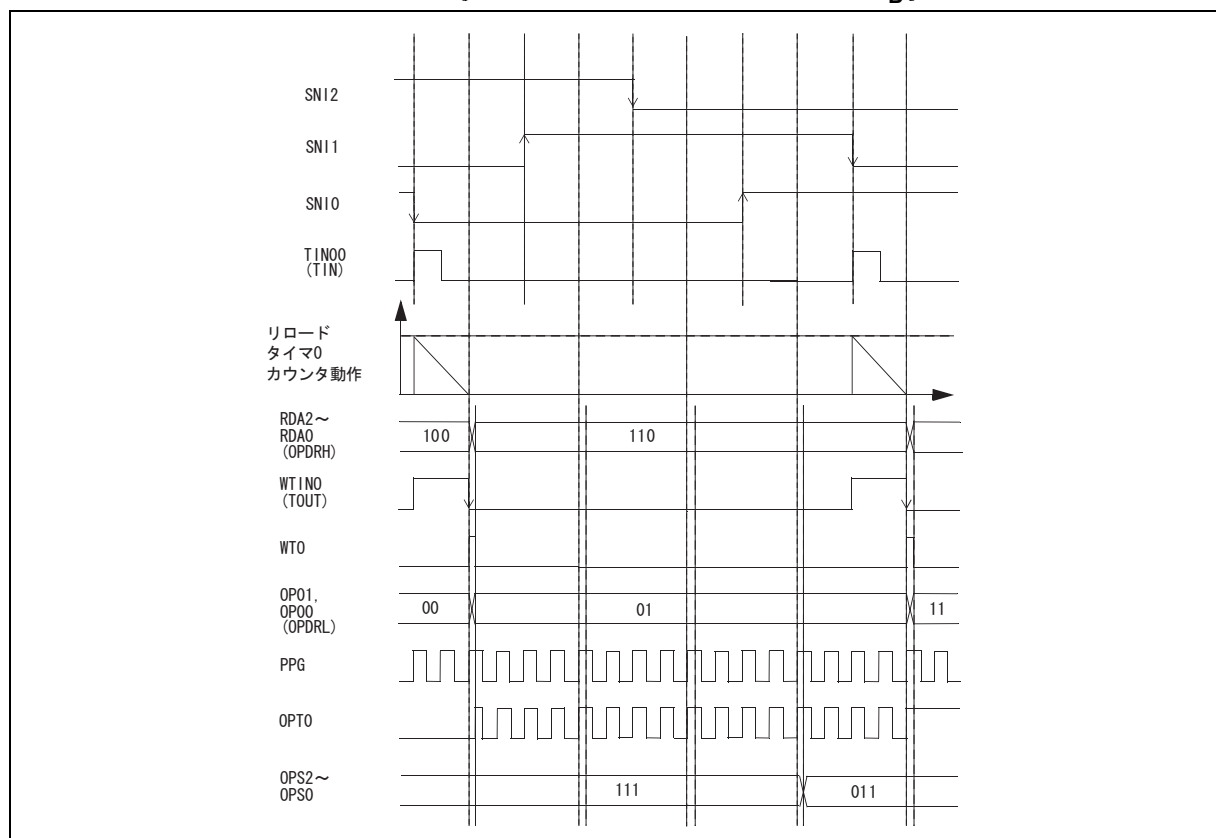
#### ■ "ワンショット位置検出と 16 ビットリロードタイマアンダフロー" 方式

最初の有効位置が検出されてから動作モードが変更されるまでに位置検出が認識されない点を除き, "位置検出と 16 ビットリロードタイマアンダフロー" 方式の動作と同じです。OPTx 端子出力波形を図 15.6-23 に示します。

この方式を使用するには, リロードタイマは"ワンショットモード"で使用しなければなりません。TIN00 は, 2 マシンサイクルより長くなければなりません。

- "ワンショット位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=111<sub>B</sub>)

図 15.6-23 "ワンショット位置検出と 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=111<sub>B</sub>)



### 15.6.4.8 "ワンショット位置検出または 16 ビットリロードタイマアンダフロー" 方式

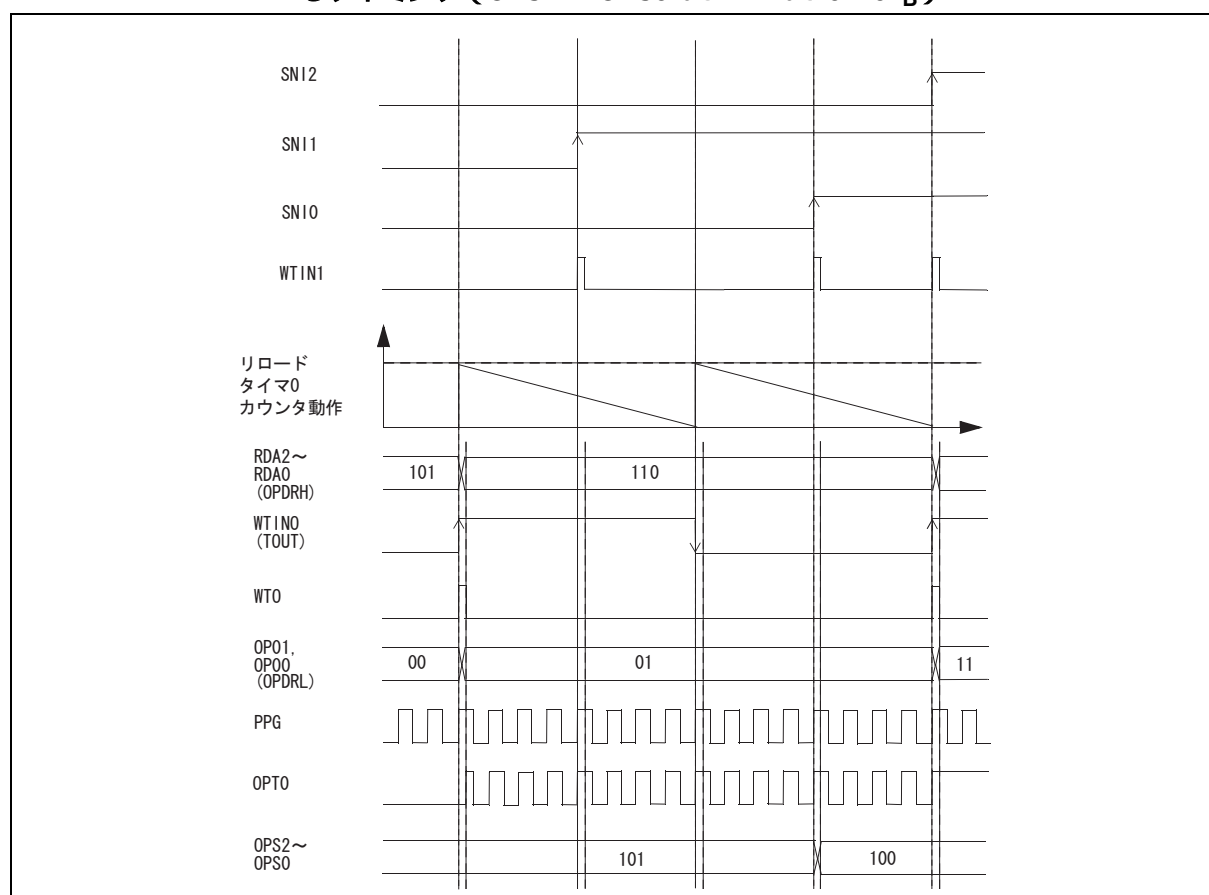
"ワンショット位置検出または 16 ビットリロードタイマアンダフロー" 方式転送時における出力タイミング更新を図 15.6-24 に示します。この方式は、出力制御レジスタ上位 (OPCUR) の OPS2 ~ OPS0:bit12 ~ bit10=101<sub>B</sub> の設定により選択されます。

#### ■ "ワンショット位置検出または 16 ビットリロードタイマアンダフロー" 方式

最初の有効位置が検出されてから動作モードが変更されるまでに位置検出が認識されない点を除き "位置検出または 16 ビットリロードタイマアンダフロー" 方式の転送動作と同じです。OPTx 端子出力波形を図 15.6-24 に示します。

- "ワンショット位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=101<sub>B</sub>)

図 15.6-24 "ワンショット位置検出または 16 ビットリロードタイマアンダフロー" 方式で生成されるタイミング (OPS2 ~ OPS0:bit12 ~ bit10=101<sub>B</sub>)



## 15.6.5 DTTI 入力制御の動作

DTTI 入力制御回路の動作について説明します。

### ■ DTTI 入力制御の動作

DTTI 回路は、PORTx となる OPTx 端子への PDRx (PORTx データレジスタ) 値の出力を制御します (OPTx 端子は出力制御レジスタ下位 (OPCLR) の OPEx:bit4 ~ bit1 または 0 に "1" を設定すると選択されます)。動作モードは、出力制御レジスタ上位 (OPCUR) の DTIE:bit15 に "1" を設定すると有効になります。

< 注意事項 > DTTI 回路が有効になる前に、OPTx 端子となる PORTx がポート方向レジスタにより出力ポートとして設定されていることを確認してください。

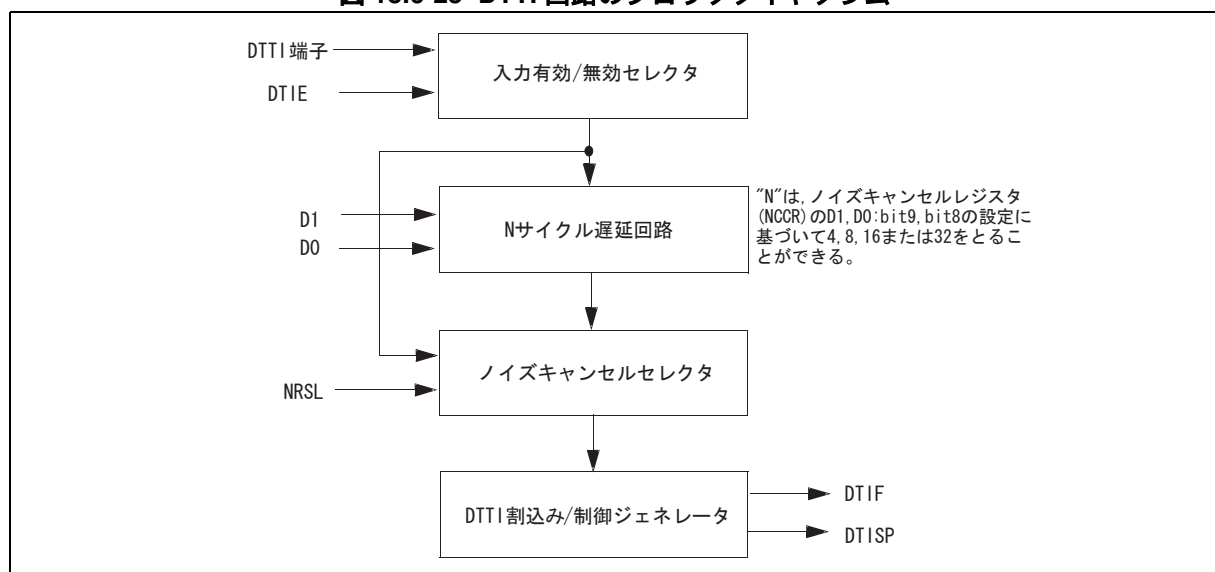
出力制御レジスタ上位 (OPCUR) の DTIE:bit14 に "1" が設定されると、OPT5 ~ OPT0 端子の波形出力は DTTI 端子の有効レベルで有効になります。DTTI 端子が low 入力レベルに設定されると、OPTx 出力はインアクティブレベルに固定されます。ソフトウェアは、PORTx の PDRx に対する各 OPTx 端子をインアクティブレベルに設定できます。OPTx 端子は、PORTx の PDRx に書き込まれたデータで動作します。

出力が DTTI 端子の入力によりインアクティブレベルに固定されている間でもタイマは動作を継続しており、位置検出機能は停止せず、かつ出力データバッファレジスタ (OPDBR0 ~ OPDBRB) から出力データレジスタ (OPDRH, OPDRL) へのデータ転送は継続され、波形が生成されます。しかし、OPT5 ~ OPT0 端子へ波形は出力されません。

DTTI 回路のブロックダイアグラムを図 15.6-25 に示し、タイマ状態制御レジスタ (TCSR) の D1, D0:bit9, bit8 に "00<sub>B</sub>" が設定されているときの DTTI 回路のタイミングダイアグラムを図 15.6-26 に示します。

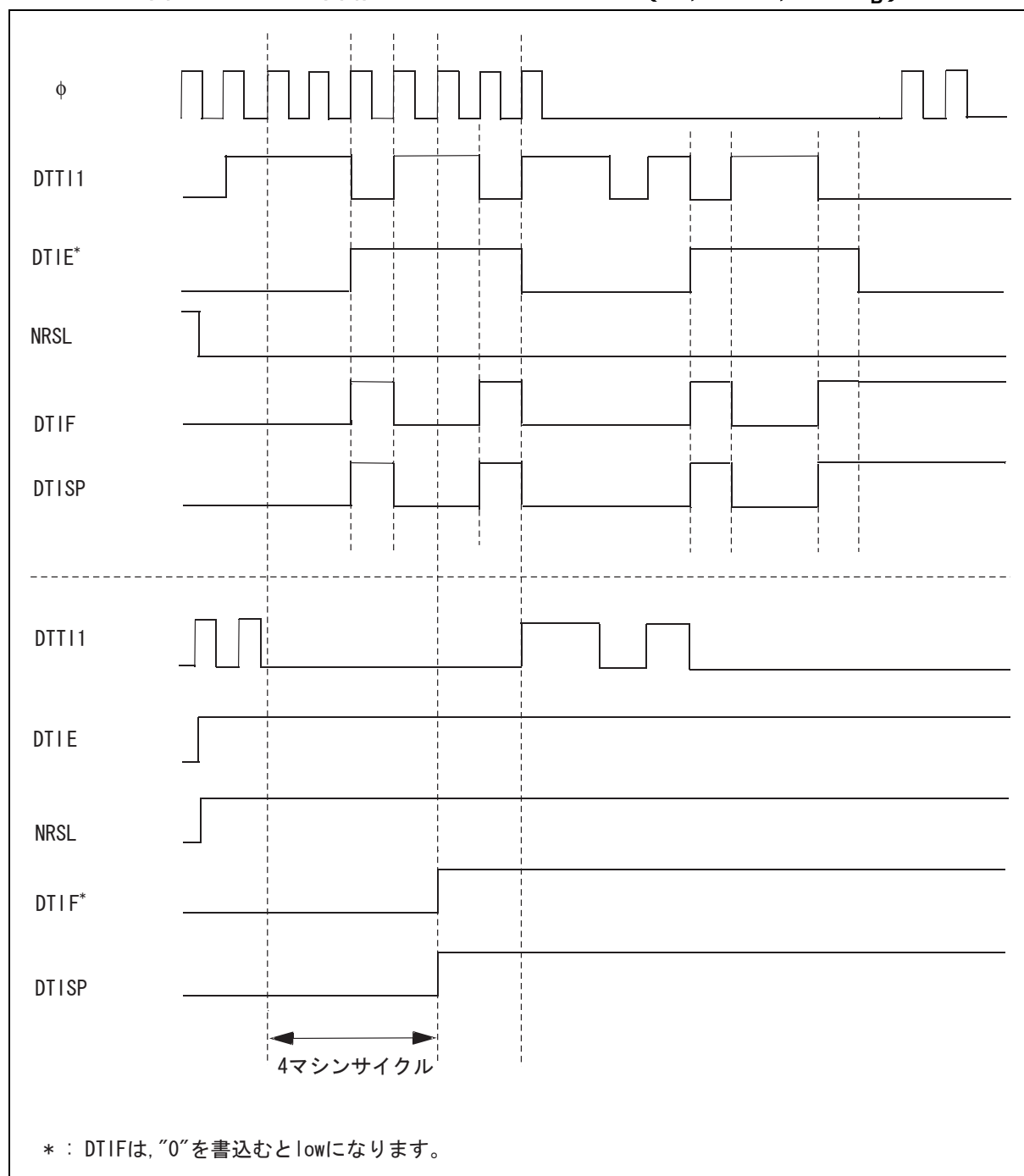
### ● DTTI 回路のブロックダイアグラム

図 15.6-25 DTTI 回路のブロックダイアグラム



● DTTI 回路のタイミングダイアグラム (D1, D0:bit9, bit8=00<sub>B</sub>)

図 15.6-26 DTTI 回路のタイミングダイアグラム (D1, D0:bit9, bit8=00<sub>B</sub>)



< 注意事項 > 認識された状態の DTTI1 (ノイズ除去後) から DTISP への時間が 2 サイクル必要とするのはワーストなケースであり, 1 サイクルを必要とするのがベストなケースです。

## ■ DTTI と OPTx 出力の関係

表 15.6-4 DTTI と OPTx 出力の関係

NRSL	DTIE	DTTI1	機 能
X	0	X	DTTI は OPTx に影響を与えません（初期値）。
0	1	0	DTTI が有効になります。ノイズフィルタは有効になりません。DTTI 端子の "L" は、PDRx が設定されているインアクティブレベルの出力をトリガとします。DTTI 割込みが生成されます。
0	1	1	DTTI は OPTx に影響を与えません。
1	1	0	DTTI が有効になります。ノイズフィルタは有効になります。DTTI 端子の "L" は、PDRx が設定されているインアクティブレベルの出力をトリガとします。DTTI 割込みが生成されます。
1	1	1	DTTI は OPTx に影響を与えません。



## 15.6.6 ノイズ除去機能の動作

---

**SNIx 端子および DTTI 端子のノイズ除去機能について説明します。**

---

### ■ ノイズ除去機能の動作

#### ● DTTI 端子のノイズ除去機能

出力制御レジスタ上位 (OPCUR) の NRSL:bit13 に "1" を設定すると, DTTI 端子入力のノイズ除去機能を使用できます。ノイズ除去機能が起動すると, ノイズ除去回路は, 出力端子をインアクティブレベルに固定するタイミングを約 4, 8, 16 または 32 マシンサイクルだけ遅延させます。ノイズ除去回路は内部クロックを使用するので, DTTI 入力が有効になった場合でも, 入力発振が停止するモード (ストップモードなど) 時には無効になります。

#### ● SNI2 ~ SNI0 端子のノイズ除去機能

入力制御レジスタ下位 (IPCLR) の SNC2 ~ SNC0:bit5 ~ bit3 に "1" を設定すると, SNI2 ~ SNI0 端子入力のノイズ除去機能を使用できます。ノイズ除去機能が選択されると, ノイズ除去回路は入力を約 4 マシンサイクルだけ遅延させます。ノイズ除去回路は内部クロックを使用するので, SNIx 入力が有効になった場合でも, 入力発振が停止するモード (ストップモードなど) 時には無効になります。

#### ● 設定可能なノイズ除去回路

除去されるノイズのパルス幅は, 4, 8, 16 または 32 マシンサイクル未満になるように設定することが可能です。すなわち, 16 MHz マシンクロック時は, 回路は 0.25ms ~ 2 ms の幅のパルスをフィルタリングできます。SNIx 端子と DTTI 端子のノイズ除去回路を設定するための制御は分離されています。ノイズキャンセル制御レジスタを図 15.4-12 に示します。

## 15.6.7 16 ビットタイマの動作

16 ビットタイマはバッファ機能とコンペアクリア機能を持っており、モータ速度検査タイマや異常検出タイマとして使用します。16 ビットタイマは、リセットが完了しカウント有効ビットが設定された後、カウンタ値 "0000<sub>H</sub>" からカウントアップを開始します。

### ■ 16 ビットタイマの動作

カウンタ値は、以下のイベントが発生した場合にクリアされます。

- オーバフローが発生した場合。
- コンペアクリアレジスタ (CPCR<sub>H</sub>, CPCRL) と一致した場合。
- 動作中にタイマ状態制御レジスタ (TCSR) の TCLR:bit7 に "1" が設定された場合。
- 書込みタイミング信号が生成され、タイマ状態制御レジスタ (TCSR) の MODE:bit6 に "0" が設定された場合。
- 位置検出信号が生成され、タイマ状態制御レジスタ (TCSR) の MODE:bit6 に "1" が設定された場合。
- リセットされた場合。

割込みは、コンペアクリアレジスタとの一致でカウンタがクリアされると、生成されません。

< 注意事項 > コンペアクリアレジスタやタイマバッファレジスタへアクセスする場合は、ワードアクセス命令をご使用ください。

図 15.6-27 オーバフローによるカウンタクリア

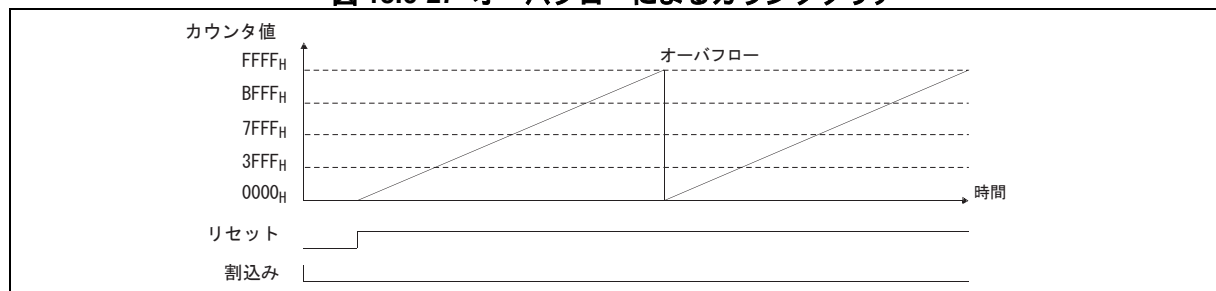
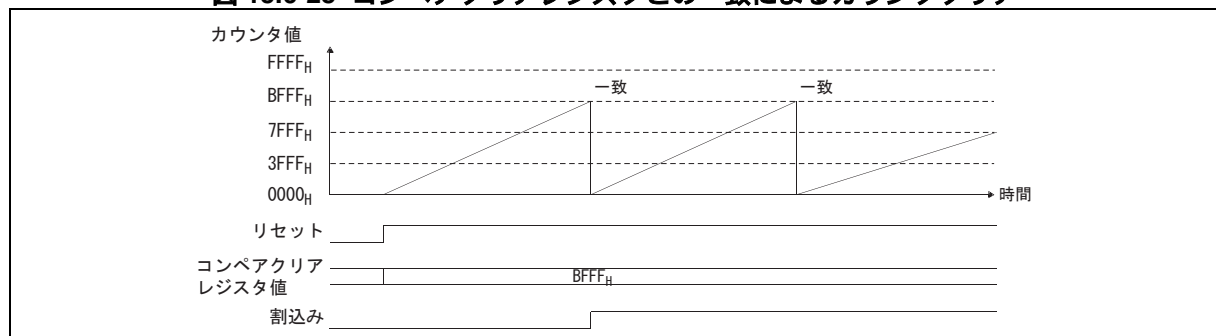


図 15.6-28 コンペアクリアレジスタとの一致によるカウンタクリア

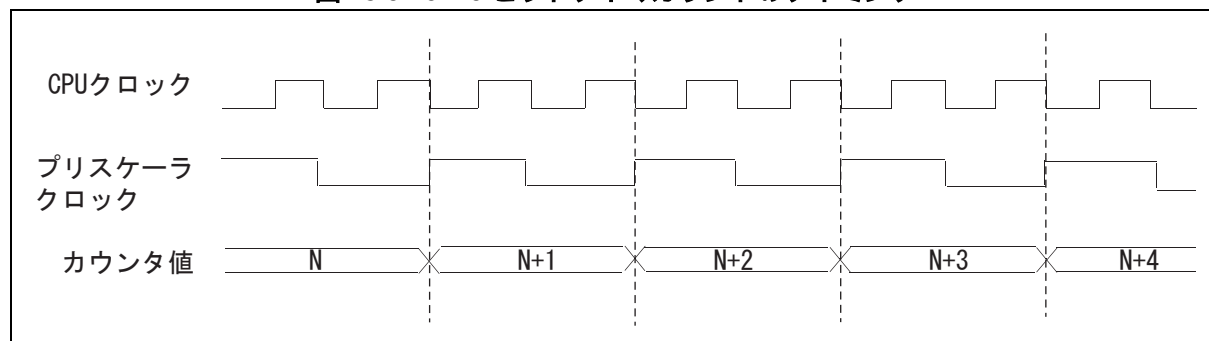


## ■ 16 ビットタイマのタイミング

16ビットタイマは、プリスケールクロックに基づいてインクリメントされ、立上りエッジでカウントアップします。

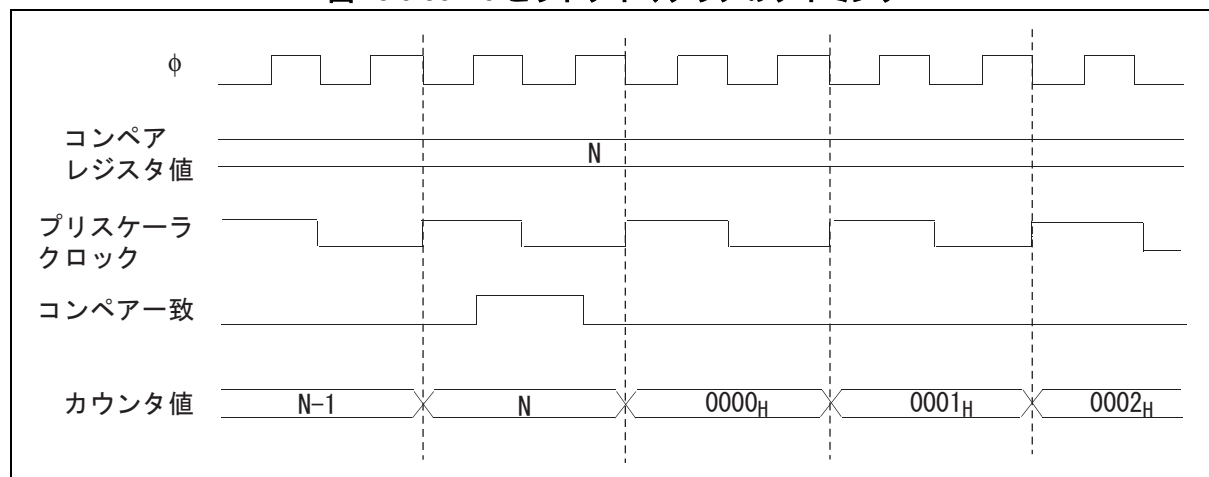
< 注意事項 > プリスケールクロックが変化する前に、タイマ状態制御レジスタ (TCSR) の TMEN:bit3 に "0" を設定してタイマカウンタを無効にしなければなりません。

図 15.6-29 16 ビットタイマカウントのタイミング



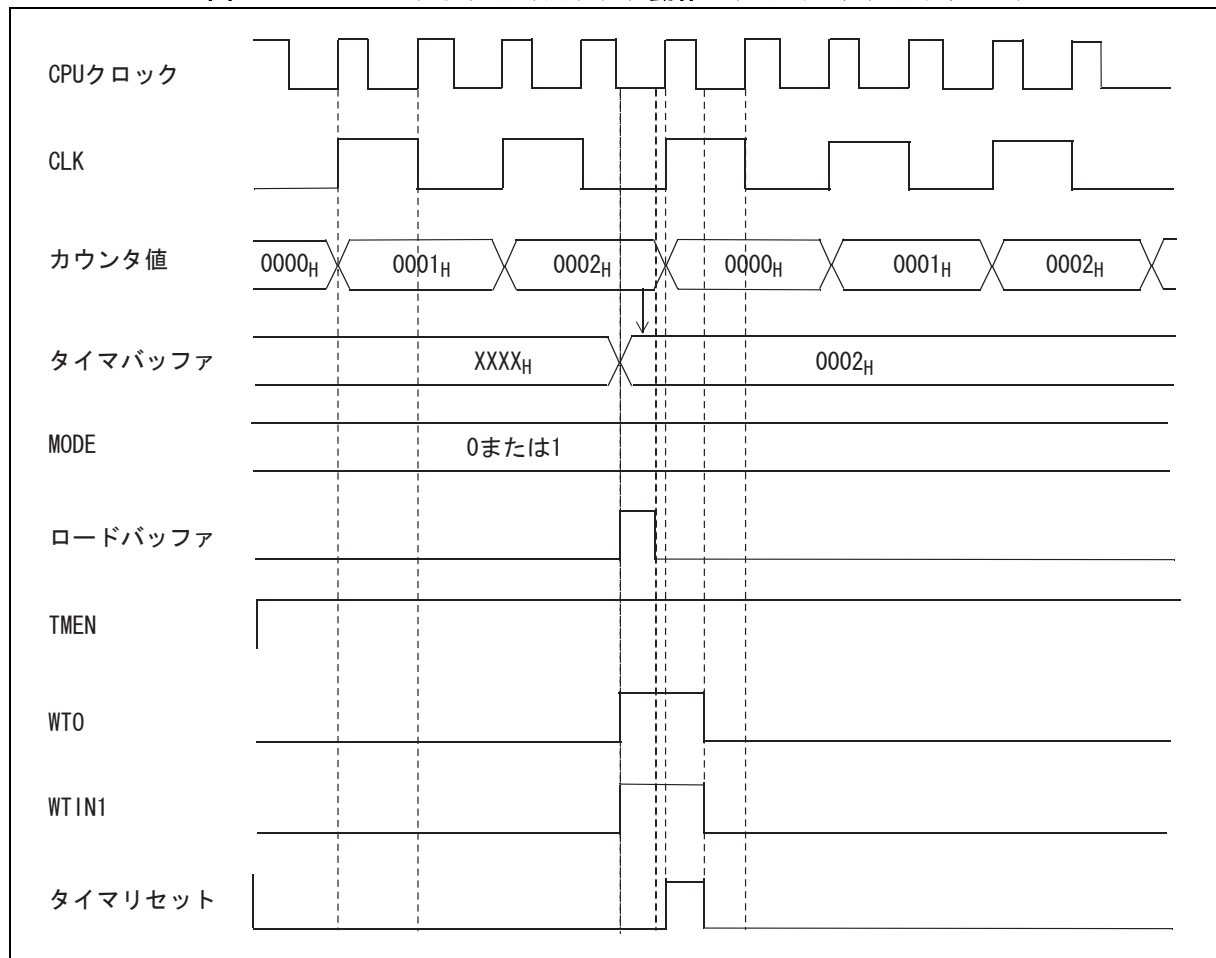
カウンタは、リセットが発生した場合、ソフトウェアクリア (TCLR) が発生した場合、コンペアクリアレジスタとの一致が発生した場合または書き込みタイミング信号か位置検出信号が生成された場合、直ちにクリアできます。カウンタは、リセットが発生した場合は直ちにクリアされ、コンペアクリアレジスタとの一致が発生した場合、ソフトウェアクリア (TCLR) が発生した場合または書き込みタイミング信号か位置検出信号が生成された場合は、カウントタイミングと同期してクリアされます。

図 15.6-30 16 ビットタイマクリアのタイミング



## ■ 16 ビットタイマバッファ動作のタイミングダイアグラム

図 15.6-31 16 ビットタイマバッファ動作のタイミングダイアグラム



## ■ マルチパルスジェネレータの 16 ビットタイマの使用

16 ビットタイマは、書込みタイミング割込みフラグまたは位置検出割込みフラグが設定されるとリセットされます（どちらを設定するかは、タイマ状態制御レジスタ（TCSR）の MODE:bit6 で選択可能）。

16 ビットタイマは、タイマ状態制御レジスタ（TCSR）の TMEN:bit3 で起動または停止できます。タイマオーバーフロー割込みは存在しません。16 ビットタイマは、再起動されるたびに現カウンタ値がバッファにラッチされ、速度が計算されます。

カウンタ値がコンペアクリアレジスタ（CPCR<sub>H</sub>, CPCR<sub>L</sub>）と一致すると、タイマはリセットされます。

---

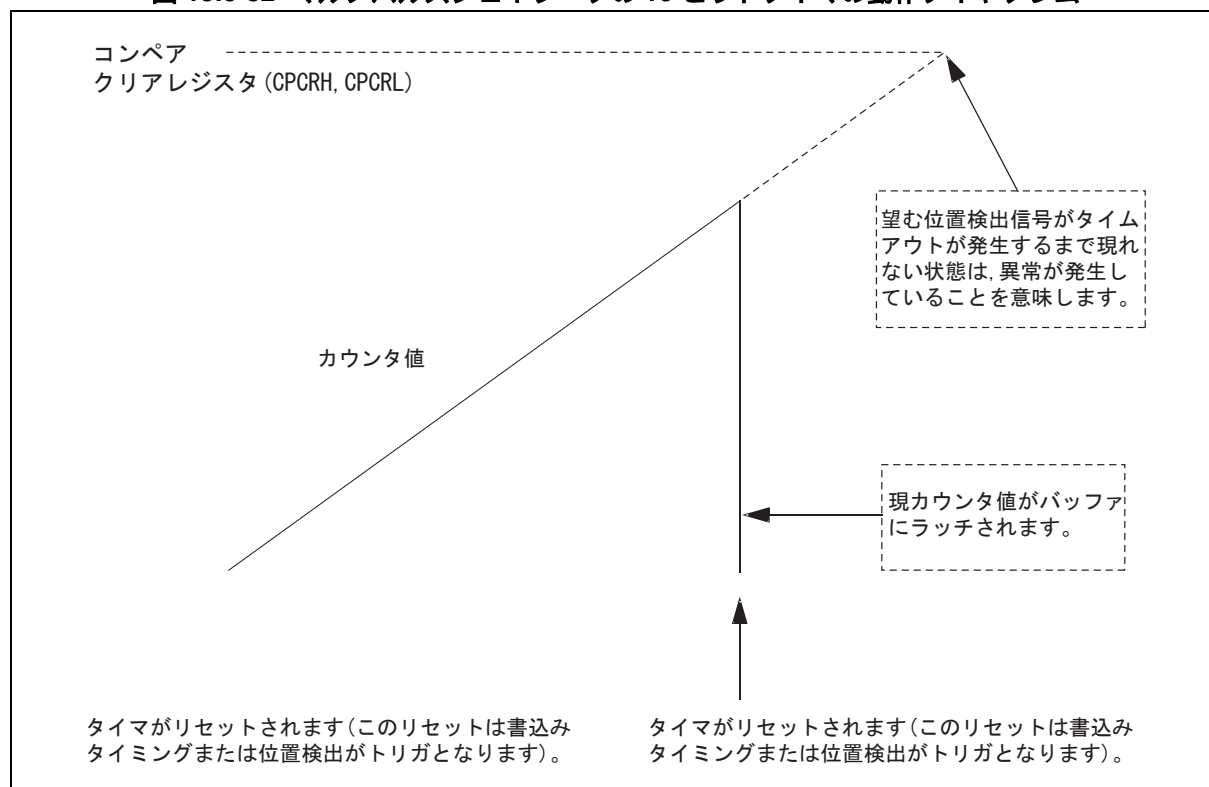
< 注意事項 > コンペアクリアレジスタ（CPCR<sub>H</sub>, CPCR<sub>L</sub>）にタイマカウンタ値と同じ値がロードされると、比較動作は次回これと同じイベントが発生するまで行われません。

---

コンペアクリア割込みは同じ割込みベクタを書込みタイミング割込みと共用しますが、コンペア一致割込みは同じ割込みベクタを位置検出割込みと共用します。

## ■ マルチパルスジェネレータの 16 ビットタイマの動作ダイアグラム

図 15.6-32 マルチパルスジェネレータの 16 ビットタイマの動作ダイアグラム



## 15.7 マルチパルスジェネレータ使用上の注意

マルチパルスジェネレータ使用上の注意を以下に示します。

### ■ 波形シーケンサ使用上の注意

#### ● 設定プログラム使用上の注意

- ある PPG 同期モードから別の PPG 同期モードへの切換え(たとえば、立上りエッジ同期 (IPCUR レジスタの WTS1, WTS0:bit15, bit14=01<sub>B</sub>) から立下りエッジ同期 (IPCUR レジスタの WTS1, WTS0:bit15, bit14=10<sub>B</sub>) への切換えまたはその逆への切換え)は、禁止されています。同期モード (IPCUR レジスタの WTS1, WTS0:bit15, bit14=00<sub>B</sub>) は、このような切換えにおける通過点であってはなりません。
- データ転送方式を変更すると、選択される次のデータバッファレジスタは、必ず出力データレジスタ上位 (OPDRH) の BNKF, RDA2 ~ RDA0:bit15 ~ bit12 で指定されます。これは、OPDBR0 書込み方式 (OPCUR レジスタの OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>) には適用されません。OPDBR0 書込み方式では、BNKF, RDA2 ~ RDA0:bit12 ~ bit10 は無視されます。
- 出力データレジスタ (OPDRH, OPDRL) へアクセスする際には、ワードアクセスをご使用ください。
- データ転送に OPDBR0 書込み方式 (OPCUR レジスタの OPS2 ~ OPS0:bit12 ~ bit10=000<sub>B</sub>) を使用する場合、出力データバッファレジスタ 0 へのアクセスは、ワードアクセスをご使用ください。出力データバッファレジスタ (下位レジスタまたは上位レジスタ) へバイトアクセスしても、転送動作は開始しません。
- 16 ビットリロードタイマアンダフロー転送方式 (OPCUR レジスタの OPS2 ~ OPS0:bit12 ~ bit10=010<sub>B</sub>) を使用するには、16 ビットリロードタイマは "リロードモード" で使用しなければなりません。16 ビットリロードタイマを起動するには、ソフトウェアトリガを使用する必要があります。16 ビットリロードタイマは、更新時間の事前設定および連続制御動作を実行するために必要となります。
- "位置検出と 16 ビットリロードタイマアンダフロー" 方式 (OPCUR レジスタの OPS2 ~ OPS0:bit12 ~ bit10=011<sub>B</sub> または 111<sub>B</sub>) を使用するには、16 ビットリロードタイマは "ワンショットモード" でなければなりません。TIN0 は、2 マシンサイクルより長くなければなりません。
- DTTI1 入力制御回路が有効になる (OPCUR レジスタの DTIE:bit15=1) 前に、OPTx と多重化されている PORTx がポート方向レジスタ (DDR<sub>x</sub>) で出力ポートとして設定されていることを確認してください。
- DTTI1 入力制御回路は内部クロックを使用するので、入力は無効となります。発振が停止するストップモードなどで DTTI1 入力が無効になった (OPCUR レジスタの DTIE:bit15=1) 場合でも、無効となります。
- ワーストの場合は、DTTI1 信号が認識されてから DTISP 信号が無効になるまでに 2 マシンサイクルを要します。ベストの場合は、1 マシンサイクルを要します。
- ノイズ除去機能が無効になっている (OPCUR レジスタの NRSL:bit13=0) 場合は、必ずノイズキャンセル制御レジスタ (NCCR) の D1, D0:bit7, bit8 を書換えてください。

- ノイズ除去機能が無効になっている( IPCLRレジスタのSNC2 ~ SNC0:bit5 ~ bit3=000<sub>B</sub> )  
場合は、必ずノイズキャンセル制御レジスタ ( NCCR ) の S21, S20, S11, S10, S01, S00:  
bit15 ~ bit10 を書換えてください。

● 割込みに関する注意

- 出力制御レジスタ上位 ( OPCUR ) の DTIF:bit14 が "1" に設定されたままになっていると、割込み処理から復帰できません。DTIF:bit14 は必ずクリアしてください。
- 出力制御レジスタ上位 ( OPCUR ) の WTIF:bit14 が "1" に設定されたままになっていると、割込み処理から復帰できません。WTIF:bit14 は必ずクリアしてください。
- 出力制御レジスタ下位 ( OPCLR ) の PDIF:bit7 が "1" に設定されたままになっていると、割込み処理から復帰できません。PDIF:bit7 は必ずクリアしてください。
- 入力制御上位レジスタ ( IPCUR ) の CPIF:bit13 が "1" に設定されたままになっていると、割込み処理から復帰できません。CPIF:bit13 は必ずクリアしてください。
- 上記割込みは割込みベクタをほかのリソースと共用するので、割込みが使用されている場合は、割込み要因を割込み処理ルーチンで注意深く確認しなければなりません。
- また、これらの割込みで EI<sup>2</sup>OS を使用している場合は、共用されるリソース割込みは禁止しなければなりません。

## ■ 16 ビットタイマ使用上の注意

● 設定プログラム使用上の注意

- コンペアクリアレジスタ( CPRRH, CPCRL )およびタイマバッファレジスタ( TMBRH, TMBRL ) へアクセスする際には、ワードアクセスをご使用ください。
- プリスケラクロックを切り換える場合は、タイマ状態制御レジスタ( TCSR )のTMEN: bit3 に "0" を設定することによって前もってタイマカウンタを無効にしておかなければなりません。タイマ制御状態レジスタの CLK2 ~ CLK0:bit2 ~ bit6 の書換えは、タイマがカウントしていない間に行ってください。
- コンペアクリアレジスタ ( CPRRH, CPCRL ) にタイマカウンタ値と同じ値がロードされた場合は、比較動作は次の同じタイマカウンタ値まで行われません。

● 割込みに関する注意

- タイマ制御状態レジスタ ( TCSR ) の ICLR:bit5 に "1" が設定されており、かつ割込み要求が許可になっている ( TCSR レジスタの ICRE:bit4=1 ) と、制御は割込み処理から復帰できません。ICLR ビットは必ずクリアしてください。
- 16 ビットタイマは割込みベクタをほかのリソースと共用するので、割込みが使用されている場合は、割込み要因を割込み処理ルーチンで注意深く確認しなければなりません。
- また、16 ビットタイマで EI<sup>2</sup>OS を使用している場合は、共用されるリソース割込みは禁止にしなければなりません。

## 15.8 マルチパルスジェネレータのプログラム例

マルチパルスジェネレータのプログラム例を示します。

### ■ マルチパルスジェネレータのプログラム例

#### ● 処理

- ・ タイミング割込みが生成されたとき、PPG の出力は OPT0 へ、PPG の反転割込みは OPT1 へ指示されます。
- ・ OPDBR0 書込みモードは、アウトプットデータレジスタ (OPDRH, OPDRL) へデータを転送するために使用されます。
- ・ 16 ビット PPG タイマは PWM において使用され、かつソフトウェアトリガにより開始します。
- ・ EI<sup>2</sup>OS は使用しません。
- ・ 16MHz はマシクロック用、62.5ns は 16 ビット PPG タイマのカウントクロック用です。

#### ● コーディング用

```

ICR07 EQU 0000B7H ; 波形シーケンサ用割込み制御レジスタ
PCSR1 EQU 000042H ; PPG 周期設定レジスタ
PDUT1 EQU 000044H ; PPG デューティ設定レジスタ
PCNT1 EQU 000046H ; PPG 制御状態レジスタ
OPCLR EQU 00008AH ; 出力制御レジスタ
OPDBR0 EQU 003FE0H ; 出力データバッファレジスタ
WTIF EQU OPCR:9 ; 割込み要求フラグビット
;----- メインプログラム -----
CODE CSEG
START
; : ; スタックポインタ (SP) が既に初期化されているものと仮定
AND CCR, #0BFH ; 割込み禁止
MOV I:ICR01, #00H ; 割込みレベル 0 (最強)
MOVW I:PCSR0, #0064H ; PPG 出力の周期を設定
MOVW I:PDUT0, #003CH ; PPG 出力のデューティ比の設定
MOVW I:PCNT0, #01100000000000110B
; 正常極性における PPG 出力を許可
; 16 ビット PPG タイマを有効, 62.5ns クロック
; ソフトウェアトリガ PPG
; PWM モードを選択
; 割込みフラグをクリア, カウンタ開始
MOVW I:OPCR, #0103H ; OPT0 と OPT1 出力を許可
; データ転送用 OPDBR0 書込みモードを選択
; 書込みタイミング割込みを許可
; 割込みフラグをクリア
MOVW I:OPDBR0, #0009H ; PPG 出力の OPT0 端子を設定
; 反転 PPG 出力の OPT1 を設定
; データ転送開始
MOV ILM, #07H ; PS の ILM をレベル 7 に設定
OR CCR, #40H ; 割込み許可

```



## 第 15 章 マルチパルスジェネレータ

```
LOOP:  MOV    A,#00H    ; 無限ループ
        MOV    A,#01H    ;
        BRA    LOOP      ;
;----- 割込みプログラム -----
WARI:
        CLRB   I:WTIF    ; 割込み要求フラグをクリア
;      :
;   ユーザ処理
;      :
        RETI            ; 割込みから復帰

CODE    ENDS
;----- ベクタ設定 -----
VECT    CSEG   ABS=0FFH
        ORG    0FF94H    ; 割込み #26 (1AH) 用ベクタをクリア
        DSL    WARI
        ORG    0FFDCH    ; リセットベクタを設定
        DSL    START
        DB     00H        ; シングルチップモードを設定
VECT    ENDS
        END    START
```

# 第16章

---

## PWC タイマ

PWC タイマの機能と動作について説明します  
( MB90467 では , PWC タイマ ch.0 未搭載となります )

- 16.1 PWC タイマの概要
- 16.2 PWC タイマのブロックダイアグラム
- 16.3 PWC タイマの端子
- 16.4 PWC タイマのレジスタ
- 16.5 PWC タイマの割込み
- 16.6 PWC タイマの動作
- 16.7 PWC タイマ使用上の注意
- 16.8 PWC タイマのプログラム例

## 16.1 PWC タイマの概要

---

PWC（パルス幅測定）タイマはリロード機能を備えた多機能 16 ビットアップカウンタであり，入力信号のパルス幅を算出する機能も備えています。

PWC タイマは，16 ビットカウンタ，入力パルス分周器，分周比制御レジスタ，カウント入力端子，パルス出力端子および 16 ビット制御レジスタから構成されています。

---

### ■ PWC タイマの特性

MB90460/465 シリーズは 2 つの PWC タイマチャンネルを装備しており，PWC タイマは以下の特性を備えています。

#### ● タイマ機能

- 指定された時間間隔で割込み要求が生成されます。
- タイマ周期と同期したパルス信号を生成できます。
- 3 つの内部クロックからカウンタクロックを選択できます。

#### ● パルス幅測定機能

- 外部パルス入力イベント間の時間を測定できます。
- 3 つの内部クロックからカウンタクロックを選択できます。
- カウントモード
- H パルス幅（立上りエッジから立下りエッジまでの幅），L パルス幅（立下りエッジから立上りエッジまでの幅）
- 立上りエッジ周期（立上りエッジから立上りエッジまでの周期），立下りエッジ周期（立下りエッジから立下りエッジまでの周期）
- 中間エッジカウント（立上りエッジから立下りエッジまでのカウントまたは立下りエッジから立上りエッジまでのカウント）
- 6 ビット入力分周器を使用して入力パルスを 22, 24, 26 または 28 で分周して周期を測定します。
- カウント完了時に割込み要求が生成されます。
- 単一カウントまたは連続カウントを選択できます。

### ■ PWC タイマの動作

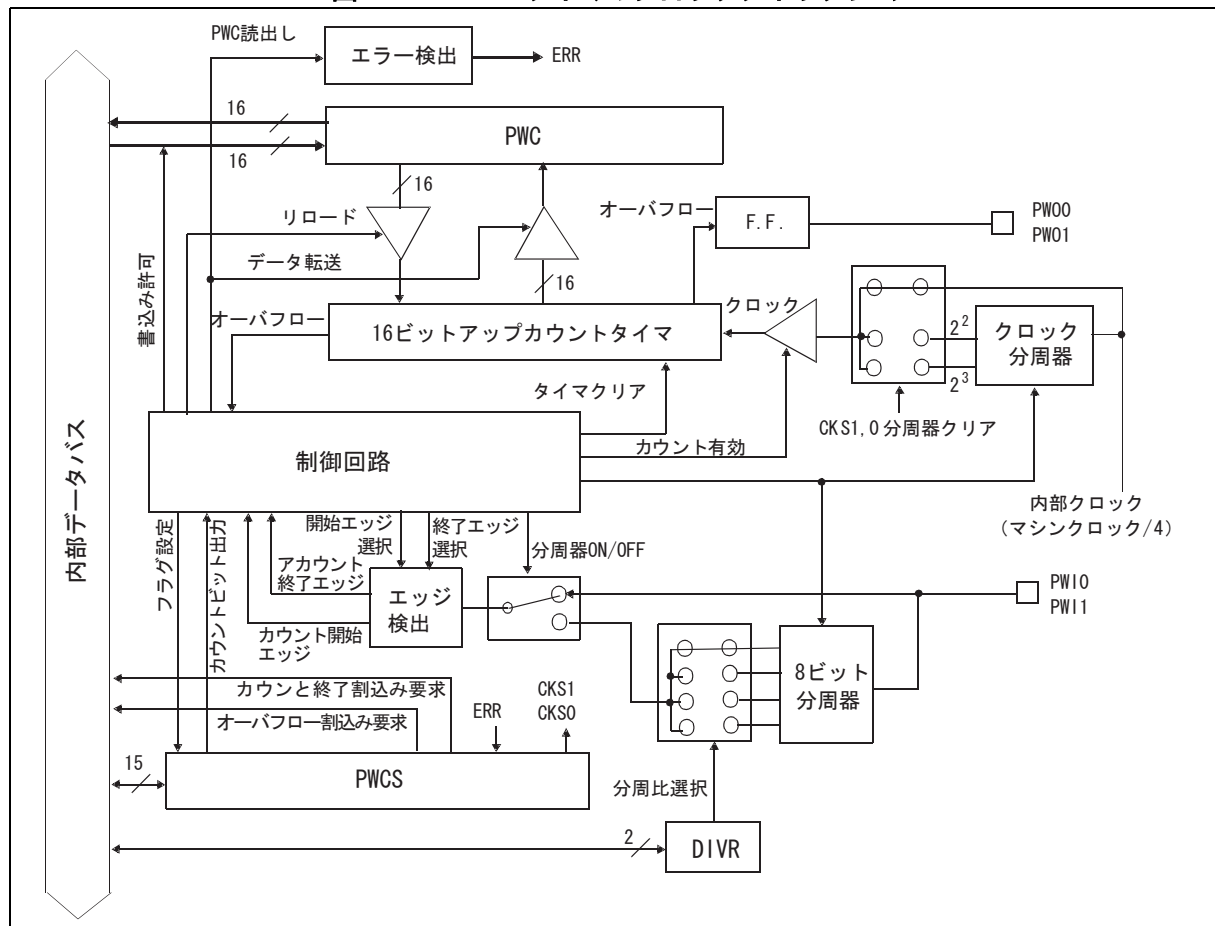
PWC タイマは，16 ビットアップカウントタイマを中核とした多機能タイマであり，カウント入力端子と 8 ビット入力分周器を備えています。PWC タイマは 2 つの主要機能（タイマ機能とパルス幅測定機能）を備えており，これらの機能はそれぞれ 2 種類のカウンタクロックを選択できます。

## 16.2 PWC タイマのブロックダイアグラム

PWC タイマのブロックダイアグラムを示します。

### ■ PWC タイマのブロックダイアグラム

図 16.2-1 PWC タイマのブロックダイアグラム



## 16.3 PWC タイマの端子

PWC タイマの端子について説明します。また端子のブロックダイアグラムも示します。

### ■ PWC タイマの端子

PWC タイマの端子は、汎用入出力ポートと共用されます。端子機能と I/O 形式および PWC タイマを使用するために必要な設定を表 16.3-1 に示します。

表 16.3-1 16 ビット PWC タイマの端子

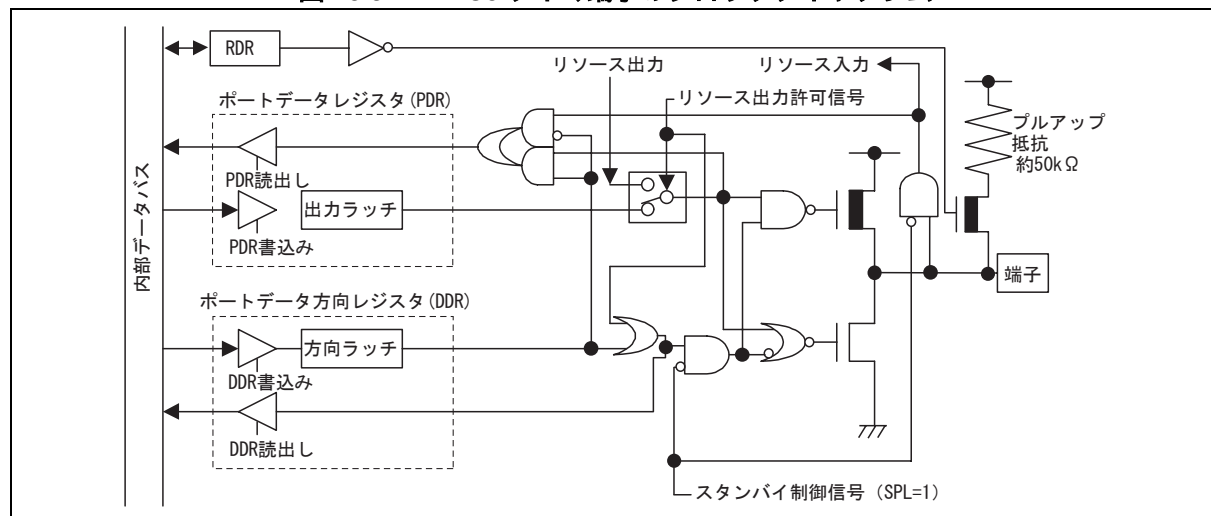
端子名	端子機能	I/O 形式	プルアップ オプション	待機制御	設 定
P06/PWI0*	ポート 0 入出力， タイマ入力	CMOS 出力， CMOS ヒステ リシス入力	選択可能	使用可能	入力ポートの設定 ( DDR0:bit6=0 )
P07/PWO0*	ポート 0 入出力， タイマ出力				タイマ有効の設定 ( PWCSL0:MOD2 ~ MOD0 が "0" ではな い)
P22/PWI1	ポート 2 入出力， タイマ入力		提供なし		入力ポートの設定 ( DDR2:bit2=0 )
P23/PWO1	ポート 2 入出力， タイマ出力				タイマ有効の設定 ( PWCSL1:MOD2 ~ MOD0 が "0" ではな い)

\*:MB90467 では、リソースが搭載されていないため、削除となります。

### ■ PWC0 タイマ端子のブロックダイアグラム

PWC0 タイマ端子のブロックダイアグラムを図 16.3-1 に示します。

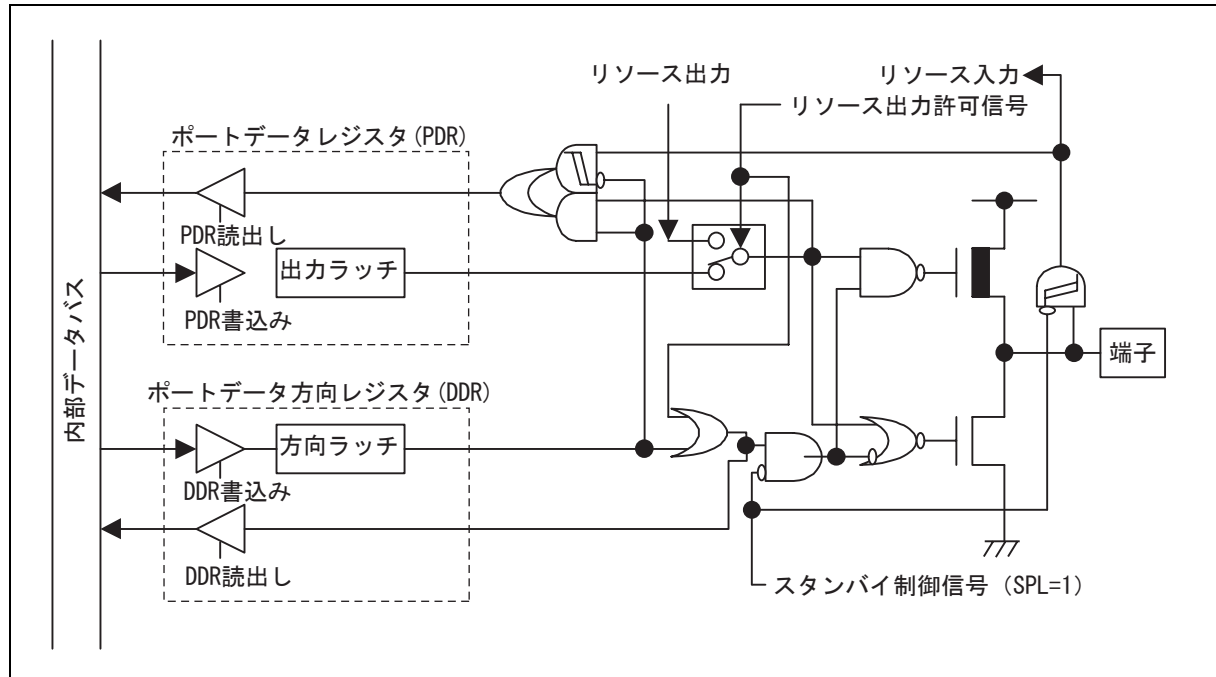
図 16.3-1 PWC0 タイマ端子のブロックダイアグラム



## ■ PWC1 タイマ端子のブロックダイヤグラム

PWC1 タイマ端子のブロックダイヤグラムを図 16.3-2 に示します。

図 16.3-2 PWC1 タイマ端子のブロックダイヤグラム



## 16.4 PWC タイマのレジスタ

PWC タイマのレジスタを以下に示します。

### ■ PWC タイマのレジスタ

図 16.4-1 PWC タイマのレジスタ

PWC状態制御レジスタ (上位)										
	bit	15	14	13	12	11	10	9	8	
		STRT	STOP	EDIR	EDIE	OVIR	OVIE	ERR	POUT	PWCSH0, PWCSH1
リード/ライト	⇒	R/W	R/W	R	R/W	R/W	R/W	R	R/W	
初期値	⇒	0	0	0	0	0	0	0	0	
PWC状態制御レジスタ (下位)										
	bit	7	6	5	4	3	2	1	0	
		CKS1	CKS0	未使用	未使用	S/C	MOD2	MOD1	MOD0	PWCSL0, PWCSL1
リード/ライト	⇒	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初期値	⇒	0	0	0	0	0	0	0	0	
PWCデータバッファレジスタ (上位)										
	bit	15	14	13	12	11	10	9	8	
		PW15	PW14	PW13	PW12	PW11	PW10	PW09	PW08	PWCH0, PWCH1
リード/ライト	⇒	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	
PWCデータバッファレジスタ (下位)										
	bit	7	6	5	4	3	2	1	0	
		PW07	PW06	PW05	PW04	PW03	PW02	PW01	PW00	PWCL0, PWCL1
リード/ライト	⇒	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	
分周比制御レジスタ										
	bit	7	6	5	4	3	2	1	0	
		—	—	—	—	—	—	DIV1	DIV0	DIV0, DIV1
リード/ライト	⇒	—	—	—	—	—	—	R/W	R/W	
初期値	⇒	—	—	—	—	—	—	0	0	

## 16.4.1 PWC 状態制御レジスタ ( PWCSH0, PWCSH1, PWCSL0, PWCSL1 )

PWC 状態制御レジスタ ( PWCSH0, PWCSH1, PWCSL0, PWCSL1 ) は, PWC タイマの動作を制御し, PWC タイマの状態を読み出します。

### ■ PWC 状態制御レジスタ, 上位バイト ( PWCSH0, PWCSH1 )

図 16.4-2 PWC 状態制御レジスタ, 上位バイト ( PWCSH0, PWCSH1 )

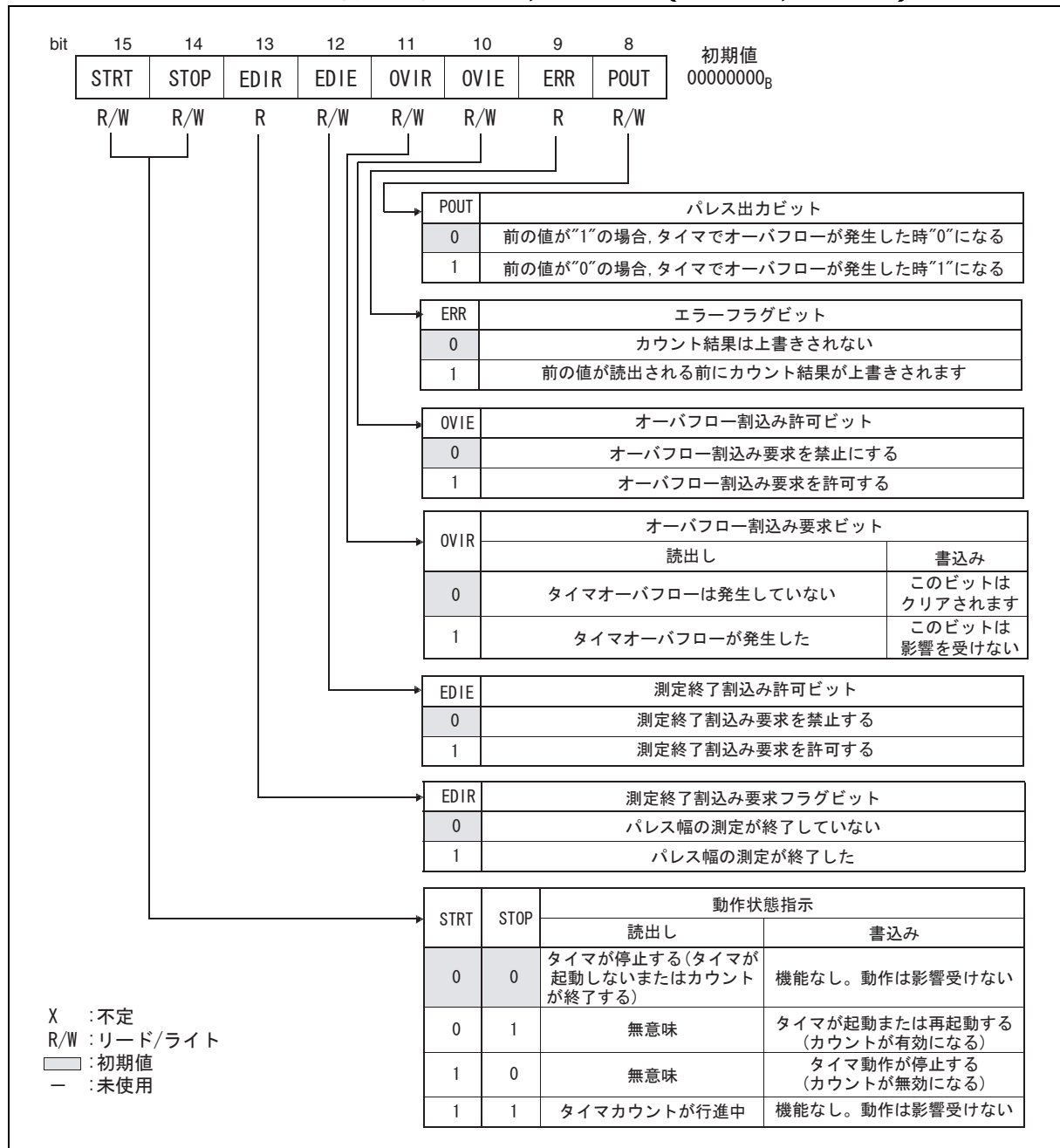




表 16.4-1 PWC 状態制御レジスタ, 上位バイト (PWCSH0, PWCSH1)( 1 / 2 )

ビット名		機 能
bit15, bit14	STRT, STOP: 起動 / 停止 ビット	<ul style="list-style-type: none"> <li>これらのビットは, 16 ビットアップカウントタイマを起動, 再起動および停止するために使用します。</li> <li>これらのビットを読み出すと, タイマ動作状態がわかります。</li> <li>これらのビットは, 読み / 書き可能です。これらのビットの意味は, これらのビットを読み出すか書き込むかに依存します。</li> <li>リードモディファイライト動作時は, 必ず "11<sub>B</sub>" が読み出されます。</li> <li>STRT, STOP ビットを書き込んでタイマを起動, 停止する場合, ビット操作命令 (ビットクリア命令など) を使用できます。ただし, 動作状態 (タイマが動作中であるといったことなどを必ず示す) の読出しに, ビット操作命令は使用できません。</li> </ul>
bit13	EDIR: 測定終了割込み要求フラグ ビット	<ul style="list-style-type: none"> <li>このビットは, パルス幅測定モードによる測定が終了したことを示します。</li> <li>パルス幅測定が終了すると, このビットは設定されます (PWC0, PWC1 には測定結果が入ります)。</li> <li>PWC データバッファレジスタ内の測定結果や PWC0, PWC1 を読み出したとき, このビットは自動的にクリアされます。</li> <li>タイマモード時はこのビットは意味を持ちません。</li> <li>このビットは読出し専用ビットであり, 書き込んでも意味がありません。</li> </ul>
bit12	EDIE: 測定終了割込み許可ビット	<ul style="list-style-type: none"> <li>このビットは, パルス幅カウントモード時の測定終了割込み要求を制御するために使用します。</li> <li>このビットが "1" で, かつ EDIR ビットに "1" が設定されたとき, 測定終了割込み要求が CPU に対して生成されます。</li> <li>タイマモード時はこのビットには必ず "0" が設定されます。</li> </ul>
bit11	OVIR: オーバーフロー割込み要求 ビット	<ul style="list-style-type: none"> <li>このビットは, 16 ビットアップカウントタイマのオーバーフローを発生させるタイミングを指定するために使用します。動作はすべてのモードに影響を与えます。</li> <li>タイマオーバーフローが ( "FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲において ) 発生すると, このビットは設定されます。</li> <li>このビットに "0" を設定した場合: このビットはクリアされます。</li> <li>このビットに "1" を設定した場合: このビットは影響を受けません。</li> <li>リードモディファイライト動作時は, 必ず "1" が読み出されます。</li> </ul> <p>( 注意事項 )</p> <p>H/L パルス幅カウントモード時は, このビットを使用してパルス幅時間を測定してはいけません。</p>
bit10	OVIE: オーバーフロー割込み要求 ビット	<ul style="list-style-type: none"> <li>このビットは, タイマオーバーフロー割込み要求を許可するために使用します。</li> <li>このビットが "1" で, かつ OVIR に "1" が設定されたとき, 測定終了割込み要求が CPU に対して生成されます。</li> </ul> <p>( 注意事項 )</p> <p>H/L パルス幅カウントモード時は, このビットに "0" を設定してください。</p>

表 16.4-1 PWC 状態制御レジスタ, 上位バイト (PWCSH0, PWCSH1) ( 2 / 2 )

ビット名		機 能
bit9	ERR: エラーフラグ ビット	<ul style="list-style-type: none"> <li>このビットは、パルス幅カウントモードで連続カウントするために使用します。このフラグは、前のカウント結果が PWC レジスタから読み出される前に次のカウント動作が完了していることを示します。この状態が発生すると、PWC レジスタは新しいカウント結果で上書きされ、前のカウント結果は失われます。カウント動作は、このビット値とは無関係に継続します。</li> <li>このビットは読み出し専用ビットであり、このビットへ値を書き込んでも値は変化しません。</li> <li>読み出されていない現カウント結果が次のカウント結果で上書きされると、このビットが設定されます。</li> <li>PWC データバッファレジスタ内の測定結果や PWC0, PWC1 を読み出したとき、このビットは自動的にクリアされます。</li> </ul>
bit8	POUP: パルス出力 ビット	<ul style="list-style-type: none"> <li>タイマモード時 16 ビットアップカウントタイマオーバフローが発生すると、このビットは反転します。</li> <li>パルス幅カウントモード時は、このビットは意味を持ちません。</li> <li>このビットは、読み / 書き可能です。ただし、このビットへの書込みは、タイマが停止した場合に限り可能です。タイマ動作中 (STRT:bit15 と STOP:bit14 の両方に "1" が設定されます) にこのビットに値を書き込んでも、このビット値は変化しません。</li> <li>このビット値が "0" であり、かつタイマオーバフローが "FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲において発生した場合またはタイマが停止しこのビットに "1" が書き込まれた場合、このビットは設定されます。</li> <li>このビット値が "0" であり、かつタイマオーバフローが "FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲において発生した場合またはタイマが停止しこのビットに "0" が書き込まれた場合、このビットはクリアされます。このビットは、リセットでもクリアされます。</li> </ul>

## ■ PWC 状態制御レジスタ, 下位バイト (PWCSL0, PWCSL1)

図 16.4-3 PWC 状態制御レジスタ, 下位バイト (PWCSL0, PWCSL1)

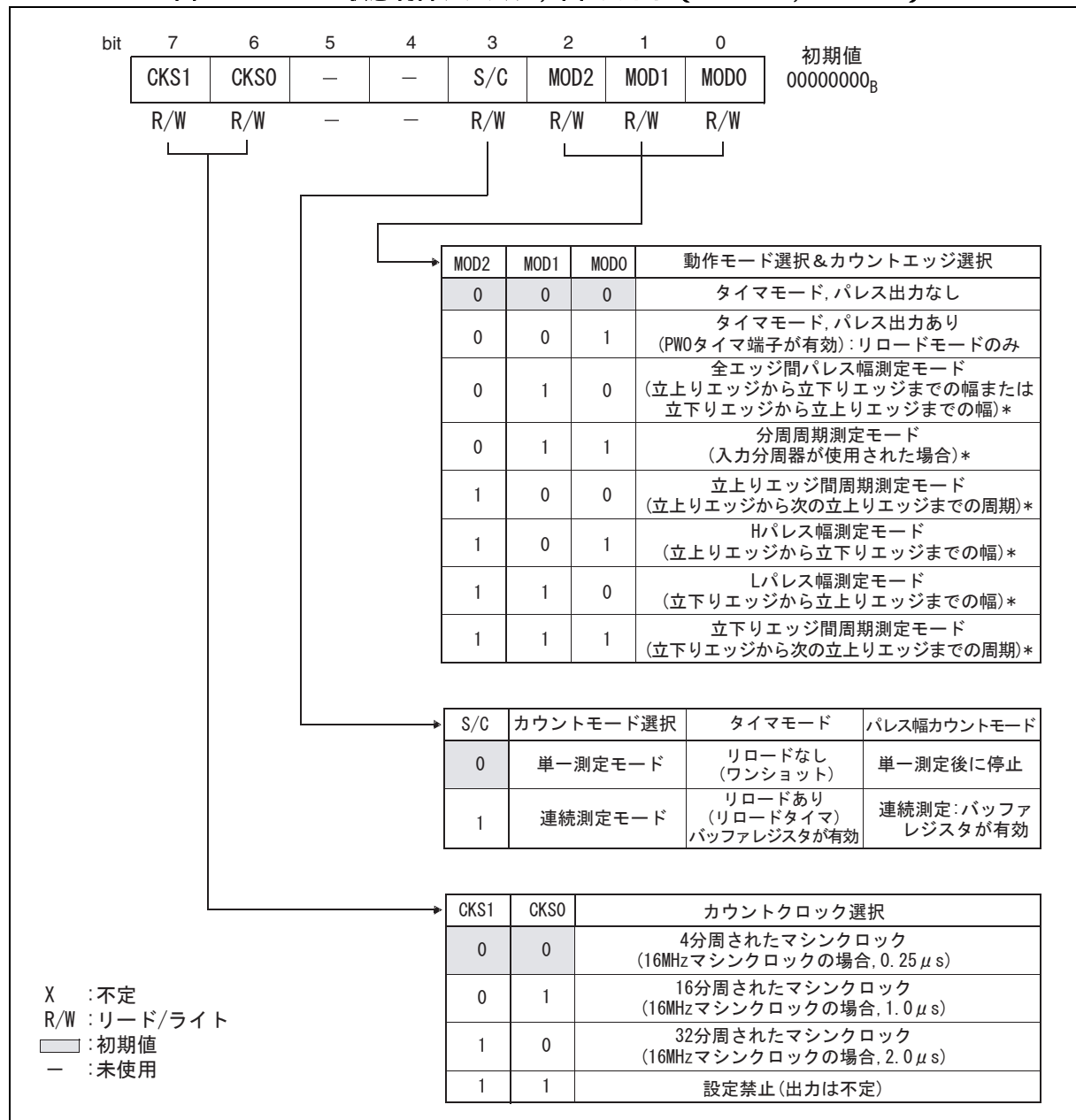


表 16.4-2 PWC 状態制御レジスタ, 下位バイト (PWCSL0, PWCSL1)

ビット名		機 能
bit7, bit6	CLK1, CLK0: クロック選択 ビット	<ul style="list-style-type: none"> <li>CKS1, 0 ビットは, 内部カウントクロックを選択するために使用します。</li> <li>リセットにより, これらのビットは "00<sub>B</sub>" に初期化されます。これらのビットは, 読み / 書き可能です。ただし, "11<sub>B</sub>" は設定できません。</li> </ul> <p>(注意事項)</p> <p>タイマ起動後にこの設定を変更することは禁止されています。これらのビットへの書込みは, タイマ起動前またはタイマ停止後に行ってください。</p>
bit5, bit4	未使用ビット	<ul style="list-style-type: none"> <li>これらのビットは, 未使用ビットです。必ず "00<sub>B</sub>" を書き込んでください。</li> </ul>
bit3	S/C: 単一 / 連続 ビット	<ul style="list-style-type: none"> <li>このビットは, カウントモードを選択するために使用します。</li> <li>リセットにより, このビットは "0" に初期化されます。このビットは, 読み / 書き可能です。</li> </ul> <p>(注意事項)</p> <p>タイマ起動後にこの設定を変更することは禁止されています。このビットへの書込みは, タイマ起動前またはタイマ停止後に行ってください。</p>
bit2 ~ bit0	MOD2, MOD1, MOD0: 動作モード ビット	<ul style="list-style-type: none"> <li>これらのビットに値を設定すると, 動作モードの選択およびパルス幅カウンタに適したパルスエッジの選択が可能になります。</li> <li>リセットにより, これらのビットは "000<sub>B</sub>" に初期化されます。これらのビットは, 読み / 書き可能です。</li> </ul> <p>(注意事項)</p> <p>タイマ起動後にこの設定を変更することは禁止されています。これらのビットへの書込みは, タイマ起動前またはタイマ停止後に行ってください。</p> <ul style="list-style-type: none"> <li>連続測定モードを * マーク (前ページ) が付いた設定に対して設定すると, エッジ総数が算出され, 内部カウントクロックの分周器はカウント終了時にクリアされません。ほかのすべてのモード時は, 内部カウントクロックの分周器はカウント終了時にクリアされます。</li> </ul>

## 16.4.2 PWC データバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1)

PWC データバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1) は, PWC タイマの動作モードに依存する機能を備えています。

### ■ PWC データバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1)

図 16.4-4 PWC データバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1)

図 10-1-10 PWCデータバッファレジスタ (PWCH0, PWCH1, PWCL0, PWCL1)

PWCデータバッファレジスタ (上位)

bit	15	14	13	12	11	10	9	8	
	PW15	PW14	PW13	PW12	PW11	PW10	PW09	PW08	PWCH0, PWCH1
リード/ライト⇒	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	

PWCデータバッファレジスタ (下位)

bit	7	6	5	4	3	2	1	0	
	PW07	PW06	PW05	PW04	PW03	PW02	PW01	PW00	PWCL0, PWCL1
リード/ライト⇒	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	

#### ● タイマモード

リロードタイマ動作モード (PWCSL0, PWCSL1 レジスタの S/C:bit3=1) 時は, このレジスタはリロード値を格納します。このレジスタは, 読み / 書き可能です。

単一タイマ動作モード (PWCSL0, PWCSL1 レジスタの S/C:bit3=0) 時は, このレジスタへ直接アクセスすると, アップカウントタイマがアクセスされます。このモード時は, このレジスタは読み / 書き可能です。ただし, このレジスタへの書込みは, タイマが停止した場合にのみ行われます。このレジスタの読出しは常に可能であり, 読出し値は現タイマ値です。

#### ● パルス幅測定モード (読出し専用)

連続測定モード (PWCSL0, PWCSL1 レジスタの S/C:bit3=1) 時は, このレジスタはバッファレジスタとして機能し, 前のカウント結果を格納します。このレジスタは読出し専用レジスタであり, 値を書き込んでもレジスタ値は変化しません。

単一測定モード (PWCSL0, PWCSL1 レジスタの S/C:bit3=0) 時は, このレジスタへ直接アクセスすると, アップカウントタイマがアクセスされます。このモード時も, このレジスタは読出し専用レジスタであり, 値を書き込んでもレジスタ値は変化しません。このレジスタの読出しは常に可能で, 読出し値は現タイマ値です。カウントが実行されると, レジスタはカウント結果を格納します。

< 注意事項 > このレジスタへアクセスする場合は, 必ずワード転送命令をご使用ください。リセットで, このレジスタは "0000<sub>H</sub>" に初期化されます。

16.4.3 分周比制御レジスタ (DIV0, DIV1)

分周比制御レジスタ (DIV0, DIV1) は, 分周周期測定モード (PWCSL0, PWCSL1 レジスタの MOD2 ~ MOD0:bit2 ~ bit0=011<sub>B</sub>) 時に使用します。このレジスタは, ほかのモード時は意味を持ちません。

■ 分周比制御レジスタ (DIV0, DIV1)

図 16.4-5 分周比制御レジスタ (DIV0, DIV1)

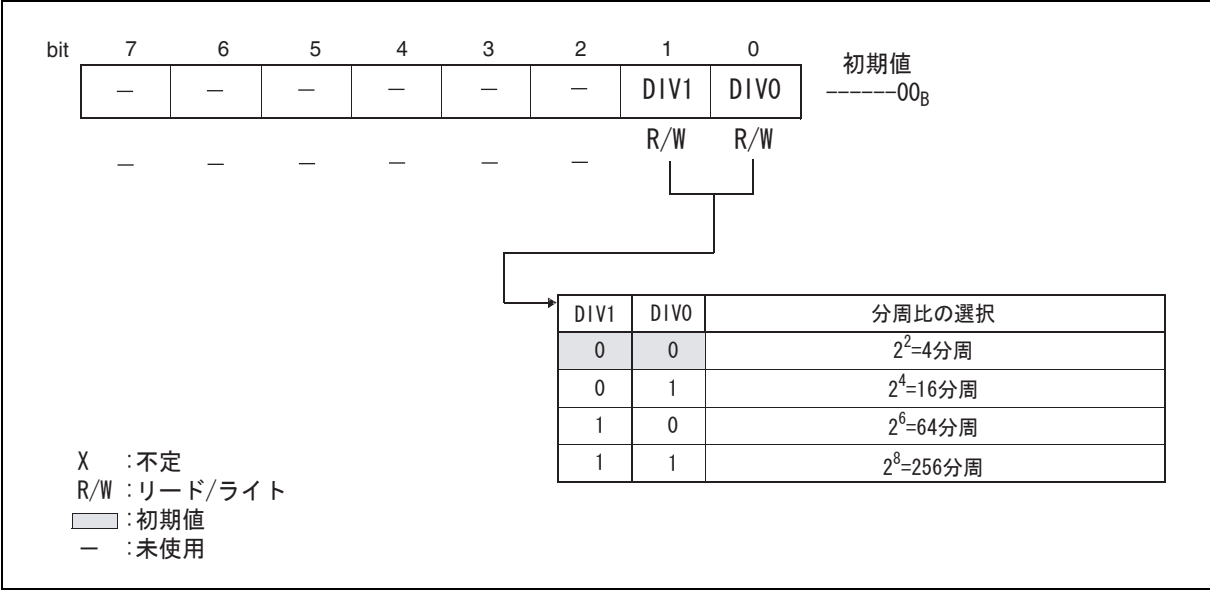


表 16.4-3 分周比制御レジスタ (DIV0, DIV1)

ビット名		機 能
bit7, ~ bit2	未使用ビット	<ul style="list-style-type: none"><li>読出し値は不定です。</li><li>これらのビットへ値を書き込んでも動作は影響を受けません。</li></ul>
bit1, bit0	DIV1, DIV0: 分周比選択 ビット	<ul style="list-style-type: none"><li>分周範囲管理モード時は, このレジスタは測定端子からのパルス入力を分周するためおよび分周後の 1 周期幅を測定するために使用します。</li><li>リセットされると, これらのビットは "00<sub>B</sub>" に初期化されます。これらのビットは, 読み / 書き可能です。</li></ul> <p>(注意事項)</p> <p>タイマ起動後にこの設定は変更できません。これらのビットへの書込みは, タイマ起動前またはタイマ停止後に行ってください。</p>

## 16.5 PWC タイマの割込み

PWC タイマは、カウンタオーバフローが発生した場合またはパルス幅測定モード時に測定が終了した場合、割込み要求を生成できます。PWC タイマはまた、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) とも連係しています。

### ■ PWC タイマの割込み

PWC タイマの割込み制御ビットと割込み要因を表 16.5-1 に示します。

表 16.5-1 PWC タイマの割込み制御ビットと割込み要因

	PWC タイマ 0		PWC タイマ 1	
割込み要求フラグビット	PWCSH0:OVIR ( bit11 )	PWCSH0:EDIR ( bit13 )	PWCSH1:OVIR ( bit11 )	PWCSH1:EDIR ( bit13 )
割込み要求許可ビット	PWCSH0:OVIE ( bit10 )	PWCSH0:EDIE ( bit12 )	PWCSH1:OVIE ( bit10 )	PWCSH1:EDIE ( bit12 )
割込み要因	16 ビットアップカウンタのオーバフロー	パルス幅測定モード時の測定終了	16 ビットアップカウンタのオーバフロー	パルス幅測定モード時の測定終了

PWC タイマでは PWC 状態制御レジスタ上位 (PWCSH) の OVIR:bit11 は、アップカウンタでオーバフローが ("FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲において) 発生すると、"1" が設定されます。この状態において割込み要求が許可 (PWCSH レジスタの OVIE:bit10=1) になると、割込み要求が割込みコントローラへ出力されます。

PWC 状態制御レジスタ (PWCSH) の EDIR:bit13 は、パルス幅測定モード時の測定が終了すると "1" がセットされます。この状態において割込み要求が許可 (PWCSH レジスタの EDIE:bit12=1) になると、割込み要求が割込みコントローラへ出力されます。

### ■ PWC タイマ割込みと EI<sup>2</sup>OS

PWC タイマ割込みと EI<sup>2</sup>OS を表 16.5-2 に示します。

表 16.5-2 PWC タイマ割込みと EI<sup>2</sup>OS

チャネル	割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
PWC タイマ 0 <sup>*1</sup>	#13 ( 0D <sub>H</sub> )	ICR01	0000B1 <sub>H</sub>	FFFFC8 <sub>H</sub>	FFFFC9 <sub>H</sub>	FFFFCA <sub>H</sub>	
PWC タイマ 1 <sup>*2</sup>	#24 ( 18 <sub>H</sub> )	ICR06	0000B6 <sub>H</sub>	FFFF9C <sub>H</sub>	FFFF9D <sub>H</sub>	FFFF9E <sub>H</sub>	

\*1: 16 ビット PPG タイマ 0 の割込み番号と同じ番号が PWC タイマ 0 に割当てられます。

\*2: アウトプットコンペア ch.5 一致の割込み番号と同じ番号が PWC タイマ 1 に割当てられます。

## ■ PWC タイマの EI<sup>2</sup>OS 機能

PWC タイマは EI<sup>2</sup>OS と関係する回路を持っているので、カウンタは、オーバフローまたは測定終了が発生すると EI<sup>2</sup>OS を起動できます。

ただし、EI<sup>2</sup>OS は、割込み制御レジスタ (ICR) を共用するほかのリソースが割込みを使用しない場合に限り使用可能です。たとえば、PWC タイマ "0" が EI<sup>2</sup>OS を使用している場合、16 ビット PPG タイマ "0" の割込みは禁止にしなければなりません。



## 16.6 PWC タイマの動作

PWC タイマは、16 ビットアップカウントタイマを中核とした多機能タイマであり、カウント入力端子と 8 ビット入力分周器を備えています。PWC タイマは 2 つの主要機能（タイマ機能とパルス幅測定機能）を備えており、これらの機能はそれぞれ 2 種類のカウントクロックを選択できます。

### ■ タイマ機能

タイマ機能は、単一モードまたはリロードモード時の動作選択を有効にするアップカウントタイマです。

タイマが起動すると、各カウントクロックでタイマはカウントします。

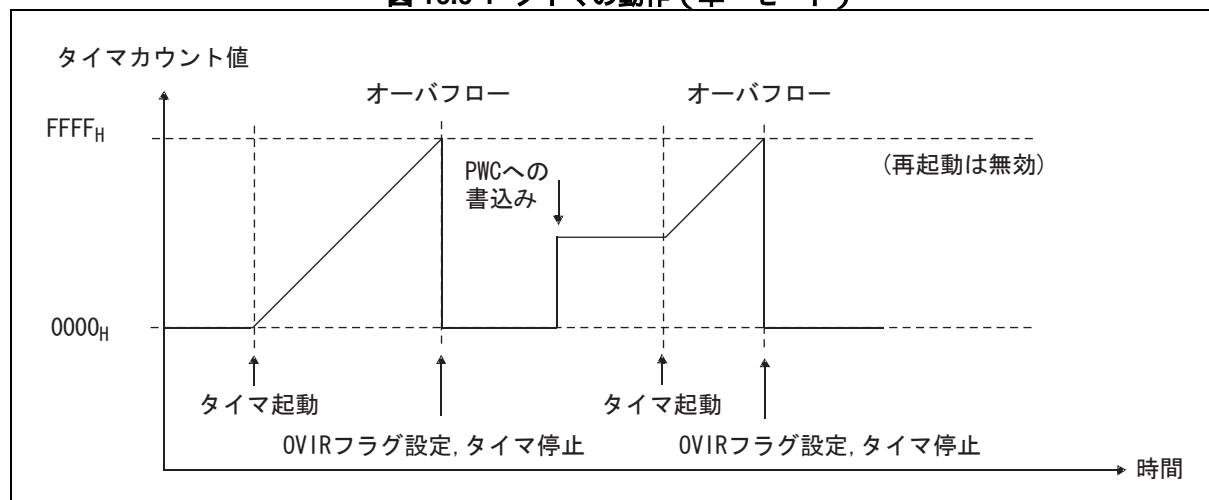
"FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲においてオーバフローが発生すると、割込み要求が送出されます。

オーバフローが発生すると、以下のイベントが発生します。

#### ● 単一モード

カウントは停止します（図 16.6-1 を参照）。

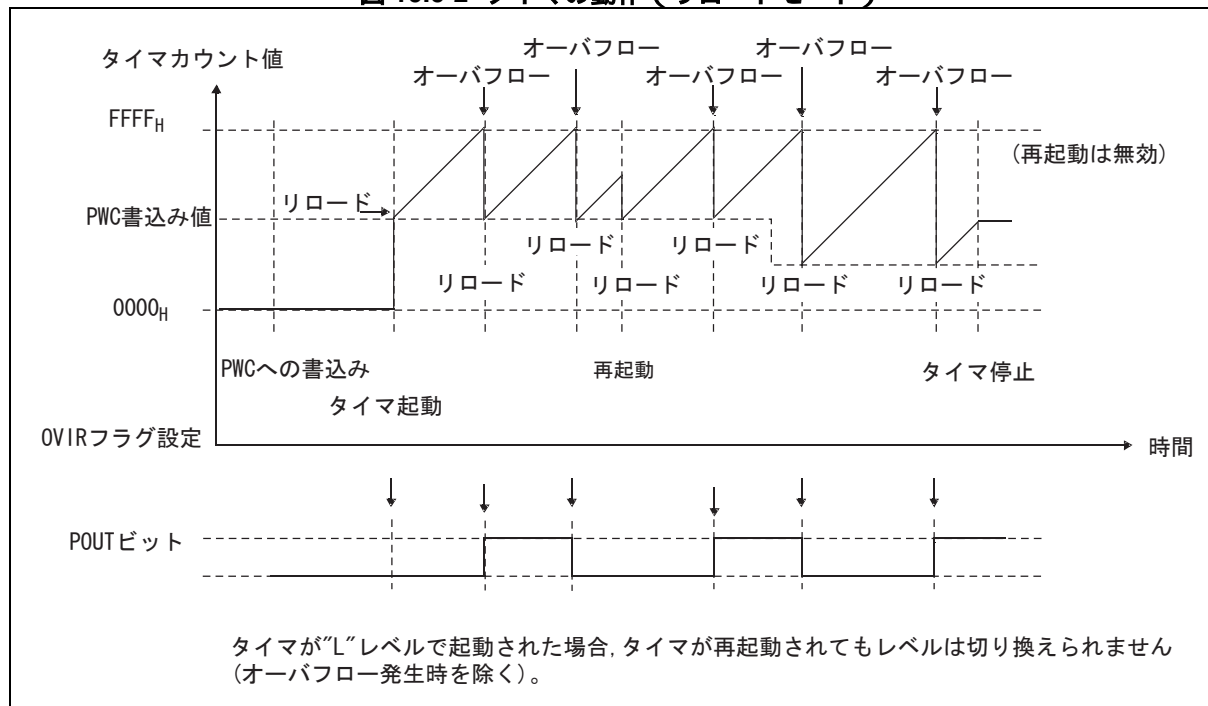
図 16.6-1 タイマの動作（単一モード）



## ● リロードモード

リロードレジスタの値がタイマにリロードされ、カウントが再起動します(図 16.6-2 を参照)。

図 16.6-2 タイマの動作 (リロードモード)



## ■ パルス幅測定機能

パルス幅測定機能は、入力パルスに関する " 指定されたイベント間 " の時間を算出します。

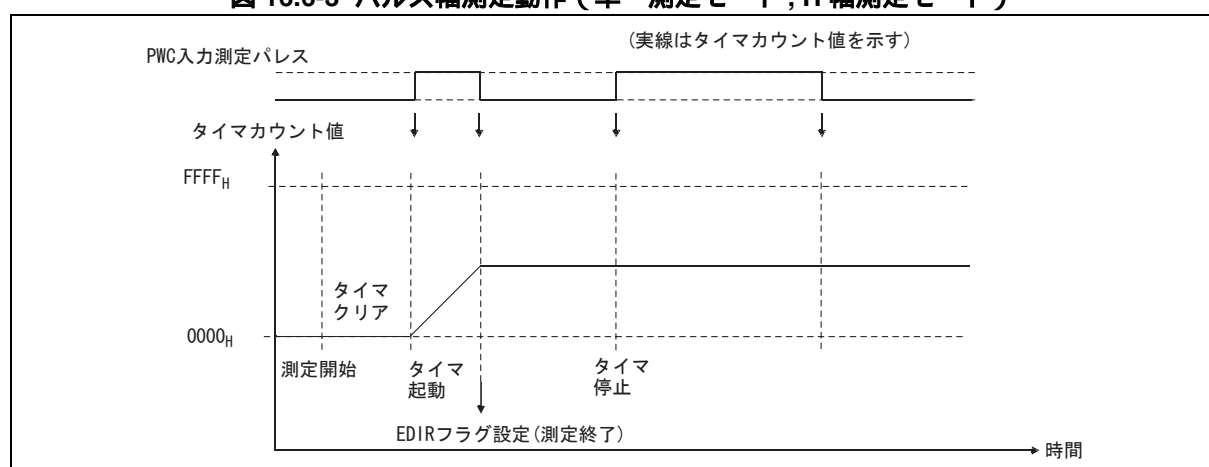
この機能がアクティブになると、指定されたカウント開始エッジが入力された後にカウントが開始します。カウンタが "0000<sub>H</sub>" にクリアされると、カウントは開始エッジ検出時に開始され、次に停止エッジが検出されます。この期間中のカウント値は、パルス幅としてレジスタに保持されます。

割込み要求は、測定終了時またはオーバーフロー発生時に生成できます。測定が完了すると、以下のイベントが発生します。

### ● 単一測定モード

動作が停止します ( 図 16.6-3 を参照 )。

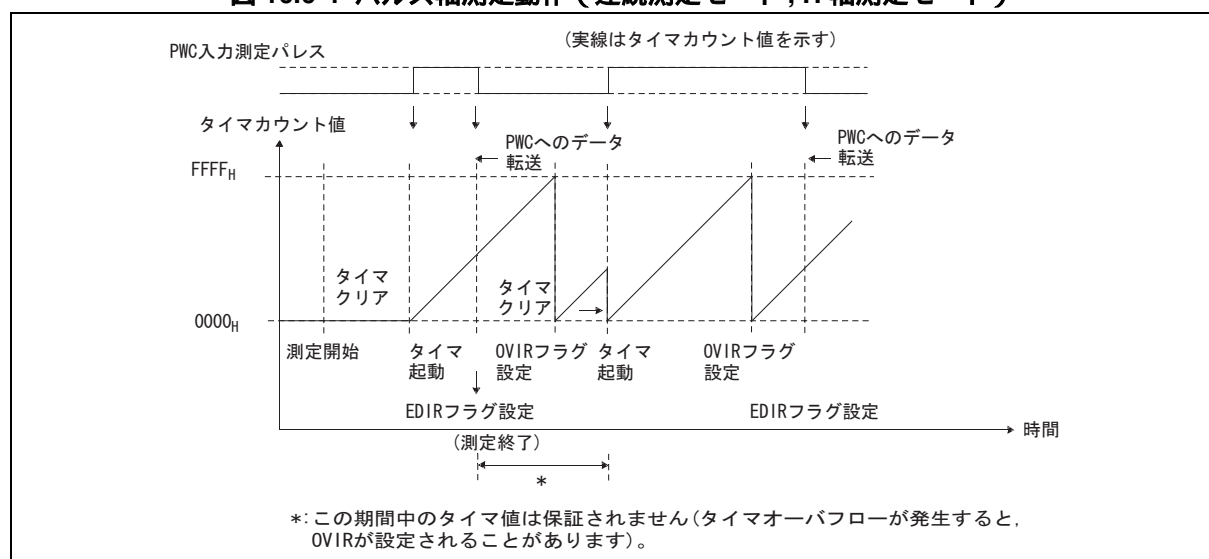
図 16.6-3 パルス幅測定動作 ( 単一測定モード, H 幅測定モード )



### ● 連続測定モード

タイマ値がバッファレジスタへ転送され、次のエッジが入力されるまでタイマはフリーラン状態になります ( 図 16.6-4 を参照 )。

図 16.6-4 パルス幅測定動作 ( 連続測定モード, H 幅測定モード )



## 16.6.1 動作モードの選択

動作モードとカウントモードは、PWCSL0, PWCSL1 レジスタの設定に従って選択されます。

### ■ 動作モードの選択

以下のレジスタは、動作モードとカウントモードの選択を設定するために使用されます。

- 動作モードの設定 :PWCSL0, PWCSL1 レジスタの MOD2 ~ MOD0:bit2 ~ bit0

カウント動作の制御を指定するには、タイマモードまたはパルス幅測定モードを選択してください。

- カウントモードの設定 :PWCSL0, PWCSL1 レジスタの S/C:bit3

単一測定、連続測定、リロード動作またはワンショット動作を選択してください。

動作モードビットで選択された動作モードを表 16.6-1 に示します。

表 16.6-1 動作モードの選択

動作モード		S/C	MOD2	MOD1	MOD0
タイマ	ワンショットタイマ	0	0	0	0
	リロードタイマ	1	0	0	0/1
	設定禁止	0	0	0	1
パルス幅測定	立上りエッジから立下りエッジまでまたは立下りエッジから立上りエッジまで:全エッジ間測定	単一測定:バッファは無効	0	0	1
		連続測定:バッファは有効	1	0	1
	分周カウント:4分周から256分周まで	単一測定:バッファは無効	0	0	1
		連続測定:バッファは有効	1	0	1
	立上りエッジから次の立上りエッジまで:立上りエッジから次の立上りエッジまでの周期測定	単一測定:バッファは無効	0	1	0
		連続測定:バッファは有効	1	1	0
	立上りエッジから立下りエッジまで:Hパルス幅測定	単一測定:バッファは無効	0	1	0
		連続測定:バッファは有効	1	1	0
	立下りエッジから立上りエッジまで:Lパルス幅測定	単一測定:バッファは無効	0	1	1
		連続測定:バッファは有効	1	1	1
	立下りエッジから次の立下りエッジまで:立下りエッジから次の立下りエッジまでの周期測定	単一測定:バッファは無効	0	1	1
		連続測定:バッファは有効	1	1	1

リセットされると、ワンショットタイマが初期値として選択されます。

< 注意事項 > タイマを起動する場合は、必ず前もって動作モードを選択してください。

## 16.6.2 タイマとパルス幅測定 of 起動 / 停止と , タイマクリア

タイマとパルス幅測定を強制的に起動 , 再起動または停止するには , PWCSH0, PWCSH1 レジスタ STRT:bit15 と STOP:bit14 を使用してください。

16 ビットアップカウントタイマは , リセット時 , 測定開始エッジ検出時またはパルス幅測定モード時のカウント開始時に "0000<sub>H</sub>" にクリアされます。

### ■ タイマとパルス幅測定 of 起動 / 停止

STRT:bit15 に "0" を書き込むと , 動作が起動または再起動され , STOP:bit14 に "0" を書き込むと , 動作が停止します。ただし , これら 2 つのビットに書き込まれる値が同じである限り , これらのビットは動作を実行しません。ビット操作命令以外の命令 ( バイト命令またはワード命令 ) を使用した場合は , 値の書き込みは以下のビット組合せに対してのみ行われます。

表 16.6-2 パルス幅測定動作 ( 単一測定モード , H 幅測定モード )

機 能	STRT	STOP
タイマまたはパルス幅測定を起動 / 再起動する	0	1
タイマまたはパルス幅測定を停止する	1	0

ビット操作命令のクリアビット命令を使用する場合 , ハードウェアは自動的に上記の値の組合せを書き込みます。ユーザは , どの値を書き込むべきかについて意識する必要はありません。

#### ● 起動後の動作

タイマモード : カウント動作を直ちに開始します。

パルス幅測定モード : 測定は , 測定開始エッジが入力された後に開始します。測定開始エッジが検出されると , 16 ビットアップカウントタイマは "0000<sub>H</sub>" にクリアされ , カウントを開始します。

#### ● タイマの再起動

タイマモードまたはパルス幅測定モード時にタイマを起動し , その後にタイマ動作が継続されている間にタイマを再起動する ( PWCSH0, PWCSH1 レジスタの STRT:bit15 に "0" を書き込む ) ことを , " タイマ再起動 " とよびます。再起動中に実行される動作は , 以下に示すモードによって異なります。

ワンショットモード : 動作は影響を受けません。

リロードタイマモード : リロードが実行され , 動作は継続します。オーバフロー発生時にタイマが再起動されると , オーバフローフラグ ( PWCSH0, PWCSH1 レジスタの OVIR:bit11 ) が設定され , PWCSH0, PWCSH1 レジスタの POUT:bit8 が反転されます。

パルス幅測定モード : 測定開始エッジ待ち状態時は、動作は影響を受けません。測定中、カウントは停止され、タイマ状態は"測定開始エッジ待ち"状態へ戻ります。測定終了時にタイマが再起動されると、測定終了フラグ (PWCSH0, PWCSH1 レジスタの EDIR:bit13) が設定され、連続測定モード時に測定結果は PWC へ転送されます。

### ● タイマの停止

ワンショットタイマモードまたは単一測定モード時は、測定はタイマオーバーフロー発生時またはカウント終了時に自動的に停止します。ユーザは、タイマが停止したか否かを意識する必要はありません。ただし、ほかのモード時は、タイマは停止しなければなりません。これは、タイマが自動的に停止する前にタイマを停止する場合も含まれます。

### ● 動作状態の検査

すでに述べた STRT ビットと STOP ビットは、読出し中、タイマの動作状態を示すビットとして機能します。動作状態指示ビットの機能を示します。

表 16.6-3 動作状態指示ビットの機能

STRT	STOP	動作状態
0	0	タイマは停止している（測定開始エッジ待ち状態を除く）。これらのビットは、タイマが起動されていないか、あるいは測定が終了していることを示します。
1	1	測定開始エッジ待ち状態またはタイマカウント動作

読出し中、STRT ビットと STOP ビットは両方とも同じ値となります。ただし、リードモディファイライト命令を使用しての読出し中は、これらのビットの値は常に "11<sub>B</sub>" です。リードモディファイライト命令を使用して、これらのビットの値を読出してはいけません。

## ■ タイマクリア

以下の場合、16 ビットアップカウントタイマは "0000<sub>H</sub>" にクリアされます。

- リセット時
- パルス幅測定モード時にカウント開始エッジが検出された後でカウントが開始したとき

## 16.6.3 タイマモード時の動作

タイマモードには、ワンショット動作モードとリロード動作モードがあります。

### ■ ワンショット動作モード

このモード時にタイマが起動すると、カウンタはカウントクロックごとにインクリメントされます。タイマは、オーバーフローが "FFFF<sub>H</sub>" から "0000<sub>H</sub>" の範囲において発生すると、自動的に停止します。

タイマが起動される前にPWCが設定されると、カウンタはこの設定値から開始します。オーバーフローが発生すると、設定値が削除され、現カウンタ値が PWC に残ります。

オーバーフローが発生すると、PWCSH0, PWCSH1 レジスタの POUT:bit8 )は反転します。

### ■ リロード動作モード

このモード時にタイマが起動すると、PWC のリロード値がタイマへ設定され、カウンタがカウントクロックごとにインクリメントされます。タイマが "FFFF<sub>H</sub>" から "0000<sub>H</sub>" までをカウントしている間にオーバーフローが発生すると、PWC のリロード値がタイマへ再び設定され、PWCSH0, PWCSH1 レジスタの POUT:bit8 が反転し、カウンタ動作が反復します。タイマは、PWCSH0, PWCSH1 レジスタの STOP:bit14( タイマを停止する ) に値が書き込まれるリセットが発生するまで停止しません。ポートビットは、パルス出力モードが指定されると、PWO 端子に出力されます。

タイマが起動される前にPWCに設定されたリロード値は、カウント時に格納されます。タイマが起動または再起動し、オーバーフローが発生すると、タイマにリロード値が必ず設定されます。カウント時に設定された値が変更されると、新しいリロード値は、次のオーバーフローが発生したときまたはタイマが再起動されたときに有効になります。

### ■ タイマ値とリロード値

ワンショット動作モード時は、PWC レジスタへ直接アクセスすると、アップカウントタイマがアクセスされます。PWC に値が書き込まれると、この値はタイマへ直接書き込まれます。カウント動作時に PWC が読み出されると、現タイマ値が読み出されます。タイマが起動される前に値が PWC に設定されると、タイマは指定された値からカウントを開始します。

リロード動作モード時は、アップカウントタイマへアクセスすることはできず、PWC はリロードレジスタ(リロード値を格納する)として機能します。タイマが起動または再起動し、オーバーフローが発生すると、PWC に書き込まれた値は必ずタイマへ設定されます。PWC が読み出されると、格納されたリロード値が読み出されます。

PWC 値とタイマ値は、リロードモード時に停止された後でタイマがワンショットモードで設定されると、不定となります。したがって、値は必ずタイマを使用する前に設定してください。

PWC 値は、ワンショット動作モード時に強制停止された後でタイマがリロードモードで設定されると、不定となります。したがって、値は必ずタイマを使用する前に設定してください。

■ 割込み要求の生成

タイマモード時の動作時に、オーバーフローが発生すると、割込み要求が生成されます。タイマカウン트의インクリメントによりオーバーフローが発生すると、オーバーフローフラグが設定され、オーバーフロー割込み要求がイネーブルになり、割込み要求が生成されます。

■ タイマ周期

PWC に "0000<sub>H</sub>" が設定された後にタイマがワンショットモードで起動された場合、カウントが "65536" を超えると、タイマオーバーフローが発生し、カウントは停止します。以下の公式では、タイマが起動されてから停止されるまでの時間を計算します。

$$T_1 = (65536 - n_1) \times t \quad \left\{ \begin{array}{l} T_1 \cdots \cdots \text{タイマが起動されてから停止されるまでの時間} (\mu s) \\ n_1 \cdots \cdots \text{タイマ起動時にPWCに設定されたタイマ値} \\ t \cdots \cdots \text{カウントクロック周期} (\mu s) \end{array} \right.$$

PWC に "0000<sub>H</sub>" が設定された後にタイマが起動されると、カウントが "65536" を超えるたびにタイマオーバーフローが発生します。以下の公式では、リロード周期と PWO 端子出力パルス周期を計算します。

$$T_R = (65536 - N_R) \times t \quad \left\{ \begin{array}{l} T_R \cdots \cdots \text{リロード周期(オーバーフロー周期)} (\mu s) \\ T_{POUT} \cdots \cdots \text{PWO端子出力パルス周期} (\mu s) \\ N_R \cdots \cdots \text{PWCに格納されたリロード値} (\mu s) \\ t \cdots \cdots \text{タイマが起動されてから停止されるまでの時間} (\mu s) \end{array} \right.$$

■ カウントクロック周期と最大周期

タイマモード時は、PWC に "0000<sub>H</sub>" が設定されると、最大周期が発生します。  
内部動作周波数 16 MHz (表中では  $\phi$  で示されています) に対応するカウントクロック周期とタイマ最大周期を表 16.6-4 に示します。

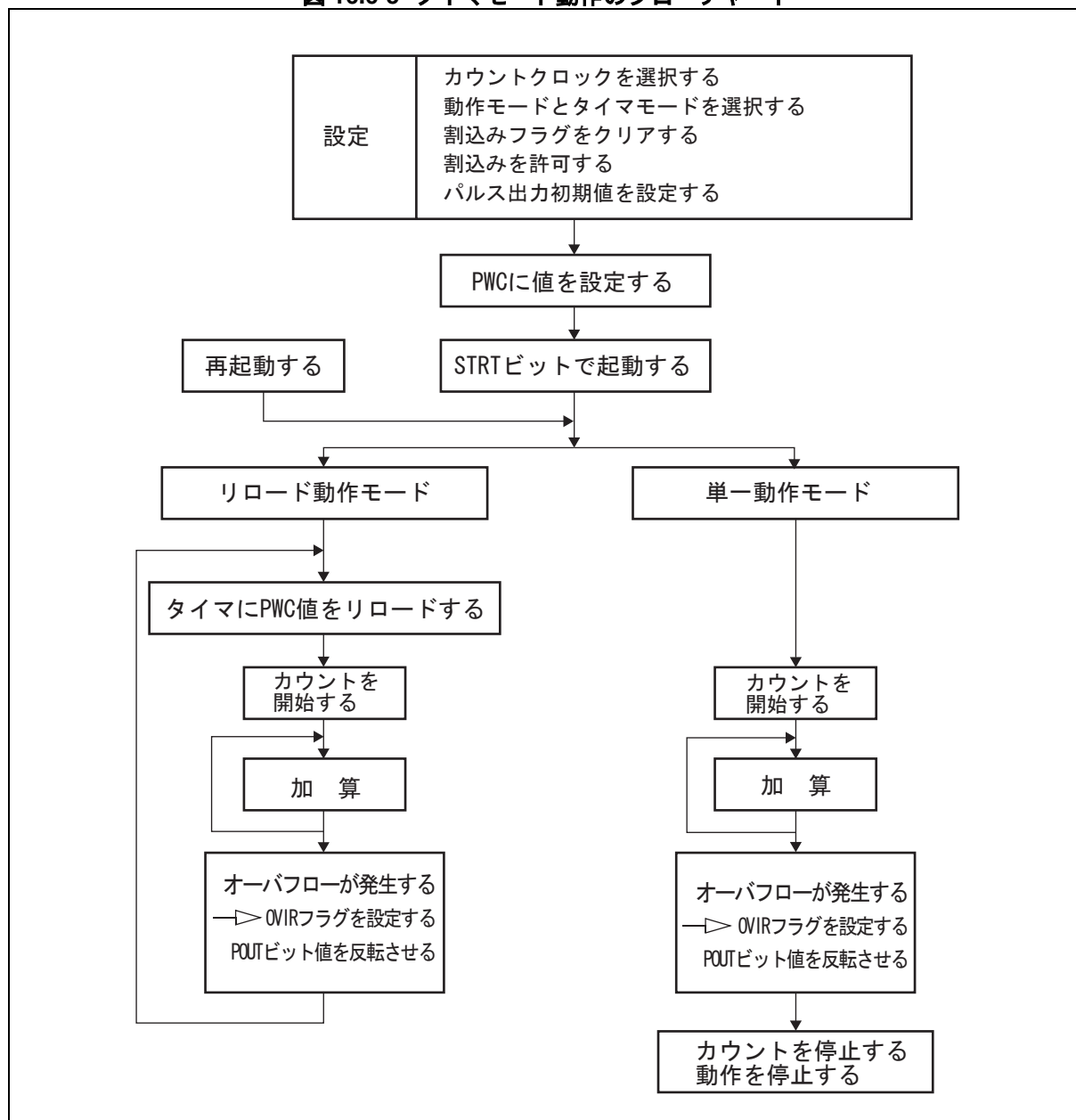
表 16.6-4 カウントクロック周期と最大周期

カウントクロック選択	CKS1, CKS0:bit7, bit6=00 <sub>B</sub> の場合 ( $\phi / 4$ )	CKS1, CKS0:bit7, bit 6=00 <sub>B</sub> の場合 ( $\phi / 16$ )	CKS1, CKS0:bit7, bit6=00 <sub>B</sub> の場合 ( $\phi / 32$ )
カウントクロック周期	0.25 $\mu$ s	1 $\mu$ s	2 $\mu$ s
タイマ最大周期	16.38ms	65.5ms	131.1ms



## ■ タイマモード動作のフローチャート

図 16.6-5 タイマモード動作のフローチャート



## 16.6.4 パルス幅測定モード時の動作

パルス幅を測定するための信号が PWI 端子から入力されます。

パルス幅測定モードには、1 回のみカウントする単一測定モードと、パルス幅が連続的に測定される連続測定モードがあります。

### ■ 単一測定モードと連続測定モード

単一測定モードと連続測定モードの違いを以下に示します。

#### ● 単一測定モード

先頭のカウント終了エッジが入力されると、タイマはカウントを停止し、PWCSH0, 1 レジスタの測定終了割込み要求フラグビット (EDIR:bit13) が設定され、以降は測定しません。ただし、タイマの再起動も指定されると、タイマ状態は測定開始エッジ待ち状態に変化します。

#### ● 連続測定モード

##### [H/L パルス幅測定モード]

カウント終了エッジが入力されると、PWCSH0, 1 レジスタの測定終了割込み要求フラグビット (EDIR:bit13) が設定され、タイマカウント結果が PWC に転送され、タイマはフリーラン状態でカウントのインクリメントを継続できます。次のカウント開始エッジが入力されると、タイマは "0000<sub>H</sub>" にクリアされ、パルス幅カウントを開始します。

カウント終了エッジが入力され、タイマがフリーラン状態になると、タイマオーバフローが発生してオーバフロー割込み要求ビット (OVIR:bit11) が設定されることがあります。H/L パルス幅測定モード時は、OVIR:bit11 を使用してパルス幅時間を測定してはいけません。

##### [全エッジ間パルス幅測定モード, 分周周期測定モード, 立上りエッジ間周期測定モード, 立下りエッジ間周期測定モード]

カウント終了エッジ (カウント開始エッジ) が入力されると、PWCSH0, PWCSH1 レジスタの測定終了割込み要求フラグビット (EDIR:bit13) が設定され、タイマカウント結果が PWC へ転送され、タイマは "0000<sub>H</sub>" にクリアされ、カウントは再び開始します。

### ■ 測定結果データ

測定結果、タイマ値、PWC 機能に対する処理は、単一測定モードと連続測定モード時では以下のように異なります。

#### ● 単一測定モード

タイマ動作時に PWC が読み出されると、現タイマ値が読み出されます。

測定終了後に PWC が読み出されると、測定結果が読み出されます。

### ● 連続測定モード

測定終了時にタイマ測定結果が PWC へ転送されます。

PWC 読出し時には、前の測定結果が読み出されます。測定が進行中の場合は、前の測定結果が PWC に格納されます。測定時には、タイマ値を読み出すことはできません。

連続測定モード時は、既存の値は新しい測定結果で上書きされます。ただし、次の測定が完了する前に前の測定結果を読み出さない場合は除きます。この場合は、PWCSH0、PWCSH1 レジスタのエラーフラグビット (ERR:bit9) が設定されます。PWC が読み出されると、エラーフラグビット (ERR:bit9) は自動的にクリアされます。

### ■ 最小入力パルス幅

以下の最小入力パルス幅よりも長いパルスを入力パルス幅カウンタ入力端子 (PWI) へ入力しなければなりません。

**パルス幅 : 2 マシンサイクル (内部動作周波数 16MHz の場合は 0.125  $\mu$ s 以上)**

ただし、上記の仕様よりも短い入力パルスも有効パルスとして認識できます。

### ■ パルス幅 / 周期の算出

測定オブジェクトのパルス幅またはパルス周期は、以下のように、カウンタ終了時に PWC から読み出されたカウンタ結果に基づいて算出されます。

$$T_w = n \times t / \text{Div} (\mu s) \quad \left\{ \begin{array}{l} T_w \cdots \cdots \text{測定されたパルス幅またはパルス周期} (\mu s) \\ n \cdots \cdots \text{PWC に格納されている測定結果} \\ t \cdots \cdots \text{カウンタクロック周期} (\mu s) \\ \text{Div} \cdots \cdots \text{分周比レジスタ (DIV0, 1) に設定されている分周比} \\ \quad (\text{値 "1" は分周カウンタモード以外のモードで使用されます}) \end{array} \right.$$

### ■ パルス幅 / 周期測定範囲

測定可能なパルス幅 / 周期の範囲は、入力分周器のカウントクロックと分周比によって異なります。

内部動作周波数 16MHz 時の (      で示されます ) 測定範囲を表 16.6-5 に示します。

表 16.6-5 パルス幅測定範囲

分周比	DIV0, DIV1	PWCSL0, PWCSL1 レジスタ CKS1, CKS0=00 <sub>B</sub> ( $\phi/4$ )	PWCSL0, PWCSL1 レジスタ CKS1, CKS0=01 <sub>B</sub> ( $\phi/16$ )	PWCSL0, PWCSL1 レジスタ CKS1, CKS0=10 <sub>B</sub> ( $\phi/32$ )
分周なし	-	0.125 $\mu$ s ~ 16.38ms[0.25 $\mu$ s]	0.125 $\mu$ s ~ 65.5ms[1.0 $\mu$ s]	0.125 $\mu$ s ~ 131ms[2 $\mu$ s]
4 分周	00 <sub>B</sub>	0.125 $\mu$ s ~ 4.10ms[62.5ns]	0.125 $\mu$ s ~ 16.38ms[0.25 $\mu$ s]	0.125 $\mu$ s ~ 32.75ms[500ns]
16 分周	01 <sub>B</sub>	0.125 $\mu$ s ~ 1024 $\mu$ s[15.6ns]	0.125 $\mu$ s ~ 4.10ms[62.5ns]	0.125 $\mu$ s ~ 8.19ms[125ns]
64 分周	00 <sub>B</sub>	0.125 $\mu$ s ~ 256 $\mu$ s[3.91ns]	0.125 $\mu$ s ~ 1024 $\mu$ s[15.6ns]	0.125 $\mu$ s ~ 2.048ms[31.25ns]
256 分周	11 <sub>B</sub>	0.125 $\mu$ s ~ 64 $\mu$ s[0.98ns]	0.125 $\mu$ s ~ 256 $\mu$ s[3.91ns]	0.125 $\mu$ s ~ 512 $\mu$ s[7.81ns]

[ ] 内の数字はビット当りの分解能を示します。

## ■ 割込み要求の生成

パルス幅測定モード時は、以下の 2 つの割込み要求を生成できます。

### ● タイマオーバフロー割込み要求

カウント時にオーバフローが発生すると、オーバフロー割込み要求ビットが設定されます。オーバフロー割込み要求が許可になると、割込み要求が生成されます。

### ● 測定終了割込み要求

測定終了エッジが検出されると、PWCSH0, PWCSH1 レジスタの測定終了割込み要求フラグビット (EDIR:bit13) が設定されます。測定終了割込みが許可になると、割込み要求が生成されます。

測定終了割込み要求フラグビット (EDIR:bit13) は、PWC が読み出されると自動的にクリアされます。

## ■ 測定モードと測定動作

測定モード時における動作を表 16.6-6 に示します。

表 16.6-6 測定モード時における動作 ( 1 / 2 )

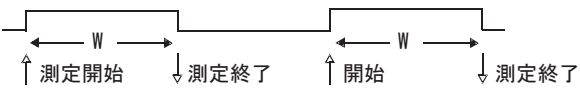
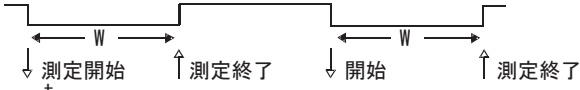
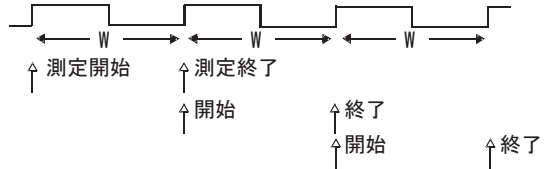
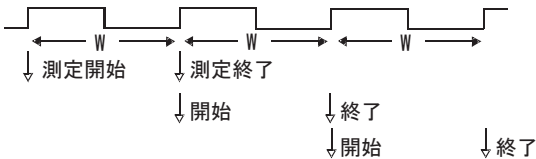
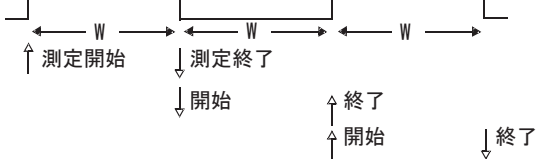
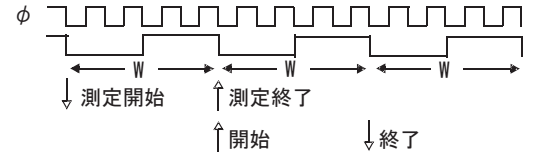
測定モード	MOD2	MOD1	MOD0	測定動作
H パルス幅測定	1	0	1	 <p>H 周期幅が測定されます。 測定開始: 立上りエッジが検出されたとき 測定終了: 立上りエッジが検出されたとき</p>
L パルス幅測定	1	1	0	 <p>L 周期幅が測定されます。 測定開始: 立下りエッジが検出されたとき 測定終了: 立下りエッジが検出されたとき</p>
立上りエッジ間周期測定	1	0	0	 <p>立上りエッジ間時間が測定されます。 測定開始: 立上りエッジが検出されたとき 測定終了: 立上りエッジが検出されたとき</p>

表 16.6-6 測定モード時における動作 ( 2 / 2 )

測定モード	MOD2	MOD1	MOD0	測定動作
立下りエッジ間周期測定	1	1	1	 <p>立下りエッジ間時間が測定されます。 測定開始: 立下りエッジが検出されたとき 測定終了: 立下りエッジが検出されたとき</p>
全エッジ間パルス幅測定	0	1	0	 <p>連続入力エッジ間の幅が測定されます。 測定開始: エッジが検出されたとき 測定終了: エッジが検出されたとき</p>
分周周期測定	0	1	1	 <p>(上記の例では4分周されている) 入力パルスは分周比レジスタ (DIV0, 1) に設定されている分周比で分周され、測定周期が求められます。 測定開始: 動作が起動された後、立下りエッジが検出されたとき 測定終了: 分周信号の1周期が終了したとき</p>

W: 測定されるパルス幅

すべてのモードにおいて、測定が開始されてから測定開始エッジが入力されるまでの間、タイマはカウントを開始しません。測定開始エッジが入力されると、タイマは "0000<sub>H</sub>" にクリアされ、カウントは測定終了エッジが入力されるまでカウントクロックごとにインクリメントされます。

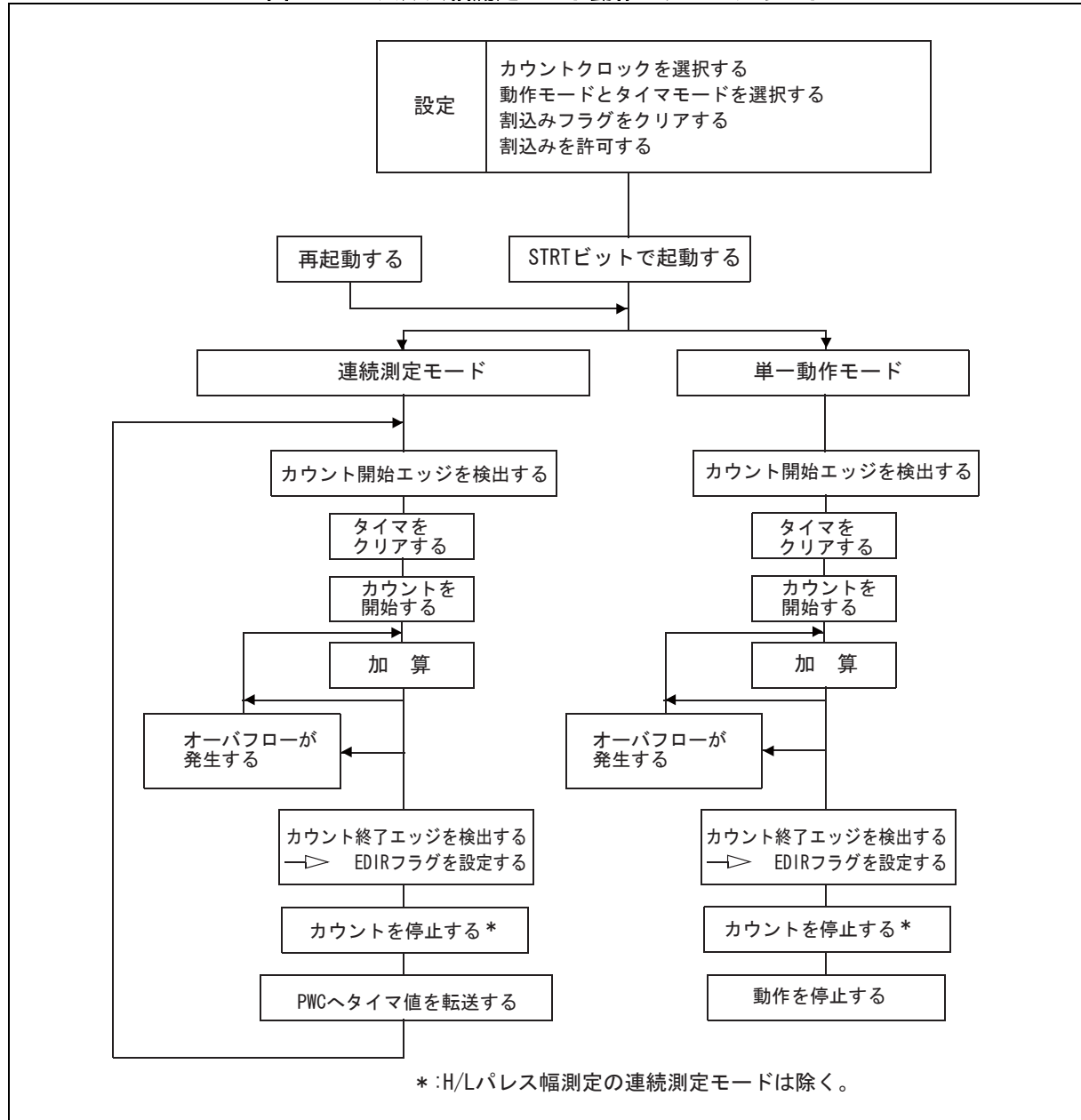
測定終了エッジが入力されると、以下の動作が実行されます。

- PWCSH0, PWCSH1 レジスタの測定終了割り込み要求フラグビット (EDIR:bit13) が設定されます。
- タイマがカウント動作を停止します (タイマが測定終了エッジ入力時と同時に再起動された場合または H/L パルス幅測定の連続測定モードが使用された場合は除きます)。
- 連続測定モード: タイマ値 (測定結果) が PWC へ転送されます。
- 単一測定モード: 測定が終了します (タイマが測定終了エッジ入力時と同時に再起動された場合は除く)。

連続測定モードで全エッジ間パルス幅測定、周期測定、立下りエッジ間周期測定または立上りエッジ間周期測定が行われる場合、終了エッジが次の測定開始エッジになります。

## ■ パルス幅測定動作のフローチャート

図 16.6-6 パルス幅測定モード動作のフローチャート



## 16.7 PWC タイマ使用上の注意

---

PWC タイマ使用上の注意を以下に示します。

---

### ■ PWC タイマ使用上の注意

#### ● プログラムで設定する際の注意

- 以下の PWCSH0, PWCSH1 と PWCSL0, PWCSL1 レジスタのビット値は、タイマ動作時に変更することは禁止されています。これらのビット値の変更は、タイマが起動される前かまたは動作が停止された後でのみ行われます。
- [bit7, bit6] CKS1, CKS0: カウントクロック選択ビット
- [bit3] S/C: 測定モード（単一または連続）選択ビット
- bit2 ~ bit0] MOD2, MOD1, MOD0: 動作モード & 測定エッジ選択ビット
- タイマ動作時にパルス出力ビット（POUT:bit8）に値が書き込まれても、このビットの値は変化しませんので注意してください。
- タイマ動作時における DIVR 値の変更は禁止されています。DIV0, DIV1 レジスタ値の変更は、タイマが起動される前かまたは動作が停止された後で行ってください。
- PWC 状態制御レジスタ下位（PWCSL0, PWCSL1）のカウントクロック選択ビット（CKS1, CKS0:bit7, bit6）に "11<sub>B</sub>" を設定することは禁止されています。
- PWC 値とタイマ値は、タイマがワンショットモードで設定されたときまたは動作がリロードタイマモードで終了した後に決定されます。したがって、これらの値の設定は必ずタイマを使用した後で行ってください。
- PWC 値は、動作がワンショットモードで停止された後にタイマがリロードタイマモードで設定されると不定となります。したがって、PWC 値は必ずタイマを使用する前に設定してください。
- モードをパルス幅測定モードからタイマモードへ変更する場合は、PWC への値は必ずタイマが起動される前に設定してください。
- パルス幅測定モード時に分周周期測定モードを使用すると、入力パルスは分周されます。カウント結果から算出されたパルス幅は平均値になるので注意してください。
- パルス幅測定モード時に連続測定が実行されている間、内部カウントクロックの分周回路はクリアされず、カウントクロックよりも小さいエッジ数がカウント結果に加算されます。

#### ● プログラムで状態を検査する際の注意

- タイマモード時は、PWCSH0, PWCSH1 レジスタの測定終了割込み要求フラグビット（EDIR:bit13）の値は意味がありません。したがって、PWCSH0, PWCSH1 レジスタの測定終了割込み許可ビット（EDIE:bit12）には必ず "0" を設定してください。
- PWC 状態制御レジスタ上位（PWCSH0, PWCSH1）の STRT:bit15 と STOP:bit14 の動作は、読み出し時と書き込み時では異なります（「16.4.1 PWC 状態制御レジスタ（PWCSH0, PWCSH1, PWCSL0, PWCSL1）」を参照）。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必ず "11<sub>B</sub>" が読み出されます。したがって、ビット操

作命令を使用して動作状態を読み出すことはできません。

ただし、ビット操作命令（ビットクリア命令）は、STRT:bit15 または STOP:bit14 に値を書き込むことによってタイマを起動または停止する際には使用できます。

- パルス幅測定モード時は、測定開始エッジが検出されると、タイマはクリアされます。また、前のタイマデータは意味がありません。

● パルス幅測定入力端子に入力されるパルスに関する注意

- 最小パルス幅は、マシンサイクルを 2 分周した幅です（内部動作周波数が 16MHz の場合は  $0.125\ \mu\text{s}$  以上）。
- 最大入力周波数は、マシンサイクルを 4 分周した幅です（内部動作周波数が 16MHz の場合は 4MHz 未満）。
- 上記よりも小さいパルス幅または上記よりも大きい周波数が入力されると、タイマ動作は保証されません。上記の制約を超える入力信号のノイズは、低減させなければなりません。

● 動作時のタイマ再起動に関する注意

- リロードタイマモード時にオーバフローが発生すると、タイマが再起動しますが、オーバフロー割込み要求ビット（OVIR:bit11）が設定され、POUT:bit8 が反転します（すなわち、通常のオーバフローの場合と同じ動作です）。
- ワンショットパルス幅測定モード時に終了エッジが検出されると、タイマが再起動し、測定開始エッジ待ち状態になりますが、測定終了割込み要求ビット（EDIR:bit13）も設定されます。
- 連続パルス幅測定モード時に測定終了エッジが検出されると、タイマが再起動し、測定開始エッジ待ち状態になり、測定終了割込み要求フラグビット（EDIR:bit13）が設定され、測定結果が PWC へ転送されます。
- 動作時にタイマを再起動する場合は、割込み生成およびほかの制御用フラグビット（OVIR:bit11, EDIR:bit13）の動作に注意してください。

● 割込みについての注意

- PWC 状態制御レジスタ（PWCSH0, PWCSH1）の OVIR:bit11 に "1" を設定し、次に割込み要求を許可にすると（PWCSH0, PWCSH1 レジスタの OVIE:bit10=1）、制御は割込み処理から戻ることができません。OVIR:bit11 は、必ずクリアしてください。
- PWC 状態制御レジスタ（PWCSH0, PWCSH1）の EDIR:bit13 に "1" を設定し、次に割込み要求を許可にすると（PWCSH0, PWCSH1 レジスタの EDIE:bit12=1）、制御は割込み処理から戻ることができません。OVIR:bit11 は、必ずクリアしてください。
- PWC タイマは割込みベクトルをほかのリソースと共用するので、割込みを使用する場合は、割込み処理ルーチンで割込み要因を注意深くチェックしなければなりません。
- また、PWC タイマで EI<sup>2</sup>OS を使用する場合は、共用リソース割込みを禁止しなければなりません。



## 16.8 PWC タイマのプログラム例

PWC タイマのプログラム例を示します。

### ■ PWC タイマのプログラム例

#### ● 処理

- 30.6Hz の出力 PWO0 は PWC タイマ 0 により生成します。
- PWC はオーバーフロー割込みの繰返し生成のためにリロードタイマモード時に使用します。
- EI<sup>2</sup>OS は使用しません。
- 16MHz はマシナクロック用, 0.25  $\mu$ s はカウントクロック用です。

#### ● コーディング例

```
ICR01 EQU 0000B1H ; PWC タイマ用割込み制御レジスタ
PWCS0 EQU 000008H ; PWC 制御状態レジスタ
PWC0 EQU 00000AH ; PWC データバッファレジスタ
OVIR EQU PWCS0:11 ; 割込み要求フラグビット
;----- メインプログラム -----
CODE CSEG
START:
;      ; スタックポインタ (SP) が既に初期化されているものと仮定
AND CCR, #0BFH ; 割込み禁止
MOV I:ICR01, #00H ; 割込みレベル 0 (最強)
MOVW I:PSC0, #0FF00H ; リロード値を設定
MOVW I:PWCSL0, #4409H ; リロードタイマモードと 0.25  $\mu$ s クロックを設定
;      ; 有効 PWC 出力
;      ; オーバフロー割込みを設定
;      ; 割込みフラグをクリア, PWC タイマ開始
MOV ILM, #07H ; PS 内の ILM をレベル 7 に設定
OR CCR, #40H ; 割込み許可
LOOP: MOV A, #00H ; 無限ループ
MOV A, #01H ;
BRA LOOP ;
;----- 割込みプログラム -----
WARI:
CLRBI:OVIR ; 割込み要求フラグをクリア
;      ;
;      ; ユーザ処理
;      ;
RETI ; 割込みから復帰
CODE ENDS
```

```
;----- ベクタ設定 -----  
VECT      CSEG  ABS=0FFH  
          ORG   0FFC8H      ; 割込み #13 (0DH) 用ベクタを設定  
          DSL   WARI  
          ORG   0FFDCH      ; リセットベクタを設定  
          DSL   START  
          DB    00H          ; シングルチップモードを設定  
VECT      ENDS  
          END    START
```



# 第17章

---

## UART

UART の機能と動作について説明します。

- 17.1 UART の概要
- 17.2 UART のブロックダイアグラム
- 17.3 UART の構成
- 17.4 UART の割込み
- 17.5 UART のボーレート
- 17.6 UART の動作説明
- 17.7 UART 使用上の注意
- 17.8 UART のプログラム例

## 17.1 UART の概要

UART は、外部装置と同期通信または非同期通信を行うための汎用のシリアルデータ通信インタフェースです。

- クロック同期、クロック非同期の双方向通信機能を備えています。
- マスタ/スレーブ型通信機能（マルチプロセッサモード）を備えています（マスタ側のみ使用可能）。
- 送信完了、受信完了および受信エラー検出で、割込み要求を発生できます。
- 拡張インテリジェント I/O サービス EI<sup>2</sup>OS に対応しています。

### ■ UART の機能

UARTは、外部装置と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あり、表 17.1-1 の機能を備えています。

表 17.1-1 UART の機能

	機能
データバッファ	全二重ダブルバッファ
転送モード	<ul style="list-style-type: none"> <li>• クロック同期（スタート/ストップビットなし、パリティビットなし）</li> <li>• クロック非同期（調歩周期）</li> </ul>
ボーレート	<ul style="list-style-type: none"> <li>• 専用ボーレートジェネレータを内蔵、8 種類の値を選択可</li> <li>• 外部クロック入力可</li> <li>• 内部タイマ（16 ビットリロードタイマ 0）から供給されるクロックを利用可</li> </ul>
データ長	<ul style="list-style-type: none"> <li>• 7 ビット（非同期ノーマルモードのみ）</li> <li>• 8 ビット</li> </ul>
信号方式	NRZ（Non Return to Zero）方式
受信エラー検出	<ul style="list-style-type: none"> <li>• フレーミングエラー</li> <li>• オーバランエラー</li> <li>• パリティエラー（動作モード 1（マルチプロセッサモード）時は検出不可）</li> </ul>
割込み要求	<ul style="list-style-type: none"> <li>• 受信割込み（受信完了、受信エラー検出）</li> <li>• 送信割込み（送信完了）</li> <li>• 送受信とも拡張インテリジェント I/O サービス（EI<sup>2</sup>OS）に対応</li> </ul>
マスタ/スレーブ型通信機能（非同期マルチプロセッサモード時）	1（マスタ）対 n（スレーブ）間の通信が可能（マスタとしてのみ使用できます）

< 注意事項 > クロック同期転送時にスタートビット/ストップビットは付加されず、データのみを転送します。

表 17.1-2 UART 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
		パリティなし	パリティあり		
0	非同期モード (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 <sup>*2</sup>
1	マルチプロセッサモード	8 ビット + 1 ビット <sup>*1</sup>	-	非同期	
2	クロック同期モード	8 ビット	-	クロック同 期	なし

- : 設定不可

\*1: "+" は通信制御に使用するアドレス / データ選択ビット ( SCR0, SCR1 レジスタの A/D:bit11 ) です。

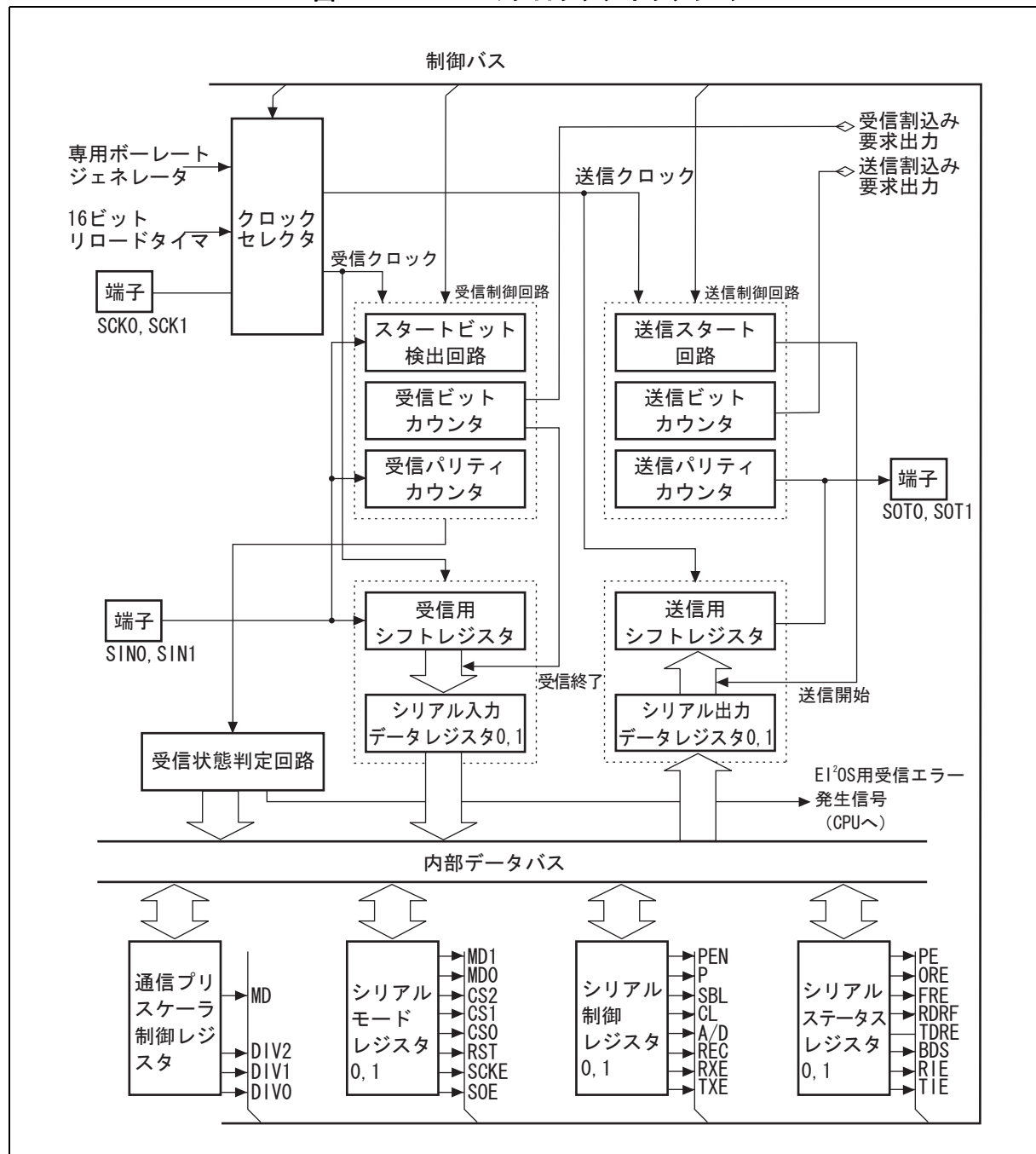
\*2: 受信時のストップビットは 1 ビットのみを検出します。

## 17.2 UART のブロックダイアグラム

UART は、以下のブロックで構成されています。

### ■ UART のブロックダイアグラム

図 17.2-1 UART のブロックダイアグラム



### ● ブロックダイヤグラム中の端子などの詳細

UART の実際の端子名および割込み要求番号は以下のとおりです。

- SIN0 端子 :P40/SIN0
- SIN1 端子 :P60/SIN1
- SCK0 端子 :P42/SCK0
- SCK1 端子 :P62/SCK1
- SOT0 端子 :P41/SOT0
- SOT1 端子 :P61/SOT1
- 送信割込み番号 0:#40 (28<sub>H</sub>)
- 受信割込み番号 0:#39 (27<sub>H</sub>)
- 送信割込み番号 1:#38 (26<sub>H</sub>)
- 受信割込み番号 1:#37 (25<sub>H</sub>)

### ● クロックセクタ

専用ボーレートジェネレータ, 外部入力クロック, 内部タイマ(16 ビットリロードタイマ) から供給されるクロックからの送受信クロックを選択してください。

### ● 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で構成されます。受信ビットカウンタは受信データをカウントして, 指定されたデータ長の 1 フレームのデータ受信を完了すると, 受信割込み要求を出力します。

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, 受信データをシリアル入力データレジスタへ指定された転送速度に応じて 1 ビットずつシフトしながらシリアル入力データレジスタ (SIDR0, SIDR1) へ書き込みます。受信パリティカウンタは, 受信データのパリティビットを検出します。

### ● 送信制御回路

送信制御回路は, 送信ビットカウンタ, 送信スタート回路, 送信パリティカウンタで構成されます。送信ビットカウンタは送信データをカウントして, 指定されたデータ長の 1 フレームのデータ送信を完了すると, 送信割込み要求を出力します。送信スタート回路は, シリアル出力データレジスタ (SODR0, SODR1) への書き込みにより送信を開始します。送信パリティカウンタは, "パリティあり" の場合にパリティビットを生成します。

### ● 受信用シフトレジスタ

SIN 端子から入力された受信データを, 1 ビットずつシフトしながら取込み, 受信が終了すると, シリアル入力データレジスタ (SIDR0, SIDR1) へ受信データを転送します。

### ● 送信用シフトレジスタ

シリアル出力データレジスタ (SODR0, SODR1) に書き込まれたデータを送信用シフトレジスタ自身へ転送し, 1 ビットずつシフトしながら SOT 端子へ出力します。



● シリアルモードレジスタ 0, 1 (SMR0, SMR1)

このレジスタは、以下の動作を行います。

- 動作モードの選択
- クロック入力ソース（ボーレート）の選択
- 専用ボーレートジェネレータのセットアップ
- 専用ボーレートジェネレータを使用したときのクロック速度（クロック分周値）の選択
- シリアルデータおよびクロック端子出力の許可または禁止を設定
- UART の初期化

● シリアル制御レジスタ 0, 1 (SCR0, SCR1)

このレジスタは、以下の動作を行います。

- パリティの有無の設定
- パリティの種類の選択
- ストップビット長の設定
- データ長の設定
- 動作モード 1（非同期マルチプロセッサ）時におけるフレームデータ形式の選択
- エラーフラグのクリア
- 送信の許可または禁止の設定
- 受信の許可または禁止の設定

● シリアルステータスレジスタ 0, 1 (SSR0, SSR1)

送受信やエラーの状態を確認し、送受信割込み要求の許可または禁止を設定します。

● シリアル入力データレジスタ 0, 1 (SIDR0, SIDR 1)

受信データを保持します。シリアル入力データは変換されてこのレジスタに格納されます。

● シリアル出力データレジスタ 0, 1 (SODR0, SODR1)

送信データを設定します。このレジスタに書き込まれたデータは、シリアル変換されて出力されます。

● 通信プリスケアラ制御レジスタ (CDCR)

ボーレートジェネレータのボーレートを設定するレジスタです。通信プリスケアラの起動 / 停止、マシクロックの分周比を設定します。

# 17.3    UART の構成

UART の端子，割込み要因，レジスタの一覧および詳細を記載します。

## ■ UART の端子

UART で使用する端子は，汎用入出力ポートと共用します。  
端子の機能と UART で使用する場合の設定を表 17.3-1 に示します。

表 17.3-1    UART の端子

端子名	端子機能	UART の使用に必要な設定
SOT	汎用入出力ポート / シリアルデータ出力	出力許可に設定 ( SMR0, SMR1 レジスタの SOE:bit0=1 )
SCK	汎用入出力ポート / シリアルクロック入出力	クロック入力時はポート方向レジスタ ( DDR ) で入力ポートに設定
		クロック出力時は出力許可に設定 ( SMR0, SMR1 レジスタの SCKE:bit1=1 )
SIN	汎用入出力ポート / シリアルデータ入力	ポート方向レジスタ ( DDR ) で入力ポートに設定

## ■ UART の端子のブロックダイアグラム

< 参照 >        端子のブロックダイアグラムは，「第 9 章 I/O ポート」をご参照ください。

## ■ UART のレジスタ一覧

図 17.3-1 UART のレジスタと初期値の一覧

シリアル制御レジスタ0, 1 (SCR0, SCR1)	bit	15	14	13	12	11	10	9	8
		0	0	0	0	0	1	0	0
シリアルモードレジスタ0, 1 (SMR0, SMR1)	bit	7	6	5	4	3	2	1	0
		0	0	0	0	0	0	0	0
シリアルステータスレジスタ0, 1 (SSR0, SSR1)	bit	15	14	13	12	11	10	9	8
		0	0	0	0	1	0	0	0
シリアル入力データレジスタ0, 1 (SIDR0, SIDR1) /シリアル出力データレジスタ0, 1 (SODR0, SODR1)	bit	7	6	5	4	3	2	1	0
		X	X	X	X	X	X	X	X
(注意事項) 読出しの場合はSIDR0, SIDR1として機能し、書込みの場合はSODR0, SODR1として機能します。									
通信プリスケールコントロールレジスタ0, 1 (CDCR0, CDCR1)	bit	15	14	13	12	11	10	9	8
		0	X	X	X	0	0	0	0
X: 不定									

## ■ UART での割込み要求の発生

## ● 受信割込み

- 受信データがシリアル入力データレジスタ (SIDR0, SIDR1) へロードされると、シリアルステータスレジスタ (SSR0, SSR1) の受信データロードフラグビット (RDRF:bit12) に "1" が設定されます。受信割込みが許可されている場合は (RIE:bit9=1), 割込みコントローラへ受信割込み要求を発生します。
- フレーミングエラー、オーバランエラーまたはパリティエラーのいずれかが発生すると、発生したエラーに応じてシリアルステータスレジスタ (SSR0, SSR1) のフレーミングエラーフラグビット (FRE:bit13), オーバランエラーフラグビット (ORE:bit14), パリティエラーフラグビット (PE:bit15) に "1" が設定されます。受信割込みが許可されている場合は (RIE:bit9=1), 割込みコントローラへ受信割込み要求を発生します。

## ● 送信割込み

送信データがシリアル出力データレジスタ (SODR0, SODR1) から送信用シフトレジスタに転送されると、シリアルステータスレジスタ (SSR0, SSR1) の送信データエンプティフラグビット (TDRE:bit11) に "1" が設定されます。送信割込みが許可されている場合は (TIE:bit8=1), 割込みコントローラへ割込み要求を発生します。

17.3.1 シリアル制御レジスタ 0, 1 (SCR0, SCR1)

シリアル制御レジスタ 0, 1 (SCR0, SCR1) 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, 動作モード 1 (非同期マルチプロセッサモード) 時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, 送受信の許可または禁止を設定します。

■ シリアル制御レジスタ 0, 1 (SCR0, SCR1)

図 17.3-2 シリアル制御レジスタ 0, 1 (SCR0, SCR1)

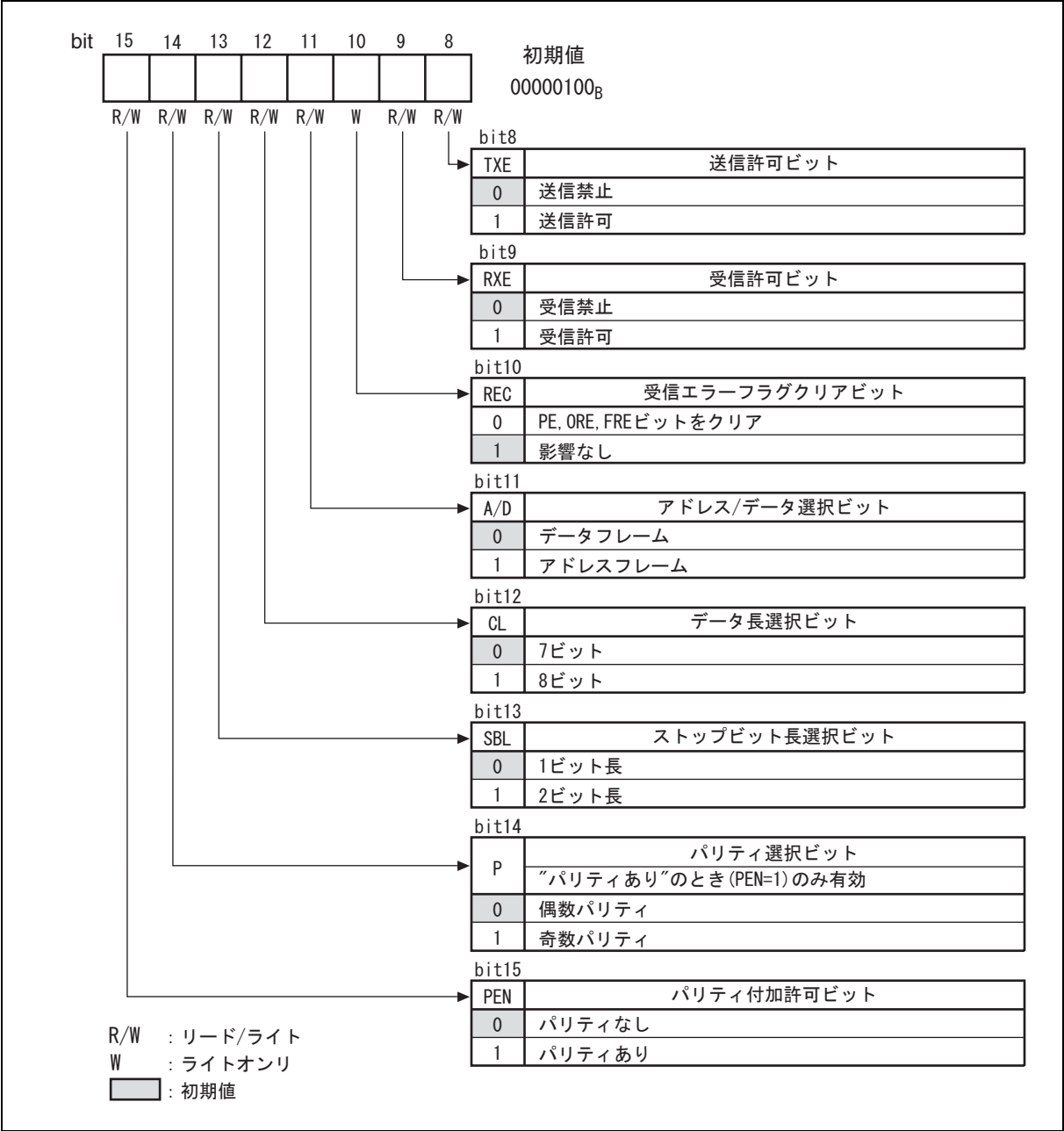


表 17.3-2 シリアル制御レジスタ 0, 1 (SCR0, SCR1) の機能

ビット名		機能
bit15	PEN: パリティ付加 許可ビット	パリティビットの付加（送信時）と検出（受信時）を行うかどうかを設定します。 （注意事項） 動作モード 1, 2（マルチプロセッサモード、同期モード）の場合はパリティビットは付加できません。必ず "0" を設定してください。
bit14	P: パリティ選択 ビット	"パリティビットあり"（PEN=1）を設定した場合、奇数パリティか偶数パリティのいずれかを設定してください。
bit13	SBL: ストップビッ ト長選択ビッ ト	動作モード 0, 1（マルチプロセッサモード、同期モード）の場合のストップビット（送信データのフレームエンドマーク）のビット長を設定します。 （注意事項） 受信時は、常にストップビットの 1 ビット目だけを検出します。
bit12	CL: データ長選択 ビット	送受信データのデータ長を設定します。 （注意事項） "7 ビット" を選択できるのは、動作モード 0（非同期ノーマルモード）の場合のみです。動作モード 1, 2（非同期マルチプロセッサモード、クロック同期モード）の場合は、必ず "8 ビット" を設定してください。
bit11	A/D: アドレス / データ選択 ビット	動作モード 1（非同期マルチプロセッサモード）の場合に、送受信するフレームのデータ形式を設定します。 このビットに "0" を設定した場合："データフレーム" が設定されます。 このビットに "1" を設定した場合："アドレスフレーム" が設定されます。
bit10	REC: 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（SSR0, SSR1）の受信エラーフラグ（bit15 ~ bit13: PE, ORE, FRE）を "0" にクリアします。 このビットに "0" を設定した場合：PE, ORE, FRE ビットがクリアされます。 このビットに "1" を設定した場合：影響しません。 リードした場合：常に "1" が読み出されます。 （注意事項） 受信割込みを "許可" に設定していれば（RIE:bit9=1）、PE, ORE, FRE ビットのいずれかに "1" が設定されている場合のみ、REC:bit10 に "0" を設定してください。
bit9	RXE: 受信許可ビッ ト	UART の受信を許可または禁止します。 このビットに "0" を設定した場合：受信は禁止されます。 このビットに "1" を設定した場合：受信は許可されます。 （注意事項） 受信中に受信を "禁止" に設定した場合は、受信中のデータをシリアル入力データレジスタに格納した後に受信が停止します。
bit8	TXE: 送信許可ビッ ト	UART の送信を許可または禁止します。 このビットに "0" を設定した場合：送信は禁止されます。 このビットに "1" を設定した場合：送信は許可されます。 （注意事項） 送信中に送信を禁止に設定した場合は、シリアル出力データレジスタ（SODR0, SODR1）のデータの送信が完了した後に送信が停止します。 "0" を設定する場合は、シリアル出力データレジスタ（SODR0, SODR1）にデータを書き込んだ後に非同期モードの場合であればボーレートの 1/16 時間、同期モードの場合であればボーレートと同じ時間以上待つてから設定してください。

17.3.2 シリアルモードレジスタ 0, 1 (SMR0, SMR1)

シリアルモードレジスタ 0, 1 (SMR0, SMR1) は、動作モードおよび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の許可または禁止を設定します。

■ シリアルモードレジスタ 0, 1 (SMR0, SMR1)

図 17.3-3 シリアルモードレジスタ 0, 1 (SMR0, SMR1)

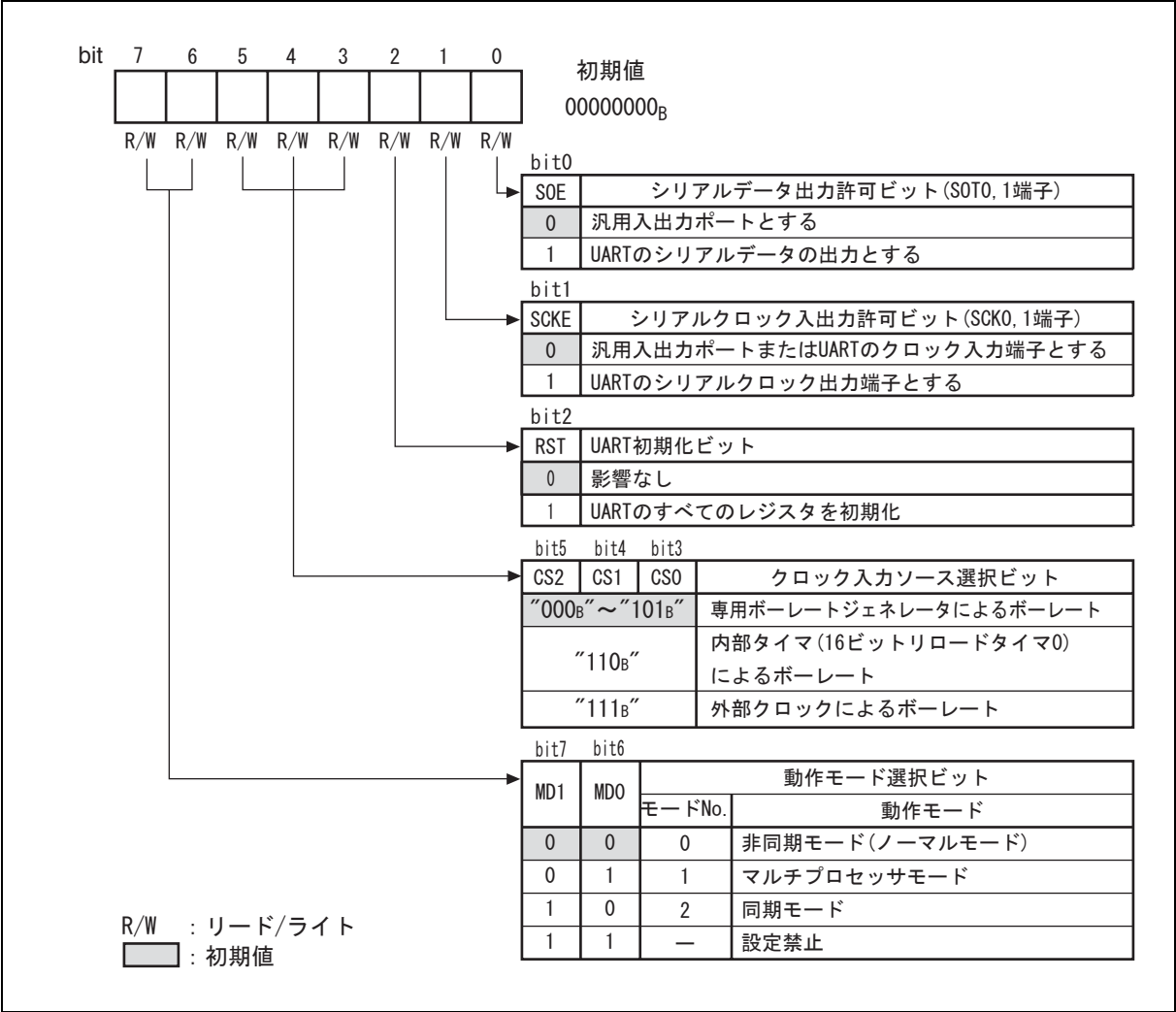


表 17.3-3 シリアルモードレジスタ 0, 1 (SMR0, SMR1) の機能

ビット名		機能
bit7, bit6	MD0, MD1: 動作モード選 択ビット	UART の動作モードを設定します。 (注意事項) 1) 動作モード 1 (非同期マルチプロセッサモード) では、マスタスレーブ型通信の、マスタとしてのみ使用できます。動作モード 1 は 9 ビット目のアドレス / データビットを受信できないので、スレーブとしては使用できません。 2) 動作モード 1 (非同期マルチプロセッサモード) では、パリティチェック機能は使用できないので、パリティ付加許可ビットには "パリティなし" (SCR0, SCR1 レジスタの PEN:bit15=0) を設定してください。
bit5 ~ bit3	CS0 ~ CS2: クロック入力 ソース選択 ビット	ボーレートのクロック入力ソースを設定します。 • クロック入力ソースは、外部クロック (SCK0, SCK1 端子), 内部タイマ (16 ビットリロードタイマ), 専用ボーレートジェネレータから選択します。 • ボーレートジェネレータを選択する場合は、ボーレートを設定してください。
bit2	RST: UART 初期化 ビット	UART のすべてのレジスタを初期化します。 このビットに "0" を設定した場合: 動作に影響ありません。 このビットに "1" を設定した場合: UART のすべてのレジスタを初期化します。
bit1	SCKE: シリアルク ロック入出力 許可ビット	シリアルクロックの入出力を切り換えます。 このビットに "0" を設定した場合: "汎用入出力ポート" または "シリアルクロック入力端子" が設定されま す。 このビットに "1" を設定した場合: "シリアルクロック出力端子" が設定されます。 (注意事項) 1) SCK0, SCK1 端子をシリアルクロック入力として使用する場合は、ポート 方向レジスタ (DDR) で端子を "入力ポート" に設定してください。ま た、クロック入力ソース選択ビットによって外部クロックを選択 (CS2 ~ CS0:bit5 ~ bit3=111 <sub>B</sub> ) してください。 2) シリアルクロック出力として使用する場合は、クロック入力ソース選択 ビットを外部クロック以外 (CS2 ~ CS0:bit5 ~ bit3=111 <sub>B</sub> 以外) に設定 してください。
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。 このビットに "0" を設定した場合: "汎用入出力ポート" が設定されます。 このビットに "1" を設定した場合: "シリアルデータ出力端子" が設定されま す。

### 17.3.3 シリアルステータスレジスタ 0, 1 (SSR0, SSR1)

シリアルステータスレジスタ 0, 1 (SSR0, SSR1) は, 送受信やエラー状態の確認, 割込みの許可または禁止を設定します。

■ シリアルステータスレジスタ 0, 1 (SSR0, SSR1)

図 17.3-4 シリアルステータスレジスタ 0, 1 (SSR0, SSR1)

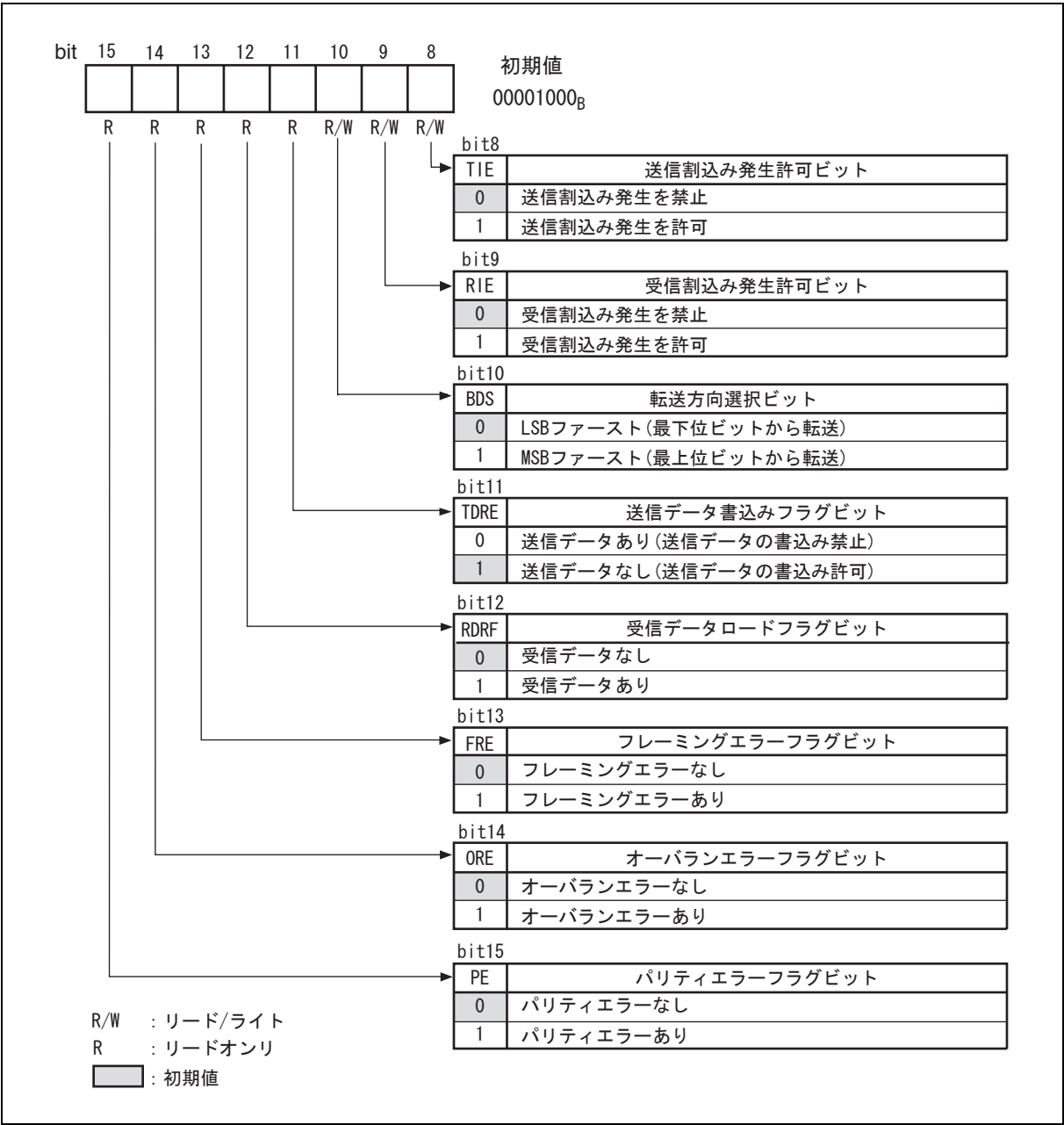




表 17.3-4 シリアルステータスレジスタ 0, 1 (SSR0, SSR1) の機能 ( 1 / 2 )

ビット名		機能
bit15	PE: パリティエ ラーフラグ ビット	<p>受信データのパリティエラーを検出します。</p> <ul style="list-style-type: none"> <li>パリティエラーが発生すると "1" が設定されます。</li> <li>受信エラーフラグクリアビット (SCR0, SCR1 レジスタの REC:bit10) に "0" を書き込むとクリアされます。</li> <li>受信割込みが許可されている場合は (RIE:bit9=1), パリティエラーが発生すると受信割込み要求が発生します。</li> <li>パリティエラーフラグビットが設定された場合は (PE:bit15=1), シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) のデータは無効です。</li> </ul>
bit14	ORE: オーバランエ ラーフラグ ビット	<p>受信時のオーバランエラーを検出します。</p> <ul style="list-style-type: none"> <li>オーバランエラーが発生すると "1" が設定されます。</li> <li>受信エラーフラグクリアビット (SCR0, SCR1 レジスタの REC:bit10) に "0" を書き込むとクリアされます。</li> <li>受信割込みが許可されている場合は (RIE:bit9=1), オーバランエラーが発生すると受信割込み要求が発生します。</li> <li>オーバランエラーフラグビットがセットされた場合は (ORE:bit14=1), シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) のデータは無効です。</li> </ul>
bit13	FRE: フレーミング エラーフラグ ビット	<p>受信データのフレーミングエラーを検出します。</p> <ul style="list-style-type: none"> <li>フレーミングエラーが発生すると "1" が設定されます。</li> <li>受信エラーフラグクリアビット (SCR0, SCR1 レジスタの REC:bit10) に "0" を書き込むとクリアされます。</li> <li>受信割込みの発生が許可されている場合は (bit9: RIE=1), フレーミングエラーが発生すると受信割込み要求が発生します。</li> <li>フレーミングエラーフラグビットがセット (FRE:bit13=1) された場合は, シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) のデータは無効です。</li> </ul>
bit12	RDRF: 受信データ ロードフラグ ビット	<p>シリアル入力データレジスタ 0, 1 の状態を示します。</p> <ul style="list-style-type: none"> <li>シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) へ受信データがロードされると, "1" が設定されます。</li> <li>シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) のデータを読み出すと "0" にクリアされます。</li> <li>受信割込みを許可した場合は (RIE:bit9=1), 受信データがシリアル入力データレジスタ 1 (SIDR1) へロードされると, 受信割込み要求が発生します。</li> </ul>
bit11	TDRE: 送信データ書 込みフラグ ビット	<p>シリアル出力データレジスタ 0, 1 の状態を示します。</p> <ul style="list-style-type: none"> <li>シリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ送信データを書き込むと, "0" にクリアされます。</li> <li>送信用シフトレジスタにデータがロードされ送信が開始されると "1" が設定されます。</li> <li>送信割込みを許可した場合は (TIE:bit8=1), シリアル出力データレジスタ 0, 1 (SODR0, SODR1) に書き込まれたデータが送信用シフトレジスタへロードされる (TDRE:bit11=1) と送信割込み要求が発生します。</li> </ul>
bit10	BDS: 転送方向選択 ビット	<p>シリアルデータの転送方向を設定します。</p> <p><b>このビットに "0" を設定した場合:</b> 最下位ビット側から転送します (LSB ファースト)。</p> <p><b>このビットに "1" を設定した場合:</b> 最上位ビット側から転送します (MSB ファースト)。</p> <p>(注意事項)</p> <p>シリアルデータレジスタへの読み出しおよび書き込み時にデータの上位側と下位側を入換えるため, シリアルアウトプットデータレジスタ (SODR0, SODR1) にデータを書き込んだ後に, 転送方向選択ビット (BDS) を書き換えた場合は, 書き込まれたデータは無効になります。</p>

表 17.3-4 シリアルステータスレジスタ 0, 1 (SSR0, SSR1) の機能 ( 2 / 2 )

ビット名		機能
bit9	RIE: 受信割込み発生許可ビット	<p>受信割込みを許可または禁止します。</p> <p><b>このビットに "1" を設定した場合：</b></p> <p>シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) に受信データがロードされるか (RDRF:bit12=1), 受信エラーが発生すると (PE:bit15=1 また ORE:bit14=1 また FRE:bit13=1), 受信割込み要求が発生します。</p>
bit8	TIE: 送信割込み発生許可ビット	<p>送信割込みを許可または禁止します。</p> <p><b>このビットに "1" を設定した場合：</b></p> <p>シリアル出力データレジスタ 0, 1 (SODR0, SODR1) に書き込まれたデータが送信用シフトレジスタへロードされると (TDRE:bit11=1), 送信割込み要求が発生します。</p>

### 17.3.4 シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1), シリアル出力データレジスタ 0, 1 (SODR0, SODR1)

シリアル入力データレジスタとシリアル出力データレジスタは同一アドレスに配置されています。読み出した場合は、シリアルデータ入力用レジスタとして機能し、書き込んだ場合は、シリアルデータ出力用レジスタとして機能します。

#### ■ シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1)

図 17.3-5 シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1)

bit 7	6	5	4	3	2	1	bit 0	初期値
D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX <sub>B</sub>
R	R	R	R	R	R	R	R	

R : リードオンリ  
X : 不定

シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) は、シリアルデータ受信用のデータバッファレジスタです。

- シリアルデータ入力端子 (SIN0, SIN1) 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、シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) へ格納されます。
- データ長が 7 ビットの場合は、上位 1 ビット (SIDR0, SIDR1:D7) は無効データとなります。
- 受信データが、シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) へ格納されると、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) に "1" が設定されます。受信割込みが許可されている場合は (SSR0, SSR1 レジスタの RIE:bit9=1) 割込みコントローラへ受信割込み要求を発生します。
- シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) は、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) が "1" の状態で読み出してください。受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) は、シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) を読み出すと、自動的に "0" にクリアされます。
- 受信エラーが発生 (SSR0, SSR1 レジスタの PE, ORE, FRE:bit15 ~ bit13 のいずれかが "1") した場合、シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) の受信データは無効となります。

## ■ シリアル出力データレジスタ 0, 1 (SODR0, SODR1)

図 17.3-6 シリアル出力データレジスタ 0, 1 (SODR0, SODR1)

bit 7	6	5	4	3	2	1	0	初期値
D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX <sub>B</sub>
W	W	W	W	W	W	W	W	

W : ライトオンリ  
X : 不定

シリアル出力データレジスタ 0, 1 (SODR0, SODR1) は、シリアルデータ送信用のデータバッファレジスタです。

- 送信動作が許可されている場合に (SCR0, SCR1 レジスタの TXE:bit8=1), 送信するデータをシリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込むと、送信データが送信用シフトレジスタに転送され、シリアルデータに変換されて、シリアルデータ出力端子 (SOT0, SOT1 端子) から送出されます。
- データ長が 7 ビットの場合、上位 1 ビット (SODR0, SODR1 レジスタの D7:bit7) は無効データとなります。
- 送信データ書込みフラグ (SSR0, SSR1 レジスタの TDRE:bit11) は、送信データがシリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込まれると、"0" にクリアされます。
- 送信用シフトレジスタへの転送が終了すると、"1" が設定されます。
- 送信データ書込みフラグ (SSR0, SSR1 レジスタの TDRE:bit11) が "1" の場合は、次の送信用データを書き込むことができます。送信割込みが許可 (SSR0, SSR1 レジスタの TIE:bit8=1) されている場合は、割込みコントローラへ送信割込み要求を発生します。次の送信データの書込みは、送信データ書込みフラグ (SCR0, SCR1 レジスタの TDRE:bit11) が "1" の状態で行ってください。

---

< 注意事項 > シリアル出力データレジスタは書込み専用のレジスタであり、シリアル入力データレジスタは読出し専用のレジスタです。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) 命令は使用できません。

---

## 17.3.5 通信プリスケアラ制御レジスタ (CDCR0, CDCR1)

通信プリスケアラ制御レジスタ 0, 1 (CDCR0, CDCR1) は, UART の専用ポーレートジェネレータのポーレートを設定します。

- 通信プリスケアラの起動または停止
- マシンクロックの分周比の設定

### ■ 通信プリスケアラ制御レジスタ 0, 1 (CDCR0, CDCR1)

図 17.3-7 通信プリスケアラ制御レジスタ 0, 1 (CDCR0, CDCR1)



表 17.3-5 通信プリスケアラ制御レジスタ 0, 1 (CDCR0, CDCR1) の機能

ビット名		機 能
bit15	MD: 通信プリスケアラ制御ビット	通信プリスケアラの動作を許可または停止します。 このビットに "0" を設定した場合: 停止します。 このビットに "1" を設定した場合: 許可します。
bit14 ~ bit12	未定義ビット	リードした場合: 値は不定です。 ライトした場合: 影響しません。
bit11	予約: 予約ビット	<ul style="list-style-type: none"> <li>必ず "0" を設定してください。</li> </ul>
bit10 ~ bit8	DIV2 ~ DIV0: 通信プリスケアラ分周比ビット	<ul style="list-style-type: none"> <li>マシンクロックの分周比を設定します。 (注意事項) 分周比を変更する場合は、クロック周波数を安定させるために、次の通信を開始するまで最低でも分周クロックの 2 周期分の時間をあけてください。</li> </ul>

## 17.4 UART の割込み

UART の割込みには、受信割込みと送信割込みがあり、以下に示す要因で割込み要求が発生できます。

- 受信データがシリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) にロードされた場合
- 受信エラー (パリティエラー, オーバランエラー, フレーミングエラー) が発生した場合
- 送信データがシリアル出力データレジスタ 0, 1 (SODR0, SODR1) から送信用シフトレジスタへ転送された場合

なお、それぞれの拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) に対応しています。

### ■ UART の割込み

UART の割込み制御ビットと割込み要因を表 17.4-1 に示します。

表 17.4-1 UART の割込み制御ビットと割込み要因

送受信	割込み要求 フラグビット	動作モード			割込み要因	割込み要因 許可ビット	割込み要求フラグの クリア
		0	1	2			
受信	SSR0, SSR1:RDRF				受信データをシリアル入 力データレジスタ 0, 1 (SIDR0, SIDR1) へロー ド	SSR0, SSR1: RIE	受信データの読出し
	SSR0, SSR1:ORE				オーバランエラー発生		受信エラーフラグク リアビット (SCR0, SCR1 レジスタの REC: bit10) への "0" 書込み
	SSR0, SSR1:FRE			×	フレーミングエラー発生		
	SSR0, SSR1:PE		×	×	パリティエラー発生		
送信	SSR0, SSR1:TDRE				シリアル出力データレジ スタ 0, 1 (SODR0, SODR 1) から送信データ転送 終了	SSR0, SSR1: TIE	送信データの書込み

: 有  
× : 無

### ● 受信割込み

受信割込みが許可されている場合に ( SSR0, SSR1 レジスタの RIE:bit9=1 ), データ受信完了 ( SSR0, SSR1 レジスタの RDRF:bit12=1 ), オーバランエラー ( SSR0, SSR1 レジスタの ORE:bit14=1 ), フレーミングエラー ( SSR0, SSR1 レジスタの FRE:bit13=1 ), パリティエラー ( SSR0, SSR1 レジスタの PE:bit15=1 ) のいずれかが発生すると割込みコントローラへ受信割込み要求が発生します。

受信データロードフラグ ( SSR0, SSR1 レジスタの RDRF:bit12 ) は, シリアル入力データレジスタ 0, 1 ( SDR0, SDR1 ) を読み出すと, 自動的に "0" にクリアされます。各受信エラーフラグ ( SSR0, SSR1 レジスタの PE, ORE, FRE:bit15 ~ bit13 ) は受信エラーフラグクリアビット ( SCR0, SCR1 レジスタの REC:bit10 ) に "0" を書き込めば, すべて "0" にクリアされます。

---

< 注意事項 > 受信エラー ( パリティエラー, オーバランエラー, フレーミングエラー ) が発生した場合は, 必要に応じてエラーを処理した後, 受信エラーフラグクリアビット ( SCR0, SCR1 レジスタの REC:bit10 ) に "0" を書き込み, 各受信エラーフラグをクリアしてください。

---

### ● 送信割込み

送信データがシリアル出力データレジスタ 0, 1 ( SODR0, SODR1 ) から送信用シフトレジスタへ送信されると, 送信データ書込みフラグビット ( SSR0, SSR1 レジスタの TDRE:bit11 ) に "1" が設定されます。

送信割込みが許可されている場合は ( SSR0, SSR1 レジスタの TIE:bit8=1 ), 割込みコントローラへ送信割込み要求が発生します。

## ■ UART の割込みと EI<sup>2</sup>OS

---

< 参照 > 割込み番号, 割込み制御レジスタ, 割込みベクタアドレスについては, 「第 7 章 割込み」をご参照ください。

---

## ■ UART の EI<sup>2</sup>OS 機能

UART は EI<sup>2</sup>OS 機能に対応しています。受信および送信の各割込みで別々に EI<sup>2</sup>OS を起動できます。

### ● 受信時

ほかのリソースの状態に関係なく, EI<sup>2</sup>OS を使用できます。

### ● 送信時

割込み制御レジスタ ( ICR13, 14 ) を, UART の送信割込みと共有しているので, UART の受信で割込みを使用しない場合に, EI<sup>2</sup>OS を起動できます。

## 17.4.1 受信割込み発生とフラグ設定のタイミング

受信時の割込みとしては、受信完了 (SSR0, SSR 1 レジスタの RDRF:bit12) および受信エラーの発生 (SSR0, SSR1 レジスタの PE, ORE, FRE:bit15 ~ bit13) があります。

### ■ 受信割込み発生とフラグ設定のタイミング

#### ● 受信データロードフラグと各受信エラーフラグの設定

データを受信する場合は、ストップビットの検出 (動作モード 0, 1: 非同期ノーマルモード, 非同期マルチプロセッサモード) またはデータの最終ビット (SIDR0, SIDR1 レジスタの D7:bit7) の検出 (動作モード 2: クロック同期モード) によって、受信データがシリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) へ格納されます。受信エラーが発生した場合はエラーフラグ (SSR0, SSR1 レジスタの PE, ORE, FRE:bit15 ~ bit13) が設定され、受信データロードフラグ (SSR0, SSR1 レジスタの RDRF:bit12) が設定されます。各動作モードともいずれかのエラーフラグが設定された場合は、受信したシリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) の受信データは無効です。

#### 動作モード 0 (非同期ノーマルモード) 時

ストップビットの検出時に、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) が設定されます。受信エラーが発生すると、エラーフラグ (SSR0, SSR1 レジスタの PE, ORE, FRE:bit15 ~ bit13) が設定されます。

#### 動作モード 1 (非同期マルチプロセッサモード) 時

ストップビットを検出すると、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) がセットされます。受信エラーが発生すると、エラーフラグ (SSR0, SSR1 レジスタの ORE, FRE:bit14, bit13) が設定されます。パリティエラー (SSR0, SSR1 レジスタの PE:bit15) は検出できません。

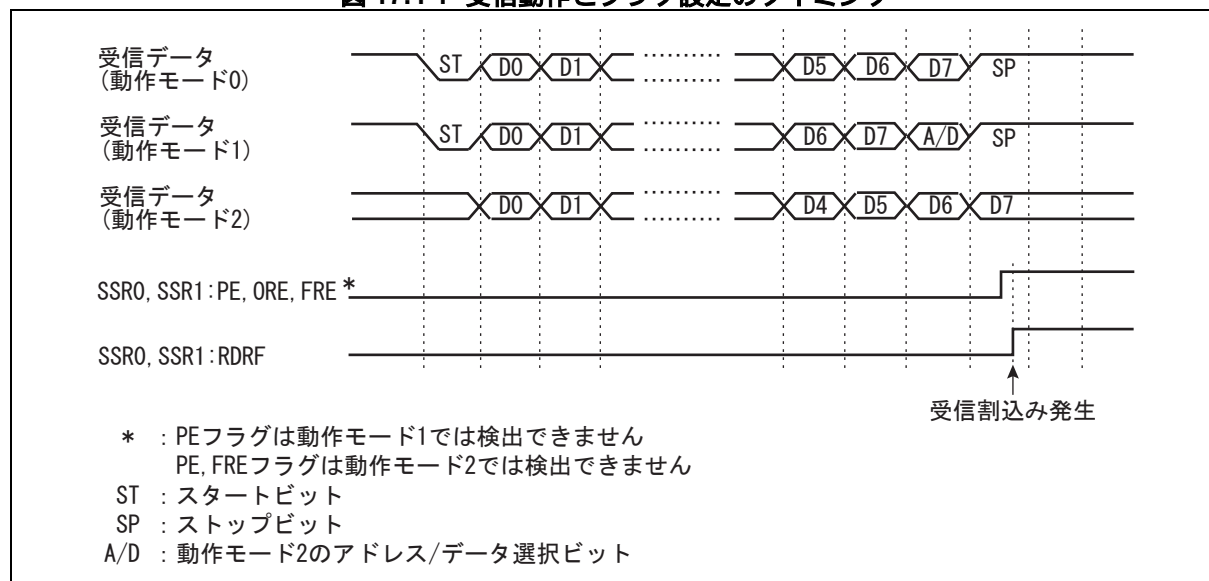
#### 動作モード 2 (クロック同期モード) 時

受信データの最終ビット (SIDR0, SIDR1 レジスタの D7:bit7) を検出すると、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) に "1" が設定されます。受信エラーが発生すると、エラーフラグ (SSR0, SSR1 レジスタの ORE:bit14) が設定されます。パリティエラー (SSR0, SSR1 レジスタの PE:bit15) およびフレーミングエラー (SSR0, SSR1 レジスタの FRE:bit13) は検出できません。

受信動作とフラグセットのタイミングを、図 17.4-1 に示します。



図 17.4-1 受信動作とフラグ設定のタイミング



### ● 受信割り込み要求発生のタイミング

受信割り込みが許可されている場合に ( SSR0, SSR1 レジスタの RIE:bit9=1 ), 受信データロードフラグ ( SSR0, SSR1 レジスタの RDRF:bit12 ), パリティエラーフラグ ( SSR0, SSR1 レジスタの PE:bit15 ), オーバランエラーフラグ ( SSR0, SSR1 レジスタの ORE:bit14 ), フレーミングエラーフラグ ( SSR0, SSR1 レジスタの FRE:bit13 ) のいずれかが設定されると, 割り込みコントローラへ受信割り込み要求を発生します。

## 17.4.2 送信割込み発生とフラグ設定のタイミング

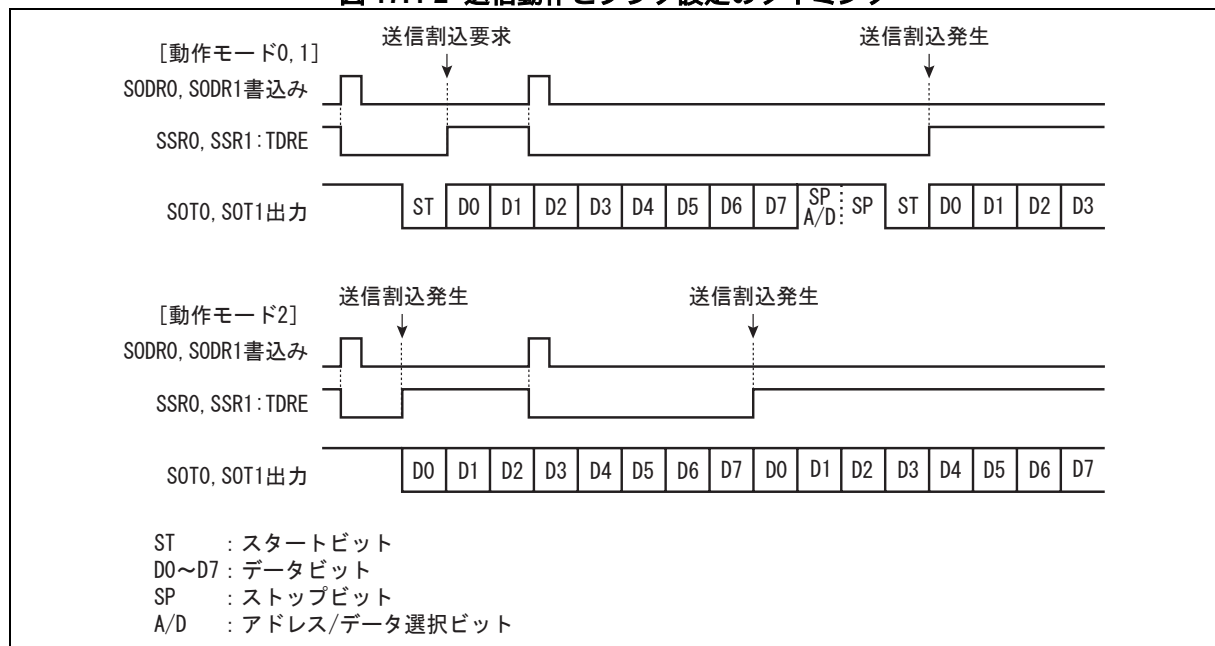
送信割込みは，シリアル出力データレジスタ 0, 1 (SODR0, SODR1) が空になり，次の送信データの書き込みが可能な状態になると発生します。

### ■ 送信割込み発生とフラグ設定のタイミング

#### ● 送信データエンプティフラグビットの設定とクリア

送信データ書き込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) は，シリアル出力データレジスタ 0, 1 (SODR0, SODR1) に書き込まれた送信データが，送信用シフトレジスタへロードされ，次のデータが書き込み可能な状態になると設定されます。シリアル出力データレジスタ 0, 1 (SODR0, SODR1) に次の送信データを書き込むと送信データ書き込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) は "0" にクリアされます。送信動作とフラグセットのタイミングを図 17.4-2 に示します。

図 17.4-2 送信動作とフラグ設定のタイミング



#### ● 送信割込み要求発生タイミング

送信割込みが許可 (SSR0, SSR1 レジスタの TIE:bit8=1) されている場合に，送信データ書き込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) が設定されると割込みコントローラへ送信割込み要求が発生します。

< 注意事項 > 送信中に送信を禁止 (SCR0, SCR1 レジスタの TXE:bit8=0, かつ動作モード 1 (非同期マルチプロセッサモード) 時の場合，受信禁止 (RXE:bit9 も含む) すると，送信データ書き込みフラグビットが設定され (SSR0, SSR1 レジスタの TDRF:bit11=1)，送信用シフトレジスタのシフト動作が停止してから UART の通信が禁止されます。送信が停止する前にシリアル出力データレジスタ 0, 1 (SODR0, SODR1) に書き込まれていた送信データは送信されます。

## 17.5 UART のボーレート

---

UART の送受信クロックとしては、以下のいずれかを選択します。

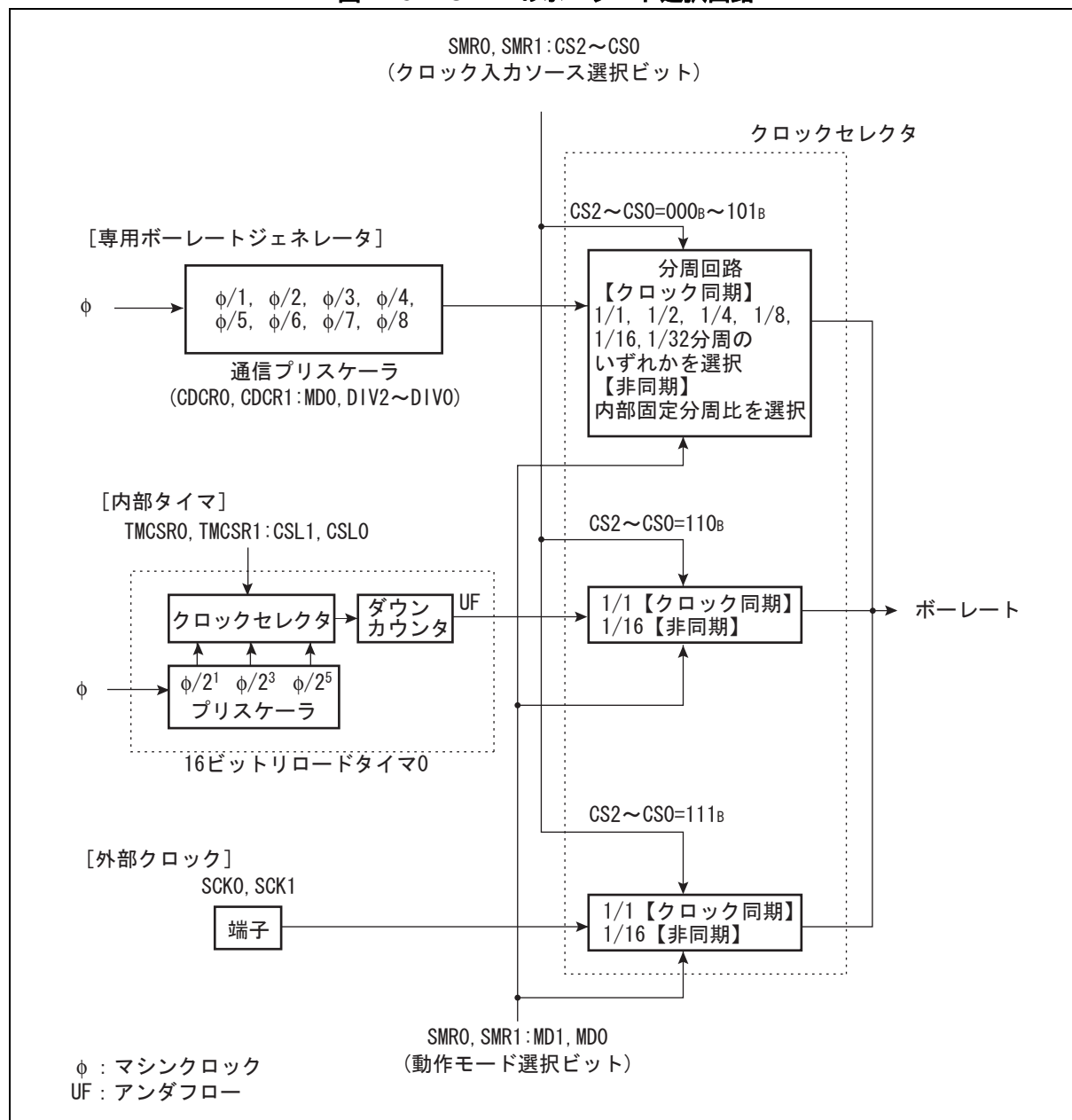
- 専用ボーレートジェネレータ
  - 内部クロック (16 ビットリロードタイマ出力)
  - 外部クロック (SCK0, SCK1 端子入力のクロック)
- 

### ■ UART のボーレートの選択

UART のボーレート選択回路は、図 17.5-1 のように構成されています。クロック入力ソースは以下の 3 種類の中から選択できます。

- 専用ボーレートジェネレータによるボーレート
  - UART に内蔵されている専用ボーレートジェネレータをクロック入力ソースとして使用する場合は、使用するボーレートに応じてシリアルモードレジスタ (SMR0, SMR1) の CS2 ~ CS0:bit5 ~ bit3 に "000<sub>B</sub>" ~ "101<sub>B</sub>" を設定してください。6 種類のボーレートから選択できます。
- 内部タイマによるボーレート
  - 16 ビットリロードタイマから供給される内部クロックを、クロック入力ソースとして使用する場合は、シリアルモードレジスタ (SMR0, SMR1) の CS2 ~ CS0:bit5 ~ bit3 に "10<sub>B</sub>" を設定してください。
  - ボーレートは、クロック同期の場合は "16 ビットリロードタイマから供給されるクロックのまま"、非同期の場合は "供給されるクロックを 16 分周した値" になります。
  - 16 ビットリロードタイマの設定値によって、自由にボーレートを設定できます。
- 外部クロックによるボーレート
  - UART のクロック入力端子 (SCK0, SCK1) から入力した外部クロックを、クロック入力ソースとして使用する場合は、シリアルモードレジスタ (SMR0, SMR1) の CS2 ~ CS0: bit5 ~ bit3 に "111<sub>B</sub>" を設定してください。
  - ボーレートは、クロック同期の場合は " 外部から供給されるクロック " のまま、クロック非同期の場合は " 入力されたクロックを 16 分周した値 " になります。

図 17.5-1 UART のボーレート選択回路



## 17.5.1 専用ボーレートジェネレータによるボーレート

UART の送受信クロックとして、専用ボーレートジェネレータの出力クロックを選択した場合のボーレートの設定を示します。

### ■ 専用ボーレートジェネレータによるボーレート

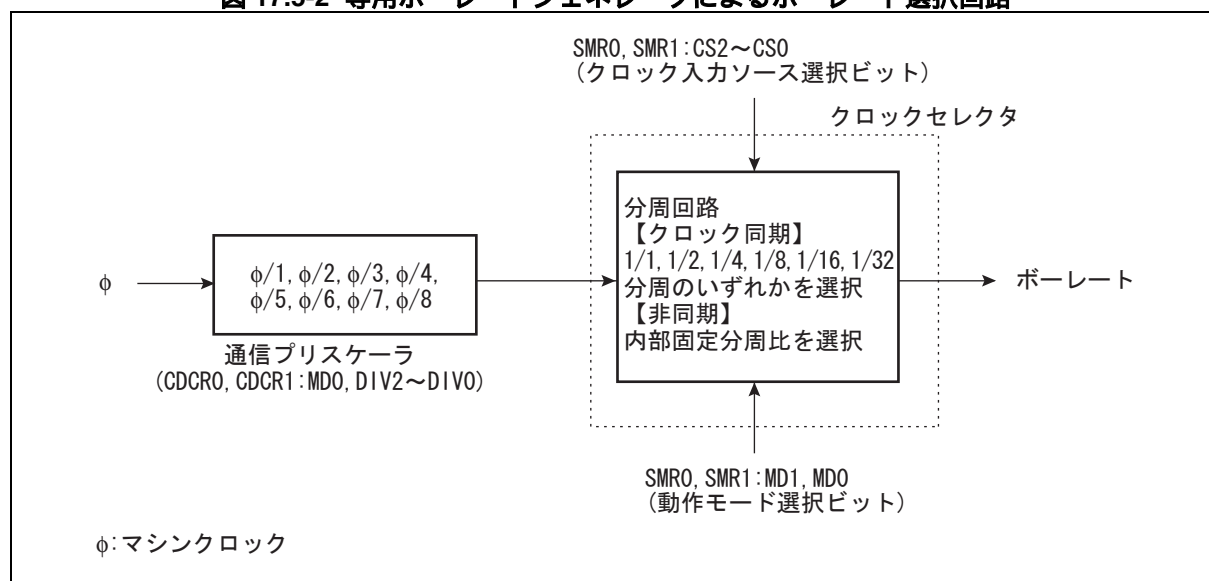
クロック入力ソース選択ビット（SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3）に "000<sub>B</sub>" ~ "101<sub>B</sub>" を設定することによって、専用ボーレートジェネレータによるボーレートが設定できます。

専用ボーレートジェネレータで送受信クロックを生成する場合は、マシンのクロック周波数が通信プリスケアラで分周された後、クロックセクタで選択されたクロック入力ソースに対して分周比を選択して、ボーレートを決定します。

通信プリスケアラによるマシンのクロックの分周比は、非同期とクロック同期とに共通ですが、ボーレート決定の分周比は、非同期とクロック同期とで異なります。

専用ボーレートジェネレータによるボーレート選択回路を図 17.5-2 に示します。

図 17.5-2 専用ボーレートジェネレータによるボーレート選択回路



#### ● ボーレート計算式

非同期ボーレート =  $\phi \times \text{div} \times (\text{非同期転送クロック分周比})$

クロック同期ボーレート =  $\phi \times \text{div} \times (\text{クロック同期転送クロック分周比})$

$\phi$  : マシンのクロック

div: 通信プリスケアラ分周比

● 通信プリスケアラによる分周比（非同期，クロック同期に共通）

マシンのクロックの分周比は，通信プリスケアラレジスタの分周比選択ビット（CDCR0, CDCR1 レジスタの DIV2 ~ DIV0:bit10 ~ bit8）で設定します。

表 17.5-1 通信プリスケアラによる分周比

MD	DIV2	DIV1	DIV0	div
0	-	-	-	停止
1	0	0	0	1 分周
1	0	0	1	2 分周
1	0	1	0	3 分周
1	0	1	1	4 分周
1	1	0	0	5 分周
1	1	0	1	6 分周
1	1	1	0	7 分周
1	1	1	1	8 分周

div: 通信プリスケアラ分周比

● ボーレート（非同期）

非同期モードのボーレートは，通信プリスケアラの出力クロックを使用して生成されます。分周比は，クロック入力ソース選択ビット（SMR0, SMR1 レジスタの CS2 ~ CS0: bit5 ~ bit3）で設定してください。

表 17.5-2 ボーレート（非同期）

CS2	CS1	CS0	非同期（スタート / ストップ同期）	算出式
0	0	0	76,923bps	$(\phi / \text{div}) / (8 \times 13 \times 2)$
0	0	1	38,461bps	$(\phi / \text{div}) / (8 \times 13 \times 4)$
0	1	0	19,230bps	$(\phi / \text{div}) / (8 \times 13 \times 8)$
0	1	1	9,615bps	$(\phi / \text{div}) / (8 \times 13 \times 16)$
1	0	0	500kbps	$(\phi / \text{div}) / (8 \times 2 \times 2)$
1	0	1	250kbps	$(\phi / \text{div}) / (8 \times 2 \times 4)$

$\phi$ : マシンのクロック周波数

div: 通信プリスケアラ分周比

## ● ボーレート（同期）

同期モードのボーレートは、通信プリスケアラの出力クロックを 1, 2, 4, 8, 16, 32 分周して生成されます。分周比は、クロック入力ソース選択ビット (SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3) で設定してください。

表 17.5-3 ボーレート（クロック同期）

CS2	CS1	CS0	CLK 同期	算出式
0	0	0	2Mbps	$(\phi / \text{div}) / 1$
0	0	1	1Mbps	$(\phi / \text{div}) / 2$
0	1	0	500kbps	$(\phi / \text{div}) / 4$
0	1	1	250kbps	$(\phi / \text{div}) / 8$
1	0	0	125kbps	$(\phi / \text{div}) / 16$
1	0	1	62.5kbps	$(\phi / \text{div}) / 32$

$\phi$ : マシンクロック周波数

div: 通信プリスケアラ分周比

## 17.5.2 内部タイマ（16 ビットリロードタイマ）によるポーレート

UART のクロック入力ソースとして、16 ビットリロードタイマ 0 から供給される内部クロックを選択する場合の設定とポーレートの計算式を示します。

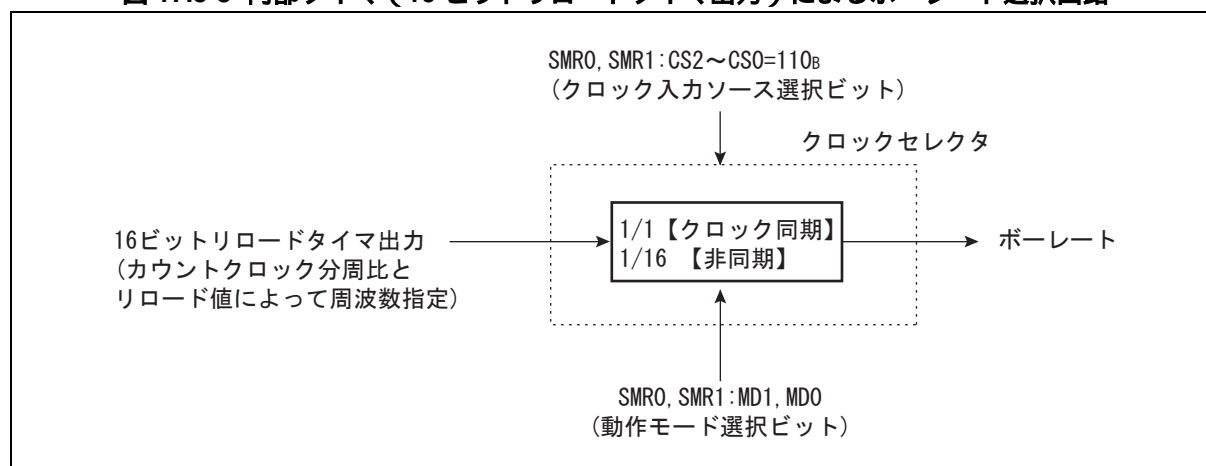
### ■ 内部タイマ（16 ビットリロードタイマ出力）によるポーレート

クロック入力ソース選択ビット（SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3）に "110<sub>B</sub>" を設定することによって、内部タイマ（16 ビットリロードタイマ出力）によるポーレートを設定できます。ポーレートは、16 ビットリロードタイマのカウントクロック分周比とリロード値を選択することによって自由に設定できます。

内部タイマによるポーレートの選択回路を図 17.5-3 に示します。

- クロック入力ソースとして内部タイマ（16 ビットリロードタイマ）を選択した（SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3）場合、16 ビットリロードタイマ出力端子（TOT）は内部接続されているので、外部クロック入力端子（SCK）に外部接続する必要はありません。
- 16 ビットリロードタイマ出力端子（TOT）は、ほかで使用していなければ汎用入出力ポートとして使用できます。

図 17.5-3 内部タイマ（16 ビットリロードタイマ出力）によるポーレート選択回路





## ● ボーレート計算式

$$\text{非同期ボーレート} = \frac{\phi/N}{16 \times 2 \times (n+1)} \text{ bps}$$

$$\text{クロック同期ボーレート} = \frac{\phi/N}{2 \times (n+1)} \text{ bps}$$

$\phi$ : マシニングロック周波数

N:16 ビットリロードタイマのプリスケアラによる分周比 ( $2^1, 2^3, 2^5$ )

n:16 ビットリロードタイマのリロード値 (0 ~ 65535)

- ボーレートとリロードレジスタ設定値の設定例 (マシニングロック周波数 :7.3728MHz の場合)

表 17.5-4 ボーレートとリロード値

ボーレート ( bps )	リロード値			
	非同期 ( 調歩同期 )		クロック同期	
	N=2 <sup>1</sup> ( マシニング ロックの 2 分周 )	N=2 <sup>3</sup> ( マシニング ロックの 8 分周 )	N=2 <sup>1</sup> ( マシニング ロックの 2 分周 )	N=2 <sup>3</sup> ( マシニング ロックの 8 分周 )
38,400	2	-	47	11
19,200	5	-	95	23
9,600	11	2	191	47
4,800	23	5	383	95
2,400	47	11	767	191
1,200	95	23	1,535	383
600	191	47	3,071	767
300	383	95	6,143	1,535

N:16 ビットリロードタイマのプリスケアラによる分周比

- : 設定不可

### 17.5.3 外部クロックによるボーレート

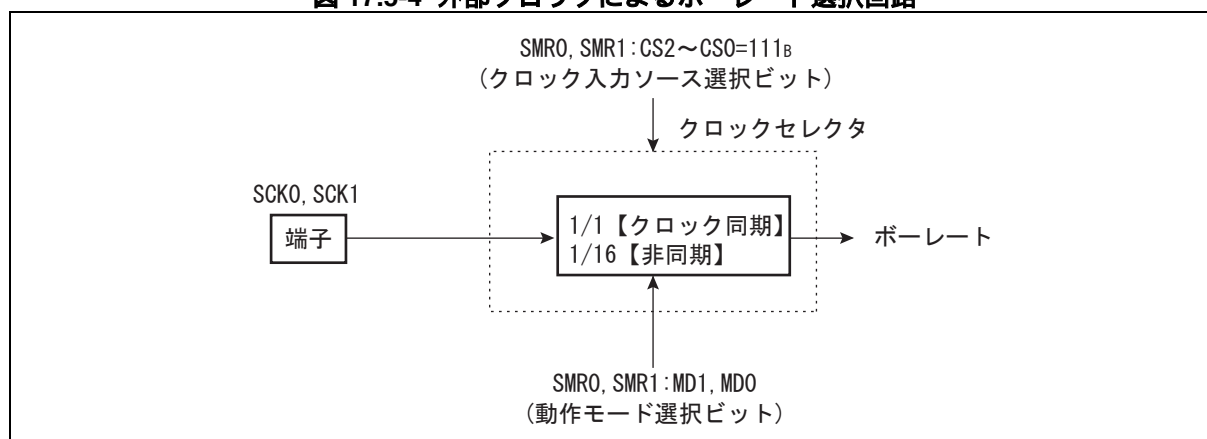
UART の送受信クロックとして外部クロックを選択した場合の設定について説明します。

#### ■ 外部クロックによるボーレート

外部クロック入力によるボーレートを選択する場合は、以下の設定が必要です。

- シリアルモードレジスタのクロック入力ソース選択ビット (SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3) に "111<sub>B</sub>" を設定します。
- SCK0, SCK1 端子をポート方向レジスタ (DDR) で入力ポートに設定します。
- シリアルクロック入出力許可ビットを (SMR0, SMR1 レジスタの SCKE:bit1) に "0" を設定します。
- SCK0, SCK1 端子から入力された外部クロックをもとに、ボーレートを設定します。内部の分周比は固定されているので、ボーレートを変更する場合は外部の入力クロックで変更する必要があります。

図 17.5-4 外部クロックによるボーレート選択回路



#### ● ボーレート計算式

非同期ボーレート =  $f/16$  bps

クロック同期ボーレート =  $f$  bps

f: 外部クロック周波数 (f は最大で 2MHz)

## 17.6 UART の動作説明

UART 機能には、双方向シリアル通信機能（動作モード 0, 2: 非同期ノーマルモード、クロック同期モード）と、マスタ/スレーブ型接続の通信機能（動作モード 1: 非同期マルチプロセッサモード）があります。

### ■ UART の動作

#### ● 動作モード

UART の動作モードは 3 種類があり、CPU 間の接続方式やデータ通信方式を設定できます。

UART の動作モードを、表 17.6-1 に示します。

表 17.6-1 UART 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
		パリティなし	パリティあり		
0	非同期モード (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 <sup>*2</sup>
1	マルチプロセッサモード	8 ビット + 1 ビット <sup>*1</sup>	-	非同期	
2	クロック同期モード	8 ビット	-	クロック同期	なし

- : 設定不可

<sup>\*1</sup>: "+1" は通信制御に使用するアドレス / データ選択ビット (SCR0, 1 レジスタの A/D:bit11) です。

<sup>\*2</sup>: 受信時のストップビットは 1 ビットのみを検出します。

< 注意事項 > UART の動作モード 1 (非同期マルチプロセッサモード) は、マスタ/スレーブ型接続時のマスタ時のみ使用できます。

#### ● CPU 間接続方式

CPU 間接続方式としては、1 対 1 接続とマスタ/スレーブ型接続のどちらかを選択できます。いずれの方式でも、データ長、パリティ有無、同期方式は、すべての CPU 間で統一しておく必要があります。動作モードは以下のように選択します。

- 1 対 1 接続では、2 つの CPU 間で動作モード 0, 2 (ノーマルモード、クロック同期モード) のいずれか同じ方式を採用する必要があります。非同期方式では動作モード 1: 非同期マルチプロセッサモード (SMR0, SMR1 レジスタの MD1, MD0:bit7, bit6=00<sub>B</sub>) を、クロック同期方式では動作モード 2: クロック同期モード (SMR0, SMR1 レジスタの MD1, MD0:bit7, bit6=10<sub>B</sub>) を設定してください。
- マスタ/スレーブ型接続では、動作モード 1: 非同期マルチプロセッサモード (SMR0, SMR1 レジスタの MD1, MD0:bit7, bit6=01<sub>B</sub>) を設定します。動作モード 1 (非同期マルチプロセッサモード) を設定した場合は、マスタとして使用してください。なお、この接続時は、パリティなしとデータ長 (8 ビット) を設定してください。

● 同期方式

非同期方式（調歩同期）またはクロック同期方式を選択します。

● 信号方式

UART は、NRZ（Non Return to Zero）形式のデータだけを扱えます。

● 送受信動作の開始

- シリアル制御レジスタの送信動作許可ビット（SCR0, SCR1 レジスタの TXE:bit8）に "1" を設定すると、送信動作を開始します。
- シリアル制御レジスタの受信動作許可ビット（SCR0, SCR1 レジスタの RXE:bit9）に "1" を設定すると、受信動作を開始します。

● 送受信動作の停止

- シリアル制御レジスタの送信動作許可ビット（SCR0, SCR1 レジスタの TXE:bit8）に "0" を設定すると、送信動作を停止します。
- シリアル制御レジスタの受信動作許可ビット（SCR0, SCR1 レジスタの RXE:bit9）に "0" を設定すると、受信動作を停止します。

● 送受信中の停止

- 受信（受信用シフトレジスタへのデータ入力中）に受信動作を禁止（SCR0, SCR1 レジスタの RXE:bit9=0）した場合は、受信中のフレームの受信を完了して、シリアル入力データレジスタ 0, 1（SIDR0, SIDR1）へ受信データを格納してから受信動作を停止します。
- 送信（送信用シフトレジスタからのデータが出力中）に送信動作を禁止（SCR0, SCR1 レジスタの TXE:bit8=0）した場合は、シリアル出力データレジスタ 0, 1（SODR0, SODR1）から送信用シフトレジスタへ 1 フレームの送信が完了してから送信動作を停止します。

## 17.6.1 非同期モード（動作モード 0, 1）時の動作

UART を動作モード 0（非同期ノーマルモード）または動作モード 1（非同期マルチプロセッサモード）で使用する場合は、通信方式は非同期となります。

### ■ 非同期モードでの動作

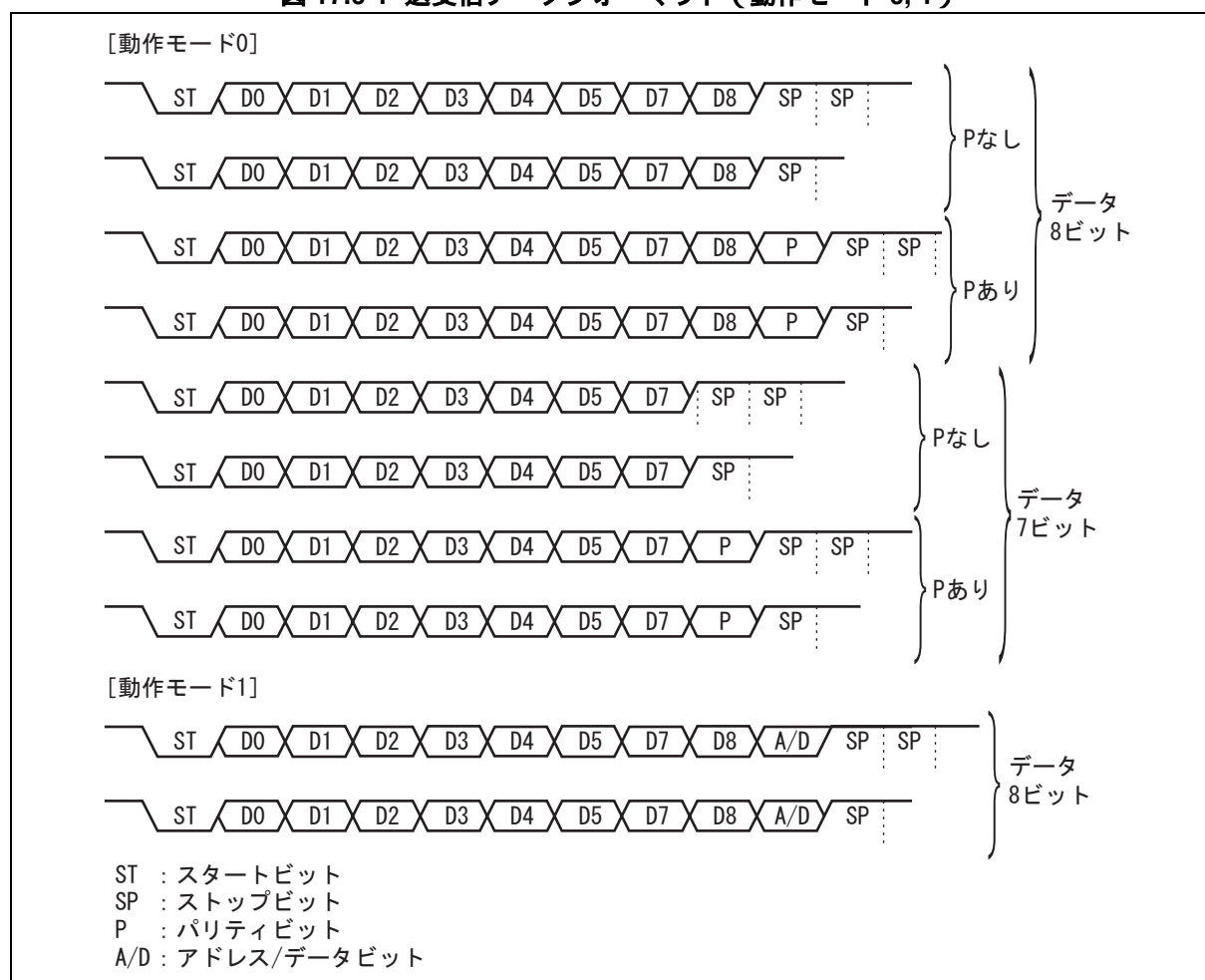
#### ● 送受信データフォーマット

送受信は、必ずスタートビット（"L" レベル）から始まり、指定されたデータビット長の送受信は LSB ファーストで行われ、ストップビット（"H" レベル）で終了します。

- 動作モード 0（非同期ノーマルモード）時は、データ長として 7 ビットまたは 8 ビットを選択します。パリティビットは、あり / なしを選択できます。
- 動作モード 1（非同期マルチプロセッサモード）時は、データ長は 8 ビットに固定されます。パリティビットは付加されません。9 ビット目に、アドレス / データビット（SCR0, SCR1 レジスタの A/D:bit11）が付加されます。

非同期モードの送受信データフォーマットを図 17.6-1 に示します。

図 17.6-1 送受信データフォーマット（動作モード 0, 1）



### ● 送信動作

- 送信データは、送信データ書込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) に "1" が設定されている状態で、シリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込みます。
- 送信データを書き込み、シリアル制御レジスタの送信動作許可ビット (SCR0, SCR1 レジスタの TXE:bit8) に "1" を設定すると、送信を開始します。
- 送信データをシリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込むと、送信データ書込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) は、一度 "0" にクリアされます。
- 送信データをシリアル出力データレジスタ 0, 1 (SODR0, SODR1) から送信用シフトレジスタへ書き込むと、送信データ書込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) には再度 "1" が設定されます。
- 送信割込み許可ビット (SSR0, SSR1 レジスタの TIE:bit8) に "1" を設定していた場合は、送信データ書込みフラグビット (SSR0, SSR1 レジスタの TDRE:bit11) に "1" が設定されると、送信割込み要求が発生します。割込み処理において、次の送信データをシリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込むことができます。

### ● 受信動作

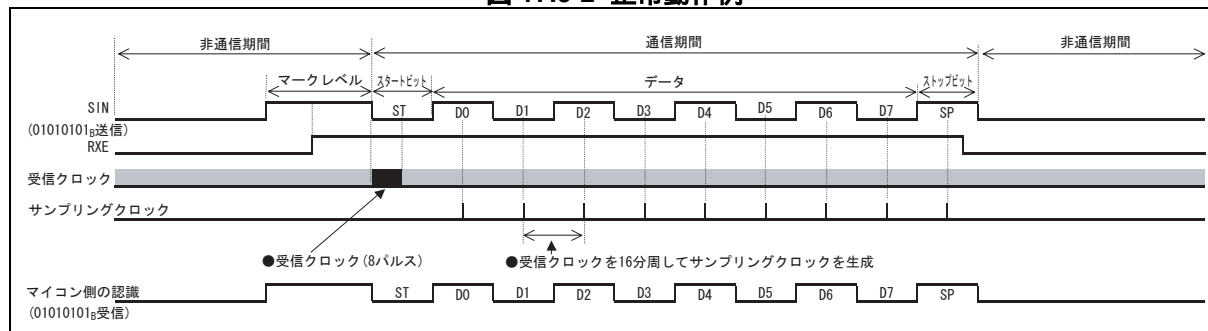
- 受信動作は、受信動作の許可が設定されている場合は (SCR0, SCR1 レジスタの RXE:bit9=1)、常に行われています。
- UART は受信データのスタートビットを検出すると、シリアル制御レジスタ 0, 1 (SCR0, SCR1) に設定しているデータフォーマットに従って、1 フレームのデータを受信しシリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) に格納します。
- 1 フレームのデータの受信が完了すると、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) に "1" が設定されます。
- 受信データを読み出す場合は、1 フレームデータの受信完了後に、シリアルステータスレジスタ 0, 1 (SSR0, SSR1) のエラーフラグの状態を確認し、正常に受信していれば、シリアル入力データレジスタ 0, 1 (SIDR0, SIDR1) から受信データを読み出してください。受信エラーが発生している場合は、エラー処理を実行してください。
- 受信データを読み出すと、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) は "0" にクリアされます。

### ● スタートビット検出方法

スタートビットを検出するには、以下のように設定してください。

- 通信期間直前は通信線を必ず "H" (マークレベルを付加) にしてください。
- 通信線が "H" (マークレベル) の期間に、受信許可 (RXE=H) にしてください。
- 非通信期間中 (マークレベルを除く) は、受信許可 (RXE=H) にしないでください。正しいデータが受信できません。
- ストップビット検出後 (RDRF フラグが "1" にセットされた後)、通信線が "H" (マークレベル) の間に受信禁止 (RXE=L) にしてください。

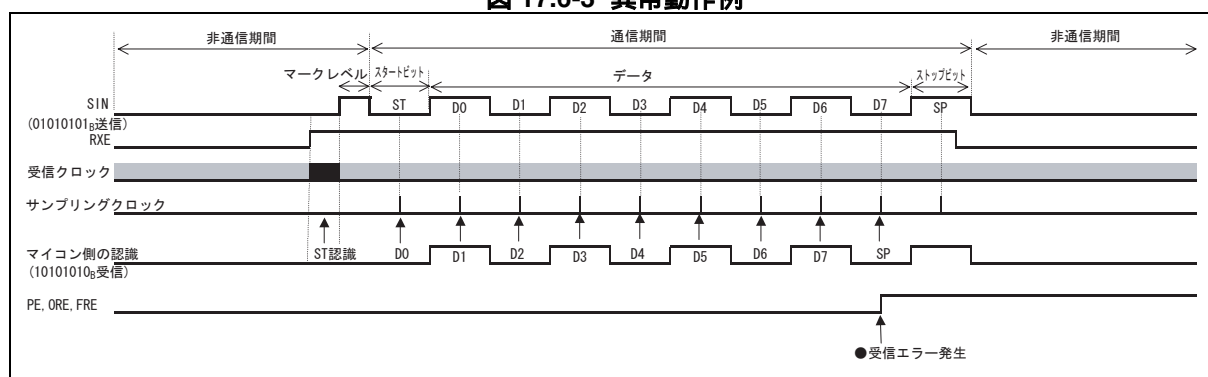
図 17.6-2 正常動作例



以下の例のようなタイミングで受信許可に設定しますとマイコン側で入力データ (SIN) が正しく認識されませんので注意してください。

- 通信線が "L" の期間に、受信許可 (RXE=H) に設定した場合の動作例

図 17.6-3 異常動作例



### ● ストップビット

1 ビットまたは 2 ビット長を選択できます。受信側では、常に最初の 1 ビットだけを検出します。

### ● エラー検出

- 動作モード 0 (非同期ノーマルモード) 時は、パリティエラー、オーバランエラー、フレームエラーを検出できます。
- 動作モード 1 (非同期マルチプロセッサモード) 時は、オーバランエラーとフレームエラーを検出できます。パリティエラーは検出できません。

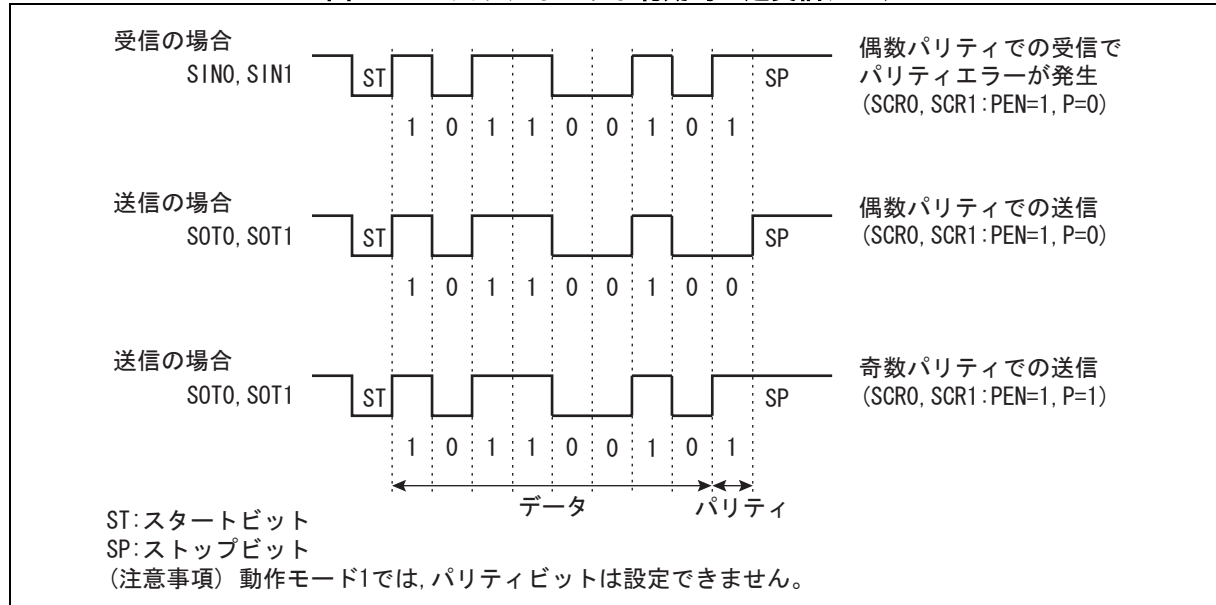
### ● パリティビット

パリティビットの付加は、動作モード 0 (非同期ノーマルモード) の場合のみ設定できます。パリティ付加許可ビット (SCR0, SCR1 レジスタの PEN:bit15) でパリティの有無を、パリティ選択ビット (SCR0, SCR1 レジスタの P:bit14) で偶数パリティ / 奇数パリティを設定できます。

動作モード 1 (非同期マルチプロセッサモード) 時は、パリティビットは付加できません。

パリティビット有効時の送受信データを図 17.6-4 に示します。

図 17.6-4 パリティビット有効時の送受信データ





## 17.6.2 クロック同期モード（動作モード 2）時の動作

UART を動作モード 2 で使用する場合は、クロック同期方式で通信を行います。

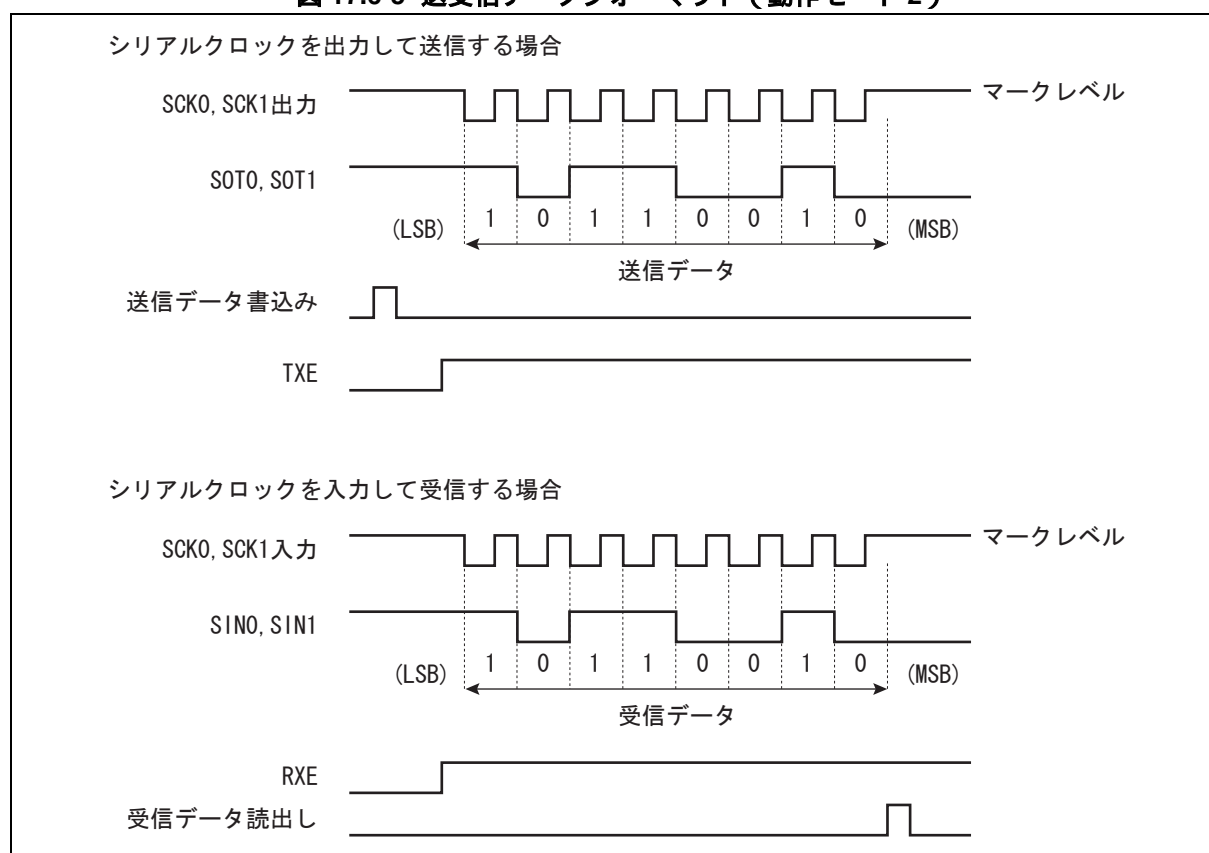
### ■ クロック同期モードでの動作

#### ● 送受信データフォーマット

クロック同期モード時は、8 ビットのデータを LSB ファーストで送受信します。送受信データには、スタートビット、ストップビットは付加されません。

クロック同期モードの送受信データフォーマットを図 17.6-5 に示します。

図 17.6-5 送受信データフォーマット（動作モード 2）



#### ● クロック供給

クロック同期方式では、送受信を行うフレームのビット数に等しい数のクロックの供給が必要になります。

- 内部クロック（専用ポーレートジェネレータまたは内部タイマ）を選択している場合（SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3=000<sub>B</sub> ~ 101<sub>B</sub> または 110<sub>B</sub>）は、データを送信するとデータ受信用同期クロックが自動的に生成されます。
- 外部クロックを選択している場合（SMR0, SMR1 レジスタの CS2 ~ CS0:bit5 ~ bit3=111<sub>B</sub>）は、UART のシリアル出力データレジスタ 0, 1（SODR0, SODR1）に送信データがあること（SSR0, SSR1 レジスタの TDRE:bit11=0）を確認してから、正確に 1 バイト分のクロックを外部から供給する必要があります。また、送信前と送信後は、必ずマークレベル（"H" レベル）に戻る必要があります。

● エラー検出

オーバランエラーのみ検出できます。パリティエラー、フレーミングエラーは検出できません。

● レジスタの設定

クロック同期モード（動作モード 2）を使用し、送信側から受信側へシリアルデータを送る場合の制御レジスタは表 17.6-2 のように設定します。

表 17.6-2 制御レジスタの設定

レジスタ名	ビット名	設定内容	
		送信側（シリアルクロックを出力）	受信側（シリアルクロックを入力）
シリアルモードレジスタ 0, 1 (SMR0, SMR1)	MD1, MD0	クロック同期モードを設定 (MD1, MD0=10 <sub>B</sub> )	
	CS2, CS1, CS0	クロック入力ソースを設定 ・ 専用ボーレートジェネレータ (CS2 ~ CS0=000 <sub>B</sub> ~ 101 <sub>B</sub> ) ・ 内部タイマ (CS2 ~ CS0=110 <sub>B</sub> )	クロック入力ソースを設定 ・ 外部クロック (CS2 ~ CS0=111 <sub>B</sub> )
	SCKE	シリアルクロック出力を設定 (SCKE=1)	シリアルクロック入力を設定 (SCKE=0)
	SOE	シリアルデータ出力端子に設定 (SOE=1)	汎用入出力ポートに設定 (SOE=0)
シリアル制御レジスタ 0, 1 (SCR0, SCR1)	PEN	パリティビットを付加しない (PEN=0)	
	CL	データ長 8 ビット (CL=1)	
	REC	エラーフラグの初期化 (REC=0)	
	TXE	送信動作を許可 (TXE=1)	送信動作を禁止 (TXE=0)
	RXE	受信動作を禁止 (RXE=0)	受信動作を許可 (RXE=1)
シリアルステータスレジスタ 0, 1 (SSR0, SSR1)	TIE	送信割込みを許可 (TIE=1)	送信割込みを禁止 (TIE=0)
	RIE	受信割込みを禁止 (RIE=0)	受信割込みを許可 (RIE=1)

● 通信開始

シリアル出力データレジスタ 0, 1 (SODR0, SODR1) に送信データを書き込むと、通信が開始します。受信のみ行う場合でも通信を開始するには、必ず仮の送信データをシリアル出力データレジスタ 0, 1 (SODR0, SODR1) へ書き込む必要があります。

● 通信終了

1 つのフレームのデータ送受信が終了すると、受信データロードフラグビット (SSR0, SSR1 レジスタの RDRF:bit12) に "1" が設定されます。データを受信した場合は、オーバランエラーフラグビット (SSR0, SSR1 レジスタの ORE:bit14) をチェックし、通信が正常に行われたかどうかを判断してください。

### 17.6.3 双方向通信機能（動作モード 0, 2）

動作モード 0, 2（非同期ノーマルモード，クロック同期モード）時は，1 対 1 接続のシリアル双方向通信ができます。同期方式は，動作モード 0（非同期ノーマルモード）の場合は非同期，動作モード 2（クロック同期モード）の場合はクロック同期となります。

#### ■ 双方向通信機能

UART を動作モード 0, 2（非同期ノーマルモード，クロック同期モード）で使用するためには，図 17.6-6 の設定が必要です。

図 17.6-6 UART の動作モード 0, 2（非同期ノーマルモード，クロック同期モード）の設定

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
SCRO, SCR1, SMRO, SMR1	PEN	P	SBL	CL	AD	REC	RXE	TXE	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE
動作モード 0	◎	◎	◎	◎	×	0	◎	◎	0	0	◎	◎	◎	0	◎	◎
動作モード 2	0	×	×	1	×	0	◎	◎	1	0	◎	◎	◎	0	◎	◎

SSRO, SSR1, SIDRO, SIDR1/SODRO, SODR1	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	送信データを設定（ライトした場合） /受信データを保持（リードした場合）							
動作モード 0	◎	◎	◎	◎	◎		◎	◎								
動作モード 2	×	◎	×	◎	◎		◎	◎								

DDRポート方向レジスタ

— : 未定義ビット

◎ : 使用ビット

×

1 : “1”を設定

0 : “0”を設定

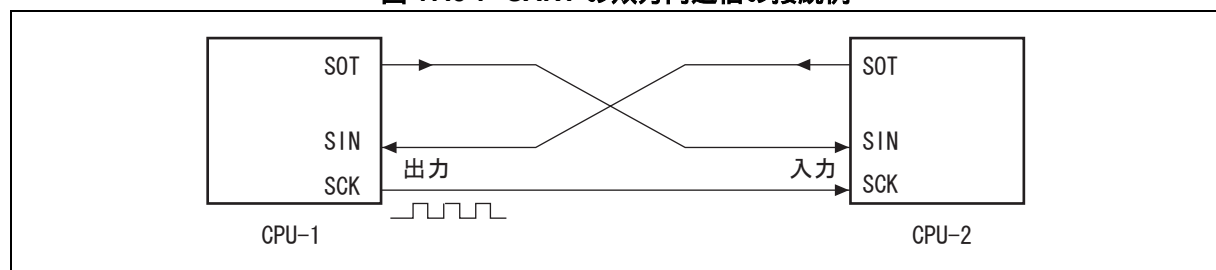
--	--	--	--	--	--	--	--

SINO, SIN1入力端子やSCK0, SCK1入力端子として使用する端子に対応するビットに“0”を設定

#### ● CPU 間の接続

2 つの CPU を接続します。

図 17.6-7 UART の双方向通信の接続例

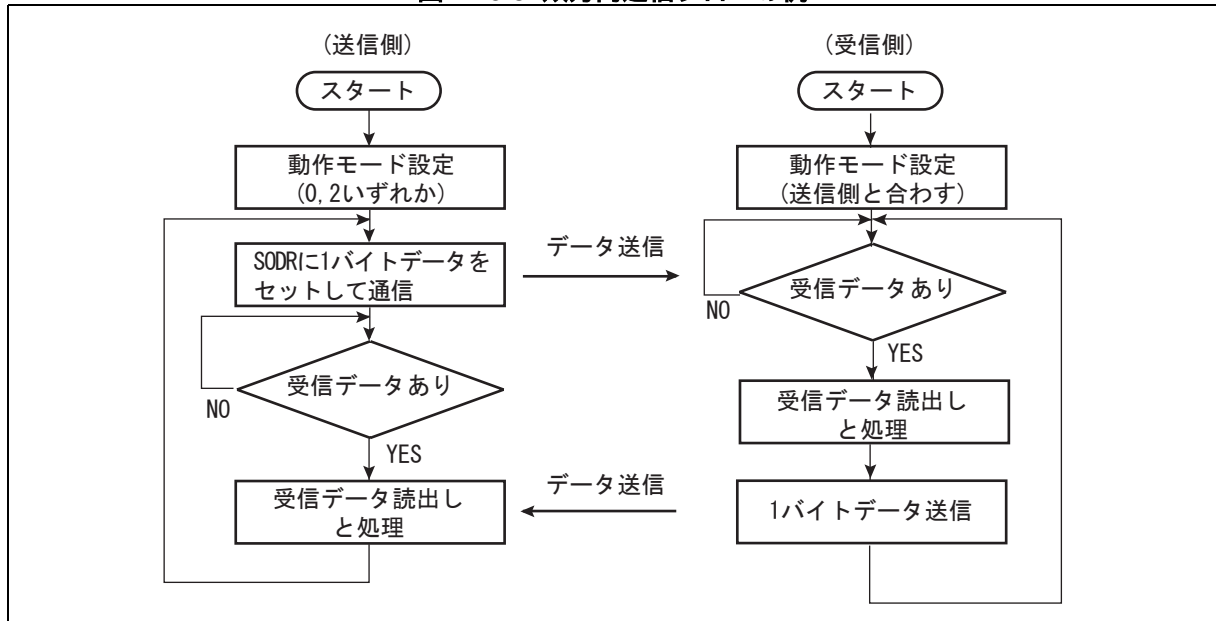


## ● 通信手順

通信は、送信データが準備できたときに任意のタイミングで送信側から開始します。送信側はシリアル出力データレジスタ (SODR0, SODR1) に送信データを設定し、シリアル制御レジスタの送信動作許可ビット (SCR0, SCR1 レジスタの TXE:bit8) に "1" を設定して、送信を開始してください。

正常に受信したことを送信側に示すために、受信データを送信側に転送する例を、図 17.6-8 に示します。

図 17.6-8 双方向通信フローの例



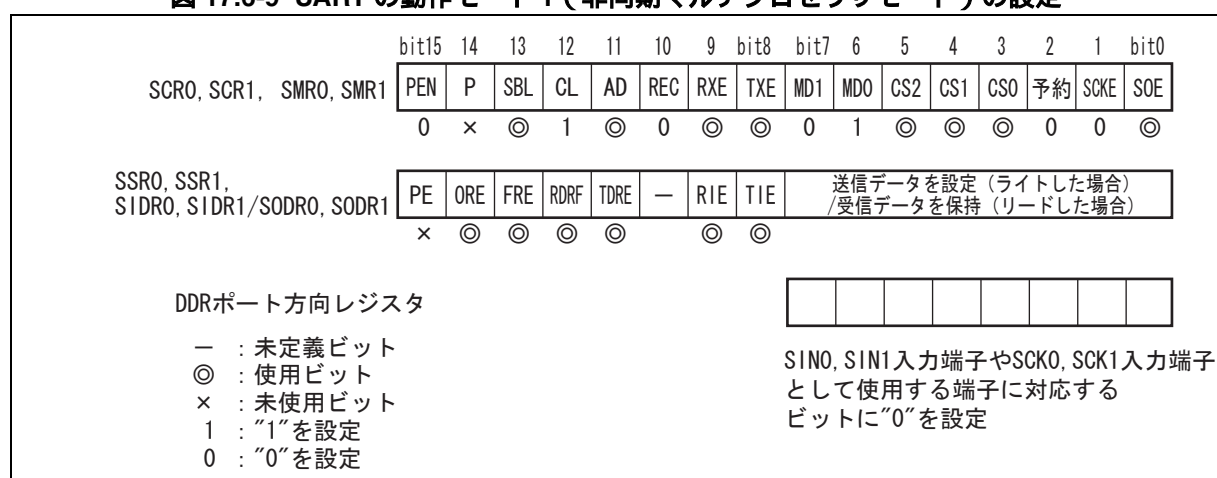
## 17.6.4 マスタ/スレーブ型通信機能（マルチプロセッサモード時）

動作モード 1（非同期マルチプロセッサモード）時は、複数 CPU のマスタ/スレーブ型接続による通信が可能です。ただし、マスタ側でのみ使用できます。

### ■ マスタ/スレーブ型通信機能

UART を動作モード 1（非同期マルチプロセッサモード）で使用するためには、図 17.6-9 の設定が必要です。

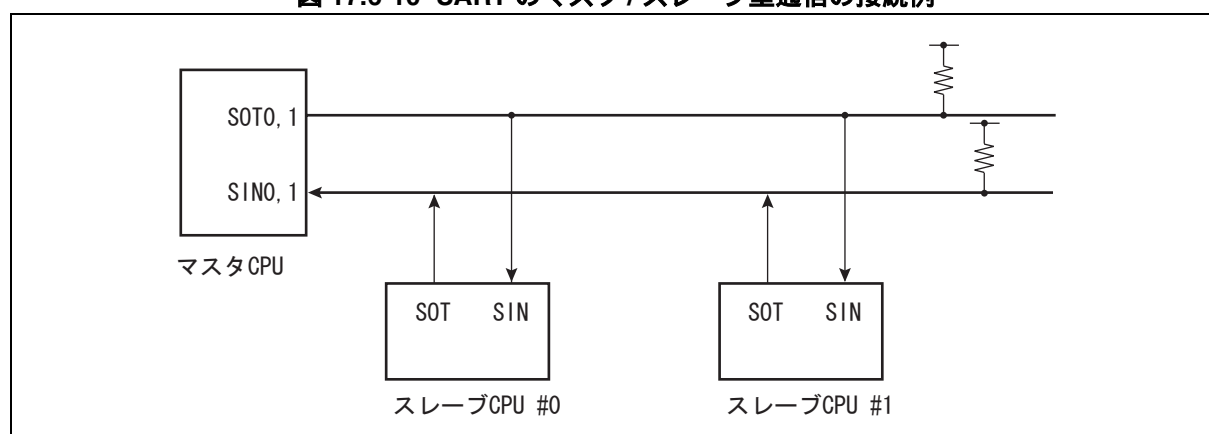
図 17.6-9 UART の動作モード 1（非同期マルチプロセッサモード）の設定



### ● CPU 間の接続

2本の共通通信ラインに1つのマスタCPUと、複数のスレーブCPUを接続して、通信システムを構成します。UARTは、マスタCPUとしてのみ使用できます。

図 17.6-10 UART のマスタ/スレーブ型通信の接続例



● 機能選択

マスタ / スレーブ型通信では、動作モードとデータ転送方式を選択してください。

動作モード 1 (非同期マルチプロセッサモード) 時は、パリティチェック機能は使用できないので、パリティ付加許可ビット (SCR0, SCR1 レジスタの PEN:bit15) には "0" を設定してください。

表 17.6-3 マスタ / スレーブ型通信機能の選択

	動作モード		データ	パリティ	同期方式	ストップビット
	マスタ CPU	スレーブ CPU				
アドレス 送受信	動作モード 1	-	A/D=1 + 8 ビットアドレス	なし	非同期	1 ビット または 2 ビット
データ 送受信			A/D=0 + 8 ビットデータ			

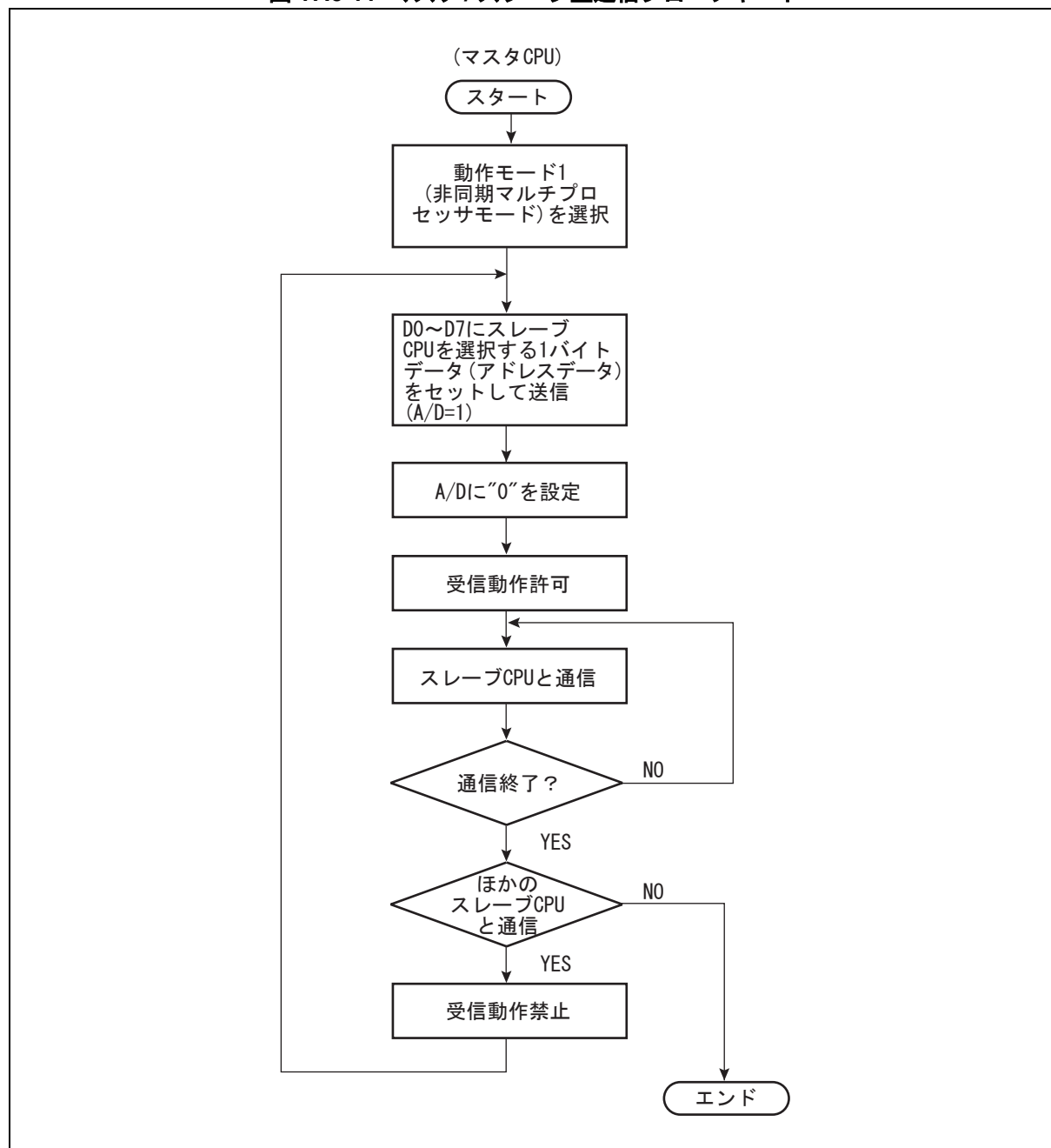
## ● 通信手順

通信は、マスタ CPU 側がアドレスデータを送信することによって開始します。

アドレスデータとは、A/D ビットが "1" のデータであり、アドレス / データ選択ビット (SCR0, SCR1 レジスタの A/D:bit11) の付加によって、通信先となるスレーブ CPU が選択されます。各スレーブ CPU はプログラムによりアドレスデータを判断し、割当てられたアドレスと一致する場合に、マスタ CPU との通信が開始します。

マスタ / スレーブ型通信のフローチャートを図 17.6-11 に示します。

図 17.6-11 マスタ / スレーブ型通信フローチャート



## 17.7 UART 使用上の注意

---

UART を使用する場合は、以下の点に注意してください。

---

### ■ UART 使用上の注意

#### ● 送受信の動作許可

UART には、送信と受信のそれぞれに対して送信動作許可ビット (SCR0, SCR1 レジスタの TXE:bit8) と受信動作許可ビット (SCR0, SCR1 レジスタの RXE:bit9) があります。

- リセット後の初期状態では、送受信とも動作禁止に設定 (SCR0, SCR1 レジスタの TXE: bit8=0, RXE:bit9=0) されているので、送受信を行う前に動作を許可する必要があります。
- 送受信動作を禁止 (SCR0, SCR1 レジスタの TXE:bit8=0, RXE:bit9=0) して、送受信を中止できます。

#### ● 動作モードの設定

動作モードは、送受信動作を禁止 (SCR0, SCR1 レジスタの TXE:bit8=0, RXE:bit9=0) してから設定してください。送受信中に動作モードを切り換えた場合、送受信されたデータは保証されません。

#### ● クロック同期モード

UART の動作モード 2 (クロック同期モード) は、クロック同期方式として設定されます。送受信データにスタートビットやストップビットは付加されません。

#### ● 送信割込み許可のタイミング

送信データ書込み許可フラグビット (SSR0, SSR1 レジスタの TDRE:bit11) はリセット値が "1" (送信データなし、送信データ書込み許可) にセットされているので、送信割込みを許可 (SSR0, SSR1 レジスタの TIE:bit8=1) したと同時に、送信割込み要求が発生します。必ず、送信データを準備してから、送信割込みを許可 (SSR0, SSR1 レジスタの TIE:bit8=1) してください。



## 17.8 UART のプログラム例

UART のプログラム例を示します。

### ■ UART のプログラム例

#### ● 処理

シリアル送受信を行うためにUARTの双方向通信機能(ノーマルモード)を使用します。

- 動作モード 0, 非同期モード(ノーマル), 8 ビットデータ長, 2 ビットストップビット長およびパリティなしを設定します。
- 通信には P60/SIN1 端子と P61/SOT1 端子を使用します。
- 専用ボーレートジェネレータを使用し, 約 9600bps にボーレートを設定します。
- 文字 13<sub>H</sub> を SOT1 端子から送信し, 割込みの発生により受信します。
- マシンクロック  $\phi$  は 16MHz と仮定します。

#### ● コーディング例

```

ICR13 EQU 0000BDH ;UART 送受信割込み制御レジスタ
DDR6 EQU 000016H ;ポート 6 データ方向レジスタ
CDCR1 EQU 00001BH ;通信プリスケアラレジスタ 1
SMR1 EQU 000024H ;モード制御レジスタ 1
SCR1 EQU 000025H ;制御レジスタ 1
SIDR1 EQU 000026H ;入力データレジスタ 1
SODR1 EQU 000026H ;出力データレジスタ 1
SSR1 EQU 000027H ;状態レジスタ 1
REC EQU SCR1:2 ;受信エラーフラグクリアビット
;----- メインプログラム -----
CODE CSEG ABS=0FFH
START:
; : ;スタックポインタ (SP) はすでに初期化されているものと
; 仮定します
AND CCR,#0BFH ;割込みを禁止します
MOV I:ICR13,#00H ;割込みレベル 0 (優先度が最も高い)
MOV I:DDR6,#0000000B;SIN1 端子を入力端子として設定します
MOV I:CDCR1,#080H ;通信プリスケアラを有効にします
MOV I:SMR1,#00010001B;動作モード 0 (非同期)
;専用ボーレートジェネレータ (9615bps) を使用します
;クロックパルス出力を無効にし, データ出力を有効にします
MOV I:SCR1,#00010011B;N パリティなし, 2 ストップビット
;8 データビットと受信エラーフラグをクリアします
;送受信動作を有効にします
MOV I:SSR1,#00000010B;送信割込みを禁止し, 受信割込みを許可にします
MOV I:SODR1,#13H ;送信データを書き込みます
MOV ILM,#07H ;PS の ILM をレベル 7 に設定します
OR CCR,#40H ;割込みを許可します
LOOP: MOV A,#00H ;無限ブール
MOV A,#01H
BRA LOOP

```

```

;----- 割り込みプログラム -----
WARI:
    MOV    A,SIDR1    ; 受信データを読み込みます
    CLRB   I:REC      ; 受信割り込み要求フラグをクリアします
;
;      ユーザ処理
;
    RETI              ; 割り込みから復帰します
CODE    ENDS

;----- ベクタ設定 -----
VECT    CSEG    ABS=0FFH
        ORG     0FF68H    ; 割り込み #37 (25H) のベクタを設定します
        DSL     WARI
        ORG     0FFDCH    ; リセットベクタを設定します
        DSL     START
        DB      00H      ; シングルチップモードを設定します
VECT    ENDS

```



# 第18章

---

## DTP/ 外部割込み

DTP/ 外部割込みの機能と動作について説明します。

18.1 DTP/ 外部割込みの概要

18.2 DTP/ 外部割込みのブロックダイアグラム

18.3 DTP/ 外部割込みの構成

18.4 DTP/ 外部割込みの動作説明

18.5 DTP/ 外部割込み使用上の注意

18.6 DTP/ 外部割込み回路のプログラム例

## 18.1 DTP/ 外部割込みの概要

DTP/ 外部割込みは、外部周辺装置が発生する割込み要求やデータ転送要求を CPU に伝達することで、外部割込み要求が発生したり、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を起動します。

### ■ DTP/ 外部割込み機能

外部周辺装置から外部割込み入力端子 (INT7 ~ INT0) に入力された割込み要求は、リソースの割込みと同様割込み要求が発生され、外部割込みが発生したりまたは拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を起動します。

割込み制御レジスタ (ICR) により拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を "無効" に設定している場合 (ISE=0) は、外部割込み機能が有効になり、割込み処理へ分岐します。

EI<sup>2</sup>OS を "有効" に設定している場合 (ISE=1) は、DTP 機能が有効となって、EI<sup>2</sup>OS による自動データ転送が行われ、指定回数のデータ転送終了後に割込み処理へ分岐します。

DTP/ 外部割込みの概要を表 18.1-1 に示します。

表 18.1-1 DTP/ 外部割込みの概要

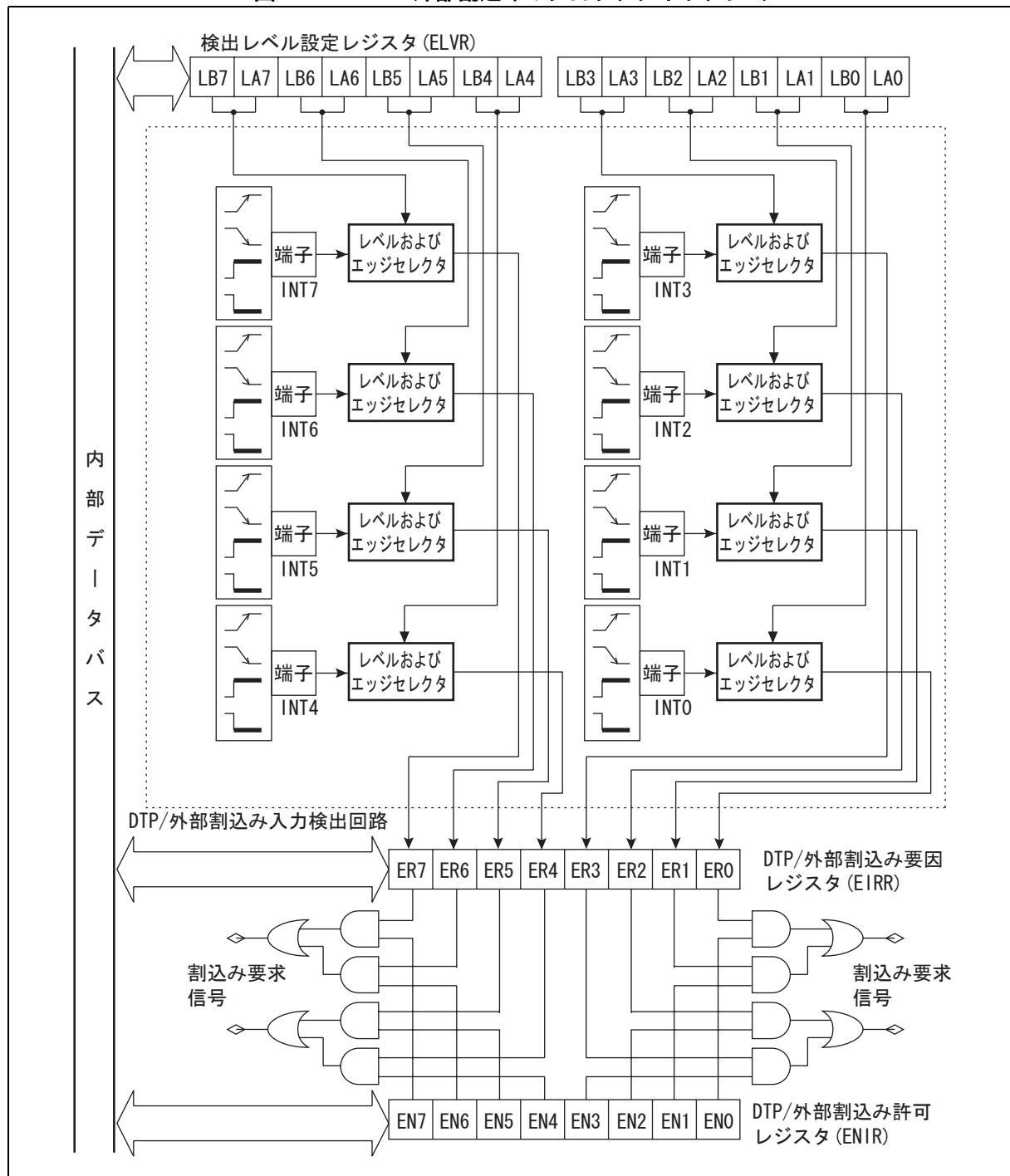
	外部割込み	DTP 機能
入力端子	8 本 (P10/INT0/DTT10 ~ P16/INT6, P63/INT7)	
割込み要因	検出レベル設定レジスタ (ELVR) にて、端子ごとにレベルまたはエッジを設定	
	"H" レベル / "L" レベル / 立上りエッジ / 立下りエッジの入力	"H" レベル / "L" レベルの入力
割込み番号	#20 (14 <sub>H</sub> ), #22 (16 <sub>H</sub> ), #25 (19 <sub>H</sub> ), #27 (1B <sub>H</sub> )	
割込み制御	DTP/ 外部割込み許可レジスタ (ENIR) による、割込み要求出力の許可または禁止	
割込みフラグ	DTP/ 外部割込み要因レジスタ (EIRR) による、割込み要因の保持	
処理選択	EI <sup>2</sup> OS を "無効" に設定 (ICR:ISE=0)	EI <sup>2</sup> OS を "有効" に設定 (ICR:ISE=1)
処理	外部割込み処理へ分岐	EI <sup>2</sup> OS によるデータ自動転送を指定回数処理後、割込み処理へ分岐

## 18.2 DTP/ 外部割込みのブロックダイアグラム

DTP/ 外部割込みのブロックダイアグラムを以下に示します。

### ■ DTP/ 外部割込みのブロックダイアグラム

図 18.2-1 DTP/ 外部割込みのブロックダイアグラム



● DTP/ 外部割込み入力検出回路

外部周辺装置が発生する割込み要求またはデータ転送要求を検出します。

検出レベル設定レジスタで設定したレベルまたはエッジを検出したい端子に対応する割込み要求フラグに "1" を設定してください (EIRR: ER)。

● 検出レベル設定レジスタ (ELVRH, ELVRL)

検出レベル設定レジスタには、DTP/ 外部割込みの要因発生条件となる、外部周辺装置からの入力信号のレベルまたはエッジを設定します。

● DTP/ 外部割込み要因レジスタ (EIRR)

DTP/ 外部割込みの要因を保持するレジスタです。

DTP/ 外部割込み端子に有効な信号が入力された場合、対応する DTP/ 外部割込み要求フラグビットに "1" が設定されます。

● DTP/ 外部割込み許可レジスタ (ENIR)

DTP/ 外部割込み許可レジスタ (ENIR) は、外部周辺装置の DTP/ 外部割込み要求の "許可" または "禁止" を設定します。

## ■ 端子の詳細と割込み番号

DTP/ 外部割込みで使用する端子と割込みの詳細を表 18.2-1 に示します。

表 18.2-1 DTP/ 外部割込みで使用する端子と割込み番号

チャネル	割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
		レジスタ名	アドレス	下位	中位	上位	
INT0/INT1	#20 (14 <sub>H</sub> )	ICR04	0000B4 <sub>H</sub>	FFFFAC <sub>H</sub>	FFFFAD <sub>H</sub>	FFFFAE <sub>H</sub>	
INT2/INT3	#22 (16 <sub>H</sub> )	ICR05	0000B5 <sub>H</sub>	FFFA4 <sub>H</sub>	FFFA5 <sub>H</sub>	FFFA6 <sub>H</sub>	
INT4/INT5	#25 (19 <sub>H</sub> )	ICR07	0000B7 <sub>H</sub>	FFFF98 <sub>H</sub>	FFFF99 <sub>H</sub>	FFFF9A <sub>H</sub>	
INT6/INT7	#27 (1B <sub>H</sub> )	ICR08	0000B8 <sub>H</sub>	FFFF90 <sub>H</sub>	FFFF91 <sub>H</sub>	FFFF92 <sub>H</sub>	

:EI<sup>2</sup>OS 割込みクリア信号で割込み要求フラグがクリアされると、使用できます。

18.3 DTP/ 外部割込みの構成

DTP/ 外部割込みの端子，割込み要因，レジスタの一覧および詳細を記述します。

■ DTP/ 外部割込みの端子

DTP/ 外部割込みで使用する端子は，汎用入出力ポートと兼用です。  
端子の機能と，DTP/ 外部割込みで使用する場合の設定を，表 18.3-1 に示します。

表 18.3-1 DTP/ 外部割込みの端子

端子名	端子機能	DTP/ 外部割込みの使用に必要な端子の設定
INT0	汎用入出力ポート / DTP 外部割込み入力	ポート方向レジスタ（DDR）により " 入力ポート " に設定
INT1		
INT2		
INT3		
INT4	汎用入出力ポート / DTP 外部割込み入力 / アドレスバス出力	
INT5		
INT6		
INT7		

■ 端子のブロックダイアグラム

< 参照 >      端子のブロックダイアグラムは，「第 9 章 I/O ポート」をご参照ください。

■ DTP/ 外部割込みのレジスタと初期値の一覧

図 18.3-1 DTP/ 外部割込みのレジスタと初期値の一覧

DTP/外部割込要因レジスタ (EIRR)	bit	15	14	13	12	11	10	9	8
		X	X	X	X	X	X	X	X
DTP/外部割込許可レジスタ (ENIR)	bit	7	6	5	4	3	2	1	0
		0	0	0	0	0	0	0	0
検出レベル設定レジスタ：上位 (ELVRH)	bit	15	14	13	12	11	10	9	8
		0	0	0	0	0	0	0	0
検出レベル設定レジスタ：下位 (ELVRL)	bit	7	6	5	4	3	2	1	0
		0	0	0	0	0	0	0	0
X : 不定									



### 18.3.1 DTP/ 外部割込み要因レジスタ (EIRR)

DTP/ 外部割込みの要因を保持するレジスタです。DTP/ 外部割込み端子に有効な信号が入力された場合，対応する割込み要求フラグビットに "1" が設定されます。

#### ■ DTP/ 外部割込み要因レジスタ (EIRR)

図 18.3-2 DTP/ 外部割込み要因レジスタ (EIRR)

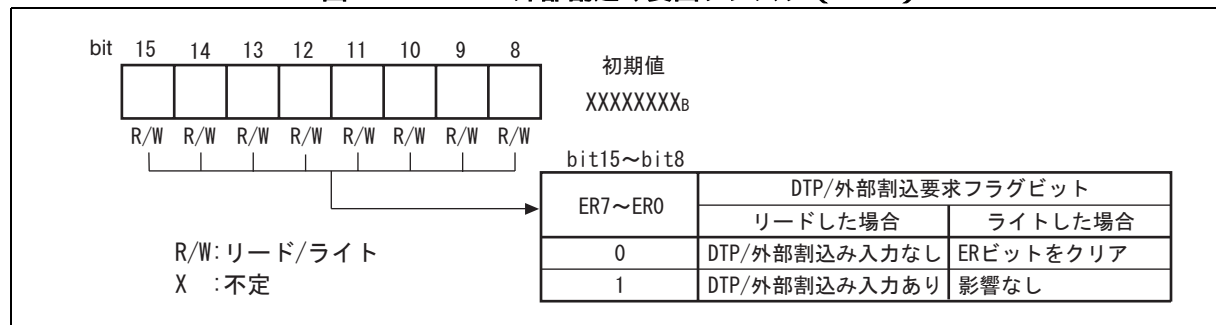


表 18.3-2 DTP/ 外部割込み要因レジスタ (EIRR) の機能

ビット名		機 能
bit15 ~ bit8	ER0 ~ ER7: DTP/ 外部割込 み要求フラグ ビット	<p>DTP/ 外部割込み端子に，検出レベル設定レジスタ (ELVRH, L) の検出条件選択ビット (LB<sub>x</sub>, LA<sub>x</sub>) で設定したエッジまたはレベル信号が入力された場合，"1" が設定されます。</p> <p><b>これらのビットに "1" を設定した場合：</b> DTP/ 外部割込み要求許可ビット (EN) に "1" を設定している場合は，対応する DTP/ 外部割込みチャンネルの割込み要求が発生します。</p> <p><b>これらのビットに "0" を設定した場合：</b>クリアされます。</p> <p><b>これらのビットに "1" を設定した場合：</b>影響しません。</p> <p>(注意事項) 複数の DTP/ 外部割込み要求が許可されている場合 (ENIR: EN=1) は，割込みを受付けたチャンネルのビットだけをクリアしてください (EIRR: ER<sub>x</sub>=0)。</p> <p>(参考) 拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) が起動された場合は，1 つのデータ転送が終了した時点で，対応する割込み要求フラグビットが自動的にクリアされます (EIRR: ER<sub>x</sub>=0)。</p>

- < 注意事項 >
- DTP/ 外部割込み要求フラグビット (EIRR:ER) の値は，対応する DTP/ 外部割込み要求許可ビット (ENIR:EN) が "1" に設定されているときのみ有効です。DTP/ 外部割込みが許可されていない状態 (ENIR:EN=0) では，DTP/ 外部割込み要因の有無にかかわらず DTP/ 外部割込み要因ビットがセットされる可能性があります。
  - DTP/ 外部割込みを許可 (ENIR:EN=1) する直前に，対応する DTP/ 外部割込み要求フラグビット (EIRR:ER) をクリアしてください。

18.3.2 DTP/ 外部割込み許可レジスタ (ENIR)

DTP/ 外部割込み許可レジスタ (ENIR) は , 外部割込み端子 (INT7 ~ INT0) ごとに外部割込み要求の " 許可 " または " 禁止 " を設定します。

■ DTP/ 外部割込み許可レジスタ (ENIR)

図 18.3-3 DTP/ 外部割込み許可レジスタ (ENIR)

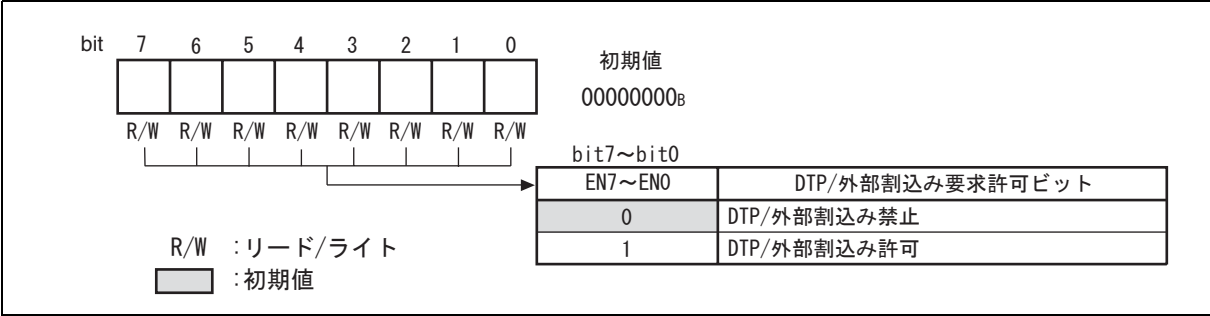


表 18.3-3 DTP/ 外部割込み許可レジスタ (ENIR) の機能

ビット名		機 能
bit7 ~ bit0	EN7 ~ EN0: DTP/ 外部割込 み要求許可 ビット	DTP/ 外部割込み端子に対して , DTP/ 外部割込み要求の " 許可 " または " 禁止 " を設定します。 DTP/ 外部割込み要求許可ビット (EN <sub>x</sub> ) に "1" を設定している場合に DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ER <sub>x</sub> ) に "1" が設定されると , 対応する DTP/ 外部割込み端子の割込み要求が発生します。 (参考) DTP/ 外部割込み要求許可ビットの設定にかかわらず , DTP/ 外部割込み端子の状態は ポートデータレジスタ (PDR) により , 直接読み出せます。

< 注意事項 > DTP/ 外部割込み許可 (ENIR:EN=1) する直前に , 対応する DTP/ 外部割込み要求フラグビット (EIRR:ER) をクリアしてください。

**表 18.3-4 DTP/ 外部割込み端子と割込み要求フラグビット，割込み許可ビットの対応**

DTP/ 外部割込み端子	DTP/ 外部割込み要求 フラグビット	DTP/ 外部割込み要求 許可ビット
INT0	ER0	EN0
INT1	ER1	EN1
INT2	ER2	EN2
INT3	ER3	EN3
INT4	ER4	EN4
INT5	ER5	EN5
INT6	ER6	EN6
INT7	ER7	EN7

18.3.3 検出レベル設定レジスタ，上位（ELVRH）

検出レベル設定レジスタ（上位）は，DTP/ 外部割込み端子のうちの INT7 ~ INT4 に対して，割込み要因となる入力信号レベルまたはエッジを設定します。

■ 検出レベル設定レジスタ，上位（ELVRH）

図 18.3-4 検出レベル設定レジスタ，上位（ELVRH）

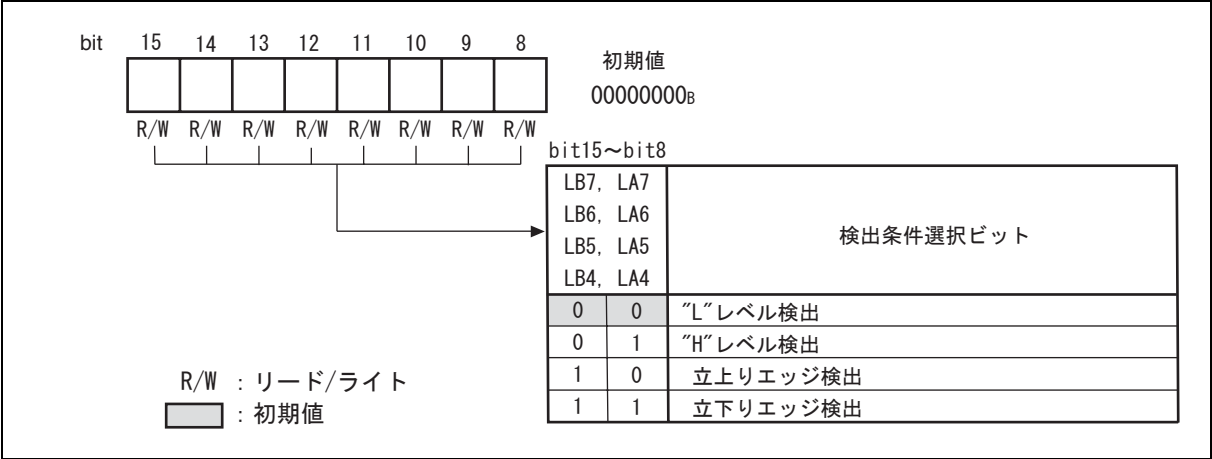


表 18.3-5 検出レベル設定レジスタ，上位（ELVRH）の機能

ビット名		機 能
bit15 ~ bit8	LB4, LA4 ~ LB7, LA7: 検出条件選択 ビット	DTP/ 外部割込み端子に対して，割込み要因の発生条件となる，外部周辺装置からの入力信号のレベルまたはエッジを設定します。 • 外部割込みの場合は 2 種類のレベルまたは 2 種類のエッジから，EI <sup>2</sup> OS の場合は 2 種類のレベルから選択してください。 (参考) 設定した検出信号が，DTP/ 外部割込み端子に入力されると，DTP/ 外部割込み要求を " 禁止 " に設定していても (ENIR:ENx=0)，DTP/ 外部割込み要求フラグビットには "1" が設定されます。

表 18.3-6 検出レベル設定レジスタ：上位と各チャネルの対応

DTP/ 外部割込み端子	ビット名
INT4	LB4, LA4
INT5	LB5, LA5
INT6	LB6, LA6
INT7	LB7, LA7

18.3.4 検出レベル設定レジスタ，下位（ELVRL）

検出レベル設定レジスタ（下位）は，DTP/ 外部割込み端子のうちの INT3 ~ INT0 に対して，割込み要因となる入力信号レベルまたはエッジを設定します。

■ 検出レベル設定レジスタ，下位（ELVRL）

図 18.3-5 検出レベル設定レジスタ，下位（ELVRL）

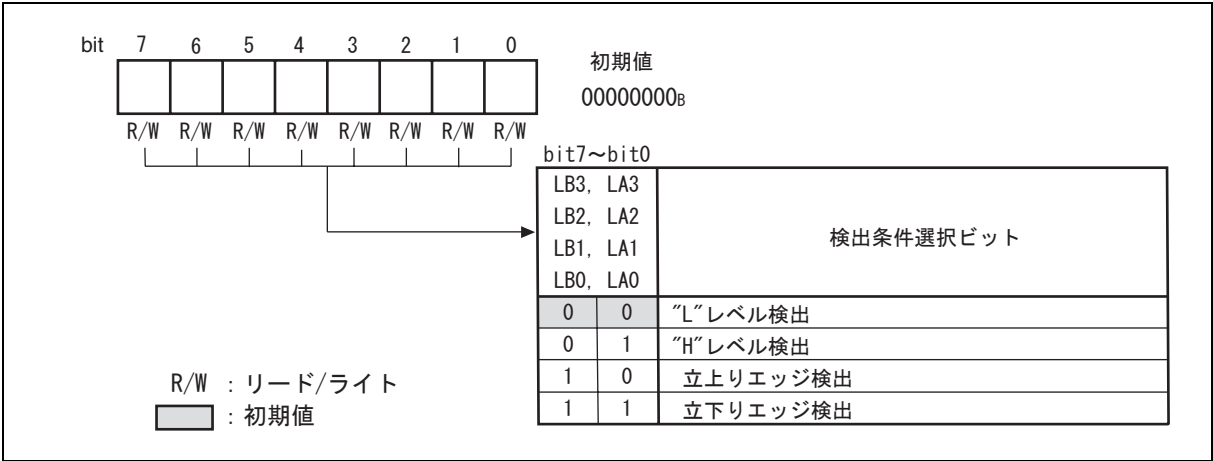


表 18.3-7 検出レベル設定レジスタ，下位（ELVRL）の機能

ビット名		機 能
bit7 ~ bit0	LB0, LA0 ~ LB3, LA3: 検出条件選択 ビット	DTP/ 外部割込み端子に対して，割込み要因の発生条件となる，外部周辺装置からの入力信号のレベルまたはエッジを設定します。 • 外部割込みの場合は 2 種類のレベルまたは 2 種類のエッジから，EI <sup>2</sup> OS の場合は 2 種類のレベルから選択してください。 (参考) 設定した検出信号が，DTP/ 外部割込み端子に入力されると，DTP/ 外部割込み要求を " 禁止 " に設定していても (ENIR:EN <sub>x</sub> =0)，DTP/ 外部割込み要求フラグビットには "1" が設定されます。

表 18.3-8 検出レベル設定レジスタ：下位と各チャネルの対応

DTP/ 外部割込み端子	ビット名
INT0	LB0, LA0
INT1	LB1, LA1
INT2	LB2, LA2
INT3	LB3, LA3

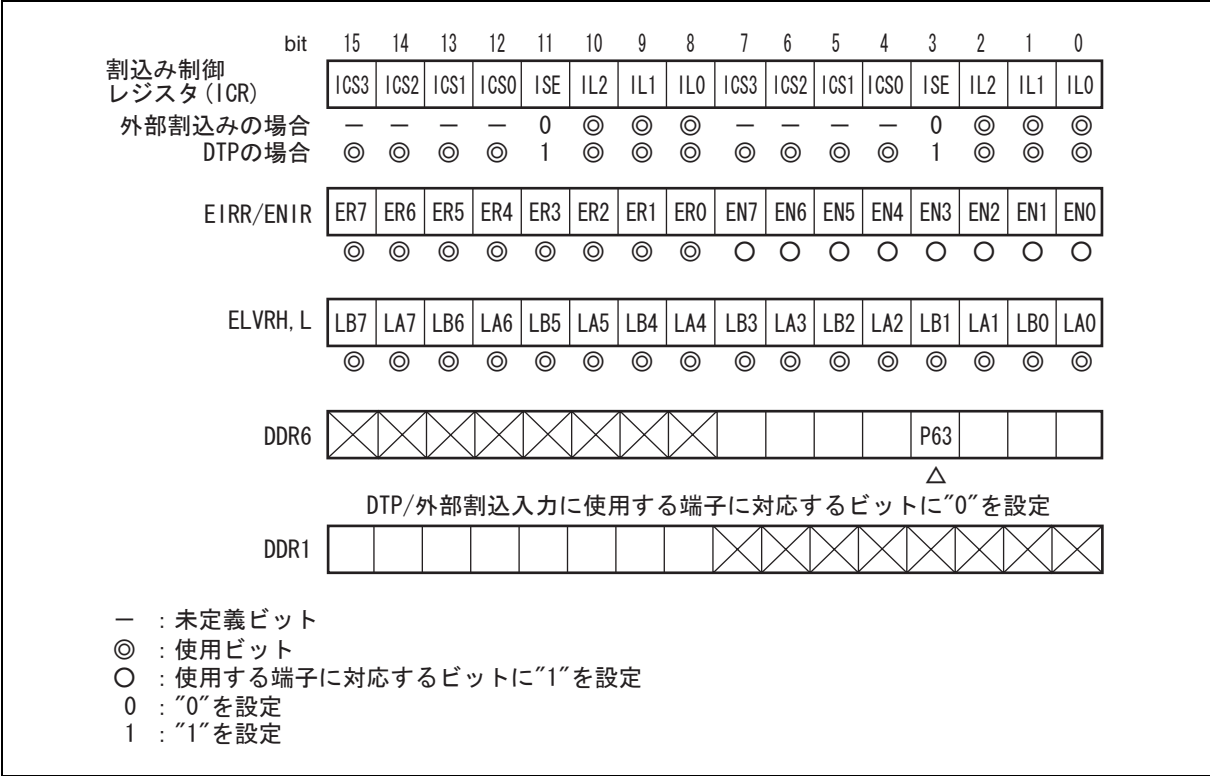
18.4 DTP/ 外部割込みの動作説明

DTP/ 外部割込みには、外部割込み機能と DTP 機能があります。各機能の設定と動作について説明します。

■ DTP/ 外部割込みの設定

DTP/ 外部割込みを使用するには、図 18.4-1 の設定が必要です。

図 18.4-1 DTP/ 外部割込みの設定



● 設定の手順

- DTP/ 外部割込みを使用する場合は、以下の手順で各レジスタを設定してください。
- 外部割込み入力として使用する端子と兼用する汎用入出力ポートを入力ポートに設定する。
  - DTP/ 外部割込み許可レジスタ (ENIR) のビットの内、使用する DTP/ 外部割込みチャンネルに対応する割込み要求許可ビット (EN<sub>x</sub>) に "0" を設定します。
  - 検出レベル設定レジスタ (ELVRH, L) のビットの内、使用する DTP/ 外部端子に対応する検出条件選択ビット (LA<sub>x</sub>, LB<sub>x</sub>) により、検出するエッジまたはレベルを設定します。
  - DTP/ 外部割込み要因レジスタ (EIRR) のビットの内、使用する DTP/ 外部割込みチャンネルに対応する割込み要求フラグビット (ER<sub>x</sub>) に "0" を設定します。
  - DTP/ 外部割込み許可レジスタ (ENIR) のビットの内、使用する DTP/ 外部割込みチャンネルに対応する割込み要求許可ビット (EN<sub>x</sub>) に "1" を設定します。

- DTP/ 外部割込みのレジスタは、外部割込み要求を " 禁止 " ( ENIR:ENx=0 ) に設定してから行ってください。
- DTP/ 外部割込み要求を " 許可 " ( ENIR:ENx=1 ) する場合は、先に DTP/ 外部割込み要因レジスタ ( EIRR ) のビットの内、対応する DTP/ 外部割込み要求フラグビットをクリア ( ERx=0 ) する必要があります。これは、レジスタ設定時に誤って割込み要求が発生するのを防ぐためです。

#### ● 外部割込み機能と DTP 機能の選択

外部割込み機能と DTP 機能のどちらが実行されるかは、対応する割込み制御レジスタ ( ICR ) の EI<sup>2</sup>OS 許可ビット ( ISE:bit11 ) の設定によって決定されます。

ISE ビットに "1" を設定した場合は、拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) が許可され、DTP 機能が実行されます。

ISE ビットに "0" を設定した場合は、EI<sup>2</sup>OS は禁止され、外部割込み機能が実行されます。

- 
- < 注意事項 >
- 1 つの割込み制御レジスタに割当てられている割込み要求は、すべて割込みレベル ( IL2 ~ IL0 ) が同じになります。
  - 1 つの割込み制御レジスタに複数の割込み要求が割当てられている場合、そのうちの 1 つが EI<sup>2</sup>OS を使用すると、ほかの割込み要求は使用できません。
- 

#### ■ DTP/ 外部割込み動作

DTP/ 外部割込みの制御ビットと割込み要因を、表 18.4-1 に示します。

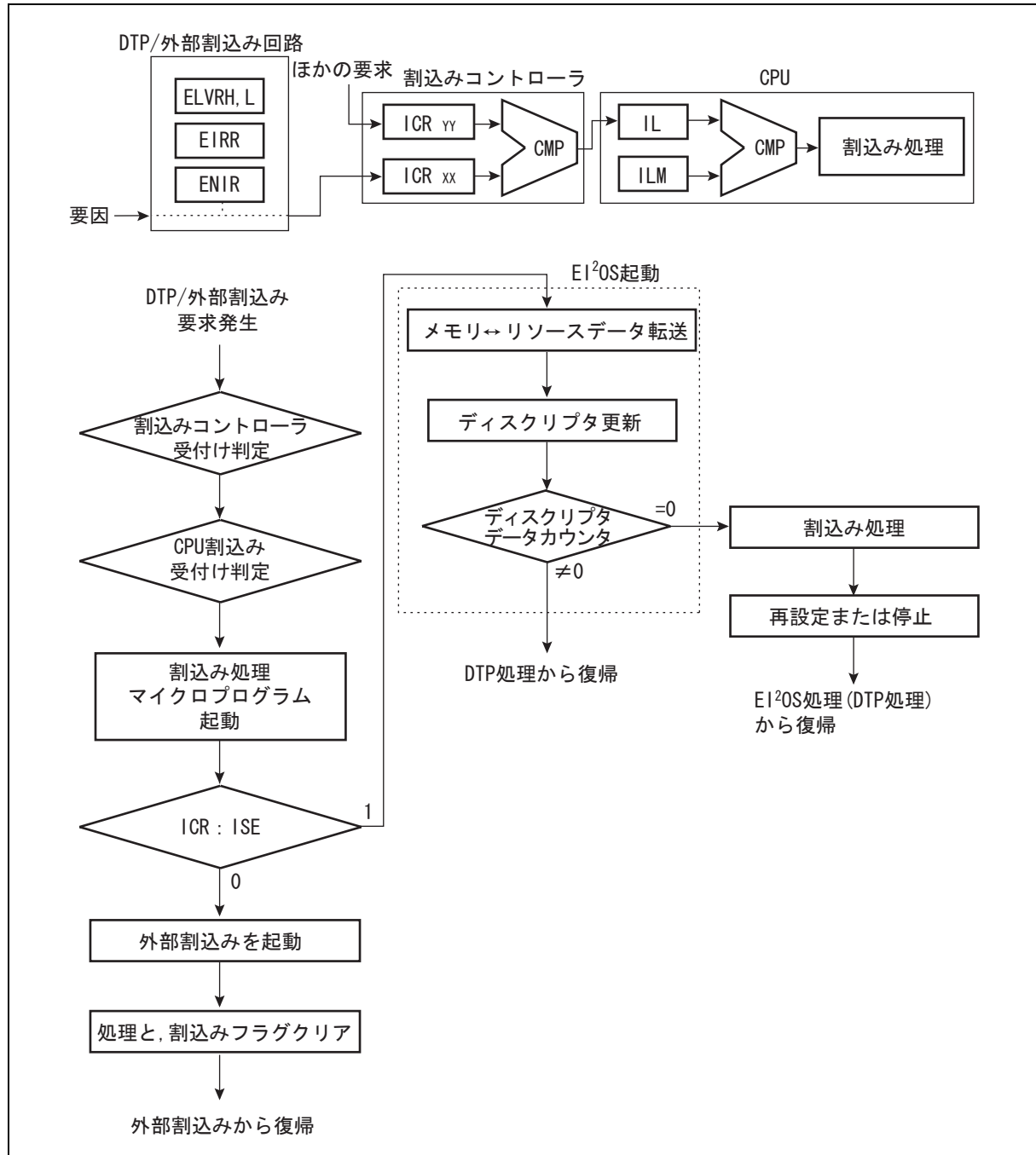
表 18.4-1 DTP/ 外部割込みの制御ビットと割込み要因

	DTP/ 外部割込み
割込み要求フラグビット	EIRR:ER7 ~ ER0
割込み要求許可ビット	ENIR:EN7 ~ EN0
割込み要因	INT7 ~ INT0 端子への、有効エッジ / レベルの入力

DTP/ 外部割込みの割込み要求が割込みコントローラに対して出力された場合、割込み制御レジスタ ( ICR ) の EI<sup>2</sup>OS 許可ビット ( ISE:bit11 ) に "0" を設定している場合は、割込み処理が実行されます。"1" を設定している場合は、拡張インテリジェント I/O サービス ( EI<sup>2</sup>OS ) が実行されます。

DTP/ 外部割込みの動作を図 18.4-2 に示します。

図 18.4-2 DTP/ 外部割込みの動作





## 18.4.1 外部割込み機能

---

DTP/ 外部割込み機能は、DTP/ 外部割込み端子における信号（エッジまたはレベル）検出により、割込み要求が発生します。

---

### ■ 外部割込み機能

DTP/ 外部割込み端子において、検出レベル設定レジスタ (ELVRH, L) で設定した信号（エッジまたはレベル）が検出されると、DTP/ 外部割込み要因レジスタ (EIRR) の割込み要求フラグビット (ERx) に "1" がセットされます。

割込み要求フラグビット (ERx) に "1" がセットされた場合に、DTP/ 外部割込み許可レジスタ (ENIR) の割込み要求許可ビットを "許可" (ENx=1) に設定していると、割込みコントローラに対して割込み要求が発生します。

割込みコントローラが、ほかの割込み要求よりも割込み優先順位が高いと判定した場合は、割込み処理を開始します。

CPU は、プロセッサステータス (PS) 内のインタラプトレベルマスクビット (ILM) と割込み要求のレベル (ICR:IL) を比較し、割込み要求レベルが ILM より高く、かつ割込み許可ビットが "許可" に設定されていた場合は (PS:CCR:I=1)、現在実行中の命令の終了後に割込み処理を実行し、割込み処理へ分岐します。

割込み処理ルーチン内で、DTP/ 外部割込み要因レジスタ (EIRR) のビットの内、対応する DTP/ 外部割込み要求フラグビット (ERx) に "0" を設定し、DTP/ 外部割込み要求をクリアしてください。

- 
- < 注意事項 >
- DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ERx) は、DTP/ 外部割込み許可レジスタ (ENIR) の対応する DTP/ 外部割込み要求許可ビット (ENx) の設定にかかわらず、DTP/ 外部割込み起動要因が発生すると "1" が設定されます。
  - 割込み処理が起動された場合は、起動要因となった DTP/ 外部割込み要求フラグビット (ERx) をクリアしてください。DTP/ 外部割込み要求フラグビット (ERx) が "1" のままでは割込みから復帰できません。クリアする場合は、受け付けられている DTP/ 外部割込み要因以外のフラグビットをクリアしないように注意してください。
-

## 18.4.2 DTP 機能

DTP/ 外部割込みは、外部周辺装置からの信号を DTP/ 外部割込み端子において検出し、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を起動します。

### ■ DTP 機能

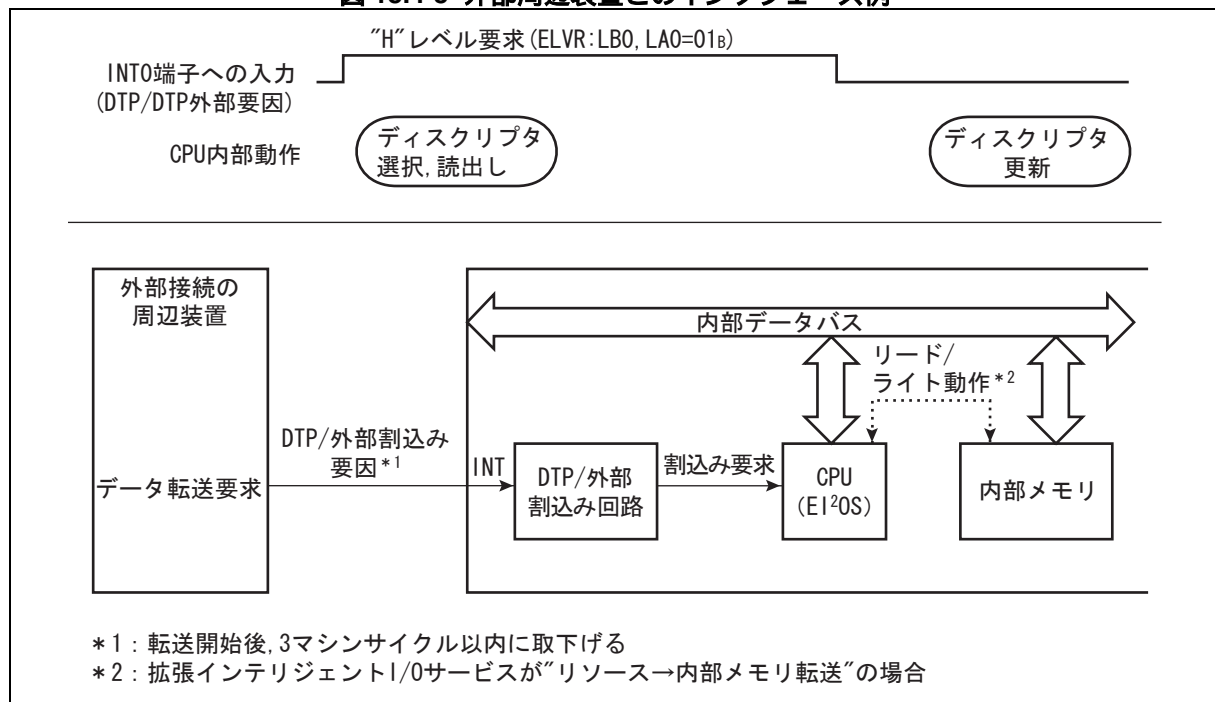
DTP/ 外部割込み機能の検出レベル設定レジスタ (ELVRH, L) で設定した信号レベルを検出することにより、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) を起動します。

EI<sup>2</sup>OS の動作が "許可" (ICR レジスタの ISE:bit11=1) されていれば、割込み要求が受け付けられた時点で EI<sup>2</sup>OS を起動し、データ転送を開始します。

1 つのデータ転送が終了するとディスクリプタの更新などが行われ、DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ER<sub>x</sub>) がクリアされて、DTP/ 外部割込み端子からの次の要求に備えます。

EI<sup>2</sup>OS による転送がすべて終了すると、割込み処理へ分岐します。

図 18.4-3 外部周辺装置とのインタフェース例



## 18.5 DTP/ 外部割込み使用上の注意

DTP/ 外部割込みを使用する際は、以下の点にご注意ください。

### ■ DTP/ 外部割込み使用上の注意

#### ● DTP 機能を使用する場合の外部接続する周辺装置の条件

DTP 機能を使用する場合は、転送により自動的にデータ転送要求をクリアしてください。

転送動作を開始してから 3 マシンサイクル以内に転送要求信号を無効にしてください。有効にしたままの場合、DTP/ 外部割込みは、次の転送要求が発生したものとして、再度データ転送動作を行います。

#### ● 外部割込みの入力極性

検出レベル設定レジスタでエッジ検出を設定している場合は、エッジを検出するためのパルス幅は最小 3 マシンサイクルが必要です。

要求入力レベルがレベル設定の場合、パルス幅はデータシートに記載されている最小パルス幅以上の期間を必要とします。また割込み入力端子がアクティブレベルを保持しつづけている限りは、DTP/ 外部割込み要因レジスタをクリアしても、割込みコントローラへの割込み要求は発生し続けます。

検出レベル設定レジスタ (ELVRH, L) でレベル検出を設定している場合は、割込み要因となるレベルが入力されると、DTP/ 外部割込み要因レジスタ (EIRR) の割込み要求フラグビット (EIRR:ER<sub>x</sub>) に "1" が設定され、図 18.5-1 に示すように、要因が保持されます。

割込み要求フラグビット (EIRR:ER<sub>x</sub>) に要因が保持された場合は、DTP/ 外部割込み要因が取下げられても、割込み要求許可状態 (ENIR:EN<sub>x</sub>=1) であれば、割込みコントローラへの要求はアクティブのままです。割込みコントローラへの要求を取下げるには DTP/ 外部割込み要因レジスタ (EIRR) の割込み要求フラグビット (EIRR:ER<sub>x</sub>) をクリアする必要があります。

図 18.5-1 レベル検出が設定されている場合の割込み要求フラグビット (EIRR:ER<sub>x</sub>) のクリア

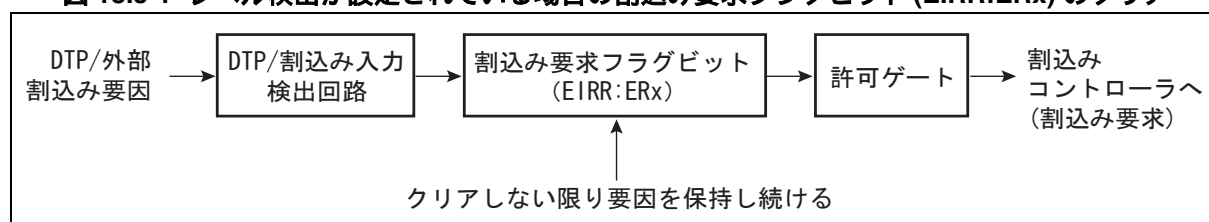
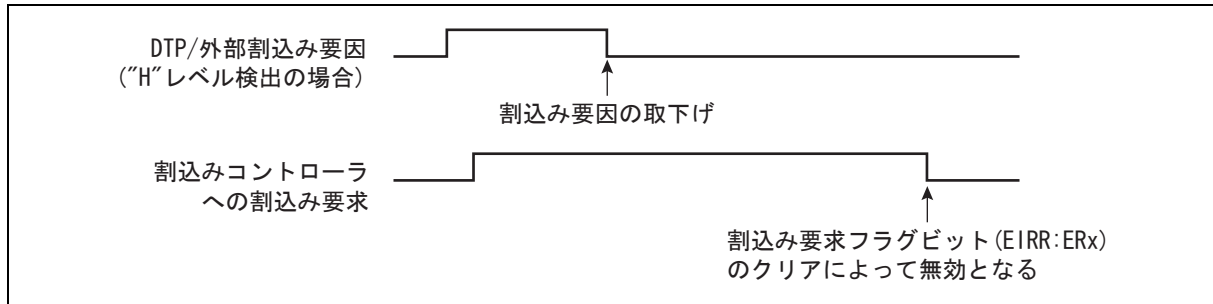


図 18.5-2 割込み要求許可の場合の DTP/ 外部割込み要因と割込み要求



● 割込みに関する注意

外部割込み機能を使用している場合は、DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ERx) に "1" が設定され、DTP/ 外部割込み要求を "許可" に設定した状態 (ENIR:ENx=1) では、割込み処理から復帰できません。割込み処理ルーチン内で必ず DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ERx) に "0" を設定してください。

検出レベル設定レジスタ (ELVRH, L) でレベル検出を設定している場合は、割込み要因となるレベルが入力されたままであれば、DTP/ 外部割込み要因レジスタ (EIRR) の DTP/ 外部割込み要求フラグビット (ERx) をクリアしても、すぐに再セットされます。必要に応じて DTP/ 外部割込み要求を禁止するか (ENIR:ENx=0)、割込み要因を解消してください。

## 18.6 DTP/ 外部割込み回路のプログラム例

DTP/ 外部割込み機能のプログラム例を示します。

### ■ DTP/ 外部割込み機能のプログラム例

#### ● 処理仕様

INT0 端子に入力されるパルスの立上りエッジを検出して外部割込みを発生します。

#### ● コーディング例

```

ICR04 EQU 0000B4H ;DTP/ 外部割込み制御レジスタ
DDR6 EQU 000016H ; ポート 6 方向レジスタ
DDR1 EQU 000011H ; ポート 1 方向レジスタ
ENIR EQU 000030H ;DTP/ 外部割込み許可レジスタ
EIRR EQU 000031H ;DTP/ 外部割込み要因レジスタ
ELVRL EQU 000032H ; 検出レベル設定レジスタ:L
ELVRH EQU 000033H ; 検出レベル設定レジスタ:H
ERO EQU EIRR:0 ;INT0 割込み要求フラグビット
ENO EQU ENIR:0 ;INT0 割込み要求許可ビット
;
;----- メインプログラム -----
CODE CSEG
START: ; スタックポインタ (SP) などは初期化済みとする
MOV I:DDR1,#00000000B;DDR1 を " 入力ポート " に設定
AND CCR,#0BFH ; 割込み禁止
MOV I:ICR04,#00H ; 割込みレベル 0 (最強), EI2OS 禁止
CLRB I:ENO ;ENIR で INT0 を禁止
MOV I:ELVRL,#00000010B;INT0 は, 立上がりエッジ選択
CLRB I:ERO ;EIRR で INT0 の割込み要求フラグクリア
SETB I:ENO ;ENIR で INT0 の割込み要求許可
MOV ILM,#07H ;PS 内の ILM をレベル 7 に設定
OR CCR,#40H ; 割込み許可
LOOP: MOV A,#00H ; 無限ループ
MOV A,#01H ;
BRA LOOP ;
;----- 割込みプログラム -----
WARI:
CLRB I:ERO ; 割込み要求フラグをクリア
.
; ユーザ処理
.
RETI ; 割込み処理からの復帰
CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
ORG 00FFACH ; 割込み番号 #20 (14H) にベクタを設定
DSL WARI
ORG 00FFDCH ; リセットベクタ設定
DSL START
DB 00H ; シングルチップモードに設定
VECT ENDS
END START

```

## ■ DTP 機能のプログラム例

### ● 処理仕様

INT0 端子に入力される信号の "H" レベルを検出して、拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) のチャンネル 0 を起動します。

DTP 処理 (EI<sup>2</sup>OS) で RAM 上のデータをポート 0 へ出力します。

### ● コーディング例

```

ICR04 EQU 0000B4H ;DTP/ 外部割込み制御レジスタ
DDR0 EQU 000010H ; ポート 0 方向レジスタ
DDR1 EQU 000011H ; ポート 1 方向レジスタ
ENIR EQU 000030H ;DTP/ 外部割込み許可レジスタ
EIRR EQU 000031H ;DTP/ 外部割込み要因レジスタ
ELVRL EQU 000032H ; 検出レベル設定レジスタ:L
ELVRH EQU 000033H ; 検出レベル設定レジスタ:H
ER0 EQU EIRR:0 ;INT0 割込み要求フラグビット
EN0 EQU ENIR:0 ;INT0 割込み要求許可ビット
;
BAPL EQU 000100H ; バッファアドレスポインタ下位
BAPM EQU 000101H ; バッファアドレスポインタ中位
BAPH EQU 000102H ; バッファアドレスポインタ上位
ISCS EQU 000103H ;EI2OS ステータスレジスタ
IOAL EQU 000104H ;I/O アドレスレジスタ下位
IOAH EQU 000105H ;I/O アドレスレジスタ上位
DCTL EQU 000106H ; データカウンタ下位
DCTH EQU 000107H ; データカウンタ上位
;
;----- メインプログラム -----
CODE CSEG
START: ; スタックポインタ (SP) などは初期化済みとする
MOV I:DDR0,#11111111B;DDR0 で " 出力ポート " に設定
MOV I:DDR1,#00000000B;DDR1 で " 入力ポート " に設定
AND CCR,#0BFH ; 割込み禁止
MOV I:ICR04,#08H ; 割込みレベル 0 (最強) EI2OS 許可 チャンネル 0
MOV BAPL,#00H ; 出力データの格納先アドレスの設定
MOV BAPM,#06H ; (600H ~ 60AH を使用)
MOV BAPH,#00H
MOV ISCS,#12H ; バイト転送, バッファアドレス + 1,
;I/O アドレス固定, メモリ I/O へ転送
MOV IOAL,#00H ; 転送先アドレスポインタとして
MOV IOAH,#00H ; ポート 0 (PDR0) を設定
MOV DCTL,#0AH ; 転送回数を 10 回に設定
MOV DCTH,#00H
;
CLRB I:EN0 ;ENIR で INT0 を禁止
MOV I:ELVRL,#00000001B;INT0 は, "H" レベル検出を設定
CLRB I:ER0 ;EIRR で INT0 の割込み要求フラグクリア
SETB I:EN0 ;ENIR で INT0 の割込み要求許可
MOV ILM,#07H ;PS 内の ILM をレベル 7 に設定
OR CCR,#40H ; 割込み許可
LOOP: MOV A,#00H ; 無限ループ
MOV A,#01H ;
BRA LOOP ;

```

```

;----- 割込みプログラム -----
WARI:
    CLRB   I:ERO           ;INT0 割込み要求フラグをクリア
    ;
    .
    ユーザ処理             ;EI2OS の終了などの処理を再指定。処理を終了するために
    ;
    .
    RETI                  ; 割込み処理からの復帰
CODE   ENDS
;----- ベクタ設定 -----
VECT   CSEG   ABS=0FFH
        ORG    00FFACH      ; 割込み番号 #20 (14H) にベクタを設定
        DSL    WARI
        ORG    00FFDCH      ; リセットベクタ設定
        DSL    START
        DB     00H          ; シングルチップモードに設定
VECT   ENDS
        END     START

```

# 第19章

## 遅延割込み発生モジュール

遅延割込み発生モジュールの機能と動作について説明します。

- 19.1 遅延割込み発生モジュールの概要
- 19.2 遅延割込み発生モジュールのブロックダイヤグラム
- 19.3 遅延割込み発生モジュールの構成
- 19.4 遅延割込み発生モジュールの動作説明
- 19.5 遅延割込み発生モジュール使用上の注意
- 19.6 遅延割込み発生モジュールのプログラム例



## 19.1 遅延割込み発生モジュールの概要

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。

ソフトウェアによってハードウェア割込み要求を発生させることができます。

### ■ 遅延割込み発生モジュールの概要

遅延割込み発生モジュールを使用することにより、ソフトウェアによってハードウェア割込み要求を発生または解除できます。

表 19.1-1 に、遅延割込み発生モジュールの概要を示します。

表 19.1-1 遅延割込み発生モジュールの概要

	機能と制御
割込み要因	遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に "1" を設定することにより割込み要求を発生 遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に "0" を設定することにより割込み要求を解除
割込み番号	#42 (2A <sub>H</sub> )
割込み制御	レジスタによる許可設定はありません。
割込みフラグ	遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に保持されています。
EI <sup>2</sup> OS	拡張インテリジェント I/O サービスには対応していません。

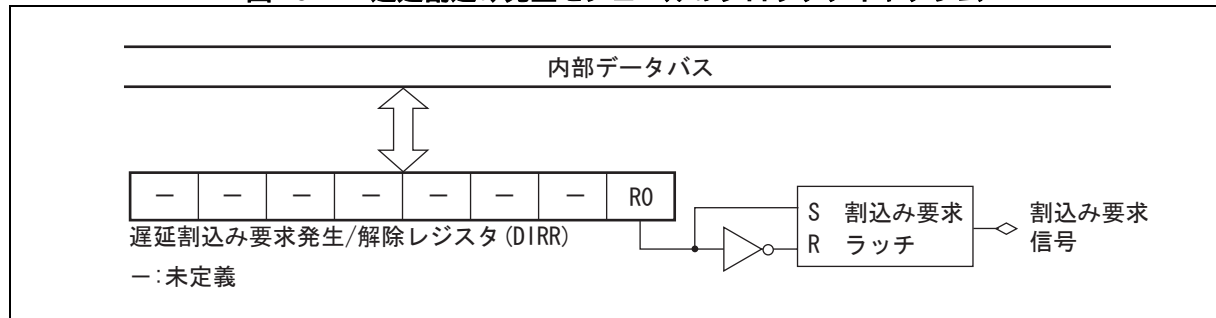
## 19.2 遅延割込み発生モジュールのブロックダイアグラム

遅延割込み発生モジュールは、以下のブロックで構成されています。

- 割込み要求ラッチ
- 遅延割込み要求発生 / 解除レジスタ (DIRR)

### ■ 遅延割込み発生モジュールのブロックダイアグラム

図 19.2-1 遅延割込み発生モジュールのブロックダイアグラム



#### ● 割込み要求ラッチ

遅延割込み要求発生 / 解除レジスタ (DIRR) の設定内容 (遅延割込み要求の発生または解除) を保持するラッチです。

#### ● 遅延割込み要求発生 / 解除レジスタ (DIRR)

遅延割込み要求を発生または解除します。

### ■ 割込み番号

遅延割込み発生モジュールで使用する割込み番号を以下に示します。

- 割込み番号 #42 ( $2A_H$ )

## 19.3 遅延割込み発生モジュールの構成

遅延割込み発生モジュールのレジスタ一覧および詳細を記述します。

### ■ レジスタと初期値の一覧

図 19.3-1 遅延割込み発生モジュールのレジスタと初期値の一覧

	bit	15	14	13	12	11	10	9	8
遅延割込み要求発生/解除レジスタ (DIRR)		X	X	X	X	X	X	X	0
X : 不定									

19.3.1 遅延割込み要求発生 / 解除レジスタ (DIRR)

遅延割込み要求発生 / 解除レジスタ (DIRR) は、遅延割込み要求を発生または解除します。

■ 遅延割込み要求発生 / 解除レジスタ (DIRR)

図 19.3-2 遅延割込み要求発生 / 解除レジスタ (DIRR)

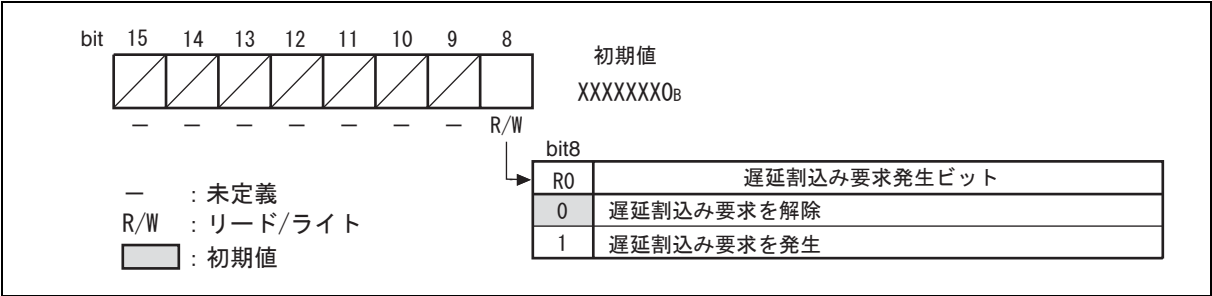


表 19.3-1 遅延割込み要求発生 / 解除レジスタ (DIRR) の機能

ビット名		機 能
bit15 ~ bit9	未定義ビット	リードした場合：値は不定です。 ライトした場合：影響しません。
bit8	R0: 遅延割込み要求発生ビット	遅延割込み要求を発生または解除します。 このビットに "0" を設定した場合：遅延割込み要求を解除します。 このビットに "1" を設定した場合：遅延割込み要求を発生します。

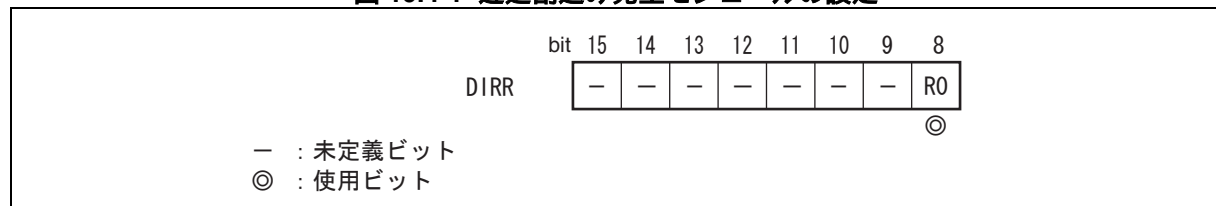
## 19.4 遅延割込み発生モジュールの動作説明

遅延割込み発生モジュールには、ソフトウェアによって割込み要求を発生または解除する機能があります。

### ■ 遅延割込み発生モジュールの動作説明

遅延割込み発生モジュールを使用するには、図 19.4-1 の設定が必要です。

図 19.4-1 遅延割込み発生モジュールの設定



遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に "1" を設定した場合、割込み要求を発生します。割込み要求の許可ビットはありません。

#### ● 遅延割込み発生モジュールの動作

遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に "1" を設定した場合、割込み要求ラッチが "1" に設定され、割込みコントローラに対して割込み要求を発生します。

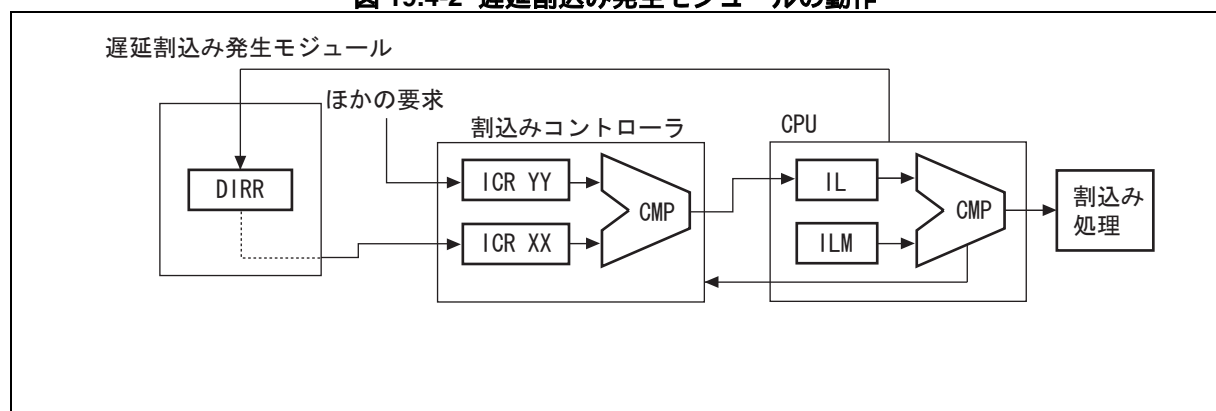
要求された割込みに対して割込みコントローラが、ほかの割込み要求よりも割込み優先順位が高いと判定した場合は、CPU に対して割込み要求が発生します。

CPU 側は、プロセッサステータス (PS) 内のインタラプトレベルマスクビット (ILM) と割込み要求のレベル (ICR: IL) を比較し、割込み要求レベルが ILM より高かった場合は、現在実行中の命令の終了後に、遅延割込み処理を実行します。

割込み処理ルーチン内で、ユーザプログラムによって R0:bit8 に "0" を設定することで割込み要求を解除し、タスクを切り換えます。

図 19.4-2 に遅延割込み発生モジュールの動作を示します。

図 19.4-2 遅延割込み発生モジュールの動作



## 19.5 遅延割込み発生モジュール使用上の注意

---

遅延割込み発生モジュールを使用する場合は、以下の点に注意してください。

---

### ■ 遅延割込み発生モジュール使用上の注意

割込み処理ルーチン内で遅延割込み要求発生 / 解除レジスタ (DIRR) の R0:bit8 に "0" を設定せずに、割込み処理から復帰した場合は、再度割込み処理を起動します。

遅延割込み発生モジュールにおける割込みは、ソフトウェア割込みとは異なり、遅延が伴います。

## 19.6 遅延割込み発生モジュールのプログラム例

遅延割込み発生モジュールのプログラム例を示します。

### ■ 遅延割込み発生モジュールのプログラム例

#### ● 処理仕様

メインプログラムで遅延割込み要因発生 / 解除レジスタ (DIRR) の R0:bit8 に "1" を設定し、遅延割込み要求を発生させ、タスク切換えを処理します。

#### ● コーディング例

```
ICR15 EQU 0000BFH ; 割込み制御レジスタ
DIRR EQU 00009FH ; 遅延割込み要求発生 / 解除レジスタ
DIRR_R0 EQU DIRR:0 ; 遅延割込み要求発生ビット
;----- メインプログラム -----
CODE CSEG
START: ; スタックポインタ (SP) などは初期化済みとする
    AND CCR, #0BFH ; 割込み禁止
    MOV I:ICR15, #00H ; 割込みレベル 0 (最強)
    MOV ILM, #07H ; PS 内の ILM をレベル 7 に設定
    OR CCR, #40H ; 割込み許可
    SETB I:DIRR_R0 ; 遅延割込み要求発生
LOOP MOV A, #00H ; 無限ループ
    MOV A, #01H
    BRA LOOP
;----- 割込みプログラム -----
WARI:
    CLRB I:DIRR_R0 ; 割込み要求フラグをクリア
    :
; ユーザ処理
; :
    RETI ; 割込みから復帰
CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
    ORG 0FF54H ; 割込み #42 (2AH) にベクタを設定
    DSL WARI
    ORG 0FFDCH ; リセットベクタ設定
    DSL START
    DB 00H ; シングルチップモードに設定
VECT ENDS
END START
```

# 第20章

## 8/10 ビット A/D コンバータ

8/10 ビット A/D コンバータの機能と動作について説明します。

- 20.1 8/10 ビット A/D コンバータの概要
- 20.2 8/10 ビット A/D コンバータのブロックダイアグラム
- 20.3 8/10 ビット A/D コンバータの端子
- 20.4 8/10 ビット A/D コンバータのレジスタ構成
- 20.5 8/10 ビット A/D コンバータの割込み
- 20.6 8/10 ビット A/D コンバータの動作
- 20.7 8/10 ビット A/D コンバータ使用上の注意
- 20.8 8/10 ビット A/D コンバータのプログラム例 1 (EI<sup>2</sup>OS を使用した場合の単発変換モード)
- 20.9 8/10 ビット A/D コンバータのプログラム例 2 (EI<sup>2</sup>OS を使用した場合の連続変換モード)
- 20.10 8/10 ビット A/D コンバータのプログラム例 3 (EI<sup>2</sup>OS を使用した場合の停止変換モード)



## 20.1 8/10 ビット A/D コンバータの概要

8/10 ビット A/D コンバータは、RC 逐次比較変換方式によりアナログ入力電圧を 10 ビットまたは 8 ビットのディジタル値へ変換します。入力信号は、アナログ入力端子の 8 チャンネルから選択します。変換はソフトウェア、16 ビットリロードタイマ 1 出力（立上りエッジ）および 16 ビットフリーランタイムゼロ検出エッジにより起動できます。

### ■ 8/10 ビット A/D コンバータの機能

8/10 ビット A/D コンバータは、アナログ入力端子へのアナログ電圧入力（入力電圧）をディジタル値へ変換します。8/10 ビット A/D コンバータの特長は、以下のとおりです。

- 最小変換時間は、6.13 $\mu$ s（この値は、マシクロックが 16MHz の場合の値であり、サンプリング時間を含んでいます）。
- 最小サンプリング時間は、3.75 $\mu$ s（この値は、マシクロックが 16MHz の場合の値です）。
- サンプルホールド回路における RC 逐次比較変換方式。
- 10 ビットまたは 8 ビットの分解能を選択可能。
- プログラムによるアナログ入力端子に対するチャンネル（最高 8 チャンネル可能）選択可能。
- A/D 変換終了時に、割込み要求の生成と EI<sup>2</sup>OS の起動可能。
- 割込み許可時、変換データ保護機能は連続変換によるデータ消失の発生を完全に防止。
- 変換は、ソフトウェア、16 ビットリロードタイマ 1 出力（立上りエッジ）および 16 ビットフリーランタイムゼロ検出エッジにより起動。

3 つの異なる変換モードを表 20.1-1 に示します。

表 20.1-1 8/10 ビット A/D コンバータの変換モード

	単一変換	スキャン変換
単発変換モード	設定したチャンネル（1 チャンネル）の入力を 1 回変換後終了。	2 つ以上の連続したチャンネル（最大 8 チャンネル）の入力を 1 回だけ変換後終了。
連続変換モード	設定したチャンネル（1 チャンネル）の入力を繰り返し変換。	2 つ以上の連続したチャンネル（最大 8 チャンネル）の入力を繰り返し変換。
停止変換モード	設定したチャンネル（1 チャンネル）を 1 回変換実行後、一時停止し、次の起動がかかるまで待機。	2 つ以上の連続したチャンネル（最大 8 チャンネル）を 1 回変換実行後、一時停止し、次の起動がかかるまで待機。

## ■ 8/10 ビット A/D コンバータ割込みと EI<sup>2</sup>OS

表 20.1-2 8/10 ビット A/D コンバータ割込みと EI<sup>2</sup>OS

割込み番号	割込み制御レジスタ		ベクトルテーブルアドレス			EI <sup>2</sup> OS
	レジスタ名	アドレス	下位	上位	バンク	
#11 (0B <sub>H</sub> )	ICR00	0000B0 <sub>H</sub>	FFFFD0 <sub>H</sub>	FFFFD1 <sub>H</sub>	FFFFD2 <sub>H</sub>	

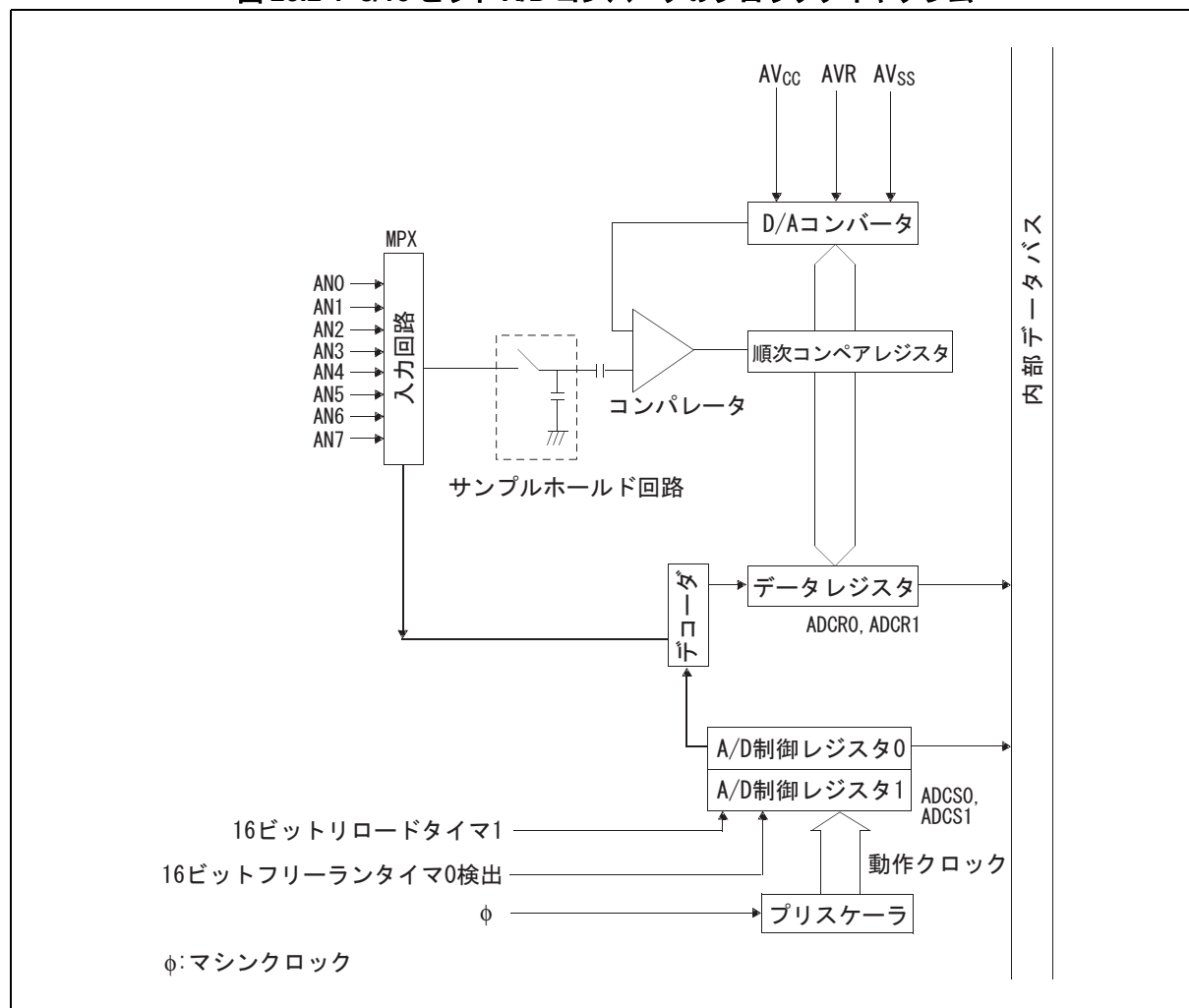
:EI<sup>2</sup>OS 割込みクリア信号により割込み要求フラグがクリアされると、使用可能。

## 20.2 8/10 ビット A/D コンバータのブロックダイアグラム

8/10 ビット A/D コンバータは，9 ブロックから構成されています。ブロックダイアグラムを図 20.2-1 に示します。

## ■ 8/10 ビット A/D コンバータのブロックダイアグラム

**図 20.2-1 8/10 ビット A/D コンバータのブロックダイヤグラム**



- ### ● A/D 制御ステータスレジスタ (ADCS0, ADCS1)

このレジスタは、A/D 変換モードの設定と A/D 変換開始/終了チャネルの設定を行う機能があります。

- ## ● A/D データレジスタ ( ADCR0, ADCR1 )

このレジスタは、A/D 変換起動要因の設定、割込みの許可 / 禁止の設定、割込み要求の確認、A/D 変換の停止中 / 動作中の確認を行う機能があります。

- クロックセレクト

クロックセレクトは、A/D 変換を起動するためのクロックを選択します。16 ビットリロードタイマチャネル 1 出力か 16 ビットフリーランタイムゼロ検出のいずれかを起動クロックとして使用できます。

- デコーダ

この回路は、A/D 制御ステータスレジスタ (ADCS0) の ANE2 ~ ANE0:bit2 ~ bit0 と ANS2 ~ ANS0:bit5 ~ bit3 の設定値に基づいて、使用するアナログ入力端子を選択します。

- アナログチャネルセレクト

この回路は、使用するピンを 8 つのアナログ入力端子の中から選択します。

- サンプルホールド回路

この回路は、アナログチャネルセレクトで選択されたチャネルの入力電圧を保持します。この回路は、A/D 変換が起動された直後に得た入力電圧をサンプリングし、保持することで A/D 変換中 (コンペア中) は、A/D 変換における入力電圧変動の影響を受けなくします。

- D/A コンバータ

この回路は、サンプルホールド回路で保持されている入力電圧と、比較される基準電圧を生成します。

- コンパレータ

この回路は、サンプルホールド回路で保持されている入力電圧を D/A コンバータの出力電圧と比較し、どちらが大きいかを判断します。

- 制御回路

この回路は、コンパレータで生成された信号に基づいて A/D 変換値を決定します。A/D 変換が完了すると、この回路は A/D データレジスタ (ADCR0, ADCR1) に変換結果を格納し、割込み要求を生成します。

## 20.3 8/10 ビット A/D コンバータの端子

8/10 ビット A/D コンバータの端子について説明します。また、端子のブロックダイヤグラムも示します。

### ■ 8/10 ビット A/D コンバータの端子

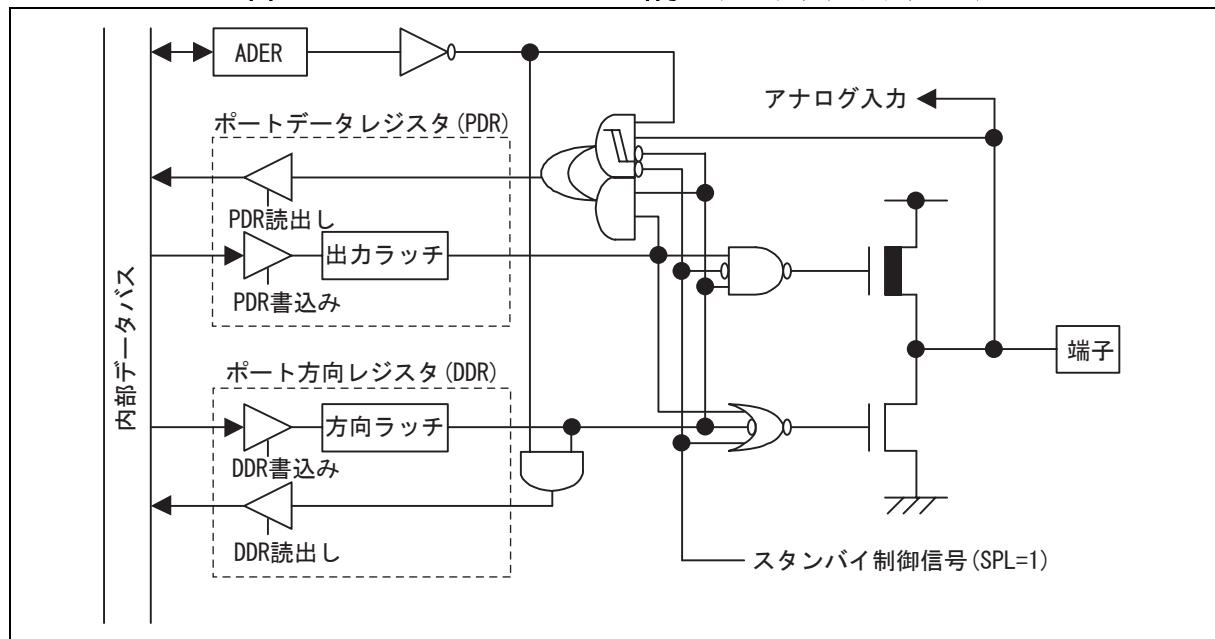
A/D コンバータの端子は、汎用端子としても使用されます。端子機能、I/O 形式および 8/10 ビット A/D コンバータを使用するために必要な設定値を表 20.3-1 に示します。

表 20.3-1 8/10 ビット A/D コンバータの端子

機 能	端子名	端子機能	入力信号種別	プルアップ 設定	スタンバイ 制御	端子を使用するために 必要な設定値
チャンネル 0	P50/AN0	ポート 5 入出力またはアナログ入力	CMOS 出力 / CMOS ヒステリシス入力または アナログ入力	なし	なし	ポート 5 を入力ポートとして設定します (DDR5:bit0 ~ bit7=0)。 ポート 5 をアナログ入力ポートとして設定します (ADER:bit0 ~ bit7=11111111 <sub>B</sub> )。
チャンネル 1	P51/AN1					
チャンネル 2	P52/AN2					
チャンネル 3	P53/AN3					
チャンネル 4	P54/AN4					
チャンネル 5	P55/AN5					
チャンネル 6	P56/AN6					
チャンネル 7	P57/AN7					

## ■ 8/10 ビット A/D コンバータの端子のブロックダイアグラム

図 20.3-1 P50/AN0 ~ P57/AN7 端子のブロックダイアグラム



- < 注意事項 >
- 端子を入力ポートとして使用する場合は、DDR5 レジスタの対応するビットに "0" を設定し、アナログ入力許可 (ADER) の対応するビットに "0" を設定してください。
  - 端子をアナログ入力端子として使用する場合は、アナログ入力許可 (ADER) レジスタの対応するビットに "1" を設定してください。PDR5 レジスタから読み出される値は "0" です。

## 20.4 8/10 ビット A/D コンバータのレジスタ構成

8/10 ビット A/D コンバータのレジスタ構成を示します。

### ■ 8/10 ビット A/D コンバータのレジスタ構成

図 20.4-1 8/10 ビット A/D コンバータのレジスタ構成

アナログ入力許可レジスタ								
	bit 7	6	5	4	3	2	1	0
	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 ⇨	1	1	1	1	1	1	1	1
A/D制御ステータスレジスタ1								
	bit 15	14	13	12	11	10	9	8
	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	RESV
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W
初期値 ⇨	0	0	0	0	0	0	0	0
A/D制御ステータスレジスタ0								
	bit 7	6	5	4	3	2	1	0
	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 ⇨	0	0	0	0	0	0	0	0
A/Dデータレジスタ1								
	bit 15	14	13	12	11	10	9	8
	S10	ST1	ST0	CT1	CT0	—	D9	D8
リード/ライト ⇨	R/W	W	W	W	W	—	R	R
初期値 ⇨	0	0	1	0	1	—	X	X
A/Dデータレジスタ0								
	bit 7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
リード/ライト ⇨	R	R	R	R	R	R	R	R
初期値 ⇨	X	X	X	X	X	X	X	X

## 20.4.1 A/D 制御ステータスレジスタ 1 (ADCS1)

A/D 制御ステータスレジスタ 1 (ADCS1) は、A/D 変換起動要因の設定、割込みの許可 / 禁止、割込み要求の確認および A/D 変換の停止中 / 動作中の確認を行う機能があります。

### ■ A/D 制御ステータスレジスタ 1 (ADCS1)

図 20.4-2 A/D 制御ステータスレジスタ 1 (ADCS1)

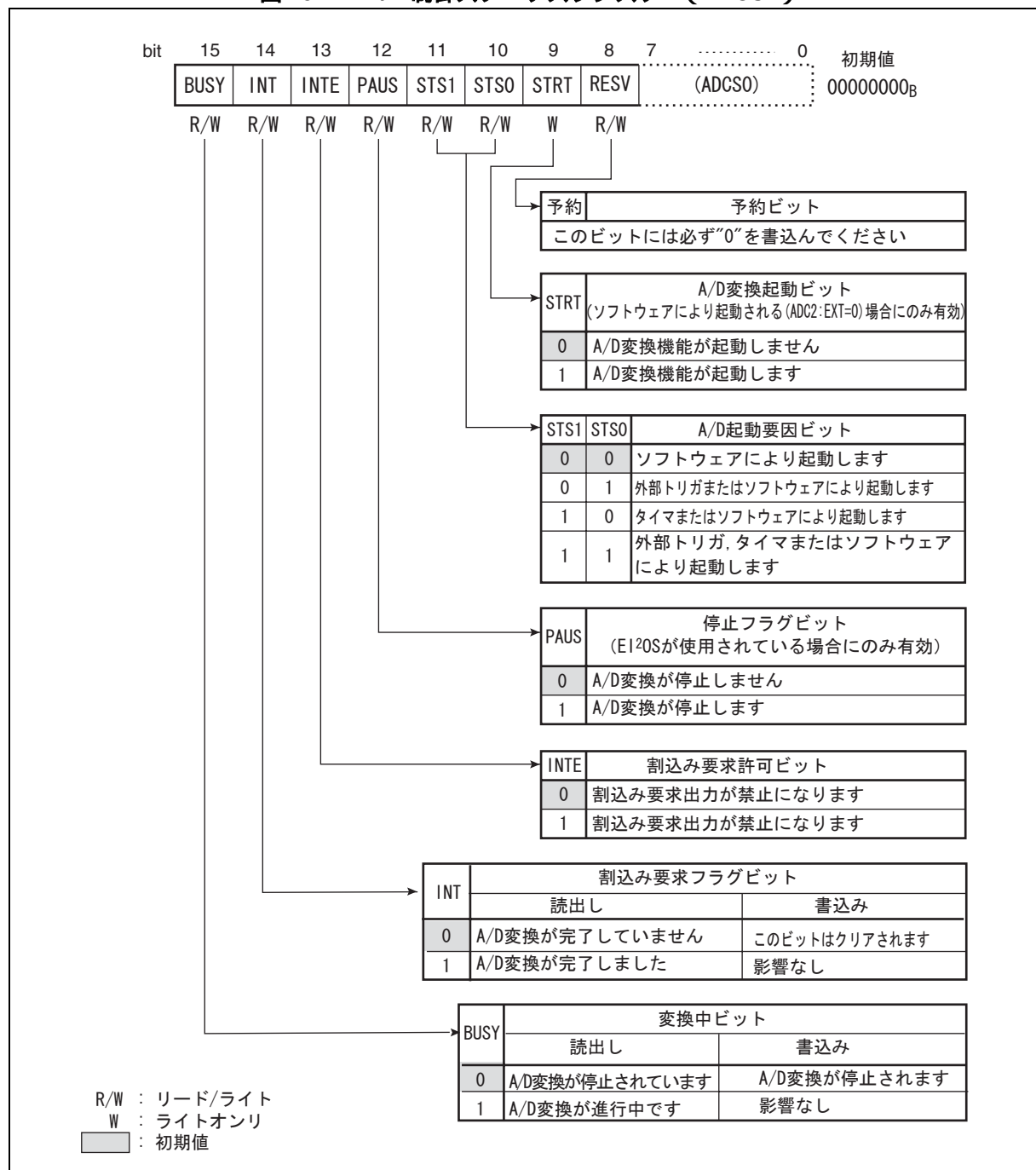




表 20.4-1 A/D 制御ステータスレジスタ 1 (ADCS1) の各ビットの機能説明 ( 1 / 2 )

ビット名		機 能
bit15	BUSY: 変換中ビット	<ul style="list-style-type: none"> <li>このビットは、A/D コンバータの動作状態を示します。</li> <li>このビットの読み出し値が "0" の場合、A/D 変換は停止されています。"1" の場合、A/D 変換動作中を示します。</li> <li><b>このビットに "0" を設定した場合：</b>A/D 変換は停止します。</li> <li><b>このビットに "1" を設定した場合：</b>このビットの値は変化せず、動作に影響しません。</li> </ul> <p>(注意事項)</p> <p>強制停止 (BUSY=0) とソフトウェアによる起動 (STRT=1) を同時に設定しないでください。</p>
bit14	INT: 割込み要求フラグビット	<ul style="list-style-type: none"> <li>A/D データレジスタに A/D 変換データが格納されると、このビットに "1" が設定されます。</li> <li>割込み要求許可ビット (INTE) に "1" が設定されている場合に "1" がセットされると、割込み要求が生成されます。EI<sup>2</sup>OS が許可になっている場合は、EI<sup>2</sup>OS が起動します。</li> <li><b>このビットに "0" を設定した場合：</b>このビットはクリアされます。</li> <li><b>このビットに "1" を設定した場合：</b>このビットの値は変化せず、動作に影響しません。</li> <li>EI<sup>2</sup>OS が起動すると、このビットはクリアされます。</li> </ul> <p>(注意事項)</p> <p>このビットに "0" を書き込むことによってこのビットをクリアする処理は、A/D コンバータが動作していないときにのみ行ってください。</p>
bit13	INTE: 割込み要求許可ビット	<ul style="list-style-type: none"> <li>このビットは、CPU への割込み出力を許可または禁止します。</li> <li><b>このビットに "1" を設定した場合：</b> 割込み要求フラグビット (INT) に "1" がセットされると割込み要求が生成されます。</li> <li>EI<sup>2</sup>OS を使用する場合は "1" を設定してください。</li> </ul>
bit12	PAUS: 停止フラグビット	<ul style="list-style-type: none"> <li>A/D 変換が一時停止すると、このビットには "1" が設定されます。</li> <li>この A/D コンバータは、A/D データレジスタを 1 つしか備えていません。したがって、連続変換モード時は、CPU が前の変換結果を読み出す前に新しい変換結果が書き込まれると、前の変換結果は失われます。連続変換モードが選択されると、変換が完了するたびに EI<sup>2</sup>OS が自動的に変換結果をメモリへ転送するようにしてください。このビットはまた、次の変換を開始する前に、変換データ転送の完了を複数の割込みが発生するのを防止します。変換が完了すると、このビットには "1" が設定されます。この状態は、EI<sup>2</sup>OS がデータレジスタのデータを転送し終えるまで保持されます。</li> <li><b>このビットに "1" を設定した場合：</b> A/D 変換は停止します。したがって、変換データを格納できません。 EI<sup>2</sup>OS が変換データの転送を完了すると、A/D コンバータは自動的に変換を再開します。</li> </ul> <p>(注意事項)</p> <p>このビットは、EI<sup>2</sup>OS が使用されている場合にのみ有効です。</p>
bit11, bit10	STS1, STS0: A/D 起動要因ビット	<ul style="list-style-type: none"> <li>これらのビットは、A/D 変換の起動方法を選択します。</li> <li>複数の起動要因が兼用されている場合、最初に発生した起動要因により起動されます。</li> </ul> <p>(注意事項)</p> <p>A/D 変換時におけるこれらのビットの設定変更後直ちに有効になるので、対応する起動要因が存在しない間にのみ行ってください。</p>

表 20.4-1 A/D 制御ステータスレジスタ 1 (ADCS1) の各ビットの機能説明 ( 2 / 2 )

ビット名		機 能
bit9	STRT: A/D 変換起動 ビット	<ul style="list-style-type: none"> <li>• このビットを使用すると、ソフトウェアは A/D 変換を起動できます。</li> <li>• <b>このビットに "1" を設定した場合</b>:A/D 変換を起動できます。</li> <li>• このビットは、停止変換モード時に変換の再起動はできません。</li> <li>• バイト/ワード命令では "1" が読み出されます。</li> <li>• リードモディファイライト系命令では "0" が読み出されます。</li> </ul> <p>( 注意事項 ) 強制停止 ( BUSY=0 ) とソフトウェアによる起動 ( STRT=1 ) を同時に設定しないでください。</p>
bit8	予約 : 予約ビット	このビットには必ず "0" を書き込んでください。

## 20.4.2 A/D 制御ステータスレジスタ 0 (ADCS0)

A/D 制御ステータスレジスタ 0 (ADCS0) は、変換モードと A/D 変換チャネルを選択します。

### ■ A/D 制御ステータスレジスタ 0 (ADCS0)

図 20.4-3 A/D 制御ステータスレジスタ 0 (ADCS0)

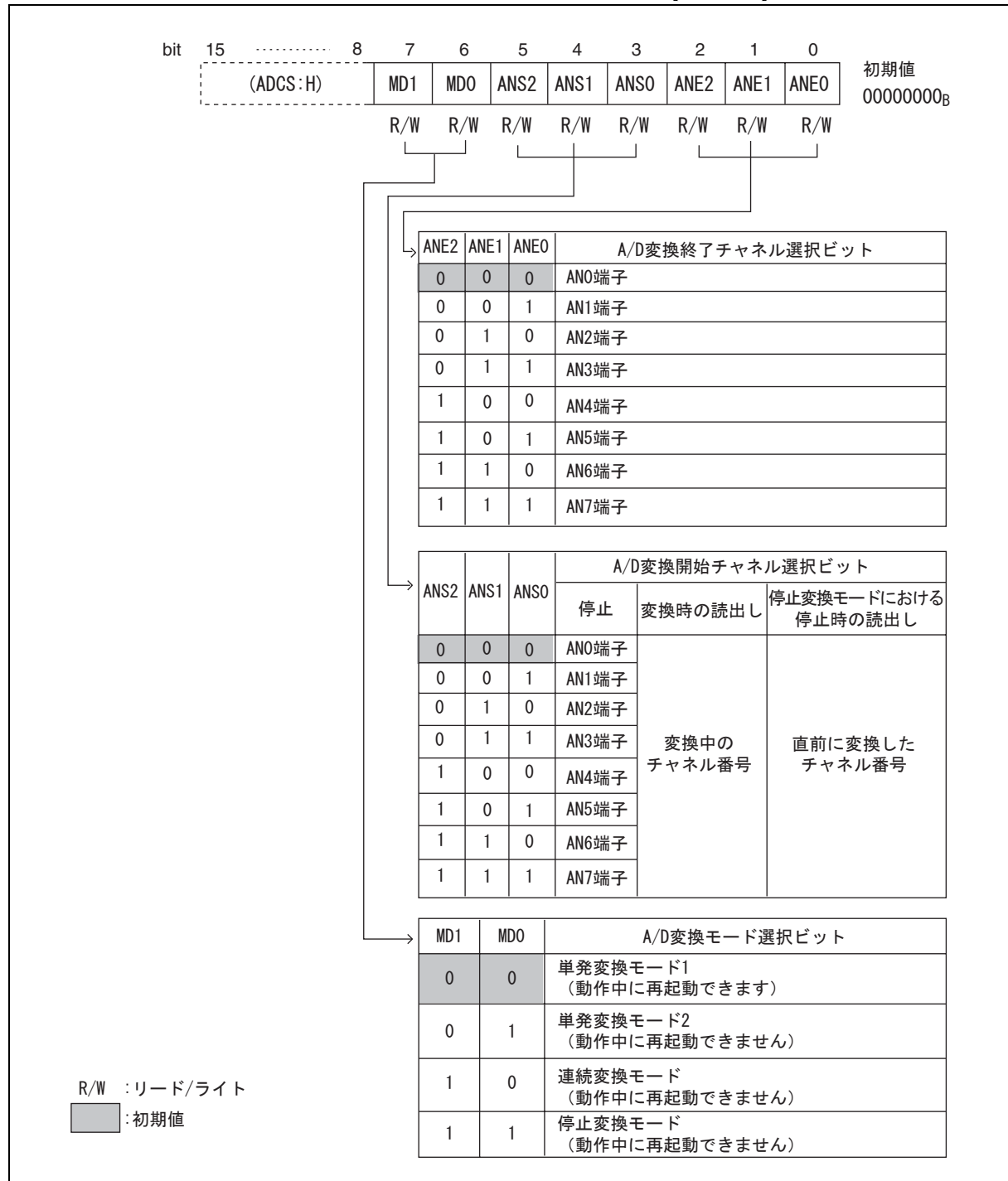


表 20.4-2 A/D 制御ステータスレジスタ 0 (ADCS0) の各ビットの機能説明 ( 1 / 2 )

ビット名	機 能
bit8, bit7	<p>MD1, MD0: A/D 変換モード選択ビット</p> <ul style="list-style-type: none"> <li>これらのビットは、A/D 変換機能の変換モードを選択します。</li> <li>MD1, MD0 ビットの値により、単発変換モード 1, 単発変換モード 2, 連続変換モードおよび停止変換モードの 4 つのモードを選択できます。</li> <li>各モードにおける動作を以下に説明します。</li> </ul> <p><b>単発変換モード 1:</b> ANS2 ~ ANS0:bit5 ~ bit3 で設定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で設定されたチャネルへの A/D 変換が 1 回のみ行われます。 動作中、再起動ができます。</p> <p><b>単発変換モード 2:</b> ANS2 ~ ANS0:bit5 ~ bit3 で設定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で設定されたチャネルへの A/D 変換が 1 回のみ行われます。 動作中、再起動はできません。</p> <p><b>連続変換モード:</b> ANS2 ~ ANS0:bit5 ~ bit3 で設定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で設定されたチャネルへの A/D 変換が繰返し行われます。この連続変換は、A/D 制御ステータスレジスタ 1 (ADCS1) の BUSY:bit15 で停止されるまで継続します。 動作中、再起動はできません。</p> <p><b>停止変換モード:</b> ANS2 ~ ANS0:bit5 ~ bit3 で設定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で設定されたチャネルへの A/D 変換が繰返し行われますが、各チャネルの変換が完了するたびに一時停止が発生します。この連続変換は、A/D 制御ステータスレジスタ 1 (ADCS1) の BUSY:bit15 で停止されるまで継続します。 動作中、再起動はできません。 一時停止が発生しているとき、A/D 制御ステータスレジスタ 1 (ADCS1) の STS1, STS0:bit11, bit10 で選択された起動要因が生成されると、変換は再起動されます。</p> <p>(注意事項) 単発変換モード 1, 単発変換モード 2, 連続変換モードおよび停止変換モード時は、16 ビットフリーランタイムゼロ検出, 16 ビットリロードタイマ 1, ソフトウェアすべての変換の再起動はできません。</p>

表 20.4-2 A/D 制御ステータスレジスタ 0 (ADCS0) の各ビットの機能説明 ( 2 / 2 )

ビット名	機 能
bit5 ～ bit3	<p>ANS2, ANS1, ANS0: A/D 変換開始 チャンネル選択 ビット</p> <ul style="list-style-type: none"> <li>これらのビットは、A/D 変換開始チャンネルの設定および変換中のチャンネル番号を表示します。</li> <li>A/D 変換を起動すると、これらのビットで指定されたチャンネルから変換を開始します。</li> <li>A/D 変換時、これらのビットは変換チャンネル番号を表示します。停止変換モードにおける一時停止時、これらのビットは直前の変換チャンネル番号を表示します。</li> <li>本ビットの読出し値は、本ビットに値を設定した場合でも、A/D 変換が開始されるまでは、設定した値ではなく、前回に A/D 変換したチャンネル番号が読み出されます。</li> <li>リセット時は、"000<sub>B</sub>" に初期化されます。</li> </ul> <p>( 注意事項 )</p> <ul style="list-style-type: none"> <li>A/D 変換開始チャンネル選択ビット (ANS2, ANS1, ANS0) を設定した後に、サンプリング時間選択ビット (ST1, ST0)、コンペア時間選択ビット (CT1, CT0) および A/D 変換終了チャンネル選択ビット (ANE2, ANE1, ANE0) をリードモディファイライト系命令で設定しないでください。</li> <li>ANS2, ANS1, ANS0 ビットは A/D 変換動作が開始するまでは前回の変換チャンネルが読み出されるため、ANS2, ANS1, ANS0 ビット設定後に ST1, ST0 ビット、CT1, CT0 ビットおよび ANE2, ANE1, ANE0 ビットをリードモディファイライト系命令で設定した場合、ANS2, ANS1, ANS0 ビットの値が書き換わる可能性があります。</li> </ul>
bit2 ～ bit0	<p>ANE2, ANE1, ANE0: A/D 変換終了 チャンネル選択 ビット</p> <ul style="list-style-type: none"> <li>これらのビットは、A/D 変換終了チャンネルを設定します。</li> <li>A/D 変換は、起動されると、これらのビットで指定されたチャンネルに対してまで行います。</li> <li>これらのビットが、ANS2 ～ ANS0 で指定されたチャンネルを指定すると、そのチャンネルのみを A/D 変換します。連続変換モードまたは停止変換モード時は、ANS2 ～ ANS0:bit5 ～ bit3 で指定された開始チャンネルの A/D 変換は、ANE2 ～ ANE0 で指定されたチャンネルの A/D 変換後に行われます。開始チャンネルの番号が終了チャンネルの番号より大きい場合、A/D 変換は、開始チャンネル、AN7, AN0、終了チャンネルの順序で行います。</li> </ul>

### 20.4.3 A/D データレジスタ (ADCR0, ADCR1)

A/D データレジスタ (ADCR0, ADCR1) は、A/D 変換結果の格納および A/D 変換分解能の選択を行います。

#### ■ A/D データレジスタ (ADCR0, ADCR1)

図 20.4-4 A/D データレジスタ (ADCR0, 1)

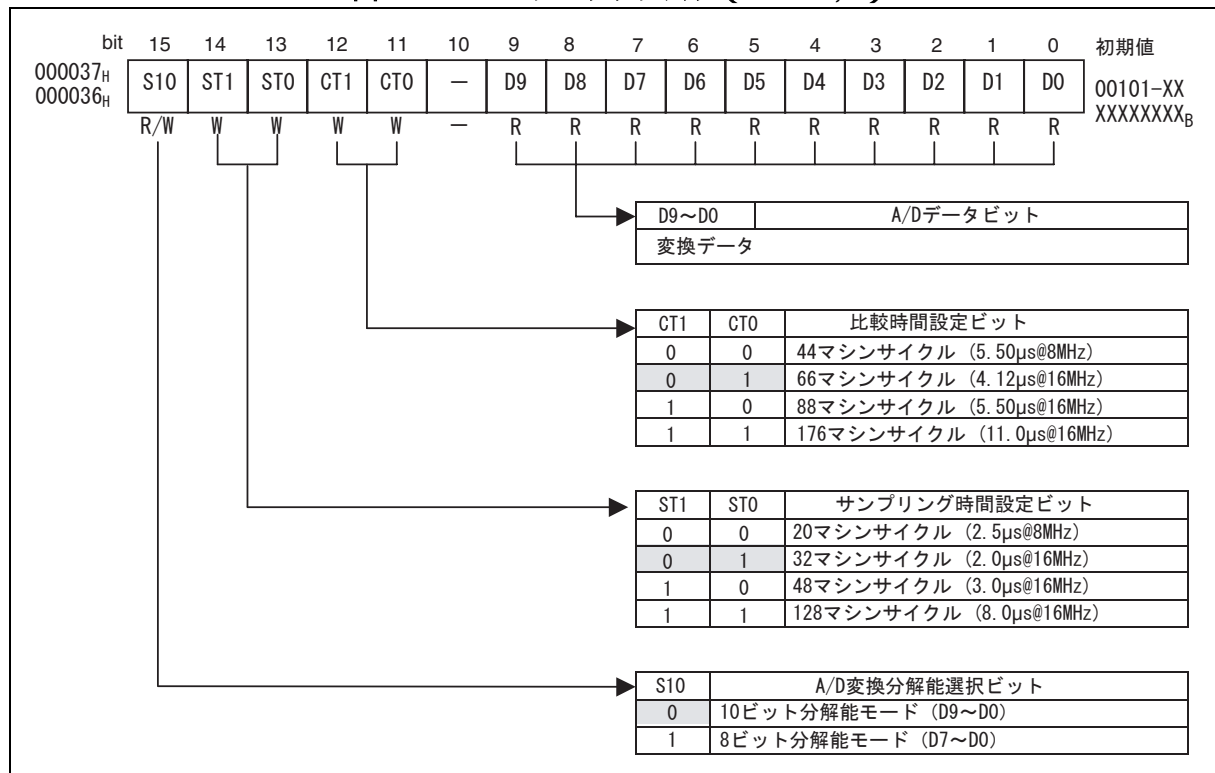


表 20.4-3 A/D データレジスタ (ADCR0, ADCR1) の各ビットの機能説明

ビット名		機 能
bit15	S10: A/D 変換分解 能選択ビット	<ul style="list-style-type: none"> <li>このビットは、A/D 変換の分解能を設定します。</li> <li>このビットに "0" を設定した場合:10 ビット分解能が設定されます。</li> <li>このビットに "1" を設定した場合:8 ビット分解能が設定されます。</li> </ul> (注意事項) 分解能によって使用される A/D データビットが異なります。
bit14, bit13	ST1, ST0: サンプリング 時間設定ビッ ト	<ul style="list-style-type: none"> <li>これらのビットは、A/D 変換のサンプリング時間を設定します。</li> <li>A/D 変換が起動されると、このビットで設定されている時間にアナログ入力 が取込まれます。</li> </ul> (注意事項) 16MHz 動作時にこれらのビットに "00 <sub>B</sub> " (8MHz) を設定すると、アナログ 電圧の取込みを正常に行えないことがあります。
bit12, bit11	CT1, CT0: 比較時間設定 ビット	<ul style="list-style-type: none"> <li>これらのビットは、A/D 変換の比較時間を設定します。</li> <li>アナログ入力が取込まれた後 (サンプリング時間経過後)、これらのビット で指定されている時間が経過後に、変換結果データが確定され、このレジス タの D9 ~ D0:bit9 ~ bit0 に格納されます。</li> </ul> (注意事項) 16MHz 動作時にこれらのビットに "00 <sub>B</sub> " (8MHz) を設定すると、正常なア ナログ変換値が得られない場合があります。
bit10	未使用ビット	<ul style="list-style-type: none"> <li>読出し値は不定です。</li> <li>このビットへの書き込みは、動作に影響しません。</li> </ul>
bit9 ~ bit0	D9 ~ D0: A/D データ ビット	<ul style="list-style-type: none"> <li>変換が終了するたびに、これらのビットに A/D 変換結果が格納され、レジス タが書換えられます。</li> <li>通常、最終の変換値が格納されます。</li> <li>このレジスタの初期値は不定です。</li> </ul> (注意事項) 変換データ保護機能があります (「20.6 8/10 ビット A/D コンバータの動作」 を参照)。A/D 変換中にこれらのビットへはデータを書き込まないでくださ い。

- S10:bit15 を書換える場合は、必ず A/D 変換動作前に A/D 変換動作を停止させてから  
書き換えてください。変換開始後にこのビットを書換えると、ADCR レジスタのデー  
タは不定となります。
- 10 ビットモード時に ADCR1, ADCR0 レジスタのデータを読み出す場合は、ワード  
転送命令 (MOVW A, 0036H など) を使用してください。

## 20.5 8/10 ビット A/D コンバータの割込み

8/10 ビット A/D コンバータは、A/D データレジスタに A/D 変換結果が格納されると、割込み要求を生成できます。拡張インテリジェント I/O サービス (EI<sup>2</sup>OS) にも対応しています。

### ■ 8/10 ビット A/D コンバータの割込み

8/10 ビット A/D コンバータの割込み制御ビットと割込み要因を表 20.5-1 に示します。

表 20.5-1 8/10 ビット A/D コンバータの割込み制御ビットと割込み要因

	8/10 ビット A/D コンバータ
割込み要求フラグビット	ADCS1:INT:bit14
割込み要求許可ビット	ADCS1:INTE:bit13
割込み要因	A/D データレジスタへの A/D 変換結果の書き込み

A/D 変換が行われ、A/D 変換結果が A/D データレジスタ (ADCR) へ書き込まれると、A/D 制御ステータスレジスタ (ADCS1) の INT:bit14 に "1" が設定されます。このとき、割込み要求が許可になっている (ADCS1 レジスタの INTE:bit13=1) 場合、割込みコントローラに割込み要求が出力されます。

### ■ 8/10 ビット A/D コンバータの割込みと EI<sup>2</sup>OS

表 20.5-2 8/10 ビット A/D コンバータの割込みと EI<sup>2</sup>OS

割込み番号	割込み制御レジスタ		ベクタテーブルアドレス			EI <sup>2</sup> OS
	レジスタ名	アドレス	下位	上位	バンク	
#11 (0B <sub>H</sub> )	ICR00	0000B0 <sub>H</sub>	FFFFD0 <sub>H</sub>	FFFFD1 <sub>H</sub>	FFFFD2 <sub>H</sub>	

: 使用可能

### ■ 8/10 ビット A/D コンバータの EI<sup>2</sup>OS 機能

8/10 ビット A/D コンバータは、EI<sup>2</sup>OS 機能を使用して A/D 変換結果をメモリへ転送できます。転送が行われると、変換データ保護機能は、A/D 変換データがメモリへ転送されるまで A/D 変換を停止し、ADCS1 レジスタの INT:bit14 をクリアします。変換データ保護機能は、データ欠落の発生を完全に防止できます。



## 20.6 8/10 ビット A/D コンバータの動作

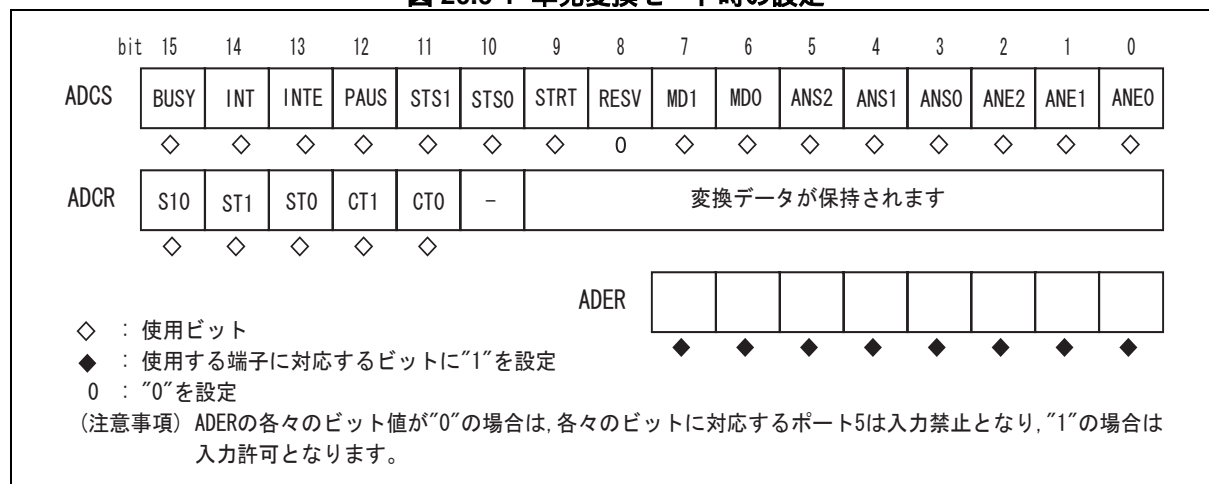
8/10 ビット A/D コンバータは、3 つの変換モード（単発変換モード、連続変換モード、停止変換モード）を備えています。各モードでの動作について説明します。

### ■ 単発変換モード時の動作

単発変換モード時は、A/D 制御ステータスレジスタ 0 (ADCS0) の ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャンネルから ADCS0 レジスタの ANE2 ~ ANE0:bit2 ~ bit0 で指定されたチャンネルまでのアナログ入力順に変換されます。ANE2 ~ ANE0:bit2 ~ bit0 で指定された終了チャンネルまでのすべてのチャンネルの変換が完了すると、A/D 変換が停止します。開始チャンネルと終了チャンネルが同じ (ANS2 ~ ANS0=ANE2 ~ ANE0) 場合は、ANS2 ~ ANS0 ビットで指定されたチャンネルのみを変換します。

単発変換モード時の動作に必要な設定値を図 20.6-1 に示します。

図 20.6-1 単発変換モード時の設定



単発変換モード時の変換順序の例を以下に示します。

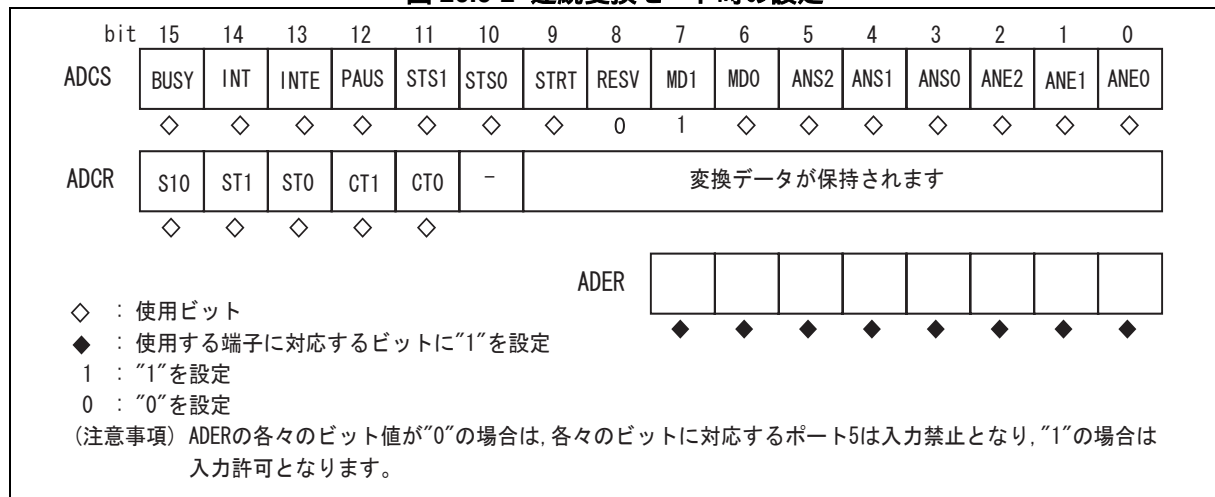
- ANS=000<sub>B</sub>, ANE=011<sub>B</sub>: AN0   AN1   AN2   AN3   End
- ANS=110<sub>B</sub>, ANE=010<sub>B</sub>: AN6   AN7   AN0   AN1   AN2   End
- ANS=011<sub>B</sub>, ANE=011<sub>B</sub>: AN3   End

## ■ 連続変換モード時の動作

連続変換モード時は、ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で指定されたチャネルまでのアナログ入力を順次 A/D 変換します。ANE2 ~ ANE0:bit2 ~ bit0 で指定された終了チャネルの変換が完了すると、A/D 変換は、ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャネルから再開されます。開始チャネルと終了チャネルが同じ (ANS2 ~ ANS0=ANE2 ~ ANE0) 場合は、ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャネルの変換が繰返されます。

連続変換モード時の動作に必要な設定値を図 20.6-2 に示します。

図 20.6-2 連続変換モード時の設定

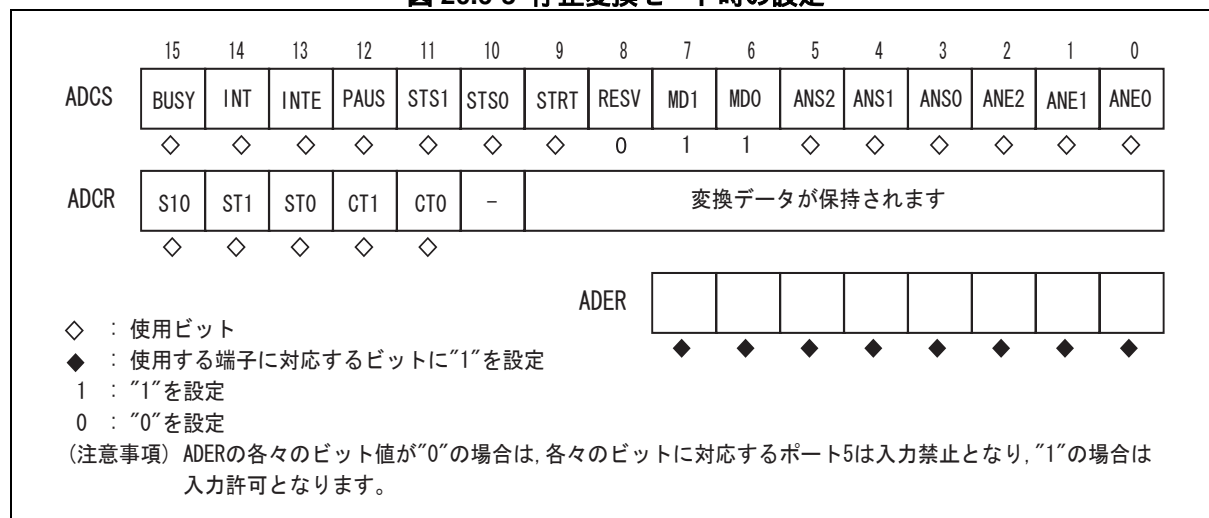


## ■ 停止変換モード時の動作

停止変換モード時は、ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャネルから ANE2 ~ ANE0:bit2 ~ bit0 で指定されたチャネルまでのアナログ入力を順次 A/D 変換しますが、各チャネルの変換が完了するたびに一時停止が発生します。ANE2 ~ ANE0:bit5 ~ bit3 で指定された終了チャネルの変換が完了すると、A/D 変換は、一時停止後、ANS2 ~ ANS0:bit5 ~ bit3 で指定されたチャネルから再開されます。開始チャネルと終了チャネルが同じ (ANS2 ~ ANS0=ANE2 ~ ANE0) 場合は、ANS ビットで指定されたチャネルの変換が繰返されます。一時停止時に A/D を再起動する場合には、STS1, STS0=bit11, bit10 で指定された起動要因を生成してください。

停止変換モード時の動作に必要な設定値を図 20.6-3 に示します。

**図 20.6-3 停止変換モード時の設定**



停止変換モード時の変換順序の例を以下に示します。

- ANS=000<sub>B</sub>, ANE=011<sub>B</sub>:AN0 一時停止 AN1 一時停止 AN2 一時停止  
AN0 反復
- ANS=110<sub>B</sub>, ANE=001<sub>B</sub>:AN6 Pause AN7 一時停止 AN0 一時停止  
AN1 AN6 反復
- ANS=011<sub>B</sub>, ANE=011<sub>B</sub>:AN3 一時停止 AN3 一時停止 繰返し

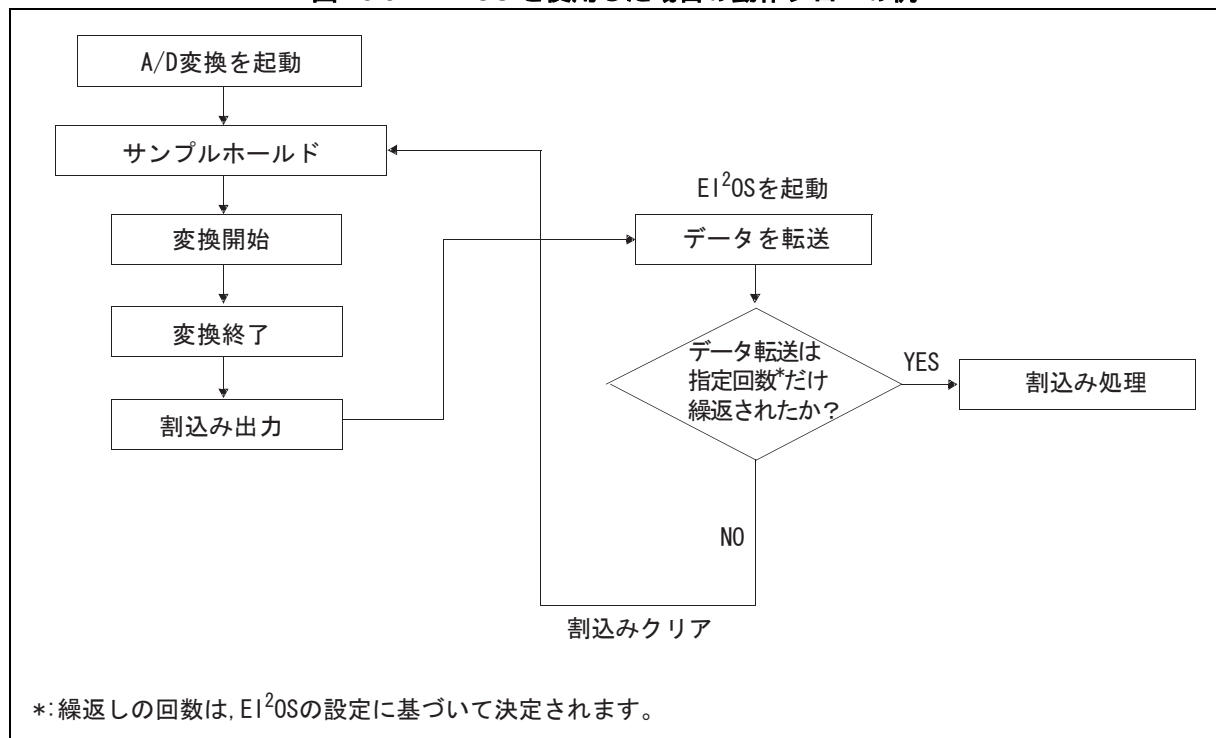
## 20.6.1 EI<sup>2</sup>OS を使用した場合の変換動作

8/10 ビット A/D コンバータは、EI<sup>2</sup>OS を使用して A/D 変換結果をメモリへ転送できます。

### ■ EI<sup>2</sup>OS を使用した場合の変換動作

EI<sup>2</sup>OS を使用した場合の動作フローを図 20.6-4 に示します。

図 20.6-4 EI<sup>2</sup>OS を使用した場合の動作フローの例



EI<sup>2</sup>OS を使用した場合は、変換データ保護機能が、連続変換モード時でもデータ欠落の発生を完全に防止します。したがって、メモリへの複数データ項目の転送を安全に行えます。

## 20.6.2 A/D 変換データ保護機能

---

割込み許可状態で A/D 変換を行うと、A/D 変換データ保護機能が動作します。

---

### ■ A/D 変換データ保護機能

A/D コンバータは、変換データを保持するデータレジスタを 1 つしか備えていません。1 つの A/D 変換が完了すると、このデータレジスタのデータは書き換えられます。

次の変換データが格納される前に変換データがメモリへ転送されなかった場合は、変換データの一部は失われます。A/D 変換データ保護機能は、割込みが許可になっているとき (ADCS1 レジスタの INTE:bit13=1)、以下で説明されているように動作し、データの消失を防止します。

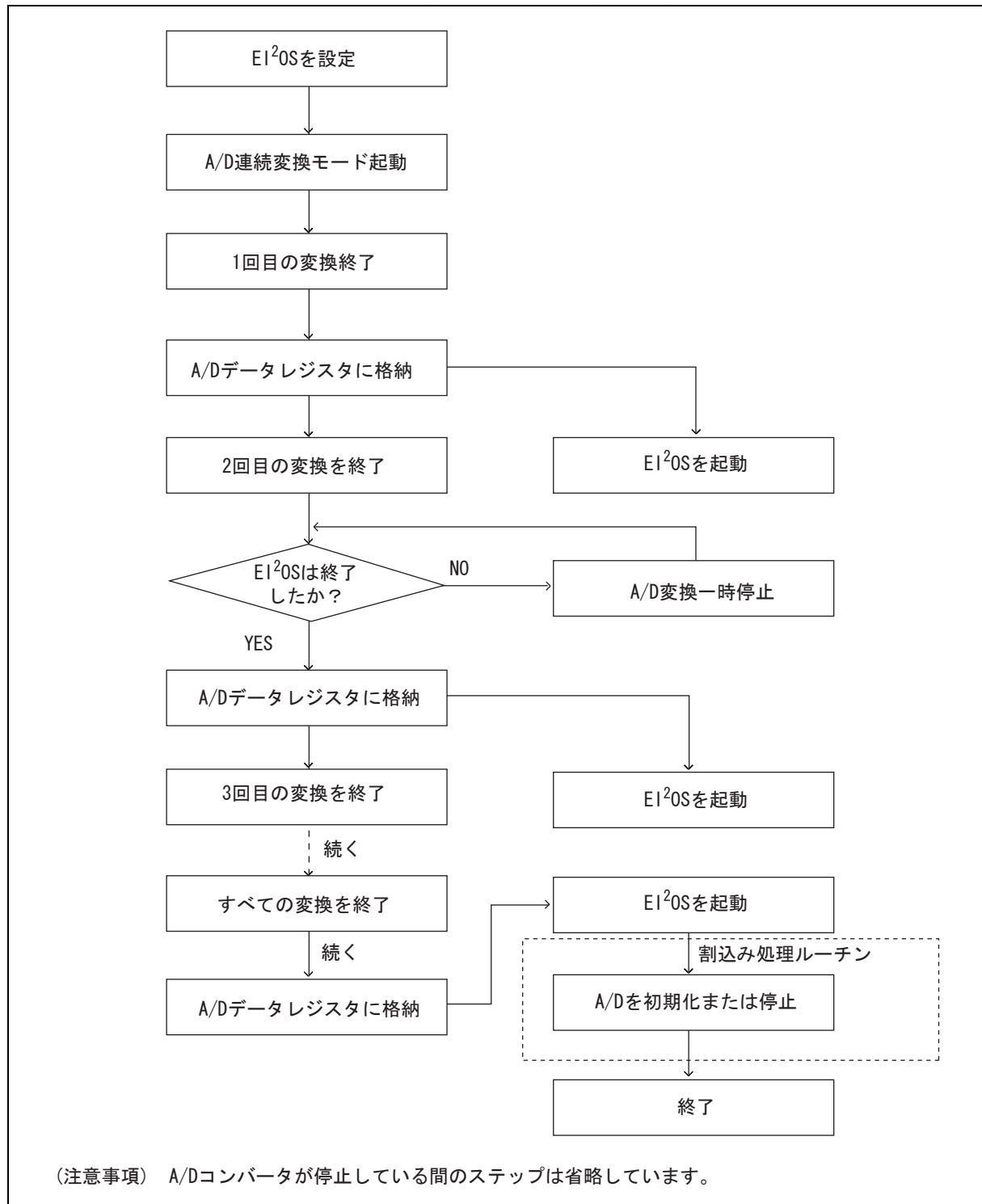
#### ● EI<sup>2</sup>OS を使用していない場合のデータ保護機能の動作

A/D データレジスタ (ADCR1, ADCR0) へ変換データが格納されると、A/D 制御ステータスレジスタ 1 (ADCS1) の INT:bit14 に "1" が設定されます。INT:bit14 に "1" が設定されると、A/D 変換が停止します。この停止状態は、割込みルーチンにより A/D データレジスタ (ADCR1, ADCR0) のデータがメモリへ転送された後、INT:bit14 がクリアされると、解除されます。

#### ● EI<sup>2</sup>OS を使用した場合のデータ保護機能の動作

EI<sup>2</sup>OS を使用した連続変換時は、変換が終了すると、A/D 制御ステータスレジスタ 1 (ADCS1) の PAUS:bit12 に "1" が設定されます。この状態は、EI<sup>2</sup>OS が変換データを A/D データレジスタからメモリへ転送し終えるまで続きます。その間、A/D 変換は停止され、次の変換データは格納されません。メモリへのデータ転送が完了すると、PAUS:bit12 は "0" にクリアされ、変換が再開されます。

EI<sup>2</sup>OS を使用した場合のデータ保護機能の動作フローを図 20.6-5 に示します。

図 20.6-5 EI<sup>2</sup>OS を使用した場合のデータ保護機能の動作フローチャート

- A/D 変換データ保護機能は、割り込みが許可状態 (ADCS1 レジスタの INTE:bit13=1) でのみ動作します。
- EI<sup>2</sup>OS が動作しているとき、A/D 変換における一時停止時に割り込みが禁止されると、A/D 変換が再開されることがあります。その結果、前のデータが転送される前に現データが書き込みされることになります。一時停止時に変換を再起動すると、古いデータは破壊されます。
- 一時停止時に変換を再起動すると、待機データは破壊されます。

## 20.7 8/10 ビット A/D コンバータ使用上の注意

---

8/10 ビット A/D コンバータ使用上の注意を以下に示します。

---

### ■ 8/10 ビット A/D コンバータ使用上の注意

#### ● アナログ入力端子

A/D 入力端子は、ポート 5 の I/O 端子としても使用されます。これらはポート 5 データレジスタ (DDR5) とアナログ入力許可レジスタ (ADER) は切り換えて使用します。端子をアナログ入力として使用するには、DDR5 の対応するビットに "0" を書き込み、ポート設定を "入力" に変更してください。次に、ADER レジスタにアナログ入力モード (ADEx=1) を設定し、ポートの入力ゲートを決定してください。ポート入力モード (ADEx=0) 時に中間レベルの信号を入力すると、入力ゲートにリーク電流が流れます。

#### ● 内部タイマ使用上の注意

内部タイマによって A/D コンバータを起動するには、A/D 制御ステータスレジスタ 1 (ADCS1) の STS1, STS0:bit11, bit10 で設定してください。内部タイマの入力値は、インアクティブレベル (内部タイマの場合は "L") に設定してください。内部タイマの入力値をインアクティブレベルに設定しない場合は、ADCS レジスタへの書込みと同時に動作が開始することがあります。

#### ● A/D コンバータの電源投入および切断順序

デジタル電源 ( $V_{CC}$ ) をオンにする前は、A/D コンバータ ( $AV_{CC}$ , AVR) の電源およびアナログ入力 (AN0 ~ AN7) の電源をオンにしないでください。

A/D コンバータの電源およびアナログ入力の電源をオフにする前は、デジタル電源 ( $V_{CC}$ ) をオフにしないでください。

#### ● A/D コンバータの電源電圧

A/D コンバータ ( $AV_{CC}$ ) への供給電圧は、デジタル電源 ( $V_{CC}$ ) を超えないでください。デジタル電源を超えた場合は、ラッチアップが発生する可能性があります。

## 20.8 8/10 ビット A/D コンバータのプログラム例 1 (EI<sup>2</sup>OS を使用した場合の単発変換モード)

EI<sup>2</sup>OS が使用されている単発変換モードにおいて A/D 変換を行うためのプログラム例を示します。

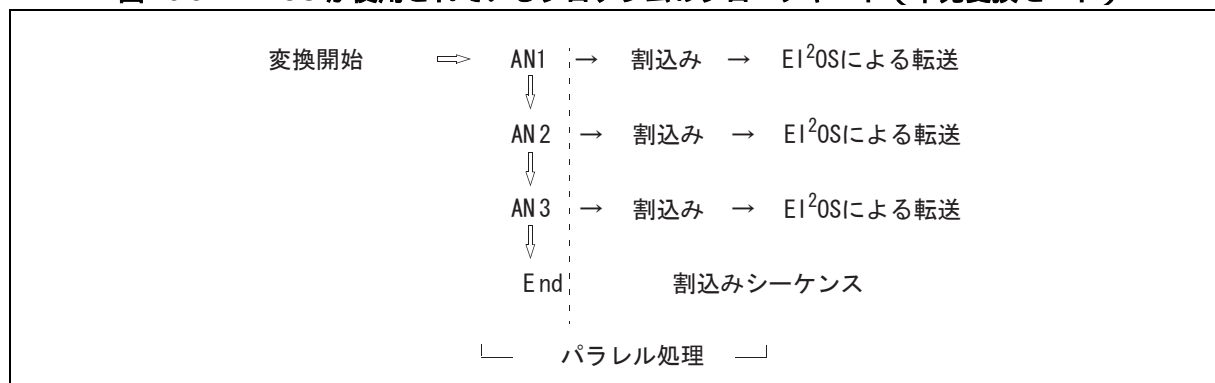
### ■ EI<sup>2</sup>OSが使用されている単発変換モードにおいてA/D変換を行うためのプログラム例

#### ● 処理

- アナログ入力 (AN1 ~ AN3) を 1 回変換します。
- 変換データをアドレス 200<sub>H</sub> ~ 205<sub>H</sub> へ順次転送します。
- 10 ビット分解能を選択します。
- 変換をソフトウェアにより起動します。

EI<sup>2</sup>OS が使用されているプログラムのフローチャート (単発変換モード) を図 20.8-1 に示します。

図 20.8-1 EI<sup>2</sup>OS が使用されているプログラムのフローチャート (単発変換モード)



#### ● コーディング例

BAPL	EQU	000100H	; バッファアドレスポインタ下位
BAPM	EQU	000101H	; バッファアドレスポインタ中位
BAPH	EQU	000102H	; バッファアドレスポインタ上位
ISCS	EQU	000103H	; EI <sup>2</sup> OS 状態レジスタ
IOAL	EQU	000104H	; I/O アドレスレジスタ下位
IOAH	EQU	000105H	; I/O アドレスレジスタ上位
DCTL	EQU	000106H	; データカウンタ下位
DCTH	EQU	000107H	; データカウンタ上位
DDR5	EQU	000015H	; ポート 5 方向レジスタ
ADER	EQU	000017H	; アナログ入力許可レジスタ
ICR00	EQU	0000B0H	; A/D 割込み制御レジスタ
ADCS0	EQU	000034H	; A/D 制御ステータスレジスタ
ADCS1	EQU	000035H	;
ADCR0	EQU	000036H	; A/D データレジスタ
ADCR1	EQU	000037H	;



```

;----- メインプログラム -----
CODE    CSEG
START:                                     ; スタックポインタ (SP) はすでに初期化されているもの
と
                                           ; 仮定します
        AND    CCR, #0BFH ; 割込みを禁止します
        MOV    ICR00, #00H ; 割込みレベル 0 (優先順位が最も高い)
        MOV    BAPL, #00H ; 変換データを転送し格納するアドレスを設定します
        MOV    BAPM, #02H ; (200H ~ 205H を使用します)
        MOV    BAPH, #00H ;
        MOV    ISCS, #18H ; ワードデータを転送し, 次にアドレスに "1" を加算し,
                                           ; 次にデータを I/O からメモリへ転送します
        MOV    IOAL, #36H ; アナログデータレジスタのアドレスを転送
        MOV    IOAH, #00H ; ソースアドレスポインタとして設定します
        MOV    DCTL, #03H ; EI2OS が行う転送の回数を 3 (変換回数と同じ値) に設定
                                           ; します
        MOV    DDR5, #11110001B; P51 ~ P53 を "入力" として設定します
        MOV    ADER, #00001110B; P51/AN1 ~ P53/AN3 をアナログ入力として設定し
ます
        MOV    DCTH, #00H ;
        MOV    ADCS0, #0BH ; 単一変換モード。AN1 ~ AN3 を変換します
        MOV    ADCS1, #0A2H; ソフトウェアによる起動。A/D 変換を開始します。
                                           ; 割込みを許可します
        MOV    ILM, #07H ; PS の ILM をレベル 7 に設定します
        OR     CCR, #40H ; 割込みを許可します
LOOP:    MOV    A, #00H ; 無限ループ
        MOV    A, #01H
        BRA    LOOP
;----- 割込みプログラム -----
ED_INT1:
        MOV    I:ADCS1, #00H; A/D 変換を終了します。割込みフラグをクリアし禁止しま
す
        RETI                                     ; 割込みから復帰します
CODE    ENDS
;----- ベクタ設定 -----
VECT    CSEG ABS=0FFH
        ORG    0FFD0H ; 割込み #11 (0BH) のベクタを設定します
        DSL    ED_INT1
        ORG    0FFDCH ; リセットベクタを設定します
        DSL    START
        DB     00H ;
VECT    ENDS
        END    START

```

## 20.9 8/10 ビット A/D コンバータのプログラム例 2 (EI<sup>2</sup>OS を使用した場合の連続変換モード)

EI<sup>2</sup>OS が使用されている連続変換モードにおいて A/D 変換を行うためのプログラム例を示します。

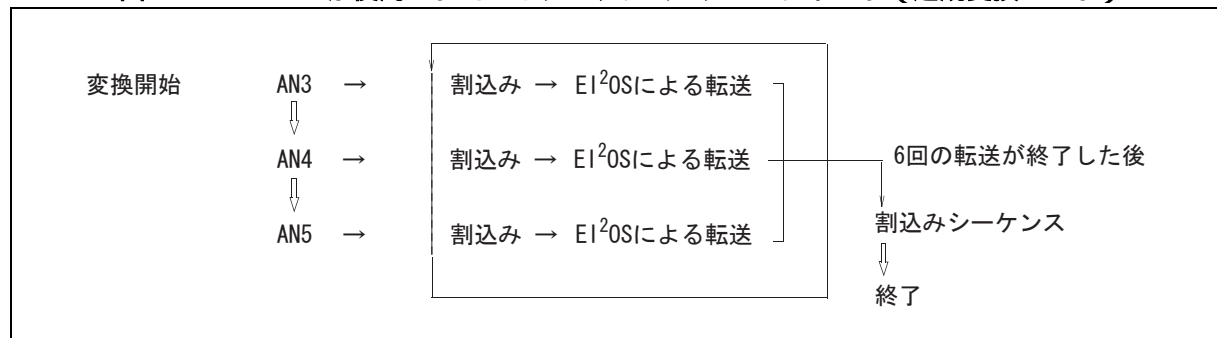
### ■ EI<sup>2</sup>OSが使用されている連続変換モードにおいてA/D変換を行うためのプログラム例

#### ● 処理

- アナログ入力 (AN3 ~ AN5) を 2 回変換します。各チャンネルに対して 2 つの変換データ項目を取得します。
- 変換データをアドレス 600<sub>H</sub> ~ 60B<sub>H</sub> へ順次転送します。
- 10 ビット分解能を選択します。
- 変換を 16 ビットリロードタイマ 1 で起動します。

EI<sup>2</sup>OS が使用されているプログラムのフローチャート (連続変換モード) を図 20.9-1 に示します。

図 20.9-1 EI<sup>2</sup>OS が使用されているプログラムのフローチャート (連続変換モード)



#### ● コーディング例

BAPL	EQU	000100H	; バッファアドレスポインタ下位
BAPM	EQU	000101H	; バッファアドレスポインタ中位
BAPH	EQU	000102H	; バッファアドレスポインタ上位
ISCS	EQU	000103H	; EI <sup>2</sup> OS 状態レジスタ
IOAL	EQU	000104H	; I/O アドレスレジスタ下位
IOAH	EQU	000105H	; I/O アドレスレジスタ上位
DCTL	EQU	000106H	; データカウンタ下位
DCTH	EQU	000107H	; データカウンタ上位
DDR5	EQU	000015H	; ポート 5 方向レジスタ
ADER	EQU	000017H	; アナログ入力許可レジスタ
ICR00	EQU	0000B0H	; A/D 割込み制御レジスタ
ADCS0	EQU	000034H	; A/D 制御状態レジスタ
ADCS1	EQU	000035H	;
ADCR0	EQU	000036H	; A/D データレジスタ
ADCR1	EQU	000037H	;
TMCSRL1	EQU	000086H	; 制御状態レジスタ 1 下位

## 第 20 章 8/10 ビット A/D コンバータ

```

TMCSRH1 EQU 000087H ;
TMRD1 EQU 000088H ;16 ビットリロードレジスタ 1
;----- メインプログラム -----
CODE CSEG
START: ; スタックポインタ (SP) はすでに初期化されているもの
と
; 仮定します
AND CCR, #0BFH ; 割込みを禁止します
MOV ICR10, #08H ; 割込みレベル 0 (優先順位が最も高い) 割込みを許可しま
す
MOV BAPL, #00H ; 変換データを格納するアドレスを設定します
MOV BAPM, #06H ; (600H ~ 60BH を使用します)
MOV BAPH, #00H ;
MOV ISCS, #18H ; ワードデータを転送し, 次にアドレスに "1" を加算し,
; 次にデータを I/O からメモリへ転送します
MOV IOAL, #36H ; アナログデータレジスタのアドレスを転送
MOV IOAH, #00H ; ソースアドレスポインタとして設定します
MOV DCTL, #06H ; EI2OS が転送を 6 回行います (3 チャンネルのそれぞれに対
; して転送を 2 回行います)
MOV DDR5, #00000000B; P50 ~ P57 を "入力" として設定します
MOV ADER, #00111000B; P53/AN3 ~ P55/AN5 をアナログ入力として設定し
ます
MOV DCTH, #00H ;
MOV ADCS0, #9DH ; 連続変換モード。AN3 ~ AN5 を変換します
MOV ADCS1, #0A8H; 16 ビットタイマを起動し, 次に A/D 変換を開始し, 次
に
; 割込みを許可します
MOVW TMRD1, #0320H; タイマ値として 800 (320H) (100 μs) を設定します
MOV TMCSRH1, #00H; クロックソースを 125ns に設定し, 次に外部トリガを
禁止
; します
MOV TMCSRL1, #12H; タイマ出力を禁止し, 次に割込みを禁止し, 次にリ
ロード
; を許可します
MOV TMCSRL1, #13H; 16 ビットリロードタイマ 1 を起動します
MOV ILM, #07H ; PS の ILM をレベル 7 に設定します
OR CCR, #40H ; 割込みを許可します
LOOP: MOV A, #00H ; 無限ループ
MOV A, #01H
BRA LOOP
;----- 割込みプログラム -----
ED_INT1:
MOV I:ADCS1, #80H ; A/D 変換を終了しません。割込みフラグをクリアし,
; 禁止します
RETI ; 割込みから復帰します
CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
ORG 0FFD0H ; 割込み #11 (0BH) のベクタを設定します
DSL ED_INT1
ORG 0FFDCH ; リセットベクタを設定します
DSL START
DB 00H ;
VECT ENDS

```

END    START

## 20.10 8/10 ビット A/D コンバータのプログラム例 3 (EI<sup>2</sup>OS を使用した場合の停止変換モード)

EI<sup>2</sup>OS が使用されている停止変換モードにおいて A/D 変換を行うためのプログラム例を示します。

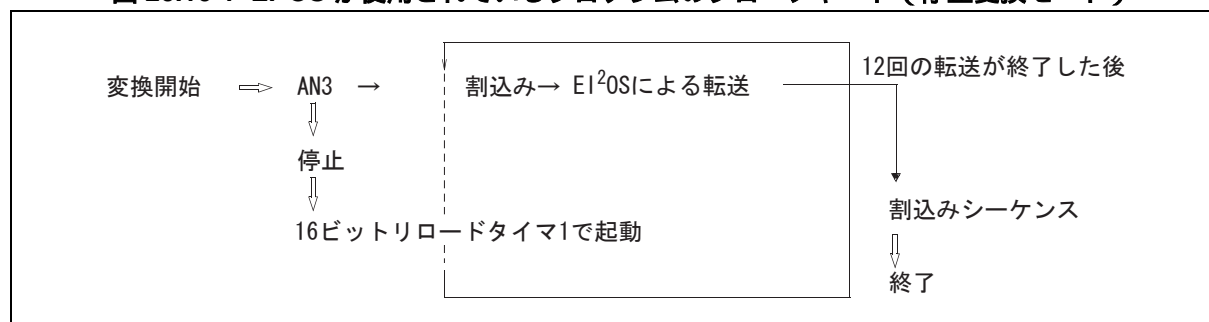
### ■ EI<sup>2</sup>OSが使用されている停止変換モードにおいてA/D変換を行うためのプログラム例

#### ● 処理

- アナログ入力 (AN3) を一定のインタバルで 12 回変換します。
- 変換データをアドレス 600<sub>H</sub> ~ 617<sub>H</sub> へ順次転送します。
- 10 ビット分解能を選択します。
- 変換を 16 ビットリロードタイマ 1 で起動します。

EI<sup>2</sup>OS が使用されているプログラムのフローチャート (停止変換モード) を図 20.10-1 に示します。

図 20.10-1 EI<sup>2</sup>OS が使用されているプログラムのフローチャート (停止変換モード)



#### ● コーディング例

BAPL	EQU	000100H	; バッファアドレスポインタ下位
BAPM	EQU	000101H	; バッファアドレスポインタ中位
BAPH	EQU	000102H	; バッファアドレスポインタ上位
ISCS	EQU	000103H	; EI <sup>2</sup> OS 状態レジスタ
IOAL	EQU	000104H	; I/O アドレスレジスタ下位
IOAH	EQU	000105H	; I/O アドレスレジスタ上位
DCTL	EQU	000106H	; データカウンタ下位
DCTH	EQU	000107H	; データカウンタ上位
DDR5	EQU	000015H	; ポート 5 方向レジスタ
ADER	EQU	000017H	; アナログ入力許可レジスタ
ICR00	EQU	0000B0H	; A/D 割り込み制御レジスタ
ADCS0	EQU	000034H	; A/D 制御状態レジスタ
ADCS1	EQU	000035H	;
ADCR0	EQU	000036H	; A/D データレジスタ
ADCR1	EQU	000037H	;
TMCSRL1	EQU	000086H	; 制御状態レジスタ 1 下位
TMCSRH1	EQU	000087H	;
TMRD1	EQU	000088H	; 16 ビットリロードレジスタ 1

```

;----- メインプログラム -----
CODE    CSEG
START:   ; スタックポインタ (SP) はすでに初期化されているものと
        ; 仮定します
        AND    CCR, #0BFH      ; 割り込みを禁止します
        MOV    ICR00, #08H     ; 割り込みレベル : TMCSSR1:L0 (優先順位が最も高い)
        MOV    BAPL, #00H     ; 変換データを格納するアドレスを設定します
        MOV    BAPM, #06H     ; (600H ~ 617H を使用します)
        MOV    BAPH, #00H     ;
        MOV    ISCS, #19H     ; ワードデータを転送し、次にアドレスに "1" を加算し、
        ; 次にデータを I/O からメモリへ転送し、次にリソース要
        ; 求を送出して終了します
        MOV    IOAL, #36H     ; アナログデータレジスタのアドレスを転送
        MOV    IOAH, #00H     ; ソースアドレスポインタとして設定します
        MOV    DCTL, #0CH     ; EI2OS が転送をチャネル 3 のみに対して 12 回行います
        MOV    DDR5, #00000000B ; P50 ~ P57 を " 入力 " として設定します
        MOV    ADER, #00001000B ; P53/AN3 をアナログ入力として設定します
        MOV    ADCS0, #0DBH    ; 停止変換モード。AN3 を変換します
        MOV    ADCS1, #0A8H    ; 16 ビットタイマを起動し、次に A/D 変換を開始し、次に
        ; 割り込みを許可します
        MOVW   TMRD1, #0320H   ; タイマ値として 800 (320H) (100 μs) に設定します
        MOV    TMCSRH1, #00H   ; クロックソースを 125ns に設定し、次に外部トリガを禁止
        ; します
        MOV    TMCSRL1, #12H   ; タイマ出力を禁止し、次に割り込みを禁止し、次にリロード
        ; を許可します
        MOV    TMCSRL1, #13H   ; 16 ビットリロードタイマ 1 を起動します
        MOV    ILM, #07H       ; PS の ILM をレベル 7 に設定します
        OR     CCR, #40H       ; 割り込みを許可します
LOOP:    MOV    A, #00H        ; 無限ループ
        MOV    A, #01H
        BRA    LOOP
;----- 割り込みプログラム -----
ED_INT1:
        MOV    I:ADCS1, #80H   ; A/D 変換を終了しません。割り込みフラグをクリアし、禁止
        ; します
        RETI                   ; 割り込みから復帰します
CODE    ENDS
;----- ベクタ設定 -----
VECT    CSEG ABS=0FFH
        ORG    0FFD0H          ; 割り込み #11 (0BH) のベクタを設定します
        DSL    ED_INT1
        ORG    0FFDCH          ; リセットベクタを設定します
        DSL    START
        DB     00H             ;
VECT    ENDS
        END    START

```



# 第21章

---

## アドレス一致検出機能

アドレス一致検出機能の機能と動作について説明します。

- 21.1 アドレス一致検出機能の概要
- 21.2 アドレス一致検出機能のブロックダイアグラム
- 21.3 アドレス一致検出機能のレジスタ構成
- 21.4 アドレス一致検出機能の動作説明
- 21.5 アドレス一致検出機能のプログラム例



## 21.1 アドレス一致検出機能の概要

---

アドレス一致検出機能は、プログラムが現在処理している命令の次に処理される命令のアドレスが、プログラムアドレス検出レジスタに設定したアドレスと一致した場合に、プログラムで次に処理される命令を強制的に INT9 命令に置換え、割込み処理プログラムへ分岐する機能です。INT9 割込みを利用できるので、本機能はプログラムのパッチ処理による修正に利用できます。

---

### ■ アドレス一致検出機能の概要

- プログラムが現在処理している命令の次に処理される命令のアドレスは、内部データバスを介して常にアドレスラッチに保持されます。アドレス一致検出機能により、アドレスラッチに保持されたアドレス値と、検出アドレス設定レジスタに設定したアドレス値は常に比較されます。比較されたアドレス値が一致した場合は、CPU が次に実行する命令は強制的に INT9 命令に置換えられ、割込み処理プログラムが実行されます。
- プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L) は 2 つあり、レジスタごとに割込み許可ビットが用意されています。アドレスラッチに保持されたアドレスと検出アドレス設定レジスタに設定したアドレスの一致による割込みの発生を、レジスタごとに許可または禁止できます。

## 21.2 アドレス一致検出機能のブロックダイアグラム

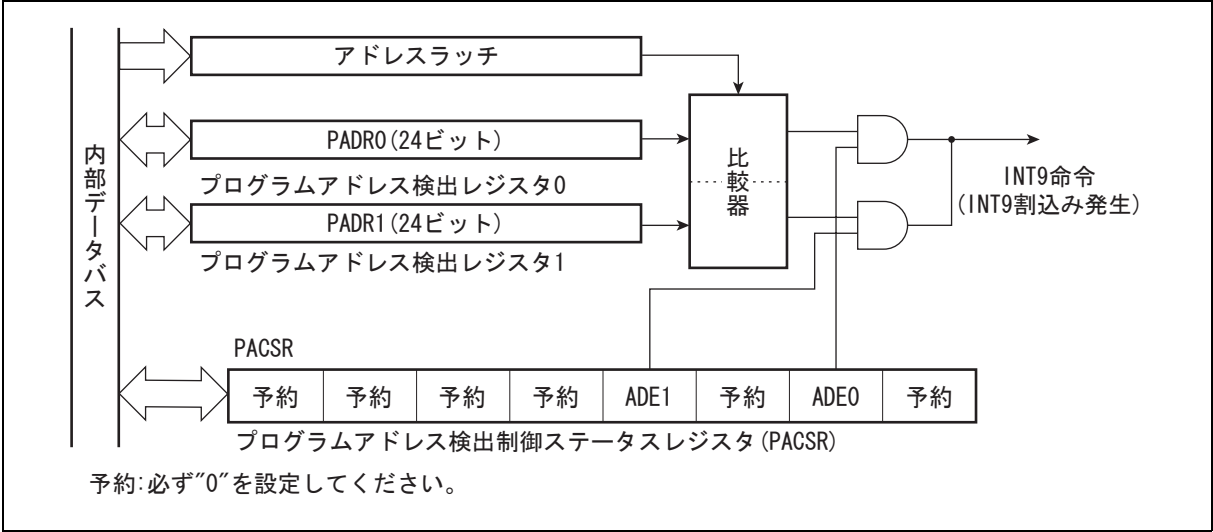
アドレス一致検出モジュールは、以下のブロックで構成されています。

- アドレスラッチ
- プログラムアドレス検出制御ステータスレジスタ（PACSR）
- プログラムアドレス検出レジスタ（RADR）

### ■ アドレス一致検出機能のブロックダイアグラム

アドレス一致検出機能のブロックダイアグラムを図 21.2-1 に示します。

図 21.2-1 アドレス一致検出機能のブロックダイアグラム



- アドレスラッチ  
内部データバスへ出力されたアドレス値を保持します。
- プログラムアドレス検出制御ステータスレジスタ（PACSR）  
アドレスが一致した場合の割込み出力の許可または禁止を設定します。
- プログラムアドレス検出レジスタ（PADR0H, M, L, PADR1H, M, L）  
アドレスラッチの値と比較するためのアドレスを設定します。

## 21.3 アドレス一致検出機能のレジスタ構成

アドレス一致検出機能で使用するレジスタの一覧と詳細を記述します。

### ■ アドレス一致検出機能のレジスタとリセット値の一覧

図 21.3-1 アドレス一致検出機能のレジスタとリセット値の一覧

プログラムアドレス検出制御ステータス レジスタ (PACSR)	bit 7 6 5 4 3 2 1 0	0 0 0 0 0 0 0 0
プログラムアドレス検出レジスタ0 (PADR0H) : 上位	bit 7 6 5 4 3 2 1 0	X X X X X X X X
プログラムアドレス検出レジスタ0 (PADR0M) : 中位	bit 15 14 13 12 11 10 9 8	X X X X X X X X
プログラムアドレス検出レジスタ0 (PADR0L) : 下位	bit 7 6 5 4 3 2 1 0	X X X X X X X X
プログラムアドレス検出レジスタ1 (PADR1H) : 上位	bit 7 6 5 4 3 2 1 0	X X X X X X X X
プログラムアドレス検出レジスタ1 (PADR1M) : 中位	bit 15 14 13 12 11 10 9 8	X X X X X X X X
プログラムアドレス検出レジスタ1 (PADR1L) : 下位	bit 7 6 5 4 3 2 1 0	X X X X X X X X
X: 不定		

### 21.3.1 プログラムアドレス検出制御ステータスレジスタ (PACSR)

アドレス一致による割込み出力の許可または禁止を設定します。アドレス一致による割込み出力を許可した場合にアドレス一致が検出されると、INT9 の割込みが出力されます。

■ プログラムアドレス検出制御ステータスレジスタ (PACSR)

図 21.3-2 プログラムアドレス検出制御ステータスレジスタ (PACSR)

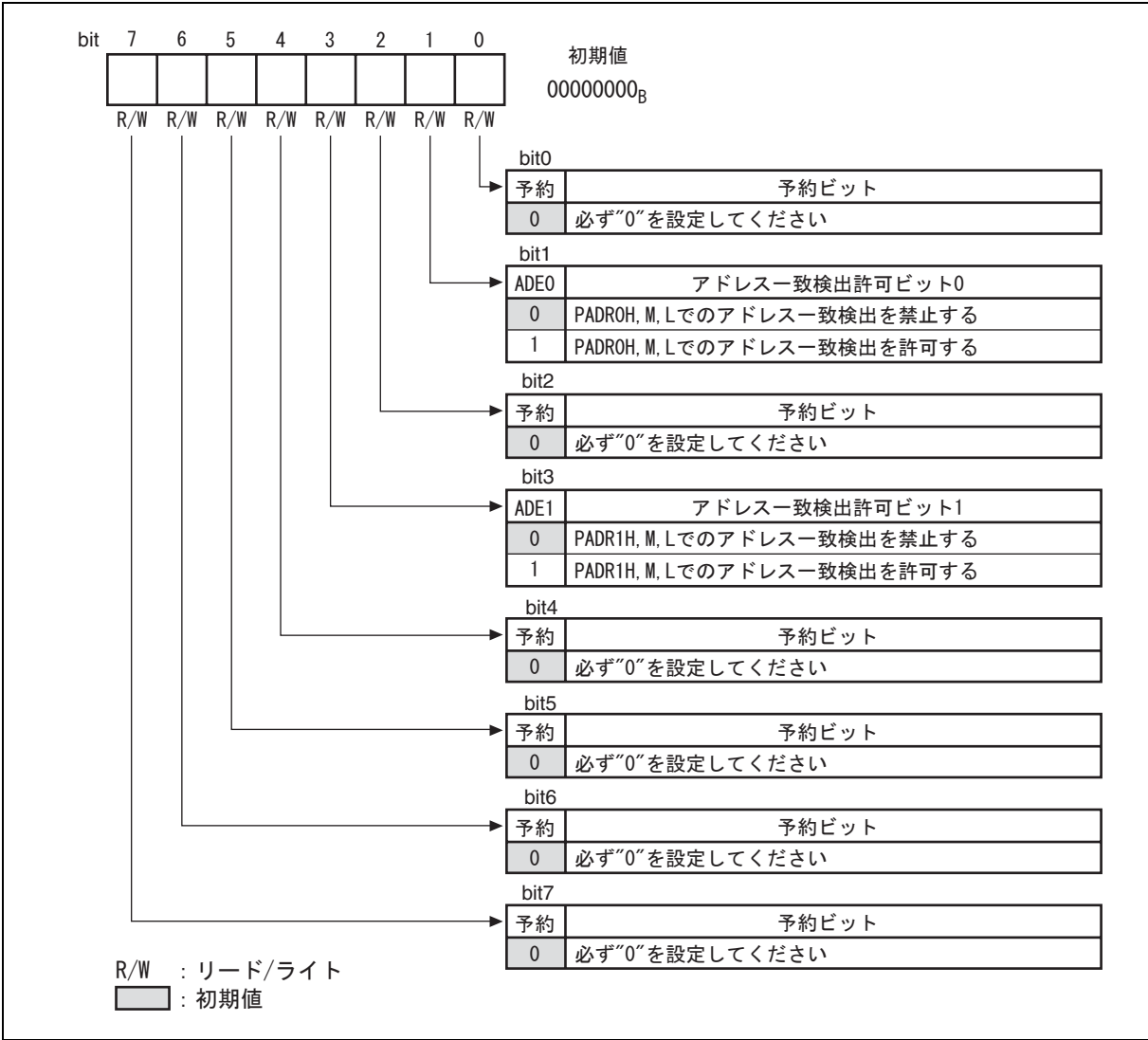


表 21.3-1 プログラムアドレス検出制御ステータスレジスタ (PACSR) の機能

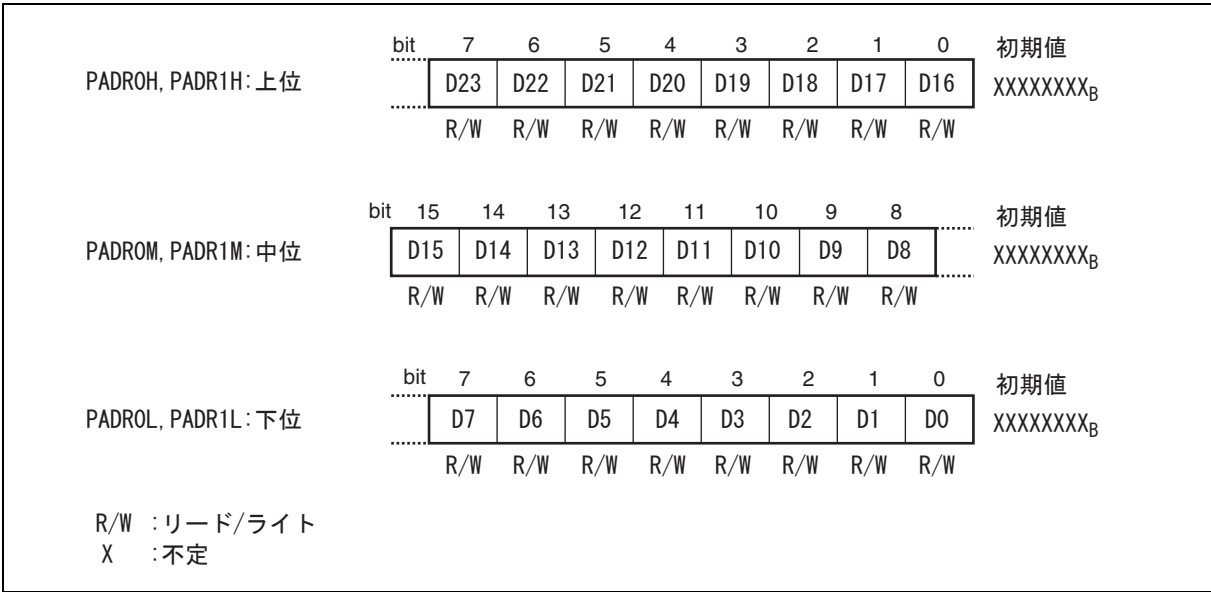
ビット名		機 能
bit14 ~ bit7	予約： 予約ビット	必ず "0" を設定してください。
bit3	ADE1: アドレス一致 検出許可ビッ ト 1	<p>プログラムアドレス検出レジスタ 1 (PADR1H, M, L) とのアドレス一致検出動作を許可または禁止します。</p> <p><b>このビットに "0" を設定した場合:</b> アドレス一致検出動作を禁止します。</p> <p><b>このビットに "1" を設定した場合:</b> アドレス一致検出動作を許可します。</p> <ul style="list-style-type: none"> <li>• アドレス一致検出動作を許可した場合に (ADE1=1), アドレスラッチの値とプログラムアドレス検出レジスタ 1 (PADR1H, M, L) の値が一致した場合は, 直ちに INT9 命令が実行されます。</li> </ul>
bit2	予約： 予約ビット	必ず "0" を設定してください。
bit1	ADE0: アドレス一致 検出許可ビッ ト 0	<p>プログラムアドレス検出レジスタ 0 (PADR0H, M, L) とのアドレス一致検出動作を許可または禁止します。</p> <p><b>このビットに "0" を設定した場合:</b> アドレス一致検出動作を禁止します。</p> <p><b>このビットに "1" を設定した場合:</b> アドレス一致検出動作を許可します。</p> <ul style="list-style-type: none"> <li>• アドレス一致検出動作を許可した場合に (ADE0=1), アドレスラッチの値とプログラムアドレス検出レジスタ 0 (PADR0H, M, L) の値が一致した場合は, 直ちに INT9 命令が実行されます。</li> </ul>
bit0	予約： 予約ビット	必ず "0" を設定してください。

### 21.3.2 プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L)

プログラムアドレス検出レジスタには、検出するアドレス値を設定します。プログラムで実行されている命令のアドレスがプログラムアドレス検出レジスタに設定されたアドレスと一致すると、次に実行される命令は強制的に INT9 命令に置換えられ、割込み処理プログラムが実行されます。

■ プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L)

図 21.3-3 プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L)



## ■ プログラムアドレス検出レジスタの機能

- プログラムアドレス検出レジスタは 2 本 (PADR0H, M, L, PADR1H, M, L) 用意されており、それぞれ上位、中位、下位の 3 バイト、合計 24 ビットで構成されています。

表 21.3-2 プログラムアドレス検出レジスタのアドレス設定

レジスタ名	割込み出力許可	アドレス設定	
プログラムアドレス 検出レジスタ 0 (PADR0H, M, L)	PACSR:ADE0	上位	プログラムアドレス 0 の上位 8 ビットを設定 (バンク)
		中位	プログラムアドレス 0 の中位 8 ビットを設定
		下位	プログラムアドレス 0 の下位 8 ビットを設定
プログラムアドレス 検出レジスタ 1 (PADR1H, M, L)	PACSR:ADE1	上位	プログラムアドレス 1 の上位 8 ビットを設定 (バンク)
		中位	プログラムアドレス 1 の中位 8 ビットを設定
		下位	プログラムアドレス 1 の下位 8 ビットを設定

- プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L) には、INT9 命令と置換える命令の先頭アドレス (1 バイト目) を設定する必要があります。

図 21.3-4 INT9 命令と置換える命令コードの先頭アドレスの設定

アドレス	命令コード	検出アドレスに設定 (上位:FF <sub>H</sub> , 中位:00 <sub>H</sub> , 下位:1F <sub>H</sub> )	ニモニック
FF001C :	A8 00 00		MOVW RWO, #0000
FF001F :	4A 00 00		MOVW A, #0000
FF0022 :	4A 80 08		MOVW A, #0880

- < 注意事項 >
- プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L) に、1 バイト目以外のアドレスを設定した場合は、命令コードが INT9 命令に置換えられず、割込み処理プログラムは実行されません。また、2 バイト目以降に設定した場合は、命令コードで指定した番地が "01<sub>H</sub>" (INT9 命令コード) に置換わってしまい、誤動作を起こす原因になる場合があるの注意してください。
  - プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L) は、対応するプログラムアドレス一致制御レジスタのアドレス一致検出動作を禁止してから (PACSR レジスタの ADE0:bit1=0 または ADE1:bit1=0) 設定してください。アドレス一致検出動作を禁止せずにプログラムアドレス検出レジスタを変更した場合、アドレスの書き込み中にアドレスが一致すると直ちにアドレス一致検出機能が働いてしまい、誤動作を起こす原因になります。
  - アドレス一致検出機能は、内部 ROM のアドレスに対してのみ使用できます。外部メモリ領域のアドレスを設定してもアドレス一致検出機能は働かず、INT9 命令は実行されません。

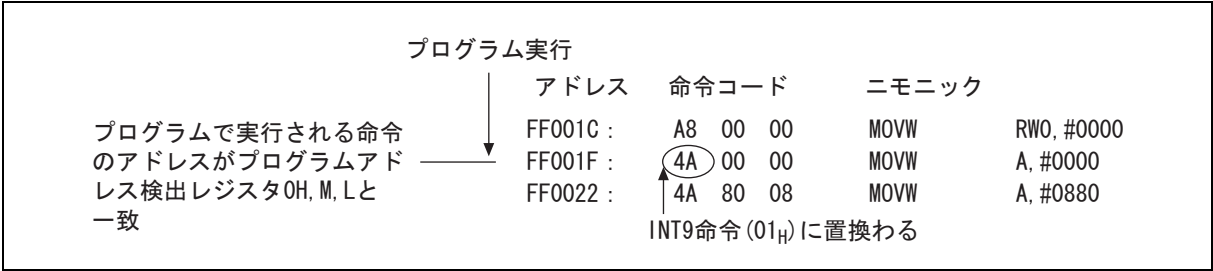
## 21.4 アドレス一致検出機能の動作説明

アドレス一致検出機能は、プログラムで実行される命令のアドレスが、プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L) で設定したアドレスと一致した場合に、CPU で実行される先頭の命令コードを INT9 (01<sub>H</sub>) 命令に置換え、割り込み処理プログラムへ分岐します。

### ■ アドレス一致検出機能の動作

検出アドレスの設定と、アドレス一致が検出された場合の動作を、図 21.4-1 に示します。

図 21.4-1 アドレス一致検出機能の動作



### ■ 検出アドレスの設定

1. 検出アドレスを設定するプログラムアドレス検出レジスタ 0 (PADR0H, M, L) のアドレス一致検出動作を禁止してください (PACSR レジスタの ADE0:bit1=0)。
2. プログラムアドレス検出レジスタ 0 (PADR0H, M, L) に、検出するアドレスを設定します。プログラムアドレス検出レジスタ 0 (PADR0H, M, L) の上位に "FF<sub>H</sub>", 中位に "00<sub>H</sub>", 下位に "1F<sub>H</sub>" を設定してください。
3. 検出アドレスを設定するプログラムアドレス検出レジスタ 0 (PADR0H, M, L) のアドレス一致検出動作を許可してください (PACSR レジスタの ADE0:bit1=1)。

### ■ プログラムの実行

1. プログラムで実行される命令のアドレスが、設定した検出アドレスと一致すると、一致したアドレスの先頭の命令コードが INT9 命令コード ("01<sub>H</sub>") に置換わります。
2. INT9 命令が実行されます。INT9 割り込みが発生し、割り込み処理プログラムが実行されます。



## 21.4.1 アドレス一致検出機能の使用例

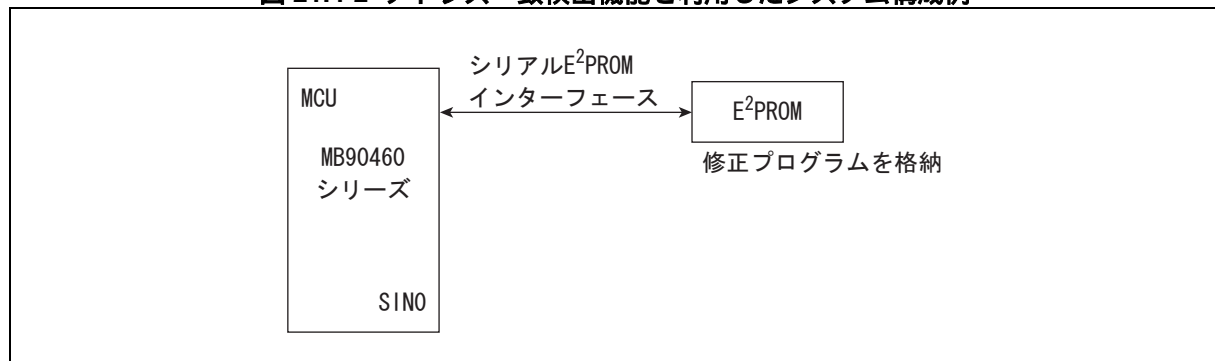
アドレス一致検出機能を利用した，プログラム修正のパッチ処理の例を示します。

### ■ システム構成と E<sup>2</sup>PROM のメモリ構成

#### ● システム構成

アドレス一致検出機能を利用したシステム構成例を図 21.4-2 に示します。

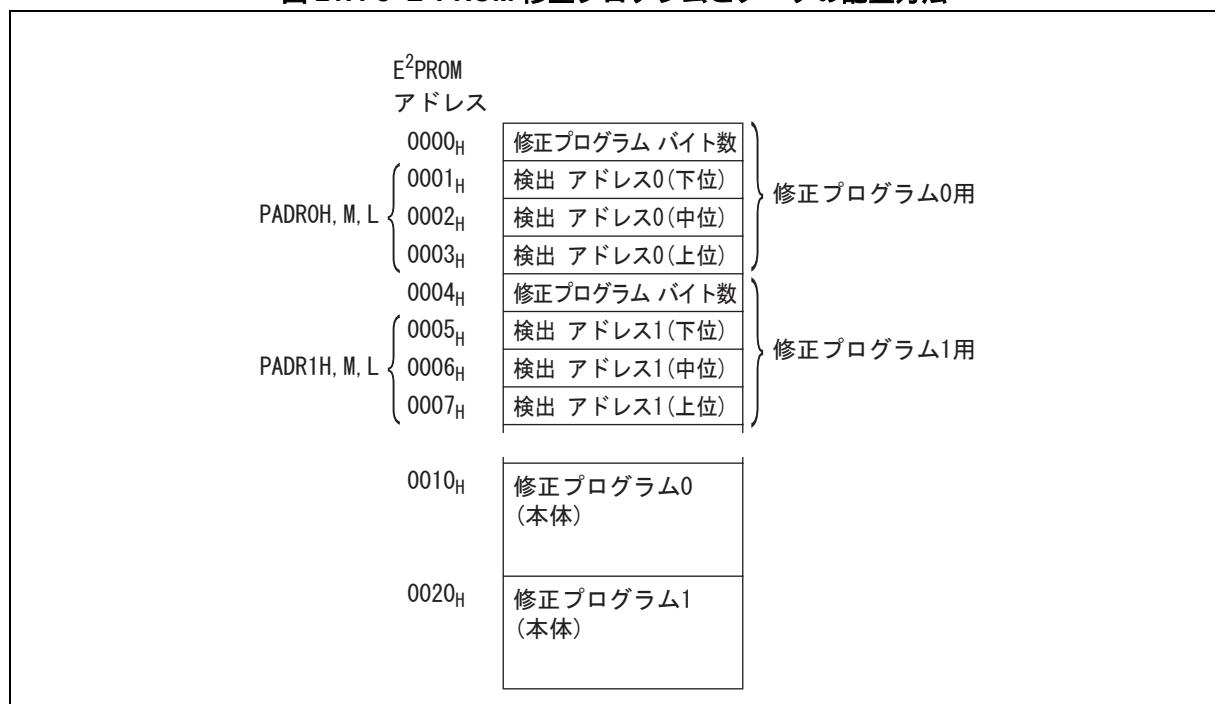
図 21.4-2 アドレス一致検出機能を利用したシステム構成例



### ■ E<sup>2</sup>PROM のメモリマップ

修正プログラムを E<sup>2</sup>PROM に格納する場合の，修正プログラムとデータの配置方法を図 21.4-3 に示します。

図 21.4-3 E<sup>2</sup>PROM 修正プログラムとデータの配置方法



- 修正プログラムバイト数

修正プログラム（本体）の総バイト数を格納します。バイト数が "00<sub>H</sub>" の場合は、"修正プログラムは存在しない"ことを示します。

- 検出アドレス（24 ビット）

プログラムミスにより INT9 命令に置換える箇所のアドレスを格納します。このアドレスはプログラムアドレス検出レジスタ（PADR0H, M, L, PADR1H, M, L）に設定するアドレスです。

- 修正プログラム（本体）

プログラムのアドレスと検出アドレスが一致した場合の、INT9 割込みによって実行するプログラムを格納します。修正プログラム 0 は、任意に決めたアドレスから配置します。修正プログラム 1 は、＜修正プログラム 0 の先頭アドレス + 修正プログラム 0 の総バイト数＞のアドレスから配置します。

## ■ 設定と動作状態

- 初期設定

- E<sup>2</sup>PROM のデータはすべて "00<sub>H</sub>" にクリアします。

- プログラムミスが発生した場合

- コネクタ接続（UART）を使用して、外部から MCU（MB90460/465）へ、E<sup>2</sup>PROM 修正プログラムとデータの配置方法に従って、修正プログラムの情報を送ります。
- MCU（MB90460/465）側では、外部から受取った修正プログラムの情報を E<sup>2</sup>PROM へ格納します。

- リセットシーケンス

- リセット後、MCU（MB90460/465）は、E<sup>2</sup>PROM 修正プログラムのバイト数を読み出して、修正プログラムの有無を確認します。
- 修正プログラムのバイト数が "00<sub>H</sub>" でない場合は、検出アドレス 0, 1 の上位、中位、下位を読み出してプログラムアドレス検出レジスタ 0, 1（PADR0H, M, L, PADR1H, M, L）へ読み出したデータを設定します。また、修正プログラムバイト数に従って、修正プログラム（本体）を読み出して、MCU（MB90460/465）の RAM へ書き込みます。
- 修正プログラム（本体）は、アドレス一致検出機能によって INT9 割込み処理で実行するアドレスに配置します。
- アドレス一致検出動作を許可します（PACSR レジスタの ADE0:bit1=1, ADE1:bit3=1）。

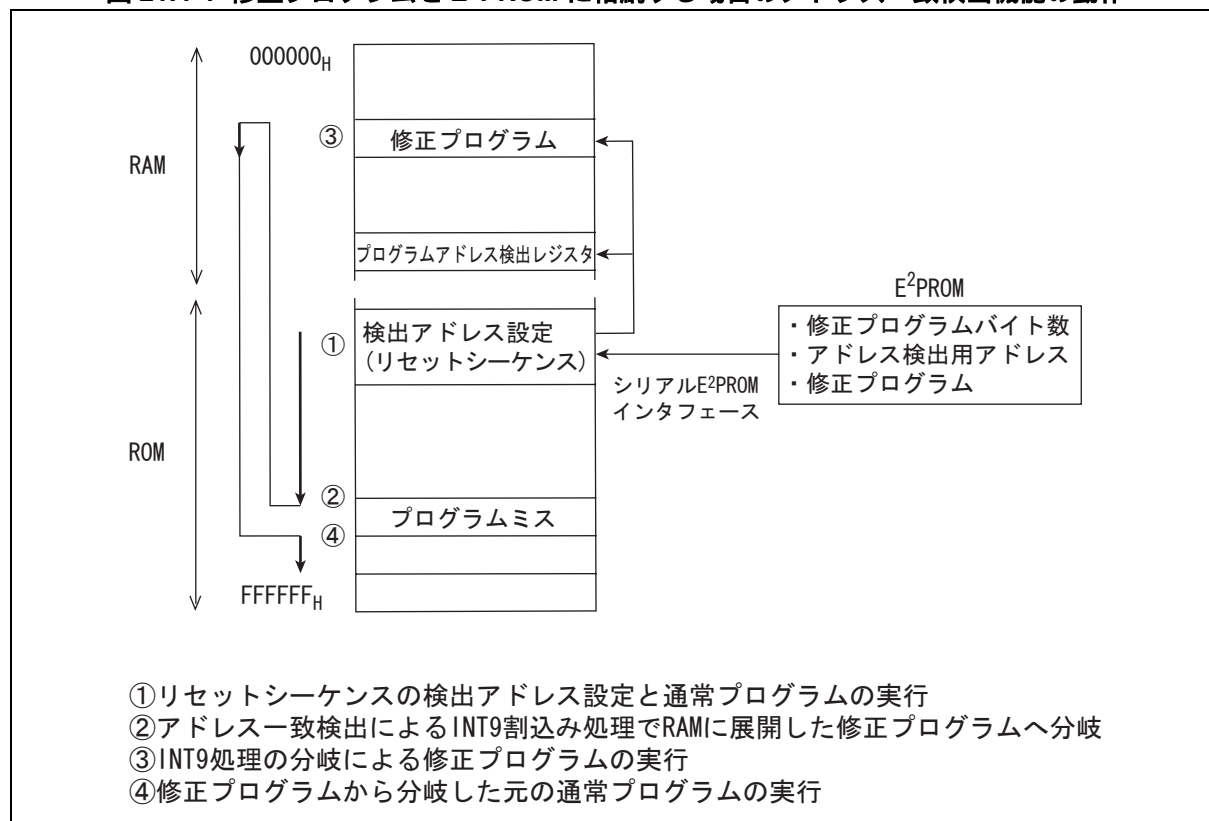
- INT9 割込み処理

- INT9 命令によって割込み処理が実行されます。MB90460/465 シリーズには、アドレス一致検出による割込み要求フラグがありません。したがって、プログラムカウンタのスタック情報が破棄されると検出アドレスを確認できません。検出アドレスを確認する場合は、割込み処理ルーチンでスタックされているプログラムカウンタの値を確認してください。
- 修正プログラムの実行後に通常のプログラムに分岐します。

## ■ 修正プログラムを E<sup>2</sup>PROM に格納する場合のアドレス一致検出機能の動作

修正プログラムを E<sup>2</sup>PROM に格納する場合のアドレス一致検出機能の動作を、図 21.4-4 に示します。

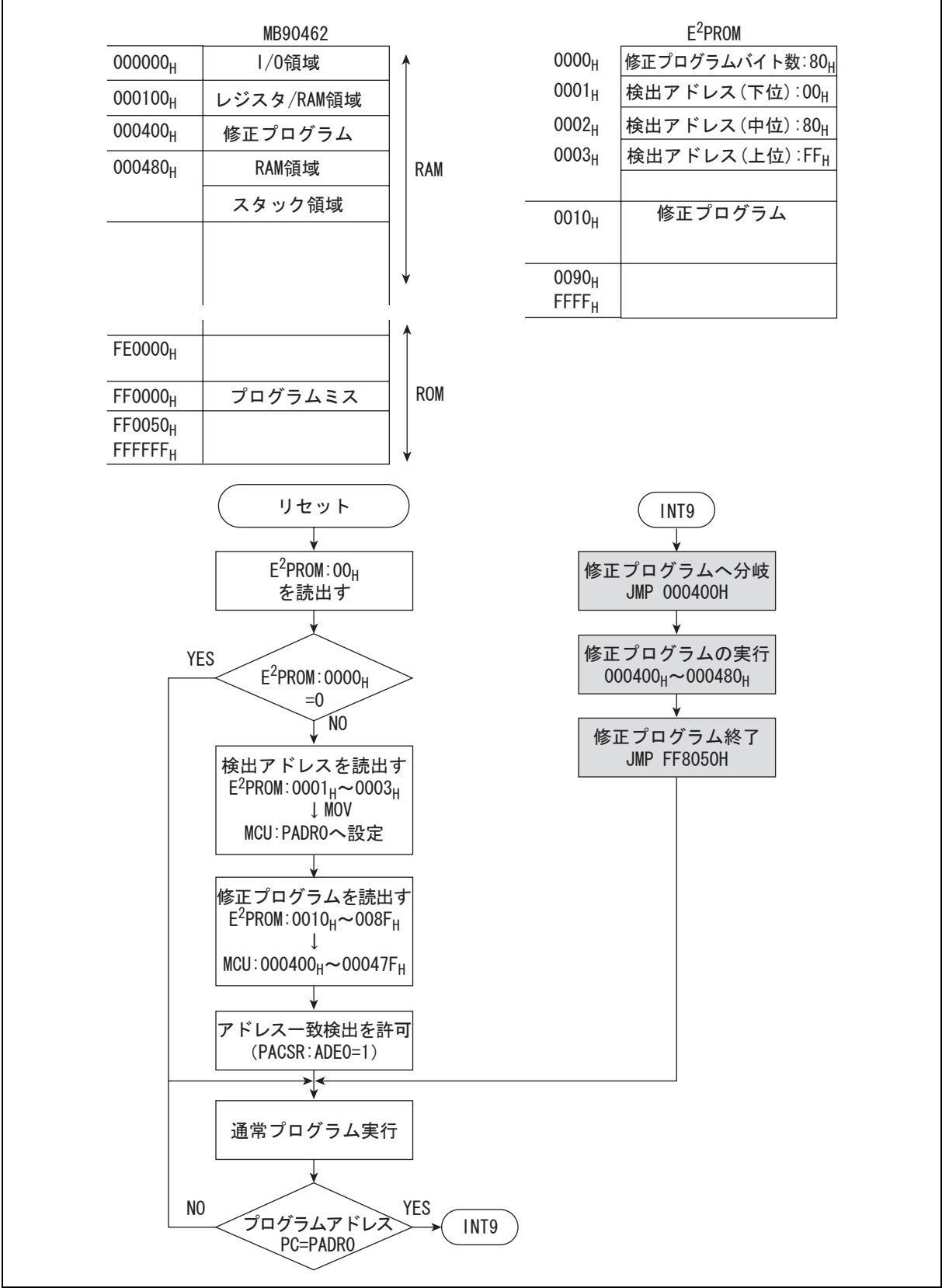
図 21.4-4 修正プログラムを E<sup>2</sup>PROM に格納する場合のアドレス一致検出機能の動作



■ プログラム修正のパッチ処理フロー

アドレス一致検出機能を利用したプログラム修正のパッチ処理のフローを図 21.4-5 に示します。

図 21.4-5 プログラム修正のパッチ処理フロー



## 21.5 アドレス一致検出機能のプログラム例

アドレス一致検出機能のプログラム例を示します。

### ■ アドレス一致検出機能のプログラム例

#### ● 処理仕様

- プログラムで実効される命令のアドレスと、プログラムアドレス検出レジスタ (PADR0H, M, L) に設定したアドレスが一致した場合に、INT9 命令が実行されます。

#### ● コーディング例

```

PACSR EQU 00009EH ; プログラムアドレス検出制御
; ステータスレジスタ

PADRL EQU 000001H ; プログラムアドレス検出レジスタ 0 下位
PADRM EQU 000002H ; プログラムアドレス検出レジスタ 0 中位
PADRH EQU 000003H ; プログラムアドレス検出レジスタ 0 上位
;
; ----- メインプログラム -----
CODE CSEG
START:
; スタックポインタ (SP) などは
; 初期化済みとする

MOV PADRL, #00H ; プログラムアドレス検出レジスタ 0 下位設定
MOV PADRM, #00H ; プログラムアドレス検出レジスタ 0 中位設定
MOV PADRH, #00H ; プログラムアドレス検出レジスタ 0 上位設定
;
MOV I: PACSR, #00000010B; アドレス一致検出の許可
.
ユーザ処理
.
LOOP:
.
ユーザ処理
.
BRA LOOP
; ----- 割込みプログラム -----
WARI:
.
ユーザ処理
.
RETI ; 割込み処理からの復帰
CODE ENDS
; ----- ベクタ設定 -----
VECT CSEG ABS=0FFH
ORG 00FFDCH ; リセットベクタ設定
DSL START
DB 00H ; シングルチップモードに設定
VECT ENDS
END START

```

## 第22章

---

# ROM 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の機能と動作について説明します。

22.1 ROM ミラー機能選択モジュールの概要

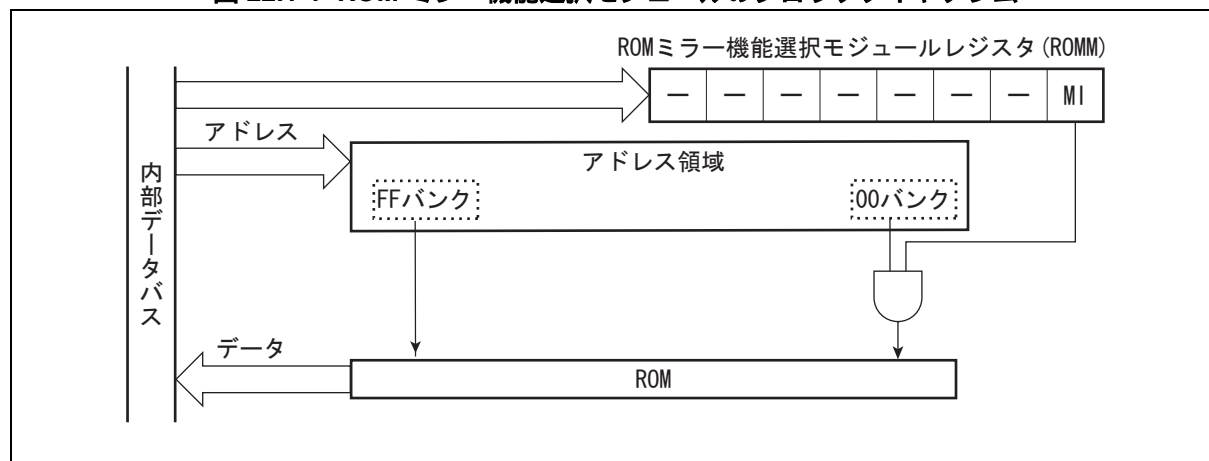
22.2 ROM ミラー機能選択モジュールレジスタ (ROMM)

## 22.1 ROM ミラー機能選択モジュールの概要

ROM ミラー機能選択モジュールにおいては、FF バンクに配置されている ROM 内のデータを、00 バンクへのアクセスによって読み出すことができます。

### ■ ROM ミラー機能選択モジュールのブロックダイアグラム

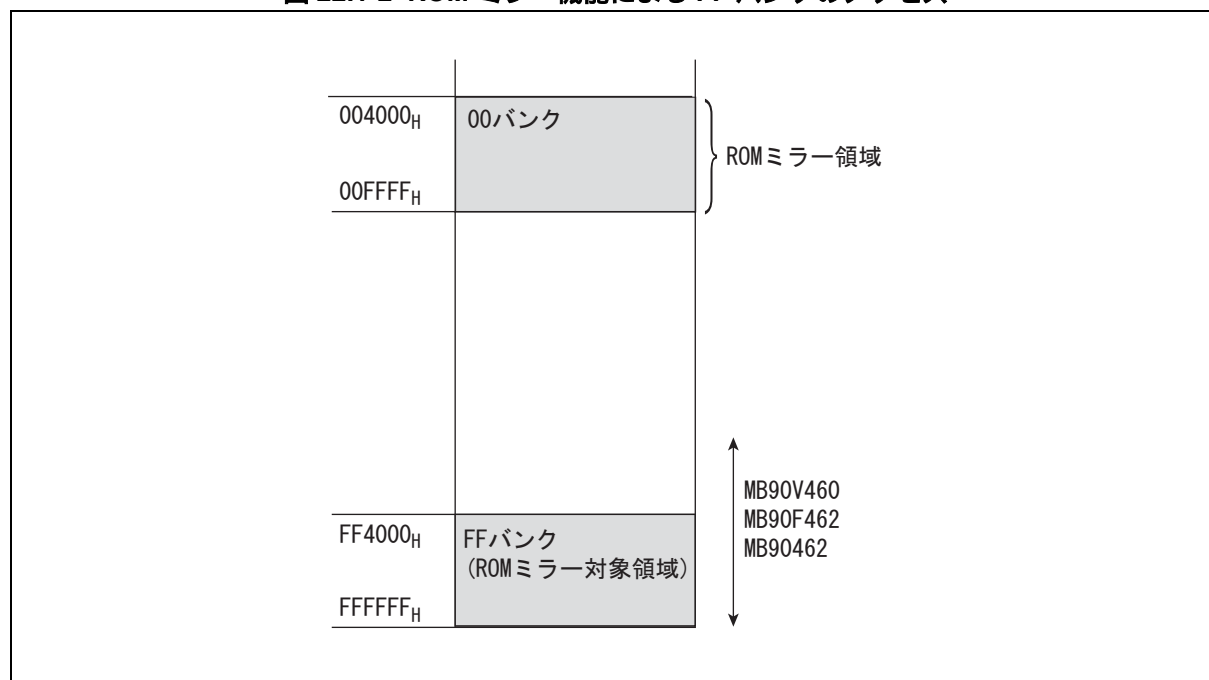
図 22.1-1 ROM ミラー機能選択モジュールのブロックダイアグラム



### ■ ROM ミラー機能による FF バンクへのアクセス

ROM ミラー機能によって、00 バンクへのアクセスにより、FF バンクの ROM データを読み出す場合の、メモリ上の位置関係を図 22.1-2 に示します。

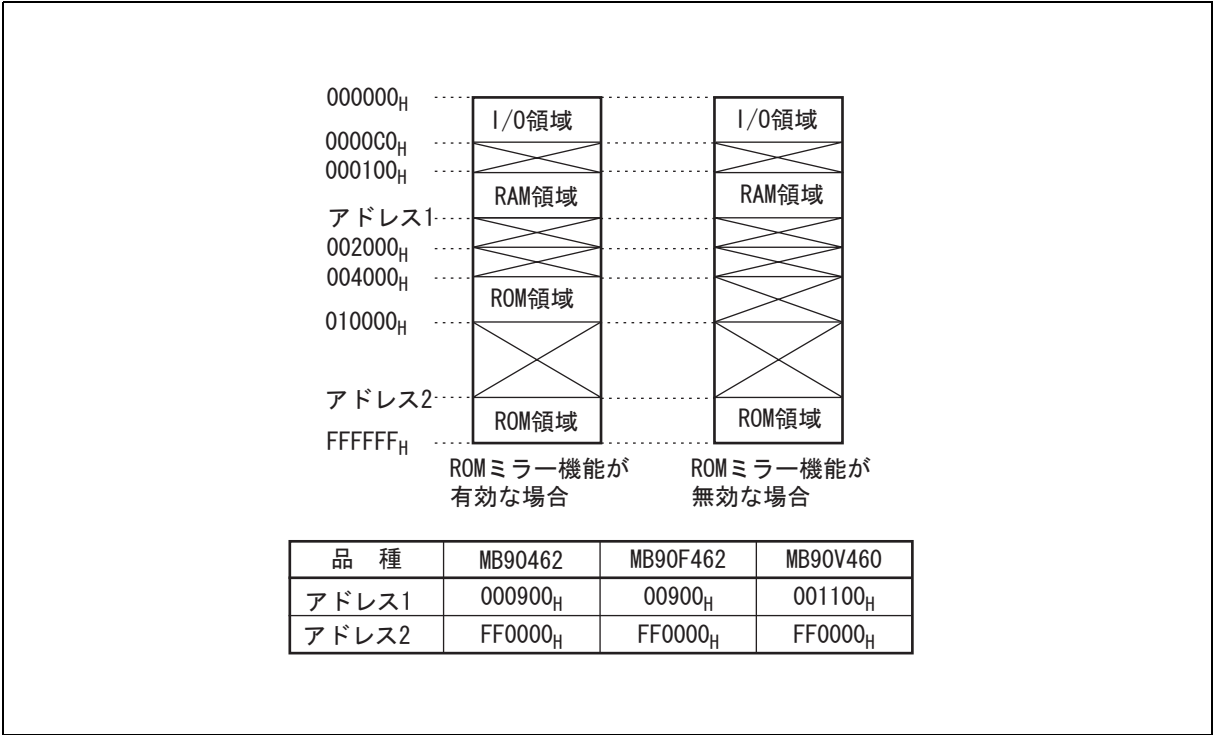
図 22.1-2 ROM ミラー機能による FF バンクのアクセス



■ ROM ミラー機能の有効 / 無効時のメモリ空間

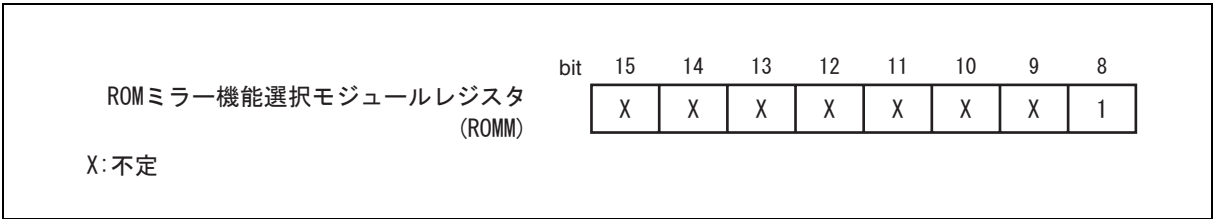
ROM ミラー機能を " 有効 " または " 無効 " にした場合のメモリ空間へのアクセス可否を図 22.1-3 に示します。

図 22.1-3 ROM ミラー機能の有効 / 無効時のメモリ空間（シングルチップモード時）



■ ROM ミラー機能選択モジュールのレジスタと初期値の一覧

図 22.1-4 ROM ミラー機能選択モジュールのレジスタと初期値の一覧





## 22.2 ROM ミラー機能選択モジュールレジスタ (ROMM)

ROM ミラー機能選択モジュールレジスタは, ROM ミラー機能を " 有効 " にするか, " 無効 " にするかを設定します。" 有効 " を設定した場合は, 00 バンクを読み出すことで, FF バンクの ROM データを読み出せます。

### ■ ROM ミラー機能選択モジュールレジスタ (ROMM)

図 22.2-1 ROM ミラー機能選択モジュールレジスタ (ROMM)

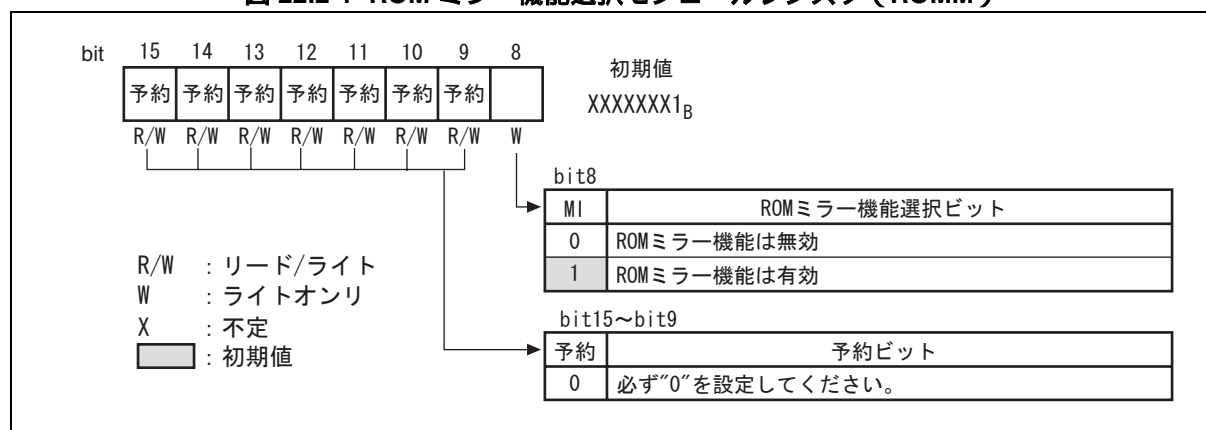


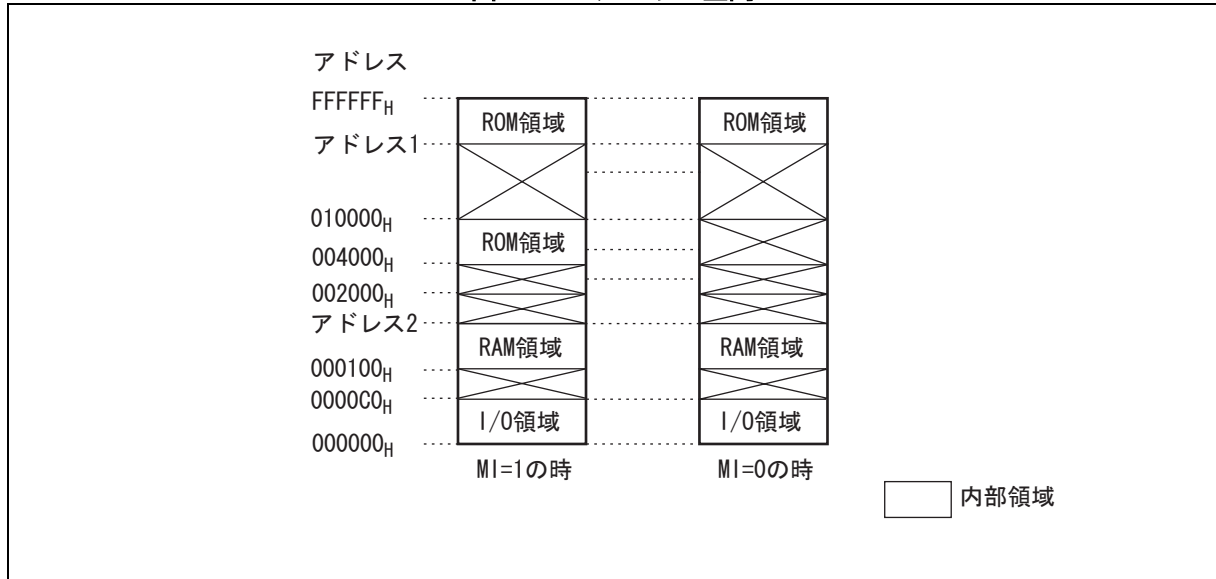
表 22.2-1 ROM ミラー機能選択モジュールレジスタ (ROMM) の機能

ビット名		機 能
bit15 ~ bit9	予約ビット	読出し値は不定です。 必ず "0" を設定してください。
bit8	MI: ROM ミラー機能選択ビット	ROM ミラー機能を " 有効 " にするか, " 無効 " にするかを設定します。 このビットに "0" を設定した場合:ROM ミラー機能を無効にします。 このビットに "1" を設定した場合:ROM ミラー機能を有効にします。 ROM ミラー機能を有効に設定した場合 (MI=1), アドレス 004000 <sub>H</sub> ~ 00FFFF <sub>H</sub> へアクセスすることにより, ROM アドレス FF4000 <sub>H</sub> ~ FFFFFFF <sub>H</sub> のデータを読み出せます。

< 注意事項 > バンク 00 は, 004000<sub>H</sub> ~ 00FFFF<sub>H</sub> 番地から FF4000<sub>H</sub> ~ FFFFFFF<sub>H</sub> 番地を参照しますので FF3FFF<sub>H</sub> 番地以下のアドレスは, ROM ミラー機能を設定しても参照できません。

	MB90462	MB90467	MB90F462	MB90F462A	MB90F463A	MB90V460
アドレス 1	FF0000 <sub>H</sub>	FF0000 <sub>H</sub>	FF0000 <sub>H</sub>	FF0000 <sub>H</sub>	FE0000 <sub>H</sub>	FF0000 <sub>H</sub>
アドレス 2	000900 <sub>H</sub>	000900 <sub>H</sub>	000900 <sub>H</sub>	000900 <sub>H</sub>	000900 <sub>H</sub>	002100 <sub>H</sub>

図 22.2-2 メモリー空間





## 第23章

---

# 512K/1024K ビット フラッシュメモリ

512K/1024K ビットフラッシュメモリの機能と動作について説明します。

- 23.1 512K/1024K ビットフラッシュメモリの概要
- 23.2 フラッシュメモリのレジスタとセクタ構成
- 23.3 フラッシュメモリ制御ステータスレジスタ (FMCS)
- 23.4 フラッシュメモリ自動アルゴリズム起動方法
- 23.5 自動アルゴリズム実行状態の確認
- 23.6 フラッシュメモリ書込み / 消去の詳細説明
- 23.7 フラッシュセキュリティの特長
- 23.8 512K ビットフラッシュメモリのプログラム例

## 23.1 512K/1024K ビットフラッシュメモリの概要

---

フラッシュメモリへのデータ書込み / 消去の方法としては、以下の 3 とおりがあります。

1. パラレルライタ
2. シリアル専用ライタ
3. プログラム実行による書込み / 消去

本章では、「3. プログラム実行による書込み / 消去」について解説します。

---

### ■ 512K/1024K ビットフラッシュメモリの概要

512K ビットフラッシュメモリは、CPU メモリマップ上の FF<sub>H</sub> バンクに配置されています。1024K ビットフラッシュメモリは、CPU メモリマップ上の FE<sub>H</sub> バンクと FF<sub>H</sub> バンクに配置されています。フラッシュメモリインターフェース回路の機能により、CPU からのリードアクセスおよびプログラムアクセスができます。

フラッシュメモリへの書込み / 消去は、フラッシュメモリインターフェース回路を介して CPU からの命令で行えるため、実装状態での書換えができ、プログラムやデータの改善を効率よく行うことができます。

### ■ 512K/1024K ビットフラッシュメモリの特長

- 512K ビット : 64K ワード × 8 ビット / 32K ワード × 16 ビット (16K+8K+8K+32K) セクタ構成
- 1024K ビット : 128K ワード × 8 ビット / 64K ワード × 16 ビット (64K+16K+8K+8K+32K) セクタ構成
- 自動プログラムアルゴリズム (Embedded Algorithm : MBM29LV200 と同様)
- 消去一時停止 / 消去再開機能の搭載
- データポーリング、トグルビットによる書込み / 消去完了検出
- CPU 割込みによる書込み / 消去の完了検出
- セクタごとの消去が可能 (セクタ組合わせ自由)
- 書込み / 消去回数 (最小) : 10,000 回
- フラッシュ読出しサイクルタイム (最小) : 2 マシンサイクル

---

< 注意事項 > マニファクチャコードとデバイスコードの読出し機能はありません。  
また、これらのコードは、コマンドによってもアクセスできません。

---

### ■ フラッシュメモリ書込み / 消去

- フラッシュメモリは、書込みと読出しを同時に行うことはできません。
- フラッシュメモリにおいてデータの書込み / 消去する場合は、フラッシュメモリ上にあるプログラムを一度 RAM にコピーし、RAM にコピーしたプログラムを実行することにより、フラッシュメモリへ書き込みます。

## 23.2 フラッシュメモリのレジスタとセクタ構成

フラッシュメモリのセクタ構成を示します。

### ■ 512K/1024K ビットフラッシュメモリのセクタ構成

512K ビットフラッシュメモリのセクタ構成を図 23.2-1 に、1024K ビットフラッシュメモリのセクタ構成を図 23.2-2 に示します。図中アドレスには、各セクタの上位アドレスと下位アドレスを示します。

図 23.2-1 512K ビットフラッシュメモリのセクタ構成

フラッシュメモリ	CPUアドレス	ライターアドレス *
SA3 (16Kバイト)	FFC000 <sub>H</sub>	7C000 <sub>H</sub>
	FFFFFF <sub>H</sub>	7FFFF <sub>H</sub>
SA2 (8Kバイト)	FFA000 <sub>H</sub>	7A000 <sub>H</sub>
	FFBFFF <sub>H</sub>	7BFFF <sub>H</sub>
SA1 (8Kバイト)	FF8000 <sub>H</sub>	78000 <sub>H</sub>
	FF9FFF <sub>H</sub>	79FFF <sub>H</sub>
SA0 (32Kバイト)	FF0000 <sub>H</sub>	70000 <sub>H</sub>
	FF7FFF <sub>H</sub>	77FFF <sub>H</sub>

\* :ライターアドレスとは、フラッシュメモリにパラレルライターによりデータを書込む場合、CPUアドレスに相当するアドレスです。  
汎用ライターを使用し書込み/消去する場合は、このライターアドレスで書込み/消去します。

図 23.2-2 1024K ビットフラッシュメモリのセクタ構成

フラッシュメモリ	CPUアドレス	ライターアドレス *
SA4 (16Kバイト)	FFFFFF <sub>H</sub>	7FFFF <sub>H</sub>
	FFC000 <sub>H</sub>	7C000 <sub>H</sub>
SA3 (8Kバイト)	FFBFFF <sub>H</sub>	7BFFF <sub>H</sub>
	FFA000 <sub>H</sub>	7A000 <sub>H</sub>
SA2 (8Kバイト)	FF9FFF <sub>H</sub>	79FFF <sub>H</sub>
	FF8000 <sub>H</sub>	78000 <sub>H</sub>
SA1 (32Kバイト)	FF7FFF <sub>H</sub>	77FFF <sub>H</sub>
	FF0000 <sub>H</sub>	70000 <sub>H</sub>
SA0 (64Kバイト)	FEFFFF <sub>H</sub>	6FFFF <sub>H</sub>
	FE0000 <sub>H</sub>	60000 <sub>H</sub>

\* :ライターアドレスとは、フラッシュメモリにパラレルライターによりデータを書込む場合、CPUアドレスに相当するアドレスです。  
汎用ライターを使用し書込み/消去する場合は、このライターアドレスで書込み/消去します。

## 23.3 フラッシュメモリ制御ステータスレジスタ (FMCS)

フラッシュメモリ制御ステータスレジスタ (FMCS) の機能をに示します。

### ■ フラッシュメモリ制御ステータスレジスタ (FMCS)

図 23.3-1 フラッシュメモリ制御ステータスレジスタ (FMCS)

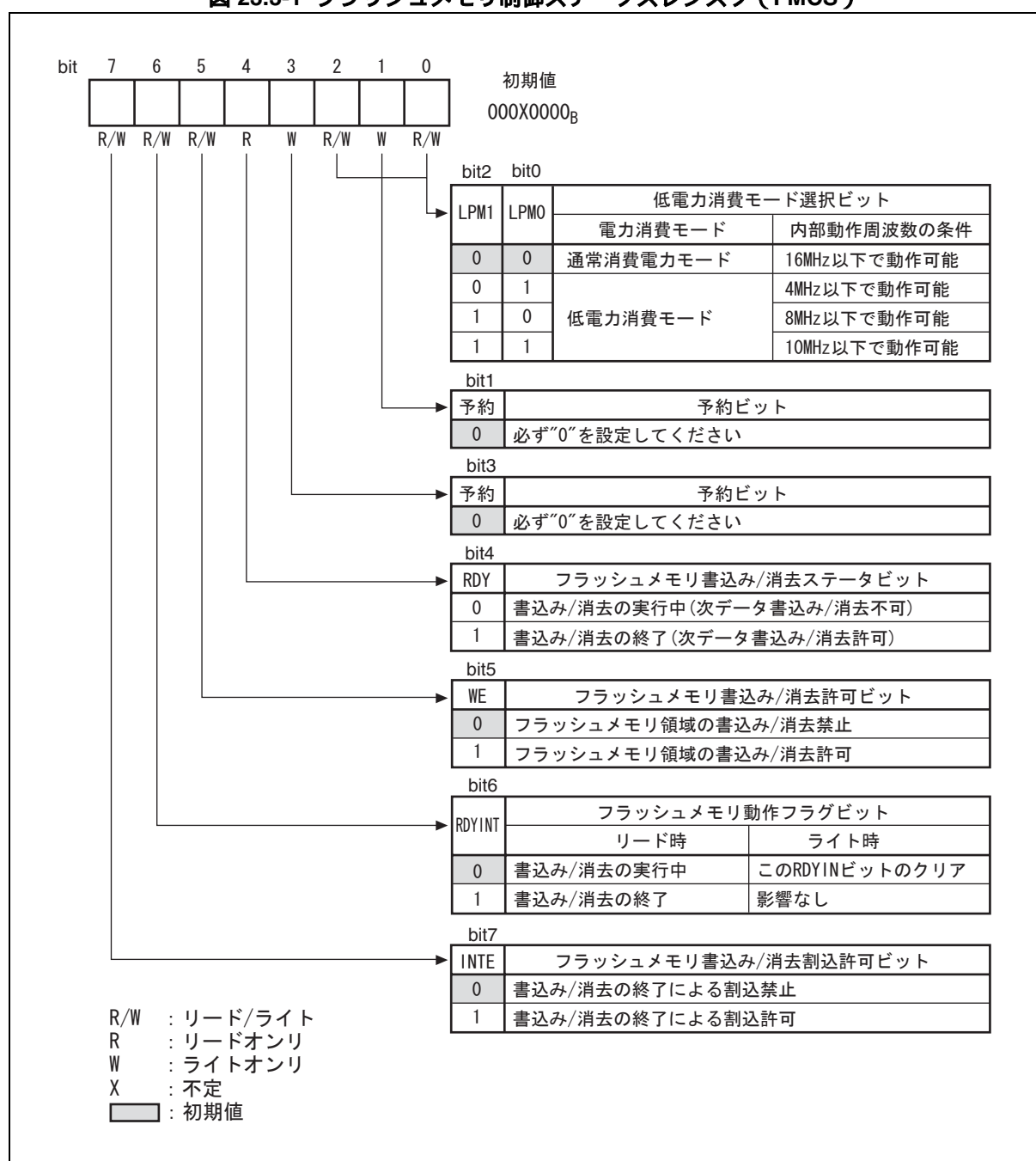
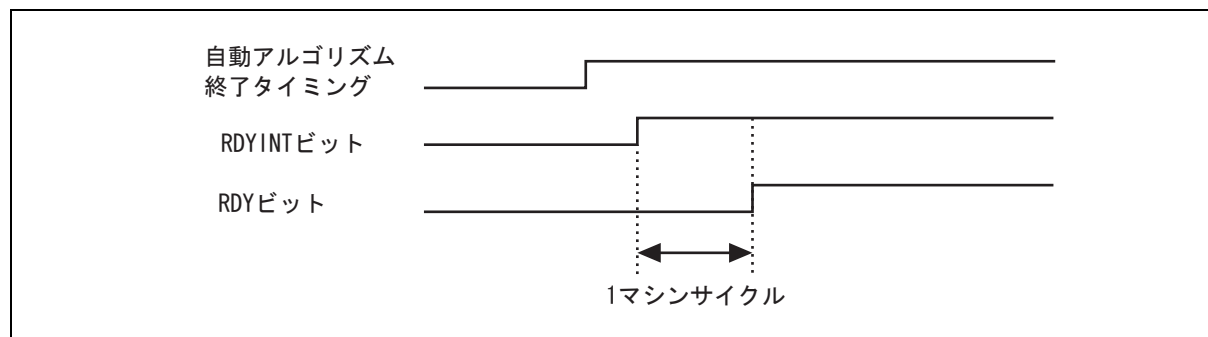


表 23.3-1 フラッシュメモリ制御ステータスレジスタ (FMCS) の機能

ビット名		機 能
bit7	INTE: フラッシュメモリ割込み / 消去割込み許可ビット	フラッシュメモリの書込み / 消去の終了による割込み要求発生を許可または禁止します。 <b>このビットに "1" に設定した場合:</b> フラッシュメモリ動作フラグビットに "1" を設定した場合に (FMCS: RDYINT=1), 割込み要求が発生します。
bit6	RDYINT: フラッシュメモリ動作フラグビット	フラッシュメモリの動作状態を示します。 フラッシュメモリの書込み / 消去が終了した場合に, フラッシュメモリ自動アルゴリズム終了のタイミングで, この RDYINT ビットに "1" が設定されます。 <ul style="list-style-type: none"> <li>フラッシュメモリ書込み / 消去の終了による割込みを許可に設定している場合は (FMCS: INTE=1), この RDYINT ビットに "1" が設定されると割込み要求が発生します。</li> <li>この RDYINT ビットが "0" の場合は, フラッシュメモリへの書込み / 消去はできません。</li> </ul> <b>このビットに "0" に設定した場合:</b> クリアされます。 <b>このビットに "1" に設定した場合:</b> 影響しません。 リードモディファイライト (RMW) 命令を使用した場合は, 必ず "1" が読み出されます。
bit5	WE: フラッシュメモリ書込み / 消去許可ビット	フラッシュメモリ領域への書込み / 消去を許可または禁止します。 この WE ビットは, フラッシュメモリの書込み / 消去のコマンドを起動する前に設定してください。 <b>このビットに "0" に設定した場合:</b> FF バンクへの書込み / 消去コマンドを入力しても, 書込み / 消去の信号は発生しません。 <b>このビットに "1" に設定した場合:</b> FF バンクへの書込み / 消去コマンド入力後, フラッシュメモリへの書込み / 消去ができます。 <ul style="list-style-type: none"> <li>書込み / 消去しない場合は, 誤ってフラッシュメモリに書き込んだり, 消去しないように, この WE ビットに "0" を設定してください。</li> </ul>
bit4	RDY: フラッシュメモリ書込み / 消去ステータスビット	フラッシュメモリの書込み / 消去の状態を示します。 <ul style="list-style-type: none"> <li>この RDY ビットが "0" の場合は, フラッシュメモリへの書込み / 消去はできません。</li> <li>この RDY ビットが "0" の場合でも, 読出し / リセットコマンド, セクタ消去一時停止などのサスペンドコマンドは受付けることができます。書込み / 消去動作を終了すると, この RDY ビットには "1" が設定されます。</li> </ul>
bit3, bit1	予約: 予約ビット	必ず "0" を設定してください。
bit2, bit0	LPM1, LPM0: 低電力消費モード選択ビット	フラッシュメモリ本体の消費電流を制御します。 <ul style="list-style-type: none"> <li>フラッシュメモリへのアクセスタイムが動作周波数によって大きく異なるため, CPU の内部動作周波数の条件を参照して設定してください。</li> </ul>



< 注意事項 > フラッシュメモリ動作フラグビット (RDYINT) とフラッシュメモリ書込み / 消去ステータスビット (RDY) は同時には変化しません。どちらかのビットで書込み / 消去の終了を判定するようにプログラムを作成してください。



## 23.4 フラッシュメモリ自動アルゴリズム起動方法

フラッシュメモリの自動アルゴリズムを起動するコマンドには、読出し / リセット、書込みおよびチップ消去の 4 種類があり、セクタ消去については一時停止と再開の制御ができます。

### ■ コマンドシーケンス表

表 23.4-1 に、フラッシュメモリの書込み / 消去時に使用するコマンドの一覧を示します。コマンドレジスタにライトするデータはすべてバイト長ですが、ワードアクセスで書き込んでください。ワードアクセスで書き込んだ場合の上位バイトのデータは無視されます。

表 23.4-1 コマンドシーケンス表

コマンド シーケンス	バス ライト アクセス	1stバス ライトサイクル		2ndバス ライトサイクル		3rdバス ライトサイクル		4thバス ライトサイクル		5thバス ライトサイクル		6thバス ライトサイクル	
		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	FFXXXX	XXF0	—	—	—	—	—	—	—	—	—	—
読出し/ リセット*	4	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XXF0	RA	RD	—	—	—	—
書込み プログラム	4	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XXA0	PA (even)	PD (word)	—	—	—	—
チップ 消去	6	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX80	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX10
セクタ 消去	6	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX80	FFAAAA	XXAA	FF5554	XX55	SA (even)	XX30
セクタ消去一時停止		Address"FFXXXX"Data(xxB0 <sub>H</sub> )の入力で、セクタ消去中の消去一時停止											
セクタ消去再開		Address"FFXXXX"Data(xx30 <sub>H</sub> )の入力で、セクタ消去中の消去一時停止後、消去開始											
Auto Select	3	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX90	—	—	—	—	—	—

\*: 2種類の読出し/リセットコマンドは、フラッシュメモリを読出しモードにリセットできます。

(注意事項) ・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進数で表記しています。ただし"X"は任意の値です。

- ・RA: 読出しアドレス
- ・PA: 書込みアドレス、偶数アドレスのみ指定可
- ・SA: セクタアドレス
- ・RD: 読出しデータ
- ・PD: 書込みデータ、ワードデータのみ指定可

表 23.4-1 における AutoSelect は、セクタ保護の状態を知るためのコマンドです。

表 23.4-1 のコマンドと共に、下記のようにアドレスを設定する必要があります。

表 23.4-2 AutoSelect 時のアドレス設定

	AQ13 ~ AQ16	AQ7	AQ2	AQ1	AQ0	DQ7 ~ DQ0
セクタ保護	セクタ アドレス	L	H	L	L	CODE <sup>*</sup>

\*: 保護されたセクタアドレスでの出力は "01<sub>H</sub>"

保護されていないセクタアドレスでの出力は "00<sub>H</sub>"

## 23.5 自動アルゴリズム実行状態の確認

フラッシュメモリでは、書込み / 消去のフローを自動アルゴリズムで実行するため、フラッシュメモリ内部の動作状態をハードウェアシーケンスによって確認ができます。

### ■ ハードウェアシーケンスフラグ

#### ● ハードウェアシーケンスフラグの概要

ハードウェアシーケンスフラグは、以下の 5 ビットの出力で構成されます。

- 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)
- セクタ消去タイマフラグ (DQ3)
- トグルビット 2 フラグ (DQ2)

ハードウェアシーケンスフラグにより、書込み / チップ・セクタ消去の終了、消去コード書込みができるかを確認できます。

ハードウェアシーケンスフラグを参照するには、コマンドシーケンス設定後にフラッシュメモリ内部の対象セクタのアドレスにリードアクセスすることで参照できます。表 23.5-1 にハードウェアシーケンスフラグのビット割当てを示します。

表 23.5-1 ハードウェアシーケンスフラグのビット割当て

ビット No.	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	-	DQ3	DQ2	-	-

- 自動書込み / チップ・セクタ消去が実行中か終了しているか判断するためには、ハードウェアシーケンスフラグを確認するかまたはフラッシュメモリ制御ステータスレジスタのフラッシュメモリ書込み / 消去ステータスビット (FMCS: RDY) を確認してください。書込み / 消去の終了後は、読出し / リセット状態に戻ります。
- 書込み / 消去プログラムを作成する場合には、DQ7, DQ6, DQ5, DQ3, DQ2 のフラグで自動書込み / 消去の終了を確認後に、データの読出しを処理してください。
- 2 回目以降のセクタ消去コード書込みが有効であるかについても、ハードウェアシーケンスフラグによって確認できます。

## ● ハードウェアシーケンスフラグの説明

表 23.5-2 に、ハードウェアシーケンスフラグ機能の一覧を示します。

表 23.5-2 ハードウェアシーケンスフラグ機能の一覧

状 態		DQ7	DQ6	DQ5	DQ3	DQ2
正常動作時の状態変化	書込み 書込み完了 (書込みアドレス指時)	$\overline{\text{DQ7}}$ DATA: 7	Toggle DATA: 6	0 DATA: 5	0 DATA: 3	1 DATA: 2
	チップ・セクタ消去 消去完了	0 1	Toggle Stop	0 1	1	Toggle Stop
	セクタ消去ウェイト 消去開始	0	Toggle	0	0 1	Toggle
	消去 セクタ消去一時止 (消去中のセクタ)	0 1	Toggle 1	0	1 0	Toggle
	セクタ消去一時停止 消去再開 (消去中のセクタ)	1 0	1 Toggle	0	0 1	Toggle
	セクタ消去一時停止中 (消去中ではないセクタ)	DATA: 7	DATA: 6	DATA: 5	DATA: 3	DATA: 2
異常動作	書込み	$\overline{\text{DQ7}}$	Toggle	1	0	1
	チップ・セクタ消去	0	Toggle	1	1	*

\* : DQ5 が "1" の場合 (タイミングリミット超過), 書込み / 消去中セクタへの連続的な読出しに対して DQ2 はトグル動作を行い, ほかのセクタへの読出しに対してはトグル動作を行いません。

## 23.5.1 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、自動アルゴリズムが実行中もしくは終了状態であることをデータポーリング機能によって知らせるハードウェアシーケンスフラグです。

### ■ データポーリングフラグ (DQ7)

表 23.5-3 と表 23.5-4 に、データポーリングフラグの状態遷移を示します。

表 23.5-3 データポーリングフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み 書込み完了	チップ・ セクタ消去 消去完了	セクタ消去 ウェイト 消去開始	セクタ消去 セクタ消去 一時停止 (消去中の セクタ)	セクタ消去 一時停止 消去再開 (消去中の セクタ)	セクタ消去 一時停止中 (消去中では ないセクタ)
DQ7	$\overline{\text{DQ7}}$ DATA: 7	0 1	0	0 1	1 0	DATA: 7

表 23.5-4 データポーリングフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み	チップ・セクタ 消去
DQ7	$\overline{\text{DQ7}}$	0

#### ● 書込みの場合

- 自動書込みアルゴリズム実行中にリードアクセスした場合、最後に書き込まれたデータの bit7 を反転させた値を出力します。
- 自動書込みアルゴリズム終了時にリードアクセスした場合、フラッシュメモリはリードアクセスしたアドレスの bit7 を出力します。

#### ● チップ消去 / セクタ消去の場合

- チップ消去 / セクタ消去のアルゴリズム実行中に現在消去しているセクタへリードアクセスすると、フラッシュメモリの bit7 は "0" を出力します。チップ消去 / セクタ消去が終了すると、フラッシュメモリの bit7 は "1" を出力します。

#### ● セクタ消去一時停止の場合

- セクタ消去一時停止状態でリードアクセスした場合、フラッシュメモリは、リードアドレスが消去中のセクタであれば、"1" を出力します。消去中のセクタでなければ、リードアドレスの読出し値の bit7 (DATA: 7) を出力します。
- トグルビットフラグ (DQ6) と共に参照することで、セクター一時停止状態であるか、どのセクタが消去中であるかを判定できます。

---

< 注意事項 > 自動アルゴリズムを起動した場合は、指定したアドレスへのリードアクセスは無視されます。データの読出しは、データポーリングフラグ（DQ7）に "1" が設定された後、可能になります。自動アルゴリズム終了後のデータの読出しは、データポーリング終了を確認したリードアクセスの後に行ってください。

---

## 23.5.2 トグルビットフラグ (DQ6)

トグルビットフラグ (DQ6) は、自動アルゴリズムが実行中、もしくは終了状態であることをトグルビット機能によって知らせるハードウェアシーケンスフラグです。

### ■ トグルビットフラグ (DQ6)

表 23.5-5 と表 23.5-6 に、トグルビットフラグの状態遷移を示します。

表 23.5-5 トグルビットフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み 書込み完了	チップ・ セクタ消去 消去完了	セクタ消去 ウェイト 消去開始	セクタ消去 セクタ消去 一時停止 (消去中の セクタ)	セクタ消去 一時停止 消去再開 (消去中の セクタ)	セクタ消去 一時停止中 (消去中では ないセクタ)
DQ6	Toggle DATA: 6	Toggle Stop	Toggle	Toggle 1	1 Toggle	DATA: 6

表 23.5-6 トグルビットフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み	チップ・セクタ 消去
DQ6	Toggle	Toggle

#### ● 書込みとチップ消去 / セクタ消去の場合

- 自動書込みアルゴリズムおよびチップ消去 / セクタ消去のアルゴリズム実行中に、リードアクセスを連続して行った場合、フラッシュメモリは、リードごとに "1" と "0" を交互にトグル出力します。
- 自動書込みアルゴリズムおよびチップ消去 / セクタ消去のアルゴリズム終了後に、リードアクセスを連続して行った場合、フラッシュメモリはリードごとにリードアドレスの読出し値の bit6 (DATA: 6) を出力します。

#### ● セクタ消去一時停止の場合

- セクタ消去一時停止状態でリードアクセスした場合、フラッシュメモリは、リードアドレスが消去中のセクタであるならば、"1" を出力します。消去中のセクタでなければ、リードアドレスの読出し値の bit6 (DATA: 6) を出力します。

< 参考 > 書込み対象のセクタが書換え保護されている場合は、約 2ms 間トグル出力した後、データを書換えることなくトグル出力を終了します。  
消去するすべてのセクタが書換え保護されている場合、トグルビットフラグ (DQ6) は、約 100ms の間トグル出力した後、データを書換えることなく読出し / リセット状態に戻ります。



### 23.5.3 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規定時間 (書き込み / 消去に要する時間) を超えてしまったことをフラッシュメモリ内部に知らせるハードウェアシーケンスフラグです。

#### ■ タイミングリミット超過フラグ (DQ5)

表 23.5-7 と表 23.5-8 に、タイミングリミット超過フラグの状態遷移を示します。

表 23.5-7 タイミングリミット超過フラグの状態遷移 (正常動作時の状態変化)

動作状態	書き込み 書き込み完了	チップ・ セクタ消去 消去完了	セクタ消去 ウェイト 消去開始	セクタ消去 セクタ消去 一時停止 (消去中の セクタ)	セクタ消去 一時停止 消去再開 (消去中の セクタ)	セクタ消去 一時停止中 (消去中では ないセクタ)
DQ5	0 DATA: 5	0 1	0	0	0	DATA: 5

表 23.5-8 タイミングリミット超過フラグの状態遷移 (異常動作時の状態変化)

動作状態	書き込み	チップ・セクタ 消去
DQ5	1	1

#### ● 書き込みとチップ消去 / セクタ消去の場合

- 書き込みまたはチップ消去 / セクタ消去の自動アルゴリズム起動後にリードアクセスをした場合、規定時間 (書き込み / 消去に要する時間) 内であれば "0" を、規定時間を超えている場合は "1" を出力します。
- タイミングリミット超過フラグ (DQ5) は、自動アルゴリズムが実行中か終了状態にかかわらず、書き込み / 消去の成功または失敗を判定できます。タイミングリミット超過フラグ (DQ5) が "1" を出力した場合、データポーリング機能またはトグルビット機能により自動アルゴリズムが実行中であれば、書き込みが失敗していると判断できます。
- 例えば "0" が書き込まれているフラッシュメモリアドレスに "1" を設定した場合は、フラッシュメモリはロックされ、自動アルゴリズムは終了せず、データポーリングフラグ (DQ7) から有効なデータが出力されません。またトグルビットフラグ (DQ6) はトグル動作を終了せず、タイムリミットを超えてしまい、タイミングリミット超過フラグ (DQ5) は "1" を出力します。タイミングリミット超過フラグ (DQ5) が "1" を出力した場合は、フラッシュメモリが不良ではなく、正しく使用されなかったことを示していますので、リセットコマンドを実行してください。

## 23.5.4 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ウェイト期間中であるか否かを知らせるハードウェアシーケンスフラグです。

### ■ セクタ消去タイマフラグ (DQ3)

表 23.5-9 と表 23.5-10 に、セクタ消去タイマフラグの状態遷移を示します。

表 23.5-9 セクタ消去タイマフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み 書込み完了	チップ・ セクタ消去 消去完了	セクタ消去 ウェイト 消去開始	セクタ消去 セクタ消去 一時停止 (消去中の セクタ)	セクタ消去 一時停止 消去再開 (消去中の セクタ)	セクタ消去 一時停止中 (消去中では ないセクタ)
DQ3	0 DATA: 3	1	0 1	1 0	0 1	DATA: 3

表 23.5-10 セクタ消去タイマフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み	チップ・セクタ 消去
DQ3	0	1

#### ● セクタ消去の場合

- セクタ消去コマンド起動後にリードアクセスした場合に、セクタ消去ウェイト期間中であれば "0" を出力します。セクタ消去ウェイト期間を超えている場合は "1" を出力します。
- データポーリング機能やトグルビット機能により消去アルゴリズムが実行中を示している場合 (DQ=0, DQ6 がトグル出力)、セクタ消去タイマフラグ (DQ3) が "1" であれば、セクタ消去しています。続けて消去一時停止以外のコマンドを設定した場合は、消去が終了されるまで無視されます。
- セクタ消去タイマフラグ (DQ3) が "0" であった場合、フラッシュメモリはセクタ消去コマンドを受け付けることができます。セクタ消去コマンドを書き込む場合は、セクタ消去タイマフラグ (DQ3) が "0" であることを確認してください。セクタ消去タイマ (DQ3) が "1" であった場合、一時停止のセクタ消去コマンドが受け付けられない場合があります。

#### ● セクタ消去一時停止の場合

- セクタ消去一時停止状態でリードアクセスした場合、フラッシュメモリは、リードアドレスが消去中のセクタであるならば、"1" を出力します。消去中のセクタでなければ、リードアドレスの読出し値の bit3 (DATA: 3) を出力します。

## 23.5.5 トグルビット 2 フラグ (DQ2)

トグルビット 2 フラグ (DQ2) は、セクタ消去一時停止中であることをトグルビット機能によって知らせるハードウェアシーケンスフラグです。

### ■ トグルビット 2 フラグ (DQ2)

表 23.5-11 と表 23.5-12 に、トグルビット 2 フラグの状態遷移を示します。

表 23.5-11 トグルビット 2 フラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み 書込み完了	チップ・ セクタ消去 消去完了	セクタ消去 ウェイト 消去開始	セクタ消去 セクタ消去 一時停止 (消去中の セクタ)	セクタ消去 一時停止 消去再開 (消去中の セクタ)	セクタ消去 一時停止中 (消去中では ないセクタ)
DQ2	1 DATA:2	Toggle Stop	Toggle	Toggle	Toggle	DATA: 2

表 23.5-12 トグルビット 2 フラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み	チップ・セクタ 消去
DQ2	1	*

\* : DQ5 が "1" の場合 (タイミングリミット超過), 書込み / 消去セクタ中への連続的な読出しに対して DQ2 はトグル動作を行い, ほかのセクタへの読出しに対してはトグル動作を行いません。

#### ● セクタ消去の場合

- チップ消去 / セクタ消去のアルゴリズム実行中に、リードアクセスを連続して行った場合、フラッシュメモリは、リードごとに "1" と "0" を交互にトグル出力します。
- チップ消去 / セクタ消去のアルゴリズムが終了したあとに、リードアクセスを連続して行った場合、フラッシュメモリは、リードごとにリードアドレスの読出し値の bit2 (DATA: 2) を出力します。

● セクタ消去一時停止の場合

- セクタ消去一時停止状態でリードアクセスを連続して行った場合，フラッシュメモリは，リードアドレスが消去中のセクタであるならば，"1" と "0" を交互にトグル出力します。消去中のセクタでなければ，リードアドレスの読出し値の bit2 (DATA: 2) を出力します。
- セクタ消去一時停止状態で書込みする場合，消去一時停止していないセクタからリードアクセスを連続して行くと，"1" を出力します。
- トグルビット 2 フラグ (DQ2) は，トグルビットフラグ (DQ6) と共に使用し，消去一時停止中であるかを検出するために使用します (DQ2 はトグル動作を行います，DQ6 はトグル動作を行いません)。
- トグルビット 2 フラグ (DQ2) は，消去しているセクタからのリードアクセスがされた場合，トグル動作を行いますので，消去しているセクタの検出にも使用できます。

---

< 参考 >	消去対象のすべてのセクタが書換え保護されている場合，トグルビット 2 フラグ (DQ2) は，約 100 $\mu$ s の間トグル出力した後，データを書換えることなく読出し / リセット状態に戻ります。
--------	--

---

## 23.6 フラッシュメモリ書込み / 消去の詳細説明

---

自動アルゴリズムを起動するコマンドを入力し、フラッシュメモリに読出し / 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動作を行う手順を説明します。

---

### ■ フラッシュメモリ書込み / 消去の詳細説明

自動アルゴリズムは、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、消去再開のコマンドシーケンスを CPU からフラッシュメモリへ書き込むことにより起動できます。CPU からフラッシュメモリへの書込みは、必ず連続して行ってください。また、自動アルゴリズムはデータポーリング機能により終了状態を確認できます。正常終了後は読出し / リセット状態に戻ります。

各動作について、下記の順に説明します。

- 読出し / リセット状態にする
- データを書き込む
- 全データを消去する（チップ全消去）
- 任意のデータを消去する（セクタ消去）
- セクタ消去を一時停止する
- セクタ消去を再開する

## 23.6.1 フラッシュメモリを讀出し / リセット状態にする

---

讀出し / リセットコマンドを入力し、フラッシュメモリを讀出し / 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讀出し / リセット状態にする

- フラッシュメモリを讀出し / リセット状態にするには、コマンドシーケンス表の讀出し / リセットコマンドを CPU からフラッシュメモリへ連続して送ってください。
- 讀出し / リセットコマンドには 1 回と 3 回のバス動作を行う 2 通りのコマンドシーケンスがありますが、違いはありません。
- 讀出し / リセット状態はフラッシュメモリの初期状態ですので、電源投入後、コマンドの正常終了後は常に讀出し / リセット状態になります。讀出し / リセット状態は、コマンドの入力待ち状態でもあります。
- 讀出し / リセット状態では、フラッシュメモリへ、リードアクセスすることによりデータを読み出すことができます。マスク ROM と同様に CPU からのプログラムアクセスができます。フラッシュメモリへリードアクセスする場合は、讀出し / リセットコマンドは必要ありません。コマンドが正常に終了しなかった場合は、自動アルゴリズムを初期化するために、讀出し / リセットコマンドを使用してください。

## 23.6.2 フラッシュメモリヘデータを書き込む

書込みコマンドを入力し、フラッシュメモリヘデータを書き込む手順について説明します。

### ■ フラッシュメモリヘデータを書き込む

- 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の書込みコマンドをCPUからフラッシュメモリへ連続して送ってください。
- 4 サイクル目に目的のアドレスへのデータ書込みが終了した場合、自動アルゴリズムが起動され自動書込みを開始します。

#### ● アドレス指定方法

- 書込みデータサイクルで指定する書込みアドレスは、偶数アドレスのみ設定できます。奇数アドレスを設定した場合は、正しく書き込むことができません。偶数アドレスへはワードデータ単位で書き込んでください。
- 書込みはどのようなアドレスの順番でも、セクタの境界を越えても可能です。1 回の書込みコマンドによって書き込まれるデータは 1 ワードのみです。

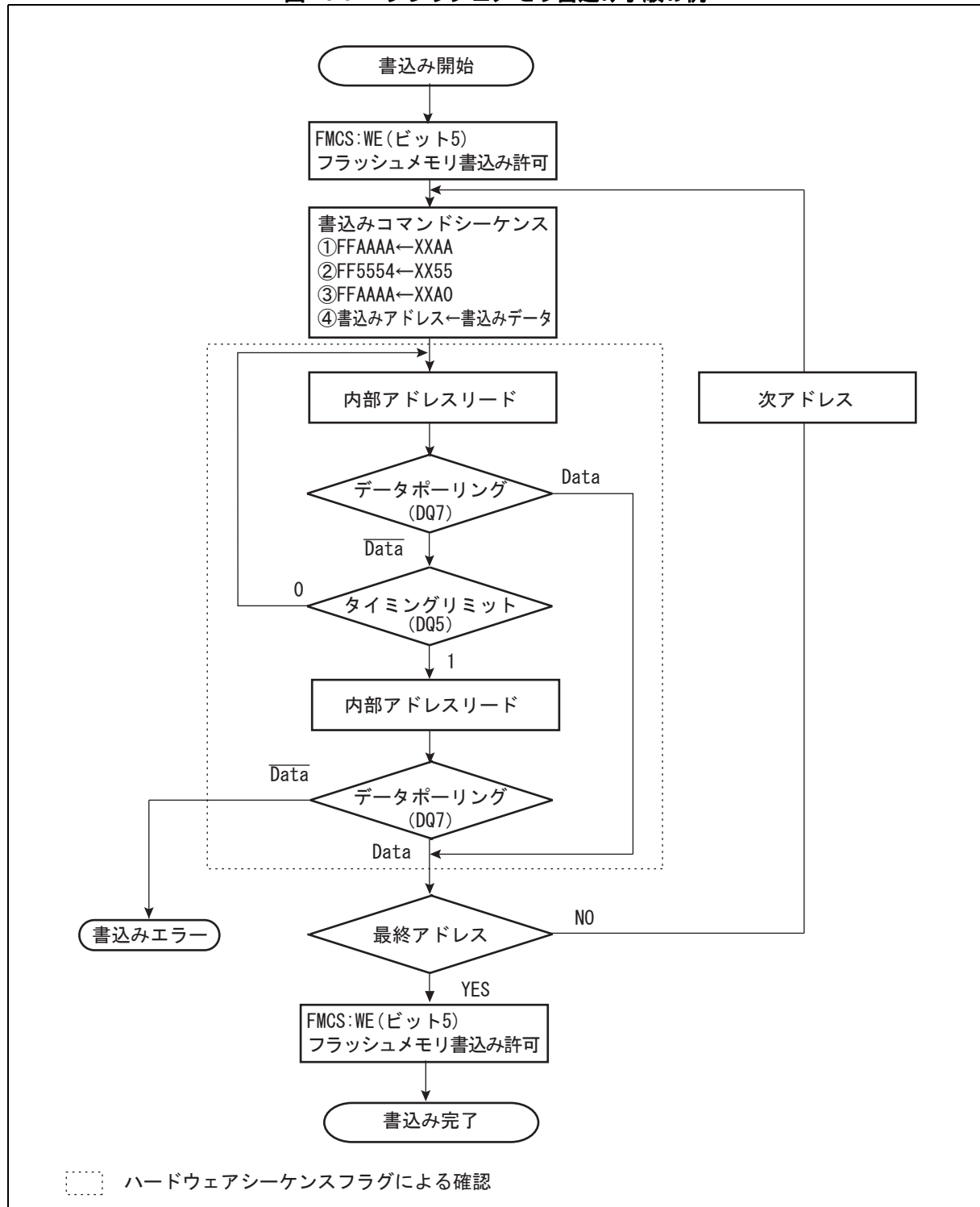
#### ● データ書込み上の注意

- 書込みによって、ビットデータを "0" から "1" へ戻すことはできません。ビットデータ "0" にビットデータ "1" を書き込むと、データポーリングアルゴリズム (DQ7) またはトグル動作 (DQ6) が終了せず、フラッシュメモリ素子が不良と判定され、書込み規定時間を超えてタイミングリミット超過フラグ (DQ5) がエラーと判定します。
- 読出し / リセット状態でデータを読み出した場合、ビットデータは "0" のままです。ビットデータを "0" から "1" に戻すには、フラッシュメモリを消去をしてください。
- 自動書込み実行中は、全てのコマンドが無視されます。書込み中にハードウェアリセットが起動された場合は、書込みアドレスのデータは保証されません。

### ■ フラッシュメモリ書込み手順

- 図 23.6-1 にフラッシュメモリ書込み手順の例を示します。ハードウェアシーケンスフラグ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できます。ここでは、フラッシュメモリへの書込み終了の確認にデータポーリングフラグ (DQ7) を用いています。
- フラグチェックのために読込むデータは、最後に書き込んだアドレスからの読込みとなります。
- データポーリングフラグ (DQ7) は、タイミングリミット超過フラグ (DQ5) と同時に変化するので、タイミングリミット超過フラグ (DQ5) が "1" であった場合でもデータポーリングフラグビット (DQ7) を確認してください。
- トグルビットフラグ (DQ6) も、タイミングリミット超過フラグビット (DQ5) が "1" に変化すると同時にトグル動作を止めますので、トグルビットフラグ (DQ6) を確認してください。

図 23.6-1 フラッシュメモリ書き込み手順の例





### 23.6.3 フラッシュメモリのデータを消去する（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。

---

#### ■ フラッシュメモリのデータを消去する（チップ消去）

- フラッシュメモリ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のチップ消去コマンドを CPU からフラッシュメモリ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、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。
- チップ消去の場合は、消去前にユーザがフラッシュメモリへ書き込む必要はありません。自動消去アルゴリズム実行中に、フラッシュメモリはすべてのセルを消去する前に "0" を自動的に書き込んでから消去します。

## 23.6.4 フラッシュメモリの任意のデータを消去する（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セクタを消去する（セクタ消去）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。

### ■ フラッシュメモリの任意のデータを消去する（セクタ消去）

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のセクタ消去コマンドを CPU からフラッシュメモリへ連続して送ってください。

#### ● セクタ指定方法

- セクタ消去コマンドは 6 回のバス動作で行われます。6 サイクル目のアドレスを目的のセクタ内の偶数アドレスに設定し、データにセクタ消去コード (30<sub>H</sub>) を書き込むことにより 50 $\mu$ s のセクタ消去ウェイトが開始します。
- 複数のセクタを消去する場合は、上記に続き消去する目的のセクタ内のアドレスに消去コード (30<sub>H</sub>) を書き込みます。

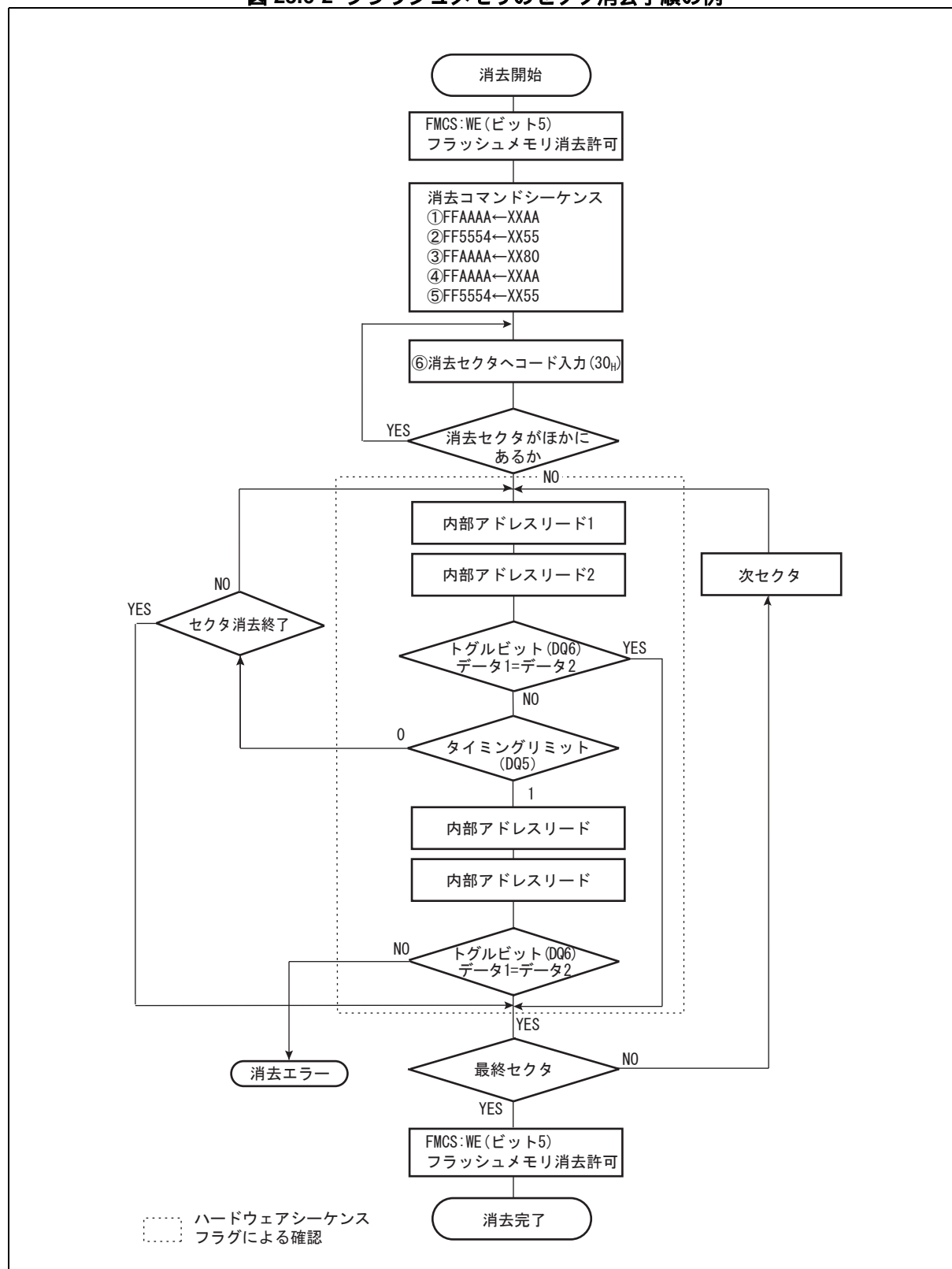
#### ● 複数のセクタを指定する場合の注意

- 最後のセクタ消去コードの書込みから 50 $\mu$ s のセクタ消去ウェイト期間終了により消去が開始します。
- 複数のセクタを同時に消去する場合は、消去セクタのアドレスと消去コード（コマンドシーケンス 6 サイクル目）を 50 $\mu$ s 以内に入力してください。50 $\mu$ s 以降に消去コードを入力した場合は、受け付けられません。
- 連続したセクタ消去コードの書込みが有効であるかは、セクタ消去タイマフラグ (DQ3) によって確認できます。
- セクタ消去タイマフラグ (DQ3) をリードする場合のアドレスは、消去しようとしているセクタを指定してください。

### ■ フラッシュメモリのセクタ消去手順

- ハードウェアシーケンスフラグ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できます。図 23.6-2 にフラッシュメモリセクタ消去手順の例を示します。ここでは、消去終了の確認にトグルビットフラグ (DQ6) を用いています。
- トグルビットフラグ (DQ6) は、タイミングリミット超過フラグ (DQ5) が "1" に変化するのと同時にトグル動作を終了します。タイミングリミット超過フラグ (DQ5) が "1" の場合でも、トグルビットフラグ (DQ6) を確認してください。
- データポーリングフラグ (DQ7) は、タイミングリミット超過フラグ (DQ5) と同時に変化するので、データポーリングフラグ (DQ7) を確認してください。

図 23.6-2 フラッシュメモリのセクタ消去手順の例



## 23.6.5 フラッシュメモリのセクタ消去を一時停止する

---

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を一時停止する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。

---

### ■ フラッシュメモリのセクタ消去を一時停止する

- フラッシュメモリのセクタ消去を一時停止するためには、コマンドシーケンス表のセクタ消去一時停止コマンドを CPU からフラッシュメモリへ送ってください。
- セクタ消去一時停止コマンドは、セクタ消去中に消去を一時停止し、消去中でないセクタからデータを読み出すことができます。セクタ消去一時停止状態では、読出しのみが可能であり、書込みはできません。
- セクタ消去一時停止コマンドは、消去ウェイト時間を含むセクタ消去中のみ有効であり、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コマンドは、消去一時停止コード (B0<sub>H</sub>) を書き込むことで実行されます。このときのアドレスは、フラッシュメモリ内の任意のアドレスを設定してください。消去一時停止中に再度、消去一時停止コマンドを実行した場合、再度入力したコマンドは無視されます。
- セクタ消去ウェイト期間中に、セクタ消去一時停止コマンドを入力した場合、直ちにセクタ消去ウェイトを終了し、消去動作を中断して消去停止状態になります。
- セクタ消去ウェイト期間後のセクタ消去中に消去一時停止コマンドを入力した場合、最大 20 $\mu$ s 後に消去一時停止状態になります。セクタ消去一時停止コマンドは、セクタ消去コマンドあるいはセクタ消去再開コマンド発行後、20 $\mu$ s 以上後に行ってください。

## 23.6.6 フラッシュメモリのセクタ消去を再開する

---

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を再開する

- 一時停止したセクタ消去を再開させるためには、コマンドシーケンス表のセクタ消去再開コマンドを CPU からフラッシュメモリへ送ってください。
- 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るセクタ消去一時停止状態から、セクタ消去を再開するためのコマンドです。セクタ消去再開コマンドは消去再開コード (30<sub>H</sub>) を書き込むことで実行されますが、アドレスはフラッシュメモリ内の任意のアドレスを指定します。
- セクタ消去中のセクタ消去再開コマンドの入力は無視されます。

## 23.7 フラッシュセキュリティの特長

---

フラッシュセキュリティコントローラ機能により、フラッシュメモリの内容を外部端子から読み出されることを防止できます。

---

### ■ フラッシュセキュリティの特長

事前定義されたフラッシュメモリのアドレスが、フラッシュセキュリティコントローラ (MB90F462, MB90F462A: FF0001<sub>H</sub>; MB90F463A: FE0001<sub>H</sub>) に割り当てられます。このアドレスに保護コード "01<sub>H</sub>" が書き込まれると、フラッシュメモリへのアクセスが制限されます。フラッシュメモリが一度保護されると、チップ消去を行うまで、機能のロックを解除することはできません。ロックを解除しない限り、いずれの外部端子からもフラッシュメモリへの読出し / 書込みはできません。

この機能は、フラッシュメモリに格納される自己完結型プログラムやデータのセキュリティを必要とするアプリケーションに適しています。マイクロコントローラの外部に位置するプログラム部分を対象アプリケーションが必要とする場合には、フラッシュセキュリティコントローラは、目的の機能を提供できません。このため、保護コードを設定した場合は、外部ベクタフェッチモードは使用しません。

標準パラレルプログラマによるフラッシュマイクロコントローラのプログラミングには、独特のセットアップが必要になることがあります。たとえば、ミナトエレクトロニクスのプログラマでは、デバイスチェックを切断する必要があります。保護コードは、フラッシュプログラミングの終わりにコーディングすることを推奨されます。これは、プログラミング中の不要な保護を回避するためです。

一度保護されたフラッシュメモリを再度プログラムするには、チップ消去操作を行う必要があります。

詳細については、弊社担当者にお問い合わせください。

## 23.8 512K ビットフラッシュメモリのプログラム例

512K ビットフラッシュメモリのプログラム例を掲載します。

### ■ 512K ビットフラッシュメモリのプログラム例

```

NAME      FLASHWE
TITLE     FLASHWE

;-----
;512kbit-FLASH サンプルプログラム
;1:FLASH にあるプログラム ( アドレス FFBC00H セクタ SA2) を RAM( アドレス 000700H)
; に転送する。
;2:RAM 上でプログラムを実行する。
;3:PDR1 の値を FLASH ( アドレス FF0000H セクタ SA0 ) に書き込む。
;4: 書き込んだ値 ( アドレス FF0000H セクタ SA0 ) を読み出し PDR2 に出力する。
;5: 書き込んだセクタ ( SA0 ) を消去する。
;6: 消去データ確認の出力
; 条件
;   ・ RAM 転送バイト数 : 100H ( 256 バイト )
;   ・ 書込み , 消去の終了判定
;       DQ5 ( タイミングリミット超過フラグ ) での判定
;       DQ6 ( トグルビットフラグ ) での判定
;       RDY ( FMCS ) での判定
;   ・ エラー時の処理
;       P00 ~ P07 に "H" を出力する
;       リセットコマンド発行
;-----
;
RESOUS    IOSEG ABS=00          ; "RESOUS" I/O セグメントの定義
          ORG     0000H
PDR0      RB      1
PDR1      RB      1
PDR2      RB      1
PDR3      RB      1
          ORG     0010H
DDR0      RB      1
DDR1      RB      1
DDR2      RB      1
DDR3      RB      1
          ORG     00A1H
CKSCR     RB      1
          ORG     00AEH
FMCS      RB      1
          ORG     006FH
ROMM      RB      1
RESOUS    ENDS
;
SSTA      SSEG
          RW      0127H
STA__T    RW      1
SSTA      ENDS
;

```

```

DATA    DSEG  ABS=0FFH      ;FLASH コマンドアドレス
        ORG    5554H
COMADR2  RW    1
        ORG    0AAAAH
COMADR1  RW    1
DATA     ENDS

;-----
; メインプログラム ( SA1 )
;-----
CODE     CSEG
START:
;-----
; 初期化
;-----
MOV      CKSCR,#0BAH  ; 3 週に設定
MOV      RP,#0
MOV      A,#!STA_T
MOV      SSB,A
MOVW     A,#STA_T
MOVW     SP,A
MOV      ROMM,#00H     ; ミラー OFF
MOV      PDR0,#00H     ; エラー確認用
MOV      DDR0,#0FFH
MOV      PDR1,#00H     ; データ入力用ポート
MOV      DDR1,#00H
MOV      PDR2,#00H     ; データ出力用ポート
MOV      DDR2,#0FFH
;-----
; RAM ( 700H 番地 ) に "FLASH 書き込み消去プログラム ( FFBC00H )" を転送する
;-----
MOVW     A,#0700H      ; 転送先 RAM 領域
MOVW     A,#0BC00H     ; 転送元アドレス ( プログラムのある位置 )
MOVW     RW0,#100H     ; 転送するバイト数
MOVS     ADB,PCB       ; FFBC00H から 000700H へ 100H 転送
CALLP    000700H       ; 転送したプログラムのあるアドレスへジャンプ
;-----
; データ出力
;-----
OUT      MOV      A,#0FFH
        MOV      ADB,A
        MOVW     RW2,#0000H
        MOVW     A,@RW2+00
        MOV      PDR2,A
END      JMP      *
CODE     ENDS

;-----
; FLASH 書き込み消去プログラム ( SA2 )
;-----
RAMPRG   CSEG  ABS=0FFH
        ORG    0BC00H
;-----
; 初期化
;-----
MOVW     RW0,#0500H  ; RW0: 入力データ確保用 RAM 空間 00:0500 ~
MOVW     RW2,#0000H  ; RW2: フラッシュメモリ書き込みアドレス FD:0000 ~

```



## 第 23 章 512K/1024K ビット フラッシュメモリ

```

MOV      A, #00H      ;DTB 変更
MOV      DTB, A       ;@RW0 用バンク指定
MOV      A, #0FFH     ;ADB 変更 1
MOV      ADB, A       ; 書込みモード指定アドレス用バンク指定
MOV      PDR3, #00H   ; スイッチ初期化
MOV      DDR3, #00H

;
WAIT1    BBC          PDR3:0, WAIT1; PDR3:0 "H" で書込みスタート
;
; -----
; 書込み (SA0)
; -----
MOV      A, PDR1
MOVW     @RW0+00, A    ;RAM に PDR1 データを確保
MOV      FMCS, #20H    ; 書込みモード設定
MOVW     ADB:COMADR1, #00AAH ; フラッシュ書込みコマンド 1
MOVW     ADB:COMADR2, #0055H ; フラッシュ書込みコマンド 2
MOVW     ADB:COMADR1, #00A0H ; フラッシュ書込みコマンド 3
;
MOVW     A, @RW0+00    ; 入力データ (RW0) をフラッシュメモリ (RW2) へ書込む
;
MOVW     @RW2+00, A
WRITE    ; 待ち時間チェック
; -----
; タイムリミット超過チェック - フラグが立ちトグル動作中である場合
; ERROR
; -----
MOVW     A, @RW2+00
AND      A, #20H       ;DQ5 タイムリミットチェック
BZ       NTOW          ; タイムリミットオーバー
MOVW     A, @RW2+00    ;AH
MOVW     A, @RW2+00    ;AL
XORW     A             ;AH AL の XOR (値が違えば "1")
AND      A, #40H       ;DQ6 トグルビットは違っているか
BNZ      ERROR         ; 違えば ERROR へ
; -----
; 書込み終了チェック (FMCS-RDY)
; -----
NTOW     MOVW     A, FMCS
AND      A, #10H       ;FMCS RDY ビット (4bit) 抽出
BZ       WRITE         ; 書込み終了か?
MOV      FMCS, #00H    ; 書込みモード解除
; -----
; 書込みデータ出力
; -----
MOVW     RW2, #0000H    ; 書込みデータ出力
MOVW     A, @RW2+00
MOV      PDR2, A
;
WAIT2    BBC          PDR3:1, WAIT2; PDR3:1 "H" でセクタ消去スタート
;

```

```

;-----
; セクタ消去 (SA0)
;-----
        MOV    @RW2+00,#0000H; アドレス初期化
        MOV    FMCS,#20H      ; 消去モード設定
        MOVW   ADB:COMADR1,#00AAH      ; フラッシュ消去コマンド 1
        MOVW   ADB:COMADR2,#0055H      ; フラッシュ消去コマンド 2
        MOVW   ADB:COMADR1,#0080H      ; フラッシュ消去コマンド 3
        MOVW   ADB:COMADR1,#00AAH      ; フラッシュ消去コマンド 4
        MOVW   ADB:COMADR2,#0055H      ; フラッシュ消去コマンド 5
        MOV    @RW2+00,#0030H      ; 消すセクタに消去コマンド発行 6
ELS                                           ; 待ち時間チェック

;-----
                                           ; タイムリミット超過チェック - フラグが
                                           ; 立ちトグル動作中である場合

        ; ERROR
;-----
        MOVW   A,@RW2+00
        AND    A,#20H           ; DQ5 タイムリミットチェック
        BZ     NTOE             ; タイムリミットオーバー
        MOVW   A,@RW2+00        ; AH 書込み動作中は, DQ6 から
        MOVW   A,@RW2+00        ; AL リード毎 "H/L" が交互出力される
        XORW   A                ; AH と AL の XOR (DQ6 の値が違えば
                                ; "1" 書込み動作中である)
        AND    A,#40H           ; DQ6 トグルビットは "H" か
        BNZ    ERROR           ; "H" なら ERROR へ

;-----
        ; 消去終了チェック (FMCS-RDY)
;-----
NTOE     MOVW   A,FMCS           ;
        AND    A,#10H           ; FMCS RDY ビット (4bit) 抽出
        BZ     ELS              ; セクタ消去終了か?
        MOV    FMCS,#00H        ; FLASH 消去モード解除
        RETP                    ; メインプログラムに戻る

;-----
; エラー
;-----
ERROR    MOV    ADB:COMADR1,#0F0H
                                ; リセットコマンド (読出しが可能になる)
        MOV    FMCS,#00H        ; FLASH モード解除
        MOV    PDR0,#0FFH      ; エラー処理の確認
        RETP                    ; メインプログラムに戻る
RAMPRG   ENDS

;-----
VECT     CSEG   ABS=0FFH
        ORG    0FFDCH
        DSL    START
        DB     00H
VECT     ENDS

;
        END    START

```



# 第24章

---

## フラッシュシリアル 書込み接続例

横河ディジタルコンピュータ株式会社製フラッシュマイコンプログラマを用いた場合の、シリアル書込み接続例について説明します。

- 24.1 F<sup>2</sup>MC-16LX ファミリ MB90F462/F462A/F463A シリアル書込み接続の基本構成
- 24.2 シングルチップモード時の接続例（ユーザ電源）
- 24.3 シングルチップモード時の接続例（ライター電源）
- 24.4 フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源）
- 24.5 フラッシュマイコンプログラマとの最小限の接続例（ライター電源）

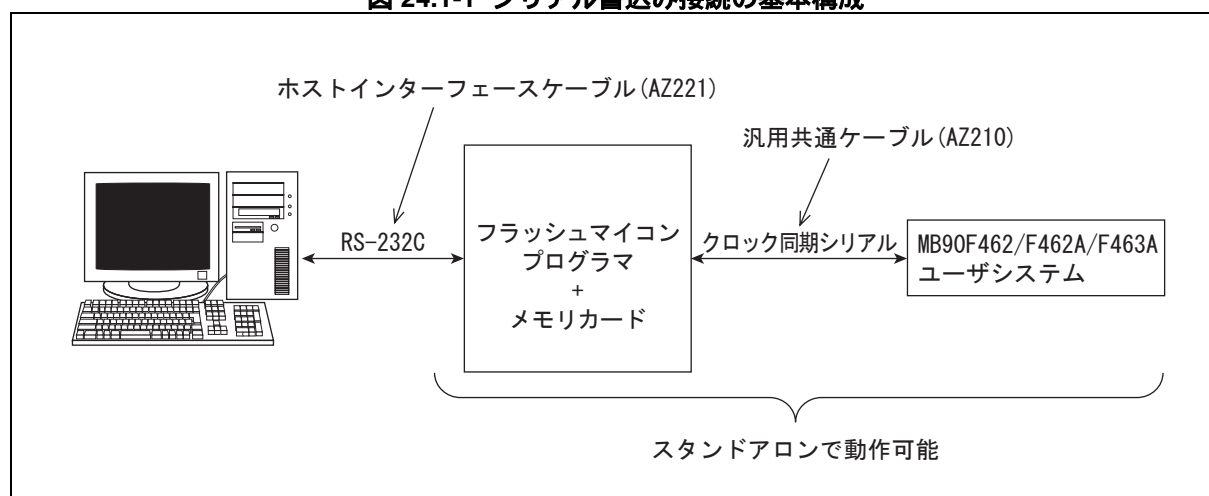
## 24.1 F<sup>2</sup>MC-16LX ファミリ MB90F462/F462A/F463A シリアル書込み接続の基本構成

MB90F462/F462A/F463A は、フラッシュ ROM の富士通標準シリアルオンボード書込みをサポートしています。オンボード書込み仕様について以下に解説します。

### ■ MB90F462/F462A/F463A シリアル書込み接続の基本構成

富士通標準シリアルオンボード書込みには、横河デジタルコンピュータ株式会社製フラッシュマイコンプログラマを使用します。

図 24.1-1 シリアル書込み接続の基本構成



< 注意事項 > フラッシュマイコンプログラマ (AF220/AF210/AF120/AF110) の機能や操作方法、接続用汎用共通ケーブル (AZ210)、コネクタにつきましては、横河デジタルコンピュータ株式会社へお問い合わせください。

表 24.1-1 富士通標準シリアルオンボード書込みに使用する端子

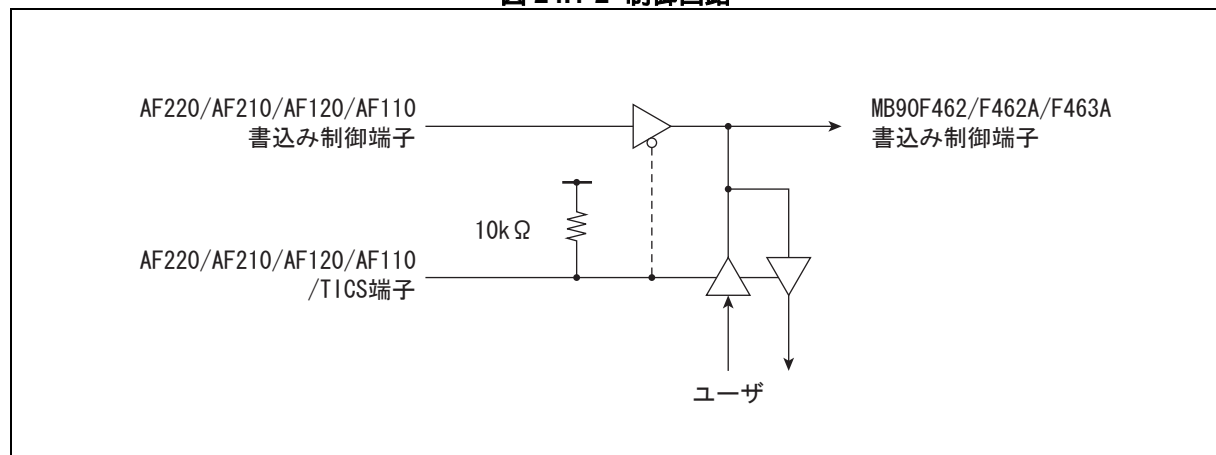
端 子	端子機能説明	補足説明
MD2, MD1, MD0	モード入力端子	MD2=1, MD1=1, MD0=0 に設定した場合, フラッシュシリアル書込みモードになります。
X0, X1	発振クロック端子	フラッシュシリアル書込みモード時は, CPU 内部動作クロックは PLL クロックの 1 通倍になります。発振クロック周波数が CPU 内部動作クロックになりますので, シリアル書込みに使用できる振動子は 1MHz ~ 16MHz になります。
P00, P01	書込みプログラム起動端子	書込みプログラムを起動するには P00 を "0" に, P01 を "1" に設定してください。
$\overline{\text{RST}}$	リセット端子	-
SIN0	シリアルデータ入力端子	UART をクロック同期モード (モード 0) で使用します。
SOT0	シリアルデータ出力端子	
SCK0	シリアルクロック入力端子	
C	C 端子	電源安定化の容量端子です。外部に 0.1 $\mu$ F 程度のセラミックコンデンサを接続してください。
V <sub>CC</sub>	電源電圧供給端子	書込み電圧 (5V $\pm$ 10%)
V <sub>SS</sub>	GND 端子	フラッシュマイコンプログラムの GND と共通にしてください。

< 注意事項 > P00, SIN0, SOT0, SCK0 端子をユーザシステムでも使用される場合には, 図 24.1-2 の制御回路が必要になります。フラッシュマイコンプログラムの /TICS 信号により, シリアル書込み中の場合にはユーザ回路を切り離すことができます。

次節以降に, 以下のシリアル書込み接続例を示しますのでご参照ください。

- ・ シングルチップモード接続例 (ユーザ電源)
- ・ シングルチップモード接続例 (ライタ電源)
- ・ フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源)
- ・ フラッシュマイコンプログラムの最小限の接続例 (ライタ電源)

図 24.1-2 制御回路



## ■ 発振クロック周波数とシリアルクロック入力周波数

MB90F462/F462A/F463Aの入力可能なシリアルクロック周波数は、以下の計算式によって求めることができますので、ご使用の発振クロック周波数に合わせて、シリアルクロック入力周波数をフラッシュマイコンプログラマに設定してください。

入力可能なシリアルクロック周波数 =  $0.125 \times$  発振クロック周波数

表 24.1-2 入力可能なシリアルクロック周波数の例

発振クロック周波数	マイコンの入力可能な最大シリアルクロック周波数	AF220/AF210/AF120/AF110 の設定可能な最大シリアルクロック周波数	AF200 の設定可能な最大シリアルクロック周波数
4MHz 時	500kHz	500kHz	500kHz
8MHz 時	1MHz	850kHz	500kHz
16MHz 時	2MHz	1.25MHz	500kHz

## ■ フラッシュマイコンプログラマのシステム構成（横河デジタルコンピュータ株式会社製）

表 24.1-3 フラッシュマイコンプログラマのシステム構成  
（横河デジタルコンピュータ株式会社製）

型 格		機 能
本体	AF220/AC4P	イーサネットインタフェース内蔵モデル /100V ~ 220V 電源アダプタ
	AF210/AC4P	スタンダードモデル /100V ~ 220V 電源アダプタ
	AF120/AC4P	単キー イーサネットインタフェース内蔵モデル /100V ~ 220V 電源アダプタ
	AF110/AC4P	単キーモデル /100V ~ 220V 電源アダプタ
AZ221		ライター専用 PC/AT 用 RS232C ケーブル
AZ210		標準ターゲットプローブ (a) 長さ : 1m
FF201		富士通製 F <sup>2</sup> MC-16LX フラッシュマイコン用コントロールモジュール
/P2		2MB PC Card ( Option ) FLASH メモリ容量 ~ 128KB 対応
/P4		4MB PC Card ( Option ) FLASH メモリ容量 ~ 512KB 対応

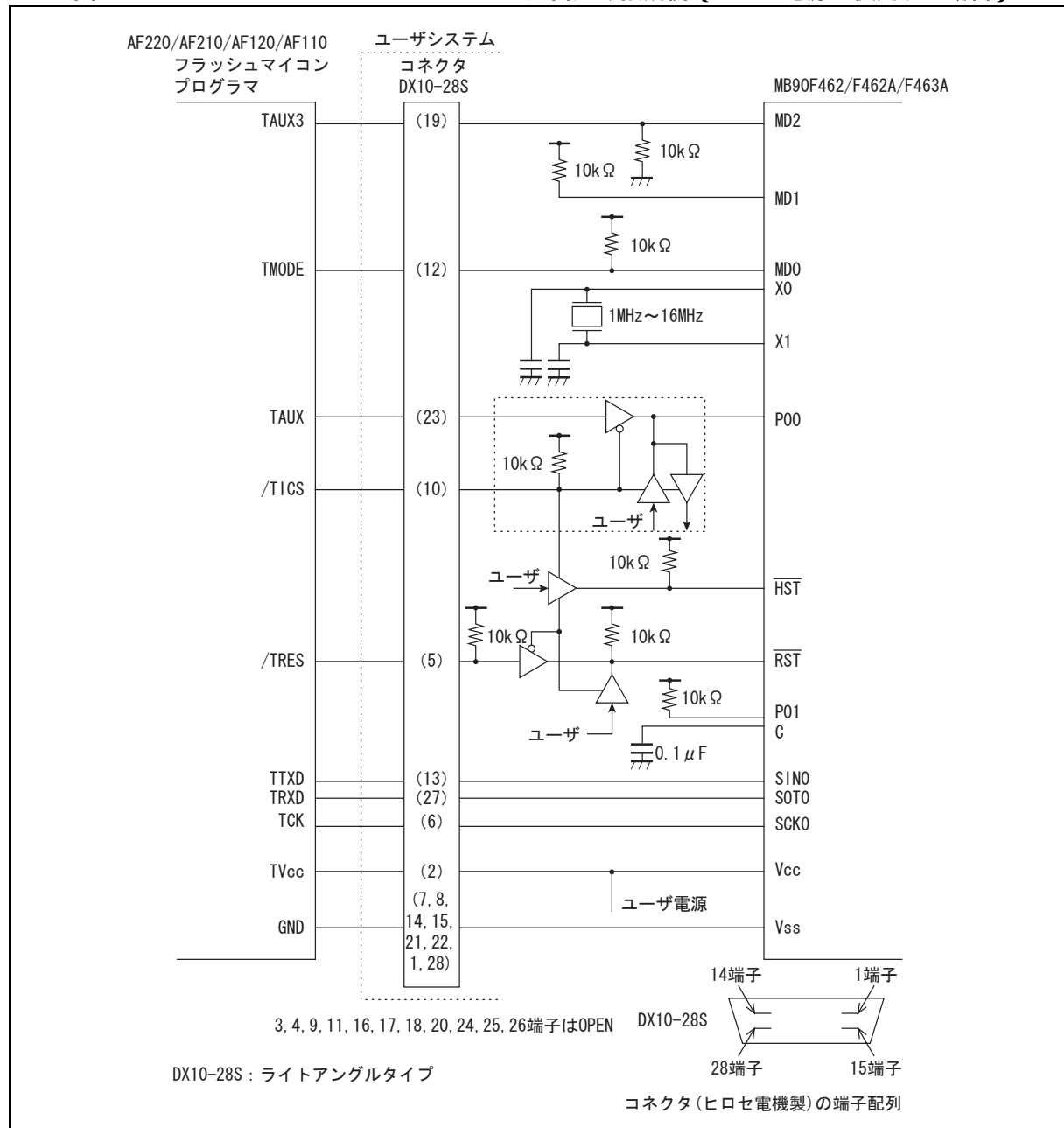
< 注意事項 > AF200 フラッシュマイコンプログラマは終息製品ですが、コントロールモジュール FF201 を用いることにより使用できます。シリアル書込み接続例を、次節以降に示します。

## 24.2 シングルチップモード時の接続例（ユーザ電源）

シングルチップモードに設定されたユーザシステムのモード端子（MD2, MD0）へ，AF220/AF210/AF210/AF120/AF110 の端子（TAUX3, TMODE）より MD2=1, MD0=0 が設定され，フラッシュシリアル書込みモードになります。ユーザ電源時の接続例を示します。

### ■ シングルチップモード時の接続例（ユーザ電源を使用する場合）

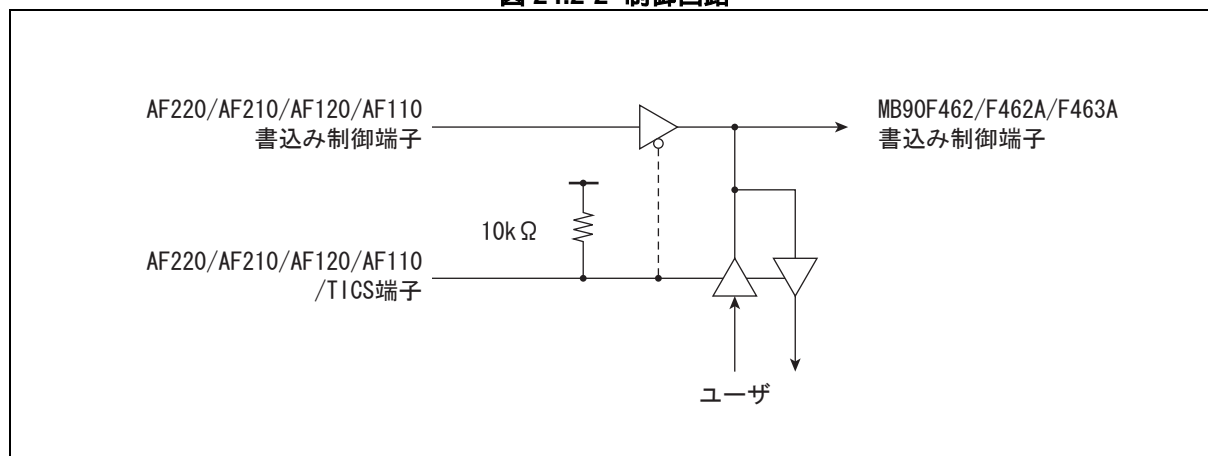
図 24.2-1 MB90F462/F462A/F463A シリアル書込み接続例（ユーザ電源を使用する場合）





- 
- < 注意事項 >
- SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, P00 に接続されている制御回路と同様の回路 (図 24.2-2) が必要となります。フラッシュマイコンプログラムの /TICS 信号により, シリアル書込み中の場合にはユーザ回路を切離すことができます。
  - AF220/AF210/AF120/AF110 との接続は, ユーザ電源が OFF の状態で行ってください。
- 

図 24.2-2 制御回路

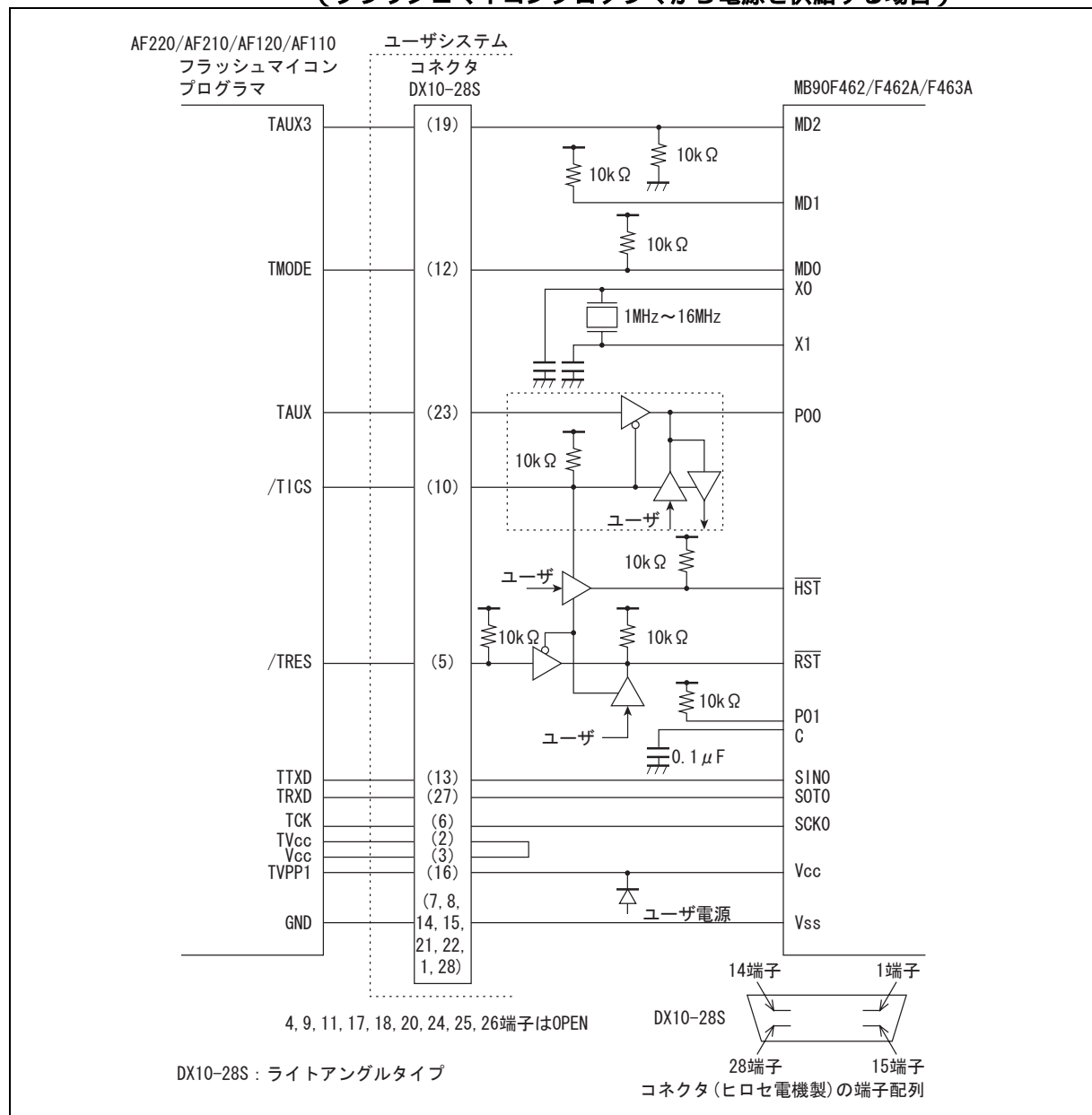


## 24.3 シングルチップモード時の接続例（ライタ電源）

シングルチップモードに設定されたユーザシステムのモード端子（MD2, MD0）へ，AF220/AF210/AF210/AF120/AF110 の端子（TAUX3, TMODE）より MD2=1, MD0=0 が設定され，フラッシュシリアル書込みモードになります。ライタ電源時の接続例を示します。

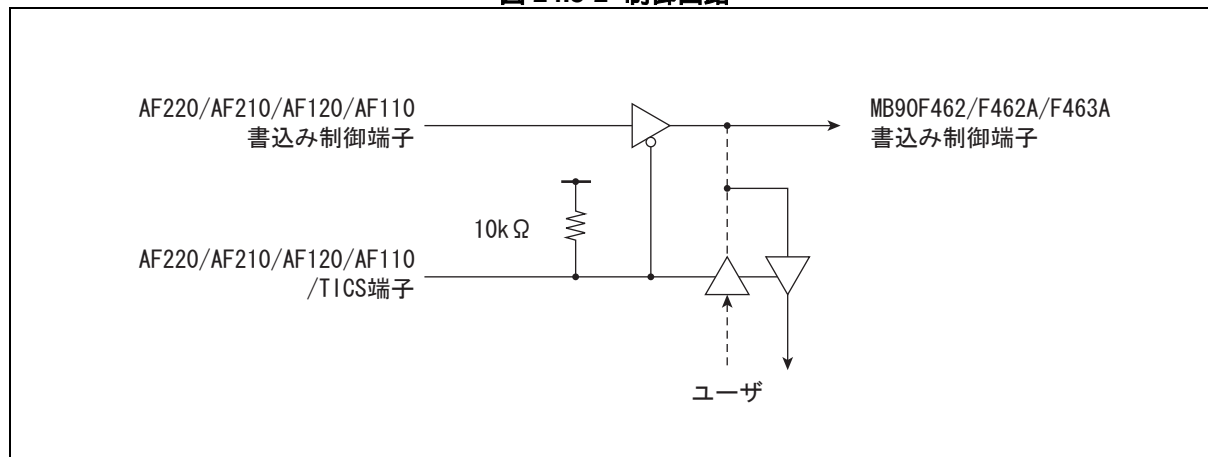
### ■ シングルチップモード時の接続例（フラッシュマイコンプログラマから電源を供給する場合）

図 24.3-1 MB90F462/F462A/F463A シリアル書込み接続例  
（フラッシュマイコンプログラマから電源を供給する場合）



- < 注意事項 >
- SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, P00 に接続されている制御回路と同様の制御回路 (図 24.3-2) が必要となります。フラッシュマイコンプログラムの /TICS 信号により, シリアル書込み中の場合にはユーザ回路を切離すことができます。
  - AF220/AF210/AF120/AF110 との接続は, ユーザ電源が OFF の状態で行ってください。
  - 書込み電源を AF220/AF210/AF120/AF110 から供給する場合は, ユーザ電源と短絡しないでください。

図 24.3-2 制御回路

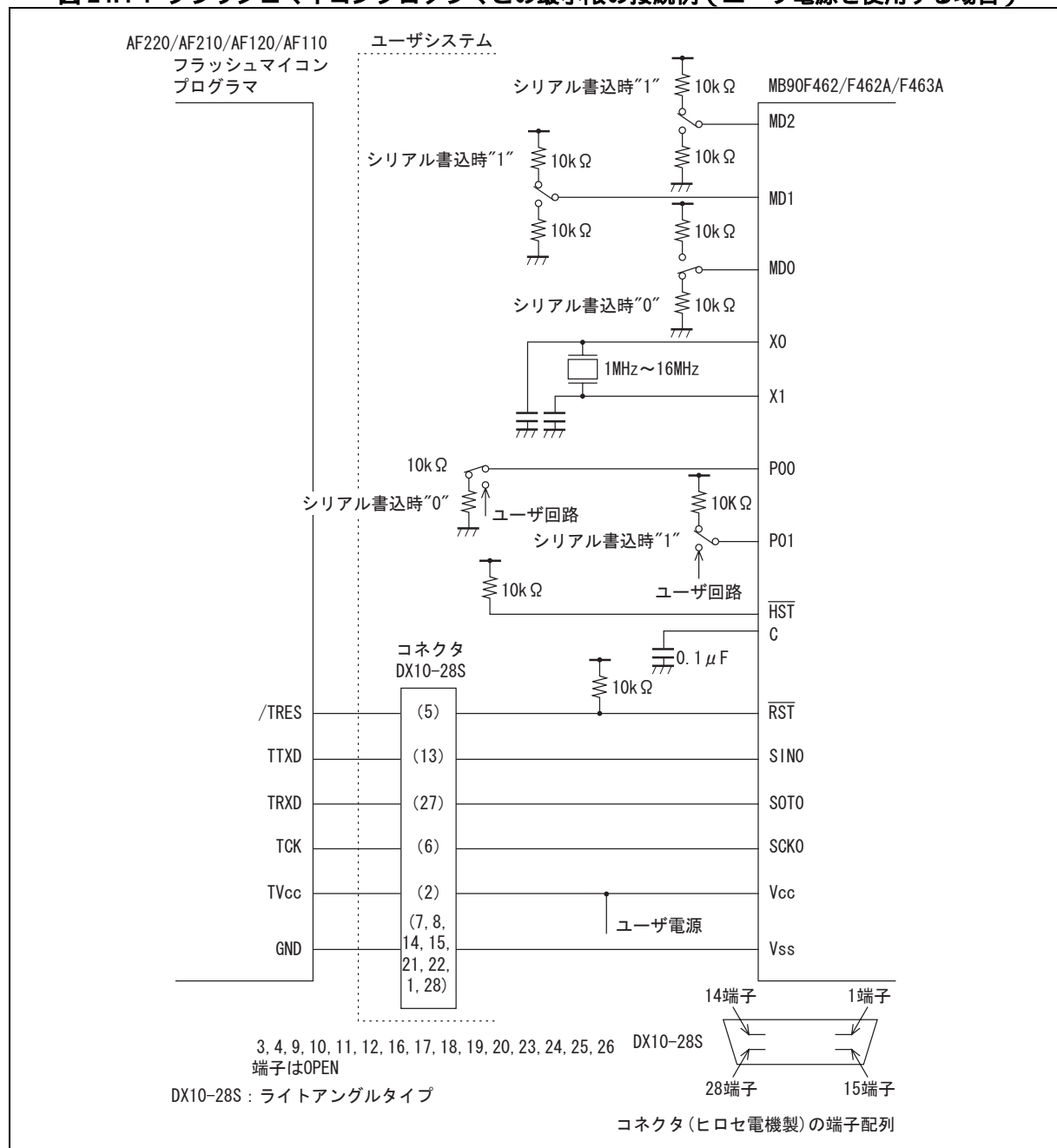


## 24.4 フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源)

シリアル書込み時に各端子 (MD2, MD0, P00) を図 24.4-1 のように設定した場合は, MD2, MD0, P00 とフラッシュマイコンプログラムの接続は必要ありません。

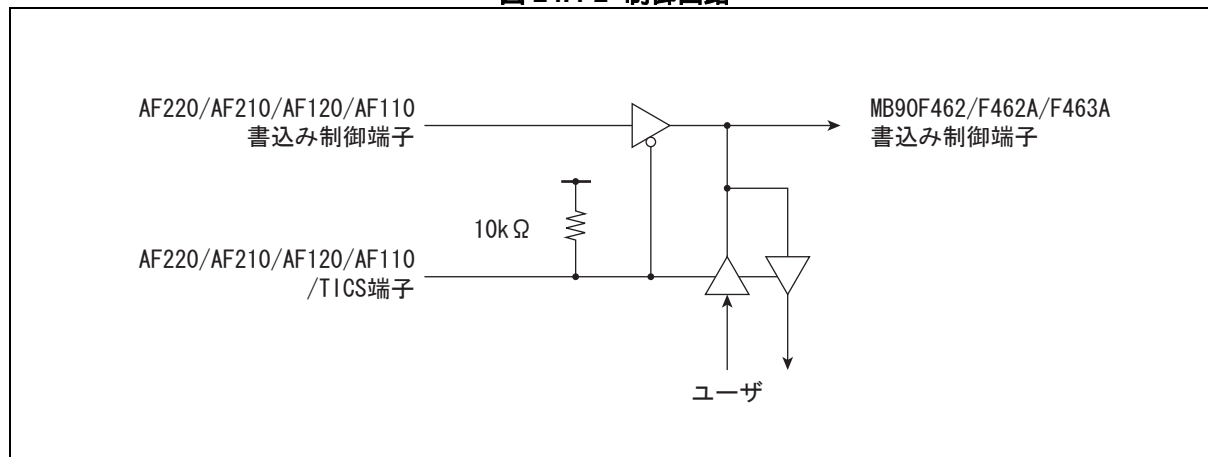
### ■ フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源を使用する場合)

図 24.4-1 フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源を使用する場合)



- < 注意事項 >
- SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, 図 24.4-2 の制御回路が必要となります。フラッシュマイコンプログラムの /TICS 信号により, シリアル書込み中の場合にはユーザ回路を切離すことができます。
  - AF220/AF210/AF120/AF110 との接続は, ユーザ電源が OFF の状態で行ってください。

図 24.4-2 制御回路

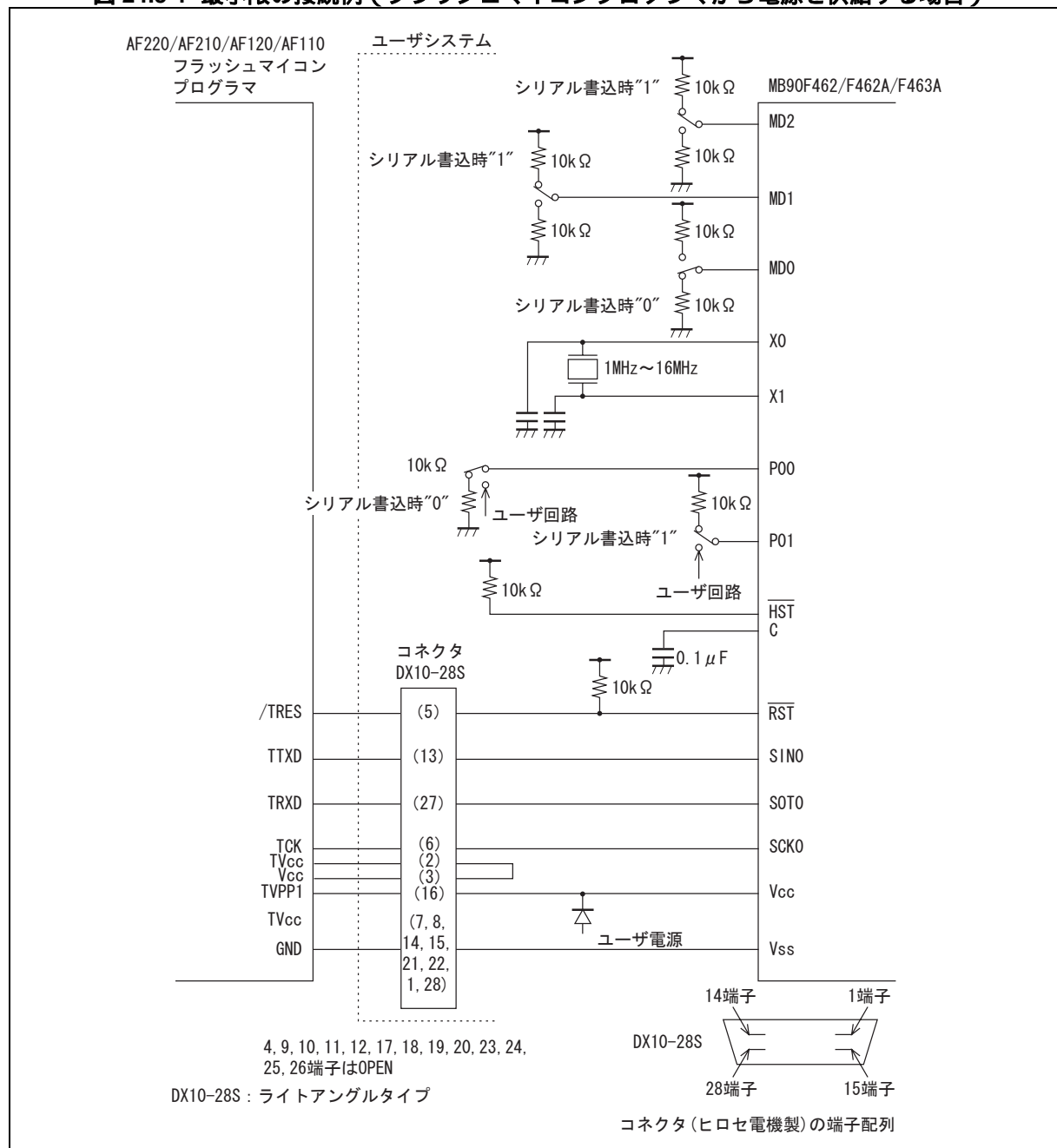


## 24.5 フラッシュマイコンプログラムの最小限の接続例 (ライタ電源)

シリアル書込み時に各端子 (MD2, MD0, P00) を図 24.5-1 のように設定した場合は, MD2, MD0, P00 とフラッシュマイコンプログラムの接続は必要ありません。

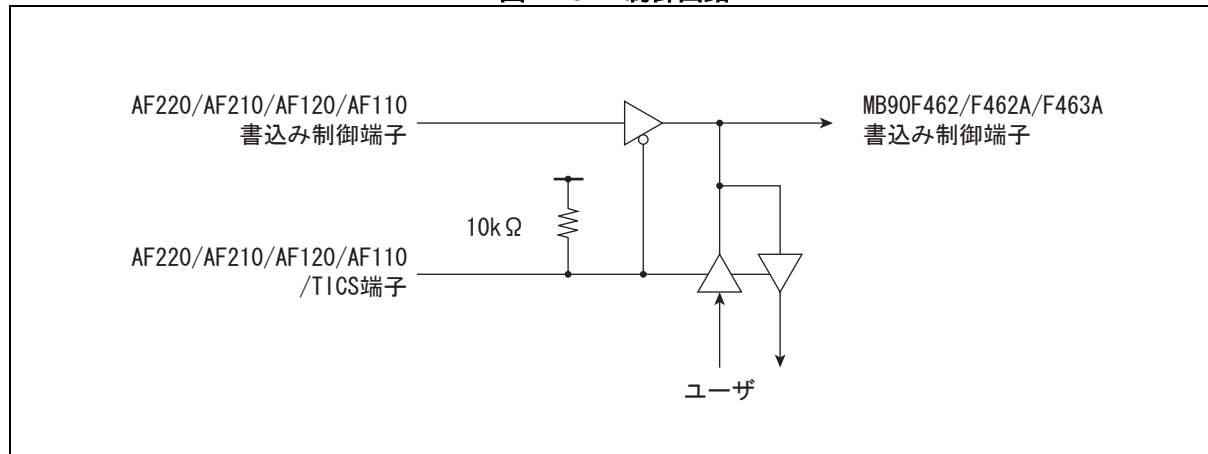
### ■ 最小限の接続例 (フラッシュマイコンプログラマから電源を供給する場合)

図 24.5-1 最小限の接続例 (フラッシュマイコンプログラマから電源を供給する場合)



- < 注意事項 >
- SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, 図 24.5-2 の制御回路が必要となります。フラッシュマイコンプログラムの /TICS 信号により, シリアル書込み中の場合にはユーザ回路を切離すことができます。
  - AF220/AF210/AF120/AF110 との接続は, ユーザ電源が OFF の状態で行ってください。
  - 書込み電源を AF220/AF210/AF120/AF110 から供給する場合は, ユーザ電源と短絡しないでください。

図 24.5-2 制御回路



## ■ シリアルライター対応

表 24.5-1 シリアルライター対応表

シリアルライター	概 要
AF200 (横河デジタルコンピュータ (株) 製)	ハンディタイプの 1 個書きシリアルライター
ITF2000 (インタフェース (株) 製)	最大 10 個同時書込み可能なシリアルギャングライター

付録では、I/O マップ、命令などを記載します。

付録 A I/O マップ

付録 B 命令



## 付録 A I/O マップ

MB90460/465 シリーズに内蔵されたリソースの各レジスタは、表 A-1 に示すようなアドレスに割当てられています。

### ■ I/O マップ

表 A-1 I/O マップ (1 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000000 <sub>H</sub>	PDR0	ポート 0 データレジスタ	R/W	R/W	ポート 0	XXXXXXXX <sub>B</sub>
000001 <sub>H</sub>	PDR1	ポート 1 データレジスタ	R/W	R/W	ポート 1	XXXXXXXX <sub>B</sub>
000002 <sub>H</sub>	PDR2	ポート 2 データレジスタ	R/W	R/W	ポート 2	XXXXXXXX <sub>B</sub>
000003 <sub>H</sub>	PDR3	ポート 3 データレジスタ	R/W	R/W	ポート 3	XXXXXXXX <sub>B</sub>
000004 <sub>H</sub>	PDR4	ポート 4 データレジスタ	R/W	R/W	ポート 4	-XXXXXXXX <sub>B</sub>
000005 <sub>H</sub>	PDR5	ポート 5 データレジスタ	R/W	R/W	ポート 5	XXXXXXXX <sub>B</sub>
000006 <sub>H</sub>	PDR6	ポート 6 データレジスタ	R/W	R/W	ポート 6	----XXXX <sub>B</sub>
000007 <sub>H</sub>	使用禁止					
000008 <sub>H</sub>	PWCSL0	PWC 状態制御レジスタ CH0 ( 下位 )	R/W	R/W	PWC タイマ ( CH0 )	00000000 <sub>B</sub>
000009 <sub>H</sub>	PWCSH0	PWC 状態制御レジスタ CH0 ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
00000A <sub>H</sub>	PWCH, L0	PWC データバッファレジスタ CH0	-	R/W		XXXXXXXX <sub>B</sub>
00000B <sub>H</sub>						XXXXXXXX <sub>B</sub>
00000C <sub>H</sub>	DIV0	分周比制御レジスタ CH0	R/W	R/W		-----00 <sub>B</sub>
00000D <sub>H</sub> ~ 00000F <sub>H</sub>	使用禁止					
000010 <sub>H</sub>	DDR0	ポート 0 データ方向レジスタ	R/W	R/W	ポート 0	00000000 <sub>B</sub>
000011 <sub>H</sub>	DDR1	ポート 1 データ方向レジスタ	R/W	R/W	ポート 1	00000000 <sub>B</sub>
000012 <sub>H</sub>	DDR2	ポート 2 データ方向レジスタ	R/W	R/W	ポート 2	00000000 <sub>B</sub>
000013 <sub>H</sub>	DDR3	ポート 3 データ方向レジスタ	R/W	R/W	ポート 3	00000000 <sub>B</sub>
000014 <sub>H</sub>	DDR4	ポート 4 データ方向レジスタ	R/W	R/W	ポート 4	-0000000 <sub>B</sub>
000015 <sub>H</sub>	DDR5	ポート 5 データ方向レジスタ	R/W	R/W	ポート 5	00000000 <sub>B</sub>
000016 <sub>H</sub>	DDR6	ポート 6 データ方向レジスタ	R/W	R/W	ポート 6	----0000 <sub>B</sub>

表 A-1 I/O マップ (2 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000017 <sub>H</sub>	ADER	アナログデータ入力許可レジスタ	R/W	R/W	ポート 5, A/D	11111111 <sub>B</sub>
000018 <sub>H</sub>	使用禁止					
000019 <sub>H</sub>	CDCR0	通信スケーラ制御レジスタ 0	R/W	R/W	通信プリス ケーラ 0	0XXX0000 <sub>B</sub>
00001A <sub>H</sub>	使用禁止					
00001B <sub>H</sub>	CDCR1	通信スケーラ制御レジスタ 1	R/W	R/W	通信プリス ケーラ 1	0XXX0000 <sub>B</sub>
00001C <sub>H</sub>	RDR0	ポート 0 プルアップ抵抗設定 レジスタ	R/W	R/W	ポート 0	00000000 <sub>B</sub>
00001D <sub>H</sub>	RDR1	ポート 1 プルアップ抵抗設定 レジスタ	R/W	R/W	ポート 1	00000000 <sub>B</sub>
00001E <sub>H</sub> , 00001F <sub>H</sub>	使用禁止					
000020 <sub>H</sub>	SMR0	シリアルモードレジスタ 0	R/W	R/W	UART0	00000000 <sub>B</sub>
000021 <sub>H</sub>	SCR0	シリアル制御レジスタ 0	R/W	R/W		00000100 <sub>B</sub>
000022 <sub>H</sub>	SIDR0/ SODR0	シリアル入力レジスタ 0/ シリ アル出力レジスタ 0	R	R		XXXXXXXX <sub>B</sub>
000023 <sub>H</sub>	SSR0	シリアルステータスレジスタ 0	R/W	R/W		00001000 <sub>B</sub>
000024 <sub>H</sub>	SMR1	シリアルモードレジスタ 1	R/W	R/W	UART1	00000000 <sub>B</sub>
000025 <sub>H</sub>	SCR1	シリアル制御レジスタ 1	R/W	R/W		00000100 <sub>B</sub>
000026 <sub>H</sub>	SIDR1/ SODR1	シリアル入力レジスタ 1/ シリ アル出力レジスタ 1	R	R		XXXXXXXX <sub>B</sub>
000027 <sub>H</sub>	SSR1	シリアルステータスレジスタ 1	R/W	R/W		00001000 <sub>B</sub>
000028 <sub>H</sub>	PWCSL1	PWC 状態制御レジスタ CH1 ( 下位 )	R/W	R/W	PWC タイマ ( CH1 )	00000000 <sub>B</sub>
000029 <sub>H</sub>	PWCSH1	PWC 状態制御レジスタ CH1 ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
00002A <sub>H</sub>	PWCH, L1	PWC データバッファレジスタ CH1	-	R/W		XXXXXXXX <sub>B</sub>
00002B <sub>H</sub>						XXXXXXXX <sub>B</sub>
00002C <sub>H</sub>	DIV1	分周比制御レジスタ CH1	R/W	R/W		-----00 <sub>B</sub>
00002D <sub>H</sub> ~ 2F <sub>H</sub>	使用禁止					

表 A-1 I/O マップ (3 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000030 <sub>H</sub>	ENIR	DTP/ 外部割込み許可レジスタ	R/W	R/W	DTP/ 外部割 込み	00000000 <sub>B</sub>
000031 <sub>H</sub>	EIRR	DTP/ 外部割込み要因レジスタ	R/W	R/W		XXXXXXXX <sub>B</sub>
000032 <sub>H</sub>	ELVRL	検出レベル設定レジスタ ( 下位 , 上位 )	R/W	R/W		00000000 <sub>B</sub>
000033 <sub>H</sub>	ELVRH		R/W	R/W		00000000 <sub>B</sub>
000034 <sub>H</sub>	ADCS0	A/D 制御ステータスレジスタ 0 ( 下位 )	R/W	R/W	8/10 ビット A/D コン バータ	00000000 <sub>B</sub>
000035 <sub>H</sub>	ADCS1	A/D 制御ステータスレジスタ 1 ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
000036 <sub>H</sub>	ADCR0	A/D データレジスタ 0	R	R		XXXXXXXX <sub>B</sub>
000037 <sub>H</sub>	ADCR1	A/D データレジスタ 1	R/W	R/W		00101-XX <sub>B</sub>
000038 <sub>H</sub>	PDCRH, L0	PPG0 ダウンカウンタレジス タ ( 上位 , 下位 )	-	R	16 ビット PPG タイマ ( CH0 )	11111111 <sub>B</sub>
000039 <sub>H</sub>						11111111 <sub>B</sub>
00003A <sub>H</sub>	PCSRH, L0	PPG0 同期設定バッファレジ スタ ( 上位 , 下位 )	-	W		XXXXXXXX <sub>B</sub>
00003B <sub>H</sub>						XXXXXXXX <sub>B</sub>
00003C <sub>H</sub>	PDUTH, L0	PPG0 デューティ設定バッ ファレジスタ ( 上位 , 下位 )	-	W		XXXXXXXX <sub>B</sub>
00003D <sub>H</sub>						XXXXXXXX <sub>B</sub>
00003E <sub>H</sub>	PCNTL0	PPG0 状態制御レジスタ ( 下位 )	R/W	R/W		--000000 <sub>B</sub>
00003F <sub>H</sub>	PCNTH0	PPG0 状態制御レジスタ ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
000040 <sub>H</sub>	PDCRH, L1	PPG1 ダウンカウンタレジス タ ( 上位 , 下位 )	-	R	16 ビット PPG タイマ ( CH1 )	11111111 <sub>B</sub>
000041 <sub>H</sub>						11111111 <sub>B</sub>
000042 <sub>H</sub>	PCSRH, L1	PPG1 同期設定バッファレジ スタ ( 上位 , 下位 )	-	W		XXXXXXXX <sub>B</sub>
000043 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000044 <sub>H</sub>	PDUTH, L1	PPG1 デューティ設定バッ ファレジスタ ( 上位 , 下位 )	-	W		XXXXXXXX <sub>B</sub>
000045 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000046 <sub>H</sub>	PCNTL1	PPG1 状態制御レジスタ ( 下位 )	R/W	R/W		--000000 <sub>B</sub>
000047 <sub>H</sub>	PCNTH1	PPG1 状態制御レジスタ ( 上位 )	R/W	R/W		00000000 <sub>B</sub>

表 A-1 I/O マップ (4 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000048 <sub>H</sub>	PDCRH, L2	PPG2 ダウンカウンタレジスタ (上位, 下位)	-	R	16 ビット PPG タイマ ( CH2 )	11111111 <sub>B</sub>
000049 <sub>H</sub>						11111111 <sub>B</sub>
00004A <sub>H</sub>	PCSRH, L2	PPG2 同期設定バッファレジスタ (上位, 下位)	-	W		XXXXXXXX <sub>B</sub>
00004B <sub>H</sub>						XXXXXXXX <sub>B</sub>
00004C <sub>H</sub>	PDUTH, L2	PPG2 デューティ設定バッファレジスタ (上位, 下位)	-	W		XXXXXXXX <sub>B</sub>
00004D <sub>H</sub>						XXXXXXXX <sub>B</sub>
00004E <sub>H</sub>	PCNTL2	PPG2 状態制御レジスタ ( 下位 )	R/W	R/W		--000000 <sub>B</sub>
00004F <sub>H</sub>	PCNTH2	PPG2 状態制御レジスタ ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
000050 <sub>H</sub>	TMRRH, L0	16 ビットデッドタイムレジスタ 0 ( 上位, 下位 )	-	R/W	波形ジェネレータ	XXXXXXXX <sub>B</sub>
000051 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000052 <sub>H</sub>	TMRRH, L1	16 ビットデッドタイムレジスタ 1 ( 上位, 下位 )	-	R/W		XXXXXXXX <sub>B</sub>
000053 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000054 <sub>H</sub>	TMRRH, L2	16 ビットデッドタイムレジスタ 2 ( 上位, 下位 )	-	R/W		XXXXXXXX <sub>B</sub>
000055 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000056 <sub>H</sub>	DTCR0	16 ビットデッドタイム制御レジスタ 0 ( 上位, 下位 )	R/W	R/W		00000000 <sub>B</sub>
000057 <sub>H</sub>	DTCR1	16 ビットデッドタイム制御レジスタ 1 ( 上位, 下位 )	R/W	R/W		00000000 <sub>B</sub>
000058 <sub>H</sub>	DTCR2	16 ビットデッドタイム制御レジスタ 2 ( 上位, 下位 )	R/W	R/W		00000000 <sub>B</sub>
000059 <sub>H</sub>	SIGCR	波形制御レジスタ	R/W	R/W		00000000 <sub>B</sub>
00005A <sub>H</sub>	CPCLRBH, L/ CPCLRH, L	コンペアクリアバッファレジスタ / コンペアクリアレジスタ ( 上位, 下位 )	-	R/W	16 ビットフ リーラン タイマ	11111111 <sub>B</sub>
00005B <sub>H</sub>						11111111 <sub>B</sub>
00005C <sub>H</sub>	TCDTH, L	タイマデータレジスタ ( 上位, 下位 )	-	R/W		00000000 <sub>B</sub>
00005D <sub>H</sub>						00000000 <sub>B</sub>
00005E <sub>H</sub>	TCCSL	タイマ状態制御レジスタ ( 上位, 下位 )	R/W	R/W		X0100000 <sub>B</sub>
00005F <sub>H</sub>	TCCSH		R/W	R/W		00000000 <sub>B</sub>

表 A-1 I/O マップ (5 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000060 <sub>H</sub>	IPCPH, L0	インプットキャプチャデータ レジスタ CH0	-	R	16 ビットイ ンプット キャプチャ ( CH0 ~ CH3 )	XXXXXXXX <sub>B</sub>
000061 <sub>H</sub>		インプットキャプチャデータ レジスタ CH0	-	R		XXXXXXXX <sub>B</sub>
000062 <sub>H</sub>	IPCPH, L1	インプットキャプチャデータ レジスタ CH1	-	R		XXXXXXXX <sub>B</sub>
000063 <sub>H</sub>		インプットキャプチャデータ レジスタ CH1	-	R		XXXXXXXX <sub>B</sub>
000064 <sub>H</sub>	IPCPH, L2	インプットキャプチャデータ レジスタ CH2	-	R		XXXXXXXX <sub>B</sub>
000065 <sub>H</sub>		インプットキャプチャデータ レジスタ CH2	-	R		XXXXXXXX <sub>B</sub>
000066 <sub>H</sub>	IPCPH, L3	インプットキャプチャデータ レジスタ CH3	-	R		XXXXXXXX <sub>B</sub>
000067 <sub>H</sub>		インプットキャプチャデータ レジスタ CH3	-	R		XXXXXXXX <sub>B</sub>
000068 <sub>H</sub>	PICSL01	PPG 出力制御 / インプット キャプチャ状態制御レジスタ ( ch.0, ch.1 )( 下位 )	R/W	R/W		00000000 <sub>B</sub>
000069 <sub>H</sub>	PICSH01	PPG 出力制御 / インプット キャプチャ状態制御レジスタ ( ch.0, ch.1 )( 上位 )	R/W	R/W		00000000 <sub>B</sub>
00006A <sub>H</sub>	ICSL23	インプットキャプチャ状態制 御レジスタ ( ch.2, ch.3 ) ( 下位 )	R/W	R/W		00000000 <sub>B</sub>
00006B <sub>H</sub>	ICSH23	インプットキャプチャ状態制 御レジスタ ( ch.2, ch.3 ) ( 上位 )	R	R		XXXXXX00 <sub>B</sub>
00006C <sub>H</sub> ~ 6E <sub>H</sub>	使用禁止					
00006F <sub>H</sub>	ROMM	ROM ミラー機能選択モ ジュールレジスタ	W	W	ROM ミラー 機能	XXXXXXXX1 <sub>B</sub>

表 A-1 I/O マップ (6 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000070 <sub>H</sub>	OCCPBH, L0/ OCCPH, L0	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 0 (上位, 下位)	-	R/W	16 ビットア ウトプット コンペア (CH0 ~ CH5)	XXXXXXXX <sub>B</sub>
000071 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000072 <sub>H</sub>	OCCPBH, L1/ OCCPH, L1	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 1 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
000073 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000074 <sub>H</sub>	OCCPBH, L2/ OCCPH, L2	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 2 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
000075 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000076 <sub>H</sub>	OCCPBH, L3/ OCCPH, L3	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 3 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
000077 <sub>H</sub>						XXXXXXXX <sub>B</sub>
000078 <sub>H</sub>	OCCPBH, L4/ OCCPH, L4	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 4 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
000079 <sub>H</sub>						XXXXXXXX <sub>B</sub>
00007A <sub>H</sub>	OCCPBH, L5/ OCCPH, L5	アウトプットコンペアバッ ファレジスタ / アウトプット コンペアレジスタ 5 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
00007B <sub>H</sub>						XXXXXXXX <sub>B</sub>
00007C <sub>H</sub>	OCSL0	コンペア制御レジスタ 0 (下位)		R/W		00001100 <sub>B</sub>
00007D <sub>H</sub>	OCSH1	コンペア制御レジスタ 1 (上位)	R/W	R/W		X1100000 <sub>B</sub>
00007E <sub>H</sub>	OCSL2	コンペア制御レジスタ 2 (下位)	R/W	R/W		00001100 <sub>B</sub>
00007F <sub>H</sub>	OCSH3	コンペア制御レジスタ 3 (上位)	R/W	R/W		X1100000 <sub>B</sub>
000080 <sub>H</sub>	OCSL4	コンペア制御レジスタ 4 (下位)	R/W	R/W		00001100 <sub>B</sub>
000081 <sub>H</sub>	OCSH5	コンペア制御レジスタ 5 (上位)	R/W	R/W		X1100000 <sub>B</sub>
000082 <sub>H</sub>	TMCSRL0	タイマ制御状態レジスタ 0 (下位)	R/W	R/W	16 ビットリ ロードタイ マ (CH0)	00000000 <sub>B</sub>
000083 <sub>H</sub>	TMCSRH0	タイマ制御状態レジスタ 0 (上位)	R/W	R/W		----0000 <sub>B</sub>
000084 <sub>H</sub>	TMRH, L0/ TMRDH, L0	16 ビットタイマレジスタ 0/16 ビットリロードレジスタ 0 (上位, 下位)	-	R/W		XXXXXXXX <sub>B</sub>
000085 <sub>H</sub>						XXXXXXXX <sub>B</sub>

表 A-1 I/O マップ (7 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
000086 <sub>H</sub>	TMCSRL1	タイマ制御状態レジスタ 1 ( 下位 )	R/W	R/W	16 ビットリ ロードタイ マ ( CH1 )	00000000 <sub>B</sub>
000087 <sub>H</sub>	TMCSRH1	タイマ制御状態レジスタ 1 ( 上位 )	R/W	R/W		----0000 <sub>B</sub>
000088 <sub>H</sub>	TMRH, L1/ TMRDH, L1	16 ビットタイマレジスタ 1/16 ビットリロードレジスタ 1 ( 上位 , 下位 )	-	R/W		XXXXXXXX <sub>B</sub>
000089 <sub>H</sub>						XXXXXXXX <sub>B</sub>
00008A <sub>H</sub>	OPCLR	出力制御レジスタ ( 下位 )	R/W	R/W	マルチパル スジェネ レータ	00000000 <sub>B</sub>
00008B <sub>H</sub>	OPCUR	出力制御レジスタ ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
00008C <sub>H</sub>	IPCLR	入力制御レジスタ ( 下位 )	R/W	R/W		00000000 <sub>B</sub>
00008D <sub>H</sub>	IPCUR	入力制御レジスタ ( 上位 )	R/W	R/W		00000000 <sub>B</sub>
00008E <sub>H</sub>	TCSR	タイマ状態制御レジスタ	R/W	R/W		00000000 <sub>B</sub>
00008F <sub>H</sub>	NCCR	ノイズキャンセル制御レジス タ	R/W	R/W		00000000 <sub>B</sub>
000090 <sub>H</sub> ～ 9D <sub>H</sub>	使用禁止					
00009E <sub>H</sub>	PACSR	プログラムアドレス検出制御 ステータスレジスタ	R/W	R/W	アドレス一 致検出	00000000 <sub>B</sub>
00009F <sub>H</sub>	DIRR	遅延割込み要求発生 / 解除レ ジスタ	R/W	R/W	遅延割込み 入力発生	XXXXXXXX0 <sub>B</sub>
0000A0 <sub>H</sub>	LPMCR	低消費電力モード制御レジス タ	R/W	R/W	低消費電力 制レジスタ	00011000 <sub>B</sub>
0000A1 <sub>H</sub>	CKSCR	クロック選択レジスタ	R/W	R/W		11111100 <sub>B</sub>
0000A2 <sub>H</sub> ～ A7 <sub>H</sub>	使用禁止					
0000A8 <sub>H</sub>	WDTC	ウォッチドッグタイマ制御レ ジスタ	R/W	R/W	ウォッチ ドットイマ	X-XXX111 <sub>B</sub>
0000A9 <sub>H</sub>	TBTC	タイムベースタイマ制御レジ スタ	R/W	R/W	タイムベ ースタイマ	1--00100 <sub>B</sub>
0000AA <sub>H</sub> ～ AD <sub>H</sub>	使用禁止					
0000AE <sub>H</sub>	FMCS	フラッシュメモリ制御ステー タスレジスタ	R/W	R/W	フラッシュ メモリイン タフェース	000X0000 <sub>B</sub>
0000AF <sub>H</sub>	使用禁止					

表 A-1 I/O マップ (8 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
0000B0 <sub>H</sub>	ICR00	割込み制御レジスタ 00	R/W	R/W	割込み制御	00000111 <sub>B</sub>
0000B1 <sub>H</sub>	ICR01	割込み制御レジスタ 01	R/W	R/W		00000111 <sub>B</sub>
0000B2 <sub>H</sub>	ICR02	割込み制御レジスタ 02	R/W	R/W		00000111 <sub>B</sub>
0000B3 <sub>H</sub>	ICR03	割込み制御レジスタ 03	R/W	R/W		00000111 <sub>B</sub>
0000B4 <sub>H</sub>	ICR04	割込み制御レジスタ 04	R/W	R/W		00000111 <sub>B</sub>
0000B5 <sub>H</sub>	ICR05	割込み制御レジスタ 05	R/W	R/W		00000111 <sub>B</sub>
0000B6 <sub>H</sub>	ICR06	割込み制御レジスタ 06	R/W	R/W		00000111 <sub>B</sub>
0000B7 <sub>H</sub>	ICR07	割込み制御レジスタ 07	R/W	R/W		00000111 <sub>B</sub>
0000B8 <sub>H</sub>	ICR08	割込み制御レジスタ 08	R/W	R/W		00000111 <sub>B</sub>
0000B9 <sub>H</sub>	ICR09	割込み制御レジスタ 09	R/W	R/W		00000111 <sub>B</sub>
0000BA <sub>H</sub>	ICR10	割込み制御レジスタ 10	R/W	R/W		00000111 <sub>B</sub>
0000BB <sub>H</sub>	ICR11	割込み制御レジスタ 11	R/W	R/W		00000111 <sub>B</sub>
0000BC <sub>H</sub>	ICR12	割込み制御レジスタ 12	R/W	R/W		00000111 <sub>B</sub>
0000BD <sub>H</sub>	ICR13	割込み制御レジスタ 13	R/W	R/W		00000111 <sub>B</sub>
0000BE <sub>H</sub>	ICR14	割込み制御レジスタ 14	R/W	R/W		00000111 <sub>B</sub>
0000BF <sub>H</sub>	ICR15	割込み制御レジスタ 15	R/W	R/W	00000111 <sub>B</sub>	
0000C0 <sub>H</sub> ～ FF <sub>H</sub>	外部領域					
001FF0 <sub>H</sub>	PADR0L	プログラムアドレス検出レジスタ 0（下位）	R/W	R/W	アドレス一致検出	XXXXXXXX <sub>B</sub>
001FF1 <sub>H</sub>	PADR0M	プログラムアドレス検出レジスタ 1（中位）	R/W	R/W		XXXXXXXX <sub>B</sub>
001FF2 <sub>H</sub>	PADR0H	プログラムアドレス検出レジスタ 2（上位）	R/W	R/W		XXXXXXXX <sub>B</sub>
001FF3 <sub>H</sub>	PADR1L	プログラムアドレス検出レジスタ 3（下位）	R/W	R/W		XXXXXXXX <sub>B</sub>
001FF4 <sub>H</sub>	PADR1M	プログラムアドレス検出レジスタ 4（中位）	R/W	R/W		XXXXXXXX <sub>B</sub>
001FF5 <sub>H</sub>	PADR1H	プログラムアドレス検出レジスタ 5（上位）	R/W	R/W		XXXXXXXX <sub>B</sub>



表 A-1 I/O マップ (9 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
003FE0 <sub>H</sub>	OPDBR0	出力データバッファレジスタ 0 (下位)	-	R/W	マルチパルス ジェネ レータ	00000000 <sub>B</sub>
003FE1 <sub>H</sub>		出力データバッファレジスタ 0 (上位)	-	R/W		00000000 <sub>B</sub>
003FE2 <sub>H</sub>	OPDBR1	出力データバッファレジスタ 1 (下位)	-	R/W		00000000 <sub>B</sub>
003FE3 <sub>H</sub>		出力データバッファレジスタ 1 (上位)	-	R/W		00000000 <sub>B</sub>
003FE4 <sub>H</sub>	OPDBR2	出力データバッファレジスタ 2 (下位)	-	R/W		00000000 <sub>B</sub>
003FE5 <sub>H</sub>		出力データバッファレジスタ 2 (上位)	-	R/W		00000000 <sub>B</sub>
003FE6 <sub>H</sub>	OPDBR3	出力データバッファレジスタ 3 (下位)	-	R/W		00000000 <sub>B</sub>
003FE7 <sub>H</sub>		出力データバッファレジスタ 3 (上位)	-	R/W		00000000 <sub>B</sub>
003FE8 <sub>H</sub>	OPDBR4	出力データバッファレジスタ 4 (下位)	-	R/W		00000000 <sub>B</sub>
003FE9 <sub>H</sub>		出力データバッファレジスタ 4 (上位)	-	R/W		00000000 <sub>B</sub>
003FEA <sub>H</sub>	OPDBR5	出力データバッファレジスタ 5 (下位)	-	R/W		00000000 <sub>B</sub>
003FEB <sub>H</sub>		出力データバッファレジスタ 5 (上位)	-	R/W		00000000 <sub>B</sub>
003FEC <sub>H</sub>	OPDBR6	出力データバッファレジスタ 6 (下位)	-	R/W		00000000 <sub>B</sub>
003FED <sub>H</sub>		出力データバッファレジスタ 6 (上位)	-	R/W		00000000 <sub>B</sub>
003FEE <sub>H</sub>	OPDBR7	出力データバッファレジスタ 7 (下位)	-	R/W		00000000 <sub>B</sub>
003FEF <sub>H</sub>		出力データバッファレジスタ 7 (上位)	-	R/W		00000000 <sub>B</sub>
003FF0 <sub>H</sub>	OPDBR8	出力データバッファレジスタ 8 (下位)	-	R/W		00000000 <sub>B</sub>
003FF1 <sub>H</sub>		出力データバッファレジスタ 8 (上位)	-	R/W		00000000 <sub>B</sub>
003FF2 <sub>H</sub>	OPDBR9	出力データバッファレジスタ 9 (下位)	-	R/W		00000000 <sub>B</sub>
003FF3 <sub>H</sub>		出力データバッファレジスタ 9 (上位)	-	R/W		00000000 <sub>B</sub>

表 A-1 I/O マップ (10 / 10)

アドレス	レジスタ 略称	レジスタ名	バイト アクセス	ワード アクセス	リソース名	初期値
003FF4 <sub>H</sub>	OPDBRA	出力データバッファレジスタ A（下位）	-	R/W	マルチパルス ジェネ レータ	00000000 <sub>B</sub>
003FF5 <sub>H</sub>		出力データバッファレジスタ A（上位）	-	R/W		00000000 <sub>B</sub>
003FF6 <sub>H</sub>	OPDBRB	出力データバッファレジスタ B（下位）	-	R/W		00000000 <sub>B</sub>
003FF7 <sub>H</sub>		出力データバッファレジスタ B（上位）	-	R/W		00000000 <sub>B</sub>
003FF8 <sub>H</sub>	OPDR	出力データレジスタ（下位）	-	R		XXXXXXXX <sub>B</sub>
003FF9 <sub>H</sub>		出力データレジスタ（上位）	-	R		0000XXXX <sub>B</sub>
003FFA <sub>H</sub>	CPCR	コンペアクリアレジスタ （下位）	-	R/W		XXXXXXXX <sub>B</sub>
003FFB <sub>H</sub>		コンペアクリアレジスタ （上位）	-	R/W		XXXXXXXX <sub>B</sub>
003FFC <sub>H</sub>	TMBR	タイマバッファレジスタ （下位）	-	R		00000000 <sub>B</sub>
003FFD <sub>H</sub>		タイマバッファレジスタ （上位）	-	R		00000000 <sub>B</sub>
003FFE <sub>H</sub> , 003FFF <sub>H</sub>	使用禁止					

● 書込み / 読出しについての説明

R/W: リード / ライト

R: リードオンリ

W: ライトオンリ

● 初期値についての説明

0: 初期値は "0" です。

1: 初期値は "1" です。

X: 初期値は不定です。

● "003FE0<sub>H</sub>" ~ "003FFF<sub>H</sub>" のレジスタ領域については, MOV, io などの I/O アドレス命令はサポートされておりません。

## 付録 B 命令

---

**F<sup>2</sup>MC-16LX に使用している命令について説明します。**

---

- B.1 命令の種類
- B.2 アドレッシング
- B.3 直接アドレッシング
- B.4 間接アドレッシング
- B.5 実行サイクル数
- B.6 実効アドレスフィールド
- B.7 命令一覧表の読み方
- B.8 F<sup>2</sup>MC-16LX 命令一覧表
- B.9 命令マップ

## B.1 命令の種類

---

F<sup>2</sup>MC-16LX には、以下に示す 351 種類の命令があります。

---

### ■ 命令の種類

- 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) 39 命令
- 論理演算命令 (ロングワード) 6 命令
- 符号反転命令 (バイト, ワード) 6 命令
- ノーマライズ命令 (ロングワード) 1 命令
- シフト命令 (バイト, ワード, ロングワード) 18 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tring命令 10 命令

## B.2 アドレッシング

---

F<sup>2</sup>MC-16LX では、命令の実効アドレスフィールドまたは命令コード自体 (インプライド) でアドレス形式が決定されます。命令コード自体でアドレス形式が決定する場合は、使用する命令コードに合わせてアドレスを指定します。命令によっては、数種類のアドレス指定方式から設定できるものがあります。

---

### ■ アドレッシング

F<sup>2</sup>MC-16LX には、以下に示す 23 種類のアドレッシングがあります。

- 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レジスタ間接 (@RWj+ j=0 ~ 3)
- ディスプレースメント付レジスタ間接 (@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)
- ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3)
- ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)
- ベースインデックス付レジスタ間接 (@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

実効アドレスフィールドで指定される、アドレス形式を表 B.2-1 に示します。

表 B.2-1 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 ea 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。	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 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 レジスタ間接	DTB
11	@RW1+disp8				DTB
12	@RW2+disp8				ADB
13	@RW3+disp8				SPB
14	@RW4+disp8				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 レジスタ間接	DTB
19	@RW1+disp16				DTB
1A	@RW2+disp16				ADB
1B	@RW3+disp16				SPB
1C	@RW0+RW7			インデックス付レジスタ間接	DTB
1D	@RW1+RW7			インデックス付レジスタ間接	DTB
1E	@PC+disp16			16 ビットディスプレースメント付 PC 間接	PCB
1F	addr16			直接アドレス	DTB

## B.3 直接アドレッシング

直接アドレッシングでは、オペランド値、レジスタおよびアドレスを直接指定します。

### ■ 直接アドレッシング

● 即値 (#imm)

オペランドの値を直接指定します。(#imm4/#imm8/#imm16/#imm32)。

図 B.3-1 に例を示します。

図 B.3-1 即値 (#imm) 例

MOVW A, #01212H (A にオペランドの値を格納する命令)		
実行前	A	2 2 3 3 : 4 4 5 5
実行後	A	4 4 5 5 : 1 2 1 2 (命令によっては AL AH に転送が行われる)

● レジスタ直接

オペランドとして、直接レジスタを指定します。指定できるレジスタを表 B.3-1 に示します。

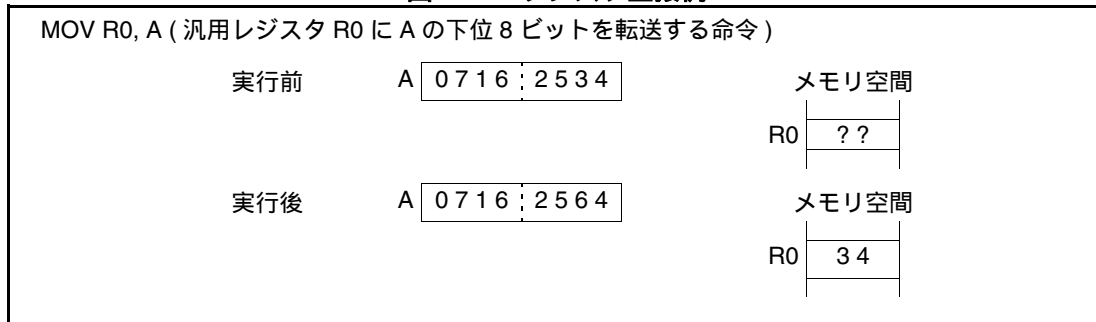
表 B.3-1 レジスタ直接

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

\*: SP は、コンディションコードレジスタ (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) または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は記述されることなく指定されます。

図 B.3-2 に例を示します。

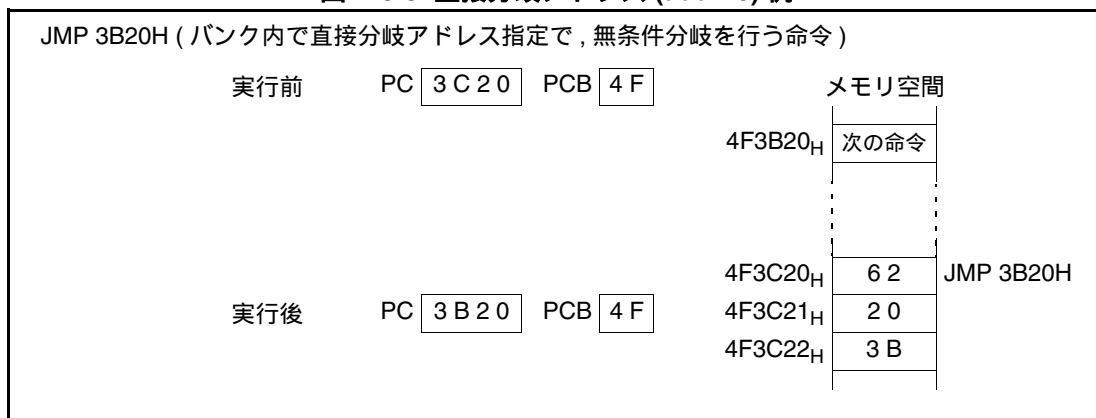
**図 B.3-2 レジスタ直接例**



● 直接分岐アドレス (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データ長は 16 ビットで、論理空間内での分岐先を示します。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されます。図 B.3-3 に例を示します。

**図 B.3-3 直接分岐アドレス (addr16) 例**

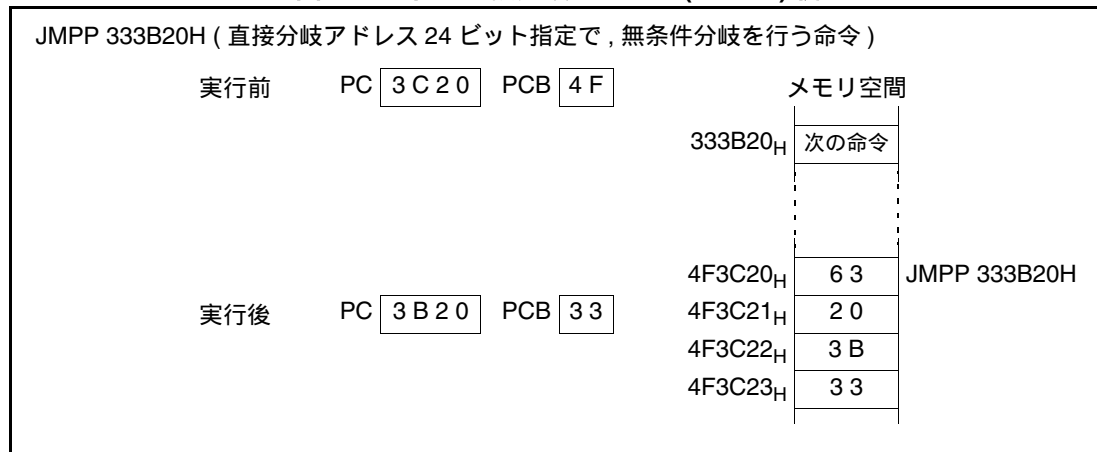




● 物理直接分岐アドレス (addr24)

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データ長は 24 ビットです。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。図 B.3-4 に例を示します。

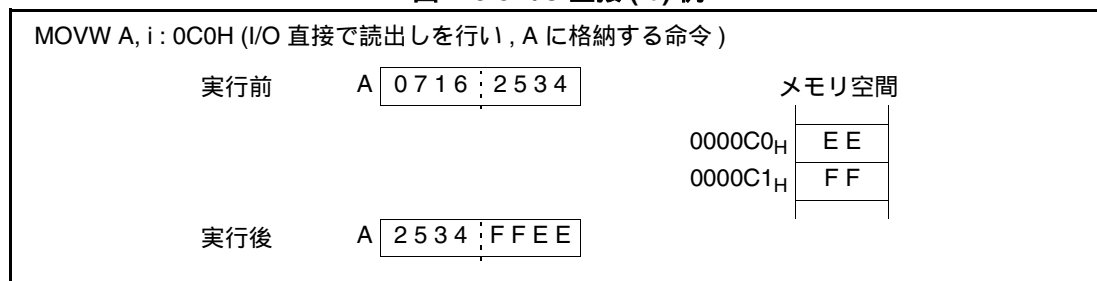
**図 B.3-4 物理直接分岐アドレス (addr24) 例**



- I/O 直接 (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イースメントで直接指定します。データバンクレジスタ (DTB)、ダイレクトページレジスタ (DPR) の値にかかわらず、物理アドレス "000000H" ~ "0000FFH" の空間の I/O 空間がアクセスされます。I/O 直接アドレス指定を使用した命令の前にバンク指定用のバンクセレクトプリフィックスを記述しても無効です。図 B.3-5 に例を示します。

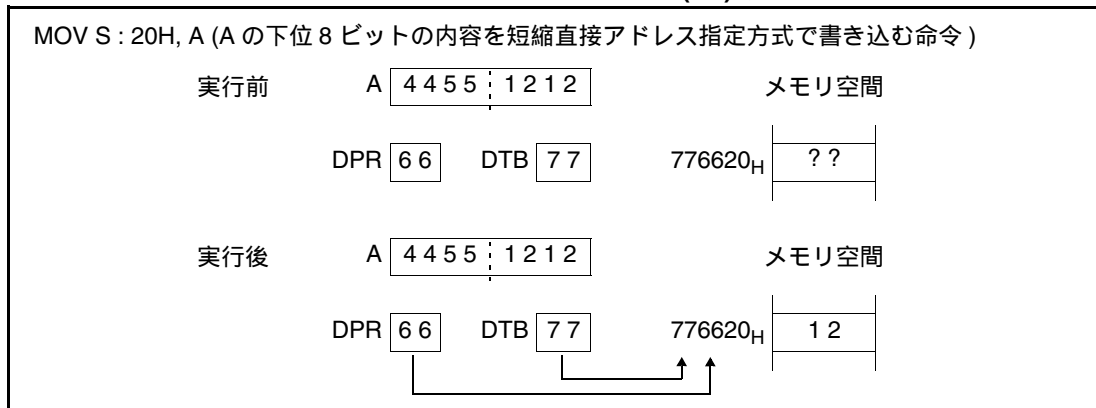
**图 B.3-5 I/O 直接 (io) 例**



● 短縮直接アドレス (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 bit15 ~ bit8 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 bit23 ~ bit16 はデータバンクレジスタ (DTB) により指定されます。図 B.3-6 に例を示します。

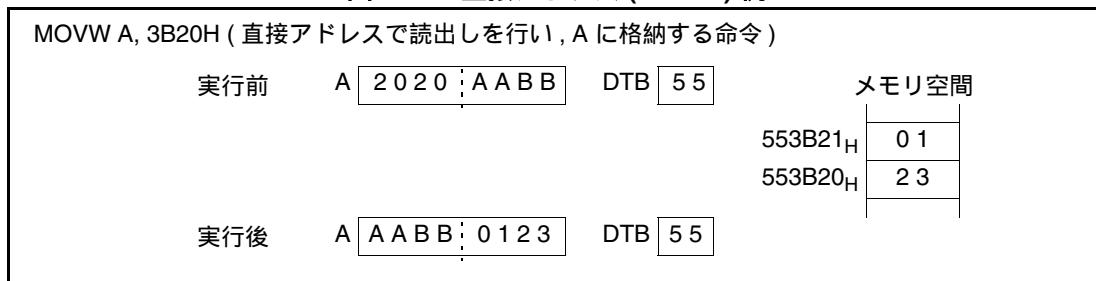
図 B.3-6 短縮直接アドレス (dir) 例



● 直接アドレス (addr16)

オペランドで、メモリアドレス下位 16 ビットを直接指定します。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。直接アドレスのアドレッシングに対しては、アクセス空間指定用のプリフィックス命令は無効です。図 B.3-7 に例を示します。

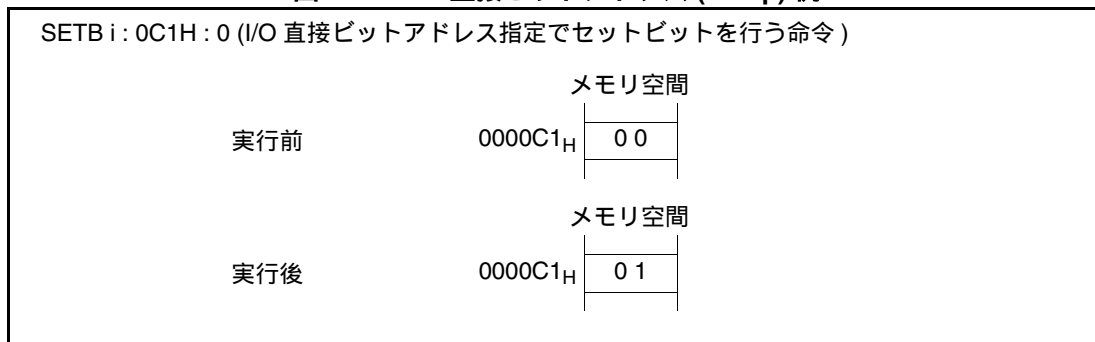
図 B.3-7 直接アドレス (addr16) 例



● I/O 直接ビットアドレス (io: bp)

物理アドレス "000000<sub>H</sub>" ~ "0000FF<sub>H</sub>" 内のビットを直接指定します。ビットの位置は ": bp" で表され, 数字の大きいほうが最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。図 B.3-8 に例を示します。

図 B.3-8 I/O 直接ビットアドレス (io: bp) 例



● 短縮直接ビットアドレス (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 bit15 ~ bit8 はダイレクトページレジスタ (DPR) で指定されます。アドレスの bit23 ~ bit16 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" で表され, 数字の大きい方が最上位ビット, 小さい方が最下位ビットとなります。図 B.3-9 に例を示します。

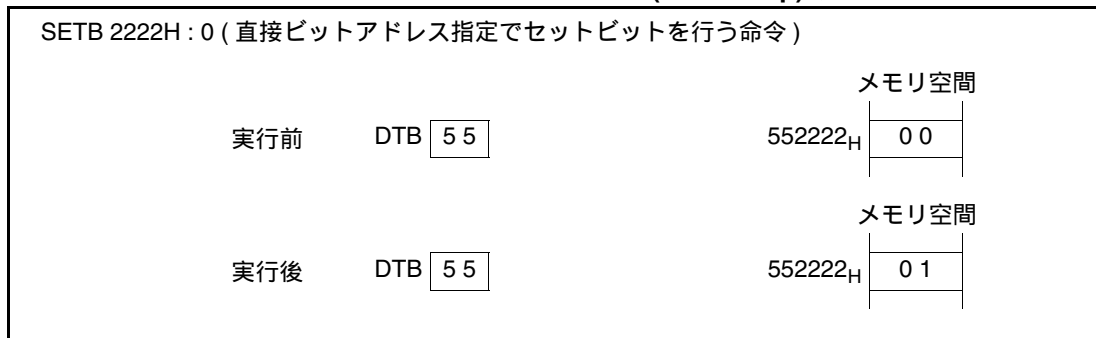
図 B.3-9 短縮直接ビットアドレス (dir: bp) 例



● 直接ビットアドレス (addr16: bp)

64K バイト内の任意のビットに対し直接指定します。アドレスの bit23 ~ bit16 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" で表され、数字の大きい方が最上位ビット、小さい方が最下位ビットとなります。図 B.3-10 に例を示します。

図 B.3-10 直接ビットアドレス (addr16: bp) 例



● ベクタアドレス (#vct)

分岐先のアドレスは指定されたベクタの内容となります。ベクタ番号のデータ長には 4 ビットと 8 ビットの 2 種類があります。サブルーチンコール命令、ソフトウェア割り込み命令に使用します。図 B.3-11 に例を示します。

図 B.3-11 ベクタアドレス (#vct) 例

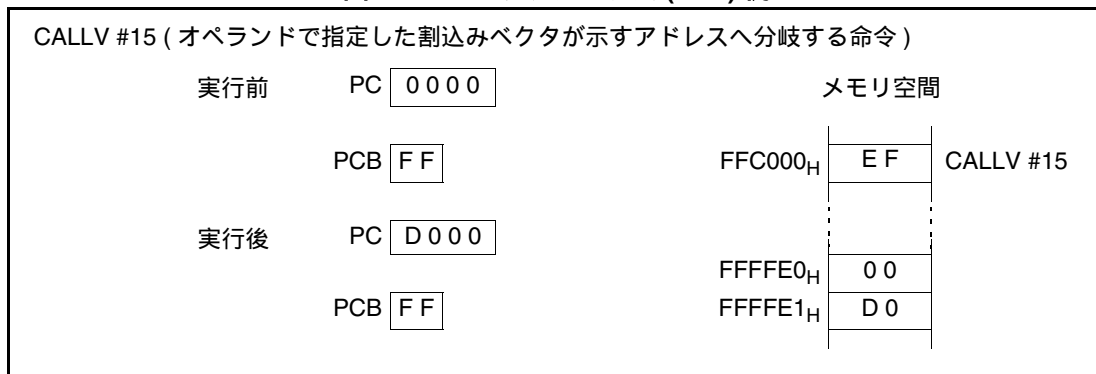


表 B.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE <sub>H</sub>	XXFFFF <sub>H</sub>
CALLV #1	XXFFFC <sub>H</sub>	XXFFFD <sub>H</sub>
CALLV #2	XXFFFA <sub>H</sub>	XXFFFB <sub>H</sub>
CALLV #3	XXFFF8 <sub>H</sub>	XXFFF9 <sub>H</sub>
CALLV #4	XXFFF6 <sub>H</sub>	XXFFF7 <sub>H</sub>
CALLV #5	XXFFF4 <sub>H</sub>	XXFFF5 <sub>H</sub>
CALLV #6	XXFFF2 <sub>H</sub>	XXFFF3 <sub>H</sub>
CALLV #7	XXFFF0 <sub>H</sub>	XXFFF1 <sub>H</sub>
CALLV #8	XXFFEE <sub>H</sub>	XXFFEF <sub>H</sub>
CALLV #9	XXFFEC <sub>H</sub>	XXFFED <sub>H</sub>
CALLV #10	XXFFEA <sub>H</sub>	XXFFEB <sub>H</sub>
CALLV #11	XXFFE8 <sub>H</sub>	XXFFE9 <sub>H</sub>
CALLV #12	XXFFE6 <sub>H</sub>	XXFFE7 <sub>H</sub>
CALLV #13	XXFFE4 <sub>H</sub>	XXFFE5 <sub>H</sub>
CALLV #14	XXFFE2 <sub>H</sub>	XXFFE3 <sub>H</sub>
CALLV #15	XXFFE0 <sub>H</sub>	XXFFE1 <sub>H</sub>

( 注意事項 ) XX には PCB レジスタの値が入ります。

---

< 注意事項 >

プログラムカウンタバンクレジスタ (PCB) が "FF<sub>H</sub>" の場合ベクタ領域は INT #vct8 (#0 ~ #7) のベクタ領域と共有しているため、使用する場合は注意が必要です。( 表 B.3-2 を参照してください )

---

## B.4 間接アドレッシング

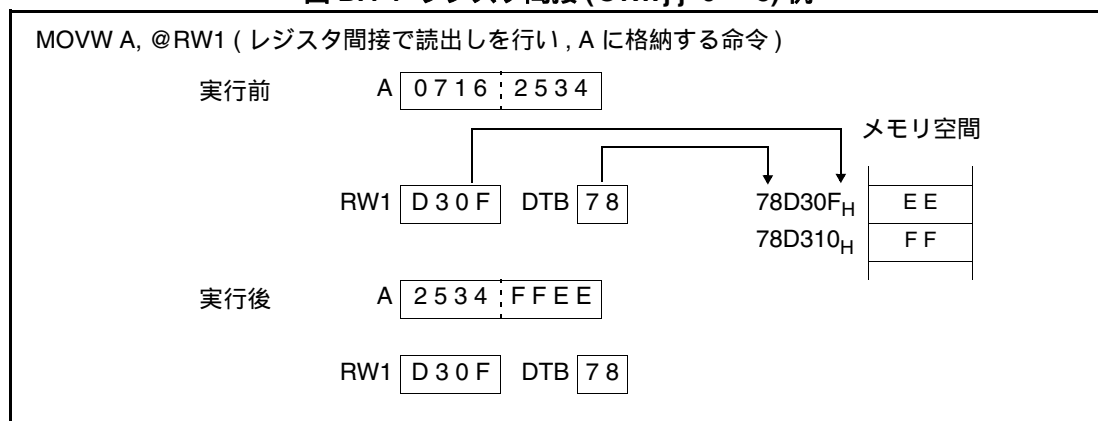
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

### ■ 間接アドレッシング

#### ● レジスタ間接 (@RWj j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の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。図 B.4-1 に例を示します。

図 B.4-1 レジスタ間接 (@RWj j=0 ~ 3) 例



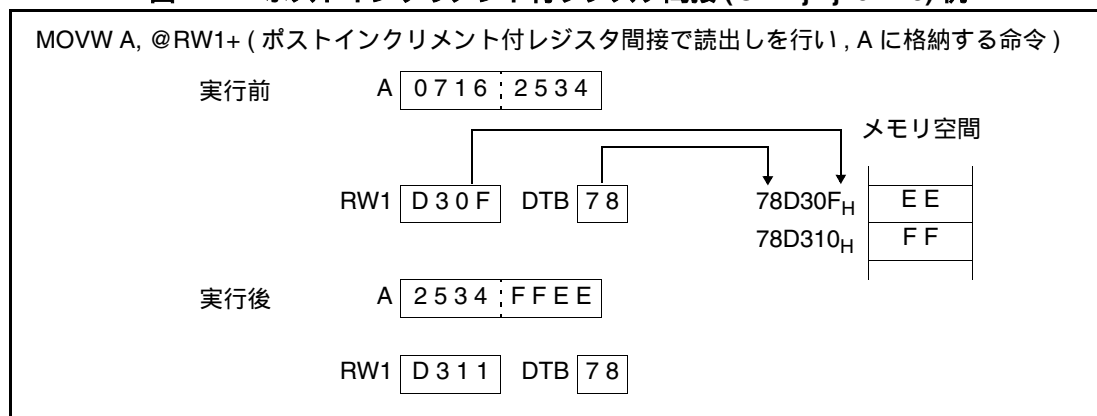
#### ● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。オペランド操作後RWjはオペランドデータ長 (バイトの場合は1, ワードの場合は2, ロングワードの場合は4) 分だけ加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の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指定したレジスタ自身のアドレスだった場合は、参照される値はインクリメントした値となり、命令が書込みだった場合は命令による書込みが優先されるので、インクリメントするはずだったレジスタは書込みデータとなります。

図 B.4-2 に例を示します。

図 B.4-2 ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3) 例

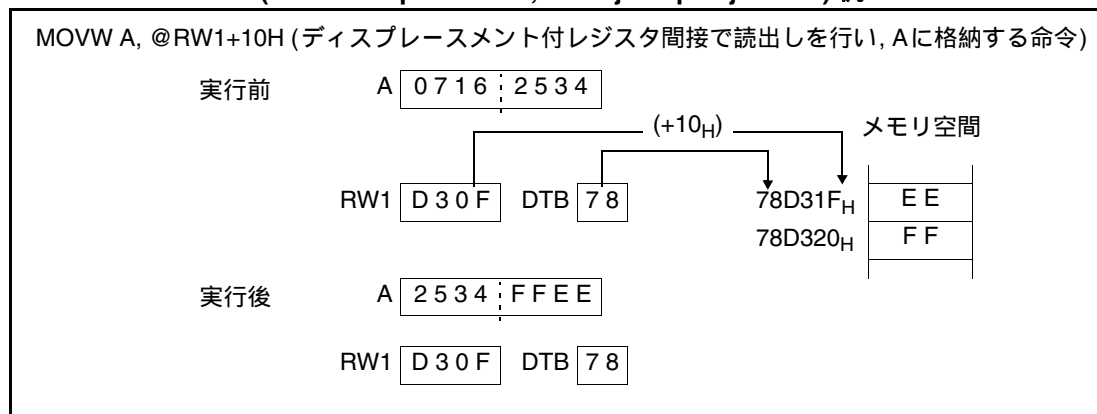


● ディスプレースメント付レジスタ間接

(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)

汎用レジスタ RWj の内容にディスプレースメントを加算したものをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは、バイトとワードの2種類があり、符号付数値として加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1, RW4, RW5 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3, RW7 を用いた場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2, RW6 を用い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。図 B.4-3 に例を示します。

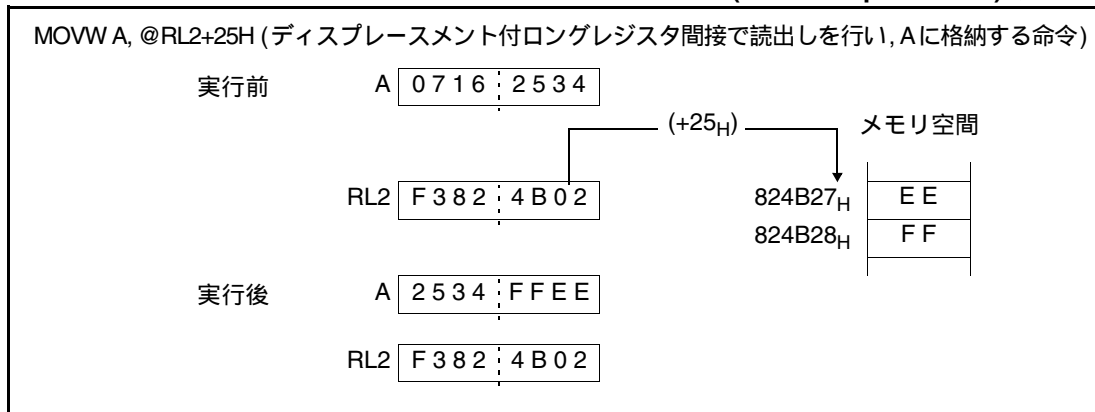
図 B.4-3 ディスプレースメント付レジスタ間接  
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● ディスプレースメント付ロングレジスタ間接 (RLi+disp8 i=0 ~ 3)

汎用レジスタ RLi の内容にディスプレースメントを加算した結果の下位 24 ビットをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは 8 ビットで符号付数値として加算されます。図 B.4-4 に例を示します。

図 B.4-4 ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3) 例



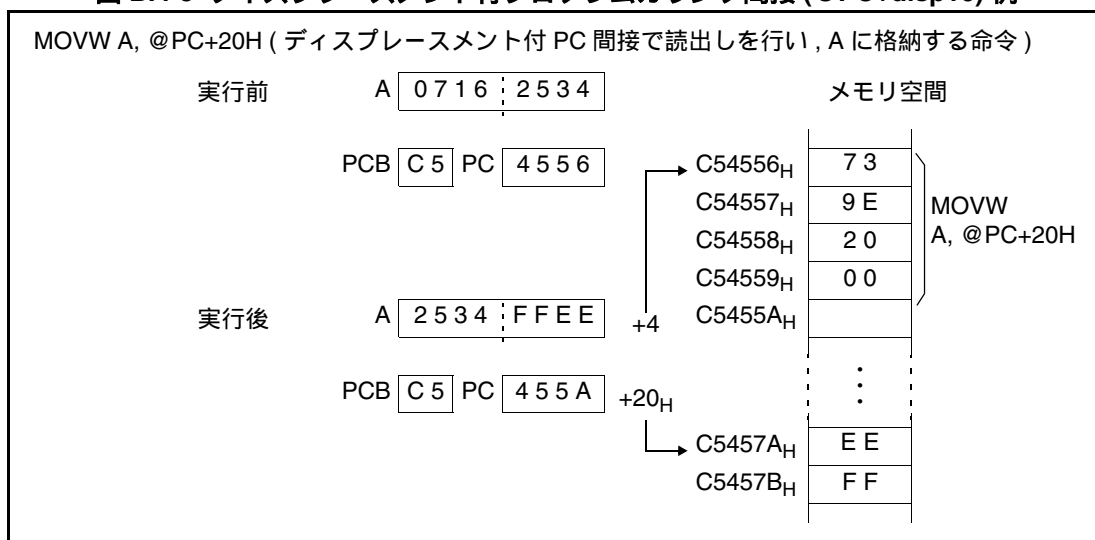
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(命令のアドレス+4+disp16) で示されるアドレスのメモリをアクセスするアドレッシングです。ディスプレースメントはワード長です。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) により指定されます。次に示す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, みなしませんでしたので注意してください。

- DBNZ eam, rel
- CBNE eam, #imm8, rel
- MOV eam, #imm8
- DWBNZ eam, rel
- CWBNE eam, #imm16, rel
- MOVW eam, #imm16

図 B.4-5 に例を示します。

図 B.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例

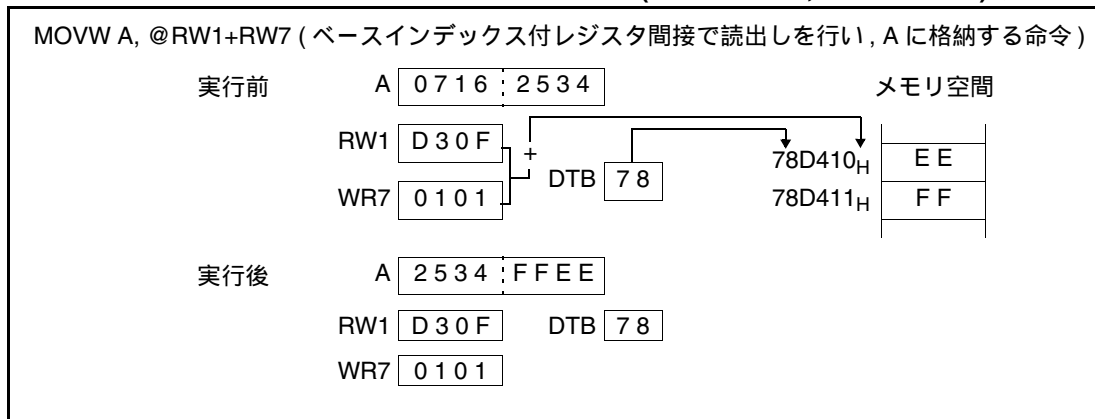




● ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7)

汎用レジスタ RW7 の内容に, RW0 あるいは RW1 を加算したものをアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で示されます。図 B.4-6 に例を示します。

図 B.4-6 ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7) 例

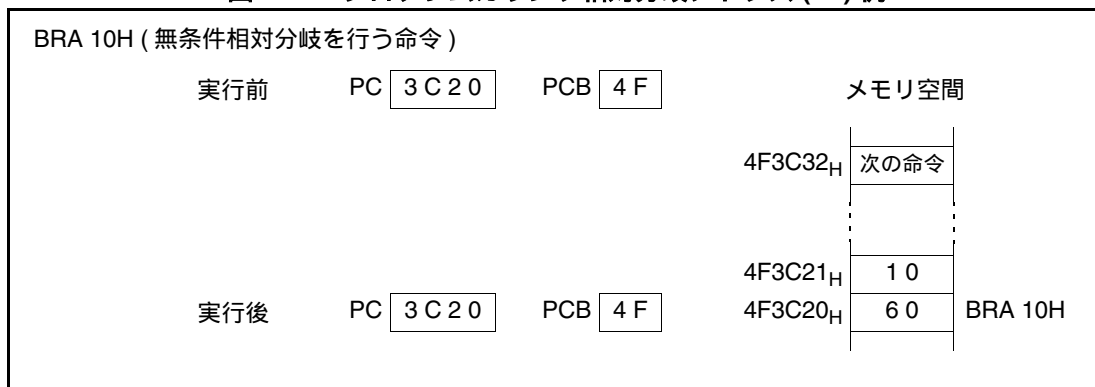


● プログラムカウンタ相対分岐アドレス (rel)

分岐先のアドレスはプログラムカウンタ (PC) の値と 8 ビットのディスプレースメントを加算した値となります。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レメントをせずに, 超えた分は無視されますので, 64K バイトのバンク内で閉じたアドレスとなります。プログラムカウンタ相対分岐アドレスによるアドレッシングは, 無条件 / 条件分岐命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で示されます。

図 B.4-7 に例を示します。

図 B.4-7 プログラムカウンタ相対分岐アドレス (rel) 例



● レジスタリスト (rlst)

スタックに対するプッシュ / ポップの対象となるレジスタを指定します。図 B.4-8 にレジスタリストの構成、図 B.4-9 に例を示します。

図 B.4-8 レジスタリストの構成

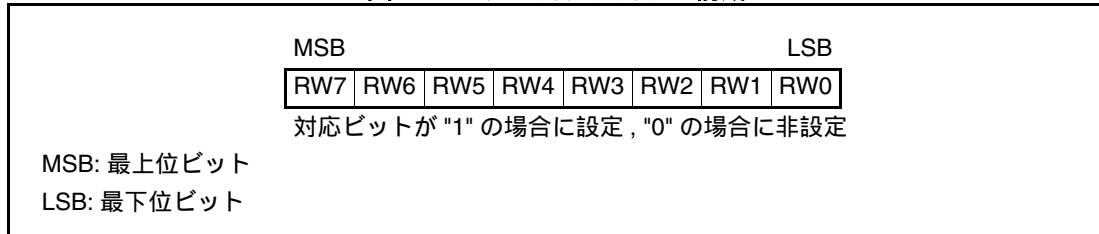
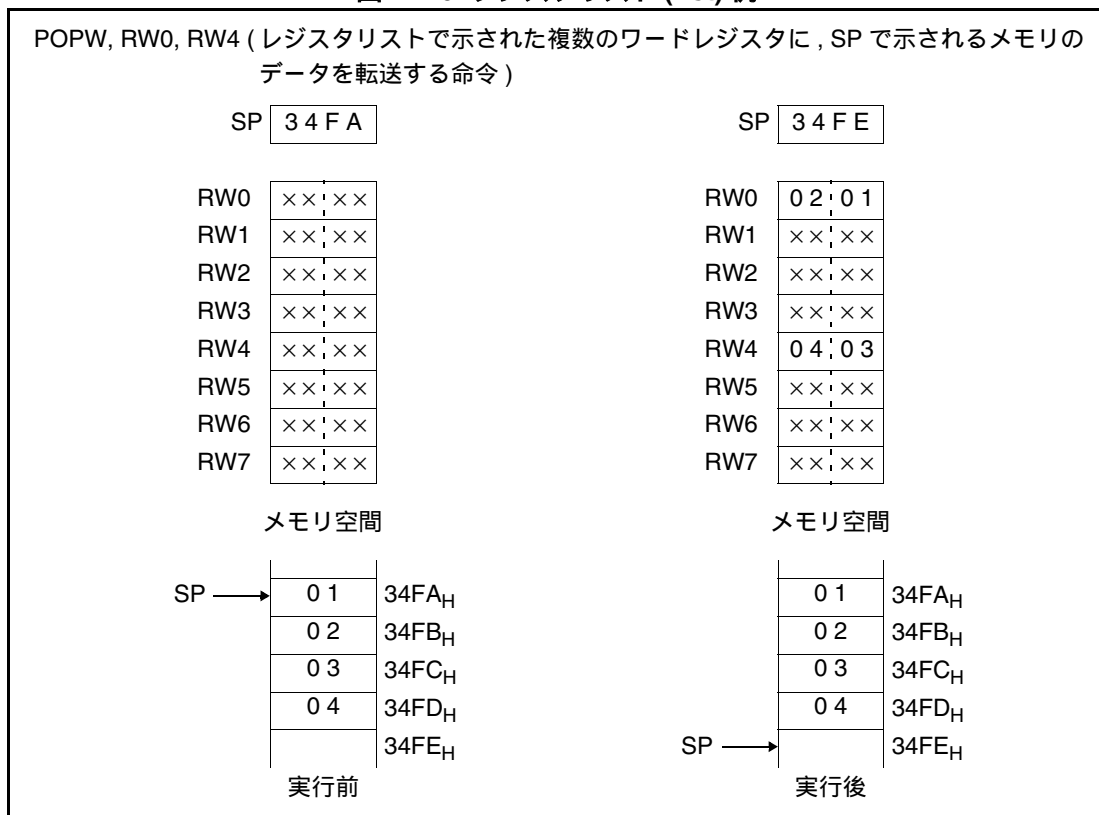


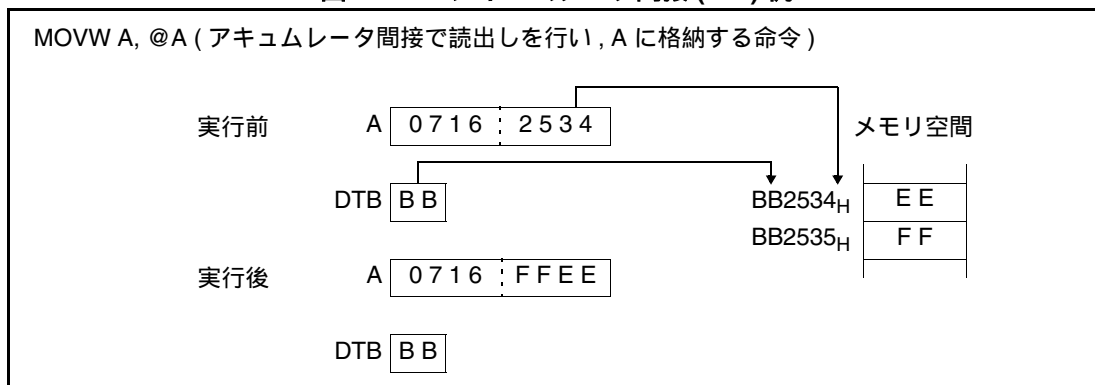
図 B.4-9 レジスタリスト (rlst) 例



### ● アキュムレータ間接 (@A)

アキュムレータの下位バイト (AL) の内容 (16 ビット) で示されるアドレスのメモリをアクセスするアドレス方式です。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) によりニーモニックで指定されます。図 B.4-10 に例を示します。

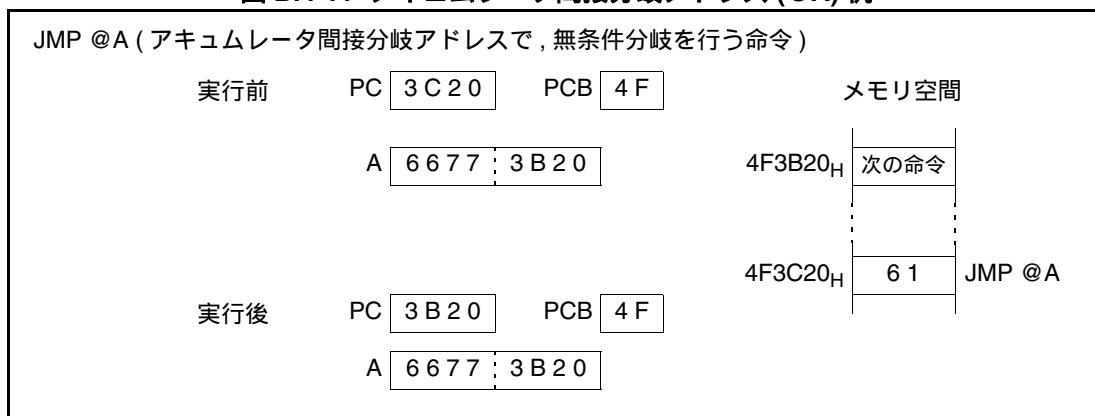
図 B.4-10 アキュムレータ間接 (@A) 例



### ● アキュムレータ間接分岐アドレス (@A)

分岐先のアドレスは, アキュムレータの下位バイト (AL) の内容 (16 ビット) となります。バンク空間内での分岐先を示し, アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されますが, JCTX (Jump Context) 命令の場合は, アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。アキュムレータ間接分岐アドレスによるアドレッシングは, 無条件分岐命令に使用します。図 B.4-11 に例を示します。

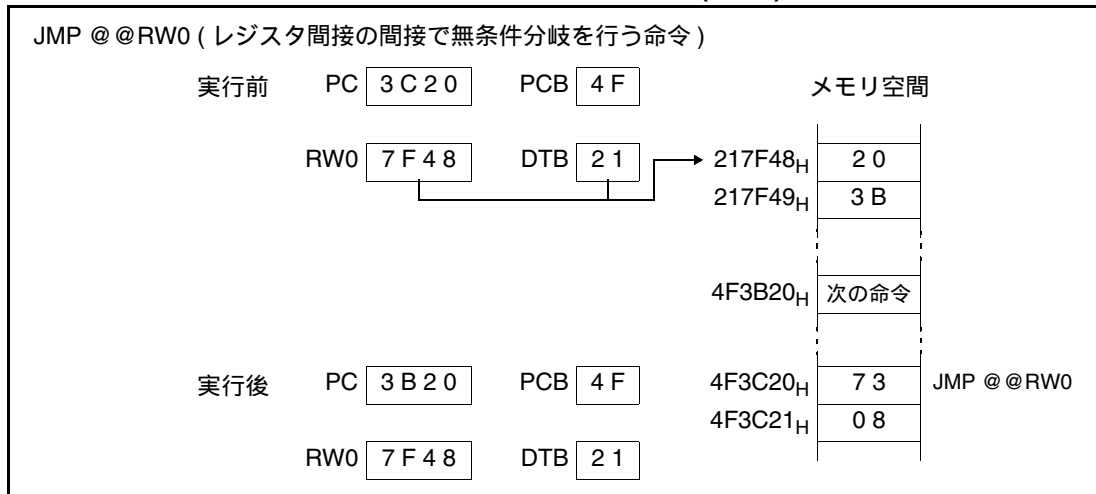
図 B.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 間接指定分岐アドレス (@ear)

ear で示される番地のワードデータが分岐先アドレスとなります。図 B.4-12 に例を示します。

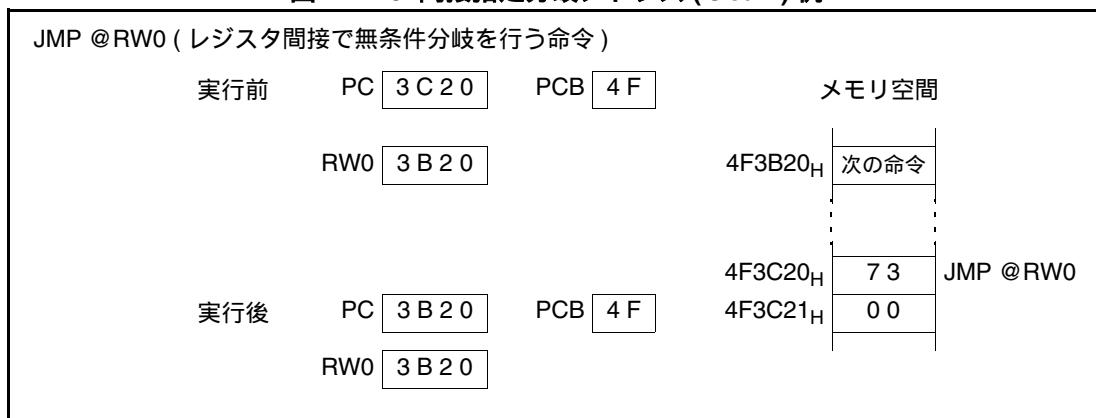
図 B.4-12 間接指定分岐アドレス (@ear) 例



● 間接指定分岐アドレス (@eam)

eam で示される番地のワードデータが分岐先アドレスとなります。図 B.4-13 に例を示します。

図 B.4-13 間接指定分岐アドレス (@eam) 例



## B.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、条件で決まる「補正值」の値およびプログラムフェッチの「サイクル数」の値を加算することで得られます。

### ■ 実行サイクル数

内蔵 ROM などの 16 ビットバスに接続されたメモリ上のプログラムをフェッチする場合には、実行中の命令がワード境界を越えるごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

外部データバスの 8 ビットバスに接続されたメモリ上のプログラムをフェッチする場合は、実行中の命令の 1 バイトごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

CPU 間欠動作時は、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスのアクセスをすると、低消費電力モード制御レジスタの CG0、CG1 ビットで指定されるサイクル数分 CPU へ供給されるクロックが一時停止しますので、CPU 間欠動作の命令の実行に要するサイクル数は、通常の実行サイクル数に、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスの「アクセス回数」×一時停止の「サイクル数」の値を「補正值」として加算してください。

### ■ 実行サイクル数計算方法

表 B.5-1、表 B.5-2、表 B.5-3 に命令実行サイクル数および補正值のデータを示します。

表 B.5-1 各種アドレッシングに対する実行サイクル数

コード	オペランド	(a) *	各種アドレッシングに対するレジスタアクセス回数
		各種アドレッシングに対する実行サイクル数	
00 ~ 07	Ri Rwi RLi	命令一覧表に記載	命令一覧表に記載
08 ~ 0B	@RWj	2	1
0C ~ 0F	@RWj+	4	2
10 ~ 17	@Rwi+disp8	2	1
18 ~ 1B	@Rwi+disp16	2	1
1C	@RW0+RW7	4	2
1D	@RW1+RW7	4	2
1E	@PC+disp16	2	0
1F	addr16	1	0

\*: (a) は「B.8 F2MC-16LX 命令一覧表」の～ (サイクル数)、B (補正值) で使用されています。"～" と "B" の意味については「B.7 命令一覧表の読み方」を参照してください。

表 B.5-2 実サイクル数算出用サイクル数の補正值

オペランド	(b) バイト * <sup>1</sup>		(c) ワード * <sup>1</sup>		(d) ロング * <sup>1</sup>	
	サイクル数	アクセス回数	サイクル数	アクセス回数	サイクル数	アクセス回数
内部レジスタ	+0	1	+0	1	+0	2
内部メモリ 偶数アドレス	+0	1	+0	1	+0	2
内部メモリ 奇数アドレス	+0	1	+2	2	+4	4
外部データバス * <sup>2</sup> 16 ビット偶数アドレス	+1	1	+1	1	+2	2
外部データバス * <sup>2</sup> 16 ビット奇数アドレス	+1	1	+4	2	+8	4
外部データバス * <sup>2</sup> 8 ビット	+1	1	+4	2	+8	4

\*1: (b), (c), (d) は「B.8 F2MC-16LX 命令一覧表」の～(サイクル数), B(補正值)で使用されています。

\*2: 外部データバスを使用した場合は、レディ入力および自動レディによりウェイトしたサイクル数も加算する必要があります。

表 B.5-3 プログラムフェッチサイクル数算出用サイクル数の補正值

命令	バイト境界	ワード境界
内部メモリ	-	+2
外部データバス 16 ビット	-	+3
外部データバス 8 ビット	+3	-

(注意事項) ・ 外部データバスを使用した場合は、レディ入力および自動レディでウェイトしたサイクル数も加算する必要があります。  
 ・ 実際にはすべてのプログラムフェッチで、命令実行が遅くなるわけではないので、この補正值は最悪ケースを算出する場合に使用してください。

## B.6 実効アドレスフィールド

表 B.6-1 に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

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記			アドレス形式	アドレス拡張部の バイト数*
00	R0	RW0	RL0	レジスタ直接 ea 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 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 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				

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記	アドレス形式	アドレス拡張部の バイト数*
18	@RW0+disp16	16 ビットディスプレースメント付 レジスタ間接	2
19	@RW1+disp16		
1A	@RW2+disp16		
1B	@RW3+disp16		
1C	@RW0+RW7	インデックス付レジスタ間接	0
1D	@RW1+RW7	インデックス付レジスタ間接	0
1E	@PC+disp16	16 ビットディスプレースメント付 PC 間接	2
1F	addr16	直接アドレス	2

\*: アドレス拡張部のバイト数は、「B.8 F2MC-16LX 命令一覧表」の#(バイト数)の "+" に当てはまります。  
 "#" の意味については「B.7 命令一覧表の読み方」を参照してください。



## B.7 命令一覧表の読み方

「B.8 F2MC-16LX 命令一覧表」で使用している項目の説明を表 B.7-1 に、記号の説明を表 B.7-2 に示します。

### ■ 命令の表示記号の説明

表 B.7-1 命令一覧表の項目の説明

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表記します。 英小文字：アセンブラ上では、書き替えて記述します。 英小文字の後の数：命令中のビット幅を示します。
#	バイト数を示します。
~	サイクル数を示します。
RG	命令実行時のレジスタアクセス回数を示します。 CPU 間欠動作時の補正値を算出するのに使用します。
B	命令実行時の実サイクル数の算出用補正値を示します。 命令実行時の実サイクルは ~ 欄の数値を加算したものとなります。
オペレーション	命令の動作を示します。
LH	アキュムレータの bit15 ~ bit8 に対する特殊動作を示します。 Z: 0 を転送する。 X: 符号を拡張して転送する。 - : 転送しない。
AH	アキュムレータの上位 16 ビットに対する特殊動作を示します。 *: AL から AH へ転送する。 - : 転送しない。 Z: AH へ 00 <sub>H</sub> を転送する。 X: AL の符号拡張で AH へ 00 <sub>H</sub> または FF <sub>H</sub> を転送する。
I	I ( 割込み許可 ), S ( スタック ), T ( ステッキビット ), N ( ネガティブ ), Z ( ゼロ ), V ( オーバフロー ), C ( キャリー ) の各フラグの状態を示します。 *: 命令の実行で変化する。 - : 変化しない。 S: 命令の実行でセットされる。 R: 命令の実行でリセットされる。
S	
T	
N	
Z	
V	
C	
RMW	
	リードモディファイライト命令 (1 命令でメモリなどからデータを読み出し、メモリへ書き込む) であるかどうかを示します。 *: 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 ) 読み書きで意味の異なるアドレスには使用できません。

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
A	32 ビットアキュムレータ 命令により、使用されるビット長が変わります。 バイト : AL の下位 8 ビット ワード : AL の 16 ビット ロング : AL: 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 or SSP)
PC	プログラムカウンタ
PCB	プログラムカウンタバンクレジスタ
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 or 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
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 <sub>H</sub> ~ 0000FF <sub>H</sub> )
#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

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
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	ビットオフセット値
vct4	ベクタ番号 (0 ~ 15)
vct8	ベクタ番号 (0 ~ 255)
( ) b	ビットアドレス
rel	PC 相対分岐指定
ear	実効アドレス指定 (コード 00 ~ 07)
eam	実効アドレス指定 (コード 08 ~ 1F)
rlst	レジスタ並び

## B.8 F<sup>2</sup>MC-16LX 命令一覧表

F<sup>2</sup>MC-16LX で使用している命令の一覧を示します。

### ■ F<sup>2</sup>MC-16LX 命令一覧表

表 B.8-1 転送系命令 ( バイト ) 41 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A,dir	2	3	0	(b)	byte (A) (dir)	Z	*	-	-	-	*	*	-	-	-
MOV A,addr16	3	4	0	(b)	byte (A) (addr16)	Z	*	-	-	-	*	*	-	-	-
MOV A,Ri	1	2	1	0	byte (A) (Ri)	Z	*	-	-	-	*	*	-	-	-
MOV A,ear	2	2	1	0	byte (A) (ear)	Z	*	-	-	-	*	*	-	-	-
MOV A,eam	2+	3 + (a)	0	(b)	byte (A) (eam)	Z	*	-	-	-	*	*	-	-	-
MOV A,io	2	3	0	(b)	byte (A) (io)	Z	*	-	-	-	*	*	-	-	-
MOV A,#imm8	2	2	0	0	byte (A) imm8	Z	*	-	-	-	*	*	-	-	-
MOV A,@A	2	3	0	(b)	byte (A) ((A))	Z	-	-	-	-	*	*	-	-	-
MOV A,@RLi+disp8	3	10	2	(b)	byte (A) ((RLi)+disp8)	Z	*	-	-	-	*	*	-	-	-
MOVN A,#imm4	1	1	0	0	byte (A) imm4	Z	*	-	-	-	R	*	-	-	-
MOVX A,dir	2	3	0	(b)	byte (A) (dir)	X	*	-	-	-	*	*	-	-	-
MOVX A,addr16	3	4	0	(b)	byte (A) (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX A,Ri	2	2	1	0	byte (A) (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX A,ear	2	2	1	0	byte (A) (ear)	X	*	-	-	-	*	*	-	-	-
MOVX A,eam	2+	3 + (a)	0	(b)	byte (A) (eam)	X	*	-	-	-	*	*	-	-	-
MOVX A,io	2	3	0	(b)	byte (A) (io)	X	*	-	-	-	*	*	-	-	-
MOVX A,#imm8	2	2	0	0	byte (A) imm8	X	*	-	-	-	*	*	-	-	-
MOVX A,@A	2	3	0	(b)	byte (A) ((A))	X	-	-	-	-	*	*	-	-	-
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX A,@RLi+disp8	3	10	2	(b)	byte (A) ((RLi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOV dir,A	2	3	0	(b)	byte (dir) (A)	-	-	-	-	-	*	*	-	-	-
MOV addr16,A	3	4	0	(b)	byte (addr16) (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,A	1	2	1	0	byte (Ri) (A)	-	-	-	-	-	*	*	-	-	-
MOV ear,A	2	2	1	0	byte (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOV eam,A	2+	3 + (a)	0	(b)	byte (eam) (A)	-	-	-	-	-	*	*	-	-	-
MOV io,A	2	3	0	(b)	byte (io) (A)	-	-	-	-	-	*	*	-	-	-
MOV @RLi+disp8,A	3	10	2	(b)	byte ((RLi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,ear	2	3	2	0	byte (Ri) (ear)	-	-	-	-	-	*	*	-	-	-
MOV Ri,eam	2+	4 + (a)	1	(b)	byte (Ri) (eam)	-	-	-	-	-	*	*	-	-	-
MOV ear,Ri	2	4	2	0	byte (ear) (Ri)	-	-	-	-	-	*	*	-	-	-
MOV eam,Ri	2+	5 + (a)	1	(b)	byte (eam) (Ri)	-	-	-	-	-	*	*	-	-	-
MOV Ri,#imm8	2	2	1	0	byte (Ri) imm8	-	-	-	-	-	*	*	-	-	-
MOV io,#imm8	3	5	0	(b)	byte (io) imm8	-	-	-	-	-	-	-	-	-	-
MOV dir,#imm8	3	5	0	(b)	byte (dir) imm8	-	-	-	-	-	-	-	-	-	-
MOV ear,#imm8	3	2	1	0	byte (ear) imm8	-	-	-	-	-	*	*	-	-	-
MOV eam,#imm8	3+	4 + (a)	0	(b)	byte (eam) imm8	-	-	-	-	-	-	-	-	-	-
MOV @AL,AH	2	3	0	(b)	byte ((A)) (AH)	-	-	-	-	-	*	*	-	-	-
XCH A,ear	2	4	2	0	byte (A) (ear)	Z	-	-	-	-	-	-	-	-	-
XCH A,eam	2+	5 + (a)	0	2 × (b)	byte (A) (eam)	Z	-	-	-	-	-	-	-	-	-
XCH Ri,ear	2	7	4	0	byte (Ri) (ear)	-	-	-	-	-	-	-	-	-	-
XCH Ri,eam	2+	9 + (a)	2	2 × (b)	byte (Ri) (eam)	-	-	-	-	-	-	-	-	-	-

( 注意事項 ) 表中の (a), (b) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-2 転送系命令 (ワード・ロングワード) 38 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A,dir	2	3	0	(c)	word (A) (dir)	-	*	-	-	-	*	*	-	-	-
MOVW A,addr16	3	4	0	(c)	word (A) (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW A,SP	1	1	0	0	word (A) (SP)	-	*	-	-	-	*	*	-	-	-
MOVW A,RWi	1	2	1	0	word (A) (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW A,ear	2	2	1	0	word (A) (ear)	-	*	-	-	-	*	*	-	-	-
MOVW A,eam	2+	3 + (a)	0	(c)	word (A) (eam)	-	*	-	-	-	*	*	-	-	-
MOVW A,io	2	3	0	(c)	word (A) (io)	-	*	-	-	-	*	*	-	-	-
MOVW A,@A	2	3	0	(c)	word (A) ((A))	-	-	-	-	-	*	*	-	-	-
MOVW A,#imm16	3	2	0	0	word (A) imm16	-	*	-	-	-	*	*	-	-	-
MOVW A,@RWi+disp8	2	5	1	(c)	word (A) ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW A,@RLi+disp8	3	10	2	(c)	word (A) ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW dir,A	2	3	0	(c)	word (dir) (A)	-	-	-	-	-	*	*	-	-	-
MOVW addr16,A	3	4	0	(c)	word (addr16) (A)	-	-	-	-	-	*	*	-	-	-
MOVW SP,A	1	1	0	0	word (SP) (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,A	1	2	1	0	word (RWi) (A)	-	-	-	-	-	*	*	-	-	-
MOVW ear,A	2	2	1	0	word (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOVW eam,A	2+	3 + (a)	0	(c)	word (eam) (A)	-	-	-	-	-	*	*	-	-	-
MOVW io,A	2	3	0	(c)	word (io) (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RWi+disp8,A	2	5	1	(c)	word ((RWi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RLi+disp8,A	3	10	2	(c)	word ((RLi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,ear	2	3	2	0	word (RWi) (ear)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,eam	2+	4 + (a)	1	(c)	word (RWi) (eam)	-	-	-	-	-	*	*	-	-	-
MOVW ear,RWi	2	4	2	0	word (ear) (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW eam,RWi	2+	5 + (a)	1	(c)	word (eam) (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,#imm16	3	2	1	0	word (RWi) imm16	-	-	-	-	-	*	*	-	-	-
MOVW io,#imm16	4	5	0	(c)	word (io) imm16	-	-	-	-	-	-	-	-	-	-
MOVW ear,#imm16	4	2	1	0	word (ear) imm16	-	-	-	-	-	*	*	-	-	-
MOVW eam,#imm16	4+	4 + (a)	0	(c)	word (eam) imm16	-	-	-	-	-	-	-	-	-	-
MOVW @AL,AH	2	3	0	(c)	word ((A)) (AH)	-	-	-	-	-	*	*	-	-	-
XCHW A,ear	2	4	2	0	word (A) (ear)	-	-	-	-	-	-	-	-	-	-
XCHW A,eam	2+	5 + (a)	0	2 × (c)	word (A) (eam)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, ear	2	7	4	0	word (RWi) (ear)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, eam	2+	9 + (a)	2	2 × (c)	word (RWi) (eam)	-	-	-	-	-	-	-	-	-	-
MOVL A,ear	2	4	2	0	long (A) (ear)	-	-	-	-	-	*	*	-	-	-
MOVL A,eam	2+	5 + (a)	0	(d)	long (A) (eam)	-	-	-	-	-	*	*	-	-	-
MOVL A,#imm32	5	3	0	0	long (A) imm32	-	-	-	-	-	*	*	-	-	-
MOVL ear,A	2	4	2	0	long (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOVL eam,A	2+	5 + (a)	0	(d)	long(eam) (A)	-	-	-	-	-	*	*	-	-	-

( 注意事項 ) 表中の (a), (c), (d) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-3 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	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A,#imm8	2	2	0	0	byte (A) (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD A,dir	2	5	0	(b)	byte (A) (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD A,ear	2	3	1	0	byte (A) (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD A,eam	2+	4 + (a)	0	(b)	byte (A) (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD ear,A	2	3	2	0	byte (ear) (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC A	1	2	0	0	byte (A) (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,ear	2	3	1	0	byte (A) (A) + (ear) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,eam	2+	4 + (a)	0	(b)	byte (A) (A) + (eam) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A	1	3	0	0	byte (A) (AH) + (AL) + (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
SUB A,#imm8	2	2	0	0	byte (A) (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB A,dir	2	5	0	(b)	byte (A) (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB A,ear	2	3	1	0	byte (A) (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB A,eam	2+	4 + (a)	0	(b)	byte (A) (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB ear,A	2	3	2	0	byte (ear) (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC A	1	2	0	0	byte (A) (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,ear	2	3	1	0	byte (A) (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,eam	2+	4 + (a)	0	(b)	byte (A) (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC A	1	3	0	0	byte (A) (AH) - (AL) - (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
ADDW A	1	2	0	0	word (A) (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW A,ear	2	3	1	0	word (A) (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW A,eam	2+	4+(a)	0	(c)	word (A) (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW A,#imm16	3	2	0	0	word (A) (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW ear,A	2	3	2	0	word (ear) (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW A,ear	2	3	1	0	word (A) (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW A,eam	2+	4+(a)	0	(c)	word (A) (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW A	1	2	0	0	word (A) (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW A,ear	2	3	1	0	word (A) (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW A,eam	2+	4+(a)	0	(c)	word (A) (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW A,#imm16	3	2	0	0	word (A) (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW ear,A	2	3	2	0	word (ear) (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW A,ear	2	3	1	0	word (A) (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW A,eam	2+	4+(a)	0	(c)	word (A) (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL A,ear	2	6	2	0	long (A) (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL A,eam	2+	7+(a)	0	(d)	long (A) (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL A,#imm32	5	4	0	0	long (A) (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL A,ear	2	6	2	0	long (A) (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL A,eam	2+	7+(a)	0	(d)	long (A) (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL A,#imm32	5	4	0	0	long (A) (A) - imm32	-	-	-	-	-	*	*	*	*	-

(注意事項) 表中の (a) ~ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-4 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		#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	3	2	0	byte (ear) <-- (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	5+(a)	0	2 × (b)	byte (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	3	2	0	byte (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	5+(a)	0	2 × (b)	byte (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	3	2	0	word (ear) (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	5+(a)	0	2 × (c)	word (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	3	2	0	word (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	5+(a)	0	2 × (c)	word (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	7	4	0	long (ear) (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	9+(a)	0	2 × (d)	long (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	7	4	0	long (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	9+(a)	0	2 × (d)	long (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*

( 注意事項 ) 表中の (a) ～ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-5 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		#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	0	byte (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMP	A,ear	2	2	1	0	byte (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMP	A,eam	2+	3+(a)	0	(b)	byte (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMP	A,#imm8	2	2	0	0	byte (A) - imm8	-	-	-	-	-	*	*	*	*	-
CMPW	A	1	1	0	0	word (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMPW	A,ear	2	2	1	0	word (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPW	A,eam	2+	3+(a)	0	(c)	word (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPW	A,#imm16	3	2	0	0	word (A) - imm16	-	-	-	-	-	*	*	*	*	-
CMPL	A,ear	2	6	2	0	long (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPL	A,eam	2+	7+(a)	0	(d)	long (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPL	A,#imm32	5	3	0	0	long (A) - imm32	-	-	-	-	-	*	*	*	*	-

( 注意事項 ) 表中の (a) ～ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-6 符号なし乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU A	1	*1	0	0	word (AH) / byte (AL) 商 byte (AL) 余り byte (AH)	-	-	-	-	-	-	-	*	*	-
DIVU A,ear	2	*2	1	0	word (A) / byte (ear) 商 byte (A) 余り byte (ear)	-	-	-	-	-	-	-	*	*	-
DIVU A,eam	2+	*3	0	*6	word (A) / byte (eam) 商 byte (A) 余り byte (eam)	-	-	-	-	-	-	-	*	*	-
DIVUW A,ear	2	*4	1	0	long (A) / word (ear) 商 word (A) 余り word (ear)	-	-	-	-	-	-	-	*	*	-
DIVUW A,eam	2+	*5	0	*7	long (A) / word (eam) 商 word (A) 余り word (eam)	-	-	-	-	-	-	-	*	*	-
MULU A	1	*8	0	0	byte (AH) * byte (AL) word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,ear	2	*9	1	0	byte (A) * byte (ear) word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,eam	2+	*10	0	(b)	byte (A) * byte (eam) word (A)	-	-	-	-	-	-	-	-	-	-
MULUW A	1	*11	0	0	word (AH) * word (AL) Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,ear	2	*12	1	0	word (A) * word (ear) Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,eam	2+	*13	0	(c)	word (A) * word (eam) Long (A)	-	-	-	-	-	-	-	-	-	-

\*1: ゼロディバイトの場合, 3    オーバフローの場合, 7    正常の場合, 15  
 \*2: ゼロディバイトの場合, 4    オーバフローの場合, 8    正常の場合, 16  
 \*3: ゼロディバイトの場合, 6+(a)    オーバフローの場合, 9+(a)    正常の場合, 19+(a)  
 \*4: ゼロディバイトの場合, 4    オーバフローの場合, 7    正常の場合, 22  
 \*5: ゼロディバイトの場合, 6+(a)    オーバフローの場合, 8+(a)    正常の場合, 26+(a)  
 \*6: ゼロディバイト, またはオーバフローの場合, (b)    正常の場合, 2 × (b)  
 \*7: ゼロディバイト, またはオーバフローの場合, (c)    正常の場合, 2 × (c)  
 \*8: byte(AH) がゼロの場合, 3    byte(AH) がゼロでない場合, 7  
 \*9: byte(ear) がゼロの場合, 4    byte(ear) がゼロでない場合, 8  
 \*10: byte(eam) がゼロの場合, 5+(a)    byte(eam) がゼロでない場合, 9+(a)  
 \*11: word(AH) がゼロの場合, 3    word(AH) がゼロでない場合, 11  
 \*12: word(ear) がゼロの場合, 4    word(ear) がゼロでない場合, 12  
 \*13: word(eam) がゼロの場合, 5+(a)    word(eam) がゼロでない場合, 13+(a)  
 (注意事項) 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。



表 B.8-7 符号付き乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック		#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	*1	0	0	word (AH) / byte (AL) 商 --> byte (AL) 余り byte (AH)	Z	-	-	-	-	-	-	*	*	-
DIV	A,ear	2	*2	1	0	word (A) / byte (ear) 商 --> byte (A) 余り byte (ear)	Z	-	-	-	-	-	-	*	*	-
DIV	A,eam	2+	*3	0	*6	word (A) / byte (eam) 商 byte (A) 余り byte (eam)	Z	-	-	-	-	-	-	*	*	-
DIVW	A,ear	2	*4	1	0	long (A) / word (ear) 商 --> word (A) 余り word (ear)	-	-	-	-	-	-	-	*	*	-
DIVW	A,eam	2+	*5	0	*7	long (A) / word (eam) 商 word (A) 余り word (eam)	-	-	-	-	-	-	-	*	*	-
MUL	A	2	*8	0	0	byte (AH) * byte (AL) word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A,ear	2	*9	1	0	byte (A) * byte (ear) word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A,eam	2+	*10	0	(b)	byte (A) * byte (eam) word (A)	-	-	-	-	-	-	-	-	-	-
MULW	A	2	*11	0	0	word (AH) * word (AL) Long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A,ear	2	*12	1	0	word (A) * word (ear) Long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A,eam	2+	*13	0	(c)	word (A) * word (eam) Long (A)	-	-	-	-	-	-	-	-	-	-

\*1: ゼロディバイトの場合, 3      オーバフローの場合, 8 or 18      正常の場合, 18

\*2: ゼロディバイトの場合, 4      オーバフローの場合, 11 or 22      正常の場合, 23

\*3: ゼロディバイトの場合, 5+(a)      オーバフローの場合 12+(a) or 23+(a)      正常の場合, 24+(a)

\*4: 被除数が正: ゼロディバイトの場合, 4      オーバフローの場合, 12 or 30      正常の場合, 31

被除数が負: ゼロディバイトの場合, 4      オーバフローの場合, 12 or 31      正常の場合, 32

\*5: 被除数が正: ゼロディバイトの場合, 5+(a)      オーバフローの場合, 12+(a) or 31+(a)      正常の場合, 32+(a)

被除数が負: ゼロディバイトの場合, 5+(a)      オーバフローの場合, 13+(a) or 32+(a)      正常の場合, 33+(a)

\*6: ゼロディバイト, またはオーバフローの場合, (b)      正常の場合, 2 × (b)

\*7: ゼロディバイト, またはオーバフローの場合, (c)      正常の場合, 2 × (c)

\*8: byte(AH) がゼロの場合, 3      結果が正の場合, 12      結果が負の場合, 13

\*9: byte(ear) がゼロの場合, 4      結果が正の場合, 13      結果が負の場合, 14

\*10: byte(eam) がゼロの場合, 5+(a)      結果が正の場合, 14+(a)      結果が負の場合, 15+(a)

\*11: word(AH) がゼロの場合, 3      結果が正の場合, 16      結果が負の場合, 19

\*12: word(ear) がゼロの場合, 4      結果が正の場合, 17      結果が負の場合, 20

\*13: word(eam) がゼロの場合, 5+(a)      結果が正の場合, 18+(a)      結果が負の場合, 21+(a)

(注意事項) ・ DIV, DIVW 命令でオーバフロー発生時の実行サイクル数に 2 種類ありますが, 演算前と演算後に検出する場合があります。

DIV, DIVW 命令でオーバフロー発生時には, AL の内容は破壊されます。

・ 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-8 論理 1 命令 ( バイト・ワード ) 39 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
AND A,#imm8	2	2	0	0	byte (A) (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND A,ear	2	3	1	0	byte (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND A,eam	2+	4+(a)	0	(b)	byte (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND ear,A	2	3	2	0	byte (ear) (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR A,#imm8	2	2	0	0	byte (A) (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR A,ear	2	3	1	0	byte (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR A,eam	2+	4+(a)	0	(b)	byte (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR ear,A	2	3	2	0	byte (ear) (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR A,#imm8	2	2	0	0	byte (A) (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR A,ear	2	3	1	0	byte (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR A,eam	2+	4+(a)	0	(b)	byte (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR ear,A	2	3	2	0	byte (ear) (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT A	1	2	0	0	byte (A) not (A)	-	-	-	-	-	*	*	R	-	-
NOT ear	2	3	2	0	byte (ear) not (ear)	-	-	-	-	-	*	*	R	-	-
NOT eam	2+	5+(a)	0	2 × (b)	byte (eam) not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW A	1	2	0	0	word (A) (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW A,#imm16	3	2	0	0	word (A) (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW A,ear	2	3	1	0	word (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW A,eam	2+	4+(a)	0	(c)	word (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW ear,A	2	3	2	0	word (ear) (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW A	1	2	0	0	word (A) (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW A,#imm16	3	2	0	0	word (A) (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW A,ear	2	3	1	0	word (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW A,eam	2+	4+(a)	0	(c)	word (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW ear,A	2	3	2	0	word (ear) (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW A	1	2	0	0	word (A) (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW A,#imm16	3	2	0	0	word (A) (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW A,ear	2	3	1	0	word (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW A,eam	2+	4+(a)	0	(c)	word (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW ear,A	2	3	2	0	word (ear) (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW A	1	2	0	0	word (A) not (A)	-	-	-	-	-	*	*	R	-	-
NOTW ear	2	3	2	0	word (ear) not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW eam	2+	5+(a)	0	2 × (c)	word (eam) not (eam)	-	-	-	-	-	*	*	R	-	*

( 注意事項 ) 表中の (a), (b), (c) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-9 論理 2 演算命令 (ロングワード) 6 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A,ear	2	6	2	0	long (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL A,eam	2+	7+(a)	0	(d)	long (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL A,ear	2	6	2	0	long (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL A,eam	2+	7+(a)	0	(d)	long (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL A,ear	2	6	2	0	long (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL A,eam	2+	7+(a)	0	(d)	long (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

(注意事項) 表中の (a), (d) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-10 符号反転命令 (バイト・ワード) 6 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
NEG A	1	2	0	0	byte (A) 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG ear	2	3	2	0	byte (ear) 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG eam	2+	5+(a)	0	2 × (b)	byte (eam) 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW A	1	2	0	0	word (A) 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW ear	2	3	2	0	word (ear) 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW eam	2+	5+(a)	0	2 × (c)	word (eam) 0 - (eam)	-	-	-	-	-	*	*	*	*	*

(注意事項) 表中の (a), (b), (c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-11 ノーマライズ命令 (ロングワード) 1 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
NRML A,R0	2	*1	1	0	long (A) 最初に "1" が立っていた所まで左 シフトする byte (R0) その場合のシフト回数	-	-	-	-	-	-	*	-	-	-

\*1: アキュムレータがすべて "0" であった場合は, 4 "0" 以外であった場合は, 6+(R0)

表 B.8-12 シフト命令 ( バイト・ワード・ロングワード ) 18 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	0	byte (A) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLC A	2	2	0	0	byte (A) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC ear	2	3	2	0	byte (ear) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC eam	2+	5+(a)	0	2 × (b)	byte (eam) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ROLC ear	2	3	2	0	byte (ear) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLC eam	2+	5+(a)	0	2 × (b)	byte (eam) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ASR A,R0	2	*1	1	0	byte (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSR A,R0	2	*1	1	0	byte (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSL A,R0	2	*1	1	0	byte (A) ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRW A	1	2	0	0	word (A) 算術右シフト (A,1 ビット)	-	-	-	-	*	*	*	-	*	-
LSRW A/SHRW A	1	2	0	0	word (A) ロジカル右シフト (A,1 ビット)	-	-	-	-	*	R	*	-	*	-
LSLW A/SHLW A	1	2	0	0	word (A) ロジカル左シフト (A,1 ビット)	-	-	-	-	-	*	*	-	*	-
ASRW A,R0	2	*1	1	0	word (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRW A,R0	2	*1	1	0	word (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLW A,R0	2	*1	1	0	word (A) ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRL A,R0	2	*2	1	0	long (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRL A,R0	2	*2	1	0	long (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLL A,R0	2	*2	1	0	long (A) <- ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-

\*1 : R0 が "0" であった場合, 6 "0" 以外であった場合は, 5+(R0)

\*2 : R0 が "0" であった場合, 6 "0" 以外であった場合は, 6+(R0)

( 注意事項 ) 表中の (a),(b) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-13 分岐命令 1 31 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	0	(Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNZ/BNE rel	2	*1	0	0	(Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BC/BLO rel	2	*1	0	0	(C) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNC/BHS rel	2	*1	0	0	(C) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BN rel	2	*1	0	0	(N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BP rel	2	*1	0	0	(N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BV rel	2	*1	0	0	(V) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNV rel	2	*1	0	0	(V) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BT rel	2	*1	0	0	(T) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNT rel	2	*1	0	0	(T) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLT rel	2	*1	0	0	(V) xor (N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGE rel	2	*1	0	0	(V) xor (N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLE rel	2	*1	0	0	((V) xor (N)) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGT rel	2	*1	0	0	((V) xor (N)) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLS rel	2	*1	0	0	(C) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BHI rel	2	*1	0	0	(C) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BRA rel	2	*1	0	0	無条件分岐	-	-	-	-	-	-	-	-	-	-
JMP @A	1	2	0	0	word (PC) (A)	-	-	-	-	-	-	-	-	-	-
JMP addr16	3	3	0	0	word (PC) addr16	-	-	-	-	-	-	-	-	-	-
JMP @ear	2	3	1	0	word (PC) (ear)	-	-	-	-	-	-	-	-	-	-
JMP @eam	2+	4+(a)	0	(c)	word (PC) (eam)	-	-	-	-	-	-	-	-	-	-
JMPP @ear 注 1)	2	5	2	0	word (PC) (ear), (PCB) (ear+2)	-	-	-	-	-	-	-	-	-	-
JMPP @eam 注 1)	2+	6+(a)	0	(d)	word (PC) (eam), (PCB) (eam+2)	-	-	-	-	-	-	-	-	-	-
JMPP addr24	4	4	0	0	word (PC) ad24 0-15, (PCB) ad24 16-23	-	-	-	-	-	-	-	-	-	-
CALL @ear 注 2)	2	6	1	(c)	word (PC) (ear)	-	-	-	-	-	-	-	-	-	-
CALL @eam 注 2)	2+	7+(a)	0	2 × (c)	word (PC) (eam)	-	-	-	-	-	-	-	-	-	-
CALL addr16 注 3)	3	6	0	(c)	word (PC) addr16	-	-	-	-	-	-	-	-	-	-
CALLV #vct4 注 3)	1	7	0	2 × (c)	ベクタコール命令	-	-	-	-	-	-	-	-	-	-
CALLP @ear 注 4)	2	10	2	2 × (c)	word (PC) (ear)0-15, (PCB) (ear)16-23	-	-	-	-	-	-	-	-	-	-
CALLP @eam 注 4)	2+	11+(a)	0	*2	word (PC) (eam)0-15, (PCB) (eam)16-23	-	-	-	-	-	-	-	-	-	-
CALLP addr24 注 5)	4	10	0	2 × (c)	word (PC) addr0-15, (PCB) addr16-23	-	-	-	-	-	-	-	-	-	-

\*1 : 分岐が行われる場合 , 4 行われない場合 , 3

\*2 : 3 × (c)+(b)

注 1) 分岐先アドレスの読み込み (word)

注 2) W : スタックへの退避 (word), R : 分岐先アドレスの読み込み (word)

注 3) スタックへの退避 (word)

注 4) W : スタックへの退避 (long), R : 分岐先アドレスの読み込み (long)

注 5) スタックへの退避 (long)

(注意事項) 表中の (a) ~ (d) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-14 分岐命令 2 19 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A,#imm8,rel	3	*1	0	0	byte (A) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE A,#imm16,rel	4	*1	0	0	word (A) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE ear,#imm8,rel	4	*2	1	0	byte (ear) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE eam,#imm8,rel 注 1)	4+	*3	0	(b)	byte (eam) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE ear,#imm16,rel	5	*4	1	0	word (ear) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE eam,#imm16,rel 注 1)	5+	*3	0	(c)	word (eam) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
DBNZ ear,rel	3	*5	2	0	byte (ear) = (ear) - 1, (ear) 0 で分岐	-	-	-	-	-	*	*	*	-	-
DBNZ eam,rel	3+	*6	2	2 × (b)	byte (eam) = (eam) - 1, (eam) 0 で分岐	-	-	-	-	-	*	*	*	-	*
DWBNZ ear,rel	3	*5	2	0	word (ear) = (ear) - 1, (ear) 0 で分岐	-	-	-	-	-	*	*	*	-	-
DWBNZ eam,rel	3+	*6	2	2 × (c)	word (eam) = (eam) - 1, (eam) 0 で分岐	-	-	-	-	-	*	*	*	-	*
INT #vct8	2	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT addr16	3	16	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INTP addr24	4	17	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT9	1	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
RETI	1	*8	0	*7	割込みからの復帰	-	-	*	*	*	*	*	*	*	-
LINK #imm8	2	6	0	(c)	関数の入口で、旧フレームポインタをスタックに保存し、新フレームポインタの設定、およびローカルポインタの領域を確保する	-	-	-	-	-	-	-	-	-	-
UNLINK	1	5	0	(c)	関数の出口で、旧フレームポインタをスタックから復帰させる	-	-	-	-	-	-	-	-	-	-
RET 注 2)	1	4	0	(c)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-
RETP 注 3)	1	6	0	(d)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-

\*1：分岐が行われる場合，5 行われない場合，4

\*2：分岐が行われる場合，13 行われない場合，12

\*3：分岐が行われる場合，7+(a) 行われない場合，6+(a)

\*4：分岐が行われる場合，8 行われない場合，7

\*5：分岐が行われる場合，7 行われない場合，6

\*6：分岐が行われる場合，8+(a) 行われない場合，7+(a)

\*7：次の割込み要求へ分岐する場合，3 × (b)+2 × (c) 今の割込みから復帰の場合，6 × (c)

\*8：次の割込みへ分岐する場合，15 今の割込みから復帰する場合，17

注 1) CBNE / CWBNE 命令では，RWj+ のアドレッシングモードは，使用しないでください。

注 2) スタックからの復帰 (word)

注 3) スタックからの復帰 (long)

( 注意事項 ) 表中の (a) ~ (d) は，表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-15 その他制御命令 (バイト・ワード・ロングワード) 28 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (A)	-	-	-	-	-	-	-	-	-	-
PUSHW AH	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (AH)	-	-	-	-	-	-	-	-	-	-
PUSHW PS	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (PS)	-	-	-	-	-	-	-	-	-	-
PUSHW rlst	2	*3	*5	*4	(SP) (SP) - 2n, ((SP)) (rlst)	-	-	-	-	-	-	-	-	-	-
POPW A	1	3	0	(c)	word (A) ((SP)), (SP) (SP) + 2	-	*	-	-	-	-	-	-	-	-
POPW AH	1	3	0	(c)	word (AH) ((SP)), (SP) (SP) + 2	-	-	-	-	-	-	-	-	-	-
POPW PS	1	4	0	(c)	word (PS) ((SP)), (SP) (SP) + 2	-	-	*	*	*	*	*	*	*	-
POPW rlst	2	*2	*5	*4	(rlst) ((SP)), (SP) (SP) + 2n	-	-	-	-	-	-	-	-	-	-
JCTX @A	1	14	0	6 × (c)	コンテキストスイッチ命令	-	-	*	*	*	*	*	*	*	-
AND CCR,#imm8	2	3	0	0	byte (CCR) (CCR) and imm8	-	-	*	*	*	*	*	*	*	-
OR CCR,#imm8	2	3	0	0	byte (CCR) (CCR) or imm8	-	-	*	*	*	*	*	*	*	-
MOV RP,#imm8	2	2	0	0	byte (RP) imm8	-	-	-	-	-	-	-	-	-	-
MOV ILM,#imm8	2	2	0	0	byte (ILM) imm8	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,ear	2	3	1	0	word (RWi) ear	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,eam	2+	2+(a)	1	0	word (RWi) eam	-	-	-	-	-	-	-	-	-	-
MOVEA A,ear	2	1	0	0	word (A) ear	-	*	-	-	-	-	-	-	-	-
MOVEA A,eam	2+	1+(a)	0	0	word (A) eam	-	*	-	-	-	-	-	-	-	-
ADDSP #imm8	2	3	0	0	word (SP) (SP) + ext(imm8)	-	-	-	-	-	-	-	-	-	-
ADDSP #imm16	3	3	0	0	word (SP) (SP) + imm16	-	-	-	-	-	-	-	-	-	-
MOV A,brg1	2	*1	0	0	byte (A) (brg1)	Z	*	-	-	-	*	*	-	-	-
MOV brg2,A	2	1	0	0	byte (brg2) (A)	-	-	-	-	-	*	*	-	-	-
NOP	1	1	0	0	ノーオペレーション	-	-	-	-	-	-	-	-	-	-
ADB	1	1	0	0	AD 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
DTB	1	1	0	0	DT 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
PCB	1	1	0	0	PC 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
SPB	1	1	0	0	SP 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
NCC	1	1	0	0	フラグ無変化用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
CMR	1	1	0	0	コモンレジスタバンク用プリフィックス	-	-	-	-	-	-	-	-	-	-

\*1 : PCB,ADB,SSB,USB . . . . . 1 ステート

DTB,DPR . . . . . 2 ステート

\*2 : 7+3 × (POP する回数)+2 × (POP する最後のレジスタ番号), RLST=0(転送レジスタ無し)の場合 7

\*3 : 29+3 × (PUSH する回数)-3 × (PUSH する最後のレジスタ番号), RLST=0(転送レジスタ無し)の場合 8

\*4 : (POP する回数) × (c), または (PUSH する回数) × (c)

\*5 : (POP する回数), または (PUSH する回数)

(注意事項) 表中の (a),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-16 ビット操作命令 21 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB A,dir:bp	3	5	0	(b)	byte (A) (dir:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,addr16:bp	4	5	0	(b)	byte (A) (addr16:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,io:bp	3	4	0	(b)	byte (A) (io:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB dir:bp,A	3	7	0	2×(b)	bit (dir:bp)b (A)	-	-	-	-	-	*	*	-	-	*
MOVB addr16:bp,A	4	7	0	2×(b)	bit (addr16:bp)b (A)	-	-	-	-	-	*	*	-	-	*
MOVB io:bp,A	3	6	0	2×(b)	bit (io:bp)b (A)	-	-	-	-	-	*	*	-	-	*
SETB dir:bp	3	7	0	2×(b)	bit (dir:bp)b 1	-	-	-	-	-	-	-	-	-	*
SETB addr16:bp	4	7	0	2×(b)	bit (addr16:bp)b 1	-	-	-	-	-	-	-	-	-	*
SETB io:bp	3	7	0	2×(b)	bit (io:bp)b 1	-	-	-	-	-	-	-	-	-	*
CLRB dir:bp	3	7	0	2×(b)	bit (dir:bp)b 0	-	-	-	-	-	-	-	-	-	*
CLRB addr16:bp	4	7	0	2×(b)	bit (addr16:bp)b 0	-	-	-	-	-	-	-	-	-	*
CLRB io:bp	3	7	0	2×(b)	bit (io:bp)b 0	-	-	-	-	-	-	-	-	-	*
BBC dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC io:bp,rel	4	*2	0	(b)	(io:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS io:bp,rel	4	*2	0	(b)	(io:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
SBBS addr16:bp,rel	5	*3	0	2×(b)	(addr16:bp) b = 1 で分岐, bit (addr16:bp) b 1	-	-	-	-	-	-	*	-	-	*
WBTS io:bp	3	*4	0	*5	(io:bp) b = 1 になるまで待つ	-	-	-	-	-	-	-	-	-	-
WBTC io:bp	3	*4	0	*5	(io:bp) b = 0 になるまで待つ	-	-	-	-	-	-	-	-	-	-

\*1：分岐が行われる場合，8 分岐が行われない場合，7

\*2：分岐が行われる場合，7 分岐が行われない場合，6

\*3：条件成立の場合，10 未成立の場合，9

\*4：不定回数

\*5：条件が成立するまで

(注意事項) 表中の (b) は，表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-17 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	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	0	byte (A)0-7 (A)8-15	-	-	-	-	-	-	-	-	-	-
SWAPW	1	2	0	0	word (AH) (AL)	-	*	-	-	-	-	-	-	-	-
EXT	1	1	0	0	byte 符号拡張	X	-	-	-	-	*	*	-	-	-
EXTW	1	2	0	0	word 符号拡張	-	X	-	-	-	*	*	-	-	-
ZEXT	1	1	0	0	byte ゼロ拡張	Z	-	-	-	-	R	*	-	-	-
ZEXTW	1	1	0	0	word ゼロ拡張	-	Z	-	-	-	R	*	-	-	-



表 B.8-18 スtring命令 10 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS / MOVSI	2	*2	*5	*3	byte 転送 @AH+ @AL+, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
MOVSD	2	*2	*5	*3	byte 転送 @AH- @AL-, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
SCEQ / SCEQI	2	*1	*8	*4	byte 検索 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
SCEQD	2	*1	*8	*4	byte 検索 @AH- AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
FILS / FILSI	2	6m+6	*8	*3	byte 充填 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	-	-	-
MOVSW / MOVSWI	2	*2	*5	*6	word 転送 @AH+ @AL+, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
MOVSWD	2	*2	*5	*6	word 転送 @AH- @AL-, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
SCWEQ / SCWEQI	2	*1	*8	*7	word 検索 @AH+ - AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
SCWEQD	2	*1	*8	*7	word 検索 @AH- - AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
FILSW / FILSWI	2	6m+6	*8	*6	word 充填 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	-	-	-

\*1 : RW0 が "0" の場合, 5 カウントアウトの場合,  $4+7 \times (RW0)$  一致した場合,  $7n+5$

\*2 : RW0 が "0" の場合, 5 "0" 以外の場合,  $4+8 \times (RW0)$

\*3 : (b)  $\times (RW0)$  + (b)  $\times (RW0)$  ソースとデスティネーションで異なる領域をアクセスする場合には, それぞれ別々に (b) の項目を算出してください。

\*4 : (b) + n

\*5 :  $2 \times (b) \times (RW0)$

\*6 : (c)  $\times (RW0)$  + (c)  $\times (RW0)$  ソースとデスティネーションで異なる領域をアクセスする場合には, それぞれ別々に (c) の項目を算出してください。

\*7 : (c)  $\times n$

\*8 : (b)  $\times (RW0)$

(注意事項) ・ m : RW0 値 (カウンタ値)

n : ループした回数

・ 表中の (b), (c) は, 表 B.5-1 および表 B.5-2 を参照してください。

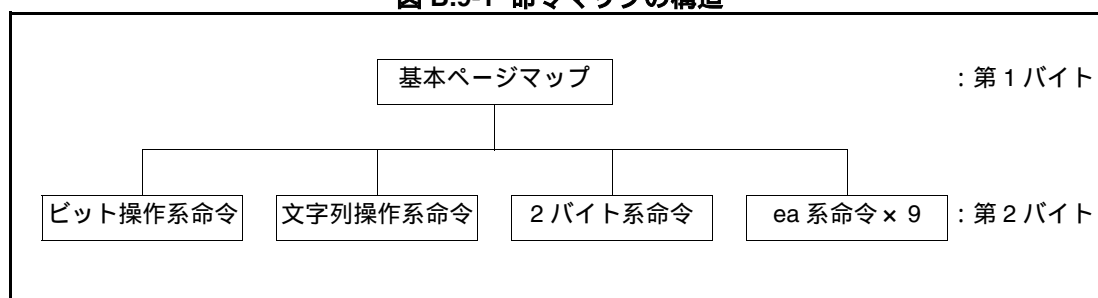
## B.9 命令マップ

F<sup>2</sup>MC-16LX の命令コードは、1 ～ 2 バイトで構成されていますので、命令マップは複数のページで構成されています。

F<sup>2</sup>MC-16LX の命令マップを示します。

### ■ 命令マップの構造

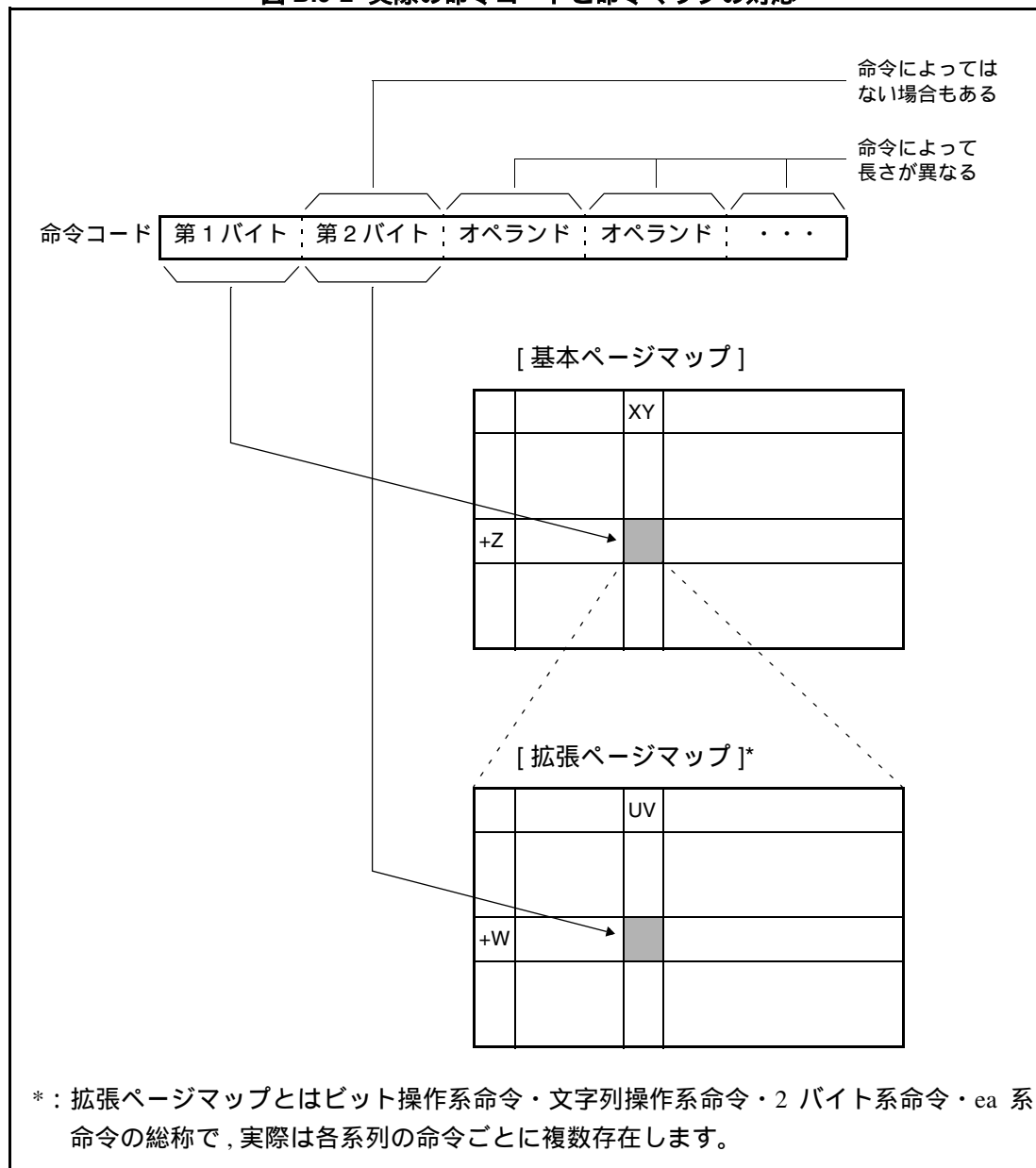
図 B.9-1 命令マップの構造



1 バイトで終わる命令 (NOP 命令など) は基本ページで完結し、2 バイト必要とする命令 (MOVS 命令など) は第 1 バイトを参照した段階で第 2 バイト用のマップの存在がわかり、第 2 バイト用のマップを参照して続く 1 バイトを調べることができます。

実際の命令コードと命令マップの対応を図 B.9-2 に示します。

図 B.9-2 実際の命令コードと命令マップの対応



命令コードの例を表 B.9-1 に示します。

表 B.9-1 命令コードの例

命令	第1バイト (基本ページマップより)	第2バイト (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, #8, rel	$70 + 0 = 70$	$F0 + 2 = F2$

表 B.9-2 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 系命令 その 1	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BZ/BEQ rel
+1	INT9	NCC	SUB A, dir	SUB A, #8	MOV dir, A	MOV io, A	JMP @A	ea 系命令 その 2	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNZ/BNL rel
+2	ADDC A	SUBDC A	ADDC A	SUBC A	MOV A, #8	MOV A, addr16	JMP addr16	ea 系命令 その 3	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BC/BLO rel
+3	NEG A	JCTX @A	CMP A	CMP A, #8	MOVX A, #8	MOV addr16, A	JMPP addr24	ea 系命令 その 4	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNC/BHS rel
+4	PCB	EXT	AND CCR, #8	AND A, #8	MOV dir, #8	MOV io, #8	CALL addr16	ea 系命令 その 5	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BN rel
+5	DTB	ZEXT	OR CCR, #8	OR A, #8	MOVX A, dir	MOVX A, io	CALLP addr24	ea 系命令 その 6	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BP rel
+6	ADB	SWAP	DIVU A	XOR A, #8	MOVW A, SP	MOVW io, #16	RETP	ea 系命令 その 7	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BV rel
+7	SPB	ADDSP #8	MULU #8	NOT A	MOVW SP, A	MOVX A, addr16	RET	ea 系命令 その 8	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNV rel
+8	LINK #imm8	ADDL A, #32	ADDW A	ADDW A, #16	MOVW A, dir	MOVW A, io	INT #vct8	ea 系命令 その 9	MOVW A, RWI	MOVW RWI, A	MOVW RWI, #16	MOV A, @RWI+d8	MOVW @RWI+d8 @RWI+d8,A	MOV A, #4	CALL #4	BT rel
+9	UNLINK	SUBL A, #32	SUBW A	SUBW A, #16	MOVW dir, A	MOVW io, A	INT	MOVEA RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNT rel
+A	MOV RP, #8	MOV ILM, #8	CBNE A, #8, rel	CWBNL A, #16, rel	MOVW A, #16	MOVW A, addr16	INTP	MOV Ri, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLT rel
+B	NEGW A	CMPL A, #32	CMPW A	CMPW A, #16	MOVL A, #32	MOVW addr16, A	RETI	MOVW RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BGE rel
+C	LSLW A	EXTW	ANDW A	ANDW A, #16	PUSHW A	POPW A	ビット操 作系命令	MOV ea, Ri	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, RWI	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BGT rel
+E	ASRW A	SWAPW A	XORW A	XORW A, #16	PUSHW PS	POPW PS	文字列操 作系命令	XCH Ri, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLS rel
+F	LSRW A	ADDSP #16	MULW A	NOTW A	PUSHW r1st	POPW r1st	2 バイト 系命令	XCHW RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BHI rel

表 B.9-3 ビット操作系命令マップ (第1バイト = 6C<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVB A, io:bp		MOVB io:bp, A		CLRB io:bp		SETB io:bp		BBC io:bp, rel		BBS io:bp, rel		WBTS io:bp		WBTC io:bp	
+1																
+2																
+3																
+4																
+5																
+6																
+7																
+8	MOVB A, dir:bp	MOVB A, addr16:bp	MOVB dir:bp, A	MOVB addr16:bp, A	CLRB dir:bp	CLRB addr16:bp	SETB dir:bp	SETB addr16:bp	BBC dir:bp, rel	BBC addr16:bp, rel	BBS dir:bp, rel	BBS addr16:bp, rel				SBBS addr16:bp
+9																
+A																
+B																
+C																
+D																
+E																
+F																

表 B.9-4 文字列操作系命令マップ (第 1 バイト = 6EH)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVSI, MOVSD, MOVSWI, MOVSWD								SCWEQI, SCEQD, PCB	SCWEQI, SCEQD, PCB	SCWEQI, SCEQD, PCB	SCWEQD, FILSI, PCB	FILSI, PCB			
+1	PCB, DTB								PCB, DTB	PCB, DTB	PCB, DTB	PCB, DTB	PCB, DTB		DTB	
+2	PCB, ADB								PCB, ADB	PCB, ADB	PCB, ADB	PCB, ADB	PCB, ADB		ADB	
+3	PCB, SPB								PCB, SPB	PCB, SPB	PCB, SPB	PCB, SPB	PCB, SPB		SPB	
+4	DTB, PCB															
+5	DTB, DTB															
+6	DTB, ADB															
+7	DTB, SPB															
+8	ADB, PCB															
+9	ADB, DTB															
+A	ADB, ADB															
+B	ADB, SPB															
+C	SPB, PCB															
+D	SPB, DTB															
+E	SPB, ADB															
+F	SPB, SPB															

表 B.9-5 2 バイト系命令マップ (第 1 バイト =6F<sub>H</sub>)

	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 @RL0+d8, A											
+1	MOV A, ADB	MOV ADB, A														
+2	MOV A, SSB	MOV SSB, A	MOVX A, @RL1+d8	MOV @RL1+d8, A	MOV @RL1+d8, A											
+3	MOV A, USB	MOV USB, A														
+4	MOV A, DPR	MOV DPR, A	MOVX A, @RL2+d8	MOV @RL2+d8, A	MOV @RL2+d8, A											
+5	MOV A, @A	MOV @AL, AH														
+6	MOV A, PCB	MOV A, @A	MOVX A, @RL3+d8	MOV @RL3+d8, A	MOV @RL3+d8, A											
+7	ROL A	ROL A														
+8				MOVW @RL0+d8, A	MOVW A, @RL0+d8			MUL A								
+9								MULW A								
+A				MOVW @RL1+d8, A	MOVW A, @RL1+d8			DIVU 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													

表 B.9-6 ea 系命令 その 1 (第 1 バイト = 70<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
					CBNE↓	CWNE↓									CBNE↓	
+0	ADDL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	RW0, @RW0+d8 #16, rel	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	R0, @RW0+d8 #8, rel	R0, @RW0+d8 #8, rel
+1	ADDL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	RW1, @RW1+d8 #16, rel	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	R1, @RW1+d8 #8, rel	R1, @RW1+d8 #8, rel
+2	ADDL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	RW2, @RW2+d8 #16, rel	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	R2, @RW2+d8 #8, rel	R2, @RW2+d8 #8, rel
+3	ADDL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	RW3, @RW3+d8 #16, rel	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	R3, @RW3+d8 #8, rel	R3, @RW3+d8 #8, rel
+4	ADDL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	RW4, @RW4+d8 #16, rel	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	R4, @RW4+d8 #8, rel	R4, @RW4+d8 #8, rel
+5	ADDL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	RW5, @RW5+d8 #16, rel	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	R5, @RW5+d8 #8, rel	R5, @RW5+d8 #8, rel
+6	ADDL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	RW6, @RW6+d8 #16, rel	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	R6, @RW6+d8 #8, rel	R6, @RW6+d8 #8, rel
+7	ADDL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	RW7, @RW7+d8 #16, rel	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	R7, @RW7+d8 #8, rel	R7, @RW7+d8 #8, rel
+8	ADDL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #16, rel	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #8, rel	@RW0, @RW0+d16 #8, rel
+9	ADDL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #16, rel	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #8, rel	@RW1, @RW1+d16 #8, rel
+A	ADDL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #16, rel	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #8, rel	@RW2, @RW2+d16 #8, rel
+B	ADDL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #16, rel	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #8, rel	@RW3, @RW3+d16 #8, rel
+C	ADDL A, A, @RW0+R7	SUBL A, A, @RW0+R7	SUBL A, A, @RW0+R7	SUBL A, A, @RW0+R7	@RW0+R7 #16, rel	CMPL A, A, @RW0+R7	CMPL A, A, @RW0+R7	CMPL A, A, @RW0+R7	ANDL A, A, @RW0+R7	ANDL A, A, @RW0+R7	ORL A, A, @RW0+R7	ORL A, A, @RW0+R7	XORL A, A, @RW0+R7	XORL A, A, @RW0+R7	@RW0+R7 #8, rel	@RW0+R7 #8, rel
+D	ADDL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	@RW1+R7 #16, rel	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ORL A, A, @RW1+R7	ORL A, A, @RW1+R7	XORL A, A, @RW1+R7	XORL A, A, @RW1+R7	@RW1+R7 #8, rel	@RW1+R7 #8, rel
+E	ADDL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	@PC+d16 #16, rel	CMPL A, A, @RW2+, @PC+d16	CMPL A, A, @RW2+, @PC+d16	CMPL A, A, @RW2+, @PC+d16	ANDL A, A, @RW2+, @PC+d16	ANDL A, A, @RW2+, @PC+d16	ORL A, A, @RW2+, @PC+d16	ORL A, A, @RW2+, @PC+d16	XORL A, A, @RW2+, @PC+d16	XORL A, A, @RW2+, @PC+d16	@PC+d16 #8, rel	@PC+d16 #8, rel
+F	ADDL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	addr16 #16, rel	CMPL A, A, @RW3+, addr16	CMPL A, A, @RW3+, addr16	CMPL A, A, @RW3+, addr16	ANDL A, A, @RW3+, addr16	ANDL A, A, @RW3+, addr16	ORL A, A, @RW3+, addr16	ORL A, A, @RW3+, addr16	XORL A, A, @RW3+, addr16	XORL A, A, @RW3+, addr16	addr16 #8, rel	addr16 #8, rel



表 B.9-7 ea 系命令 その 2 (第 1 バイト = 71<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMPP @ RL0, @ @RW0+d8	JMPP @ @ RL0, @ @RW0+d8	CALLP @ RL0, @ @RW0+d8	CALLP @ @ RL0, @ @RW0+d8	INCL @ RL0, @ @RW0+d8	INCL @ RL0, @ @RW0+d8	DECL @ RL0, @ @RW0+d8	DECL @ RL0, @ @RW0+d8	MOVL A, @ RL0, @ @RW0+d8	MOVL A, @ RL0, @ @RW0+d8	MOVL @ RL0, A, @ @RW0+d8	MOVL @ RL0, A, @ @RW0+d8	MOV R0, #8, @ @RW0+d8, #8	MOV @ RL0, #8, @ @RW0+d8, #8	MOVEA A, @ RW0, @ @RW0+d8	MOVEA A, @ RW0, @ @RW0+d8
+1	JMPP @ RL0, @ @RW1+d8	JMPP @ @ RL0, @ @RW1+d8	CALLP @ RL0, @ @RW1+d8	CALLP @ @ RL0, @ @RW1+d8	INCL @ RL0, @ @RW1+d8	INCL @ RL0, @ @RW1+d8	DECL @ RL0, @ @RW1+d8	DECL @ RL0, @ @RW1+d8	MOVL A, @ RL0, @ @RW1+d8	MOVL A, @ RL0, @ @RW1+d8	MOVL @ RL0, A, @ @RW1+d8	MOVL @ RL0, A, @ @RW1+d8	MOV R1, #8, @ @RW1+d8, #8	MOV @ RL0, #8, @ @RW1+d8, #8	MOVEA A, @ RW1, @ @RW1+d8	MOVEA A, @ RW1, @ @RW1+d8
+2	JMPP @ RL1, @ @RW2+d8	JMPP @ @ RL1, @ @RW2+d8	CALLP @ RL1, @ @RW2+d8	CALLP @ @ RL1, @ @RW2+d8	INCL @ RL1, @ @RW2+d8	INCL @ RL1, @ @RW2+d8	DECL @ RL1, @ @RW2+d8	DECL @ RL1, @ @RW2+d8	MOVL A, @ RL1, @ @RW2+d8	MOVL A, @ RL1, @ @RW2+d8	MOVL @ RL1, A, @ @RW2+d8	MOVL @ RL1, A, @ @RW2+d8	MOV R2, #8, @ @RW2+d8, #8	MOV @ RL1, #8, @ @RW2+d8, #8	MOVEA A, @ RW2, @ @RW2+d8	MOVEA A, @ RW2, @ @RW2+d8
+3	JMPP @ RL1, @ @RW3+d8	JMPP @ @ RL1, @ @RW3+d8	CALLP @ RL1, @ @RW3+d8	CALLP @ @ RL1, @ @RW3+d8	INCL @ RL1, @ @RW3+d8	INCL @ RL1, @ @RW3+d8	DECL @ RL1, @ @RW3+d8	DECL @ RL1, @ @RW3+d8	MOVL A, @ RL1, @ @RW3+d8	MOVL A, @ RL1, @ @RW3+d8	MOVL @ RL1, A, @ @RW3+d8	MOVL @ RL1, A, @ @RW3+d8	MOV R3, #8, @ @RW3+d8, #8	MOV @ RL1, #8, @ @RW3+d8, #8	MOVEA A, @ RW3, @ @RW3+d8	MOVEA A, @ RW3, @ @RW3+d8
+4	JMPP @ RL2, @ @RW4+d8	JMPP @ @ RL2, @ @RW4+d8	CALLP @ RL2, @ @RW4+d8	CALLP @ @ RL2, @ @RW4+d8	INCL @ RL2, @ @RW4+d8	INCL @ RL2, @ @RW4+d8	DECL @ RL2, @ @RW4+d8	DECL @ RL2, @ @RW4+d8	MOVL A, @ RL2, @ @RW4+d8	MOVL A, @ RL2, @ @RW4+d8	MOVL @ RL2, A, @ @RW4+d8	MOVL @ RL2, A, @ @RW4+d8	MOV R4, #8, @ @RW4+d8, #8	MOV @ RL2, #8, @ @RW4+d8, #8	MOVEA A, @ RW4, @ @RW4+d8	MOVEA A, @ RW4, @ @RW4+d8
+5	JMPP @ RL2, @ @RW5+d8	JMPP @ @ RL2, @ @RW5+d8	CALLP @ RL2, @ @RW5+d8	CALLP @ @ RL2, @ @RW5+d8	INCL @ RL2, @ @RW5+d8	INCL @ RL2, @ @RW5+d8	DECL @ RL2, @ @RW5+d8	DECL @ RL2, @ @RW5+d8	MOVL A, @ RL2, @ @RW5+d8	MOVL A, @ RL2, @ @RW5+d8	MOVL @ RL2, A, @ @RW5+d8	MOVL @ RL2, A, @ @RW5+d8	MOV R5, #8, @ @RW5+d8, #8	MOV @ RL2, #8, @ @RW5+d8, #8	MOVEA A, @ RW5, @ @RW5+d8	MOVEA A, @ RW5, @ @RW5+d8
+6	JMPP @ RL3, @ @RW6+d8	JMPP @ @ RL3, @ @RW6+d8	CALLP @ RL3, @ @RW6+d8	CALLP @ @ RL3, @ @RW6+d8	INCL @ RL3, @ @RW6+d8	INCL @ RL3, @ @RW6+d8	DECL @ RL3, @ @RW6+d8	DECL @ RL3, @ @RW6+d8	MOVL A, @ RL3, @ @RW6+d8	MOVL A, @ RL3, @ @RW6+d8	MOVL @ RL3, A, @ @RW6+d8	MOVL @ RL3, A, @ @RW6+d8	MOV R6, #8, @ @RW6+d8, #8	MOV @ RL3, #8, @ @RW6+d8, #8	MOVEA A, @ RW6, @ @RW6+d8	MOVEA A, @ RW6, @ @RW6+d8
+7	JMPP @ RL3, @ @RW7+d8	JMPP @ @ RL3, @ @RW7+d8	CALLP @ RL3, @ @RW7+d8	CALLP @ @ RL3, @ @RW7+d8	INCL @ RL3, @ @RW7+d8	INCL @ RL3, @ @RW7+d8	DECL @ RL3, @ @RW7+d8	DECL @ RL3, @ @RW7+d8	MOVL A, @ RL3, @ @RW7+d8	MOVL A, @ RL3, @ @RW7+d8	MOVL @ RL3, A, @ @RW7+d8	MOVL @ RL3, A, @ @RW7+d8	MOV R7, #8, @ @RW7+d8, #8	MOV @ RL3, #8, @ @RW7+d8, #8	MOVEA A, @ RW7, @ @RW7+d8	MOVEA A, @ RW7, @ @RW7+d8
+8	JMPP @ @ @RW0, @ @RW0+d16	JMPP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	INCL @ @RW0, @ @RW0+d16	INCL @ @RW0, @ @RW0+d16	DECL @ @RW0, @ @RW0+d16	DECL @ @RW0, @ @RW0+d16	MOVL A, @ @RW0, @ @RW0+d16	MOVL A, @ @RW0, @ @RW0+d16	MOVL @ @RW0, A, @ @RW0+d16	MOVL @ @RW0, A, @ @RW0+d16	MOV @ @RW0, #8, @ @RW0+d16, #8	MOV @ @RW0, #8, @ @RW0+d16, #8	MOVEA A, @ @RW0, @ @RW0+d16	MOVEA A, @ @RW0, @ @RW0+d16
+9	JMPP @ @ @RW1, @ @RW1+d16	JMPP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	INCL @ @RW1, @ @RW1+d16	INCL @ @RW1, @ @RW1+d16	DECL @ @RW1, @ @RW1+d16	DECL @ @RW1, @ @RW1+d16	MOVL A, @ @RW1, @ @RW1+d16	MOVL A, @ @RW1, @ @RW1+d16	MOVL @ @RW1, A, @ @RW1+d16	MOVL @ @RW1, A, @ @RW1+d16	MOV @ @RW1, #8, @ @RW1+d16, #8	MOV @ @RW1, #8, @ @RW1+d16, #8	MOVEA A, @ @RW1, @ @RW1+d16	MOVEA A, @ @RW1, @ @RW1+d16
+A	JMPP @ @ @RW2, @ @RW2+d16	JMPP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	INCL @ @RW2, @ @RW2+d16	INCL @ @RW2, @ @RW2+d16	DECL @ @RW2, @ @RW2+d16	DECL @ @RW2, @ @RW2+d16	MOVL A, @ @RW2, @ @RW2+d16	MOVL A, @ @RW2, @ @RW2+d16	MOVL @ @RW2, A, @ @RW2+d16	MOVL @ @RW2, A, @ @RW2+d16	MOV @ @RW2, #8, @ @RW2+d16, #8	MOV @ @RW2, #8, @ @RW2+d16, #8	MOVEA A, @ @RW2, @ @RW2+d16	MOVEA A, @ @RW2, @ @RW2+d16
+B	JMPP @ @ @RW3, @ @RW3+d16	JMPP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	INCL @ @RW3, @ @RW3+d16	INCL @ @RW3, @ @RW3+d16	DECL @ @RW3, @ @RW3+d16	DECL @ @RW3, @ @RW3+d16	MOVL A, @ @RW3, @ @RW3+d16	MOVL A, @ @RW3, @ @RW3+d16	MOVL @ @RW3, A, @ @RW3+d16	MOVL @ @RW3, A, @ @RW3+d16	MOV @ @RW3, #8, @ @RW3+d16, #8	MOV @ @RW3, #8, @ @RW3+d16, #8	MOVEA A, @ @RW3, @ @RW3+d16	MOVEA A, @ @RW3, @ @RW3+d16
+C	JMPP @ @ @RW0+, @ @RW0+RW7	JMPP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	INCL @ @RW0+, @ @RW0+RW7	INCL @ @RW0+, @ @RW0+RW7	DECL @ @RW0+, @ @RW0+RW7	DECL @ @RW0+, @ @RW0+RW7	MOVL A, @ @RW0+, @ @RW0+RW7	MOVL A, @ @RW0+, @ @RW0+RW7	MOVL @ @RW0+, A, @ @RW0+RW7	MOVL @ @RW0+, A, @ @RW0+RW7	MOV @ @RW0+, #8, @ @RW0+RW7, #8	MOV @ @RW0+, #8, @ @RW0+RW7, #8	MOVEA A, @ @RW0+, @ @RW0+RW7	MOVEA A, @ @RW0+, @ @RW0+RW7
+D	JMPP @ @ @RW1+, @ @RW1+RW7	JMPP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	INCL @ @RW1+, @ @RW1+RW7	INCL @ @RW1+, @ @RW1+RW7	DECL @ @RW1+, @ @RW1+RW7	DECL @ @RW1+, @ @RW1+RW7	MOVL A, @ @RW1+, @ @RW1+RW7	MOVL A, @ @RW1+, @ @RW1+RW7	MOVL @ @RW1+, A, @ @RW1+RW7	MOVL @ @RW1+, A, @ @RW1+RW7	MOV @ @RW1+, #8, @ @RW1+RW7, #8	MOV @ @RW1+, #8, @ @RW1+RW7, #8	MOVEA A, @ @RW1+, @ @RW1+RW7	MOVEA A, @ @RW1+, @ @RW1+RW7
+E	JMPP @ @ @PC+d16	JMPP @ @ @PC+d16	CALLP @ @ @PC+d16	CALLP @ @ @PC+d16	INCL @ @PC+d16	INCL @ @PC+d16	DECL @ @PC+d16	DECL @ @PC+d16	MOVL A, @ @PC+d16, A	MOVL A, @ @PC+d16, A	MOVL @ @PC+d16, A	MOVL @ @PC+d16, A	MOV @ @PC+d16, #8, @ @PC+d16, #8	MOV @ @PC+d16, #8, @ @PC+d16, #8	MOVEA A, @ @PC+d16, A	MOVEA A, @ @PC+d16, A
+F	JMPP @ @ @addr16	JMPP @ @ @addr16	CALLP @ @ @addr16	CALLP @ @ @addr16	INCL @ @addr16	INCL @ @addr16	DECL @ @addr16	DECL @ @addr16	MOVL A, @ @addr16, A	MOVL A, @ @addr16, A	MOVL @ @addr16, A	MOVL @ @addr16, A	MOV @ @addr16, #8, @ @addr16, #8	MOV @ @addr16, #8, @ @addr16, #8	MOVEA A, @ @addr16, A	MOVEA A, @ @addr16, A

表 B.9-8 ea 系命令 その 3 (第 1 バイト = 72<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ROL	R0' @RW0+d8	RORC	R0' @RW0+d8	INC	R0' @RW0+d8	DEC	R0' @RW0+d8	MOV	A, R0' @RW0+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+1	ROL	R1' @RW1+d8	RORC	R1' @RW1+d8	INC	R1' @RW1+d8	DEC	R1' @RW1+d8	MOV	A, R1' @RW1+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+2	ROL	R2' @RW2+d8	RORC	R2' @RW2+d8	INC	R2' @RW2+d8	DEC	R2' @RW2+d8	MOV	A, R2' @RW2+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+3	ROL	R3' @RW3+d8	RORC	R3' @RW3+d8	INC	R3' @RW3+d8	DEC	R3' @RW3+d8	MOV	A, R3' @RW3+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+4	ROL	R4' @RW4+d8	RORC	R4' @RW4+d8	INC	R4' @RW4+d8	DEC	R4' @RW4+d8	MOV	A, R4' @RW4+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+5	ROL	R5' @RW5+d8	RORC	R5' @RW5+d8	INC	R5' @RW5+d8	DEC	R5' @RW5+d8	MOV	A, R5' @RW5+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+6	ROL	R6' @RW6+d8	RORC	R6' @RW6+d8	INC	R6' @RW6+d8	DEC	R6' @RW6+d8	MOV	A, R6' @RW6+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+7	ROL	R7' @RW7+d8	RORC	R7' @RW7+d8	INC	R7' @RW7+d8	DEC	R7' @RW7+d8	MOV	A, R7' @RW7+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+8	ROL	R0' @RW0+d16	RORC	R0' @RW0+d16	INC	R0' @RW0+d16	DEC	R0' @RW0+d16	MOV	A, R0' @RW0+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+9	ROL	R1' @RW1+d16	RORC	R1' @RW1+d16	INC	R1' @RW1+d16	DEC	R1' @RW1+d16	MOV	A, R1' @RW1+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+A	ROL	R2' @RW2+d16	RORC	R2' @RW2+d16	INC	R2' @RW2+d16	DEC	R2' @RW2+d16	MOV	A, R2' @RW2+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+B	ROL	R3' @RW3+d16	RORC	R3' @RW3+d16	INC	R3' @RW3+d16	DEC	R3' @RW3+d16	MOV	A, R3' @RW3+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+C	ROL	R0' @RW0+R7	RORC	R0' @RW0+R7	INC	R0' @RW0+R7	DEC	R0' @RW0+R7	MOV	A, R0' @RW0+R7	MOV	MOV	MOVX	MOVX	XCH	XCH
+D	ROL	R1' @RW1+R7	RORC	R1' @RW1+R7	INC	R1' @RW1+R7	DEC	R1' @RW1+R7	MOV	A, R1' @RW1+R7	MOV	MOV	MOVX	MOVX	XCH	XCH
+E	ROL	R2' @RW2+R7	RORC	R2' @RW2+R7	INC	R2' @RW2+R7	DEC	R2' @RW2+R7	MOV	A, R2' @RW2+R7	MOV	MOV	MOVX	MOVX	XCH	XCH
+F	ROL	R3' @RW3+R7	RORC	R3' @RW3+R7	INC	R3' @RW3+R7	DEC	R3' @RW3+R7	MOV	A, R3' @RW3+R7	MOV	MOV	MOVX	MOVX	XCH	XCH

表 B.9-9 ea 系命令 その4 (第1バイト = 73<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMP @RW0, @@RW0+d8	JMP @RW0, @@RW0+d8	CALL RW0, @@RW0+d8	CALL RW0, @@RW0+d8	INCW RW0, @@RW0+d8	INCW RW0, @@RW0+d8	DECW RW0, @@RW0+d8	DECW RW0, @@RW0+d8	MOVW A, RW0, @@RW0+d8	MOVW A, RW0, @@RW0+d8	MOVW RW0, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	XCHW A, RW0, @@RW0+d8	XCHW A, RW0, @@RW0+d8
+1	JMP @RW1, @@RW1+d8	JMP @RW1, @@RW1+d8	CALL RW1, @@RW1+d8	CALL RW1, @@RW1+d8	INCW RW1, @@RW1+d8	INCW RW1, @@RW1+d8	DECW RW1, @@RW1+d8	DECW RW1, @@RW1+d8	MOVW A, RW1, @@RW1+d8	MOVW A, RW1, @@RW1+d8	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	XCHW A, RW1, @@RW1+d8	XCHW A, RW1, @@RW1+d8
+2	JMP @RW2, @@RW2+d8	JMP @RW2, @@RW2+d8	CALL RW2, @@RW2+d8	CALL RW2, @@RW2+d8	INCW RW2, @@RW2+d8	INCW RW2, @@RW2+d8	DECW RW2, @@RW2+d8	DECW RW2, @@RW2+d8	MOVW A, RW2, @@RW2+d8	MOVW A, RW2, @@RW2+d8	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	XCHW A, RW2, @@RW2+d8	XCHW A, RW2, @@RW2+d8
+3	JMP @RW3, @@RW3+d8	JMP @RW3, @@RW3+d8	CALL RW3, @@RW3+d8	CALL RW3, @@RW3+d8	INCW RW3, @@RW3+d8	INCW RW3, @@RW3+d8	DECW RW3, @@RW3+d8	DECW RW3, @@RW3+d8	MOVW A, RW3, @@RW3+d8	MOVW A, RW3, @@RW3+d8	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	XCHW A, RW3, @@RW3+d8	XCHW A, RW3, @@RW3+d8
+4	JMP @RW4, @@RW4+d8	JMP @RW4, @@RW4+d8	CALL RW4, @@RW4+d8	CALL RW4, @@RW4+d8	INCW RW4, @@RW4+d8	INCW RW4, @@RW4+d8	DECW RW4, @@RW4+d8	DECW RW4, @@RW4+d8	MOVW A, RW4, @@RW4+d8	MOVW A, RW4, @@RW4+d8	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	XCHW A, RW4, @@RW4+d8	XCHW A, RW4, @@RW4+d8
+5	JMP @RW5, @@RW5+d8	JMP @RW5, @@RW5+d8	CALL RW5, @@RW5+d8	CALL RW5, @@RW5+d8	INCW RW5, @@RW5+d8	INCW RW5, @@RW5+d8	DECW RW5, @@RW5+d8	DECW RW5, @@RW5+d8	MOVW A, RW5, @@RW5+d8	MOVW A, RW5, @@RW5+d8	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	XCHW A, RW5, @@RW5+d8	XCHW A, RW5, @@RW5+d8
+6	JMP @RW6, @@RW6+d8	JMP @RW6, @@RW6+d8	CALL RW6, @@RW6+d8	CALL RW6, @@RW6+d8	INCW RW6, @@RW6+d8	INCW RW6, @@RW6+d8	DECW RW6, @@RW6+d8	DECW RW6, @@RW6+d8	MOVW A, RW6, @@RW6+d8	MOVW A, RW6, @@RW6+d8	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	XCHW A, RW6, @@RW6+d8	XCHW A, RW6, @@RW6+d8
+7	JMP @RW7, @@RW7+d8	JMP @RW7, @@RW7+d8	CALL RW7, @@RW7+d8	CALL RW7, @@RW7+d8	INCW RW7, @@RW7+d8	INCW RW7, @@RW7+d8	DECW RW7, @@RW7+d8	DECW RW7, @@RW7+d8	MOVW A, RW7, @@RW7+d8	MOVW A, RW7, @@RW7+d8	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	XCHW A, RW7, @@RW7+d8	XCHW A, RW7, @@RW7+d8
+8	JMP @RW0, @RW0+d16	JMP @RW0, @RW0+d16	CALL @RW0, @RW0+d16	CALL @RW0, @RW0+d16	INCW @RW0, @RW0+d16	INCW @RW0, @RW0+d16	DECW @RW0, @RW0+d16	DECW @RW0, @RW0+d16	MOVW A, @RW0, @RW0+d16	MOVW A, @RW0, @RW0+d16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	XCHW A, @RW0, @RW0+d16	XCHW A, @RW0, @RW0+d16
+9	JMP @RW1, @RW1+d16	JMP @RW1, @RW1+d16	CALL @RW1, @RW1+d16	CALL @RW1, @RW1+d16	INCW @RW1, @RW1+d16	INCW @RW1, @RW1+d16	DECW @RW1, @RW1+d16	DECW @RW1, @RW1+d16	MOVW A, @RW1, @RW1+d16	MOVW A, @RW1, @RW1+d16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	XCHW A, @RW1, @RW1+d16	XCHW A, @RW1, @RW1+d16
+A	JMP @RW2, @RW2+d16	JMP @RW2, @RW2+d16	CALL @RW2, @RW2+d16	CALL @RW2, @RW2+d16	INCW @RW2, @RW2+d16	INCW @RW2, @RW2+d16	DECW @RW2, @RW2+d16	DECW @RW2, @RW2+d16	MOVW A, @RW2, @RW2+d16	MOVW A, @RW2, @RW2+d16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	XCHW A, @RW2, @RW2+d16	XCHW A, @RW2, @RW2+d16
+B	JMP @RW3, @RW3+d16	JMP @RW3, @RW3+d16	CALL @RW3, @RW3+d16	CALL @RW3, @RW3+d16	INCW @RW3, @RW3+d16	INCW @RW3, @RW3+d16	DECW @RW3, @RW3+d16	DECW @RW3, @RW3+d16	MOVW A, @RW3, @RW3+d16	MOVW A, @RW3, @RW3+d16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	XCHW A, @RW3, @RW3+d16	XCHW A, @RW3, @RW3+d16
+C	JMP @RW0+, @RW0+RW7	JMP @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	XCHW A, @RW0+, @RW0+RW7	XCHW A, @RW0+, @RW0+RW7
+D	JMP @RW1+, @RW1+RW7	JMP @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	XCHW A, @RW1+, @RW1+RW7	XCHW A, @RW1+, @RW1+RW7
+E	JMP @RW2+, @PC+d16	JMP @RW2+, @PC+d16	CALL @RW2+, @PC+d16	CALL @RW2+, @PC+d16	INCW @RW2+, @PC+d16	INCW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	XCHW A, @RW2+, @PC+d16	XCHW A, @RW2+, @PC+d16
+F	JMP @RW3+, @addr16	JMP @RW3+, @addr16	CALL @RW3+, @addr16	CALL @RW3+, @addr16	INCW @RW3+, @addr16	INCW @RW3+, @addr16	DECW @RW3+, @addr16	DECW @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	XCHW A, @RW3+, @addr16	XCHW A, @RW3+, @addr16

表 B.9-10 ea 系命令 その5 (第1バイト = 74<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD A, R0, @RW0+d8	ADD A, R0, @RW0+d8	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	CMP A, R0, @RW0+d8	CMP A, R0, @RW0+d8	AND A, R0, @RW0+d8	AND A, R0, @RW0+d8	OR A, R0, @RW0+d8	OR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	DBNZ R0, r, RW0+d8, r	DBNZ R0, r, RW0+d8, r
+1	ADD A, R1, @RW1+d8	ADD A, R1, @RW1+d8	SUB A, R1, @RW1+d8	SUB A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	CMP A, R1, @RW1+d8	CMP A, R1, @RW1+d8	AND A, R1, @RW1+d8	AND A, R1, @RW1+d8	OR A, R1, @RW1+d8	OR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	DBNZ R1, r, RW1+d8, r	DBNZ R1, r, RW1+d8, r
+2	ADD A, R2, @RW2+d8	ADD A, R2, @RW2+d8	SUB A, R2, @RW2+d8	SUB A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	CMP A, R2, @RW2+d8	CMP A, R2, @RW2+d8	AND A, R2, @RW2+d8	AND A, R2, @RW2+d8	OR A, R2, @RW2+d8	OR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	DBNZ R2, r, RW2+d8, r	DBNZ R2, r, RW2+d8, r
+3	ADD A, R3, @RW3+d8	ADD A, R3, @RW3+d8	SUB A, R3, @RW3+d8	SUB A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	CMP A, R3, @RW3+d8	CMP A, R3, @RW3+d8	AND A, R3, @RW3+d8	AND A, R3, @RW3+d8	OR A, R3, @RW3+d8	OR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	DBNZ R3, r, RW3+d8, r	DBNZ R3, r, RW3+d8, r
+4	ADD A, R4, @RW4+d8	ADD A, R4, @RW4+d8	SUB A, R4, @RW4+d8	SUB A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	CMP A, R4, @RW4+d8	CMP A, R4, @RW4+d8	AND A, R4, @RW4+d8	AND A, R4, @RW4+d8	OR A, R4, @RW4+d8	OR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	DBNZ R4, r, RW4+d8, r	DBNZ R4, r, RW4+d8, r
+5	ADD A, R5, @RW5+d8	ADD A, R5, @RW5+d8	SUB A, R5, @RW5+d8	SUB A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	CMP A, R5, @RW5+d8	CMP A, R5, @RW5+d8	AND A, R5, @RW5+d8	AND A, R5, @RW5+d8	OR A, R5, @RW5+d8	OR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	DBNZ R5, r, RW5+d8, r	DBNZ R5, r, RW5+d8, r
+6	ADD A, R6, @RW6+d8	ADD A, R6, @RW6+d8	SUB A, R6, @RW6+d8	SUB A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	CMP A, R6, @RW6+d8	CMP A, R6, @RW6+d8	AND A, R6, @RW6+d8	AND A, R6, @RW6+d8	OR A, R6, @RW6+d8	OR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	DBNZ R6, r, RW6+d8, r	DBNZ R6, r, RW6+d8, r
+7	ADD A, R7, @RW7+d8	ADD A, R7, @RW7+d8	SUB A, R7, @RW7+d8	SUB A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	CMP A, R7, @RW7+d8	CMP A, R7, @RW7+d8	AND A, R7, @RW7+d8	AND A, R7, @RW7+d8	OR A, R7, @RW7+d8	OR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	DBNZ R7, r, RW7+d8, r	DBNZ R7, r, RW7+d8, r
+8	ADD A, @RW0, @RW0+d16	ADD A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	DBNZ @RW0, r, W0+d16, r	DBNZ @RW0, r, W0+d16, r
+9	ADD A, @RW1, @RW1+d16	ADD A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	DBNZ @RW1, r, W1+d16, r	DBNZ @RW1, r, W1+d16, r
+A	ADD A, @RW2, @RW2+d16	ADD A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	DBNZ @RW2, r, W2+d16, r	DBNZ @RW2, r, W2+d16, r
+B	ADD A, @RW3, @RW3+d16	ADD A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	DBNZ @RW3, r, W3+d16, r	DBNZ @RW3, r, W3+d16, r
+C	ADD A, @RW0+, @RW0+RW7	ADD A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	DBNZ @RW0+, r, W0+RW7, r	DBNZ @RW0+, r, W0+RW7, r
+D	ADD A, @RW1+, @RW1+RW7	ADD A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	DBNZ @RW1+, r, W1+RW7, r	DBNZ @RW1+, r, W1+RW7, r
+E	ADD A, @RW2+, @PC+d16	ADD A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	DBNZ @RW2+, r, PC+d16, r	DBNZ @RW2+, r, PC+d16, r
+F	ADD A, @RW3+, A, addr16	ADD A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	DBNZ @RW3+, r, addr16, r	DBNZ @RW3+, r, addr16, r

表 B.9-11 ea 系命令 その6 (第1バイト = 75<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUBC A, R0', @RW0+d8	SUBC A, R0', @RW0+d8	NEG R0', @RW0+d8	NEG A, R0', @RW0+d8	AND R0, A', @RW0+d8, A	AND R0, A', @RW0+d8, A	OR R0, A', @RW0+d8, A	OR R0, A', @RW0+d8, A	XOR R0, A', @RW0+d8, A	XOR R0, A', @RW0+d8, A	NOT R0', @RW0+d8	NOT R0', @RW0+d8
+1	ADD R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUBC A, R1', @RW1+d8	SUBC A, R1', @RW1+d8	NEG R1', @RW1+d8	NEG A, R1', @RW1+d8	AND R1, A', @RW1+d8, A	AND R1, A', @RW1+d8, A	OR R1, A', @RW1+d8, A	OR R1, A', @RW1+d8, A	XOR R1, A', @RW1+d8, A	XOR R1, A', @RW1+d8, A	NOT R1', @RW1+d8	NOT R1', @RW1+d8
+2	ADD R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUBC A, R2', @RW2+d8	SUBC A, R2', @RW2+d8	NEG R2', @RW2+d8	NEG A, R2', @RW2+d8	AND R2, A', @RW2+d8, A	AND R2, A', @RW2+d8, A	OR R2, A', @RW2+d8, A	OR R2, A', @RW2+d8, A	XOR R2, A', @RW2+d8, A	XOR R2, A', @RW2+d8, A	NOT R2', @RW2+d8	NOT R2', @RW2+d8
+3	ADD R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUBC A, R3', @RW3+d8	SUBC A, R3', @RW3+d8	NEG R3', @RW3+d8	NEG A, R3', @RW3+d8	AND R3, A', @RW3+d8, A	AND R3, A', @RW3+d8, A	OR R3, A', @RW3+d8, A	OR R3, A', @RW3+d8, A	XOR R3, A', @RW3+d8, A	XOR R3, A', @RW3+d8, A	NOT R3', @RW3+d8	NOT R3', @RW3+d8
+4	ADD R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUBC A, R4', @RW4+d8	SUBC A, R4', @RW4+d8	NEG R4', @RW4+d8	NEG A, R4', @RW4+d8	AND R4, A', @RW4+d8, A	AND R4, A', @RW4+d8, A	OR R4, A', @RW4+d8, A	OR R4, A', @RW4+d8, A	XOR R4, A', @RW4+d8, A	XOR R4, A', @RW4+d8, A	NOT R4', @RW4+d8	NOT R4', @RW4+d8
+5	ADD R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUBC A, R5', @RW5+d8	SUBC A, R5', @RW5+d8	NEG R5', @RW5+d8	NEG A, R5', @RW5+d8	AND R5, A', @RW5+d8, A	AND R5, A', @RW5+d8, A	OR R5, A', @RW5+d8, A	OR R5, A', @RW5+d8, A	XOR R5, A', @RW5+d8, A	XOR R5, A', @RW5+d8, A	NOT R5', @RW5+d8	NOT R5', @RW5+d8
+6	ADD R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUBC A, R6', @RW6+d8	SUBC A, R6', @RW6+d8	NEG R6', @RW6+d8	NEG A, R6', @RW6+d8	AND R6, A', @RW6+d8, A	AND R6, A', @RW6+d8, A	OR R6, A', @RW6+d8, A	OR R6, A', @RW6+d8, A	XOR R6, A', @RW6+d8, A	XOR R6, A', @RW6+d8, A	NOT R6', @RW6+d8	NOT R6', @RW6+d8
+7	ADD R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUBC A, R7', @RW7+d8	SUBC A, R7', @RW7+d8	NEG R7', @RW7+d8	NEG A, R7', @RW7+d8	AND R7, A', @RW7+d8, A	AND R7, A', @RW7+d8, A	OR R7, A', @RW7+d8, A	OR R7, A', @RW7+d8, A	XOR R7, A', @RW7+d8, A	XOR R7, A', @RW7+d8, A	NOT R7', @RW7+d8	NOT R7', @RW7+d8
+8	ADD @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUBC A, @RW0', @RW0+d16	SUBC A, @RW0', @RW0+d16	NEG @RW0', @RW0+d16	NEG A, @RW0', @RW0+d16	AND @RW0, A', @RW0+d16, A	AND @RW0, A', @RW0+d16, A	OR @RW0, A', @RW0+d16, A	OR @RW0, A', @RW0+d16, A	XOR @RW0, A', @RW0+d16, A	XOR @RW0, A', @RW0+d16, A	NOT @RW0', @RW0+d16	NOT @RW0', @RW0+d16
+9	ADD @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUBC A, @RW1', @RW1+d16	SUBC A, @RW1', @RW1+d16	NEG @RW1', @RW1+d16	NEG A, @RW1', @RW1+d16	AND @RW1, A', @RW1+d16, A	AND @RW1, A', @RW1+d16, A	OR @RW1, A', @RW1+d16, A	OR @RW1, A', @RW1+d16, A	XOR @RW1, A', @RW1+d16, A	XOR @RW1, A', @RW1+d16, A	NOT @RW1', @RW1+d16	NOT @RW1', @RW1+d16
+A	ADD @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUBC A, @RW2', @RW2+d16	SUBC A, @RW2', @RW2+d16	NEG @RW2', @RW2+d16	NEG A, @RW2', @RW2+d16	AND @RW2, A', @RW2+d16, A	AND @RW2, A', @RW2+d16, A	OR @RW2, A', @RW2+d16, A	OR @RW2, A', @RW2+d16, A	XOR @RW2, A', @RW2+d16, A	XOR @RW2, A', @RW2+d16, A	NOT @RW2', @RW2+d16	NOT @RW2', @RW2+d16
+B	ADD @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUBC A, @RW3', @RW3+d16	SUBC A, @RW3', @RW3+d16	NEG @RW3', @RW3+d16	NEG A, @RW3', @RW3+d16	AND @RW3, A', @RW3+d16, A	AND @RW3, A', @RW3+d16, A	OR @RW3, A', @RW3+d16, A	OR @RW3, A', @RW3+d16, A	XOR @RW3, A', @RW3+d16, A	XOR @RW3, A', @RW3+d16, A	NOT @RW3', @RW3+d16	NOT @RW3', @RW3+d16
+C	ADD @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUBC A, @RW0+', @RW0+RW7	SUBC A, @RW0+', @RW0+RW7	NEG @RW0+', @RW0+RW7	NEG A, @RW0+', @RW0+RW7	AND @RW0+, A', @RW0+RW7, A	AND @RW0+, A', @RW0+RW7, A	OR @RW0+, A', @RW0+RW7, A	OR @RW0+, A', @RW0+RW7, A	XOR @RW0+, A', @RW0+RW7, A	XOR @RW0+, A', @RW0+RW7, A	NOT @RW0+', @RW0+RW7	NOT @RW0+', @RW0+RW7
+D	ADD @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUBC A, @RW1+', @RW1+RW7	SUBC A, @RW1+', @RW1+RW7	NEG @RW1+', @RW1+RW7	NEG A, @RW1+', @RW1+RW7	AND @RW1+, A', @RW1+RW7, A	AND @RW1+, A', @RW1+RW7, A	OR @RW1+, A', @RW1+RW7, A	OR @RW1+, A', @RW1+RW7, A	XOR @RW1+, A', @RW1+RW7, A	XOR @RW1+, A', @RW1+RW7, A	NOT @RW1+', @RW1+RW7	NOT @RW1+', @RW1+RW7
+E	ADD @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUBC A, @RW2+', @PC+d16	SUBC A, @RW2+', @PC+d16	NEG @RW2+', @PC+d16	NEG A, @RW2+', @PC+d16	AND @RW2+, A', @PC+d16, A	AND @RW2+, A', @PC+d16, A	OR @RW2+, A', @PC+d16, A	OR @RW2+, A', @PC+d16, A	XOR @RW2+, A', @PC+d16, A	XOR @RW2+, A', @PC+d16, A	NOT @RW2+', @PC+d16	NOT @RW2+', @PC+d16
+F	ADD @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUBC A, @RW3+', addr16	SUBC A, @RW3+', addr16	NEG @RW3+', addr16	NEG A, @RW3+', addr16	AND @RW3+, A', addr16, A	AND @RW3+, A', addr16, A	OR @RW3+, A', addr16, A	OR @RW3+, A', addr16, A	XOR @RW3+, A', addr16, A	XOR @RW3+, A', addr16, A	NOT @RW3+', addr16	NOT @RW3+', addr16

表 B.9-12 ea 系命令 その 7 (第 1 バイト = 76<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW A, RW0, @RW0+d8	ADDW A, RW0, @RW0+d8	SUBW A, RW0, @RW0+d8	SUBW A, RW0, @RW0+d8	ADDCW A, RW0, @RW0+d8	ADDCW A, RW0, @RW0+d8	CMPW A, RW0, @RW0+d8	CMPW A, RW0, @RW0+d8	ANDW A, RW0, @RW0+d8	ANDW A, RW0, @RW0+d8	ORW A, RW0, @RW0+d8	ORW A, RW0, @RW0+d8	XORW A, RW0, @RW0+d8	XORW A, RW0, @RW0+d8	DWBZ RW0, r' @RW0+d8, r	DWBZ RW0, r' @RW0+d8, r
+1	ADDW A, RW1, @RW1+d8	ADDW A, RW1, @RW1+d8	SUBW A, RW1, @RW1+d8	SUBW A, RW1, @RW1+d8	ADDCW A, RW1, @RW1+d8	ADDCW A, RW1, @RW1+d8	CMPW A, RW1, @RW1+d8	CMPW A, RW1, @RW1+d8	ANDW A, RW1, @RW1+d8	ANDW A, RW1, @RW1+d8	ORW A, RW1, @RW1+d8	ORW A, RW1, @RW1+d8	XORW A, RW1, @RW1+d8	XORW A, RW1, @RW1+d8	DWBZ RW1, r' @RW1+d8, r	DWBZ RW1, r' @RW1+d8, r
+2	ADDW A, RW2, @RW2+d8	ADDW A, RW2, @RW2+d8	SUBW A, RW2, @RW2+d8	SUBW A, RW2, @RW2+d8	ADDCW A, RW2, @RW2+d8	ADDCW A, RW2, @RW2+d8	CMPW A, RW2, @RW2+d8	CMPW A, RW2, @RW2+d8	ANDW A, RW2, @RW2+d8	ANDW A, RW2, @RW2+d8	ORW A, RW2, @RW2+d8	ORW A, RW2, @RW2+d8	XORW A, RW2, @RW2+d8	XORW A, RW2, @RW2+d8	DWBZ RW2, r' @RW2+d8, r	DWBZ RW2, r' @RW2+d8, r
+3	ADDW A, RW3, @RW3+d8	ADDW A, RW3, @RW3+d8	SUBW A, RW3, @RW3+d8	SUBW A, RW3, @RW3+d8	ADDCW A, RW3, @RW3+d8	ADDCW A, RW3, @RW3+d8	CMPW A, RW3, @RW3+d8	CMPW A, RW3, @RW3+d8	ANDW A, RW3, @RW3+d8	ANDW A, RW3, @RW3+d8	ORW A, RW3, @RW3+d8	ORW A, RW3, @RW3+d8	XORW A, RW3, @RW3+d8	XORW A, RW3, @RW3+d8	DWBZ RW3, r' @RW3+d8, r	DWBZ RW3, r' @RW3+d8, r
+4	ADDW A, RW4, @RW4+d8	ADDW A, RW4, @RW4+d8	SUBW A, RW4, @RW4+d8	SUBW A, RW4, @RW4+d8	ADDCW A, RW4, @RW4+d8	ADDCW A, RW4, @RW4+d8	CMPW A, RW4, @RW4+d8	CMPW A, RW4, @RW4+d8	ANDW A, RW4, @RW4+d8	ANDW A, RW4, @RW4+d8	ORW A, RW4, @RW4+d8	ORW A, RW4, @RW4+d8	XORW A, RW4, @RW4+d8	XORW A, RW4, @RW4+d8	DWBZ RW4, r' @RW4+d8, r	DWBZ RW4, r' @RW4+d8, r
+5	ADDW A, RW5, @RW5+d8	ADDW A, RW5, @RW5+d8	SUBW A, RW5, @RW5+d8	SUBW A, RW5, @RW5+d8	ADDCW A, RW5, @RW5+d8	ADDCW A, RW5, @RW5+d8	CMPW A, RW5, @RW5+d8	CMPW A, RW5, @RW5+d8	ANDW A, RW5, @RW5+d8	ANDW A, RW5, @RW5+d8	ORW A, RW5, @RW5+d8	ORW A, RW5, @RW5+d8	XORW A, RW5, @RW5+d8	XORW A, RW5, @RW5+d8	DWBZ RW5, r' @RW5+d8, r	DWBZ RW5, r' @RW5+d8, r
+6	ADDW A, RW6, @RW6+d8	ADDW A, RW6, @RW6+d8	SUBW A, RW6, @RW6+d8	SUBW A, RW6, @RW6+d8	ADDCW A, RW6, @RW6+d8	ADDCW A, RW6, @RW6+d8	CMPW A, RW6, @RW6+d8	CMPW A, RW6, @RW6+d8	ANDW A, RW6, @RW6+d8	ANDW A, RW6, @RW6+d8	ORW A, RW6, @RW6+d8	ORW A, RW6, @RW6+d8	XORW A, RW6, @RW6+d8	XORW A, RW6, @RW6+d8	DWBZ RW6, r' @RW6+d8, r	DWBZ RW6, r' @RW6+d8, r
+7	ADDW A, RW7, @RW7+d8	ADDW A, RW7, @RW7+d8	SUBW A, RW7, @RW7+d8	SUBW A, RW7, @RW7+d8	ADDCW A, RW7, @RW7+d8	ADDCW A, RW7, @RW7+d8	CMPW A, RW7, @RW7+d8	CMPW A, RW7, @RW7+d8	ANDW A, RW7, @RW7+d8	ANDW A, RW7, @RW7+d8	ORW A, RW7, @RW7+d8	ORW A, RW7, @RW7+d8	XORW A, RW7, @RW7+d8	XORW A, RW7, @RW7+d8	DWBZ RW7, r' @RW7+d8, r	DWBZ RW7, r' @RW7+d8, r
+8	ADDW A, @RW0, @RW0+d16	ADDW A, @RW0, @RW0+d16	SUBW A, @RW0, @RW0+d16	SUBW A, @RW0, @RW0+d16	ADDCW A, @RW0, @RW0+d16	ADDCW A, @RW0, @RW0+d16	CMPW A, @RW0, @RW0+d16	CMPW A, @RW0, @RW0+d16	ANDW A, @RW0, @RW0+d16	ANDW A, @RW0, @RW0+d16	ORW A, @RW0, @RW0+d16	ORW A, @RW0, @RW0+d16	XORW A, @RW0, @RW0+d16	XORW A, @RW0, @RW0+d16	DWBZ @RW0, r' @RW0+d16, r	DWBZ @RW0, r' @RW0+d16, r
+9	ADDW A, @RW1, @RW1+d16	ADDW A, @RW1, @RW1+d16	SUBW A, @RW1, @RW1+d16	SUBW A, @RW1, @RW1+d16	ADDCW A, @RW1, @RW1+d16	ADDCW A, @RW1, @RW1+d16	CMPW A, @RW1, @RW1+d16	CMPW A, @RW1, @RW1+d16	ANDW A, @RW1, @RW1+d16	ANDW A, @RW1, @RW1+d16	ORW A, @RW1, @RW1+d16	ORW A, @RW1, @RW1+d16	XORW A, @RW1, @RW1+d16	XORW A, @RW1, @RW1+d16	DWBZ @RW1, r' @RW1+d16, r	DWBZ @RW1, r' @RW1+d16, r
+A	ADDW A, @RW2, @RW2+d16	ADDW A, @RW2, @RW2+d16	SUBW A, @RW2, @RW2+d16	SUBW A, @RW2, @RW2+d16	ADDCW A, @RW2, @RW2+d16	ADDCW A, @RW2, @RW2+d16	CMPW A, @RW2, @RW2+d16	CMPW A, @RW2, @RW2+d16	ANDW A, @RW2, @RW2+d16	ANDW A, @RW2, @RW2+d16	ORW A, @RW2, @RW2+d16	ORW A, @RW2, @RW2+d16	XORW A, @RW2, @RW2+d16	XORW A, @RW2, @RW2+d16	DWBZ @RW2, r' @RW2+d16, r	DWBZ @RW2, r' @RW2+d16, r
+B	ADDW A, @RW3, @RW3+d16	ADDW A, @RW3, @RW3+d16	SUBW A, @RW3, @RW3+d16	SUBW A, @RW3, @RW3+d16	ADDCW A, @RW3, @RW3+d16	ADDCW A, @RW3, @RW3+d16	CMPW A, @RW3, @RW3+d16	CMPW A, @RW3, @RW3+d16	ANDW A, @RW3, @RW3+d16	ANDW A, @RW3, @RW3+d16	ORW A, @RW3, @RW3+d16	ORW A, @RW3, @RW3+d16	XORW A, @RW3, @RW3+d16	XORW A, @RW3, @RW3+d16	DWBZ @RW3, r' @RW3+d16, r	DWBZ @RW3, r' @RW3+d16, r
+C	ADDW A, @RW0+, @RW0+RW7	ADDW A, @RW0+, @RW0+RW7	SUBW A, @RW0+, @RW0+RW7	SUBW A, @RW0+, @RW0+RW7	ADDCW A, @RW0+, @RW0+RW7	ADDCW A, @RW0+, @RW0+RW7	CMPW A, @RW0+, @RW0+RW7	CMPW A, @RW0+, @RW0+RW7	ANDW A, @RW0+, @RW0+RW7	ANDW A, @RW0+, @RW0+RW7	ORW A, @RW0+, @RW0+RW7	ORW A, @RW0+, @RW0+RW7	XORW A, @RW0+, @RW0+RW7	XORW A, @RW0+, @RW0+RW7	DWBZ @RW0+, r' @RW0+RW7, r	DWBZ @RW0+, r' @RW0+RW7, r
+D	ADDW A, @RW1+, @RW1+RW7	ADDW A, @RW1+, @RW1+RW7	SUBW A, @RW1+, @RW1+RW7	SUBW A, @RW1+, @RW1+RW7	ADDCW A, @RW1+, @RW1+RW7	ADDCW A, @RW1+, @RW1+RW7	CMPW A, @RW1+, @RW1+RW7	CMPW A, @RW1+, @RW1+RW7	ANDW A, @RW1+, @RW1+RW7	ANDW A, @RW1+, @RW1+RW7	ORW A, @RW1+, @RW1+RW7	ORW A, @RW1+, @RW1+RW7	XORW A, @RW1+, @RW1+RW7	XORW A, @RW1+, @RW1+RW7	DWBZ @RW1+, r' @RW1+RW7, r	DWBZ @RW1+, r' @RW1+RW7, r
+E	ADDW A, @RW2+, @PC+d16	ADDW A, @RW2+, @PC+d16	SUBW A, @RW2+, @PC+d16	SUBW A, @RW2+, @PC+d16	ADDCW A, @RW2+, @PC+d16	ADDCW A, @RW2+, @PC+d16	CMPW A, @RW2+, @PC+d16	CMPW A, @RW2+, @PC+d16	ANDW A, @RW2+, @PC+d16	ANDW A, @RW2+, @PC+d16	ORW A, @RW2+, @PC+d16	ORW A, @RW2+, @PC+d16	XORW A, @RW2+, @PC+d16	XORW A, @RW2+, @PC+d16	DWBZ @RW2+, r' @PC+d16, r	DWBZ @RW2+, r' @PC+d16, r
+F	ADDW A, @RW3+, addr 16	ADDW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	DWBZ @RW3+, r' addr 16, r	DWBZ @RW3+, r' addr 16, r

表 B.9-13 ea 系命令 その8 (第1バイト = 77<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBCW A, RW0', @RW0+d8	SUBCW A, RW0', @RW0+d8	NEGW RW0', @RW0+d8	NEGW RW0', @RW0+d8	ANDW RW0, A', @RW0+d8, A	ANDW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	NOTW RW0', @RW0+d8	NOTW RW0', @RW0+d8
+1	ADDW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBCW A, RW1', @RW1+d8	SUBCW A, RW1', @RW1+d8	NEGW RW1', @RW1+d8	NEGW RW1', @RW1+d8	ANDW RW1, A', @RW1+d8, A	ANDW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	NOTW RW1', @RW1+d8	NOTW RW1', @RW1+d8
+2	ADDW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBCW A, RW2', @RW2+d8	SUBCW A, RW2', @RW2+d8	NEGW RW2', @RW2+d8	NEGW RW2', @RW2+d8	ANDW RW2, A', @RW2+d8, A	ANDW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	NOTW RW2', @RW2+d8	NOTW RW2', @RW2+d8
+3	ADDW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBCW A, RW3', @RW3+d8	SUBCW A, RW3', @RW3+d8	NEGW RW3', @RW3+d8	NEGW RW3', @RW3+d8	ANDW RW3, A', @RW3+d8, A	ANDW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	NOTW RW3', @RW3+d8	NOTW RW3', @RW3+d8
+4	ADDW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBCW A, RW4', @RW4+d8	SUBCW A, RW4', @RW4+d8	NEGW RW4', @RW4+d8	NEGW RW4', @RW4+d8	ANDW RW4, A', @RW4+d8, A	ANDW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	NOTW RW4', @RW4+d8	NOTW RW4', @RW4+d8
+5	ADDW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBCW A, RW5', @RW5+d8	SUBCW A, RW5', @RW5+d8	NEGW RW5', @RW5+d8	NEGW RW5', @RW5+d8	ANDW RW5, A', @RW5+d8, A	ANDW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	NOTW RW5', @RW5+d8	NOTW RW5', @RW5+d8
+6	ADDW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBCW A, RW6', @RW6+d8	SUBCW A, RW6', @RW6+d8	NEGW RW6', @RW6+d8	NEGW RW6', @RW6+d8	ANDW RW6, A', @RW6+d8, A	ANDW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	NOTW RW6', @RW6+d8	NOTW RW6', @RW6+d8
+7	ADDW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBCW A, RW7', @RW7+d8	SUBCW A, RW7', @RW7+d8	NEGW RW7', @RW7+d8	NEGW RW7', @RW7+d8	ANDW RW7, A', @RW7+d8, A	ANDW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	NOTW RW7', @RW7+d8	NOTW RW7', @RW7+d8
+8	ADDW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBCW A, @RW0', @RW0+d16	SUBCW A, @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	ANDW @RW0, A', @RW0+d16, A	ANDW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	NOTW @RW0', @RW0+d16	NOTW @RW0', @RW0+d16
+9	ADDW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBCW A, @RW1', @RW1+d16	SUBCW A, @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	ANDW @RW1, A', @RW1+d16, A	ANDW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	NOTW @RW1', @RW1+d16	NOTW @RW1', @RW1+d16
+A	ADDW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBCW A, @RW2', @RW2+d16	SUBCW A, @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	ANDW @RW2, A', @RW2+d16, A	ANDW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	NOTW @RW2', @RW2+d16	NOTW @RW2', @RW2+d16
+B	ADDW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBCW A, @RW3', @RW3+d16	SUBCW A, @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	ANDW @RW3, A', @RW3+d16, A	ANDW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	NOTW @RW3', @RW3+d16	NOTW @RW3', @RW3+d16
+C	ADDW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBCW A, @RW0+', @RW0+RW7	SUBCW A, @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	ANDW @RW0+, A', @RW0+RW7, A	ANDW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	NOTW @RW0+', @RW0+RW7	NOTW @RW0+', @RW0+RW7
+D	ADDW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBCW A, @RW1+', @RW1+RW7	SUBCW A, @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	ANDW @RW1+, A', @RW1+RW7, A	ANDW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	NOTW @RW1+', @RW1+RW7	NOTW @RW1+', @RW1+RW7
+E	ADDW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBCW A, @RW2+', @PC+d16	SUBCW A, @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	ANDW @RW2+, A', @PC+d16, A	ANDW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	NOTW @RW2+', @PC+d16	NOTW @RW2+', @PC+d16
+F	ADDW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBCW A, @RW3+', addr16	SUBCW A, @RW3+', addr16	NEGW @RW3+', addr16	NEGW @RW3+', addr16	ANDW @RW3+, A', addr16, A	ANDW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	NOTW @RW3+', addr16	NOTW @RW3+', addr16

表 B.9-14 ea 系命令 その9 (第1バイト = 78<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MULU A, R0', @RW0+d8	MULU A, MULU A, @RW0+d8	MULUW A, MULUW A, @RW0+d8	MULUW A, MULUW A, @RW0+d8	MUL A, R0', @RW0+d8	MUL A, @RW0+d8	MULW A, MULW A, @RW0+d8	MULW A, MULW A, @RW0+d8	DIVU A, R0', @RW0+d8	DIVU A, @RW0+d8	DIVUW A, DIVUW A, @RW0+d8	DIVUW A, DIVUW A, @RW0+d8	DIV A, R0', @RW0+d8	DIV A, @RW0+d8	DIVW A, DIVW A, @RW0+d8	DIVW A, DIVW A, @RW0+d8
+1	MULU A, R1', @RW1+d8	MULU A, MULU A, @RW1+d8	MULUW A, MULUW A, @RW1+d8	MULUW A, MULUW A, @RW1+d8	MUL A, R1', @RW1+d8	MUL A, @RW1+d8	MULW A, MULW A, @RW1+d8	MULW A, MULW A, @RW1+d8	DIVU A, R1', @RW1+d8	DIVU A, @RW1+d8	DIVUW A, DIVUW A, @RW1+d8	DIVUW A, DIVUW A, @RW1+d8	DIV A, R1', @RW1+d8	DIV A, @RW1+d8	DIVW A, DIVW A, @RW1+d8	DIVW A, DIVW A, @RW1+d8
+2	MULU A, R2', @RW2+d8	MULU A, MULU A, @RW2+d8	MULUW A, MULUW A, @RW2+d8	MULUW A, MULUW A, @RW2+d8	MUL A, R2', @RW2+d8	MUL A, @RW2+d8	MULW A, MULW A, @RW2+d8	MULW A, MULW A, @RW2+d8	DIVU A, R2', @RW2+d8	DIVU A, @RW2+d8	DIVUW A, DIVUW A, @RW2+d8	DIVUW A, DIVUW A, @RW2+d8	DIV A, R2', @RW2+d8	DIV A, @RW2+d8	DIVW A, DIVW A, @RW2+d8	DIVW A, DIVW A, @RW2+d8
+3	MULU A, R3', @RW3+d8	MULU A, MULU A, @RW3+d8	MULUW A, MULUW A, @RW3+d8	MULUW A, MULUW A, @RW3+d8	MUL A, R3', @RW3+d8	MUL A, @RW3+d8	MULW A, MULW A, @RW3+d8	MULW A, MULW A, @RW3+d8	DIVU A, R3', @RW3+d8	DIVU A, @RW3+d8	DIVUW A, DIVUW A, @RW3+d8	DIVUW A, DIVUW A, @RW3+d8	DIV A, R3', @RW3+d8	DIV A, @RW3+d8	DIVW A, DIVW A, @RW3+d8	DIVW A, DIVW A, @RW3+d8
+4	MULU A, R4', @RW4+d8	MULU A, MULU A, @RW4+d8	MULUW A, MULUW A, @RW4+d8	MULUW A, MULUW A, @RW4+d8	MUL A, R4', @RW4+d8	MUL A, @RW4+d8	MULW A, MULW A, @RW4+d8	MULW A, MULW A, @RW4+d8	DIVU A, R4', @RW4+d8	DIVU A, @RW4+d8	DIVUW A, DIVUW A, @RW4+d8	DIVUW A, DIVUW A, @RW4+d8	DIV A, R4', @RW4+d8	DIV A, @RW4+d8	DIVW A, DIVW A, @RW4+d8	DIVW A, DIVW A, @RW4+d8
+5	MULU A, R5', @RW5+d8	MULU A, MULU A, @RW5+d8	MULUW A, MULUW A, @RW5+d8	MULUW A, MULUW A, @RW5+d8	MUL A, R5', @RW5+d8	MUL A, @RW5+d8	MULW A, MULW A, @RW5+d8	MULW A, MULW A, @RW5+d8	DIVU A, R5', @RW5+d8	DIVU A, @RW5+d8	DIVUW A, DIVUW A, @RW5+d8	DIVUW A, DIVUW A, @RW5+d8	DIV A, R5', @RW5+d8	DIV A, @RW5+d8	DIVW A, DIVW A, @RW5+d8	DIVW A, DIVW A, @RW5+d8
+6	MULU A, R6', @RW6+d8	MULU A, MULU A, @RW6+d8	MULUW A, MULUW A, @RW6+d8	MULUW A, MULUW A, @RW6+d8	MUL A, R6', @RW6+d8	MUL A, @RW6+d8	MULW A, MULW A, @RW6+d8	MULW A, MULW A, @RW6+d8	DIVU A, R6', @RW6+d8	DIVU A, @RW6+d8	DIVUW A, DIVUW A, @RW6+d8	DIVUW A, DIVUW A, @RW6+d8	DIV A, R6', @RW6+d8	DIV A, @RW6+d8	DIVW A, DIVW A, @RW6+d8	DIVW A, DIVW A, @RW6+d8
+7	MULU A, R7', @RW7+d8	MULU A, MULU A, @RW7+d8	MULUW A, MULUW A, @RW7+d8	MULUW A, MULUW A, @RW7+d8	MUL A, R7', @RW7+d8	MUL A, @RW7+d8	MULW A, MULW A, @RW7+d8	MULW A, MULW A, @RW7+d8	DIVU A, R7', @RW7+d8	DIVU A, @RW7+d8	DIVUW A, DIVUW A, @RW7+d8	DIVUW A, DIVUW A, @RW7+d8	DIV A, R7', @RW7+d8	DIV A, @RW7+d8	DIVW A, DIVW A, @RW7+d8	DIVW A, DIVW A, @RW7+d8
+8	MULU A, @RW0	MULU A, MULU A, @RW0	MULUW A, MULUW A, @RW0	MULUW A, MULUW A, @RW0	MUL A, @RW0	MUL A, @RW0	MULW A, MULW A, @RW0	MULW A, MULW A, @RW0	DIVU A, @RW0	DIVU A, @RW0	DIVUW A, DIVUW A, @RW0	DIVUW A, DIVUW A, @RW0	DIV A, @RW0	DIV A, @RW0	DIVW A, DIVW A, @RW0	DIVW A, DIVW A, @RW0
+9	MULU A, @RW1	MULU A, MULU A, @RW1	MULUW A, MULUW A, @RW1	MULUW A, MULUW A, @RW1	MUL A, @RW1	MUL A, @RW1	MULW A, MULW A, @RW1	MULW A, MULW A, @RW1	DIVU A, @RW1	DIVU A, @RW1	DIVUW A, DIVUW A, @RW1	DIVUW A, DIVUW A, @RW1	DIV A, @RW1	DIV A, @RW1	DIVW A, DIVW A, @RW1	DIVW A, DIVW A, @RW1
+A	MULU A, @RW2	MULU A, MULU A, @RW2	MULUW A, MULUW A, @RW2	MULUW A, MULUW A, @RW2	MUL A, @RW2	MUL A, @RW2	MULW A, MULW A, @RW2	MULW A, MULW A, @RW2	DIVU A, @RW2	DIVU A, @RW2	DIVUW A, DIVUW A, @RW2	DIVUW A, DIVUW A, @RW2	DIV A, @RW2	DIV A, @RW2	DIVW A, DIVW A, @RW2	DIVW A, DIVW A, @RW2
+B	MULU A, @RW3	MULU A, MULU A, @RW3	MULUW A, MULUW A, @RW3	MULUW A, MULUW A, @RW3	MUL A, @RW3	MUL A, @RW3	MULW A, MULW A, @RW3	MULW A, MULW A, @RW3	DIVU A, @RW3	DIVU A, @RW3	DIVUW A, DIVUW A, @RW3	DIVUW A, DIVUW A, @RW3	DIV A, @RW3	DIV A, @RW3	DIVW A, DIVW A, @RW3	DIVW A, DIVW A, @RW3
+C	MULU A, @RW0+	MULU A, MULU A, @RW0+	MULUW A, MULUW A, @RW0+	MULUW A, MULUW A, @RW0+	MUL A, @RW0+	MUL A, @RW0+	MULW A, MULW A, @RW0+	MULW A, MULW A, @RW0+	DIVU A, @RW0+	DIVU A, @RW0+	DIVUW A, DIVUW A, @RW0+	DIVUW A, DIVUW A, @RW0+	DIV A, @RW0+	DIV A, @RW0+	DIVW A, DIVW A, @RW0+	DIVW A, DIVW A, @RW0+
+D	MULU A, @RW1+	MULU A, MULU A, @RW1+	MULUW A, MULUW A, @RW1+	MULUW A, MULUW A, @RW1+	MUL A, @RW1+	MUL A, @RW1+	MULW A, MULW A, @RW1+	MULW A, MULW A, @RW1+	DIVU A, @RW1+	DIVU A, @RW1+	DIVUW A, DIVUW A, @RW1+	DIVUW A, DIVUW A, @RW1+	DIV A, @RW1+	DIV A, @RW1+	DIVW A, DIVW A, @RW1+	DIVW A, DIVW A, @RW1+
+E	MULU A, @RW2+	MULU A, MULU A, @RW2+	MULUW A, MULUW A, @RW2+	MULUW A, MULUW A, @RW2+	MUL A, @RW2+	MUL A, @RW2+	MULW A, MULW A, @RW2+	MULW A, MULW A, @RW2+	DIVU A, @RW2+	DIVU A, @RW2+	DIVUW A, DIVUW A, @RW2+	DIVUW A, DIVUW A, @RW2+	DIV A, @RW2+	DIV A, @RW2+	DIVW A, DIVW A, @RW2+	DIVW A, DIVW A, @RW2+
+F	MULU A, @RW3+	MULU A, MULU A, @RW3+	MULUW A, MULUW A, @RW3+	MULUW A, MULUW A, @RW3+	MUL A, @RW3+	MUL A, @RW3+	MULW A, MULW A, @RW3+	MULW A, MULW A, @RW3+	DIVU A, @RW3+	DIVU A, @RW3+	DIVUW A, DIVUW A, @RW3+	DIVUW A, DIVUW A, @RW3+	DIV A, @RW3+	DIV A, @RW3+	DIVW A, DIVW A, @RW3+	DIVW A, DIVW A, @RW3+



表 B.9-15 MOVEA RWi, ea 命令 (第1バイト = 79<sub>H</sub>)[illegible]

表 B.9-16 MOV Ri, ea 命令 (第1バイト = 7A<sub>H</sub>)[illegible]

表 B.9-17 MOVW RWi, ea 命令 (第1バイト=7B<sub>H</sub>)[illegible]

表 B.9-18 MOV Ri, ea 命令 (第1バイト = 7C<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0, @RW0+d8, R0	MOV R0, R1, @RW0-d8, R1	MOV R0, R1, @RW0-d8, R1	MOV R0, R2, @RW0+d8, R2	MOV R0, R3, @RW0+d8, R3	MOV R0, R4, @RW0+d8, R4	MOV R0, R5, @RW0+d8, R5	MOV R0, R6, @RW0+d8, R6	MOV R0, R7, @RW0+d8, R7	MOV R0, R8, @RW0+d8, R8	MOV R0, R9, @RW0+d8, R9	MOV R0, R10, @RW0+d8, R10	MOV R0, R11, @RW0+d8, R11	MOV R0, R12, @RW0+d8, R12	MOV R0, R13, @RW0+d8, R13	MOV R0, R14, @RW0+d8, R14
+1	MOV R1, R0, @RW1+d8, R0	MOV R1, R1, @RW1+d8, R1	MOV R1, R2, @RW1+d8, R2	MOV R1, R3, @RW1+d8, R3	MOV R1, R4, @RW1+d8, R4	MOV R1, R5, @RW1+d8, R5	MOV R1, R6, @RW1+d8, R6	MOV R1, R7, @RW1+d8, R7	MOV R1, R8, @RW1+d8, R8	MOV R1, R9, @RW1+d8, R9	MOV R1, R10, @RW1+d8, R10	MOV R1, R11, @RW1+d8, R11	MOV R1, R12, @RW1+d8, R12	MOV R1, R13, @RW1+d8, R13	MOV R1, R14, @RW1+d8, R14	MOV R1, R15, @RW1+d8, R15
+2	MOV R2, R0, @RW2+d8, R0	MOV R2, R1, @RW2+d8, R1	MOV R2, R2, @RW2+d8, R2	MOV R2, R3, @RW2+d8, R3	MOV R2, R4, @RW2+d8, R4	MOV R2, R5, @RW2+d8, R5	MOV R2, R6, @RW2+d8, R6	MOV R2, R7, @RW2+d8, R7	MOV R2, R8, @RW2+d8, R8	MOV R2, R9, @RW2+d8, R9	MOV R2, R10, @RW2+d8, R10	MOV R2, R11, @RW2+d8, R11	MOV R2, R12, @RW2+d8, R12	MOV R2, R13, @RW2+d8, R13	MOV R2, R14, @RW2+d8, R14	MOV R2, R15, @RW2+d8, R15
+3	MOV R3, R0, @RW3+d8, R0	MOV R3, R1, @RW3+d8, R1	MOV R3, R2, @RW3+d8, R2	MOV R3, R3, @RW3+d8, R3	MOV R3, R4, @RW3+d8, R4	MOV R3, R5, @RW3+d8, R5	MOV R3, R6, @RW3+d8, R6	MOV R3, R7, @RW3+d8, R7	MOV R3, R8, @RW3+d8, R8	MOV R3, R9, @RW3+d8, R9	MOV R3, R10, @RW3+d8, R10	MOV R3, R11, @RW3+d8, R11	MOV R3, R12, @RW3+d8, R12	MOV R3, R13, @RW3+d8, R13	MOV R3, R14, @RW3+d8, R14	MOV R3, R15, @RW3+d8, R15
+4	MOV R4, R0, @RW4+d8, R0	MOV R4, R1, @RW4+d8, R1	MOV R4, R2, @RW4+d8, R2	MOV R4, R3, @RW4+d8, R3	MOV R4, R4, @RW4+d8, R4	MOV R4, R5, @RW4+d8, R5	MOV R4, R6, @RW4+d8, R6	MOV R4, R7, @RW4+d8, R7	MOV R4, R8, @RW4+d8, R8	MOV R4, R9, @RW4+d8, R9	MOV R4, R10, @RW4+d8, R10	MOV R4, R11, @RW4+d8, R11	MOV R4, R12, @RW4+d8, R12	MOV R4, R13, @RW4+d8, R13	MOV R4, R14, @RW4+d8, R14	MOV R4, R15, @RW4+d8, R15
+5	MOV R5, R0, @RW5+d8, R0	MOV R5, R1, @RW5+d8, R1	MOV R5, R2, @RW5+d8, R2	MOV R5, R3, @RW5+d8, R3	MOV R5, R4, @RW5+d8, R4	MOV R5, R5, @RW5+d8, R5	MOV R5, R6, @RW5+d8, R6	MOV R5, R7, @RW5+d8, R7	MOV R5, R8, @RW5+d8, R8	MOV R5, R9, @RW5+d8, R9	MOV R5, R10, @RW5+d8, R10	MOV R5, R11, @RW5+d8, R11	MOV R5, R12, @RW5+d8, R12	MOV R5, R13, @RW5+d8, R13	MOV R5, R14, @RW5+d8, R14	MOV R5, R15, @RW5+d8, R15
+6	MOV R6, R0, @RW6+d8, R0	MOV R6, R1, @RW6+d8, R1	MOV R6, R2, @RW6+d8, R2	MOV R6, R3, @RW6+d8, R3	MOV R6, R4, @RW6+d8, R4	MOV R6, R5, @RW6+d8, R5	MOV R6, R6, @RW6+d8, R6	MOV R6, R7, @RW6+d8, R7	MOV R6, R8, @RW6+d8, R8	MOV R6, R9, @RW6+d8, R9	MOV R6, R10, @RW6+d8, R10	MOV R6, R11, @RW6+d8, R11	MOV R6, R12, @RW6+d8, R12	MOV R6, R13, @RW6+d8, R13	MOV R6, R14, @RW6+d8, R14	MOV R6, R15, @RW6+d8, R15
+7	MOV R7, R0, @RW7+d8, R0	MOV R7, R1, @RW7+d8, R1	MOV R7, R2, @RW7+d8, R2	MOV R7, R3, @RW7+d8, R3	MOV R7, R4, @RW7+d8, R4	MOV R7, R5, @RW7+d8, R5	MOV R7, R6, @RW7+d8, R6	MOV R7, R7, @RW7+d8, R7	MOV R7, R8, @RW7+d8, R8	MOV R7, R9, @RW7+d8, R9	MOV R7, R10, @RW7+d8, R10	MOV R7, R11, @RW7+d8, R11	MOV R7, R12, @RW7+d8, R12	MOV R7, R13, @RW7+d8, R13	MOV R7, R14, @RW7+d8, R14	MOV R7, R15, @RW7+d8, R15
+8	MOV @RW0, R0, @RW0+d16, R0	MOV @RW0, R1, @RW0+d16, R1	MOV @RW0, R2, @RW0+d16, R2	MOV @RW0, R3, @RW0+d16, R3	MOV @RW0, R4, @RW0+d16, R4	MOV @RW0, R5, @RW0+d16, R5	MOV @RW0, R6, @RW0+d16, R6	MOV @RW0, R7, @RW0+d16, R7	MOV @RW0, R8, @RW0+d16, R8	MOV @RW0, R9, @RW0+d16, R9	MOV @RW0, R10, @RW0+d16, R10	MOV @RW0, R11, @RW0+d16, R11	MOV @RW0, R12, @RW0+d16, R12	MOV @RW0, R13, @RW0+d16, R13	MOV @RW0, R14, @RW0+d16, R14	MOV @RW0, R15, @RW0+d16, R15
+9	MOV @RW1, R0, @RW1+d16, R0	MOV @RW1, R1, @RW1+d16, R1	MOV @RW1, R2, @RW1+d16, R2	MOV @RW1, R3, @RW1+d16, R3	MOV @RW1, R4, @RW1+d16, R4	MOV @RW1, R5, @RW1+d16, R5	MOV @RW1, R6, @RW1+d16, R6	MOV @RW1, R7, @RW1+d16, R7	MOV @RW1, R8, @RW1+d16, R8	MOV @RW1, R9, @RW1+d16, R9	MOV @RW1, R10, @RW1+d16, R10	MOV @RW1, R11, @RW1+d16, R11	MOV @RW1, R12, @RW1+d16, R12	MOV @RW1, R13, @RW1+d16, R13	MOV @RW1, R14, @RW1+d16, R14	MOV @RW1, R15, @RW1+d16, R15
+A	MOV @RW2, R0, @RW2+d16, R0	MOV @RW2, R1, @RW2+d16, R1	MOV @RW2, R2, @RW2+d16, R2	MOV @RW2, R3, @RW2+d16, R3	MOV @RW2, R4, @RW2+d16, R4	MOV @RW2, R5, @RW2+d16, R5	MOV @RW2, R6, @RW2+d16, R6	MOV @RW2, R7, @RW2+d16, R7	MOV @RW2, R8, @RW2+d16, R8	MOV @RW2, R9, @RW2+d16, R9	MOV @RW2, R10, @RW2+d16, R10	MOV @RW2, R11, @RW2+d16, R11	MOV @RW2, R12, @RW2+d16, R12	MOV @RW2, R13, @RW2+d16, R13	MOV @RW2, R14, @RW2+d16, R14	MOV @RW2, R15, @RW2+d16, R15
+B	MOV @RW3, R0, @RW3+d16, R0	MOV @RW3, R1, @RW3+d16, R1	MOV @RW3, R2, @RW3+d16, R2	MOV @RW3, R3, @RW3+d16, R3	MOV @RW3, R4, @RW3+d16, R4	MOV @RW3, R5, @RW3+d16, R5	MOV @RW3, R6, @RW3+d16, R6	MOV @RW3, R7, @RW3+d16, R7	MOV @RW3, R8, @RW3+d16, R8	MOV @RW3, R9, @RW3+d16, R9	MOV @RW3, R10, @RW3+d16, R10	MOV @RW3, R11, @RW3+d16, R11	MOV @RW3, R12, @RW3+d16, R12	MOV @RW3, R13, @RW3+d16, R13	MOV @RW3, R14, @RW3+d16, R14	MOV @RW3, R15, @RW3+d16, R15
+C	MOV @RW0+, R0, @RW0+RW7, R0	MOV @RW0+, R1, @RW0+RW7, R1	MOV @RW0+, R2, @RW0+RW7, R2	MOV @RW0+, R3, @RW0+RW7, R3	MOV @RW0+, R4, @RW0+RW7, R4	MOV @RW0+, R5, @RW0+RW7, R5	MOV @RW0+, R6, @RW0+RW7, R6	MOV @RW0+, R7, @RW0+RW7, R7	MOV @RW0+, R8, @RW0+RW7, R8	MOV @RW0+, R9, @RW0+RW7, R9	MOV @RW0+, R10, @RW0+RW7, R10	MOV @RW0+, R11, @RW0+RW7, R11	MOV @RW0+, R12, @RW0+RW7, R12	MOV @RW0+, R13, @RW0+RW7, R13	MOV @RW0+, R14, @RW0+RW7, R14	MOV @RW0+, R15, @RW0+RW7, R15
+D	MOV @RW1+, R0, @RW1+RW7, R0	MOV @RW1+, R1, @RW1+RW7, R1	MOV @RW1+, R2, @RW1+RW7, R2	MOV @RW1+, R3, @RW1+RW7, R3	MOV @RW1+, R4, @RW1+RW7, R4	MOV @RW1+, R5, @RW1+RW7, R5	MOV @RW1+, R6, @RW1+RW7, R6	MOV @RW1+, R7, @RW1+RW7, R7	MOV @RW1+, R8, @RW1+RW7, R8	MOV @RW1+, R9, @RW1+RW7, R9	MOV @RW1+, R10, @RW1+RW7, R10	MOV @RW1+, R11, @RW1+RW7, R11	MOV @RW1+, R12, @RW1+RW7, R12	MOV @RW1+, R13, @RW1+RW7, R13	MOV @RW1+, R14, @RW1+RW7, R14	MOV @RW1+, R15, @RW1+RW7, R15
+E	MOV @RW2+, R0, @PC+d16, R0	MOV @RW2+, R1, @PC+d16, R1	MOV @RW2+, R2, @PC+d16, R2	MOV @RW2+, R3, @PC+d16, R3	MOV @RW2+, R4, @PC+d16, R4	MOV @RW2+, R5, @PC+d16, R5	MOV @RW2+, R6, @PC+d16, R6	MOV @RW2+, R7, @PC+d16, R7	MOV @RW2+, R8, @PC+d16, R8	MOV @RW2+, R9, @PC+d16, R9	MOV @RW2+, R10, @PC+d16, R10	MOV @RW2+, R11, @PC+d16, R11	MOV @RW2+, R12, @PC+d16, R12	MOV @RW2+, R13, @PC+d16, R13	MOV @RW2+, R14, @PC+d16, R14	MOV @RW2+, R15, @PC+d16, R15
+F	MOV @RW3+, R0, addr16, R0	MOV @RW3+, R1, addr16, R1	MOV @RW3+, R2, addr16, R2	MOV @RW3+, R3, addr16, R3	MOV @RW3+, R4, addr16, R4	MOV @RW3+, R5, addr16, R5	MOV @RW3+, R6, addr16, R6	MOV @RW3+, R7, addr16, R7	MOV @RW3+, R8, addr16, R8	MOV @RW3+, R9, addr16, R9	MOV @RW3+, R10, addr16, R10	MOV @RW3+, R11, addr16, R11	MOV @RW3+, R12, addr16, R12	MOV @RW3+, R13, addr16, R13	MOV @RW3+, R14, addr16, R14	MOV @RW3+, R15, addr16, R15

表 B.9-19 MOVW RWi, ea 命令 (第1バイト = 7D<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW RW0, RW0, @RW0-d8, RW0	MOVW RW0, RW1, @RW0-d8, RW1	MOVW RW0, RW2, @RW0-d8, RW2	MOVW RW0, RW3, @RW0-d8, RW3	MOVW RW0, RW4, @RW0-d8, RW4	MOVW RW0, RW5, @RW0-d8, RW5	MOVW RW0, RW6, @RW0-d8, RW6	MOVW RW0, RW7, @RW0-d8, RW7	MOVW RW0, RW4, @RW0-d8, RW4	MOVW RW0, RW5, @RW0-d8, RW5	MOVW RW0, RW6, @RW0-d8, RW6	MOVW RW0, RW7, @RW0-d8, RW7	MOVW RW0, RW8, @RW0-d8, RW8	MOVW RW0, RW9, @RW0-d8, RW9	MOVW RW0, RW10, @RW0-d8, RW10	MOVW RW0, RW11, @RW0-d8, RW11
+1	MOVW RW1, RW0, @RW1-d8, RW0	MOVW RW1, RW1, @RW1-d8, RW1	MOVW RW1, RW2, @RW1-d8, RW2	MOVW RW1, RW3, @RW1-d8, RW3	MOVW RW1, RW4, @RW1-d8, RW4	MOVW RW1, RW5, @RW1-d8, RW5	MOVW RW1, RW6, @RW1-d8, RW6	MOVW RW1, RW7, @RW1-d8, RW7	MOVW RW1, RW4, @RW1-d8, RW4	MOVW RW1, RW5, @RW1-d8, RW5	MOVW RW1, RW6, @RW1-d8, RW6	MOVW RW1, RW7, @RW1-d8, RW7	MOVW RW1, RW8, @RW1-d8, RW8	MOVW RW1, RW9, @RW1-d8, RW9	MOVW RW1, RW10, @RW1-d8, RW10	MOVW RW1, RW11, @RW1-d8, RW11
+2	MOVW RW2, RW0, @RW2-d8, RW0	MOVW RW2, RW1, @RW2-d8, RW1	MOVW RW2, RW2, @RW2-d8, RW2	MOVW RW2, RW3, @RW2-d8, RW3	MOVW RW2, RW4, @RW2-d8, RW4	MOVW RW2, RW5, @RW2-d8, RW5	MOVW RW2, RW6, @RW2-d8, RW6	MOVW RW2, RW7, @RW2-d8, RW7	MOVW RW2, RW4, @RW2-d8, RW4	MOVW RW2, RW5, @RW2-d8, RW5	MOVW RW2, RW6, @RW2-d8, RW6	MOVW RW2, RW7, @RW2-d8, RW7	MOVW RW2, RW8, @RW2-d8, RW8	MOVW RW2, RW9, @RW2-d8, RW9	MOVW RW2, RW10, @RW2-d8, RW10	MOVW RW2, RW11, @RW2-d8, RW11
+3	MOVW RW3, RW0, @RW3-d8, RW0	MOVW RW3, RW1, @RW3-d8, RW1	MOVW RW3, RW2, @RW3-d8, RW2	MOVW RW3, RW3, @RW3-d8, RW3	MOVW RW3, RW4, @RW3-d8, RW4	MOVW RW3, RW5, @RW3-d8, RW5	MOVW RW3, RW6, @RW3-d8, RW6	MOVW RW3, RW7, @RW3-d8, RW7	MOVW RW3, RW4, @RW3-d8, RW4	MOVW RW3, RW5, @RW3-d8, RW5	MOVW RW3, RW6, @RW3-d8, RW6	MOVW RW3, RW7, @RW3-d8, RW7	MOVW RW3, RW8, @RW3-d8, RW8	MOVW RW3, RW9, @RW3-d8, RW9	MOVW RW3, RW10, @RW3-d8, RW10	MOVW RW3, RW11, @RW3-d8, RW11
+4	MOVW RW4, RW0, @RW4-d8, RW0	MOVW RW4, RW1, @RW4-d8, RW1	MOVW RW4, RW2, @RW4-d8, RW2	MOVW RW4, RW3, @RW4-d8, RW3	MOVW RW4, RW4, @RW4-d8, RW4	MOVW RW4, RW5, @RW4-d8, RW5	MOVW RW4, RW6, @RW4-d8, RW6	MOVW RW4, RW7, @RW4-d8, RW7	MOVW RW4, RW4, @RW4-d8, RW4	MOVW RW4, RW5, @RW4-d8, RW5	MOVW RW4, RW6, @RW4-d8, RW6	MOVW RW4, RW7, @RW4-d8, RW7	MOVW RW4, RW8, @RW4-d8, RW8	MOVW RW4, RW9, @RW4-d8, RW9	MOVW RW4, RW10, @RW4-d8, RW10	MOVW RW4, RW11, @RW4-d8, RW11
+5	MOVW RW5, RW0, @RW5-d8, RW0	MOVW RW5, RW1, @RW5-d8, RW1	MOVW RW5, RW2, @RW5-d8, RW2	MOVW RW5, RW3, @RW5-d8, RW3	MOVW RW5, RW4, @RW5-d8, RW4	MOVW RW5, RW5, @RW5-d8, RW5	MOVW RW5, RW6, @RW5-d8, RW6	MOVW RW5, RW7, @RW5-d8, RW7	MOVW RW5, RW4, @RW5-d8, RW4	MOVW RW5, RW5, @RW5-d8, RW5	MOVW RW5, RW6, @RW5-d8, RW6	MOVW RW5, RW7, @RW5-d8, RW7	MOVW RW5, RW8, @RW5-d8, RW8	MOVW RW5, RW9, @RW5-d8, RW9	MOVW RW5, RW10, @RW5-d8, RW10	MOVW RW5, RW11, @RW5-d8, RW11
+6	MOVW RW6, RW0, @RW6-d8, RW0	MOVW RW6, RW1, @RW6-d8, RW1	MOVW RW6, RW2, @RW6-d8, RW2	MOVW RW6, RW3, @RW6-d8, RW3	MOVW RW6, RW4, @RW6-d8, RW4	MOVW RW6, RW5, @RW6-d8, RW5	MOVW RW6, RW6, @RW6-d8, RW6	MOVW RW6, RW7, @RW6-d8, RW7	MOVW RW6, RW4, @RW6-d8, RW4	MOVW RW6, RW5, @RW6-d8, RW5	MOVW RW6, RW6, @RW6-d8, RW6	MOVW RW6, RW7, @RW6-d8, RW7	MOVW RW6, RW8, @RW6-d8, RW8	MOVW RW6, RW9, @RW6-d8, RW9	MOVW RW6, RW10, @RW6-d8, RW10	MOVW RW6, RW11, @RW6-d8, RW11
+7	MOVW RW7, RW0, @RW7-d8, RW0	MOVW RW7, RW1, @RW7-d8, RW1	MOVW RW7, RW2, @RW7-d8, RW2	MOVW RW7, RW3, @RW7-d8, RW3	MOVW RW7, RW4, @RW7-d8, RW4	MOVW RW7, RW5, @RW7-d8, RW5	MOVW RW7, RW6, @RW7-d8, RW6	MOVW RW7, RW7, @RW7-d8, RW7	MOVW RW7, RW4, @RW7-d8, RW4	MOVW RW7, RW5, @RW7-d8, RW5	MOVW RW7, RW6, @RW7-d8, RW6	MOVW RW7, RW7, @RW7-d8, RW7	MOVW RW7, RW8, @RW7-d8, RW8	MOVW RW7, RW9, @RW7-d8, RW9	MOVW RW7, RW10, @RW7-d8, RW10	MOVW RW7, RW11, @RW7-d8, RW11
+8	MOVW @RW0, RW0, @RW0-d16, RW0	MOVW @RW0, RW1, +d16, RW1	MOVW @RW0, RW2, +d16, RW2	MOVW @RW0, RW3, +d16, RW3	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW7, +d16, RW7	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW7, +d16, RW7	MOVW @RW0, RW8, +d16, RW8	MOVW @RW0, RW9, +d16, RW9	MOVW @RW0, RW10, +d16, RW10	MOVW @RW0, RW11, +d16, RW11
+9	MOVW @RW1, RW0, @RW1-d16, RW0	MOVW @RW1, RW1, +d16, RW1	MOVW @RW1, RW2, +d16, RW2	MOVW @RW1, RW3, +d16, RW3	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW7, +d16, RW7	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW7, +d16, RW7	MOVW @RW1, RW8, +d16, RW8	MOVW @RW1, RW9, +d16, RW9	MOVW @RW1, RW10, +d16, RW10	MOVW @RW1, RW11, +d16, RW11
+A	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW8, +d16, RW8	MOVW @RW2, RW9, +d16, RW9	MOVW @RW2, RW10, +d16, RW10	MOVW @RW2, RW11, +d16, RW11
+B	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW8, +d16, RW8	MOVW @RW3, RW9, +d16, RW9	MOVW @RW3, RW10, +d16, RW10	MOVW @RW3, RW11, +d16, RW11
+C	MOVW @RW0+, RW0, +RW7, RW0	MOVW @RW0+, RW1, +RW7, RW1	MOVW @RW0+, RW2, +RW7, RW2	MOVW @RW0+, RW3, +RW7, RW3	MOVW @RW0+, RW4, +RW7, RW4	MOVW @RW0+, RW5, +RW7, RW5	MOVW @RW0+, RW6, +RW7, RW6	MOVW @RW0+, RW7, +RW7, RW7	MOVW @RW0+, RW4, +RW7, RW4	MOVW @RW0+, RW5, +RW7, RW5	MOVW @RW0+, RW6, +RW7, RW6	MOVW @RW0+, RW7, +RW7, RW7	MOVW @RW0+, RW8, +RW7, RW8	MOVW @RW0+, RW9, +RW7, RW9	MOVW @RW0+, RW10, +RW7, RW10	MOVW @RW0+, RW11, +RW7, RW11
+D	MOVW @RW1+, RW0, +RW7, RW0	MOVW @RW1+, RW1, +RW7, RW1	MOVW @RW1+, RW2, +RW7, RW2	MOVW @RW1+, RW3, +RW7, RW3	MOVW @RW1+, RW4, +RW7, RW4	MOVW @RW1+, RW5, +RW7, RW5	MOVW @RW1+, RW6, +RW7, RW6	MOVW @RW1+, RW7, +RW7, RW7	MOVW @RW1+, RW4, +RW7, RW4	MOVW @RW1+, RW5, +RW7, RW5	MOVW @RW1+, RW6, +RW7, RW6	MOVW @RW1+, RW7, +RW7, RW7	MOVW @RW1+, RW8, +RW7, RW8	MOVW @RW1+, RW9, +RW7, RW9	MOVW @RW1+, RW10, +RW7, RW10	MOVW @RW1+, RW11, +RW7, RW11
+E	MOVW @RW2+, RW0, +d16, RW0	MOVW @RW2+, RW1, +d16, RW1	MOVW @RW2+, RW2, +d16, RW2	MOVW @RW2+, RW3, +d16, RW3	MOVW @RW2+, RW4, +d16, RW4	MOVW @RW2+, RW5, +d16, RW5	MOVW @RW2+, RW6, +d16, RW6	MOVW @RW2+, RW7, +d16, RW7	MOVW @RW2+, RW4, +d16, RW4	MOVW @RW2+, RW5, +d16, RW5	MOVW @RW2+, RW6, +d16, RW6	MOVW @RW2+, RW7, +d16, RW7	MOVW @RW2+, RW8, +d16, RW8	MOVW @RW2+, RW9, +d16, RW9	MOVW @RW2+, RW10, +d16, RW10	MOVW @RW2+, RW11, +d16, RW11
+F	MOVW @RW3+, RW0, +d16, RW0	MOVW @RW3+, RW1, +d16, RW1	MOVW @RW3+, RW2, +d16, RW2	MOVW @RW3+, RW3, +d16, RW3	MOVW @RW3+, RW4, +d16, RW4	MOVW @RW3+, RW5, +d16, RW5	MOVW @RW3+, RW6, +d16, RW6	MOVW @RW3+, RW7, +d16, RW7	MOVW @RW3+, RW4, +d16, RW4	MOVW @RW3+, RW5, +d16, RW5	MOVW @RW3+, RW6, +d16, RW6	MOVW @RW3+, RW7, +d16, RW7	MOVW @RW3+, RW8, +d16, RW8	MOVW @RW3+, RW9, +d16, RW9	MOVW @RW3+, RW10, +d16, RW10	MOVW @RW3+, RW11, +d16, RW11

表 B.9-20 XCH Ri, ea 命令 (第1バイト = 7EH)

[illegible]

**表 B.9-21 XCHW RWi, ea 命令 (第 1 バイト = 7FH)**

[illegible]

## 索引

## Numerics

16 ビット PPG タイマ	
16 ビット PPG タイマ ( × 3 )	258
16 ビット PPG タイマ使用上の注意	277
16 ビット PPG タイマ端子のブロックダイヤ グラム	261
16 ビット PPG タイマの EI <sup>2</sup> OS 機能	272
16 ビット PPG タイマの端子	260
16 ビット PPG タイマのプログラム例	278
16 ビット PPG タイマのブロックダイヤグラム	259
16 ビット PPG タイマのレジスタ	262
16 ビット PPG タイマ割込み	271
16 ビット PPG タイマ割込みと EI <sup>2</sup> OS	272
16 ビットアウトプットコンペア	
16 ビットアウトプットコンペア ( × 6 )	282
16 ビットアウトプットコンペア使用上の注意	360
16 ビットアウトプットコンペアタイミング	345
16 ビットアウトプットコンペアの動作	341
16 ビットアウトプットコンペアのプログラム例	364
16 ビットアウトプットコンペアのブロック ダイヤグラム	286
16 ビットアウトプットコンペアのレジスタ	291
16 ビットアウトプットコンペア割込み	331
16 ビットアウトプットコンペア割込みと EI <sup>2</sup> OS	331
16 ビットインプットキャプチャ	
16 ビットインプットキャプチャ ( × 4 )	283
16 ビットインプットキャプチャ使用上の注意	361
16 ビットインプットキャプチャ入力タイミング	347
16 ビットインプットキャプチャの動作	346
16 ビットインプットキャプチャのブロックダイヤ グラム	286
16 ビットインプットキャプチャのレジスタ	292
16 ビットインプットキャプチャ割込み	332
16 ビットインプットキャプチャ割込みと EI <sup>2</sup> OS	332
16 ビットタイマ	
16 ビットタイマ使用上の注意	440
16 ビットタイマのタイミング	436
16 ビットタイマの動作	435
16 ビットタイマのブロックダイヤグラム	375
マルチパルスジェネレータの 16 ビットタイマの 使用	438
マルチパルスジェネレータの 16 ビットタイマの 動作ダイヤグラム	438
16 ビットタイマバッファ	
16 ビットタイマバッファ動作のタイミングダイヤ グラム	437
16 ビットタイマレジスタ	
16 ビットタイマレジスタ上位, 下位 (TMRH0,TMRH1, TMRL0,TMRL1)	239
16 ビットデッドタイム制御レジスタ	
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR0)	319
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)	325
16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)	322
16 ビットデッドタイムレジスタ	
16 ビットデッドタイムレジスタ (TMRRH,TMRRLO ~ TMRRLO2)	318
16 ビットフリーランタイム	
16 ビットフリーランタイム ( × 1 )	282
16 ビットフリーランタイム使用上の注意	360
16 ビットフリーランタイムのプログラム例	363
16 ビットフリーランタイムのブロックダイヤ グラム	285
16 ビットフリーランタイムのレジスタ	290
16 ビットフリーランタイム割込み	330
16 ビットフリーランタイム割込みと EI <sup>2</sup> OS	330
16 ビットリロードタイマ	
16 ビットリロードタイマ使用上の注意	252
16 ビットリロードタイマ端子のブロックダイヤ グラム	233
16 ビットリロードタイマの設定	242
16 ビットリロードタイマの端子	233
16 ビットリロードタイマの動作モード	228
16 ビットリロードタイマのブロックダイヤグラム	231
16 ビットリロードタイマのレジスタ一覧	234
16 ビットリロードタイマの割込み	241
16 ビットリロードタイマの割込みと EI <sup>2</sup> OS	230
16 ビットリロードタイマの割込みと EI <sup>2</sup> OS 機能	241
16 ビットリロードタイマアンダフロー	
16 ビットリロードタイマアンダフロー方式	420
位置検出と 16 ビットリロードタイマアンダフロー 方式	424
位置検出または 16 ビットリロードタイマアンダ フロー方式	426
ワンショット位置検出と 16 ビットリロードタイマ アンダフロー方式	429
ワンショット位置検出または 16 ビットリロード タイマアンダフロー方式	430
16 ビットリロードタイマ出力	
内部タイマ (16 ビットリロードタイマ出力) に よるポーレート	505
16 ビットリロードレジスタ	
16 ビットリロードレジスタ上位, 下位 (TMRDH0,TMRDH1,TMRDLO,TMRDLO1)	240



512K ビットフラッシュメモリ	
512K ビットフラッシュメモリの概要 .....	606
512K ビットフラッシュメモリのセクタ構成 .....	607
512K ビットフラッシュメモリの特長 .....	606
512K ビットフラッシュメモリのプログラム例 .....	632
8/10 ビット A/D コンバータ	
8/10 ビット A/D コンバータ使用上の注意 .....	576
8/10 ビット A/D コンバータの EI <sup>2</sup> OS 機能 .....	569
8/10 ビット A/D コンバータの機能 .....	554
8/10 ビット A/D コンバータの端子 .....	558
8/10 ビット A/D コンバータの端子のブロック ダイアグラム .....	559
8/10 ビット A/D コンバータのブロックダイア グラム .....	556
8/10 ビット A/D コンバータのレジスタ構成 .....	560
8/10 ビット A/D コンバータの割込み .....	569
8/10 ビット A/D コンバータの割込みと EI <sup>2</sup> OS .....	569
8/10 ビット A/D コンバータ割込みと EI <sup>2</sup> OS .....	555

## A

A/D 制御ステータスレジスタ	
A/D 制御ステータスレジスタ 0(ADCS0) .....	564
A/D 制御ステータスレジスタ 1(ADCS1) .....	561
A/D データレジスタ	
A/D データレジスタ (ADCR0,ADCR1) .....	567
A/D 変換データ保護機能	
A/D 変換データ保護機能 .....	574
ADCR	
A/D データレジスタ (ADCR0,ADCR1) .....	567
ADCS	
A/D 制御ステータスレジスタ 0(ADCS0) .....	564
A/D 制御ステータスレジスタ 1(ADCS1) .....	561

## C

CDCR	
通信プリスケラ制御レジスタ 0,1 (CDCR0,CDCR1) .....	494
CKSCR	
クロック選択レジスタ (CKSCR) .....	84
CPCLRBH,CPCLRBL	
コンペアクリアバッファレジスタ (CPCLRBH,CPCLRBL) .....	294
CPCLRH,CPCLRL	
コンペアクリアレジスタ (CPCLRH,CPCLRL) .....	294
CPCR, CPCRL	
コンペアクリアレジスタ上位, 下位 (CPCR, CPCRL) .....	398
CPU	
CPU の概要 .....	28
CPU 間欠動作モード	
CPU 間欠動作モード .....	93, 100
CPU 動作モード	
CPU 動作モードと消費電流 .....	92

## D

DIP-64P-M01	
DIP-64P-M01 の外形寸法図 .....	11
端子配列図 (DIP-64P-M01) .....	10
DIRR	
遅延割込み要求発生 / 解除レジスタ (DIRR) .....	549
DIV	
分周比制御レジスタ (DIV0,DIV1) .....	455
DIV A,Ri	
「DIV A,Ri」, 「DIVW A,RWi」命令の使用上の注意 .....	64
DIVW A,RWi	
「DIV A,Ri」, 「DIVW A,RWi」命令の使用上の注意 .....	64
DTCR	
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR0) .....	319
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2) .....	325
16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1) .....	322

DTP/ 外部割込み	
DTP/ 外部割込み機能	526
DTP/ 外部割込み使用上の注意	540
DTP/ 外部割込み動作	536
DTP/ 外部割込みの設定	535
DTP/ 外部割込みの端子	529
DTP/ 外部割込みのブロックダイヤグラム	527
DTP/ 外部割込み機能	
DTP/ 外部割込み機能のプログラム例	542
DTP/ 外部割込み許可レジスタ	
DTP/ 外部割込み許可レジスタ (ENIR)	531
DTP/ 外部割込み要因レジスタ	
DTP/ 外部割込み要因レジスタ (EIRR)	530
DTP 機能	
DTP 機能	539
DTP 機能のプログラム例	543
DTTI	
DTTI と OPTx 出力の関係	433
DTTI 端子入力	
DTTI 端子入力の動作	358
DTTI 入力制御	
DTTI 入力制御の動作	431
DTTI 割込み	
DTTI 割込み	359
<b>E</b>	
E <sup>2</sup> PROM	
E <sup>2</sup> PROM のメモリマップ	594
システム構成と E <sup>2</sup> PROM のメモリ構成	594
修正プログラムを E <sup>2</sup> PROM に格納する場合の アドレス一致検出機能の動作	596
EI <sup>2</sup> OS	
16 ビット PPG タイマの EI <sup>2</sup> OS 機能	272
16 ビット PPG タイマ割込みと EI <sup>2</sup> OS	272
8/10 ビット A/D コンバータの割込みと EI <sup>2</sup> OS	569
8/10 ビット A/D コンバータ割込みと EI <sup>2</sup> OS	555
EI <sup>2</sup> OS を使用した場合の変換動作	573
PWC タイマ割込みと EI <sup>2</sup> OS	456
タイムベースタイマ割込みと EI <sup>2</sup> OS	211
マルチパルスジェネレータの割込みと EI <sup>2</sup> OS	406
EIRR	
DTP/ 外部割込み要因レジスタ (EIRR)	530
ELVRH	
検出レベル設定レジスタ, 上位 (ELVRH)	533
ELVRL	
検出レベル設定レジスタ, 下位 (ELVRL)	534
ENIR	
DTP/ 外部割込み許可レジスタ (ENIR)	531
<b>F</b>	
F <sup>2</sup> MC-16LX	
F <sup>2</sup> MC-16LX 命令一覧表	685
FMCS	
フラッシュメモリ制御ステータスレジスタ (FMCS)	608
FPT-64P-M06	
FPT-64P-M06 の外形寸法図	12

端子配列図 (FPT-64P-M06)	8
FPT-64P-M09	
FPT-64P-M09 の外形寸法図	13
端子配列図 (FPT-64P-M09)	9
<b>I</b>	
I/O 回路	
I/O 回路の種類	19
I/O 端子	
I/O 端子と端子機能	14
I/O ポート	
I/O ポート関連のレジスタ	164
I/O ポートのプログラム例	203
I/O ポートの機能	
I/O ポートの機能	162
I/O マップ	
I/O マップ	650
I/O 領域	
I/O 領域	30
I/O レジスタアドレスポインタ	
I/O レジスタアドレスポインタ (IOA)	141
ICR	
割込み制御レジスタ (ICR)	122
割込み制御レジスタ (ICR00 ~ ICR15)	120
ICSH	
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 上位バイト (ICSH23)	311
ICSL	
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 下位バイト (ICSL23)	312
IPCLR	
入力制御レジスタ, 下位 (IPCLR)	396
IPCPH,IPCPL	
インプットキャプチャデータレジスタ (IPCPH,IPCPL0 ~ IPCPL3)	310
IPCUR	
入力制御レジスタ, 上位 (IPCUR)	394
<b>L</b>	
LPMCR	
低消費電力モード制御レジスタ (LPMCR)	97
<b>N</b>	
NCCR	
ノイズキャンセル制御レジスタ (NCCR)	400
<b>O</b>	
OCCPBH,OCCPBL	
アウトプットコンペアバッファレジスタ (OCCPBH,OCCPBL0 ~ OCCPBL5)	302
OCCPH,OCCPL	
アウトプットコンペアレジスタ (OCCPH,OCCPL0 ~ OCCPL5)	303
OCSH	
コンペア制御レジスタ, 上位バイト (OCSH1,OCSH3,OCSH5)	304

OCSL	
コンペア制御レジスタ, 下位バイト (OCSL0,OCSL2,OCSL4) .....	307
OPCLR	
出力制御レジスタ, 下位 (OPCLR) .....	386
OPCUR	
出力制御レジスタ, 上位 (OPCUR) .....	384
OPDBR	
出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBRB) .....	393
出力データバッファレジスタ, 上位 (OPDBR0 ~ OPDBRB) .....	391
OPDBR0 書込み	
OPDBR0 書込み方式 .....	419
OPDRH	
出力データレジスタ, 上位 (OPDRH) .....	388
OPDRH	
出力データレジスタ上位, 下位 (OPDRH, OPDRL) .....	408
OPDRL	
出力データレジスタ, 下位 (OPDRL) .....	390
<b>P</b>	
PACSR	
プログラムアドレス検出制御ステータスレジスタ (PACSR) .....	589
PADR	
プログラムアドレス検出レジスタ (PADR0H,M,L,PADR1H,M,L) .....	591
PCNTH	
PPG 状態制御レジスタ, 上位バイト (PCNTH0 ~ PCNTH2) .....	267
PCNTL	
PPG 状態制御レジスタ, 下位バイト (PCNTL0 ~ PCNTL2) .....	269
PCSRH,PCSRL	
PPG 周期設定バッファレジスタ (PCSRH,PCSRL0 ~ PCSRL2) .....	265
PDCRH,PDCRL	
PPG ダウンカウンタレジスタ (PDCRH,PDCRL0 ~ PDCRL2) .....	264
PDUTH,PDUTL	
PPG デューティ設定バッファレジスタ (PDUTH,PDUTL0 ~ PDUTL2) .....	266
PICSH	
PPG 出力制御 / インプットキャプチャ状態制御 レジスタ (ch.0, ch.1), 上位バイト (PICSH01) .....	314
PICSL	
インプットキャプチャ状態制御レジスタ (ch.0,ch.1), 下位バイト (PICSL01) .....	316
PLL クロック通倍率	
PLL クロック通倍率の設定 .....	86
PLL クロックモード	
メインクロックモードと PLL クロックモード .....	86
PPG0 出力	
ゲートトリガされた PPG0 出力 .....	350
PPG0 出力制御	
PPG0 出力制御 .....	350

PPG 周期設定バッファレジスタ	
PPG 周期設定バッファレジスタ (PCSRH,PCSRL0 ~ PCSRL2) .....	265
PPG 出力制御 / インプットキャプチャ状態 制御レジスタ	
PPG 出力制御 / インプットキャプチャ状態制御 レジスタ (ch.0, ch.1), 上位バイト (PICSH01) .....	314
PPG 状態制御レジスタ	
PPG 状態制御レジスタ, 下位バイト (PCNTL0 ~ PCNTL2) .....	269
PPG 状態制御レジスタ, 上位バイト (PCNTH0 ~ PCNTH2) .....	267
PPG ダウンカウンタレジスタ	
PPG ダウンカウンタレジスタ (PDCRH,PDCRL0 ~ PDCRL2) .....	264
PPG デューティ設定バッファレジスタ	
PPG デューティ設定バッファレジスタ (PDUTH,PDUTL0 ~ PDUTL2) .....	266
PPG 割込み	
PPG 割込み .....	276
PWC0 タイマ	
PWC0 タイマ端子のブロックダイアグラム .....	446
PWC1 タイマ	
PWC1 タイマ端子のブロックダイアグラム .....	447
PWCSH	
PWC 状態制御レジスタ, 上位バイト (PWCSH0,PWCSH1) .....	449
PWCSL	
PWC 状態制御レジスタ, 下位バイト (PWCSL0,PWCSL1) .....	452
PWC 状態制御レジスタ	
PWC 状態制御レジスタ, 下位バイト (PWCSL0,PWCSL1) .....	452
PWC 状態制御レジスタ, 上位バイト (PWCSH0,PWCSH1) .....	449
PWC タイマ	
PWC タイマ使用上の注意 .....	472
PWC タイマの EI <sup>2</sup> OS 機能 .....	457
PWC タイマの端子 .....	446
PWC タイマの動作 .....	444
PWC タイマの特性 .....	444
PWC タイマのプログラム例 .....	474
PWC タイマのブロックダイアグラム .....	445
PWC タイマのレジスタ .....	448
PWC タイマの割込み .....	456
PWC タイマ割込みと EI <sup>2</sup> OS .....	456
PWC データバッファレジスタ	
PWC データバッファレジスタ (PWCH0,PWCH1,PWCL0,PWCL1) .....	454
PWM モード	
PWM モード (PCNTH レジスタの MDSE:bit13=0) .....	273

**R**

RAM 領域	
RAM 領域 .....	30
ROM 領域	
ROM 領域 .....	29

ROMM	
ROM ミラー機能選択モジュールレジスタ (ROMM) .....	602
ROM ミラー機能	
ROM ミラー機能による FF バンクへのアクセス .....	600
ROM ミラー機能の有効 / 無効時のメモリ空間 .....	601
ROM 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のブロックダイアグラム .....	600
ROM ミラー機能選択モジュールレジスタ	
ROM ミラー機能選択モジュールレジスタ (ROMM) .....	602
RTO0	
RTO0 ~ RTO5 と GATE の出力状態 .....	348
RTO5	
RTO0 ~ RTO5 と GATE の出力状態 .....	348
RUN モード	
RUN モード .....	156
<b>S</b>	
SCR	
シリアル制御レジスタ 0,1(SCR0,SCR1) .....	485
SIDR	
シリアル入力データレジスタ 0,1(SIDR0,SIDR1) .....	492
SIGCR	
波形制御レジスタ (SIGCR) .....	328
SMR	
シリアルモードレジスタ 0,1(SMR0,SMR1) .....	487
SODR	
シリアル出力データレジスタ 0,1(SODR0,SODR1) .....	493
SSR	
シリアルステータスレジスタ 0,1(SSR0,SSR1) .....	489
<b>T</b>	
TBTC	
タイムベースタイマ制御レジスタ (TBTC) .....	209
TCCSH	
タイマ状態制御レジスタ, 上位バイト (TCCSH) .....	296
TCCSL	
タイマ状態制御レジスタ, 下位バイト (TCCSL) .....	299
TCDTH,TCDTL	
タイマデータレジスタ (TCDTH,TCDTL) .....	295
TCSR	
タイマ状態制御レジスタ (TCSR) .....	402
TMBRH, TMBRL	
タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL) .....	399
TMCSRH	
タイマ制御状態レジスタ上位 (TMCSRH0,TMCSRH1) .....	235
TMCSRL	
タイマ制御状態レジスタ下位 (TMCSRL0,TMCSRL1) .....	237

TMRDH,TMRDL	
16 ビットリロードレジスタ上位, 下位 (TMRDH0,TMRDH1,TMRDL0,TMRDL1) .....	240
TMRH,TMRL	
16 ビットタイマレジスタ上位, 下位 (TMRH0,TMRH1,TMRL0,TMRL1) .....	239
TMRRH,TMRRL	
16 ビットデッドタイマレジスタ (TMRRH,TMRRL0 ~ TMRRL2) .....	318
<b>U</b>	
UART	
UART 使用上の注意 .....	521
UART での割り込み要求の発生 .....	484
UART の EI <sup>2</sup> OS 機能 .....	496
UART の機能 .....	478
UART の端子 .....	483
UART の端子のブロックダイアグラム .....	483
UART の動作 .....	508
UART のプログラム例 .....	522
UART のブロックダイアグラム .....	480
UART のボーレートの選択 .....	500
UART のレジスター一覧 .....	484
UART の割り込み .....	495
UART の割り込みと EI <sup>2</sup> OS .....	496

**W**

WDTC	
ウォッチドッグタイマ制御レジスタ (WDTC) .....	220
WTIN1 出力状態	
WTIN1 出力状態とレジスタ設定 .....	410

## あ

アウトプットコンペアバッファレジスタ	
アウトプットコンペアバッファレジスタ	
(OCCPBH, OCCPBL0 ~ OCCPBL5) .....	302
アウトプットコンペアレジスタ	
アウトプットコンペアレジスタ	
(OCCPH, OCCPL0 ~ OCCPL5) .....	303
アキュムレータ	
アキュムレータ (A) .....	42
アクセス空間	
バンクレジスタとアクセス空間 .....	35
アドレス一致検出機能	
アドレス一致検出機能の概要 .....	586
アドレス一致検出機能の動作 .....	593
アドレス一致検出機能のプログラム例 .....	598
アドレス一致検出機能のブロックダイアグラム	
.....	587
アドレス一致検出機能のレジスタとリセット値の	
一覧 .....	588
修正プログラムを E <sup>2</sup> PROM に格納する場合の	
アドレス一致検出機能の動作 .....	596
アドレス引用指定	
32 ビット汎用レジスタの下位 24 ビットアドレス	
引用指定 (間接指定) .....	34
アドレス指定	
バンク方式によるアドレス指定とデフォルト空間	
.....	36
アドレス直接指定	
オペランドによる 24 ビットアドレス直接指定	
.....	34
アドレッシング	
アドレッシング .....	662
間接アドレッシング .....	671
直接アドレッシング .....	664

## い

位置検出	
位置検出と 16 ビットリロードタイマアンダフロー	
方式 .....	424
位置検出の動作 .....	409
位置検出方式 .....	422
位置検出または 16 ビットリロードタイマアンダ	
フロー方式 .....	426
位置検出回路	
位置検出回路のブロックダイアグラム .....	377
イベントカウントモード	
イベントカウントモード .....	250
イベントカウントモード (外部クロックモード)	
.....	229
イベントカウントモードのプログラム例 .....	254
インターバルタイマ機能	
インターバルタイマ機能 .....	206
インターバルタイマ機能 (タイムベースタイマ) の	
動作 .....	212
インプットキャプチャ状態制御レジスタ	
インプットキャプチャ状態制御レジスタ	
(ch.2, ch.3), 下位バイト (ICSL23) .....	312
インプットキャプチャ状態制御レジスタ	
(ch.2, ch.3), 上位バイト (ICSH23) .....	311

インプットキャプチャ状態制御レジスタ	
(ch.0, ch.1), 下位バイト (PICSL01) .....	316
インプットキャプチャデータレジスタ	
インプットキャプチャデータレジスタ	
(IPCPH, IPCPL0 ~ IPCPL3) .....	310

## う

ウォッチドッグタイマ	
ウォッチドッグタイマ使用上の注意 .....	224
ウォッチドッグタイマの動作 .....	222
ウォッチドッグタイマのプログラム例 .....	225
ウォッチドッグタイマのブロックダイアグラム	
.....	219
ウォッチドッグタイマ機能	
ウォッチドッグタイマ機能 .....	218
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制御レジスタ (WDTC)	
.....	220

## お

オペランド	
オペランドによる 24 ビットアドレス直接指定	
.....	34

## か

外形寸法図	
DIP-64P-M01 の外形寸法図 .....	11
FPT-64P-M06 の外形寸法図 .....	12
FPT-64P-M09 の外形寸法図 .....	13
外部カウントクロック	
選択された外部カウントクロック .....	340
外部クロック	
外部クロックによるボーレート .....	507
振動子と外部クロックの接続 .....	89
外部クロックモード	
イベントカウントモード (外部クロックモード)	
.....	229
外部リセット端子	
外部リセット端子のブロックダイアグラム .....	71
外部割込み機能	
外部割込み機能 .....	538
カウンタ	
カウンタの動作状態 .....	243
カウンタ動作	
カウンタ動作 .....	229
カウントクロック周期	
カウントクロック周期と最大周期 .....	465
拡張インテリジェント I/O サービス	
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS)	
.....	138
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の	
使用手順 .....	145
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の	
処理時間 (1 回のデータ転送に要する時間)	
.....	146
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の	
ステータスレジスタ (ISCS) .....	142

拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の ディスクリプタ (ISD) の構成 .....	140
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の 動作 .....	139
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の 動作フロー .....	144
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の プログラム例 .....	152
間接アドレッシング 間接アドレッシング .....	671

## く

クロック クロックの概要 .....	80
クロック供給機能 クロック供給機能 .....	207, 213
クロック供給マップ クロック供給マップ .....	81
クロック選択レジスタ クロック選択レジスタ (CKSCR) .....	84
クロック同期モード クロック同期モードでの動作 .....	514
クロック発生部 クロック発生部のブロックダイアグラム .....	82
クロックモード クロックモード .....	93
クロックモードの切換え .....	112
クロックモードの遷移 .....	86

## け

ゲートトリガ ゲートトリガ (PPG チャンネル 0 の場合のみ) .....	275
検出アドレス 検出アドレスの設定 .....	593
検出レベル設定レジスタ 検出レベル設定レジスタ, 下位 (ELVRL) .....	534
検出レベル設定レジスタ, 上位 (ELVRH) .....	533

## こ

コマンドシーケンス コマンドシーケンス表 .....	611
コモンレジスタバンクプリフィックス コモンレジスタバンクプリフィックス (CMR) .....	60
コンディションコードレジスタ コンディションコードレジスタ (CCR) の構成 .....	48
コンペアクリアバッファ コンペアクリアバッファ .....	336
コンペアクリアバッファレジスタ コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL) .....	294
コンペアクリアレジスタ コンペアクリアレジスタ (CPCLRH, CPCLRL) .....	294
コンペアクリアレジスタ上位, 下位 (CPCRH, CPCRL) .....	398

コンペア制御レジスタ コンペア制御レジスタ, 下位バイト (OCSL0, OCSL2, OCSL4) .....	307
コンペア制御レジスタ, 上位バイト (OCSH1, OCSH3, OCSH5) .....	304

## さ

最小入力パルス幅 最小入力パルス幅 .....	468
----------------------------	-----

## し

システムスタックポインタ システムスタックポインタ (SSP) .....	46
実効アドレス 実効アドレスフィールド .....	680
実効アドレスフィールド 実効アドレスフィールド .....	663
実行サイクル 実行サイクル数 .....	678
実行サイクル数計算方法 .....	678
修正プログラム 修正プログラムを E <sup>2</sup> PROM に格納する場合の アドレス一致検出機能の動作 .....	596
受信割込み発生 受信割込み発生とフラグ設定のタイミング .....	497
出力状態 RTO0 ~ RTO5 と GATE の出力状態 .....	348
出力制御レジスタ 出力制御レジスタ, 下位 (OPCLR) .....	386
出力制御レジスタ, 上位 (OPCUR) .....	384
出力データバッファレジスタ 出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBRB) .....	393
出力データバッファレジスタ, 上位 (OPDBR0 ~ OPDBRB) .....	391
出力データバッファレジスタの動作 .....	415
出力データレジスタ 出力データレジスタ, 下位 (OPDRL) .....	390
出力データレジスタ, 上位 (OPDRH) .....	388
出力データレジスタ上位, 下位 (OPDRH, OPDRL) .....	408
出力データレジスタのブロックダイアグラム .....	407
出力データレジスタへのデータ転送動作 .....	417
状態遷移図 状態遷移図 .....	109
消費電流 CPU 動作モードと消費電流 .....	92
シリアル書込み MB90F462/MB90F462A/MB90F463A シリアル書込 み接続の基本構成 .....	638
シリアルクロック入力周波数 発振クロック周波数とシリアルクロック入力周波 数 .....	640
シリアル出力データレジスタ シリアル出力データレジスタ 0,1 (SODR0, SODR1) .....	493

## 索引

シリアルステータスレジスタ	
シリアルステータスレジスタ 0,1(SSR0,SSR1)	489
シリアル制御レジスタ	
シリアル制御レジスタ 0,1(SCR0,SCR1)	485
シリアル入力データレジスタ	
シリアル入力データレジスタ 0,1(SIDR0,SIDR1)	492
シリアルモードレジスタ	
シリアルモードレジスタ 0,1(SMR0,SMR1)	487
シリアルライタ	
シリアルライタ対応	648
シングルチップモード	
シングルチップモード時の接続例 (フラッシュ マイコンプログラマから電源を供給する場合)	643
シングルチップモード時の接続例 (ユーザ電源を使用する場合)	641
振動子	
振動子と外部クロックの接続	89

## す

スタック	
スタックの設定	45
スタック動作	
割り込み処理開始時のスタック動作	149
割り込み処理からの復帰時のスタック動作	149
スタック領域	
スタック領域	150
スタンバイモード	
スタンバイモード	93
スタンバイモードでの動作状態	101
スタンバイモードの設定	111
スタンバイモードへの遷移と割り込み	111
割り込みによるスタンバイモードの解除	111
ストップモード	
外部割り込みによるストップモードの解除	112
ストップモードの解除	108
ストップモードへの遷移	107
スリープモード	
スリープモードの解除	103
スリープモードへの遷移	102

## せ

製品一覧	
MB90460/465 シリーズの製品一覧	5
セクタ消去	
フラッシュメモリのセクタ消去手順	627
フラッシュメモリのセクタ消去を一時停止する	629
フラッシュメモリのセクタ消去を再開する	630
フラッシュメモリの任意のデータを消去する (セクタ消去)	627
セクタ消去タイマフラグ	
セクタ消去タイマフラグ (DQ3)	619
専用ポーレートジェネレータ	
専用ポーレートジェネレータによるポーレート	502

専用レジスタ	
専用レジスタの構成	40

## そ

送信割り込み発生	
送信割り込み発生とフラグ設定のタイミング	499
双方向通信機能	
双方向通信機能	516
測定結果データ	
測定結果データ	467
測定モード	
測定モードと測定動作	469
ソフトウェアブルアップ抵抗	
ソフトウェアブルアップ抵抗	110
ソフトウェア割り込み	
ソフトウェア割り込みからの復帰	136
ソフトウェア割り込み動作	137
ソフトウェア割り込みの起動	136

## た

タイマ機能	
タイマ機能	458
タイマクリア	
タイマクリア	335, 463
タイマ周期	
タイマ周期	465
タイマ状態制御レジスタ	
タイマ状態制御レジスタ (TCSR)	402
タイマ状態制御レジスタ, 下位バイト (TCCSL)	299
タイマ状態制御レジスタ, 上位バイト (TCCSH)	296
タイマ制御状態レジスタ	
タイマ制御状態レジスタ下位 (TMCSRL0,TMCSRL1)	237
タイマ制御状態レジスタ上位 (TMCSRH0,TMCSRH1)	235
タイマ値	
タイマ値とリロード値	464
タイマデータレジスタ	
タイマデータレジスタ (TCDTH,TCDTL)	295
タイマとパルス幅測定	
タイマとパルス幅測定の起動 / 停止	462
タイマバッファレジスタ	
タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL)	399
タイマモード	
タイマモード	335
タイマモードの動作	352
タイマモード動作	
タイマモード動作のフローチャート	466
タイマ割り込み	
タイマ割り込み	338
タイミングダイアグラム	
16ビットタイマバッファ動作のタイミングダイア グラム	437
タイミングリミット超過フラグ	
タイミングリミット超過フラグ (DQ5)	618

タイムベースタイマ	
タイムベースタイマ使用上の注意	215
タイムベースタイマの動作	214
タイムベースタイマのプログラム例	216
タイムベースタイマのブロックダイアグラム	208
タイムベースタイマ制御レジスタ	
タイムベースタイマ制御レジスタ (TBTC)	209
タイムベースタイマモード	
タイムベースタイマモードの解除	106, 112
タイムベースタイマモードへの遷移	105
タイムベースタイマ割込み	
タイムベースタイマ割込み	211
タイムベースタイマ割込みと EI <sup>2</sup> OS	211
ダイレクトページレジスタ	
ダイレクトページレジスタ (DPR)	53
多機能タイマ	
多機能タイマの EI <sup>2</sup> OS 機能	333
多機能タイマの端子	288
多機能タイマの動作	334
多機能タイマのブロックダイアグラム	284
多機能タイマ端子	
多機能タイマ端子のブロックダイアグラム	289
多バイト長オペランド	
多バイト長オペランドの構成	37
多バイト長データ	
スタック上の多バイト長データの構成	38
多バイト長データへのアクセス	38
メモリ上の多バイト長データの格納状態	37
単一測定モード	
単一測定モードと連続測定モード	467
端子機能	
I/O 端子と端子機能	14
端子配列図	
端子配列図 (DIP-64P-M01)	10
端子配列図 (FPT-64P-M06)	8
端子配列図 (FPT-64P-M09)	9
単発変換モード	
EI <sup>2</sup> OS を使用した場合の単発変換モードにおいて	
A/D 変換を行うためのプログラム例	577
単発変換モード時の動作	570

## ち

遅延割込み発生モジュール	
遅延割込み発生モジュール使用上の注意	551
遅延割込み発生モジュールの概要	546
遅延割込み発生モジュールの動作説明	550
遅延割込み発生モジュールのプログラム例	552
遅延割込み発生モジュールのブロックダイアグラム	547
遅延割込み要求発生 / 解除レジスタ	
遅延割込み要求発生 / 解除レジスタ (DIRR)	549
チップ消去	
フラッシュメモリのデータを消去する	
(チップ消去)	626
直接アドレッシング	
直接アドレッシング	664

## つ

通信プリスケアラ制御レジスタ	
通信プリスケアラ制御レジスタ	
0,1(CDCR0,CDCR1)	494

## て

停止変換モード	
EI <sup>2</sup> OS を使用した場合の停止変換モードにおいて	
A/D 変換を行うためのプログラム例	582
停止変換モード時の動作	572
低消費電力制御回路	
低消費電力制御回路のブロックダイアグラム	95
低消費電力モード制御レジスタ	
低消費電力モード制御レジスタ (LPMCR)	97
低消費電力モード制御レジスタへのアクセス	99
ディスクリプタ	
拡張インテリジェント I/O サービス (EI <sup>2</sup> OS) の	
ディスクリプタ (ISD) の構成	140
データカウンタ	
データカウンタ (DCT)	141
データ書込み制御ユニット	
データ書込み制御ユニットの動作	411
データ書込み制御ユニットのブロックダイアグラム	376
データポーリングフラグ	
データポーリングフラグ (DQ7)	615
デッドタイムタイマモード	
デッドタイムタイマモード時の動作	354
デバイス	
デバイス取扱い上の注意事項	24
デフォルト空間	
バンク方式によるアドレス指定とデフォルト空間	36

## と

動作モード	
動作モード	156
動作モードの選択	461
特長	
MB90460/465 シリーズの特長	2
トグルビット 2 フラグ	
トグルビット 2 フラグ (DQ2)	620
トグルビットフラグ	
トグルビットフラグ (DQ6)	617

## な

内部クロックモード	
内部クロックモード	228
内部クロックモード (リロードモード) の動作	244
内部クロックモード (ワンショットモード)	247
内部クロックモードのプログラム例	253



内部タイマ	
内部タイマ (16 ビットリロードタイマ出力) によるポーレート	505

## に

入力制御レジスタ	
入力制御レジスタ, 下位 (IPCLR)	396
入力制御レジスタ, 上位 (PCUR)	394

## の

ノイズキャンセル機能	
DTTI 端子ノイズキャンセル機能	359
ノイズキャンセル制御レジスタ	
ノイズキャンセル制御レジスタ (NCCR)	400
ノイズ除去機能	
ノイズ除去機能の動作	434

## は

ハードウェアシーケンスフラグ	
ハードウェアシーケンスフラグ	613
ハードウェア割込み	
ハードウェア割込み	125
ハードウェア割込みからの復帰	128
ハードウェア割込み使用手順	131
ハードウェア割込み動作	129
ハードウェア割込みのアクティブ化	128
ハードウェア割込みの構造	126
ハードウェア割込みの処理に要する時間	134
ハードウェア割込みの抑制	126
波形シーケンサ	
波形シーケンサ使用上の注意	439
波形シーケンサの機能	368
波形シーケンサのブロックダイアグラム	373
波形ジェネレータ	
波形ジェネレータ	283
波形ジェネレータ使用上の注意	361
波形ジェネレータのブロックダイアグラム	287
波形ジェネレータのレジスタ	293
波形ジェネレータ割込み	333
波形ジェネレータ割込みと EI <sup>2</sup> OS	333
波形制御レジスタ	
波形制御レジスタ (SIGCR)	328
バスモード	
バスモード	156
バスモード設定ビット	
バスモード設定ビット	158
発振安定待ち時間	
発振安定待ち時間	88, 112
発振安定待ち時間用タイマ機能	213
リセット要因と発振安定待ち時間	70
発振クロック周波数	
発振クロック周波数とシリアルクロック入力周波数	640
パッチ処理	
プログラム修正のパッチ処理フロー	597
バッファアドレスポインタ	
バッファアドレスポインタ (BAP)	142

パルス幅 / 周期	
パルス幅 / 周期測定範囲	468
パルス幅 / 周期の算出	468
パルス幅測定	
パルス幅測定動作のフローチャート	471
パルス幅測定機能	
パルス幅測定機能	460
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 (PCB,DTB,ADB,SPB)	58
バンク方式	
バンク方式によるアドレス指定とデフォルト空間	36
リニア方式とバンク方式のメモリ指定	33
バンクレジスタ	
バンクレジスタ (PCB,DTB,USB,SSB,ADB)	54
バンクレジスタとアクセス空間	35
汎用レジスタ	
32 ビット汎用レジスタの下位 24 ビットアドレス引用指定 (間接指定)	34
汎用レジスタの構成	55
汎用レジスタ領域	
汎用レジスタ領域とレジスタバンクポインタ (RP)	50

## ひ

非同期モード	
非同期モードでの動作	510

## ふ

フラグ設定	
受信割込み発生とフラグ設定のタイミング	497
送信割込み発生とフラグ設定のタイミング	499
フラグ変化抑止プリフィックス	
フラグ変化抑止プリフィックス (NCC)	61
フラッシュマイコンプログラマ	
シングルチップモード時の接続例 (フラッシュマイコンプログラマから電源を供給する場合)	643
最小限の接続例 (フラッシュマイコンプログラマから電源を供給する場合)	647
フラッシュマイコンプログラマとの最小限の接続例 (ユーザ電源を使用する場合)	645
フラッシュマイコンプログラマのシステム構成 (横河デジタルコンピュータ株式会社製)	640
フラッシュメモリ	
フラッシュメモリ書込み手順	624
フラッシュメモリのセクタ消去手順	627
フラッシュメモリのセクタ消去を一時停止する	629
フラッシュメモリのセクタ消去を再開する	630
フラッシュメモリのデータを消去する (チップ消去)	626
フラッシュメモリの任意のデータを消去する (セクタ消去)	627
フラッシュメモリへデータを書き込む	624

フラッシュメモリを読み出し/リセット状態にする	623
フラッシュメモリ書き込み/消去	
フラッシュメモリ書き込み/消去	606
フラッシュメモリ書き込み/消去の詳細説明	622
フラッシュメモリ制御ステータスレジスタ	
フラッシュメモリ制御ステータスレジスタ (FMCS)	608
プリフィックスコード	
プリフィックスコード	57
プリフィックスコードと割込み抑止命令	62
プリフィックスコードの連続	63
プログラム	
プログラムの実行	593
プログラムアドレス検出制御ステータスレジスタ	
プログラムアドレス検出制御ステータスレジスタ (PACSR)	589
プログラムアドレス検出レジスタ	
プログラムアドレス検出レジスタ (PADR0H,M,L,PADR1H,M,L)	591
プログラムアドレス検出レジスタの機能	592
プログラムカウンタ	
プログラムカウンタ (PC)	52
プログラム修正	
プログラム修正のパッチ処理フロー	597
プロセッサステータス	
プロセッサステータス (PS) の構成	47
ブロックダイアグラム	
16ビット PPG タイマ端子のブロックダイアグラム	261
16ビット PPG タイマのブロックダイアグラム	259
16ビットアウトプットコンペアのブロックダイアグラム	286
16ビットインプットキャプチャのブロックダイアグラム	286
16ビットタイマのブロックダイアグラム	375
16ビットフリーランタイマのブロックダイアグラム	285
16ビットリロードタイマ端子のブロックダイアグラム	233
16ビットリロードタイマのブロックダイアグラム	231
8/10ビット A/D コンバータの端子のブロックダイアグラム	559
8/10ビット A/D コンバータのブロックダイアグラム	556
DTP/ 外部割込みのブロックダイアグラム	527
MB90460/465 シリーズの全体ブロックダイアグラム	7
PWC0 タイマ端子のブロックダイアグラム	446
PWC1 タイマ端子のブロックダイアグラム	447
PWC タイマのブロックダイアグラム	445
ROM ミラー機能選択モジュールのブロックダイアグラム	600
UART の端子のブロックダイアグラム	483
UART のブロックダイアグラム	480
アドレス一致検出機能のブロックダイアグラム	587
位置検出回路のブロックダイアグラム	377

ウォッチドッグタイマのブロックダイアグラム	219
外部リセット端子のブロックダイアグラム	71
クロック発生部のブロックダイアグラム	82
出力データレジスタのブロックダイアグラム	407
タイムベースタイマのブロックダイアグラム	208
多機能タイマ端子のブロックダイアグラム	289
多機能タイマのブロックダイアグラム	284
遅延割込み発生モジュールのブロックダイアグラム	547
低消費電力制御回路のブロックダイアグラム	95
データ書き込み制御ユニットのブロックダイアグラム	376
波形シーケンサのブロックダイアグラム	373
波形ジェネレータのブロックダイアグラム	287
ポート 0 の端子のブロックダイアグラム	166
ポート 1 の端子のブロックダイアグラム	172
ポート 2 の端子のブロックダイアグラム	178
ポート 3 の端子のブロックダイアグラム	183
ポート 4 の端子のブロックダイアグラム	188
ポート 5 の端子のブロックダイアグラム	193
ポート 6 の端子のブロックダイアグラム	199
マルチパルスジェネレータ端子のブロックダイアグラム	380
マルチパルスジェネレータのブロックダイアグラム	372
分周比制御レジスタ	
分周比制御レジスタ (DIV0,DIV1)	455

## ほ

ポート 0	
ポート 0 の構成	165
ポート 0 の端子	165
ポート 0 の端子のブロックダイアグラム	166
ポート 0 の動作	169
ポート 0 のレジスタ	166
ポート 0 のレジスタの機能	167
ポート 1	
ポート 1 の構成	171
ポート 1 の端子	171
ポート 1 の端子のブロックダイアグラム	172
ポート 1 の動作	175
ポート 1 のレジスタ	172
ポート 1 のレジスタの機能	173
ポート 2	
ポート 2 の構成	177
ポート 2 の端子	177
ポート 2 の端子のブロックダイアグラム	178
ポート 2 の動作	180
ポート 2 のレジスタ	178
ポート 2 のレジスタの機能	179
ポート 3	
ポート 3 の構成	182
ポート 3 の端子	182
ポート 3 の端子のブロックダイアグラム	183
ポート 3 の動作	185

## 索引

ポート 3 のレジスタ .....	183
ポート 3 のレジスタの機能 .....	184
ポート 4 .....	
ポート 4 の構成 .....	187
ポート 4 の端子 .....	187
ポート 4 の端子のブロックダイアグラム .....	188
ポート 4 の動作 .....	190
ポート 4 のレジスタ .....	188
ポート 4 のレジスタの機能 .....	189
ポート 5 .....	
ポート 5 の構成 .....	192
ポート 5 の端子 .....	192
ポート 5 の端子のブロックダイアグラム .....	193
ポート 5 の動作 .....	196
ポート 5 のレジスタ .....	193
ポート 5 のレジスタの機能 .....	194
ポート 6 .....	
ポート 6 の構成 .....	198
ポート 6 の端子 .....	198
ポート 6 の端子のブロックダイアグラム .....	199
ポート 6 の動作 .....	201
ポート 6 のレジスタ .....	199
ポート 6 のレジスタの機能 .....	200
ボーレート .....	
UART のボーレートの選択 .....	500
外部クロックによるボーレート .....	507
専用ボーレートジェネレータによるボーレート .....	502
内部タイマ (16 ビットリロードタイマ出力) によるボーレート .....	505

## ま

マシクロック .....	
マシクロック .....	87
マスタ/スレーブ型通信機能 .....	
マスタ/スレーブ型通信機能 .....	518
マルチパルスジェネレータ .....	
マルチパルスジェネレータ端子のブロックダイアグラム .....	380
マルチパルスジェネレータの 16 ビットタイマの使用 .....	438
マルチパルスジェネレータの 16 ビットタイマの動作ダイアグラム .....	438
マルチパルスジェネレータの EI <sup>2</sup> OS 機能 .....	406
マルチパルスジェネレータの端子 .....	379
マルチパルスジェネレータのプログラム例 .....	441
マルチパルスジェネレータのブロックダイアグラム .....	372
マルチパルスジェネレータのレジスタ .....	381
マルチパルスジェネレータの割込みと EI <sup>2</sup> OS .....	406
マルチパルスジェネレータの割込み要因 .....	405
マルチパルス割込み .....	
マルチパルス割込み .....	404

## め

命令 .....	
命令の種類 .....	661
命令マップの構造 .....	699

命令一覧表 .....	
F <sup>2</sup> MC-16LX 命令一覧表 .....	685
メインクロックモード .....	
メインクロックモードと PLL クロックモード .....	86
メモリ空間 .....	
ROM ミラー機能の有効 / 無効時のメモリ空間 .....	601
メモリ空間 .....	29
メモリ指定 .....	
リニア方式とバンク方式のメモリ指定 .....	33
メモリマップ .....	
E <sup>2</sup> PROM のメモリマップ .....	594
メモリマップ .....	31

## も

モード設定 .....	
モード設定 .....	156
モード端子 .....	
モード端子 .....	72
モード端子 (MD2 ~ MD0) .....	157
モード端子とモードデータの関係 .....	159
モードデータ .....	
モード端子とモードデータの関係 .....	159
モードデータの取込み .....	72
モードデータ読出し後の端子状態 .....	77
モードデータレジスタ .....	
モードデータレジスタ .....	158

## ゆ

ユーザスタックポインタ .....	
ユーザスタックポインタ (USP) .....	46
ユーザ電源 .....	
シングルチップモード時の接続例 (ユーザ電源を使用する場合) .....	641
フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源を使用する場合) .....	645

## り

リセット .....	
リセットによる各端子の状態 .....	77
リセット動作 .....	
リセット動作の概要 .....	72
リセット要因 .....	
リセット要因 .....	68
リセット要因と発振安定待ち時間 .....	70
リセット要因フラグビット .....	
リセット要因フラグビット .....	74
リセット要因フラグビットとリセット要因の対応 .....	75
リセット要因フラグビットの注意事項 .....	75
リニア方式 .....	
リニア方式とバンク方式のメモリ指定 .....	33
リロード値 .....	
タイマ値とリロード値 .....	464
リロード動作モード .....	
リロード動作モード .....	464

リロードモード	
内部クロックモード (リロードモード) の動作	244

## れ

例外処理	
例外処理	148
レジスタ	
16 ビットアウトプットコンペアのレジスタ	291
16 ビットインプットキャプチャのレジスタ	292
16 ビットタイマレジスタ上位, 下位 (TMRH0,TMRH1, TMRL0,TMRL1)	239
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR0)	319
16 ビットデッドタイム制御レジスタ, 下位バイト (DTCR2)	325
16 ビットデッドタイム制御レジスタ, 上位バイト (DTCR1)	322
16 ビットデッドタイムレジスタ (TMRRH, TMRRL0 ~ TMRRL2)	318
16 ビットフリーランタイムのレジスタ	290
16 ビットリロードレジスタ上位, 下位 (TMRDH0,TMRDH1,TMRDL0,TMRDL1)	240
A/D 制御ステータスレジスタ 0(ADCS0)	564
A/D 制御ステータスレジスタ 1(ADCS1)	561
A/D データレジスタ (ADCR0,ADCR1)	567
DTP/ 外部割込み許可レジスタ (ENIR)	531
DTP/ 外部割込み要因レジスタ (EIRR)	530
I/O ポート関連のレジスタ	164
PPG 周期設定バッファレジスタ (PCSRH,PCSR0 ~ PCSRL2)	265
PPG 出力制御 / インプットキャプチャ状態制御 レジスタ (ch.0, ch.1), 上位バイト (PICSH01)	314
PPG 状態制御レジスタ, 下位バイト (PCNTL0 ~ PCNTL2)	269
PPG 状態制御レジスタ, 上位バイト (PCNTH0 ~ PCNTH2)	267
PPG ダウンカウンタレジスタ (PDCRH,PDCRL0 ~ PDCRL2)	264
PPG デューティ設定バッファレジスタ (PDUTH,PDUTL0 ~ PDUTL2)	266
PWC 状態制御レジスタ, 下位バイト (PWCSL0,PWCSL1)	452
PWC 状態制御レジスタ, 上位バイト (PWCSH0,PWCSH1)	449
ROM ミラー機能選択モジュールレジスタ (ROMM)	602
アウトプットコンペアバッファレジスタ (OCCPBH,OCCPBL0 ~ OCCPBL5)	302
アウトプットコンペアレジスタ (OCCPH,OCCPL0 ~ OCCPL5)	303
インプットキャプチャ状態制御レジスタ (ch.0,ch.1), 下位バイト (PICSL01)	316
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 下位バイト (ICSL23)	312
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 上位バイト (ICSH23)	311

インプットキャプチャデータレジスタ (IPCPH,IPCPL0 ~ IPCPL3)	310
ウォッチドッグタイマ制御レジスタ (WDTC)	220
クロック選択レジスタ (CKSCR)	84
検出レベル設定レジスタ, 下位 (ELVRL)	534
検出レベル設定レジスタ, 上位 (ELVRH)	533
コンディションコードレジスタ (CCR) の構成	48
コンペアクリアバッファレジスタ (CPCLR0,CPCLR1)	294
コンペアクリアレジスタ (CPCLR0,CPCLR1)	294
コンペアクリアレジスタ上位, 下位 (CPCR0, CPCR1)	398
コンペア制御レジスタ, 下位バイト (OCSL0,OCSL2,OCSL4)	307
コンペア制御レジスタ, 上位バイト (OCSH1,OCSH3,OCSH5)	304
出力制御レジスタ, 下位 (OPCLR)	386
出力制御レジスタ, 上位 (OPCUR)	384
出力データバッファレジスタ, 下位 (OPDBR0 ~ OPDBR3)	393
出力データバッファレジスタ, 上位 (OPDBR4 ~ OPDBR7)	391
出力データレジスタ, 下位 (OPDRL)	390
出力データレジスタ, 上位 (OPDRH)	388
出力データレジスタ上位, 下位 (OPDRH, OPDRL)	408
シリアル出力データレジスタ 0,1(SODR0,SODR1)	493
シリアルステータスレジスタ 0,1(SSR0,SSR1)	489
シリアル制御レジスタ 0,1(SCR0,SCR1)	485
シリアル入力データレジスタ 0,1(SIDR0,SIDR1)	492
シリアルモードレジスタ 0,1(SMR0,SMR1)	487
タイマ状態制御レジスタ (TCSR)	402
タイマ状態制御レジスタ, 下位バイト (TCCSL)	299
タイマ状態制御レジスタ, 上位バイト (TCCSH)	296
タイマ制御状態レジスタ下位 (TMCSRL0,TMCSRL1)	237
タイマ制御状態レジスタ上位 (TMCSR0,TMCSR1)	235
タイマデータレジスタ (TCDTH,TCDTL)	295
タイマバッファレジスタ上位, 下位 (TMBRH, TMBRL)	399
タイムベースタイマ制御レジスタ (TBTC)	209
ダイレクトページレジスタ (DPR)	53
遅延割込み要求発生 / 解除レジスタ (DIRR)	549
通信プリスケール制御レジスタ 0,1 (CDCR0,CDCR1)	494
低消費電力モード制御レジスタ (LPMCR)	97
入力制御レジスタ, 下位 (IPCLR)	396
入力制御レジスタ, 上位 (IPCUR)	394
ノイズキャンセル制御レジスタ (NCCR)	400
波形ジェネレータのレジスタ	293
波形制御レジスタ (SIGCR)	328
バンクレジスタ (PCB,DTB,USB,SSB,ADB)	54

フラッシュメモリ制御ステータスレジスタ (FMCS) .....	608	割り込み処理	
プログラムアドレス検出制御ステータスレジスタ (PACSR) .....	589	割り込み処理	130
プログラムアドレス検出レジスタ (PADR0H,M,L,PADR1H,M,L) .....	591	割り込み処理開始時のスタック動作 .....	149
分周比制御レジスタ (DIV0,DIV1) .....	455	割り込み処理からの復帰時のスタック動作 .....	149
ポート 0 のレジスタ .....	166	割り込み処理のプログラム例 .....	151
ポート 0 のレジスタの機能 .....	167	割り込み制御レジスタ	
ポート 1 のレジスタ .....	172	割り込み制御レジスタ .....	119
ポート 1 のレジスタの機能 .....	173	割り込み制御レジスタ (ICR) .....	122
ポート 2 のレジスタ .....	178	割り込み制御レジスタ (ICR00 ~ ICR15) .....	120
ポート 2 のレジスタの機能 .....	179	割り込み制御レジスタの機能 .....	119, 122
ポート 3 のレジスタ .....	183	割り込み要因、割り込みベクタおよび	
ポート 3 のレジスタの機能 .....	184	割り込み制御レジスタ .....	117
ポート 4 のレジスタ .....	188	割り込み番号	
ポート 4 のレジスタの機能 .....	189	割り込み番号 .....	547
ポート 5 のレジスタ .....	193	割り込みベクタ	
ポート 5 のレジスタの機能 .....	194	割り込みベクタ .....	116
ポート 6 のレジスタ .....	199	割り込み要因、割り込みベクタおよび	
ポート 6 のレジスタの機能 .....	200	割り込み制御レジスタ .....	117
モードデータレジスタ .....	158	割り込みマスク	
割り込み制御レジスタ (ICR) .....	122	割り込みマスク機能 .....	338
割り込み制御レジスタ (ICR00 ~ ICR15) .....	120	割り込み要因	
割り込みレベルマスクレジスタ (ILM) .....	51	割り込み要因、割り込みベクタおよび	
レジスタバンク		割り込み制御レジスタ .....	117
レジスタバンク .....	56	割り込み要求	
レジスタバンクポインタ		割り込み要求の生成 .....	465, 469
汎用レジスタ領域とレジスタバンクポインタ (RP) .....	50	割り込み抑止命令	
レジスタバンクポインタ (RP) .....	50	プリフィックスコードと割り込み抑止命令 .....	62
連続測定モード		割り込みレベルマスクレジスタ	
単一測定モードと連続測定モード .....	467	割り込みレベルマスクレジスタ (ILM) .....	51
連続変換モード		ワンショット位置検出	
El <sup>2</sup> OS を使用した場合の連続変換モードにおいて		ワンショット位置検出と16ビットリロードタイマ	
A/D 変換を行うためのプログラム例 .....	579	アンダフロー方式 .....	429
連続変換モード時の動作 .....	571	ワンショット位置検出方式 .....	428
		ワンショット位置検出または16ビットリロード	
		タイマアンダフロー方式 .....	430
		ワンショット動作モード	
		ワンショット動作モード .....	464
		ワンショットモード	
		内部クロックモード (ワンショットモード)	
		ワンショットモード .....	247
		ワンショットモード	
		(PCNTH レジスタの MDSE:bit13=1) .....	274

## わ

## 割り込み

複数の割り込み .....	132
割り込み動作 .....	115
割り込みの種類と機能 .....	114

CM44-10120-4

---

**富士通マイクロエレクトロニクス・CONTROLLER MANUAL**

F<sup>2</sup>MC-16LX

16 ビット・マイクロコントローラ

MB90460/465 シリーズ

ハードウェアマニュアル

---

2008 年 8 月 第 4 版発行

発行	<b>富士通マイクロエレクトロニクス株式会社</b>	
編集	マーケティング統括部	ビジネス推進部

---

