



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-RAMTM、SRAM、TraveoTM マイクロコントローラー、業界唯一の PSoC[®] 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[®] 静電容量タッチセンシング コントローラー、Wireless BLE Bluetooth[®] 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。

F²MC-16LX

16 ビット・マイクロコントローラ

MB90480/485 Series

ハードウェアマニュアル

F²MC-16LX

16 ビット・マイクロコントローラ

MB90480/485 Series

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://edevicе.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顧を賜り厚くお礼申し上げます。

MB90480/485 シリーズは、民生機器などの高速リアルタイム処理が要求される用途向けに設計された 16 ビットマイクロコントローラです。PHS, 携帯, CD-ROM, VTR などの制御に適した機能を搭載しています。

本書は、実際に MB90480/485 シリーズを使用して製品を開発される技術者を対象に、MB90480/485 シリーズの機能や動作について解説したものです。本書をご一読ください。

なお、各種命令の詳細については、『インストラクション・マニュアル』をご参照ください。

■ 商標

F²MC 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。

Embedded Algorithm は、Advanced Micro Devices, Inc. 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。

■ ライセンス

本製品には、お客様が Philips 社の定めた I²C 標準仕様書に従う I²C システムの中で使用されることを条件に、Philips 社 I²C 特許がライセンスされております。

Purchase of Fujitsu I²C components conveys a license under the Philips I²C Patent Rights to use, these components in an I²C system provided that the system conforms to the I²C Standard Specification as defined by Philips.

■ 本書の全体構成

本書は、以下に示す 27 の章および付録から構成されています。

第 1 章 MB90480/485 シリーズの概要

この章では、MB90480/485 シリーズの特長、ブロックダイアグラム、機能概要など全体を知るための基本的な仕様について説明します。

第 2 章 CPU

この章では、MB90480/485 シリーズの CPU コアの機能を知るために、アーキテクチャ、仕様、命令などの基本的なことについて説明します。

第 3 章 割込み

この章では、割込みの概要、割込みベクタと割込み要因、レジスタの構成 / 機能および割込み処理の動作などについて説明します。

第4章 リセット

この章では、リセットの概要、リセット要因と発振安定待ち時間およびリセットの動作について説明します。

第5章 クロック

この章では、クロックの概要、レジスタの構成 / 機能、クロックモードおよび発振安定待ち時間について説明します。

第6章 低消費電力モード

この章では、低消費電力モードの概要、レジスタの構成 / 機能および低消費電力モードの動作について説明します。

第7章 モード設定

この章では、モード設定の概要、モード端子、モードデータおよびモード設定の各モードにおける動作について説明します。

第8章 I/O ポート

この章では、I/O ポートの概要および I/O ポートで使用するレジスタの構成 / 機能について説明します。

第9章 タイムベースタイマ

この章では、タイムベースタイマの概要、レジスタの構成 / 機能、タイムベースタイマの割込みおよびタイムベースタイマの動作について説明します。

第10章 ウォッチドッグタイマ

この章では、ウォッチドッグタイマの概要、レジスタの構成 / 機能およびウォッチドッグタイマの動作について説明します。

第11章 時計タイマ

この章では、時計タイマの概要、レジスタの構成 / 機能および時計タイマの動作について説明します。

第12章 16 ビット入出力タイマ

この章では、16 ビット入出力タイマの概要、レジスタの構成 / 機能および 16 ビット入出力タイマの動作について説明します。

第13章 8/16 ビットアップダウンカウンタ / タイマ

この章では、8/16 ビットアップダウンカウンタ / タイマの概要、レジスタの構成 / 機能および 8/16 ビットアップダウンカウンタ / タイマの動作について説明します。

第14章 16 ビットリロードタイマ

この章では、16 ビットリロードタイマの概要、レジスタの構成 / 機能および 16 ビットリロードタイマの動作について説明します。

第15章 8/16 ビット PPG タイマ

この章では、8/16 ビット PPG タイマの概要、レジスタの構成 / 機能および 8/16 ビット PPG タイマの動作について説明します。

第16章 DTP/ 外部割込み

この章では、DTP/ 外部割込みの概要、レジスタの構成 / 機能および DTP/ 外部割込みの動作について説明します。

第 17 章 8/10 ビット A/D コンバータ

この章では、8/10 ビット A/D コンバータの概要、レジスタの構成 / 機能および 8/10 ビット A/D コンバータの動作について説明します。

第 18 章 拡張 I/O シリアルインタフェース

この章では、拡張 I/O シリアルインタフェースの概要、レジスタの構成 / 機能および 拡張 I/O シリアルインタフェースの動作について説明します。

第 19 章 UART

この章では、UART の概要、レジスタの構成 / 機能、UART の動作、UART の使用上の注意および UART プログラム例について説明します。

第 20 章 チップセレクト機能

この章では、チップセレクト機能の概要、レジスタの構成 / 機能およびチップセレクト機能の動作について説明します。

第 21 章 アドレス一致検出機能

この章では、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

この章では、ROM ミラー機能選択モジュールの機能およびレジスタの構成 / 機能について説明します。

第 23 章 2M/3M ビットフラッシュメモリ

この章では、2M/3M ビットフラッシュメモリの機能、動作およびフラッシュメモリへのデータの書込み / 消去の方法について説明します。

第 24 章 MB90F481B/F482B/F488B/F489B シリアル書込み接続例

この章では、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。

第 25 章 PWC タイマ (MB90485 シリーズのみ)

この章では、PWC の概要、レジスタ構成 / 機能および PWC の動作について説明します。

第 26 章 μ PG タイマ (MB90485 シリーズのみ)

この章では、 μ PG の概要、レジスタ構成 / 機能および μ PG の動作について説明します。

第 27 章 I²C インタフェース (MB90485 シリーズのみ)

この章では、I²C インタフェースの概要、レジスタ構成 / 機能および I²C インタフェースの動作について説明します。

付 録

付録として、I/O マップ、割込みベクタおよび命令一覧について、本文に記載できなかった細かいところや、プログラミングするときに参照することがらを記載しています。

- 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に、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

第 1 章	MB90480/485 シリーズの概要	1
1.1	MB90480/485 シリーズの特長	2
1.2	MB90480/485 シリーズのブロックダイアグラム	5
1.3	外形寸法図	7
1.4	端子配列	9
1.5	端子機能	11
1.6	入出力回路形式	16
1.7	デバイスの取扱い	19
第 2 章	CPU の機能	21
2.1	CPU の概略仕様	22
2.2	メモリ空間	23
2.3	CPU のレジスタ	27
2.3.1	アキュムレータ (A)	29
2.3.2	ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP)	30
2.3.3	プロセッサステータス (PS)	31
2.3.4	プログラムカウンタ (PC)	34
2.3.5	プログラムカウンタバンクレジスタ (PCB)	35
2.3.6	ダイレクトページレジスタ (DPR)	36
2.3.7	汎用レジスタ (レジスタバンク)	37
2.4	プリフィックスコード	38
第 3 章	割込み	41
3.1	割込みの概要	42
3.2	割込み要因と割込みベクタ	44
3.3	割込み制御レジスタと周辺機能	47
3.3.1	割込み制御レジスタ (ICR00 ~ ICR15)	48
3.4	ハードウェア割込み	51
3.4.1	ハードウェア割込みの動作	54
3.4.2	ハードウェア割込みの動作フロー	56
3.4.3	ハードウェア割込みの使用手順	57
3.4.4	多重割込み	58
3.4.5	ハードウェア割込みの処理時間	60
3.5	ソフトウェア割込み	62
3.6	μDMAC による割込み	64
3.6.1	DMA ディスクリプタ	67
3.6.2	DMA ディスクリプタの各レジスタ	68
3.6.3	μDMAC の処理手順	71
3.6.4	μDMAC の処理時間	72
3.7	拡張インテリジェント I/O サービス (EI ² OS) による割込み	74
3.7.1	拡張インテリジェント I/O サービス (EI ² OS) ディスクリプタ (ISD)	76
3.7.2	拡張インテリジェント I/O サービス (EI ² OS) ディスクリプタ (ISD) の説明	77
3.7.3	拡張インテリジェント I/O サービス (EI ² OS) の動作	80
3.7.4	拡張インテリジェント I/O サービス (EI ² OS) の設定手順	81
3.7.5	拡張インテリジェント I/O サービス (EI ² OS) 処理時間	82

3.8	例外処理割込み	84
3.9	割込み処理のスタック動作.....	85
3.10	割込み処理のプログラム例.....	87
3.11	遅延割込み発生モジュール.....	89
3.11.1	遅延割込み発生モジュールの動作.....	90
第4章	リセット	91
4.1	リセットの概要	92
4.2	リセット要因と発振安定待ち時間.....	94
4.3	外部リセット端子	96
4.4	リセット動作	97
4.5	リセット要因ビット	99
4.6	リセットによる各端子の状態	101
第5章	クロック	103
5.1	クロックの概要	104
5.2	クロック発生部のブロックダイアグラム.....	106
5.3	クロック選択レジスタ (CKSCR), PLL 出力選択レジスタ (PLLOS).....	108
5.4	クロックモード	113
5.5	発振安定待ち時間.....	117
5.6	振動子と外部クロックの接続.....	118
第6章	低消費電力モード	119
6.1	低消費電力モードの概要	120
6.2	低消費電力制御回路のブロックダイアグラム.....	122
6.3	低消費電力モード制御レジスタ (LPMCR)	124
6.4	CPU 間欠動作モード	127
6.5	スタンバイモード.....	128
6.5.1	スリープモード	129
6.5.2	タイムベースタイマモード	131
6.5.3	時計モード	133
6.5.4	ストップモード	135
6.6	スタンバイモードの状態遷移図	137
6.7	スタンバイモード, ホールド, リセット時の端子状態	139
6.8	低消費電力モード使用上の注意	144
第7章	モード設定	147
7.1	モード設定.....	148
7.2	モード端子 (MD2 ~ MD0).....	149
7.3	モードデータ	150
7.4	外部メモリアクセス	153
7.4.1	自動レディ機能選択レジスタ (ARSR)	155
7.4.2	外部アドレス出力制御レジスタ (HACR).....	156
7.4.3	バス制御信号選択レジスタ (EPCR)	157
7.5	モード設定の各モードにおける動作	159
7.5.1	外部メモリアクセス制御信号.....	160
7.5.2	レディ機能	163
7.5.3	ホールド機能.....	166

第 8 章	I/O ポート	169
8.1	I/O ポートの機能	170
8.2	I/O ポートのレジスタ	171
8.2.1	ポートデータレジスタ (PDR0 ~ PDRA)	172
8.2.2	ポート方向レジスタ (DDR0 ~ DDRA)	173
8.2.3	その他のレジスタ	175
第 9 章	タイムベースタイマ	177
9.1	タイムベースタイマの概要	178
9.2	タイムベースタイマの構成	179
9.3	タイムベースタイマ制御レジスタ (TBTC)	181
9.4	タイムベースタイマの割込み	183
9.5	タイムベースタイマの動作	184
9.6	タイムベースタイマ使用上の注意	187
9.7	タイムベースタイマのプログラム例	188
第 10 章	ウォッチドッグタイマ	189
10.1	ウォッチドッグタイマの概要	190
10.2	ウォッチドッグタイマ制御レジスタ (WDTC)	191
10.3	ウォッチドッグタイマの構成	193
10.4	ウォッチドッグタイマの動作	195
10.5	ウォッチドッグタイマ使用上の注意	197
10.6	ウォッチドッグタイマのプログラム例	198
第 11 章	時計タイマ	199
11.1	時計タイマの概要	200
11.2	時計タイマの構成	201
11.3	時計タイマ制御レジスタ (WTC)	202
11.4	時計タイマの動作	204
第 12 章	16 ビット入出力タイマ	207
12.1	16 ビット入出力タイマの概要	208
12.2	16 ビット入出力タイマの構成	209
12.3	16 ビット入出力タイマのレジスタの構成と機能	213
12.3.1	フリーランタイマ	214
12.3.2	アウトプットコンペア	219
12.3.3	インプットキャプチャ	222
12.4	16 ビット入出力タイマの割込み	224
12.5	16 ビット入出力タイマの動作	227
12.5.1	フリーランタイマの動作	228
12.5.2	アウトプットコンペアの動作	229
12.5.3	インプットキャプチャの動作	231
12.5.4	フリーランタイマのタイミング	232
12.5.5	アウトプットコンペアのタイミング	233
12.5.6	インプットキャプチャのタイミング	234
12.6	16 ビット入出力タイマのプログラム例	235

第 13 章	8/16 ビットアップダウンカウンタ / タイマ	247
13.1	8/16 ビットアップダウンカウンタ / タイマの概要	248
13.2	8/16 ビットアップダウンカウンタ / タイマの構成	250
13.3	8/16 ビットアップダウンカウンタ / タイマのレジスタの構成と機能	253
13.3.1	カウンタコントロールレジスタ (ch.0) 上位 (CCRHO)	254
13.3.2	カウンタコントロールレジスタ (ch.1) 上位 (CCRHI)	256
13.3.3	カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)	258
13.3.4	カウンタステータスレジスタ 0/1 (CSR0/1)	260
13.3.5	アップダウンカウンタレジスタ (ch.0/ch.1) (UDCR0/1)	262
13.3.6	リロード / コンペアレジスタ (ch.0/ch.1) (RCR0/1)	263
13.4	8/16 ビットアップダウンカウンタ / タイマの割込み	264
13.5	8/16 ビットアップダウンカウンタ / タイマの動作	266
13.5.1	リロード / コンペア機能	269
13.5.2	アップダウンカウンタレジスタ (UDCR) へのデータの書込み	272
13.6	8/16 ビットアップダウンカウンタ / タイマのプログラム例	274
第 14 章	16 ビットリロードタイマ	281
14.1	16 ビットリロードタイマの概要	282
14.2	16 ビットリロードタイマのレジスタの構成と機能	286
14.2.1	タイマ制御ステータスレジスタ (TMCSR)	287
14.2.2	16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR)	291
14.3	16 ビットリロードタイマの割込み	293
14.4	16 ビットリロードタイマの動作	294
14.4.1	カウント動作の状態遷移	295
14.4.2	内部クロックモード (リロードモード) の動作	296
14.4.3	内部クロックモード (ワンショットモード) の動作	298
14.4.4	イベントカウントモード	300
14.5	16 ビットリロードタイマのプログラム例	302
第 15 章	8/16 ビット PPG タイマ	307
15.1	8/16 ビット PPG タイマの概要	308
15.2	8/16 ビット PPG タイマの構成	309
15.3	8/16 ビット PPG タイマのレジスタの構成と機能	312
15.3.1	PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)	313
15.3.2	PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)	315
15.3.3	PPG0 ~ 5 出力制御レジスタ (PPG01/PPG23/PPG45)	317
15.3.4	リロードレジスタ (PRL0 ~ 5, PRLH0 ~ 5)	319
15.4	8/16 ビット PPG タイマの割込み	320
15.5	8/16 ビット PPG タイマの動作	322
15.6	8/16 ビット PPG タイマのプログラム例	328
第 16 章	DTP/ 外部割込み	331
16.1	DTP/ 外部割込みの概要	332
16.2	DTP/ 外部割込みのレジスタの構成と機能	334
16.3	DTP/ 外部割込み	336
16.4	DTP/ 外部割込みの動作	338
16.5	DTP/ 外部割込み使用上の注意	340
16.6	DTP/ 外部割込みのプログラム例	341

第 17 章	8/10 ビット A/D コンバータ	345
17.1	8/10 ビット A/D コンバータの概要	346
17.2	8/10 ビット A/D コンバータの構成	347
17.3	8/10 ビット A/D コンバータのレジスタの構成と機能	349
17.3.1	コントロールステータスレジスタ 1(ADCS1)	350
17.3.2	コントロールステータスレジスタ 2(ADCS2)	352
17.3.3	データレジスタ (ADCR2, ADCR1)	355
17.4	8/10 ビット A/D コンバータの割込み	356
17.5	8/10 ビット A/D コンバータの動作	357
17.5.1	単発モードにおける μ DMAC の起動例	359
17.5.2	連続モードにおける μ DMAC の起動例	360
17.5.3	停止モードにおける μ DMAC の起動例	361
17.6	8/10 ビット A/D コンバータの変換データ保護機能	362
17.7	8/10 ビット A/D コンバータの使用上の注意	364
17.8	8/10 ビット A/D コンバータのプログラム例	365
第 18 章	拡張 I/O シリアルインタフェース	371
18.1	拡張 I/O シリアルインタフェースの概要	372
18.2	拡張 I/O シリアルインタフェースの構成	373
18.3	拡張 I/O シリアルインタフェースのレジスタの構成と機能	375
18.3.1	シリアルモードコントロールステータスレジスタ 0/1 (SMCS0/SMCS1)	376
18.3.2	シリアルデータレジスタ 0/1(SDR0/SDR1)	379
18.3.3	通信プリスケアラコントロールレジスタ 0/1(SDCR0/SDCR1)	380
18.4	拡張 I/O シリアルインタフェースの割込み	381
18.5	拡張 I/O シリアルインタフェースの動作	382
18.5.1	シフトクロックモード	383
18.5.2	シリアル I/O の動作状態	384
18.5.3	シフト動作のスタート / ストップタイミングと入出力のタイミング	386
18.5.4	割込み機能	388
18.6	拡張 I/O シリアルインタフェースのプログラム例	389
第 19 章	UART	393
19.1	UART の概要	394
19.2	UART の構成	395
19.3	UART のレジスタの構成と機能	397
19.3.1	シリアルモードレジスタ (SMR)	398
19.3.2	シリアルコントロールレジスタ (SCR)	400
19.3.3	シリアルインプット / アウトプットレジスタ (SIDR/SODR)	402
19.3.4	シリアルステータスレジスタ (SSR)	403
19.3.5	通信プリスケアラコントロールレジスタ (CDCR)	405
19.4	UART の割込み	407
19.5	UART の動作	409
19.5.1	非同期モード (動作モード 0, 1) 時の動作	412
19.5.2	同期モード (動作モード 2) 時の動作	415
19.5.3	双方向通信機能 (ノーマルモード)	417
19.5.4	マスタ / スレーブ型通信機能 (マルチプロセッサモード)	419
19.6	UART の使用上の注意	421
19.7	UART のプログラム例	422

第 20 章	チップセレクト機能	429
20.1	チップセレクト機能の概要	430
20.2	チップセレクト機能の構成	431
20.3	チップセレクト機能のレジスタの構成と機能	433
20.3.1	チップセレクト領域 MASK レジスタ (CMRx)	434
20.3.2	チップセレクト領域レジスタ (CARx)	435
20.3.3	チップセレクトコントロールレジスタ (CSCR)	436
20.3.4	チップセレクトアクティブレベルレジスタ (CALR)	437
20.4	チップセレクト機能の動作	438
第 21 章	アドレス一致検出機能	439
21.1	アドレス一致検出機能の概要	440
21.2	アドレス一致検出機能のブロックダイアグラム	441
21.3	アドレス一致検出機能のレジスタ構成	442
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)	443
21.3.2	プログラムアドレス検出レジスタ (PADR0, PADR1)	445
21.4	アドレス一致検出機能の動作説明	447
21.4.1	アドレス一致検出機能の使用例	448
21.5	アドレス一致検出機能のプログラム例	452
第 22 章	ROM ミラー機能選択モジュール	453
22.1	ROM ミラー機能選択モジュールの概要	454
22.2	ROM ミラー機能選択レジスタ (ROMM)	455
第 23 章	2M/3M ビットフラッシュメモリ	457
23.1	2M/3M ビットフラッシュメモリの概要	458
23.2	2M/3M ビットフラッシュメモリのセクタ構成	459
23.3	フラッシュメモリコントロールステータスレジスタ (FMCS)	460
23.4	フラッシュメモリの自動アルゴリズム起動方法	466
23.5	自動アルゴリズム実行状態の確認	467
23.5.1	データポーリングフラグ (DQ7)	468
23.5.2	トグルビットフラグ (DQ6)	469
23.5.3	タイミングリミット超過フラグ (DQ5)	470
23.5.4	セクタ消去タイマフラグ (DQ3)	471
23.6	フラッシュメモリの書込み / 消去	472
23.6.1	フラッシュメモリを読出し / リセット状態にする方法	473
23.6.2	フラッシュメモリにデータを書き込む方法	474
23.6.3	フラッシュメモリの全データを消去する方法 (チップ消去)	476
23.6.4	フラッシュメモリの任意のデータを消去する方法 (セクタ消去)	477
23.6.5	フラッシュメモリのセクタ消去を一時停止する方法	479
23.6.6	フラッシュメモリのセクタ消去を再開する方法	480
23.7	フラッシュセキュリティ機能	481
第 24 章	MB90F481B/F482B/F488B/F489B シリアル書込み接続例	483
24.1	MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成	484
24.2	シングルチップモード時の接続例 (ユーザ電源使用時)	488
24.3	フラッシュマイコンプログラマとの接続例 (ユーザ電源使用時)	490

第 25 章 PWC タイマ (MB90485 シリーズのみ)	493
25.1 PWC タイマの概要	494
25.2 PWC タイマの構成	495
25.3 PWC タイマのレジスタの構成と機能	497
25.3.1 PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR02)	498
25.3.2 PWC データバッファレジスタ (PWCR0 ~ PWCR2)	503
25.3.3 分周比制御レジスタ (DIVR0 ~ DIVR2)	504
25.4 PWC タイマの割込み	505
25.5 PWC タイマの動作	507
25.5.1 タイマ機能の動作	508
25.5.2 パルス幅測定機能の動作	509
25.5.3 カウントクロックおよび動作モードの選択	510
25.5.4 タイマ / パルス幅測定の起動と停止	512
25.5.5 タイマモードの動作	514
25.5.6 パルス幅測定モードの動作	517
25.6 PWC タイマの使用上の注意	523
第 26 章 μPG タイマ (MB90485 シリーズのみ)	525
26.1 μ PG タイマの概要と構成	526
26.2 μ PG タイマのレジスタの構成と機能	528
26.3 μ PG タイマのタイミングチャート	529
第 27 章 I²C インタフェース (MB90485 シリーズのみ)	531
27.1 I ² C インタフェースの概要	532
27.2 I ² C インタフェースの構成	533
27.3 I ² C インタフェースのレジスタの構成と機能	535
27.3.1 バスステータスレジスタ (IBSR)	536
27.3.2 バスコントロールレジスタ (IBCR)	538
27.3.3 クロックコントロールレジスタ (ICCR)	544
27.3.4 アドレスレジスタ (IADR)	546
27.3.5 データレジスタ (IDAR)	547
27.4 I ² C インタフェースの割込み	548
27.5 I ² C インタフェースの動作	550
付録	553
付録 A メモリマップ	554
付録 B I/O マップ	558
付録 C 割込み要因と割込みベクタ , 割込み制御レジスタ	564
付録 D 命令	566
用語索引	627
レジスタ索引	649
端子機能索引	653
割込みベクタ索引	657

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。)
566 ~ 626	「付録 D 命令」全体を変更

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。

< 参考 > 5 版 6 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。)
	型格を変更 (MB90483B MB90483C) シリーズ名を変更 (MB90480 シリーズ MB90480/485 シリーズ)
i	■ ライセンスを追加
11	表 1.5-1 端子機能 (1 / 5) の端子名 D00 ~ D07 および D08 ~ D15 の機能を変更 (出力端子 入出力端子)
90	図 3.11-2 遅延割込み発生モジュールの動作を変更 (ICR _{XX} IL) (ICR _{XX} ILM) (NTA INTA)
151	図 7.3-1 アクセス領域と物理アドレスの関係を変更 (アクセスなし アクセス禁止)
289	[bit2] UF(タイマ割込み要求フラグ) を変更 (μDMA μDMAC)
340	■ 外部に接続する周辺装置の条件を変更 ((暫定値) の記述を削除)
376	■ シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1) の図を変更 (* 1 および * 2 の記述を削除)
509	図 25.5-2 パルス幅測定動作 (単発測定モード / "H" 幅測定) を変更 図 25.5-3 パルス幅測定動作 (連続測定モード / "H" 幅測定) を変更
550	< 注意事項 > を追加
554	図 A-1 メモリマップを変更 (アクセス無 アクセス禁止)
555	表 A-1 各品種のアドレス #1, アドレス #2 およびアドレス #3 の対応を変更
556	図 A-2 MB90F489B メモリマップを変更 (アクセス無 アクセス禁止)
557	図 A-3 MB90483C メモリマップを追加

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

MB90480/485 シリーズの 概要

MB90480/485 シリーズの特長や基本的な仕様について説明します。

- 1.1 MB90480/485 シリーズの特長
- 1.2 MB90480/485 シリーズのブロックダイアグラム
- 1.3 外形寸法図
- 1.4 端子配列
- 1.5 端子機能
- 1.6 入出力回路形式
- 1.7 デバイスの取扱い

1.1 MB90480/485 シリーズの特長

MB90480/485 シリーズは、高速リアルタイム処理が要求される用途向けに設計された 16 ビットマイクロコントローラです。

■ MB90480/485 シリーズの特長

MB90480/485 シリーズには、以下の特長があります。

- 最小命令実行時間
 - 40.0ns/6.25MHz 原発振 4 通倍 (内部動作 25MHz/3.3V \pm 0.3V 時)
 - 62.5ns/4MHz 原発振 4 通倍 (内部動作 16MHz/3.0V \pm 0.3V 時)
 - PLL クロック通倍方式
- 最大メモリ空間 :16M バイト
- コントロール用途に最適化された命令体系
 - 取扱い可能なデータタイプ : ビット / バイト / ワード / ロングワード
 - 標準アドレッシングモード :23 種類
 - 32 ビットアキュムレータの採用による高精度演算の強化
 - 符号付き乗除算 , 拡張 RETI 命令
- 高級言語 (C 言語) マルチタスクに対応をとった命令体系
 - システムスタックポイントの採用
 - 命令セットの対称性とバレルシフト命令
- ノンマルチプレクスバス / マルチプレクスバス対応
- 実行速度の向上 :4 バイトのキュー
- 強力な割込み機能 (プライオリティがプログラマブルに 8 レベル設定可能) : 外部割込み 8 本
- データ転送機能 (μ DMAC) : 最大 16 チャンネル
- 内蔵 ROM: フラッシュ版 :192K バイト /256K バイト /384K バイト
マスク版 :192K バイト /256K バイト
- 内蔵 RAM: フラッシュ版 :4K バイト /6K バイト /10K バイト /24K バイト
マスク版 :10K バイト /16K バイト
- 汎用ポート : 最大 84 本
(入力プルアップ抵抗設定可能:16 本
出力オープンドレイン設定可能:16 本)
- A/D コンバータ (RC 逐次比較型) :8 チャンネル
(分解能 10 ビット / 変換時間 3.68 μ s(25MHz 動作時))
- UART:1 チャンネル
- 拡張 I/O シリアルインタフェース (SIO) :2 チャンネル

- 8/16ビットPPG:3チャンネル(8ビット×6チャンネル, 16ビット×3チャンネルモード切換え機能付き)
- 8/16ビットアップダウンタイマ:1チャンネル(8ビット×2チャンネル, 16ビット×1チャンネルモード切換え機能付き)
- 16ビットリロードタイマ:1チャンネル
- 16ビット入出力タイマ(インプットキャプチャ×2チャンネル, アウトプットコンペア×6チャンネル, フリーランタイマ×1チャンネル)
- 2系統クロックジェネレータ内蔵
- 低消費電力モード(ストップモード/スリープモード/CPU間欠動作モード/時計モード/タイムベースタイマモード)
- パッケージ:QFP100/LQFP100
- CMOS テクノロジ
- 3V 単一電源 (MB90485 シリーズのみ一部ポートで 5V 電源での駆動が可能)
- I²C インタフェース*:1チャンネル(MB90485 シリーズのみ)
MB90487B のみ I²C インタフェース搭載で, 該当する端子 (P77/P76) も N-ch オープンドレイン端子 (P-ch なし) になります。
- 16 ビット PWC: 3 チャンネル (MB90485 シリーズのみ)
3 チャンネルのうち, 2 チャンネルを入力比較機能付き
- μ PG: 1 チャンネル (MB90485 シリーズのみ)

*: I²C のライセンス

本製品には, お客様が Philips 社の定めた I²C 標準仕様書に従う I²C システムの中で使用されることを条件に, Philips 社 I²C 特許がライセンスされております。

Purchase of Fujitsu I²C components conveys a license under the Philips I²C Patent Rights to use, these components in an I²C system provided that the system conforms to the I²C Standard Specification as defined by Philips.

■ 品種構成

MB90480 シリーズの品種構成を表 1.1-1 に, MB90485 シリーズの品種構成を表 1.1-2 に示します。

表 1.1-1 MB90480 シリーズ品種構成一覧

	MB90V480B	MB90F481B	MB90F482B
品種構成	評価用品	フラッシュ メモリ品	
ROM 容量	-	192K バイト	256K バイト
RAM 容量	16K バイト	4K バイト	6K バイト
備考	ユーザ端子 3/5V 版 *1	f=25MHz ユーザ端子 3V 版 セキュリティ機能なし	f=25MHz ユーザ端子 3V 版 セキュリティ機能なし
エミュレータ専用電源 *2	あり	-	-

*1: ユーザ端子 :P20 ~ P27, P30 ~ P37, P40 ~ P47, P70 ~ P77

*2: エミュレータ (MB2147-01) をご使用いただく際のジャンパススイッチ (TOOL VCC) の設定です。
詳細につきましては, MB2147-01 または MB2147-20 ハードウェアマニュアル (3.3 エミュレータ専用電源切換え) を参照してください。

表 1.1-2 MB90485 シリーズ品種構成一覧

	MB90V485B	MB90487B	MB90F488B	MB90488B	MB90F489B	MB90483C
品種構成	評価用品	マスク ROM 品	フラッシュメモリ品	マスク ROM 品	フラッシュメモリ品	マスク ROM 品
ROM 容量	-	192K バイト	256K バイト	256K バイト	384K バイト	256K バイト
RAM 容量	16K バイト	10K バイト	10K バイト	10K バイト	24K バイト	16K バイト
備考	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2 セキュリティ機能あり	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2 セキュリティ機能あり	f=25MHz 3V/5V 電源 *1 PWC, μ PG, I ² C 搭載 *2
エミュレータ専用電源 *3	あり	-	-	-	-	-

*1: 3V/5V I/F 端子 : P20 ~ P27, P30 ~ P37, P40 ~ P47, P70 ~ P77。この端子以外は 3V 電源。

*2: I²C 搭載時に P76/P77 端子は, N-ch オープンドレイン端子 (P-ch なし) になります。

*3: エミュレータ (MB2147-01) をご使用いただく際のジャンパススイッチ (TOOL VCC) の設定です。

詳細につきましては, MB2147-01 または MB2147-20 ハードウェアマニュアル (3.3 エミュレータ専用電源切換え) を参照してください。

■ 対応品種のパッケージ

● パッケージ

パッケージの相違点については, 下表を参照してください。

表 1.1-3 MB90480/485 シリーズパッケージと品種対応

品種 パッケージ	MB90487B/488B MB90483C	MB90F481B/F482B MB90F488B/F489B	MB90V480B MB90V485B
FPT-100P-M06	○	○	×
FPT-100P-M05	○	○	×
PGA-299C	×	×	○

○ : 使用可能

× : 使用不可能

● メモリ空間

メモリ空間の相違点については, 付録のメモリマップを参照してください。

● 消費電流

消費電流の相違点については, データシートの電気的特性を参照してください。

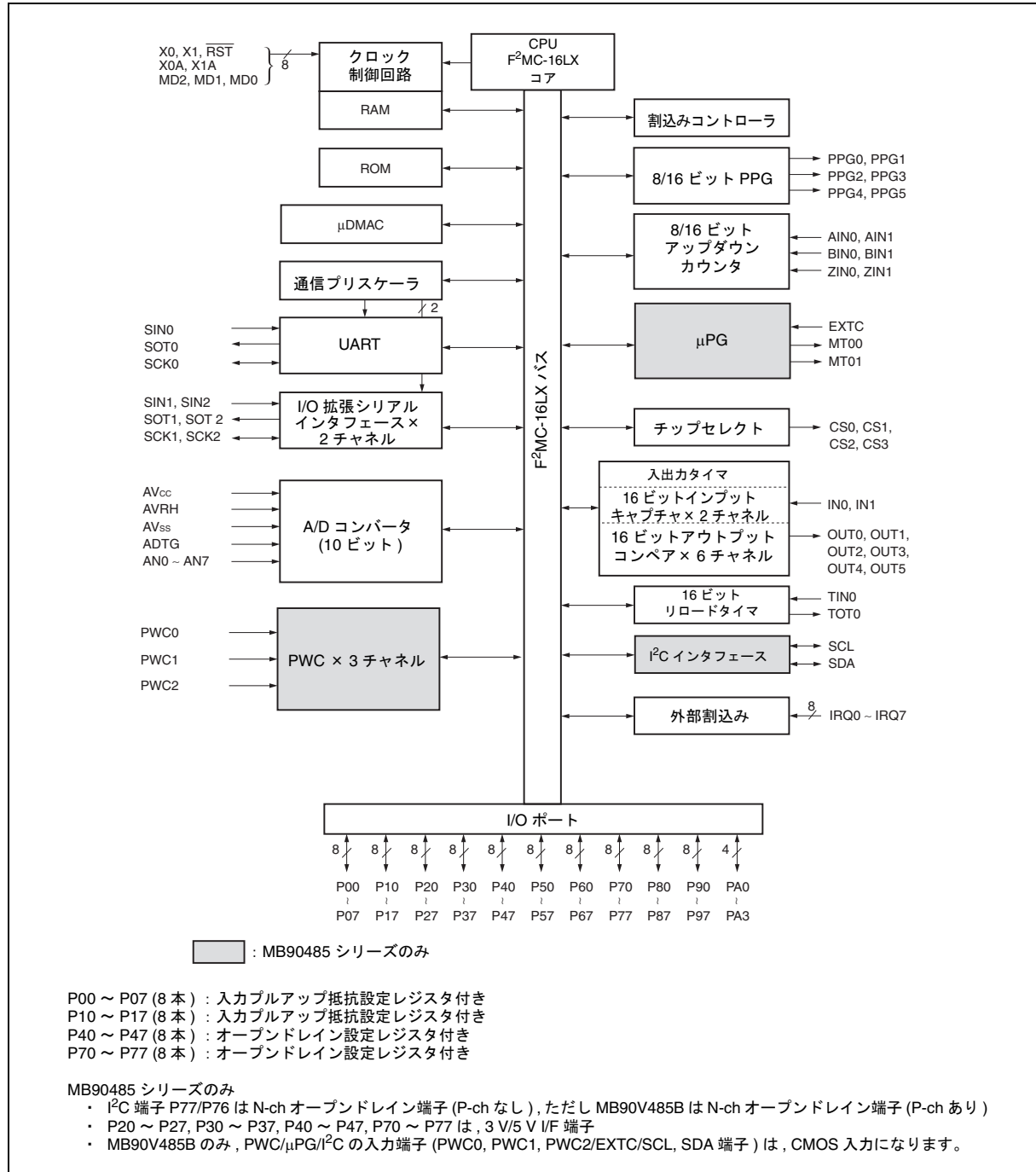
1.2 MB90480/485 シリーズのブロックダイアグラム

MB90480/485 シリーズのブロックダイアグラムを示します。

■ MB90480/485 シリーズのブロックダイアグラム

図 1.2-1 に, MB90480/485 シリーズのブロックダイアグラムを示します。

図 1.2-1 MB90480/485 シリーズのブロックダイアグラム



<注意事項>

図 1.2-1 において , I/O ポートは内蔵の各機能ブロックと端子を共用しています。内蔵モジュールの端子として使用する場合は , I/O ポートとしては使えません。

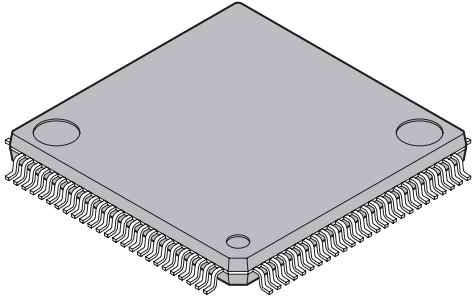
1.3 外形寸法図

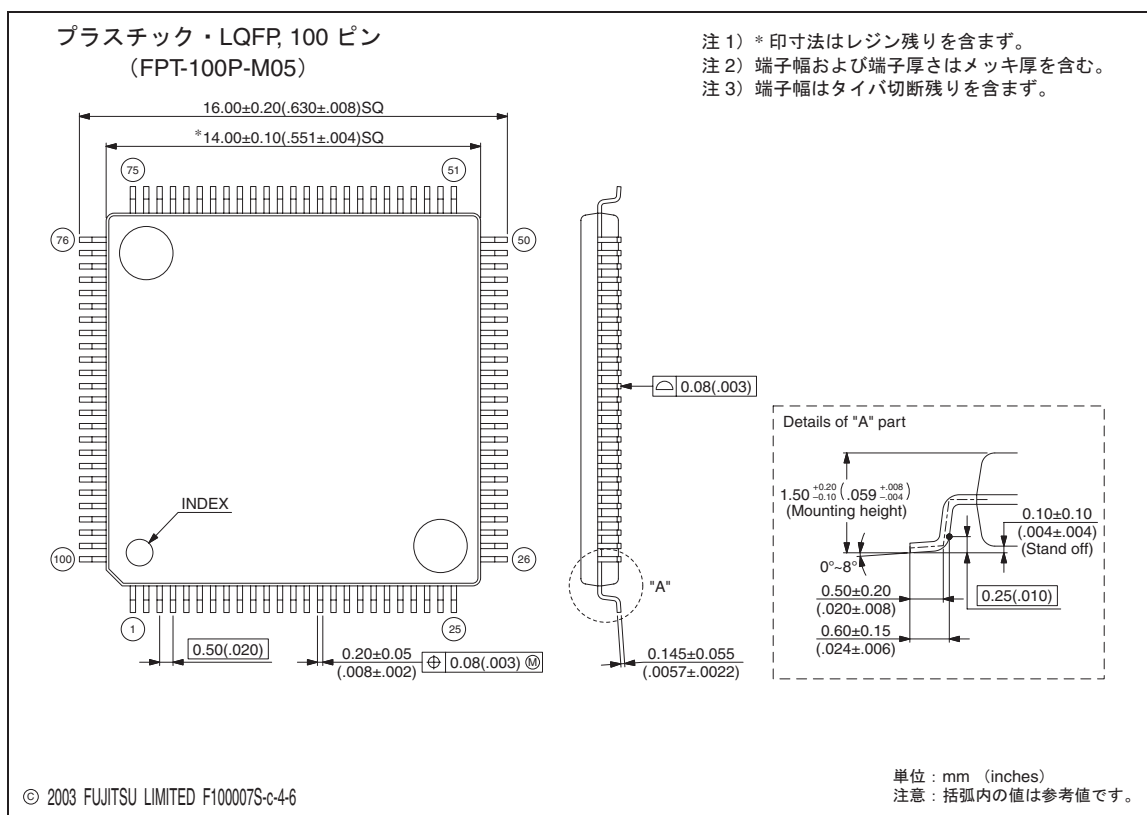
MB90480/485 シリーズには, 2 種類のパッケージが用意されています。

■ 外形寸法図 (LQFP-100)

図 1.3-1 に, LQFP-100 タイプの外形寸法図を示します。

図 1.3-1 LQFP-100 タイプの外形寸法図

<p>プラスチック・LQFP, 100 ピン</p>  <p>(FPT-100P-M05)</p>	リードピッチ	0.50mm
	パッケージ幅× パッケージ長さ	14.0 × 14.0mm
	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
	質量	0.65g
	コード (参考)	P-LFQFP100-14×14-0.50



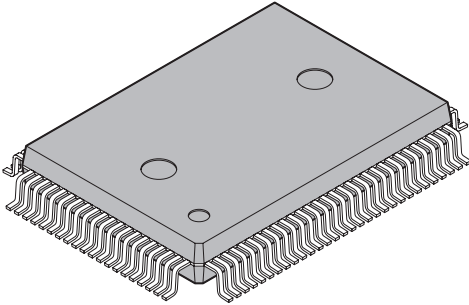
最新の外形寸法図については, 下記の URL にてご確認ください。

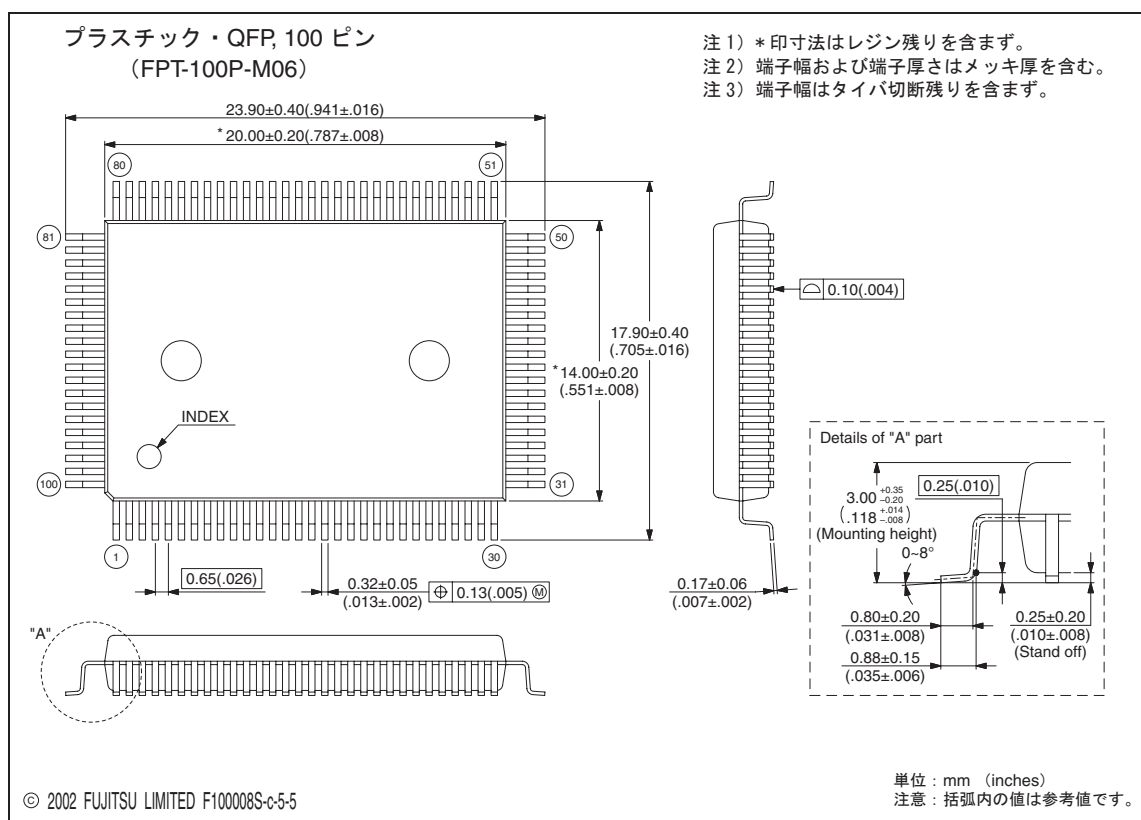
<http://edevic.fujitsu.com/jp/datasheet/jf-ovpklv.html>

■ 外形寸法図 (QFP-100)

図 1.3-2 に , QFP-100 タイプの外形寸法図を示します。

図 1.3-2 QFP-100 タイプの外形寸法図

<p>プラスチック・QFP, 100 ピン</p>  <p>(FPT-100P-M06)</p>	リードピッチ	0.65mm
	パッケージ幅× パッケージ長さ	14.00× 20.00mm
	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-QFP100-14× 20-0.65



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/jp/datasheet/jf-ovpklv.html>

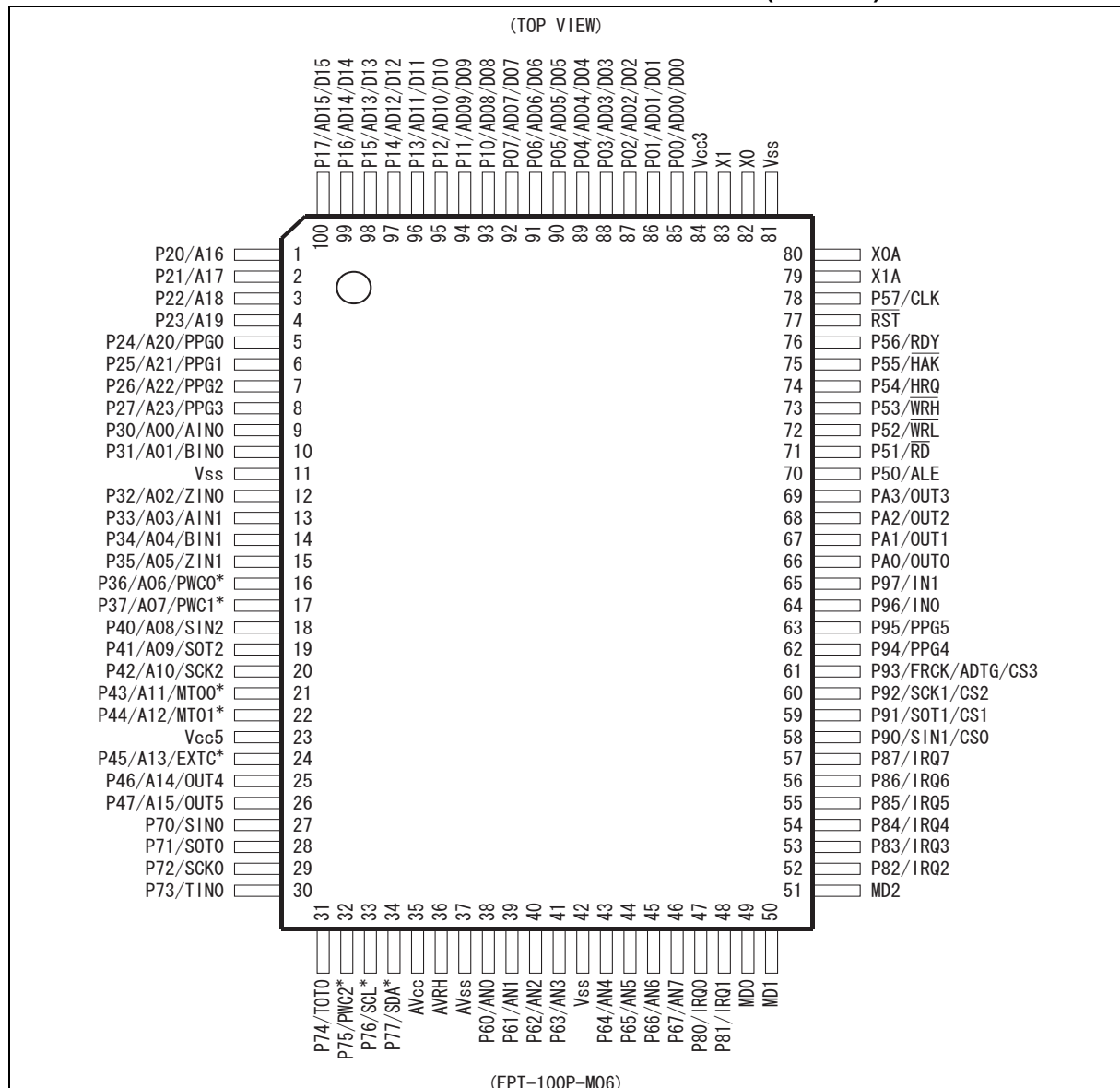
1.4 端子配列

MB90480/485 シリーズの端子配列を 2 種類のパッケージ別に示します。

■ 端子配列図 (QFP-100)

図 1.4-1 に、QFP-100 タイプの端子配列図を示します。

図 1.4-1 MB90480/485 シリーズの端子配列図 (QFP-100)



*: MB90485 シリーズでの端子です。(MB90480 シリーズでは P36/A06, P37/A07, P43/A11, P44/A12, P75, P76, P77 となります。)

(注意事項) MB90485 シリーズのみ

- ・ I²C 端子 P77, P76 は N-ch オープンドレイン端子 (P-ch なし) , ただし MB90V485B は N-ch オープンドレイン端子 (P-ch あり)
- ・ P20 ~ P27, P30 ~ P37, P40 ~ P47, P70 ~ P77 は , 3 V/5 V I/F 端子
- ・ MB90V485B のみ , PWC/μPG/I²C の入力端子 (PWC0, PWC1, PWC2/EXTC/SCL, SDA 端子) は , CMOS 入力になります。

1.5 端子機能

MB90480/485 シリーズの端子機能について説明します。

■ 端子機能

表 1.5-1 に、MB90480/485 シリーズの端子機能を示します。

表 1.5-1 端子機能 (1 / 5)

端子番号		端子名	入出力 回路形式 *3	機 能
QFP *1	LQFP *2			
82	80	X0	A	発振端子です。
83	81	X1	A	発振端子です。
80	78	X0A	A	32kHz 発振端子です。
79	77	X1A	A	32kHz 発振端子です。
77	75	$\overline{\text{RST}}$	B	リセット入力端子です。
85 ～ 92	83 ～ 90	P00 ～ P07	C (CMOS)	汎用の入出力ポートです。 ポート 0 入力抵抗レジスタ (RDR0) により、プルアップ抵抗を付加 (RD00-RD07=1) に設定できます (出力設定時は無効)。
		AD00 ～ AD07		マルチプレクスモード時、外部アドレス/データバス下位入出力端子として機能します。
		D00 ～ D07		ノンマルチプレクスモード時、外部データバス下位入出力端子として機能します。
93 ～ 100	91 ～ 98	P10 ～ P17	C (CMOS)	汎用の入出力ポートです。 ポート 1 入力抵抗レジスタ (RDR1) により、プルアップ抵抗を付加 (RD10-RD17=1) に設定できます (出力設定時は無効)。
		AD08 ～ AD15		マルチプレクスモード時、外部アドレス/データバス上位入出力端子として機能します。
		D08 ～ D15		ノンマルチプレクスモード時、外部データ上位入出力端子として機能します。
1 ～ 4	99 100 1 2	P20 ～ P23	E (CMOS/H)	汎用の入出力ポートです。外バスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "1" の場合、汎用入出力ポートとして機能します。
		A16 ～ A19		マルチプレクスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "0" の場合、アドレス上位出力端子 (A16 ～ A19) として機能します。 ノンマルチプレクスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "0" の場合、アドレス上位出力端子 (A16 ～ A19) として機能します。
5 ～ 8	3 ～ 6	P24 ～ P27	E (CMOS/H)	汎用の入出力ポートです。外バスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "1" の場合、汎用入出力ポートとして機能します。
		A20 ～ A23		マルチプレクスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "0" の場合、アドレス上位出力端子 (A20 ～ A23) として機能します。 ノンマルチプレクスモード時、外部アドレス出力制御レジスタ (HACR) の対応するビットが "0" の場合、アドレス上位出力端子 (A20 ～ A23) として機能します。
		PPG0 ～ PPG3		PPG タイマ出力端子として機能します。
9	7	P30	E (CMOS/H)	汎用の入出力ポートです。
		A00		ノンマルチプレクスモード時、外部アドレス端子として機能します。
		AIN0		8/16 ビットアップダウンタイマ入力端子です (ch.0)。

表 1.5-1 端子機能 (2 / 5)

端子番号		端子名	入出力 回路形式 *3	機 能	
QFP *1	LQFP *2				
10	8	P31	E (CMOS/H)	汎用の入出力ポートです。	
		A01		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		BIN0		8/16 ビットアップダウンタイマ入力端子です (ch.0)。	
12	10	P32	E (CMOS/H)	汎用の入出力ポートです。	
		A02		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		ZIN0		8/16 ビットアップダウンタイマ入力端子です (ch.0)。	
13	11	P33	E (CMOS/H)	汎用の入出力ポートです。	
		A03		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		AIN1		8/16 ビットアップダウンタイマ入力端子です (ch.1)。	
14	12	P34	E (CMOS/H)	汎用の入出力ポートです。	
		A04		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		BIN1		8/16 ビットアップダウンタイマ入力端子です (ch.1)。	
15	13	P35	E (CMOS/H)	汎用の入出力ポートです。	
		A05		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		ZIN1		8/16 ビットアップダウンタイマ入力端子です (ch.1)。	
16 17	14 15	P36, P37	D (CMOS)	MB90480 シリーズ	汎用の入出力ポートです。
		A06, A07			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		P36, P37	E (CMOS/H)	MB90485 シリーズ	汎用の入出力ポートです。
		A06, A07			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		PWC0, PWC1 *4			PWC 入力端子として機能します。
18	16	P40	G (CMOS/H)	汎用入出力ポートです。	
		A08		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		SIN2		単純シリアル I/O 入力端子です。	
19	17	P41	F (CMOS)	汎用入出力ポートです。	
		A09		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		SOT2		単純シリアル I/O 出力端子です。	
20	18	P42	G (CMOS/H)	汎用入出力ポートです。	
		A10		ノンマルチプレクスモード時, 外部アドレス端子として機能します。	
		SCK2		単純シリアル I/O クロック入出力端子です。	
21 22	19 20	P43, P44	F (CMOS)	MB90480 シリーズ	汎用入出力ポートです。
		A11, A12			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		P43, P44	F (CMOS)	MB90485 シリーズ	汎用入出力ポートです。
		A11, A12			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		MT00, MT01			μPG 出力端子です。

表 1.5-1 端子機能 (3 / 5)

端子番号		端子名	入出力 回路形式 *3	機 能	
QFP *1	LQFP *2				
24	22	P45	F (CMOS)	MB90480 シリーズ	汎用入出力ポートです。
		A13			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		P45	G (CMOS/H)	MB90485 シリーズ	汎用入出力ポートです。
		A13			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		EXTC*4			μPG 入力端子です。
25 26	23 24	P46, P47	F (CMOS)		汎用入出力ポートです。
		A14, A15			ノンマルチプレクスモード時, 外部アドレス端子として機能します。
		OUT4, OUT5			アウトプットコンペアイイベント出力端子として機能します。
70	68	P50	D (CMOS)		汎用入出力ポートです。外部バスモード時, ALE 端子として機能します。
		ALE			外部バスモード時, アドレス取込み許可信号 (ALE) 端子として機能します。
71	69	P51	D (CMOS)		汎用入出力ポートです。外部バスモード時, \overline{RD} 端子として機能します。
		\overline{RD}			外部バスモード時, 読出しストローブ出力 (\overline{RD}) 端子として機能します。
72	70	P52	D (CMOS)		汎用入出力ポートです。外部バスモード時, EPCR レジスタの WRE ビットが "1" の場合, \overline{WRL} 端子として機能します。
		\overline{WRL}			外部バスモード時, 下位側データ書込みストローブ出力 (\overline{WRL}) 端子として機能します。EPCR レジスタの WRE ビットが "0" の場合, 汎用入出力ポートとして機能します。
73	71	P53	D (CMOS)		汎用入出力ポートです。バス幅 16 ビットの外部バスモード時, EPCR レジスタの WRE ビットが "1" の場合, \overline{WRH} 端子として機能します。
		\overline{WRH}			バス幅 16 ビットの外部バスモード時, 上位側データ書込みストローブ出力 (\overline{WRH}) 端子として機能します。EPCR レジスタの WRE ビットが "0" の場合, 汎用入出力ポートとして機能します。
74	72	P54	D (CMOS)		汎用入出力ポートです。外部バスモード時, EPCR レジスタの HDE ビットが "1" の場合, HRQ 端子として機能します。
		HRQ			外部バスモード時, ホールド要求入力 (HRQ) 端子として機能します。EPCR レジスタの HDE ビットが "0" の場合, 汎用入出力ポートとして機能します。
75	73	P55	D (CMOS)		汎用入出力ポートです。外部バスモード時, EPCR レジスタの HDE ビットが "1" の場合, \overline{HAK} 端子として機能します。
		\overline{HAK}			外部バスモード時, ホールドアクノリッジ出力 (\overline{HAK}) 端子として機能します。EPCR レジスタの HDE ビットが "0" の場合, 汎用入出力ポートとして機能します。
76	74	P56	D (CMOS)		汎用入出力ポートです。外部バスモード時, EPCR レジスタの RYE ビットが "1" の場合, RDY 端子として機能します。
		RDY			外部バスモード時, 外部レディ入力 (RDY) 端子として機能します。EPCR レジスタの RYE ビットが "0" の場合, 汎用入出力ポートとして機能します。
78	76	P57	D (CMOS)		汎用入出力ポートです。外部バスモード時, EPCR レジスタの CKE ビットが "1" の場合, CLK 端子として機能します。
		CLK			外部バスモード時, マシンサイクルクロック出力 (CLK) 端子として機能します。EPCR レジスタの CKE ビットが "0" の場合, 汎用入出力ポートとして機能します。

表 1.5-1 端子機能 (4 / 5)

端子番号		端子名	入出力 回路形式 *3	機 能	
QFP *1	LQFP *2				
38 ～ 41	36 ～ 39	P60 ～ P63	H (CMOS)	汎用入出力ポートです。	
		AN0 ～ AN3		アナログ入力端子として機能します。	
43 ～ 46	41 ～ 44	P64 ～ P67	H (CMOS)	汎用入出力ポートです。	
		AN4 ～ AN7		アナログ入力端子として機能します。	
27	25	P70	G (CMOS/H)	汎用入出力ポートです。	
		SIN0		UART データ入力端子として機能します。	
28	26	P71	F (CMOS)	汎用入出力ポートです。	
		SOT0		UART データ出力端子として機能します。	
29	27	P72	G (CMOS/H)	汎用入出力ポートです。	
		SCK0		UART クロック入出力端子として機能します。	
30	28	P73	G (CMOS/H)	汎用入出力ポートです。	
		TIN0		16 ビットリロードタイマのイベント入力端子として機能します。	
31	29	P74	F (CMOS)	汎用入出力ポートです。	
		TOT0		16 ビットリロードタイマの出力端子として機能します。	
32	30	P75	F (CMOS)	MB90480 シリーズ	汎用入出力ポートです。
		P75	G (CMOS/H)	MB90485 シリーズ	汎用入出力ポートです。
		PWC2 *4			PWC 入力端子として機能します。
33	31	P76	F (CMOS)	MB90480 シリーズ	汎用入出力ポートです。
		P76	I (NMOS/H)	MB90485 シリーズ	汎用入出力ポートです。
		SCL *4			I ² C インタフェースデータ入出力端子として機能します。 I ² C インタフェースデータ動作中は、ポート出力を Hi-Z としてください。
34	32	P77	F (CMOS)	MB90480 シリーズ	汎用入出力ポートです。
		P77	I (NMOS/H)	MB90485 シリーズ	汎用入出力ポートです。
		SDA *4			I ² C インタフェースデータ入出力端子として機能します。 I ² C インタフェースデータ動作中は、ポート出力を Hi-Z としてください。
47 48	45 46	P80, P81	E (CMOS/H)	汎用入出力ポートです。	
		IRQ0, IRQ1		外部割込み入力端子として機能します。	
52 ～ 57	50 ～ 55	P82 ～ P87	E (CMOS/H)	汎用入出力ポートです。	
		IRQ2 ～ IRQ7		外部割込み入力端子として機能します。	
58	56	P90	E (CMOS/H)	汎用入出力ポートです。	
		SIN1		単純シリアル I/O データ入力端子として機能します。	
		CS0		チップセレクト 0 です。	
59	57	P91	D (CMOS)	汎用入出力ポートです。	
		SOT1		単純シリアル I/O データ出力端子として機能します。	
		CS1		チップセレクト 1 です。	
60	58	P92	E (CMOS/H)	汎用入出力ポートです。	
		SCK1		単純シリアル I/O クロック入出力端子として機能します。	
		CS2		チップセレクト 2 です。	

表 1.5-1 端子機能 (5 / 5)

端子番号		端子名	入出力 回路形式 *3	機 能	
QFP *1	LQFP *2				
61	59	P93	E (CMOS/H)	汎用入出力ポートです。	
		FRCK		フリーランタイム使用時, 外部クロック入力端子として機能します。	
		ADTG		A/D コンバータ使用時, 外部トリガ入力端子として機能します。	
		CS3		チップセレクト 3 です。	
62	60	P94	D (CMOS)	汎用入出力ポートです。	
		PPG4		PPG タイマ出力端子として機能します。	
63	61	P95	D (CMOS)	汎用入出力ポートです。	
		PPG5		PPG タイマ出力端子として機能します。	
64	62	P96	E (CMOS/H)	汎用入出力ポートです。	
		IN0		インプットキャプチャチャネル 0 のトリガ入力として取り込まれます。	
65	63	P97	E (CMOS/H)	汎用入出力ポートです。	
		IN1		インプットキャプチャチャネル 1 のトリガ入力として取り込まれます。	
66 ～ 69	64 ～ 67	PA0 ～ PA3	D (CMOS)	汎用入出力ポートです。	
		OUT0 ～ OUT3		アウトプットコンペアイイベント出力端子として機能します。	
35	33	AV _{CC}	-	A/D コンバータ電源端子です。	
36	34	AVRH	-	A/D コンバータ外部基準電源端子です。	
37	35	AV _{SS}	-	A/D コンバータ電源端子です。	
49 ～ 51	47 ～ 49	MD0 ～ MD2	J (CMOS/H)	動作モード指定用入力端子です。	
84	82	V _{CC3}	-	3.3V ± 0.3V 用電源端子 (V _{CC3}) です。	
23	21	V _{CC5}	-	MB90480 シリーズ	3.3V ± 0.3V 用電源端子です。 V _{CC} = V _{CC3} = V _{CC5} として 3V 系電源として使用してください。
				MB90485 シリーズ	3V/5V 用電源端子です。 5V I/F 端子として P20 ～ P27, P30 ～ P37, P40 ～ P47, P70 ～ P77 を使用する場合は 5V 電源端子です。 3V 単一の場合は, V _{CC} = V _{CC3} = V _{CC5} として 3V 系電源 として使用してください。
11 42 81	9 40 79	V _{SS}	-	電源入力 (GND) です。	

*1: QFP: FPT-100P-M06

*2: LQFP: FPT-100P-M05

*3: 入出力回路形式については「1.6 入出力回路形式」を参照してください。

*4: MB90V485B のみ CMOS 入力になります。

1.6 入出力回路形式

MB90480/485 シリーズの各端子の入出力回路形式を示します。

■ 入出力回路形式

表 1.6-1 に, MB90480/485 シリーズの各端子の入出力回路形式を示します。

表 1.6-1 入出力回路形式 (1 / 3)

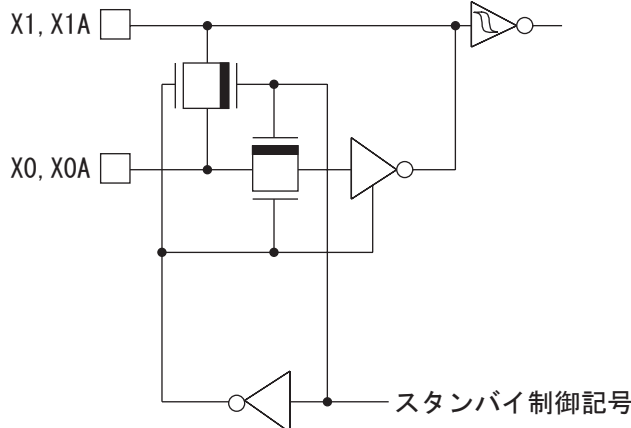
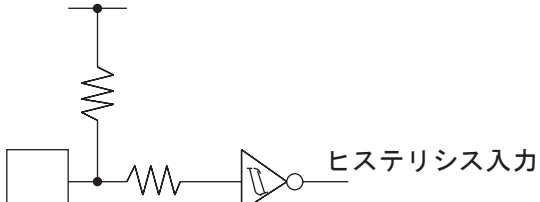
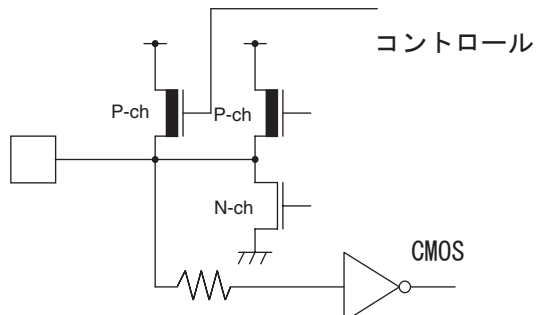
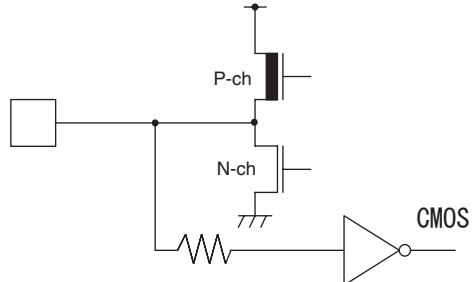
分類	回 路	備 考
A		<ul style="list-style-type: none"> • 発振帰還抵抗 X1, X0 : 約 $1\text{M}\Omega$ X1A, X0A : 約 $10\text{M}\Omega$ • スタンバイコントロール付き
B		<ul style="list-style-type: none"> • プルアップ付きヒステリシス入力
C		<ul style="list-style-type: none"> • 入力プルアップ抵抗制御付き • CMOS レベル入出力
D		<ul style="list-style-type: none"> • CMOS レベル入出力

表 1.6-1 入出力回路形式 (2 / 3)

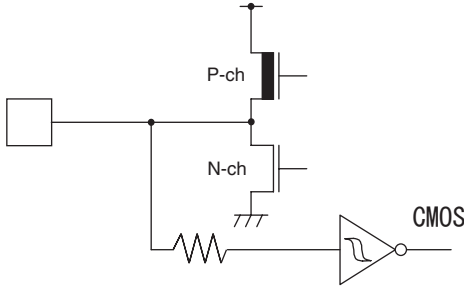
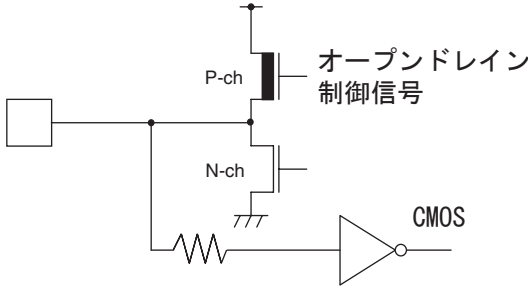
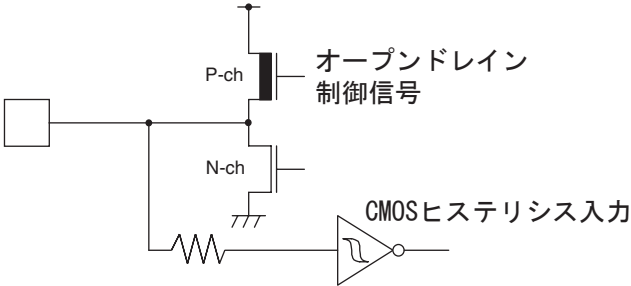
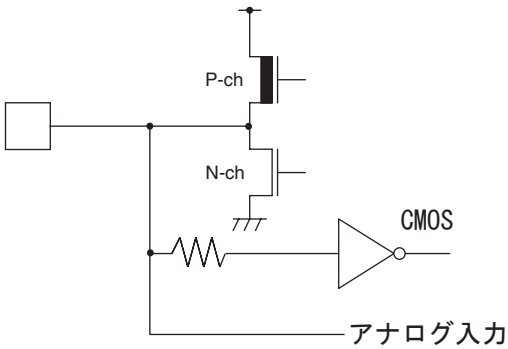
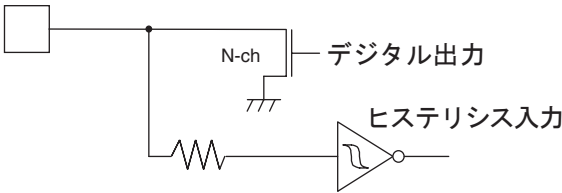
分類	回 路	備 考
E		<ul style="list-style-type: none"> • ヒステリシス入力 • CMOS レベル出力
F		<ul style="list-style-type: none"> • CMOS レベル入出力 • オープンドレインコントロール付き
G		<ul style="list-style-type: none"> • CMOS レベル出力 • ヒステリシス入力 • オープンドレインコントロール付き
H		<ul style="list-style-type: none"> • CMOS レベル入出力 • アナログ入力
I		<ul style="list-style-type: none"> • ヒステリシス入力 • N-ch オープンドレイン出力

表 1.6-1 入出力回路形式 (3 / 3)

分類	回 路	備 考
J	<p>(フラッシュ品)</p>	<p>(フラッシュ品)</p> <ul style="list-style-type: none"> • CMOS レベル入力 • フラッシュテスト用の高電圧制御あり
	<p>(マスク品)</p>	<p>(マスク品)</p> <ul style="list-style-type: none"> • CMOS ヒステリシス入力

1.7 デバイスの取扱い

MB90480/485 シリーズの取扱い上の注意について説明します。

■ デバイス取扱い上の注意

● ラッチアップの防止 , 電源投入について

本デバイスでは , 入力端子や出力端子に V_{CC} より高い電圧や V_{SS} より低い電圧が印加された場合 , V_{CC} 端子と V_{SS} 端子間に絶対最大定格を超える電圧が印加された場合 , または V_{CC} の電圧より AV_{CC} の電源が先に供給された場合に , ラッチアップ現象を生じることがあります。

ラッチアップ現象が起きると電源電流が激増し , 素子の熱破壊に至ることがありますので , 使用に際しては , 注意してください。

アナログ系の電圧印加は , 必ず V_{CC} と同時に行うか , デジタル電源の投入後に行ってください (電源 OFF の場合は , アナログ系電源を先に切断するか , 同時に切断してください) 。

● 未使用端子の処理

入力に用いる未使用端子を開放のままにしておくと , 誤動作およびラッチアップによる永久破壊の原因になることがあります。

使用していない入力端子は $2\text{ k}\Omega$ 以上の抵抗を介してプルアップまたはプルダウンの処理を行ってください。

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

● 電源端子 (V_{CC} 端子または V_{SS} 端子) の取扱い

V_{CC} 端子または V_{SS} 端子が複数ある場合 , デバイスの設計上はラッチアップ , 不要輻射の低減 , グラウンドレベルの上昇によるストロブ信号の誤動作の防止および総出力電流規格の遵守のために , すべての電源端子を , 電源およびグラウンドに接続してください。

電源供給源は , 低インピーダンスで本デバイスの V_{CC} 端子または V_{SS} 端子に接続するように配慮してください。

本デバイスの近くで , V_{CC} 端子と V_{SS} 端子の間に $0.1\mu\text{F}$ 程度のコンデンサをバイパスコンデンサとして接続することをお薦めします。

● 水晶発振回路

高速発振端子 (X0, X1) および 低速発振端子 (X0A, X1A) の近辺のノイズは , 本デバイスの誤動作の原因となります。高速発振端子と低速発振端子に対して , 水晶発振子 (またはセラミック発振子) およびグラウンドへのバイパスコンデンサは , できる限り近くに配置するようプリント基板を設計してください。また , その配線とほかの配線とはできる限り交差させないよう設計してください。

高速発振端子および 低速発振端子の回りをグラウンドで囲むようなプリント板アートワークは , 安定した動作を期待できますので強くお薦めします。

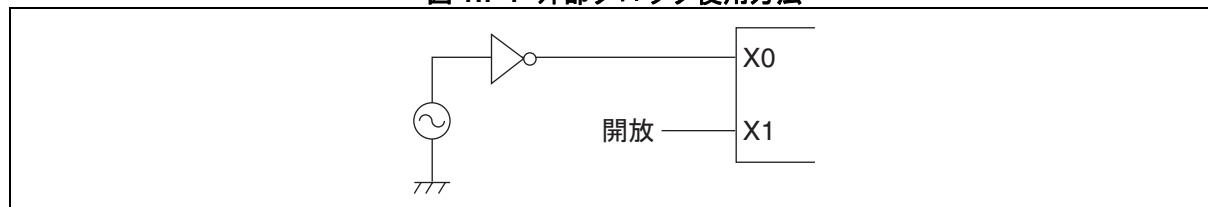
各量産品において , ご使用される発振子メーカーに発振評価依頼をしてください。

● 外部クロック使用時の注意

外部クロックを使用する場合は、X0端子のみを駆動し、X1端子は開放としてください。

図 1.7-1 に外部クロック使用方法を示します (f=25MHz 以下)。

図 1.7-1 外部クロック使用方法



● PLL クロックモード動作中の注意

PLL クロックモードで動作中に、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する場合があります。この動作は、保証外の動作です。

■ 電源に関する取扱い上の注意

● 供給電源の安定化

V_{CC} 電源電圧の動作範囲内においても、電源の急激な変化があると誤動作を生じることがありますので、 V_{CC} 電源電圧を安定させてください。

安定化の基準として、商用周波数 (50/60Hz) での V_{CC} リップル変動 (P-P 値) は、標準 V_{CC} 値の 10% 以下に、また、電源の切換え時の瞬時変化においては、過渡変動率が 0.1V/ms 以下になるように電圧変動を抑えてください。

● 1 系統で使用する場合

MB90480/485 シリーズを 1 系統で使用する場合、 $X0A=V_{SS}$ 、 $X1A=OPEN$ として使用してください。

● フラッシュへの書込みについて

フラッシュシリアル書込みは、必ず、動作電圧 $V_{CC}=3.13V \sim 3.6V$ 内で行ってください。

フラッシュへの通常書込みは、必ず、動作電圧 $V_{CC}=3V \sim 3.6V$ 内で行ってください。

● 2 電源使用時の注意

MB90485 シリーズは、通常 3V 電源を使用しますが、 $V_{CC3}=3V$ 系/ $V_{CC5}=5V$ 系とすることで、P20 ~ P27, P30 ~ P37, P40 ~ P47, P70 ~ P77 をメイン 3V 系電源とは別に、5V 系電源として共用できます。

A/D 変換時のアナログ電源 (AV_{CC} , $AVRH$) は 3V 系としてしか使用できませんので注意してください。

● P90/CS0 端子の取扱い

フラッシュシリアル書込み中は、P90/CS0 は "L" 出力しています。外部からの入力是不行わないでください。

第2章

CPU の機能

MB90480/485 シリーズの CPU の機能について説明します。

2.1 CPU の概略仕様

2.2 メモリ空間

2.3 CPU のレジスタ

2.4 プリフィックスコード

2.1 CPUの概略仕様

CPUの概略仕様を説明します。

■ CPUの概略仕様

F²MC-16LX CPU コアは、民生用機器などの高速リアルタイム処理が要求される用途向けに設計された、16ビットCPUです。F²MC-16LXの命令セットは、コントローラ用途向けに設計され、高速、かつ高効率制御処理が可能です。

F²MC-16LX CPU コアは、16ビットデータはもちろん、32ビットデータ処理も内部32ビットアキュムレータ搭載により可能です。メモリ空間は、最大16Mバイト(拡張可能)、リニアポインタまたはバンク方式のいずれかでアクセス可能です。また、命令体系はF²MC-8LのATアーキテクチャをベースに、高級言語対応命令の追加、アドレッシングモードの拡張、乗除算命令の強化およびビット処理の充実により命令が強化されています。次に、F²MC-16LX CPUの特長を示します。

● 最小命令実行時間

- 40.0ns/6.25MHz 原発振4通倍(内部動作25MHz/3.3V ± 0.3V時)
- 62.5ns/4MHz 原発振4通倍(内部動作16MHz/3.0V ± 0.3V時)
- PLLクロック通倍方式

● 最大メモリ空間:16Mバイト、リニア/バンク方式でアクセス

● コントロール用途に最適化された命令体系

- 取扱い可能なデータタイプ:ビット/バイト/ワード/ロングワード
- 標準アドレッシングモード:23種類/32ビットアキュムレータの採用による高精度演算の強化
- 符号付き乗除算、拡張RETI命令

● 強力な割込み機能: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μDMA

● 高級言語(C言語)マルチタスクに対応をとった命令体系

システムスタックポイントの採用/命令セットの対称性/バレルシフト命令

● 実行速度の向上:4バイトのキュー

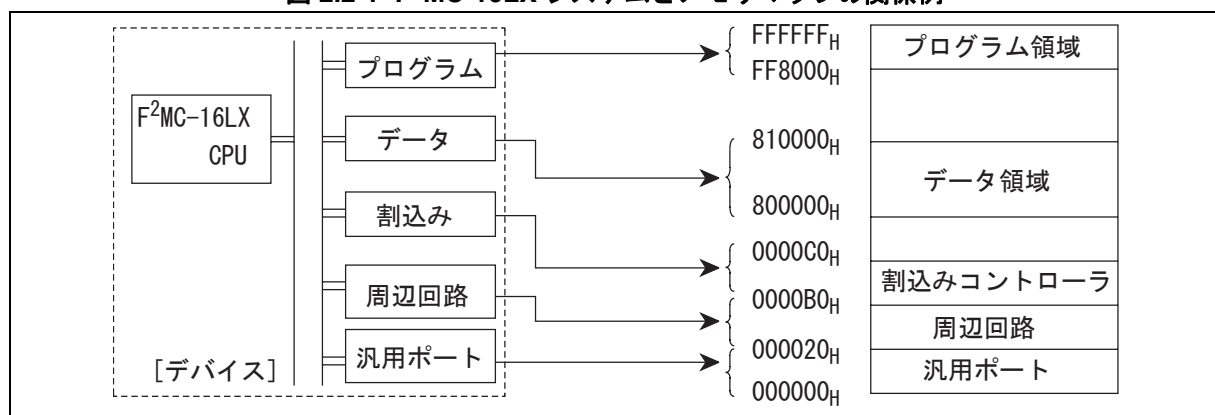
2.2 メモリ空間

F²MC-16LX CPU は 16M バイトのメモリ空間を持ち、F²MC-16LX CPU が管理するデータプログラムの入出力はすべて、この 16M バイトのメモリ空間に配置されます。CPU は 24 ビットのアドレスバスでこれらのアドレスを示し、各リソースにアクセスすることができます。

■ メモリマップ

図 2.2-1 に、F²MC-16LX システムとメモリマップの関係を示します。

図 2.2-1 F²MC-16LX システムとメモリマップの関係例



■ アドレス生成の種類

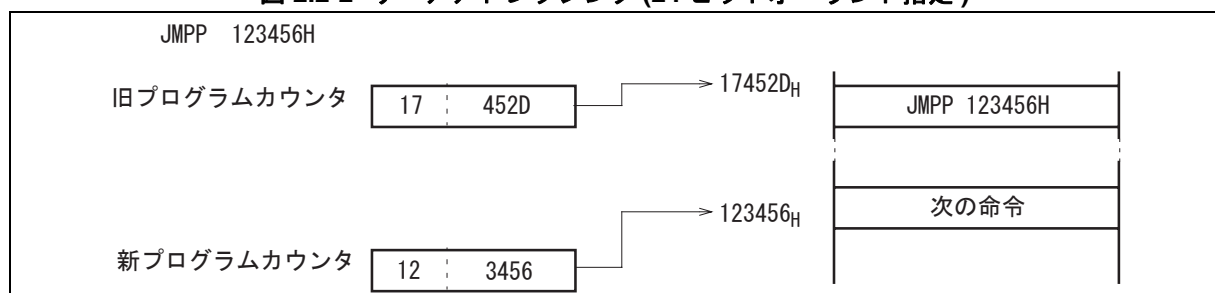
F²MC-16LX CPU には、2 つのアドレス生成方式があります。1 つは、24 ビットアドレスすべてを命令により指定するリニアアドレッシングです。もう 1 つは、アドレス上位 8 ビットを適当なバンクレジスタで、アドレス下位 16 ビットを命令により指定するバンクアドレッシングです。

リニアアドレッシングには、2 種類あり、オペランドで直接 24 ビットのアドレスを指定する方式と、32 ビットの汎用レジスタの内容下位 24 ビットをアドレスとして引用する方式があります。

● リニアアドレッシング (24 ビットオペランド指定)

図 2.2-2 に、24 ビットオペランド指定によるリニアアドレッシングの例を示します。

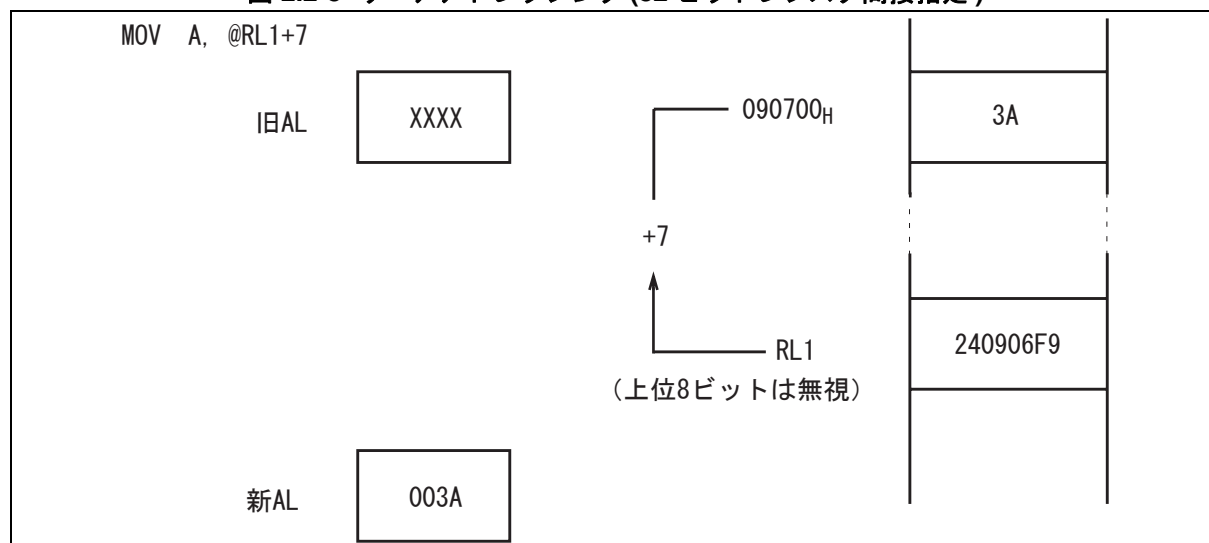
図 2.2-2 リニアアドレッシング (24 ビットオペランド指定)



● リニアアドレッシング (32 ビットレジスタ間接指定)

図 2.2-3 に, 32 ビットレジスタ間接指定によるリニアアドレッシングの例を示します。

図 2.2-3 リニアアドレッシング (32 ビットレジスタ間接指定)



● バンクによるアドレッシングの種類

バンクアドレッシングは, 16M バイトの空間を 64K バイトごとの 256 個のバンクに分割し, 次に示す 5 つのバンクレジスタで各空間に対するバンクを指定します。

- プログラムカウンタバンクレジスタ (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)

PCB によって指定される 64K バイトのバンクをプログラム (PC) 空間とよびます。この PC 空間には, 例えば, 命令コードやベクタテーブルおよび即値データなどが存在します。

DTB によって指定される 64K バイトのバンクをデータ (DT) 空間とよびます。この DT 空間には, 書き込み可能なデータや内外リソースの制御/データレジスタなどが存在します。

USB または SSB によって指定される 64K バイトのバンクをスタック (SP) 空間とよびます。この SP 空間は, プッシュ/ポップ命令や割込みレジスタの退避などの際にスタックアクセスが生じたときにアクセスされます。どのスタック空間をアクセスするかは, コンディションコードレジスタ中の S フラグの値に依存します。

ADB によって指定される 64K バイトのバンクをアディショナル (AD) 空間とよびます。この AD 空間には, 例えば, DT 空間に入りきらなかったデータなどが存在します。

表 2.2-1 に示すように, 各アドレッシングモードで使用されるデフォルト空間が, 命令のコード効率を向上するために事前に決められています。あるアドレッシングモードでデフォルト以外の空間を使用するには, 命令コードに先行してバンクに対応しているプリフィックスコードを指定します。これによって, そのプリフィックスコードに対応した任意のバンク空間にアクセスすることができます。

リセット後, DTB, USB, SSB, ADB は, 00_H に初期化され, PCB は, リセットベクタにより指定された値に初期化されます。

リセット後, DT, SP, AD の各空間は, バンク 00_H(000000_H ~ 00FFFF_H) に配置され, PC

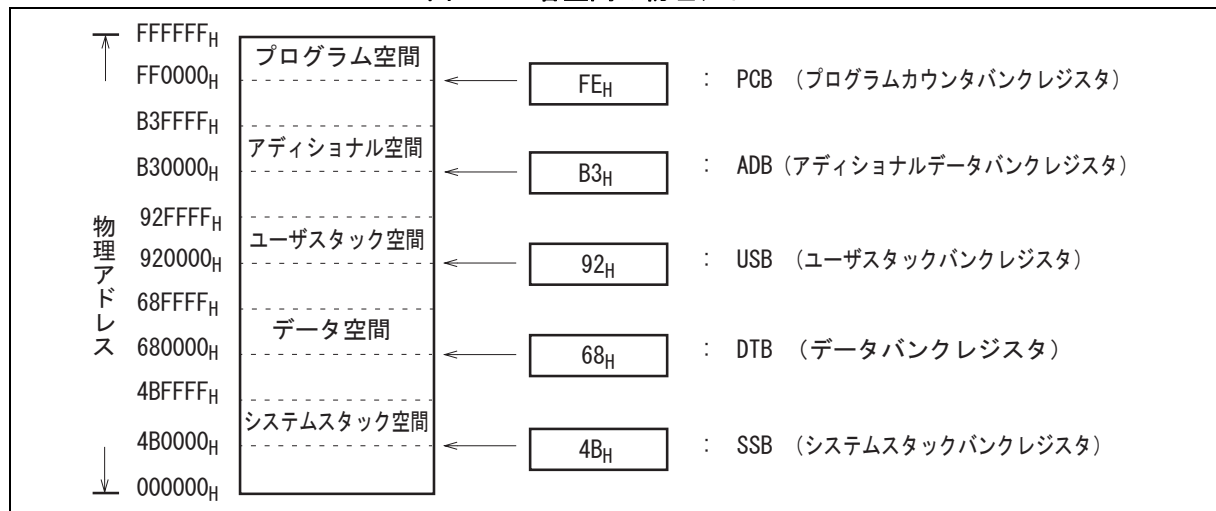
空間はリセットベクタにより指定されたバンクに配置されます。

表 2.2-1 デフォルト空間

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

図 2.2-4 にレジスタバンクの分割されたメモリ空間の例を示します。

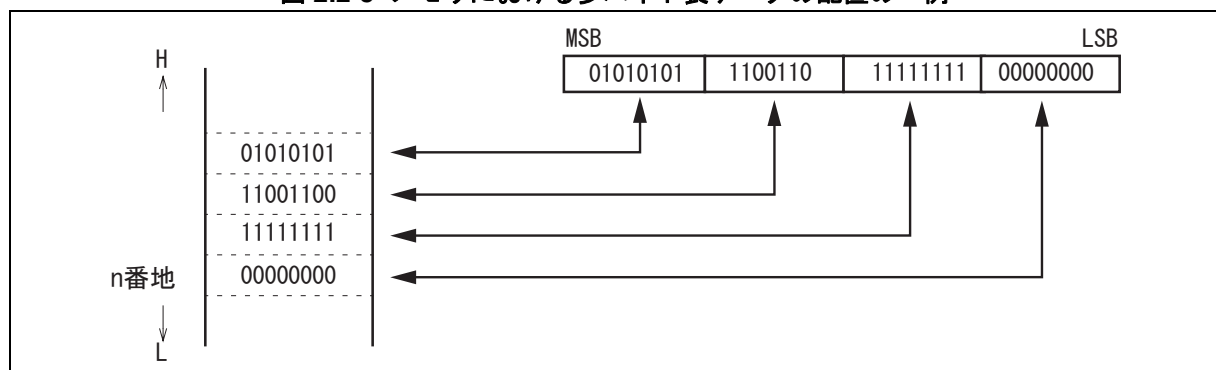
図 2.2-4 各空間の物理アドレス



■ メモリ空間における多バイト長データの配置

図 2.2-5 にメモリにおける多バイト長データの構成を示します。データ項目は, 下位 8 ビットが n 番地に, 以下 n+1 番地, n+2 番地, n+3 番地の順に格納されます。

図 2.2-5 メモリにおける多バイト長データの配置の一例



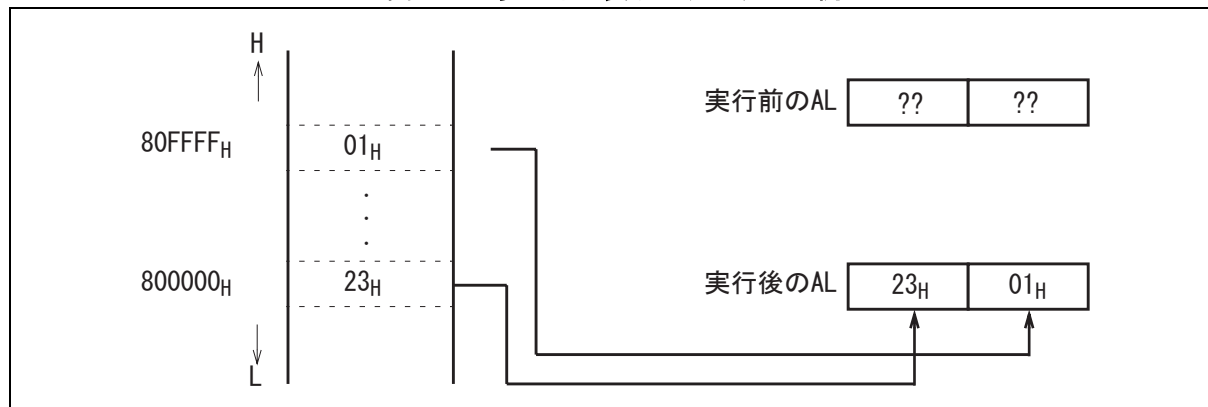
データのメモリへの書込みは, アドレスの低い方から順に行われます。したがって, 32 ビット長のデータ項目であれば下位 16 ビットが先に転送され, 続いて上位 16 ビットが転送されます。なお, 下位ビットの書込み直後にリセット信号を入力すると, 上位ビットが書き込まれないことがあります。

■ 多バイト長データアクセス

図 2.2-6 に、多バイト長データアクセスの例を示します。

これは、MOVW A, 030FFFFH の実行例です。

図 2.2-6 多バイト長データアクセス例



2.3 CPUのレジスタ

F²MC-16LXのレジスタは、CPU内部の特殊レジスタとメモリ上にある汎用レジスタの2種に分けられます。特殊レジスタはCPU内部の専用ハードウェアであり、その用途がCPUのアーキテクチャによって限定されています。汎用レジスタは、CPUのアドレス空間をRAMと共用します。汎用レジスタは、アドレスを指定しないでアクセスができるという点では、特殊レジスタと同じですが、通常のメモリ空間と同じくその用途をユーザが指定できます。

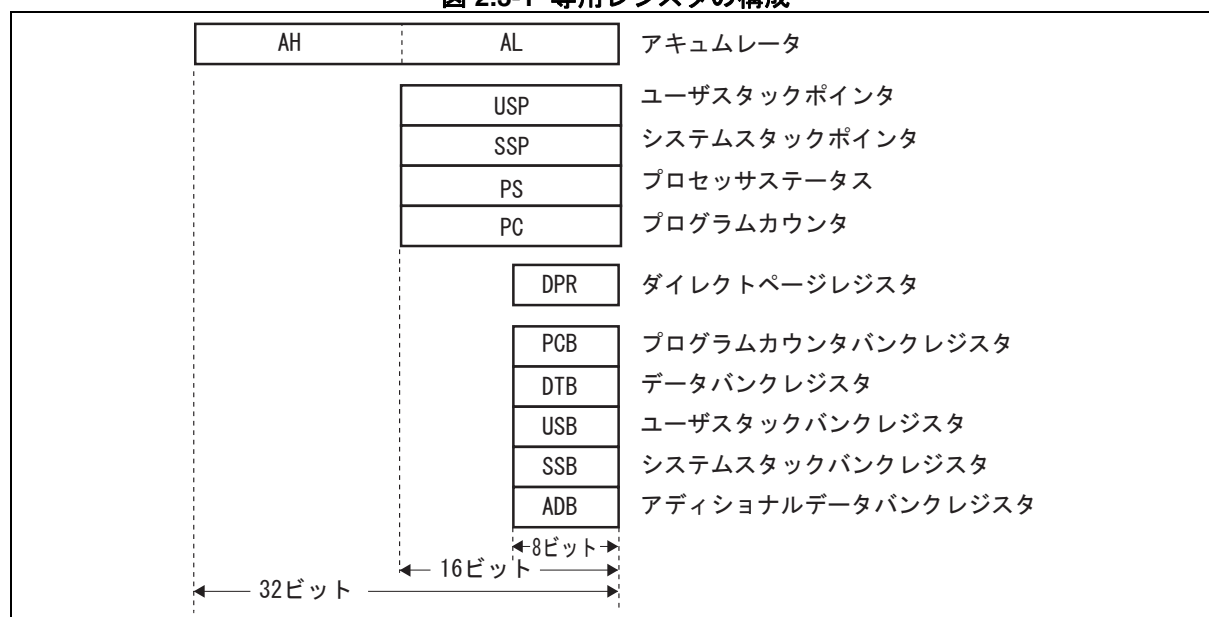
■ 専用レジスタ

F²MC-16LXの専用レジスタは、以下に示す11種類で構成されています。

- アキュムレータ (A=AH:AL):
16ビット×2本のアキュムレータ (単一の32ビットアキュムレータとしても使用可能)
- ユーザスタックポインタ (USP):
ユーザスタック領域を示す16ビットのポインタ
- システムスタックポインタ (SSP):
システムスタック領域を示す16ビットのポインタ
- プロセッサステータス (PS):
システムの状態を示す16ビットのレジスタ
- プログラムカウンタ (PC):
プログラムのアドレスを保持している16ビットのレジスタ
- ダイレクトページレジスタ (DPR):
ダイレクトページを示す8ビットのレジスタ
- プログラムカウンタバンクレジスタ (PCB):
PC空間を示す8ビットのレジスタ
- データバンクレジスタ (DTB):
DT空間を示す8ビットのレジスタ
- ユーザスタックバンクレジスタ (USB):
ユーザスタック空間を示す8ビットのレジスタ
- システムスタックバンクレジスタ (SSB):
システムスタック空間を示す8ビットのレジスタ
- アディショナルデータバンクレジスタ (ADB):
AD空間を示す8ビットのレジスタ

図 2.3-1 に、専用レジスタの構成を示します。

図 2.3-1 専用レジスタの構成



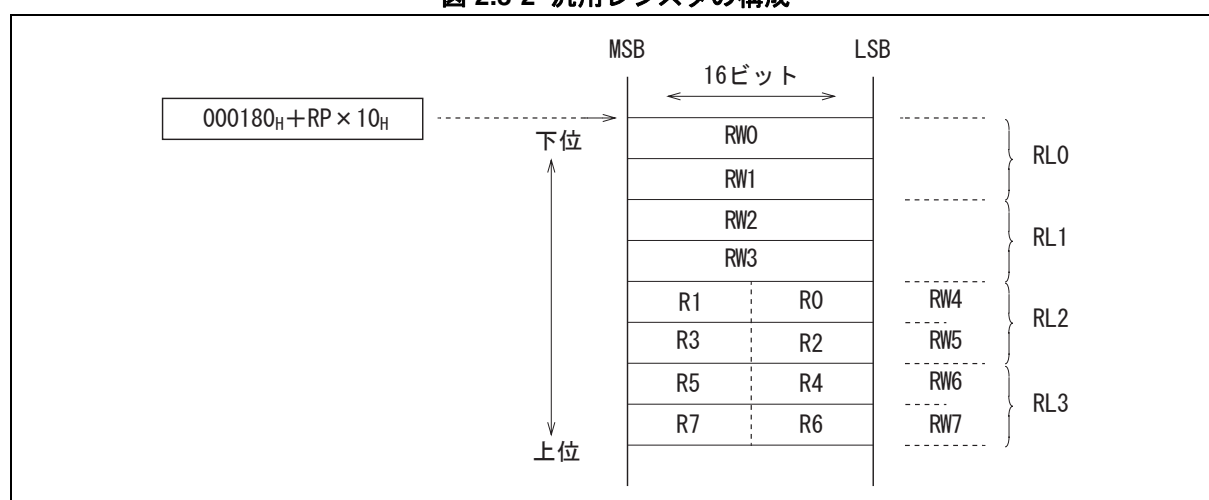
■ 汎用レジスタ

F²MC-16LX の汎用レジスタは、主記憶のアドレス 000180_H ~ 00037F_H(最大構成)に存在し、レジスタバンクレジスタ (RP) によって先に述べたアドレスのどの部分が現在レジスタバンクとして使用されているかを指定します。各バンクには、次に示す 3 種のレジスタが存在します。これらは、図 2.3-2 に示すようにお互いに依存します。

- R0 ~ R7 :8 ビットの汎用レジスタ
- RW0 ~ RW7 :16 ビットの汎用レジスタ
- RL0 ~ RL3 :32 ビットの汎用レジスタ

図 2.3-2 に、汎用レジスタの構成を示します。

図 2.3-2 汎用レジスタの構成



バイトレジスタとワードレジスタの上位/下位バイトの関係は、 $RW(i+4) = RW(i \times 2 + 1) \times 256 + R(i \times 2)$ [$i=0 \sim 3$] という式で表せます。RLi の上位/下位の関係は、 $RW(i) = RW(i \times 2 + 1) \times 65536 + RW(i \times 2)$ [$i=0 \sim 3$] という式で表せます。

2.3.1 アキュムレータ (A)

アキュムレータ (A) の機能について説明します。

■ アキュムレータ (A)

アキュムレータ (A) は、2つの16ビット長の算術演算レジスタ (AH/AL) で構成され、演算結果やデータ転送の一時記憶として使用されます。32ビットデータ処理のときには、AHとALを連結して使用し、16ビットデータ処理モードのワード処理や、8ビットデータ処理モードのバイト処理のときには、ALのみが使用されます。アキュムレータ (A) に格納されたデータは、メモリ/レジスタ (Ri, Rwi, Rli) 中のデータと演算することができます。ワード長以下のデータ項目をALに転送した場合、転送前のAL中のデータ項目が自動的にAHに転送されます (データ保持機能)。データ保持機能とAL-AH間の演算によって処理効率の向上が図られます。

ALへのバイト長以下のデータ項目の転送時は、データは符号拡張またはゼロ拡張され、16ビット長データ項目としてALに格納されます。AL中のデータは、ワード長としてもバイト長としても扱えます。

ALにバイト処理の算術演算命令を実行すると、演算前のALの上位8ビットは無視され、演算結果の上位8ビットはすべてゼロになります。アキュムレータ (A) は、リセットでは初期化されず、リセット直後は不定値となります。

図 2.3-3 に 32 ビットデータ転送の処理を、図 2.3-4 に AL-AH 転送の処理を示します。

図 2.3-3 32 ビットデータ転送

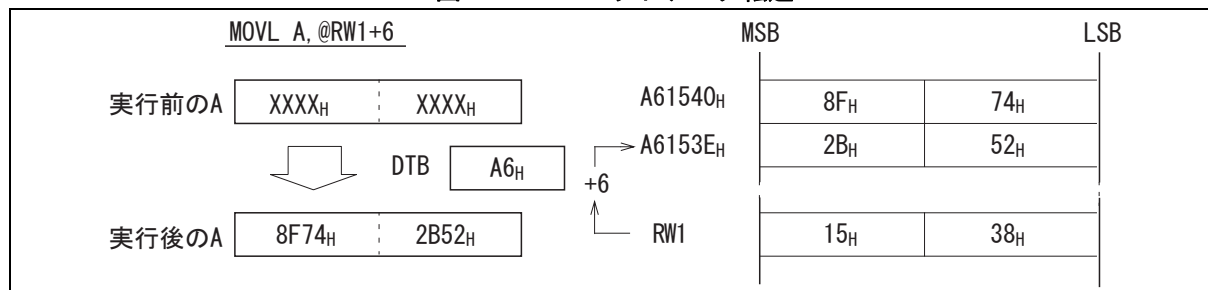
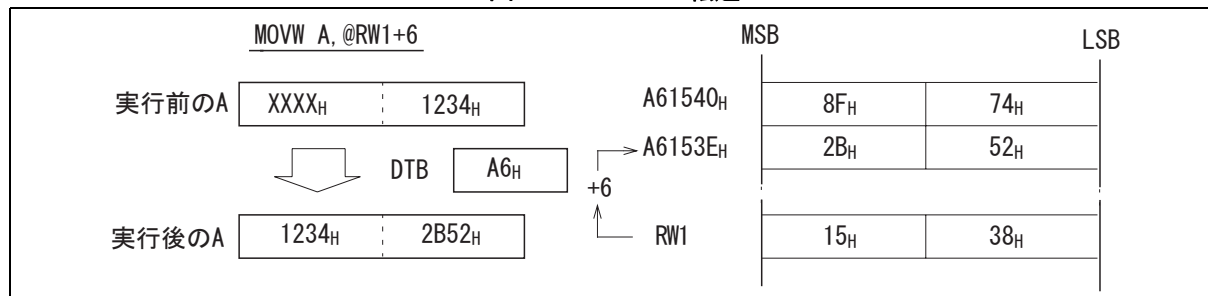


図 2.3-4 AL-AH 転送



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

ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP) の機能について説明します。

■ 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)

ユーザスタックポインタ (USP) およびシステムスタックポインタ (SSP) は、16 ビットのレジスタで、プッシュ / ポップ命令またはサブルーチン実行時のデータ退避 / 復帰のメモリアドレスを示します。USP レジスタと SSP レジスタは、スタック系の命令により使用されますが、プロセッサステータスレジスタ中の S フラグが "0" のときには、USP レジスタが有効になり、S フラグが "1" のときには、SSP レジスタが有効になります (図 2.3-5 参照)。また、割込みが受け付けられると S フラグがセットされるため、レジスタ値は割込み処理の間 SSP が示すメモリ領域に常に退避されます。割込みルーチンでのスタック処理は SSP で、割込みルーチン以外のスタック処理は USP が用いられます。スタック空間を分けない場合、SSP だけを用います。

スタック処理の間のアドレス上位 8 ビットは、SSP → SSB, USP → USB により示されます。また、USP および SSP は、リセットでは初期化されず、不定値になります。

図 2.3-5 に、S フラグが "0" の場合と "1" の場合のスタック操作命令とスタックポインタの関係を示します。

図 2.3-5 スタック操作命令とスタックポインタ



<注意事項> スタックポインタを設定する場合には、偶数アドレスを設定してください。

2.3.3 プロセッサステータス (PS)

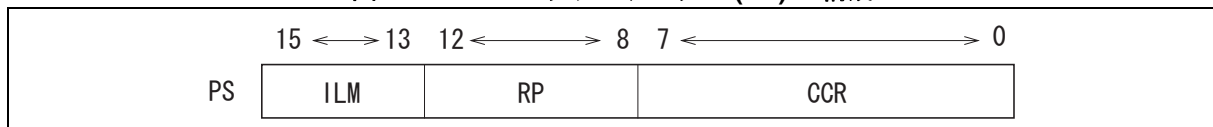
プロセッサステータス (PS) の機能について説明します。

■ プロセッサステータス (PS)

プロセッサステータス (PS) は、CPU 動作を行うビットと CPU 状態を示すビットから構成されます。図 2.3-6 に示すように、PS レジスタの上位バイトはレジスタバンクポインタ (RP) とインタラプトレベルマスクレジスタ (ILM) から構成されます。RP は、レジスタバンクの先頭アドレスを示し、PS レジスタの下位バイトは実行結果、または割込み発生によりセット/リセットされるフラグを含む、コンディションコードレジスタ (CCR) です。

図 2.3-6 に、プロセッサステータス (PS) の構成を示します。

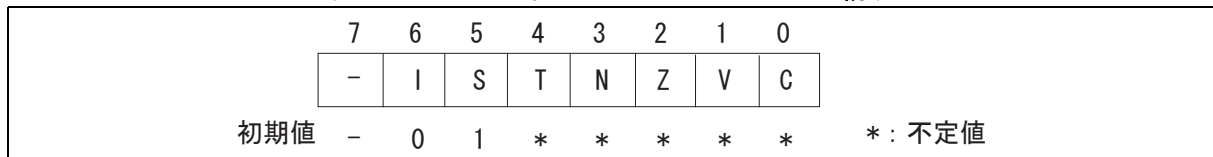
図 2.3-6 プロセッサステータス (PS) の構成



■ コンディションコードレジスタ (CCR)

図 2.3-7 に、コンディションコードレジスタの構成を示します。

図 2.3-7 コンディションコードレジスタの構成



● I: 割込み許可フラグ

ソフトウェアインタラプト以外の割込みは、I フラグが "1" の場合には許可、"0" の場合には禁止します。

I フラグは、リセット時クリアされます。

● S: スタックフラグ

S フラグが "0" の場合にはスタック操作ポインタとして USP が有効、"1" の場合には SSP が有効になります。S フラグは、割込み時、またはリセット時にセットされます。

● T: スティックイビットフラグ

論理右 / 算術右シフト命令を実行後にキャリーよりシフトアウトされたデータに 1 つ以上 "1" があると T フラグに "1" が設定されます。それ以外は "0" がセットされます。シフト量がゼロの場合も T フラグに "0" がセットされます。

● N: ネガティブフラグ

演算結果の MSB が "1" なら N フラグはセットされ、それ以外はクリアされます。

● Z: ゼロフラグ

演算結果がすべて "0" なら Z フラグはセットされ、それ以外はクリアされます。

● V: オーバフローフラグ

演算の実行結果として符号付き数値のオーバーフローが生じた場合、Vフラグはセットされそれ以外はクリアされます。

● C: キャリーフラグ

演算の実行結果として MSB より桁上げ / 桁下げが生じた場合、C フラグはセットされそれ以外はクリアされます。

■ レジスタバンクポインタ (RP)

レジスタバンクポインタ (RP) は、F²MC-16LX の汎用レジスタと内部 RAM アドレスとの関係を示します。RP は、現在使用中のレジスタバンクの先頭メモリアドレスを $[00180_H + RP \times 10_H]$ という変換式で示します。

RP は、5 ビットにより構成されており、 $00_H \sim 1F_H$ までの値をとります。

$000180_H \sim 00037F_H$ のメモリ中にレジスタバンクを配置できます。ただし、この範囲内であってもレジスタバンクが内部 RAM でなければ、汎用レジスタとして使用することはできません。命令上では RP に 8 ビットの即値を転送できますが、実際に使用されるのはそのデータの下位 5 ビットのみです。

図 2.3-8 レジスタバンクポインタ (RP) の構成

	B 4	B 3	B 2	B 1	B 0
初期値	0	0	0	0	0

■ インタラプトレベルマスクレジスタ (ILM)

インタラプトレベルマスクレジスタ (ILM) は 3 ビットから構成されており、CPU の割込みマスクのレベルを示します。この 3 ビットにより表示されるレベルより高い割込みレベルの割込み要求のみが受け付けられます。レベルの高低は "0" が最強で "7" が最弱と定義されています (表 2.3-1 参照)。したがって、割込みが受け付けられるのは、現状の ILM の値より小さいレベル値でなければなりません。割込みが受け付けられるとその割込みのレベル値が ILM にセットされ、これ以降は同じか、それよりも低い優先順位の割込みは、受け付けられなくなります。ILM はリセットによりすべてゼロに初期化されます。命令上では ILM レジスタに 8 ビットの即値を転送できますが、実際に使用されるのはそのデータの下位 3 ビットのみです。

図 2.3-9 にインタラプトレベルマスクレジスタの構成を、表 2.3-1 にインタラプトレベルマスクレジスタ (ILM) で示されるレベルを示します。

図 2.3-9 インタラプトレベルマスクレジスタの構成

	ILM2	ILM1	ILM0
初期値	0	0	0

表 2.3-1 インタラプトレベルマスクレジスタ (ILM) で示されるレベル

ILM2	ILM1	ILM0	レベル値	許可される割込みレベル
0	0	0	0	割込み禁止
0	0	1	1	"0" のみ
0	1	0	2	1 より小さいレベル値
0	1	1	3	2 より小さいレベル値
1	0	0	4	3 より小さいレベル値
1	0	1	5	4 より小さいレベル値
1	1	0	6	5 より小さいレベル値
1	1	1	7	6 より小さいレベル値

2.3.4 プログラムカウンタ (PC)

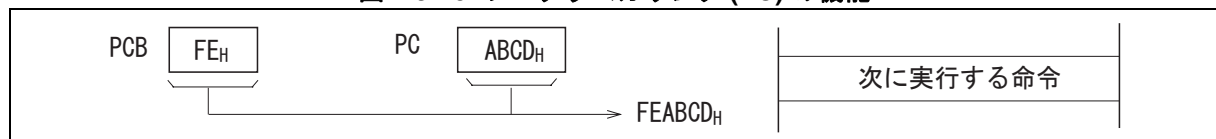
プログラムカウンタ (PC) の機能について説明します。

■ プログラムカウンタ (PC)

PCは、16ビットのカウンタであり、CPUが実行すべき命令コードのメモリアドレスの下位16ビットを示します。上位8ビットのアドレスは、プログラムカウンタバンクレジスタ (PCB) により示されます。PCは条件分岐命令、サブルーチンコール命令、割り込み、リセットなどにより内容が更新されます。また、オペランドアクセス時のベースポインタとしても使用できます。

図 2.3-10 に、プログラムカウンタ (PC) の機能を示します。

図 2.3-10 プログラムカウンタ (PC) の機能



2.3.5 プログラムカウンタバンクレジスタ (PCB)

プログラムカウンタバンクレジスタ (PCB) の機能について説明します。

■ プログラムカウンタバンクレジスタ (PCB)< 初期値 : リセットベクタ中の値 >

プログラムカウンタバンクレジスタ (PCB) は、以下のレジスタから構成されています。

- データバンクレジスタ (DTB)< 初期値 :00_H>
- ユーザスタックバンクレジスタ (USB)< 初期値 :00_H>
- システムスタックバンクレジスタ (SSB)< 初期値 :00_H>
- アディショナルデータバンクレジスタ (ADB)< 初期値 :00_H>

各バンクレジスタは、PC, DT, SP(ユーザ), SP(システム), AD の各空間が配置されるメモリバンクを示します。

すべてのバンクレジスタは1バイト長であり、リセットにより PCB は 00_H に初期化されます。PCB 以外のバンクレジスタは、読出しのみができます。PCB も読出しできますが、書込みはできません。

16M バイト全空間に分岐する JMPP, CALLP, RETP, RETI, RETF 命令実行中、または割込み発生時に PCB は更新されます。各レジスタの動作は、「2.2 メモリ空間」を参照してください。

2.3.6 ダイレクトページレジスタ (DPR)

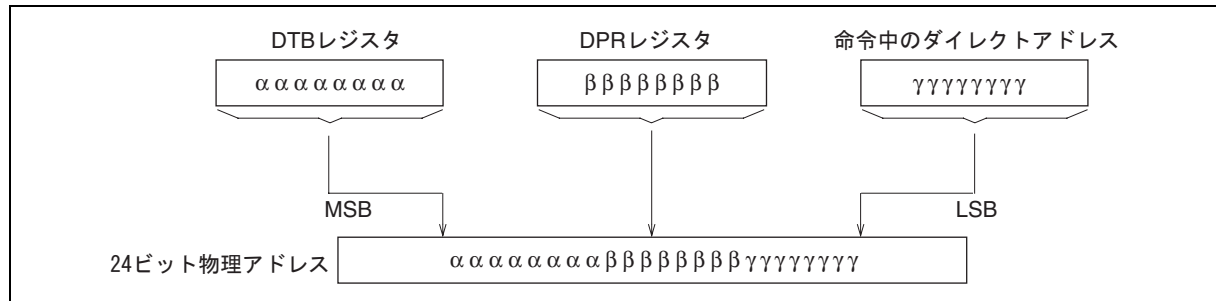
ダイレクトページレジスタ (DPR) の機能について説明します。

■ ダイレクトページレジスタ (DPR)< 初期値 :01_H>

ダイレクトページレジスタ (DPR) は、図 2.3-11 に示すように、ダイレクトアドレッシングモードでの命令オペランドのアドレス 8 ～アドレス 15 を指定します。DPR は 8 ビット長であり、リセットにより 01_H に初期化されます。また、命令により読出し / 書込みができます。

図 2.3-11 に、ダイレクトアドレッシングモードにおける物理アドレスの生成を示します。

図 2.3-11 ダイレクトアドレッシングモードにおける物理アドレスの生成



2.3.7 汎用レジスタ (レジスタバンク)

汎用レジスタ (レジスタバンク) の機能について説明します。

■ 汎用レジスタ (レジスタバンク)

レジスタバンクは 8 ワードで構成され、バイトレジスタ (R0 ~ R7)、ワードレジスタ (RW0 ~ RW7)、ロングワードレジスタ (RL0 ~ RL3) の算術演算用汎用レジスタとして使用できます。また、レジスタバンクを命令ポインタとしても使用できます。表 2.3-2 に各レジスタ機能を、図 2.3-12 に各レジスタの関係を示します。

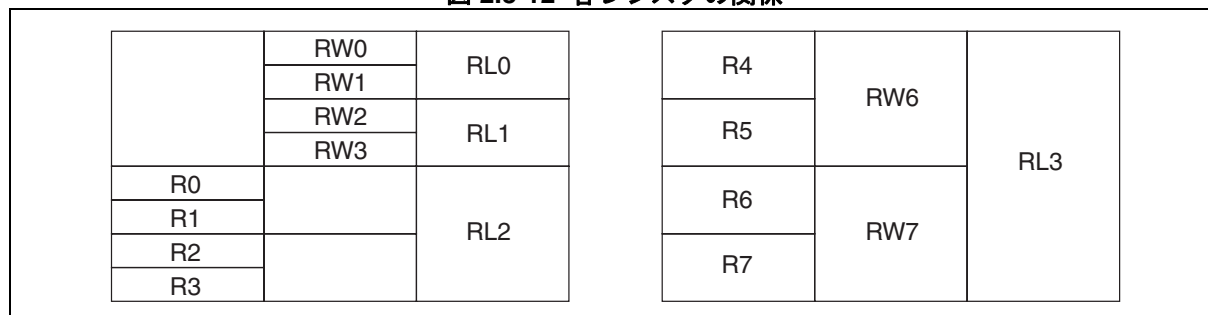
レジスタバンクの値は、通常の RAM 領域と同様に、リセットでは初期化されず、リセット前の状態が保持されます。

ただし、パワーオン時は、不定となります。

表 2.3-2 各レジスタの機能

R0 ~ R7	各種命令のオペランドとして使用 (注意事項) R0 はバレルシフトのカウンタまたは、ノーマライズ命令のカウンタとしても使用
RW0 ~ RW7	ポインタとして使用 / 各種命令のオペランドとして使用 (注意事項) RW0 はストリング命令のカウンタとして使用
RL0 ~ RL3	ロングポインタとして使用 / 各種命令のオペランドとして使用

図 2.3-12 各レジスタの関係



2.4 プリフィックスコード

命令の前にプリフィックスコードを置くことで、命令動作の一部を変更できます。バンクセレクトプリフィックス、コモンレジスタバンクプリフィックス、フラグ変化抑制プリフィックスの3種類のプリフィックスコードを使用できます。

■ バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)

データアクセスに使用されるメモリ空間は、アドレッシングモードごとに決められています。

バンクセレクトプリフィックスを命令の前に置くと、その命令によりデータアクセスに使用するメモリ空間をアドレッシングモードと無関係に選択します。

表 2.4-1 に、バンクセレクトプリフィックスと選択される空間の関係を示します。

表 2.4-1 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	選択される空間
PCB	PC 空間
DTB	データ空間
ADB	AD 空間
SPB	スタックフラグの値により SSB, または USB 空間のどちらかが用いられます。

ただし、次の命令を使用する場合は注意してください。

● スtring命令 (MOVS, MOVSW, SCEQ, SCWEQ, FILS, FILSW)

プリフィックスに無関係に、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)

プリフィックスに無関係に、S フラグに応じて SSB, または USB が使用されます。

● I/O アクセス命令

MOVA A,io/MOV io,A/MOVX A,io/MOVW A,io/MOVW io,A
 MOV io,#imm8/MOVW io,#imm8/MOVB A,io:bp/MOVB io:bp,A
 SETB io:bp/CLRB io:bp/BBC io:bp,rel/BBS io:bp,rel WBTC
 WBTS

バンクの I/O 空間は、プリフィックスの有無にかかわらず使用されます。

● フラグ変更命令 (AND CCR,#imm8, OR CCR,#imm8)

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

● POPW PS

プリフィックスに無関係に S フラグに応じて SSB, または USB が使用されます。プリフィックスの効果が次の命令まで及びます。

● MOV ILM,#imm8

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

● RETI

プリフィックスに無関係に SSB が使用されます。

■ コモンレジスタバンクプリフィックス (CMR)

複数のタスク間でのデータ交換を容易にするには、レジスタバンクポインタ (RP) の値に無関係に比較的簡単に同一のレジスタバンクをアクセスする必要があります。レジスタバンクをアクセスする命令の前に CMR を置くと、その命令は現在の RP の値に関係なく 000180_H ~ 00018F_H のアドレスにあるコモンバンク (RP=0 のときに選択されるレジスタバンク) にアクセスします。

ただし、次の命令を使用する場合は注意してください。

● スtring命令 (MOVS, MOVSW, SCEQ, SCWEQ, FILS, FILSW)

プリフィックスコードを付加したString命令実行中に割り込み要求が発生すると、割り込み処理後にString命令が復帰したとき、プリフィックスコードが無効となります。このため、割り込み処理後にString命令が誤動作します。上記のString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)

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

● MOV ILM,#imm8

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

■ フラグ変化抑止プリフィックス (NCC)

フラグ変化を抑止するには、フラグ変化抑止プリフィックスコード (NCC) を用います。命令の前に NCC を置くことで、その命令に伴うフラグ変化を抑止できます。ただし、次の命令を使用する場合は注意してください。

● String命令 (MOVS, MOVSW, SCEQ, SCWEQ, FILS, FILSW)

プリフィックスコードを付加したString命令実行中に割り込み要求が発生すると、割り込み処理後にString命令が復帰したとき、プリフィックスコードが無効となります。このため、割り込み処理後にString命令が誤動作します。上記のString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)

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

● 割り込み命令 (INT #vct8, INT9, INT addr16, INTO addr24, POPW PS)

プリフィックスに無関係に、CCR は命令の仕様どおりに変化します。

● JCTX@A

プリフィックスに無関係に、CCR は命令の仕様どおりに変化します。

● MOV ILM,#imm8

命令動作は正常ですが、プリフィックスの効果がそのまま次の命令まで及びます。

■ 割込み抑止命令

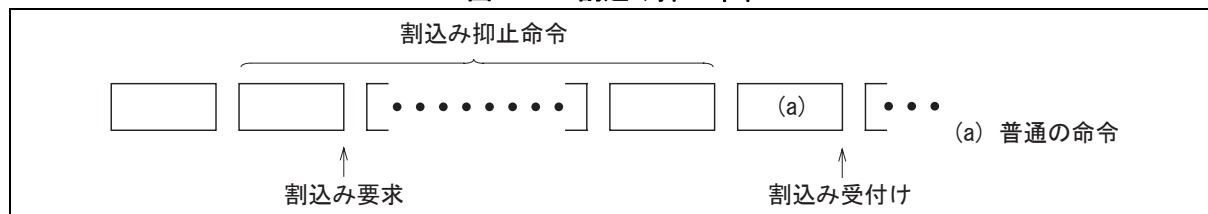
次に示す 10 種類の命令に対しては, 割込み要求は受け付けられません。

MOV ILM,#imm8/PCB/SPB/OR CCR,#imm8/NCC

AND CCR,#imm8/ADB/CMR/POPW PS/DTB

上記命令のどれかを実行中に有効な割込み要求が発生した場合, 上記以外の命令が実行されたときだけ割込み処理をすることができます。詳細は, 図 2.4-1 割込み抑止命令を参照してください。

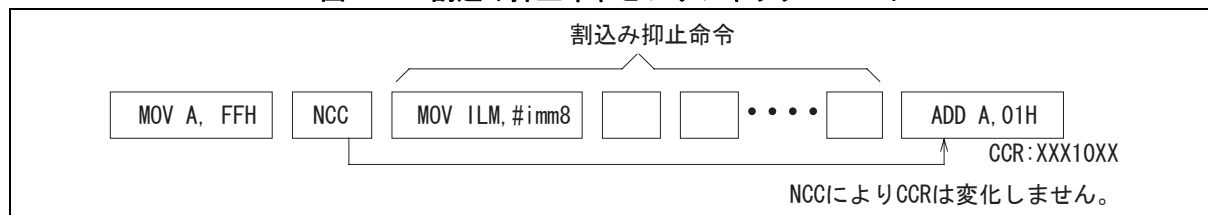
図 2.4-1 割込み抑止命令



■ 割込み抑止命令とプリフィックス命令に関する制約

プリフィックスコードを割込み抑止前に置くと, プリフィックスコードの効果が図 2.4-2 に示すように割込み抑止命令以外のコードの後にある最初の命令まで及びます。

図 2.4-2 割込み抑止命令とプリフィックスコード

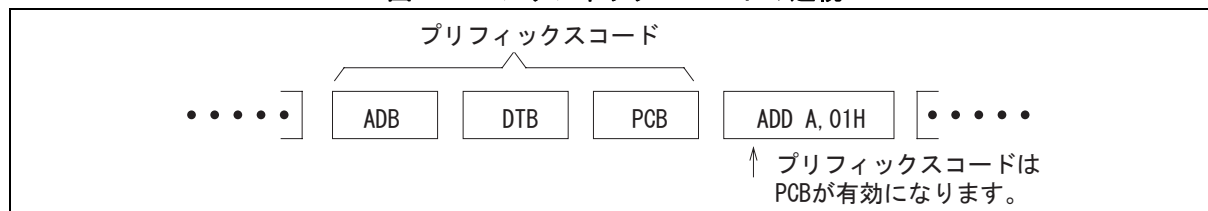


■ 連続したプリフィックスコード

競合するプリフィックスコードが連続している場合, 図 2.4-3 に示すように後方のものが有効になります。

競合するプリフィックスコードとは, 図 2.4-3 に示す PCB, ADB, DTB, SPB のことを意味します。

図 2.4-3 プリフィックスコードの連続



第3章

割込み

割込みとダイレクトアクセス (DMA) について説明します。

- 3.1 割込みの概要
- 3.2 割込み要因と割込みベクタ
- 3.3 割込み制御レジスタと周辺機能
- 3.4 ハードウェア割込み
- 3.5 ソフトウェア割込み
- 3.6 μ DMAC による割込み
- 3.7 拡張インテリジェント I/O サービス (EI²OS) による割込み
- 3.8 例外処理割込み
- 3.9 割込み処理のスタック動作
- 3.10 割込み処理のプログラム例
- 3.11 遅延割込み発生モジュール

3.1 割込みの概要

F²MC-16LX にはイベントなどが発生したとき、現在実行中の処理を中断して、別に定義したプログラムへ制御を移す以下の4つの割込み機能があります。

- ハードウェア割込み
 - ソフトウェア割込み
 - μ DMAC による割込み
 - 例外処理
-

■ 割込みの種類と機能

● ハードウェア割込み

周辺機能からの割込み要求に対して、ユーザの定義した割込み処理用プログラムへ制御を移行する機能です。

● ソフトウェア割込み

ソフトウェア割込み専用の命令 (INT 命令など) の実行によって、ユーザの定義した割込み処理用プログラムへ制御を移行する機能です。

● μ DMAC による割込み

μ DMAC は周辺機能とメモリ間の自動データ転送機能です。従来、割込み処理プログラムで行っていたデータ転送を DMAC(ダイレクトメモリアクセス)のように行うことができます。指定回数のデータ転送処理が終了すると、自動的に割込み処理プログラムを実行します。

μ DMAC による割込みは、ハードウェア割込みの一種です。

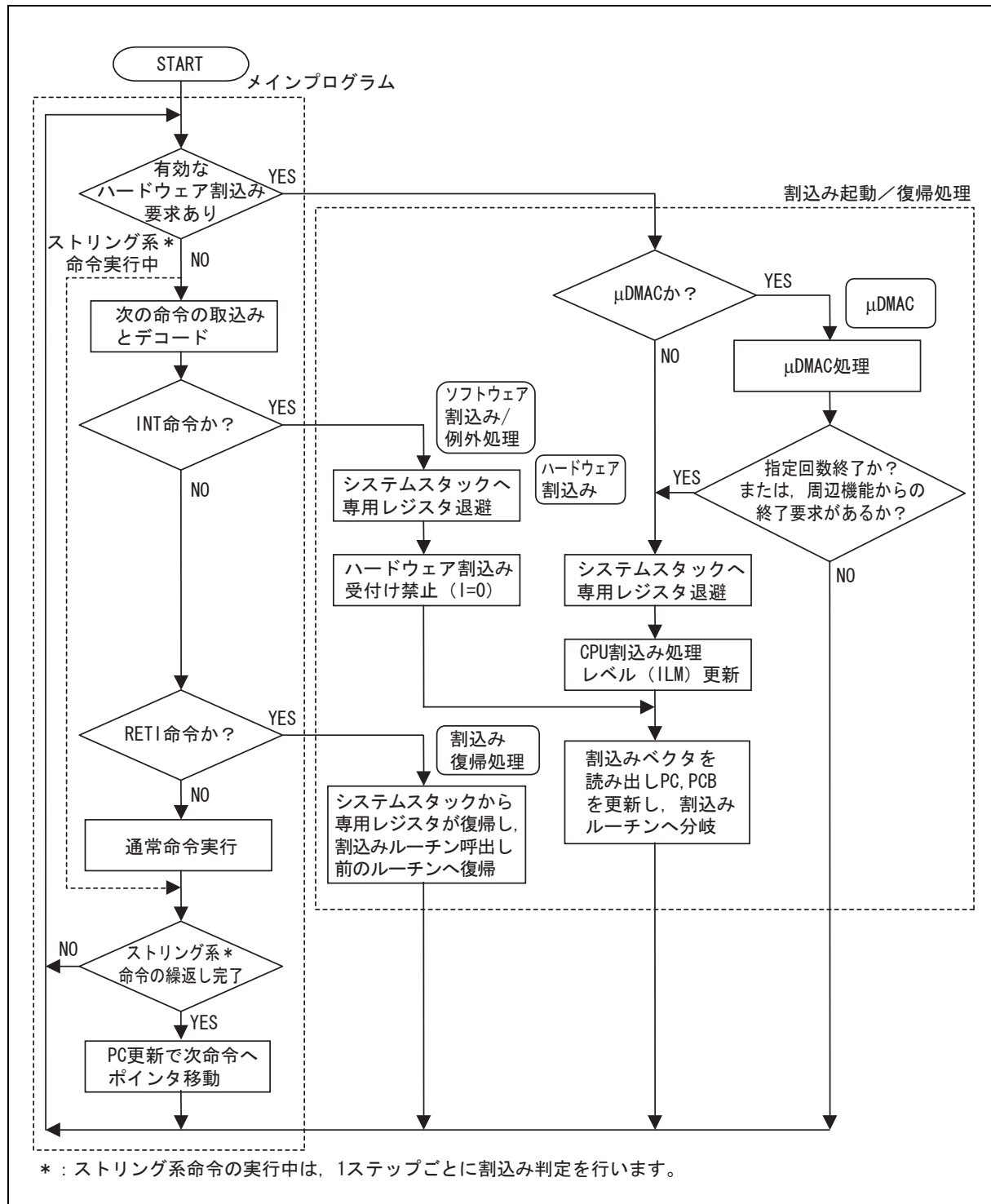
● 例外処理

例外処理は、命令の境目で例外事項(未定義命令の実行)が発生したことを検出した段階で、通常処理を中断して行われます。ソフトウェア割込み命令の "INT10" と等価です。

■ 割り込み動作の概略フロー

4種類の割り込み機能の起動および復帰処理を図3.1-1に示します。

図 3.1-1 割り込み動作の概略フロー



3.2 割り込み要因と割り込みベクタ

F²MC-16LX には、256 種類の割り込み要因に対応する機能があり、メモリの最上位アドレスに 256 組の割り込みベクタテーブルを割り当てています。この割り込みベクタは、すべての割り込みで共有します。

ソフトウェア割り込みは、このすべての割り込み (INT0 ~ INT255) を使用できます。一部の割り込みベクタはハードウェア割り込みや例外処理割り込みで共有されています。また、ハードウェア割り込みでは、各周辺機能に対し、割り込みベクタと割り込み制御レジスタ (ICR) が固定されています。

■ 割り込みベクタ

割り込み処理の際に参照する割り込みベクタテーブルは、メモリ領域の最上位アドレス ("FFFC00_H" ~ "FFFFFF_H") に割り当てられています。また、割り込みベクタは、μDMAC, ハードウェア割り込み、ソフトウェア割り込みおよび例外処理、で同じ領域を共有しています。割り込み番号と割り込みベクタの割当てを表 3.2-1 に示します。

表 3.2-1 割り込みベクター一覧表

ソフトウェア 割り込み命令	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	割り込み No	ハードウェア 割り込み
INT0	FFFFFC _H	FFFFFD _H	FFFFFE _H	未使用	#0	なし
⋮	⋮	⋮	⋮	⋮	⋮	⋮
INT7	FFFFE0 _H	FFFFE1 _H	FFFFE2 _H	未使用	#7	なし
INT8	FFFFDC _H	FFFFDD _H	FFFFDE _H	FFFFDF _H	#8	(RESET ベクタ)
INT9	FFFFD8 _H	FFFFD9 _H	FFFFDA _H	未使用	#9	なし
INT10	FFFFD4 _H	FFFFD5 _H	FFFFD6 _H	未使用	#10	< 例外処理 >
INT11	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	未使用	#11	ハードウェア 割り込み #0
INT12	FFFFCC _H	FFFFCD _H	FFFFCE _H	未使用	#12	ハードウェア 割り込み #1
INT13	FFFFC8 _H	FFFFC9 _H	FFFFCA _H	未使用	#13	ハードウェア 割り込み #2
INT14	FFFFC4 _H	FFFFC5 _H	FFFFC6 _H	未使用	#14	ハードウェア 割り込み #3
⋮	⋮	⋮	⋮	⋮	⋮	⋮
INT254	FFFC04 _H	FFFC05 _H	FFFC06 _H	未使用	#254	なし
INT255	FFFC00 _H	FFFC01 _H	FFFC02 _H	未使用	#255	なし

使われない割り込みベクタも、例外処理などのアドレスに設定することを推奨します。

■ 割込み要因と割込みベクタ，割込み制御レジスタ

表 3.2-2 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 3.2-2 割込み要因と割込みベクタ，割込み制御レジスタ

割込み要因	EI ² OS のクリア	μDMAC チャネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
リセット	×	-	#08	FFFFDC _H	-	-
INT9 命令	×	-	#09	FFFFD8 _H	-	-
例外	×	-	#10	FFFFD4 _H	-	-
INT0 (IRQ0)	○	0	#11	FFFFD0 _H	ICR00	0000B0 _H
INT1 (IRQ1)	○	×	#12	FFFFCC _H		
INT2 (IRQ2)	○	×	#13	FFFFC8 _H	ICR01	0000B1 _H
INT3 (IRQ3)	○	×	#14	FFFFC4 _H		
INT4 (IRQ4)	○	×	#15	FFFFC0 _H	ICR02	0000B2 _H
INT5 (IRQ5)	○	×	#16	FFFFBC _H		
INT6 (IRQ6)	○	×	#17	FFFFB8 _H	ICR03	0000B3 _H
INT7 (IRQ7)	○	×	#18	FFFFB4 _H		
PWC1 (MB90485 シリーズのみ)	○	×	#19	FFFFB0 _H	ICR04	0000B4 _H
PWC2 (MB90485 シリーズのみ)	○	×	#20	FFFFAC _H		
PWC0 (MB90485 シリーズのみ)	○	1	#21	FFFFA8 _H	ICR05	0000B5 _H
PPG0/PPG1 カウンタボロー	×	×	#22	FFFFA4 _H		
PPG2/PPG3 カウンタボロー	×	×	#23	FFFFA0 _H	ICR06	0000B6 _H
PPG4/PPG5 カウンタボロー	×	×	#24	FFFF9C _H		
8/16 ビットアップダウンカウンタ/タイマ (ch.0, ch.1) コンペア/アンダフロー/オーバフロー/アップダウン反転	○	×	#25	FFFF98 _H	ICR07	0000B7 _H
インプットキャプチャ (ch.0) 取込み	○	5	#26	FFFF94 _H		
インプットキャプチャ (ch.1) 取込み	○	6	#27	FFFF90 _H	ICR08	0000B8 _H
アウトプットコンペア (ch.0) 一致	○	8	#28	FFFF8C _H		
アウトプットコンペア (ch.1) 一致	○	9	#29	FFFF88 _H	ICR09	0000B9 _H
アウトプットコンペア (ch.2) 一致	○	10	#30	FFFF84 _H		
アウトプットコンペア (ch.3) 一致	○	×	#31	FFFF80 _H	ICR10	0000BA _H
アウトプットコンペア (ch.4) 一致	○	×	#32	FFFF7C _H		
アウトプットコンペア (ch.5) 一致	○	×	#33	FFFF78 _H	ICR11	0000BB _H
UART 送信完了	○	11	#34	FFFF74 _H		
16 ビットフリーランタイムオーバフロー， 16 ビットリロードタイマアンダフロー *2，	○	12	#35	FFFF70 _H	ICR12	0000BC _H
UART 受信完了	◎	7	#36	FFFF6C _H		
SIO1 (ch.0)	○	13	#37	FFFF68 _H	ICR13	0000BD _H
SIO2 (ch.1)	○	14	#38	FFFF64 _H		
I ² C インタフェース (MB90485 シリーズのみ)	×	×	#39	FFFF60 _H	ICR14	0000BE _H
A/D コンバータ	○	15	#40	FFFF5C _H		
FLASH の書き込み/消去，タイムベースタイマ， 時計タイマ *1	×	×	#41	FFFF58 _H	ICR15	0000BF _H
遅延割込み発生モジュール	×	×	#42	FFFF54 _H		

×：割込み要求フラグはクリアされません。
 ○：割込み要求フラグはクリアされます。
 ◎：割込み要求フラグはクリアされます。ストップ要求あり。
 *1：FLASH 書き込み/消去とタイムベースタイマと時計タイマを同時には使用できませんので注意願います。
 *2：リロードタイマアンダフロー割込みを許可 (TMCSR レジスタの INTE ビット = 1) から禁止 (TMCSR レジスタの INTE ビット = 0) にする場合は、割込み制御レジスタ (ICR12) の IL2 ～ IL0 ビット = "111_B" として割込みを禁止にしてから INTE ビットに "0" を書き込んでください。

<注意事項>

同一割込み番号に2つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2つの要因のどちらか1つがEI²OS/μDMAC機能を使用すると、もう1つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを"0"にして、ソフトウェアポーリング処理で対処してください。

3.3 割り込み制御レジスタと周辺機能

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あり、割り込み機能を持つすべての周辺機能に対応して存在します。このレジスタは、割り込みを制御します。

■ 割り込み制御レジスタ一覧

表 3.3-1 に、割り込み制御レジスタと対応する周辺機能の一覧を示します。

表 3.3-1 割り込み制御レジスタ一覧

アドレス	レジスタ	略称	対応する周辺機能
0000B0 _H	割り込み制御レジスタ 00	ICR00	INT0, 1
0000B1 _H	割り込み制御レジスタ 01	ICR01	INT2, 3
0000B2 _H	割り込み制御レジスタ 02	ICR02	INT4, 5
0000B3 _H	割り込み制御レジスタ 03	ICR03	INT6, 7
0000B4 _H	割り込み制御レジスタ 04	ICR04	PWC1, 2 (MB90485 シリーズのみ)
0000B5 _H	割り込み制御レジスタ 05	ICR05	8/16 ビット PPG タイマ 0, 1 PWC0 (MB90485 シリーズのみ)
0000B6 _H	割り込み制御レジスタ 06	ICR06	8/16 ビット PPG タイマ 2, 3, 4, 5
0000B7 _H	割り込み制御レジスタ 07	ICR07	8/16UD カウンタ 0, 1, インพุットキャプチャ 0
0000B8 _H	割り込み制御レジスタ 08	ICR08	インพุットキャプチャ 1, アウトプットコンペア 0
0000B9 _H	割り込み制御レジスタ 09	ICR09	アウトプットコンペア 1, 2
0000BA _H	割り込み制御レジスタ 10	ICR10	アウトプットコンペア 3, 4
0000BB _H	割り込み制御レジスタ 11	ICR11	アウトプットコンペア 5, UART 送信
0000BC _H	割り込み制御レジスタ 12	ICR12	UART 受信, 16 ビットフリーランタイム, 16 ビットリロードタイマ
0000BD _H	割り込み制御レジスタ 13	ICR13	SIO0, 1
0000BE _H	割り込み制御レジスタ 14	ICR14	A/D, I ² C (MB90485 シリーズのみ)
0000BF _H	割り込み制御レジスタ 15	ICR15	FLASH 書込み, タイムベースタイマ, 時計タイマ, 遅延割り込み発生モジュール

<注意事項>

割り込み制御レジスタ (ICR) に対するリードモディファイライト (RMW) 系命令でのアクセスは、誤動作を引き起こしますので行わないでください。

3.3.1 割り込み制御レジスタ (ICR00 ~ ICR15)

割り込み制御レジスタ (ICR00 ~ ICR15) は、割り込み機能を持つすべての周辺機能に対応し、割り込み要求発生時の処理を制御します。このレジスタは、書込み時と読出し時では、一部の機能が異なります。

■ 割り込み制御レジスタ (ICR00 ~ ICR15) の機能

割り込み制御レジスタ (ICR00 ~ ICR15) は、次に示す4つの機能を持つビットで構成されています。

- 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 (EI²OS) 許可ビット (ISE3)
- 拡張インテリジェント I/O サービス (EI²OS) チャンネル選択ビット (ICS3 ~ ICS0)
- 拡張インテリジェント I/O サービス (EI²OS) ステータスビット (S1, S0)

■ 割り込み制御レジスタ (ICR00 ~ ICR15) の構成

図 3.3-1 に、割り込み制御レジスタ (ICR00 ~ ICR15) のビット構成を示します。

図 3.3-1 割り込み制御レジスタ (ICR00 ~ ICR15) のビット構成

割り込み制御レジスタ (ICR) 書込み時

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000B0 _H									00000111 _B
0000BF _H									
	W	W	W	W	R/W	R/W	R/W	R/W	

割り込み制御レジスタ (ICR) 読出し時

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000B0 _H	—	—	S1	S0	ISE	IL2	IL1	IL0	XX000111 _B
0000BF _H	—	—	S1	S0	ISE	IL2	IL1	IL0	
	—	—	R	R	R/W	R/W	R/W	R/W	

R/W: リード/ライト可能

R : リードオンリ

W : ライトオンリ

— : 未定義

<注意事項>


- ICS3 ~ 0 ビットは、拡張インテリジェント I/O サービス (EI²OS) を起動する場合のみ有効です。EI²OS を起動する場合は、ISE ビットに "1" を設定してください。EI²OS を起動しない場合は、ISE ビットに "0" を設定してください。EI²OS を起動しない場合は ICS3 ~ 0 ビットの設定は不要です。
- ICS1, 0 ビットは書込みのみ可能です。S1, 0 ビットは、読出しのみ可能です。

■ 割込み制御レジスタ (ICR00 ~ ICR15) の各ビットの機能

● 割込みレベル設定ビット (IL2 ~ IL0)

対応する周辺機能の割込みレベルを設定します。リセットにより、レベル 7(割込みなし)に初期化されます。割込みレベル設定ビットと各割込みレベルとの関係を、表 3.3-2 に示します。

表 3.3-2 割込みレベル設定ビットと割込みレベルの対応

IL2	IL1	IL0	割込みレベル
0	0	0	0(最高割込み)
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	6(最低割込み)
1	1	1	7(割込みなし)

● 拡張インテリジェント I/O サービス (EI²OS) 許可ビット (ISE)

割込み要求生成時にこのビットが "1" の場合、EI²OS が起動されます。割込み要求発生時にこのビットが "0" の場合、割込みシーケンスが起動されます。EI²OS 終了条件が満たされる (S1, 0=00_B 以外) と、ISE ビットは "0" にクリアされます。対応するリソースが EI²OS 機能を持っていない場合は、ソフトウェアで ISE ビットに "0" を設定してください。ISE ビットは、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 (EI²OS) チャネル選択ビット (ICS3 ~ ICS0)

ICS3 ~ 0 ビットは書込み専用ビットで、EI²OS のチャネルを指定します。EI²OS ディスクリプタアドレスは、ICS3 ~ 0 ビットに設定した値により決定されます。ICS3 ~ 0 ビットは、リセットにより "0000_B" に初期化されます。

EI²OS チャ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 3.3-3 に示します。

表 3.3-3 EI²OS チャ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	選択されるチャネル	ディスクリプタアドレス
0	0	0	0	0	000100 _H
0	0	0	1	1	000108 _H
0	0	1	0	2	000110 _H
0	0	1	1	3	000118 _H
0	1	0	0	4	000120 _H
0	1	0	1	5	000128 _H
0	1	1	0	6	000130 _H
0	1	1	1	7	000138 _H
1	0	0	0	8	000140 _H
1	0	0	1	9	000148 _H
1	0	1	0	10	000150 _H
1	0	1	1	11	000158 _H
1	1	0	0	12	000160 _H
1	1	0	1	13	000168 _H
1	1	1	0	14	000170 _H
1	1	1	1	15	000178 _H

第3章 割込み

● 拡張インテリジェント I/O サービス (EI²OS) ステータスビット (S1, S0)

S1, 0 ビットは、読出し専用ビットです。S1, 0 ビット値を EI²OS 終了時に確認することにより、動作状態と終了状態を判別することができます。S1, 0 ビットは、リセットにより "00_B" に初期化されます。

S1, 0 ビットと EI²OS 状態の対応関係を表 3.3-4 に示します。

表 3.3-4 EI²OS ステータスビットと EI²OS 状態の対応関係

S1	S0	EI ² OS の状態
0	0	EI ² 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

3.4 ハードウェア割込み

ハードウェア割込みは、周辺機能からの割込み要求信号に対応して、CPU がそれまで実行していたプログラムの実行を一時中断し、ユーザの定義した割込み処理用プログラムに制御を移行する機能です。また、 μ DMAC や外部割込みなどもハードウェア割込みの一種として実行されます。

■ ハードウェア割込みの機能

● ハードウェア割込みの機能

ハードウェア割込みは、周辺機能が出力する割込み要求信号の割込みレベルと、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レジスタ (ILM) とを比較し、かつプロセッサステータス (PS) 内の I フラグの内容をハードウェアで参照し、割込みが受け付けられるかどうかを判定します。

ハードウェア割込みが受け付けられると、自動的に CPU 内部のレジスタ類をシステムスタックへ退避し、インタラプトレベルマスクレジスタ (ILM) に現在要求している割込みレベルを格納したあと、対応する割込みベクタへ分岐します。

● 多重割込み

ハードウェア割込みは、多重に起動できます。

● μ DMAC

μ DMAC は、メモリと I/O 間の自動転送機能です。転送完了時に、ハードウェア割込みが起動されます。なお、 μ DMAC は多重に起動されることはなく、ある μ DMAC の処理中は、ほかの割込み要求および μ DMAC 要求はすべて保留されます。

● 外部割込み

外部割込み (ウェイクアップ割込みを含む) は、周辺機能 (割込み要求検出回路) を通じて、ハードウェア割込みとして受け付けられます。

● 割込みベクタ

割込み処理の際に参照する割込みベクタテーブルは、メモリ "FFFC00_H" ~ "FFFFFF_H" に割り当ててあり、ソフトウェア割込みと共用しています。割込み番号と割込みベクタの割当てについては、「3.2 割込み要因と割込みベクタ」を参照してください。

■ ハードウェア割り込みの構造

ハードウェア割り込みに関連する機構は、表 3.4-1 に示す 4 つの部分に分かれて存在します。

ハードウェア割り込みを使用する場合は、あらかじめプログラムでこの 4 箇所を設定する必要があります。

表 3.4-1 ハードウェア割り込みに関連する機構

	ハードウェア割り込みに関する機構	機 能
周辺機能	割り込み許可ビット, 割り込み要求ビット	周辺機能からの割り込み要求の制御
割り込みコントローラ	割り込み制御レジスタ (ICR)	割り込みレベルの設定および μ DMAC の制御
CPU	割り込み許可フラグ (I)	割り込み許可状態の識別
	インタラプト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の比較
	マイクロコード	割り込み処理ルーチンの実行
メモリ上の "FFFC00 _H " ~ "FFFFFF _H "	割り込みベクタテーブル	割り込み処理時の分岐先アドレスを格納

■ ハードウェア割り込み抑止

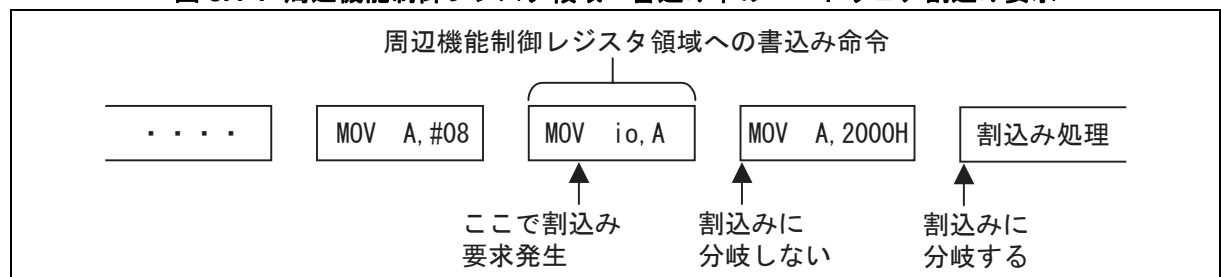
ハードウェア割り込みは、以下の条件において割り込み要求の受け付けが抑止されます。

● 周辺機能制御レジスタ領域への書き込み中のハードウェア割り込み抑止

周辺機能制御レジスタ領域への書き込み中は、ハードウェア割り込み要求を受け付けません。これは各リソースの割り込み制御レジスタ関係の書換えを行っている最中の割り込み要求に対して、CPU が割り込み関係で誤動作を起こすことを避けるためです。周辺機能制御レジスタ領域とは、"000000_H" ~ "0000FF_H" の I/O アドレッシング領域のことではなく、周辺機能制御レジスタの制御レジスタおよびデータレジスタに割り当てられている領域のことを示します。

図 3.4-1 に、周辺機能制御レジスタ領域へ書き込み中のハードウェア割り込み動作について示します。

図 3.4-1 周辺機能制御レジスタ領域へ書き込み中のハードウェア割り込み要求



● 割込み抑止命令のハードウェア割込み抑止

ハードウェア割込み抑止命令を表 3.4-2 に示します。ハードウェア割込み抑止命令を実行中にハードウェア割込み要求が発生した場合は、ハードウェア割込み抑止命令処理後、他の命令が実行された後に割込み処理が行われます。

表 3.4-2 ハードウェア割込み抑止命令

	プリフィックス コード	割込み / 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)
割込みやホールド要求 を受け付けない命令	PCB DTB ADB SPB CMR NCC	MOV ILM,#imm8 OR CCR,#imm8 AND CCR,#imm8 POPW PS

● ソフトウェア割込み実行中のハードウェア割込み抑止

ソフトウェア割込みを起動すると、I フラグを "0" にクリアするために、ほかの割込み要求は受け付けられません。

3.4.1 ハードウェア割込みの動作

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動作について説明します。

■ ハードウェア割込みの起動

● 周辺機能の動作 (割込み要求の発生)

ハードウェア割込み要求の機能を持つ周辺機能は、割込みの要求のあり / なしを示す「割込み要求フラグ」と、CPU への割込み要求の許可 / 禁止を選択する「割込み許可フラグ」を持っています。割込み要求フラグは、周辺機能固有のイベントの発生によってセットされ、割込み許可フラグが「許可」であったときに、割込みコントローラへ割込み要求を発生します。

● 割込みコントローラの動作 (割込み要求の制御)

割込みコントローラは、同時に受け取った割込み要求の割込みレベル (IL) どうしを比較し、最も高いレベルの要求 (IL の値の最も小さいもの) を採択して CPU へ通知します。同一レベルの要求が複数あった場合には、割込み番号が小さいものを優先します。

● CPU の動作 (割込み要求の受け付けと割込み処理)

CPU は受け取った割込みレベル (ICR の IL2 ~ IL0) とインタラプトレベルマスクレジスタ (ILM) とを比較し、 $IL < ILM$ で、なおかつ割込みが許可 (PC の CCR の I=1) されている場合に、現在実行中の命令が終了後、割込み処理マイクロコードを起動し、割込み処理を実行します。

割込み処理では、まず、システムスタック (SSB と SSP の示すシステムスタック空間) へ専用レジスタ (A, DPR, ADB, DTB, PCB, PC, PS の 12 バイト) の内容を退避します。次に、割込みベクタのプログラムカウンタ (PCB, PC) へのロード、ILM の更新、スタックフラグ (S) の設定 (CCR の S=1 とし、システムスタックを有効にする) を行います。

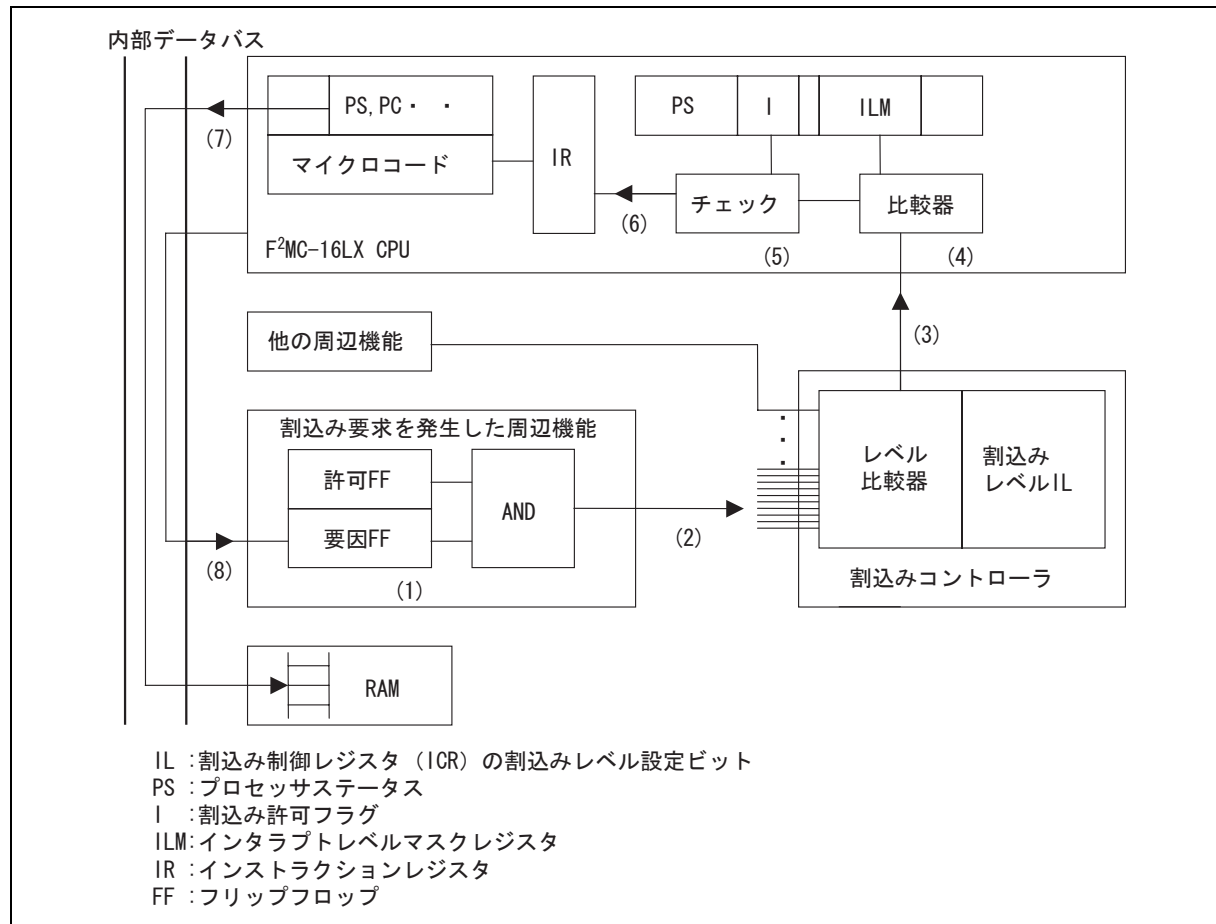
■ ハードウェア割込みからの復帰

割込み処理プログラムの中で、割込み要因となった周辺機能の割込み要求フラグをクリアして、RETI 命令を実行すると、システムスタックに退避している 12 バイトデータを専用レジスタに戻し、割込み分岐前に実行していた処理に復帰します。割込み要求フラグをクリアすることで、周辺機能が割込みコントローラへ出力していた割込み要求は自動的に取り下げられます。

■ ハードウェア割り込みの動作

図 3.4-2 に、ハードウェア割り込みの発生から割り込み処理完了までの動作を示します。

図 3.4-2 ハードウェア割り込みの動作



- (1) 周辺機能の内部で割り込み要因が発生します。
- (2) 周辺機能の割り込み許可ビットを参照し、割り込み許可になっていれば、周辺から割り込みコントローラへ割り込み要求を出力します。
- (3) 割り込み要求を受け取った割り込みコントローラは、同時に要求のあった割り込みの優先順位を判定したうえで、該当する割り込み要求に対応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から要求のあった割り込みレベル (IL) を、インタラプトレベルマスクレジスタ (ILM) と比較します。
- (5) 比較の結果が現在の割り込み処理レベルより優先順位が高い場合、コンディションコードレジスタ (CCR) の I フラグの内容をチェックします。
- (6) (5) のチェックの結果、I フラグが割り込み許可 (I=1) の場合、現在実行中の命令の実行が終了するまで待ち、終了時点で ILM に要求されたレベル (IL) を設定します。
- (7) レジスタを退避し、割り込み処理ルーチンへ分岐します。
- (8) 割り込み処理ルーチン内のソフトウェアによって、(1) で発生した割り込み要因をクリアし、RETI 命令を実行することによって割り込み処理が終了します。

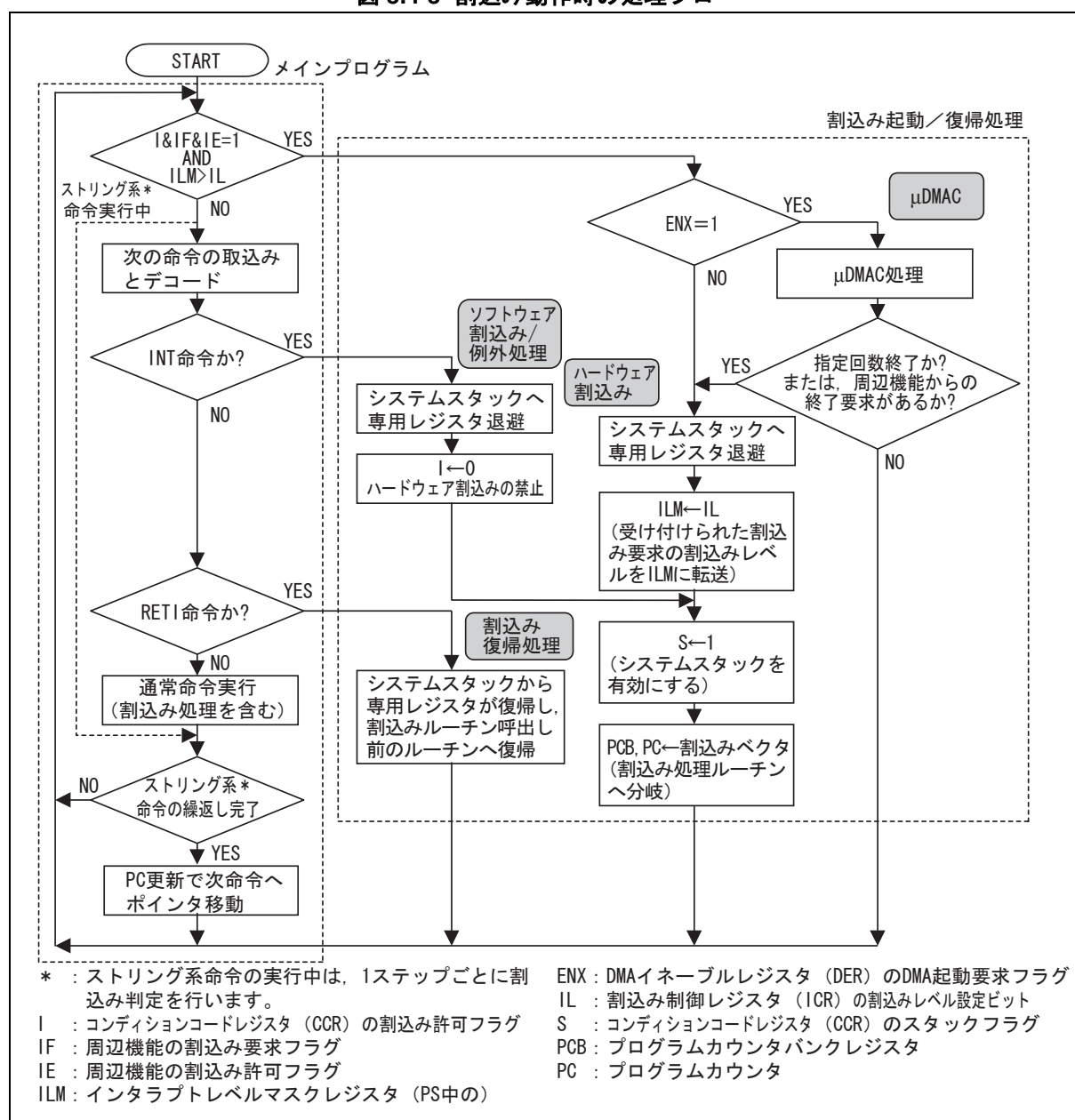
3.4.2 ハードウェア割込みの動作フロー

周辺機能から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 に伝達し、CPU が割込みを受け付けられる状態であれば、現在実行中の命令を一時中断して割込み処理ルーチンの実行、または μ DMAC の起動を行います。また、INT 命令によるソフトウェア割込みが発生した場合は、CPU の状態にかかわらず割込み処理ルーチンを実行します。このとき、ハードウェア割込みは禁止されます。

■ ハードウェア割込みの動作フロー

図 3.4-3 に、割込み動作時の処理フローを示します。

図 3.4-3 割込み動作時の処理フロー



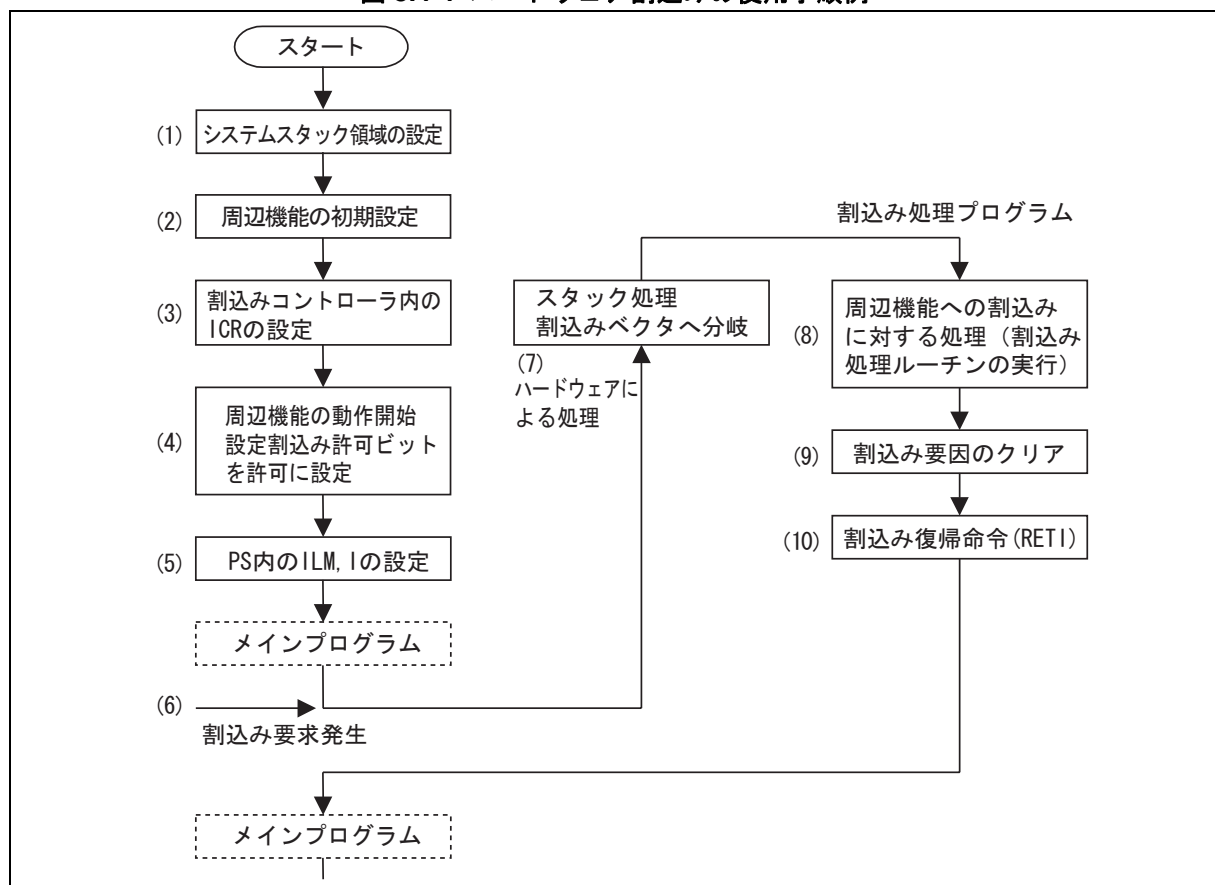
3.4.3 ハードウェア割り込みの使用手順

ハードウェア割り込みを使用するには、システムスタック領域、周辺機能および割り込み制御レジスタ (ICR) などの設定が必要です。

■ ハードウェア割り込みの使用手順

ハードウェア割り込みの使用手順の一例を、図 3.4.4 に示します。

図 3.4-4 ハードウェア割り込みの使用手順例



- (1) システムスタック領域を設定します。
- (2) 割り込み要求を発生可能な周辺機能の初期設定をします。
- (3) 割り込みコントローラ内の割り込み制御レジスタ (ICR) を設定します。
- (4) 周辺機能を動作開始状態にし、割り込み許可ビットを許可に設定します。
- (5) 割り込みレベルマスケジスタ (ILM) および割り込み許可フラグ (I) を割り込み受け可能に設定します。
- (6) 周辺機能の割り込み発生で、ハードウェア割り込み要求が発生します。
- (7) 割り込み処理ハードウェアでレジスタの退避が行われ、割り込み処理プログラムに分岐します。
- (8) 割り込み処理プログラムで、割り込み発生に対する周辺機能への処理をします。
- (9) 周辺機能の割り込み要求を解除します。
- (10) 割り込み復帰命令を実行し、分岐前のプログラムに復帰します。

3.4.4 多重割込み

ハードウェア割込みでは、周辺機能からの複数の割込み要求に対して、割込み制御レジスタ (ICR) の割込みレベル設定ビット (IL0 ~ IL2) に異なる割込みレベルを設定することによって、多重割込みを実現できます。ただし、 μ DMAC の多重起動はできません。

■ 多重割込み動作

割込み処理ルーチン実行中に、より高い割込みレベルの割込み要求が発生すると、現在の割込み処理を中断してより高い割込み要求を受け付けます。高いレベルの割込みが終了すると、元の割込み処理に戻ります。割込みレベルは0 ~ 7まで設定できますが、レベル7に設定した場合は、CPU は割込み要求を受け付けません。

割込み処理実行中にその割込みと同等か、より低いレベルの割込みが発生した場合、I フラグか ILM を変更しない限り、現在の割込み終了するまで新しい割込み要求は保留されます。また、割込み処理ルーチン内でコンディションコードレジスタ (CCR) の I フラグを割込み禁止 (CCR の I=0) に設定するか、インタラプトレベルマスクレジスタ (ILM) を割込み禁止 (ILM=000) に設定すると、割込み中の多重割込みの起動を部分的に禁止することができます。

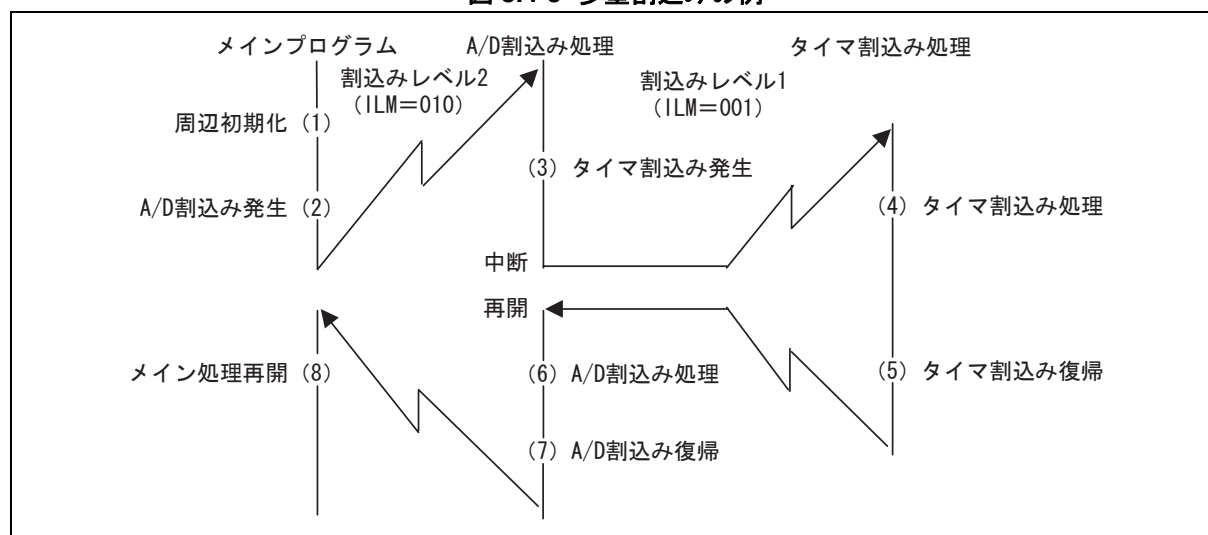
<注意事項>

μ DMACは多重起動できません。 μ DMACの処理中は、ほかの割込み要求および μ DMAC要求はすべて保留されます。

■ 多重割り込みの例

多重割り込み処理の例として、A/Dコンバータよりタイマ割り込みを優先させる場合を想定し、A/Dコンバータの割り込みレベルを"2"に、タイマ割り込みレベルを"1"に設定します。このとき、A/Dコンバータの割り込み処理中にタイマ割り込みが発生すると、図3.4-5のような処理を行います。

図 3.4-5 多重割り込みの例



● A/D 割り込み発生

A/Dコンバータ割り込み処理開始時に、インタラプトレベルマスクレジスタ (ILM) が自動的にA/Dコンバータの割り込みレベル (ICRのIL2～IL0)と同じ値 (例では2) になります。このとき、レベル1, またはレベル0の割り込み要求が発生すると、この割り込み処理を優先して行います。

● 割り込み処理の終了

割り込み処理が終了して復帰命令 (RETI) を実行すると、スタック内に退避した専用レジスタ (A, DPR, ADB, DTB, PCB, PC, PS) の値が復帰し、インタラプトレベルマスクレジスタ (ILM) は中断前の値になります。

3.4.5 ハードウェア割込みの処理時間

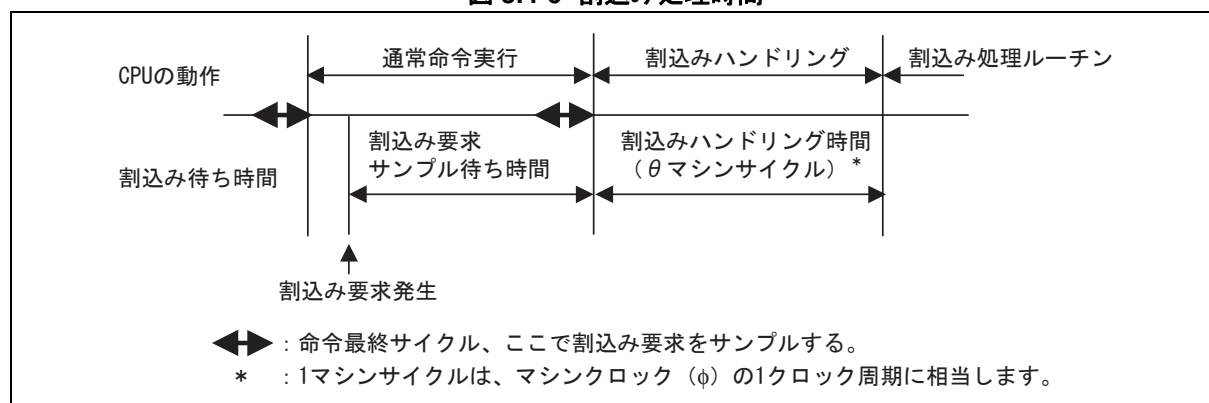
ハードウェア割込み要求が発生して割込み処理ルーチンが実行されるまでには、現在実行中の命令が終了するまでの時間と、割込みハンドリング時間が必要です。

■ ハードウェア割込み処理時間

割込み要求が発生し、割込み処理ルーチンが実行されるまでには、割込み要求サンプル待ち時間と割込みハンドリング時間（割込み処理準備に要する時間）が必要です。

図 3.4-6 に、割込み処理時間を示します。

図 3.4-6 割込み処理時間



● 割込み要求サンプル待ち時間

割込み要求サンプル待ち時間とは、割込み要求が発生してから現在実行中の命令が終了するまでの時間です。割込み要求が発生しているかどうかは、各命令の最後のサイクルで割込み要求をサンプリングして判断します。そのため、各命令の実行中は、CPU が割込み要求を認識することができず、待ち時間が発生します。

なお、割込み要求サンプル待ち時間は、最も実行サイクルの長いPCPW, PW0,...RW7 命令 (45 マシンサイクル) 開始直後に割込み要求が発生した場合、最大となります。

● 割込みハンドリング時間 (θ マシンサイクル)

CPU は、割込み要求を受け付けてから、専用レジスタのシステムスタックへの退避および割込みベクタの取込みなどを行うため、割込みハンドリング時間は、以下の式によって求められます。

- 割込み起動時 : $\theta = 24 + 6 \times Z$ マシンサイクル
- 割込み復帰時 : $\theta = 11 + 6 \times Z$ マシンサイクル (RETI 命令)

割込みハンドリング時間は、スタックポインタの指しているアドレスによって異なります。表 3.4-3 に割込みハンドリング時間の補正值 (Z) を示します。

1 マシンサイクルは、マシクロック (φ) の 1 クロック周期に相当します。

表 3.4-3 割込みハンドリング時間の補正值 (Z)

スタックポインタが指しているアドレス	補正值 (Z)
外部 8 ビットの場合	+4
外部 偶数アドレスの場合	+1
外部 奇数アドレスの場合	+4
内部 偶数アドレスの場合	0
内部 奇数アドレスの場合	+2

3.5 ソフトウェア割込み

ソフトウェア割込みは、ソフトウェア割込み命令 (INT 命令) を実行すると、CPU がそれまで実行していたプログラムからユーザの定義した割込み処理用のプログラムへ制御を移す機能です。

ソフトウェア割込みの実行中、ハードウェア割込みは停止します。

■ ソフトウェア割込みの起動

● ソフトウェア割込みの起動

ソフトウェア割込みの起動は、INT 命令を使用します。ソフトウェア割込み要求は、割込み要求フラグおよび許可フラグはなく、INT 命令を実行すると常に割込み要求が発生します。

● ハードウェア割込みの抑止

INT 命令には割込みレベルがないので、インタラプトレベルマスクレジスタ (ILM) の更新は行われません。INT 命令実行中は、コンディションコードレジスタ (CCR) の I フラグが "0" に設定され、ハードウェア割込みを抑止します。ソフトウェア割込み処理中でもハードウェア割込みを許可するときは、ソフトウェア割込み処理ルーチンの中で I フラグを "1" に設定してください。

● ソフトウェア割込みの動作

CPU が INT 命令を取り込んで実行すると、ソフトウェア割込み処理用のマイクロコードを起動します。このマイクロコードにより、CPU 内部のレジスタ類をシステムスタックに退避し、ハードウェア割込みを抑止 (CCR の I=0) したあと、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は、「3.2 割込み要因と割込みベクタ」を参照してください。

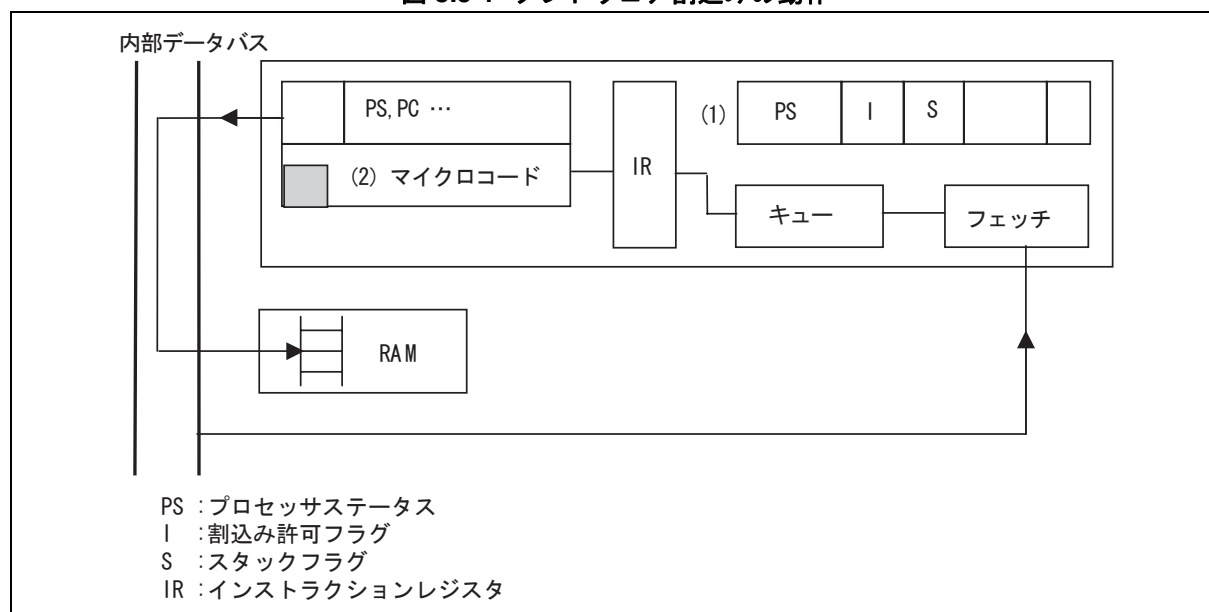
■ ソフトウェア割込みからの復帰

割込み処理プログラムの中で割込み復帰命令 (RETI 命令) を実行すると、システムスタックに退避している 12 バイトデータを専用レジスタに戻して、割込み分岐前に実行していた処理に復帰します。

■ ソフトウェア割り込みの動作

図 3.5-1 に、ソフトウェア割り込みの発生から割り込み処理完了までの動作を示します。

図 3.5-1 ソフトウェア割り込みの動作



- (1) ソフトウェア割り込み命令を実行します。
- (2) ソフトウェア割り込み命令に対応したマイクロコードに従って専用レジスタを退避するなどの必要な処理を行ったあと、分岐処理を行います。
- (3) ユーザの割り込み処理ルーチン内の RETI 命令で割り込み処理が終了します。

■ ソフトウェア割り込みに関する注意事項

プログラムカウンタバンクレジスタ (PCB) が "FF_H" の場合、CALLV 命令のベクタ領域は INT#vct8 命令のテーブルと重なります。プログラム作成の際には、CALLV 命令と INT#vct8 命令のアドレスの重複に注意してください。

3.6 μ DMAC による割り込み

μ DMAC コントローラは、EI²OS と同等の機能を持った簡易 DMA です。DMA 転送の設定は、DMA ディスクリプタを用いて行います。

■ μ DMAC の機能

μ DMAC は、以下の機能をもっています。

- 周辺リソース (I/O) とメモリとの間で自動データ転送を行います。
- DMA 起動中は、CPU のプログラム実行は停止します。
- DMA 転送チャンネルは 16 チャンネルです (チャンネルの番号の小さい方が DMA 転送の優先順位が高い)。
- 転送元、転送先アドレスのインクリメントあり、またはインクリメントなしを選択可能です。
- DMA 転送は、周辺リソース (I/O) の割り込み要因により起動します。
- DMA 転送の制御は、 μ DMAC イネーブルレジスタ、 μ DMAC ストップステータスレジスタ、 μ DMAC ステータスレジスタ、ディスクリプタ (RAM 上の 000100_H ~ 00017F_H に割り当てられている) で行います。
- リソースから DMA 転送を停止させる手段として STOP 要求があります。
- DMA 転送終了後、DMA ステータスレジスタの転送終了チャンネルに相当するビットにフラグがセットされ、割り込みコントローラへ終了割り込みが出力されます。

■ μ DMAC のレジスタ一覧

● μ DMAC イネーブルレジスタ (DER)

μ DMAC イネーブルレジスタ (DER) のビット構成を、下図に示します。

ビット	15	14	13	12	11	10	9	8	
0000AD _H	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	DERH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	初期値 00000000 _B
	7	6	5	4	3	2	1	0	
0000AC _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	DERL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	初期値 00000000 _B

μ DMAC イネーブルレジスタ (DER) の各ビットの機能は、下表のとおりです。

ENx ビット	機 能
0 [初期値]	リソースからの割り込み要求を割り込みコントローラへ出力します。 (リソースからの割り込み要求を DMA 起動要求としません。)
1	リソースからの割り込み要求を DMA 起動要求とします。 DMA 転送バイト数が "0" となったところで、"0" にクリアされます。

＜注意事項＞

スタンバイモード (スリープモード, ストップモード, 時計モード, タイムベースタイマモード) または CPU 間欠動作モード (メインクロック間欠動作モード, PLL クロック間欠モード, サブクロック間欠モード) に移行する場合, 必ず DMA イネーブルレジスタ (DER) に 0000_H を設定してからモードに移行してください。

● μDMAC ストップステータスレジスタ (DSSR)

μDMAC ストップステータスレジスタ (DSSR) のビット構成を, 下図に示します。

ビット	7	6	5	4	3	2	1	0	
0000A4 _H	STP7	STP6	STP5	STP4	STP3	STP2	STP1	STP0	DSSR
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	初期値 00000000 _B

μDMAC ストップステータスレジスタ (DSSR) の各ビットの機能は, 下表のとおりです。

STPx ビット	機能
0 [初期値]	DMA 転送中に STOP 要求を受け付けていません。
1	DMA 転送中に STOP 要求を受け付けて, DMA 動作を停止しました。 STOP 要求を受け付けるのは UART 受信 (ch.7) のみです。bit7 以外のビットは意味を持ちません。 ソフトウェアで "1" を書き込んでも意味を持ちません。

● μDMAC ステータスレジスタ (DSR)

μDMAC ステータスレジスタ (DSR) のビット構成を, 下図に示します。

ビット	15	14	13	12	11	10	9	8	
00009D _H	DE15	DE14	DE13	DE12	DE11	DE10	DE9	DE8	DSRH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	初期値 00000000 _B
ビット	7	6	5	4	3	2	1	0	
00009C _H	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0	DSRL
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	初期値 00000000 _B

μDMAC ステータスレジスタ (DSR) の各ビットの機能は, 下表のとおりです。

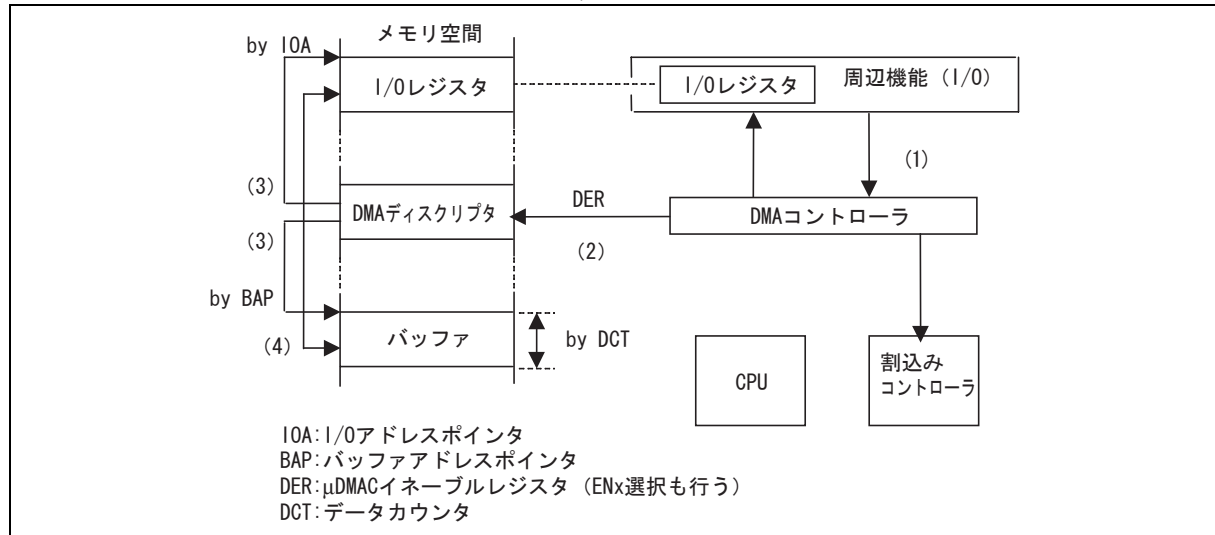
DEx ビット	機能
0 [初期値]	DMA 転送は終了していません。
1	DMA 転送が終了すると割り込み要求を割り込みコントローラへ出力します。

(注意事項) "1" を書き込んだ場合, DMA 転送は終了しません。割り込みコントローラへ割り込みを出力します。

■ μ DMAC の動作説明

図 3.6-1 に μ DMAC の動作説明図を示します。DMA を使用したデータ転送は、以下の順番で行われます。

図 3.6-1 μ DMAC の動作



- (1) 周辺リソース (I/O) が DMA 転送を要求する。
- (2) DMA コントローラがディスクリプタを読み出す。
- (3) 転送元、転送先や転送データ数をディスクリプタから読み出す。
- (4) I/O とメモリ間で DMA データ転送を開始する。
- (5) 転送終了でないとき：リソースの割り込み要求をクリアします。

転送終了時 : DMA 転送終了後、 μ DMAC ステータスレジスタに転送終了のフラグがセットされ、割り込み要求を割り込みコントローラへ出力します。

3.6.1 DMA ディスクリプタ

DMA ディスクリプタは、内部 RAM の "000100_H" ~ "00017F_H" に存在し、8 バイト×16 チャンネルで構成されています。

■ DMA ディスクリプタの構成

DMA ディスクリプタは、8 バイト× 16 チャンネルで構成されており、各 DMA ディスクリプタは図 3.6-2 に示す構造になっています。チャンネル番号と DMA ディスクリプタのアドレスの対応は、表 3.6-1 に示すとおりです。

図 3.6-2 DMA ディスクリプタの構成

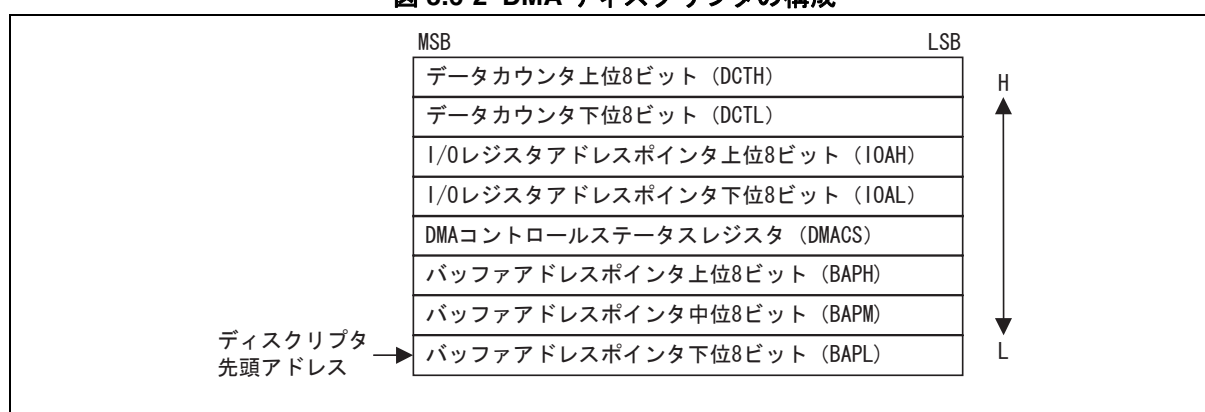


表 3.6-1 チャネル番号とディスクリプタアドレスの対応

μDMAC イネーブルレジスタ	チャンネル	ディスクリプタアドレス	リソース割込み要求
EN0	0	000100 _H	INT0
EN1	1	000108 _H	PWC0 (MB90485 シリーズのみ)
EN2	2	000110 _H	PPG0/PPG1 カウンタボロー
EN3	3	000118 _H	PPG2/PPG3 カウンタボロー
EN4	4	000120 _H	PPG4/PPG5 カウンタボロー
EN5	5	000128 _H	インプットキャプチャ (ch.0) 取込み
EN6	6	000130 _H	インプットキャプチャ (ch.1) 取込み
EN7	7	000138 _H	UART 受信完了
EN8	8	000140 _H	アウトプットコンペア (ch.0) 一致
EN9	9	000148 _H	アウトプットコンペア (ch.1) 一致
EN10	10	000150 _H	アウトプットコンペア (ch.2) 一致
EN11	11	000158 _H	UART 送信完了
EN12	12	000160 _H	16 ビット FRT/16 ビットリロードタイマオーバフロー
EN13	13	000168 _H	SIO1
EN14	14	000170 _H	SIO2
EN15	15	000178 _H	A/D コンバータ

3.6.2 DMA ディスクリプタの各レジスタ

DMA ディスクリプタの各レジスタは、次のレジスタから構成されています。

- データカウンタ (DCT)
- I/O レジスタアドレスポインタ (IOA)
- DMA コントロールステータスレジスタ (DMACS)
- バッファアドレスポインタ (BAP)

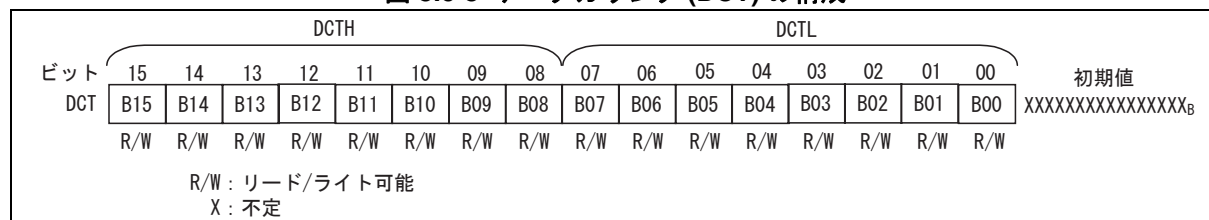
各レジスタの値は、リセット時には不定となるので、必ず初期化してください。

■ データカウンタ (DCT)

データカウンタ (DCT) は、16 ビット長のレジスタで、転送データ数に対応したカウンタとなります。各データの転送後に、カウンタは1デクリメント (減値) します。このカウンタが "0" になると、DMA は終了します。図 3.6-3 に、データカウンタ (DCT) の構成を示します。

データカウンタ (DCT) に "0" を設定すると、最大データ転送数 (65536 個) がセットされます。

図 3.6-3 データカウンタ (DCT) の構成

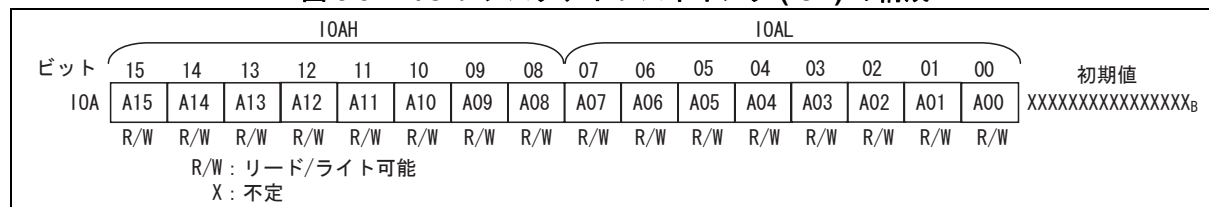


■ I/O レジスタアドレスポインタ (IOA)

I/O レジスタアドレスポインタ (IOA) は、16 ビット長のレジスタで、バッファとのデータ転送を行う I/O レジスタの下位アドレス (A15 ~ A0) を示します。上位アドレス (A23 ~ A16) はすべて "0" で、"000000_H" から "00FFFF_H" 番地までの任意の I/O をアドレスで指定できます。

図 3.6-4 に、I/O レジスタアドレスポインタ (IOA) の構成を示します。

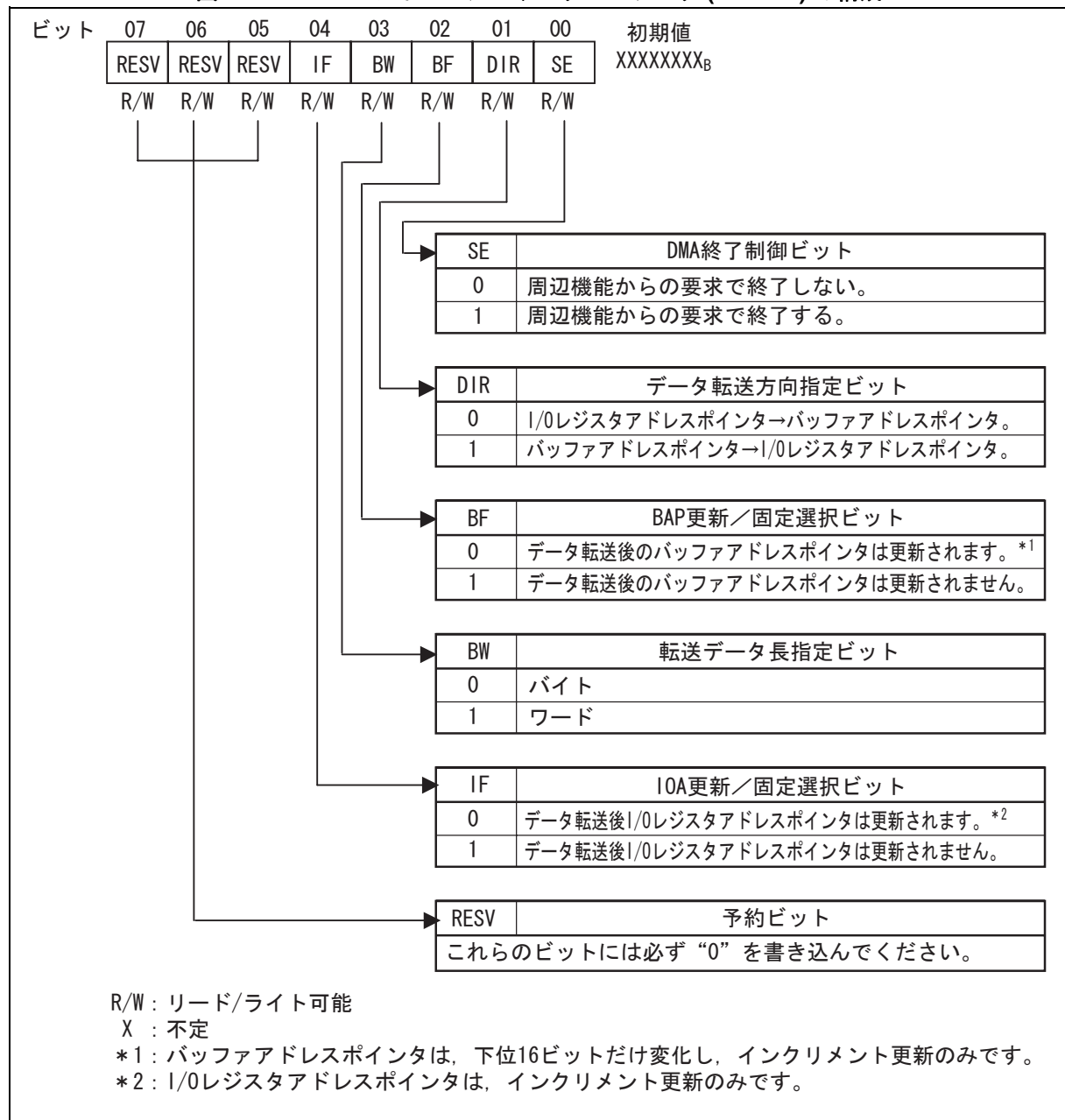
図 3.6-4 I/O レジスタアドレスポインタ (IOA) の構成



■ DMA コントロールステータスレジスタ (DMACS)

DMA コントロールステータスレジスタ (DMACS) は、8 ビット長で、バッファアドレスポインタ (BAP) と I/O レジスタアドレスポインタ (IOA) の更新/固定、転送データ形式 (バイト/ワード) および転送方向を示します。図 3.6-5 に、DMA コントロールステータスレジスタ (DMACS) の構成を示します。

図 3.6-5 DMA コントロールステータスレジスタ (DMACS) の構成



■ バッファアドレスポインタ (BAP)

バッファアドレスポインタ (BAP) は、24 ビットのレジスタで、DMA による次回の転送で使用するアドレスを保持します。バッファアドレスポインタ (BAP) は、DMA の各チャンネルに対してそれぞれ独立に存在するので、DMA の各チャンネルは、16M バイトの任意のアドレスと I/O の間でデータ転送できます。DMA コントロールステータスレジスタ (DMACS) の BF ビット (BAP 更新 / 固定選択ビット) で "更新あり" とした場合、バッファアドレスポインタ (BAP) は下位 16 ビット (BAPM, BAPL) だけ変化し、上位 8 ビット (BAPH) は変化しません。図 3.6-6 に、バッファアドレスポインタ (BAP) の BAP 構成を示します。

図 3.6-6 バッファアドレスポインタ (BAP) の構成

	23	16	15	8	7	0	初期値
BAP	BAPH			BAPM		BAPL	XXXXXX _H
	(R/W)			(R/W)		(R/W)	
R/W : リード/ライト可能							
X : 不定							

<注意事項>

- I/Oレジスタアドレスポインタ (IOA) で指定できる領域は、"000000_H" ~ "00FFFF_H" 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_H" ~ "FFFFFF_H" です。

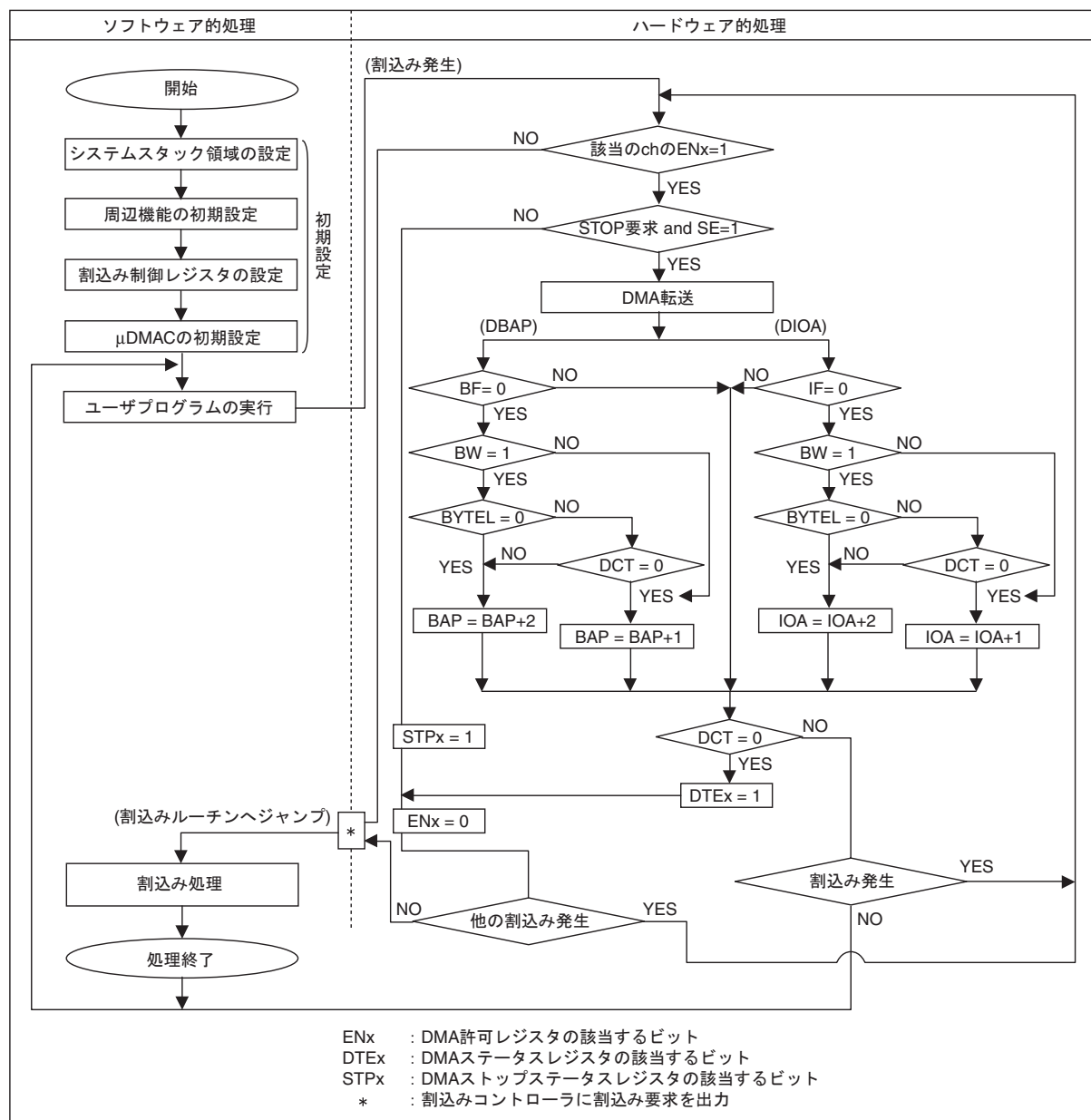
3.6.3 μ DMAC の処理手順

周辺リソース (I/O) から割り込み要求が発生し、対応する μ DMAC イネーブルレジスタ (DER) で DMA の起動が設定されていれば、DMA 転送を行います。指定回数のデータ転送が終了すれば、割り込み要求を割り込みコントローラへ出力します。

■ μ DMAC の処理手順

図 3.6-7 に簡単な μ DMAC の処理手順を示します。

図 3.6-7 μ DMAC の処理手順



3.6.4 μ DMAC の処理時間

μ DMAC の処理に必要な時間は、以下の要因によって変化します。

- DMA コントロールステータスレジスタ (DMACS) の設定
- I/O レジスタアドレスポインタ (IOA) の示すアドレス (領域)
- バッファアドレスポインタ (BAP) の示すアドレス (領域)
- 外部アクセス時の外部データバス幅
- 転送データのデータ長

また、 μ DMAC によるデータ転送終了時には、ハードウェア割込みが起動されるため、割込みハンドリング時間が加算されます。

■ μ DMAC の処理時間 (1 回の転送時間)

● データ転送継続時

データ転送継続時の μ DMAC 処理時間は、DMA コントロールステータスレジスタ (DMACS) の設定によって、表 3.6-2 のようになります。

表 3.6-2 μ DMAC の実行時間

IOA 更新 / 固定選択ビット (IF) の設定		固定	更新
BAP アドレス更新 / 固定 選択ビット (BF) の設定	固定	17	19
	更新	19	21

(注意事項) 単位はマシンサイクルです。1 マシンサイクルは、マシナクロック (ϕ) の 1 クロック周期に相当します。

さらに、表 3.6-3 に示すように、 μ DMAC 実行時の条件によって補正が必要です。

表 3.6-3 μ DMAC 実行時間のデータ転送の補正值

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	奇	B/ 偶	8/ 奇
バッファアドレス ポインタ	内部アクセス	B/ 偶	0	+2	+1	+4
		奇	+2	+4	+3	+6
	外部アクセス	B/ 偶	+1	+3	+2	+5
		8/ 奇	+4	+6	+5	+8

(注意事項) B はバイトデータ転送、8 は外部バス幅 8 ビットワード転送、偶は偶数アドレスワード転送、奇は奇数アドレスワード転送を示します。

● 転送パフォーマンス

- 最小転送速度 1.7 μ s/10MHz(マシナクロック) 時
1.07 μ s/16MHz(マシナクロック) 時
 - 内蔵 I/O → 内蔵 RAM または 内蔵 RAM → 内蔵 I/O アドレスのインクリメントなし
 - 偶数番地 → 偶数番地 または 8 ビットアクセス
- 最大転送速度 2.8 μ s/10MHz(マシナクロック) 時
1.75 μ s/16MHz(マシナクロック) 時

表 3.6-4 に、割込みハンドリング時間の補正值を示します。

表 3.6-4 割込みハンドリング時間の補正值 (Z)

スタックポインタが指しているアドレス	補正值 (Z)
外部 8 ビットの場合	+4
外部 偶数アドレスの場合	+1
外部 奇数アドレスの場合	+4
内部 偶数アドレスの場合	0
内部 奇数アドレスの場合	+2

● 周辺機能 (I/O) からの終了要求による終了時

周辺機能 (I/O) からの終了要求で、 μ DMAC によるデータ転送を途中で終了した場合 (DEx=1) は、データ転送は行わずにハードウェア割込みを起動します。この場合の μ DMAC 処理時間は、次の式で算出されます。式中の Z は、割込みハンドリング時間の補正值 (表 3.6-4 を参照) です。

途中終了時の μ DMAC 処理時間 = $36 + 6 \times Z$ マシンサイクル

1 マシンサイクルは、マシナクロック (ϕ) の 1 クロック周期に相当します。

3.7 拡張インテリジェント I/O サービス (EI²OS) による割込み

拡張インテリジェント I/O サービス (EI²OS) は、周辺機能 (リソース) とメモリの間でデータ転送を行う機能です。データ転送が終了した場合、ハードウェア割込み処理を行います。

■ 拡張インテリジェント I/O サービス (EI²OS)

拡張インテリジェント I/O サービスは、ハードウェア割込みの一種です。周辺機能 (リソース) とメモリの間でデータ転送を行う機能です。ユーザは、EI²OS を起動する場合と終了する場合にプログラムを作成し、データ転送プログラムは不要です。

● 拡張インテリジェント I/O サービス (EI²OS) の利点

割込み処理ルーチンで実行するデータ転送と比べた場合、次のような利点があります。

- 転送プログラムを記述する必要がないので、プログラムサイズを小さくできる。
- 周辺機能 (リソース) の割込み要因によって転送を起動できるので、データの転送要因をポーリングする必要がない。
- 転送アドレスのインクリメントを設定できる。
- I/O レジスタアドレスのインクリメントおよび更新なしを設定できる。

● 拡張インテリジェント I/O サービス (EI²OS) の終了割込み

EI²OS による、データ転送が終了した場合、割込み処理ルーチンへ分岐します。

割込み処理プログラムで、割込み制御レジスタ (ICR) の EI²OS ステータスビット (S1, S0) をチェックすることにより、EI²OS の終了要因を判断することができます。

<参考>

割込み番号や割込みベクタは、各周辺で固定されています。詳細は「3.2 割込み要因と割込みベクタ」を参照してください。

● 割込み制御レジスタ (ICR)

EI²OS の起動、EI²OS のチャンネルを設定することができます。また、EI²OS 終了時の EI²OS ステータスを判断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 (EI²OS) ディスクリプタ (ISD)

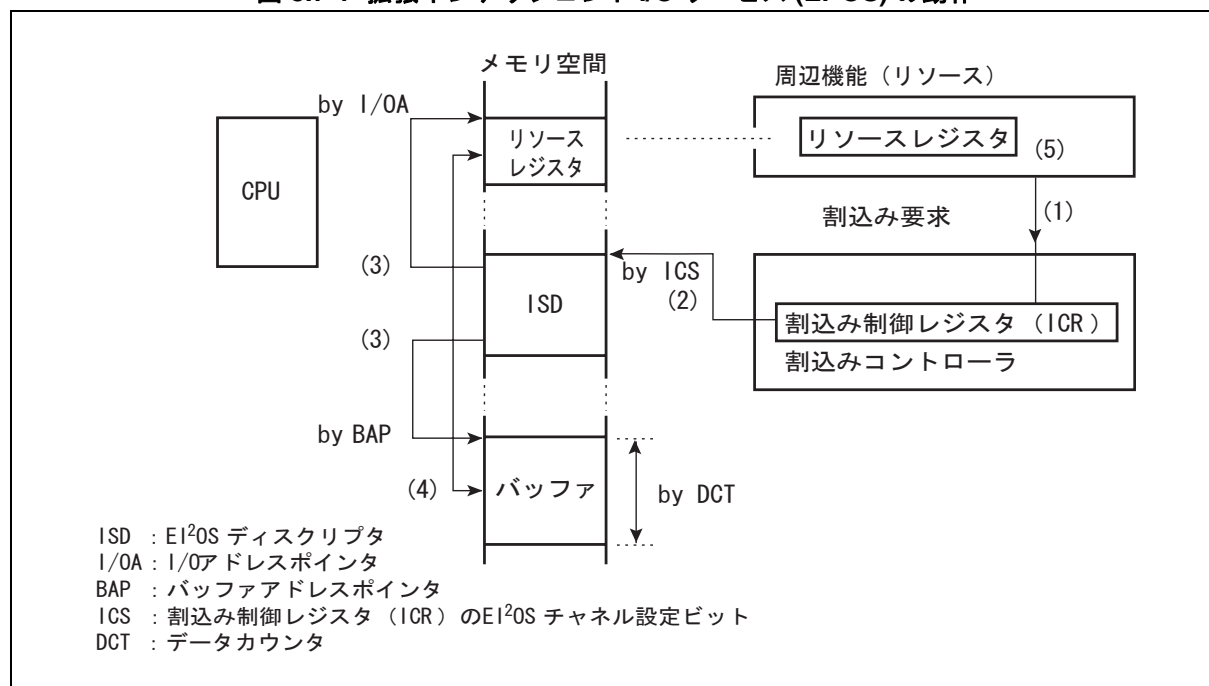
EI²OS ディスクリプタは、RAM 上の "000100_H" ~ "00017F_H" に配置されており、転送モード、周辺機能 (リソース) のアドレスや転送バイト数、転送先アドレスを設定する 8 バイト × 16 チャンネル分のレジスタです。割込み制御レジスタ (ICR) でチャンネル設定を行います。

<注意事項>

拡張インテリジェント I/O サービス (EI²OS) が動作している場合、CPU のプログラムは実行されません。

■ 拡張インテリジェント I/O サービス (EI²OS) の動作

図 3.7-1 拡張インテリジェント I/O サービス (EI²OS) の動作



- (1) 周辺機能 (リソース) から, 割り込み要求を出力します。
- (2) 割り込みコントローラが EI²OS ディスクリプタを割り込み制御レジスタ (ICR) の設定に従って設定します。
- (3) 転送元や転送先が EI²OS ディスクリプタから読み出されます。
- (4) 周辺機能 (リソース) とメモリ間でデータの転送が行われます。
- (5) データ転送終了後に周辺機能 (リソース) の割り込み要求フラグビットは "0" にクリアされます。

3.7.1 拡張インテリジェント I/O サービス (EI²OS) ディスクリプタ (ISD)

拡張インテリジェント I/O サービス (EI²OS) ディスクリプタ (ISD) は、RAM の "000100_H" ~ "00017F_H" に存在し、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 (EI²OS) 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。

図 3.7-2 EI²OS ディスクリプタ (ISD) の構成

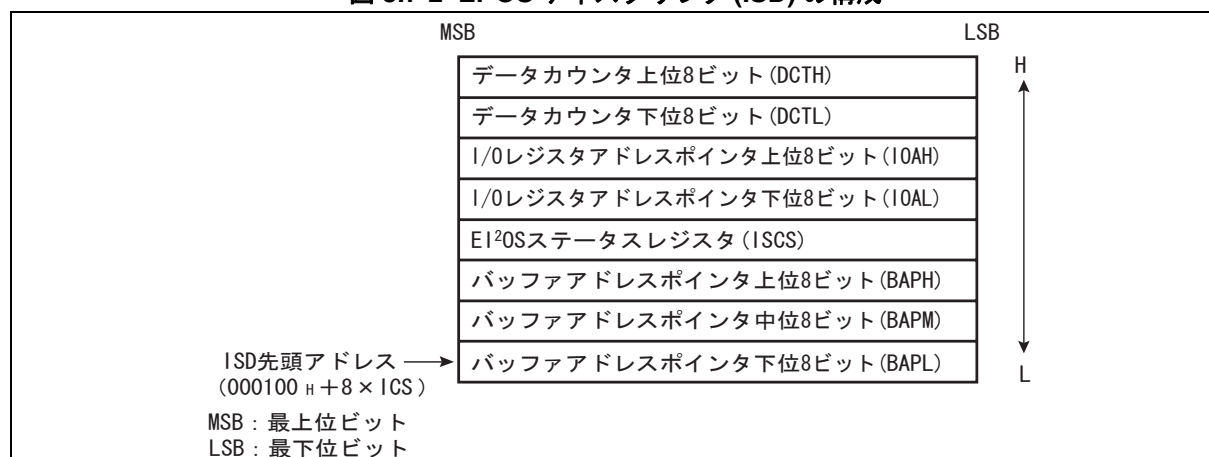


表 3.7-1 チャンネル番号とディスクリプタアドレスの対応

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

*:ISD のアドレスは、8 バイトのうちの先頭アドレスを示しています。

3.7.2 拡張インテリジェント I/O サービス (EI²OS) ディスクリプタ (ISD) の説明

拡張インテリジェント I/O サービス (EI²OS) ディスクリプタ (ISD) は、以下に示す 4 種類の 8 バイトのレジスタで構成されています。

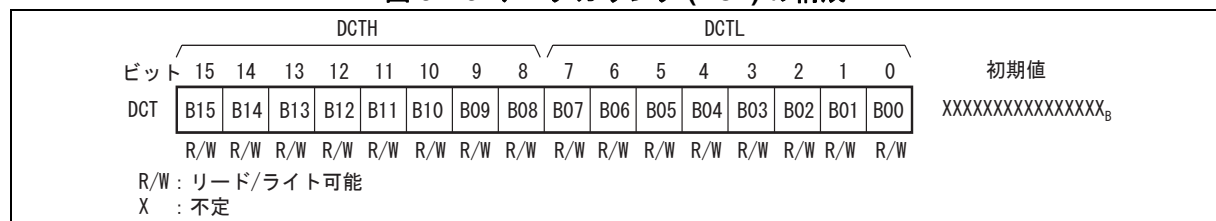
- ・ データカウンタ (DCT : 2 バイト)
- ・ I/O レジスタアドレスポインタ (IOA : 2 バイト)
- ・ EI²OS ステータスレジスタ (ISCS : 1 バイト)
- ・ バッファアドレスポインタ (BAP : 3 バイト)

各レジスタの初期値は、不定です。

■ データカウンタ (DCT)

データカウンタ (DCT) は、16 ビット長のレジスタです。転送データバイト数を設定します。データを 1 バイト転送するごとに、カウンタ値は -1 (デクリメント) されます。データカウンタ値が "0000_H" になった場合、EI²OS が終了します。

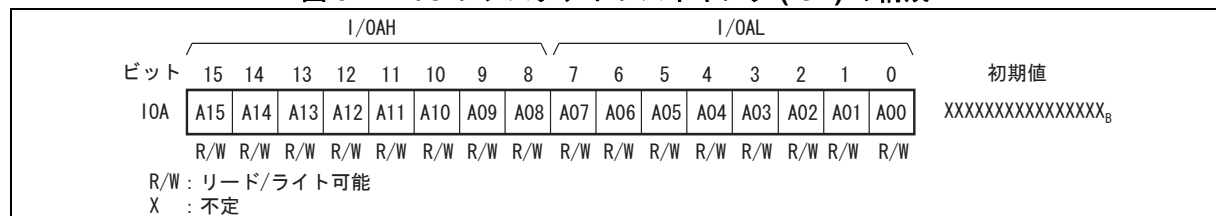
図 3.7-3 データカウンタ (DCT) の構成



■ I/O レジスタアドレスポインタ (IOA)

I/O レジスタアドレスポインタ (IOA) は、16 ビット長のレジスタです。データ転送を行うための I/O レジスタ下位アドレス (A15 ~ A0) を設定します。上位アドレス (A23 ~ A16) は "00_H" であり、"0000_H" から "FFFF_H" 番地までの I/O をアドレスで設定できます。

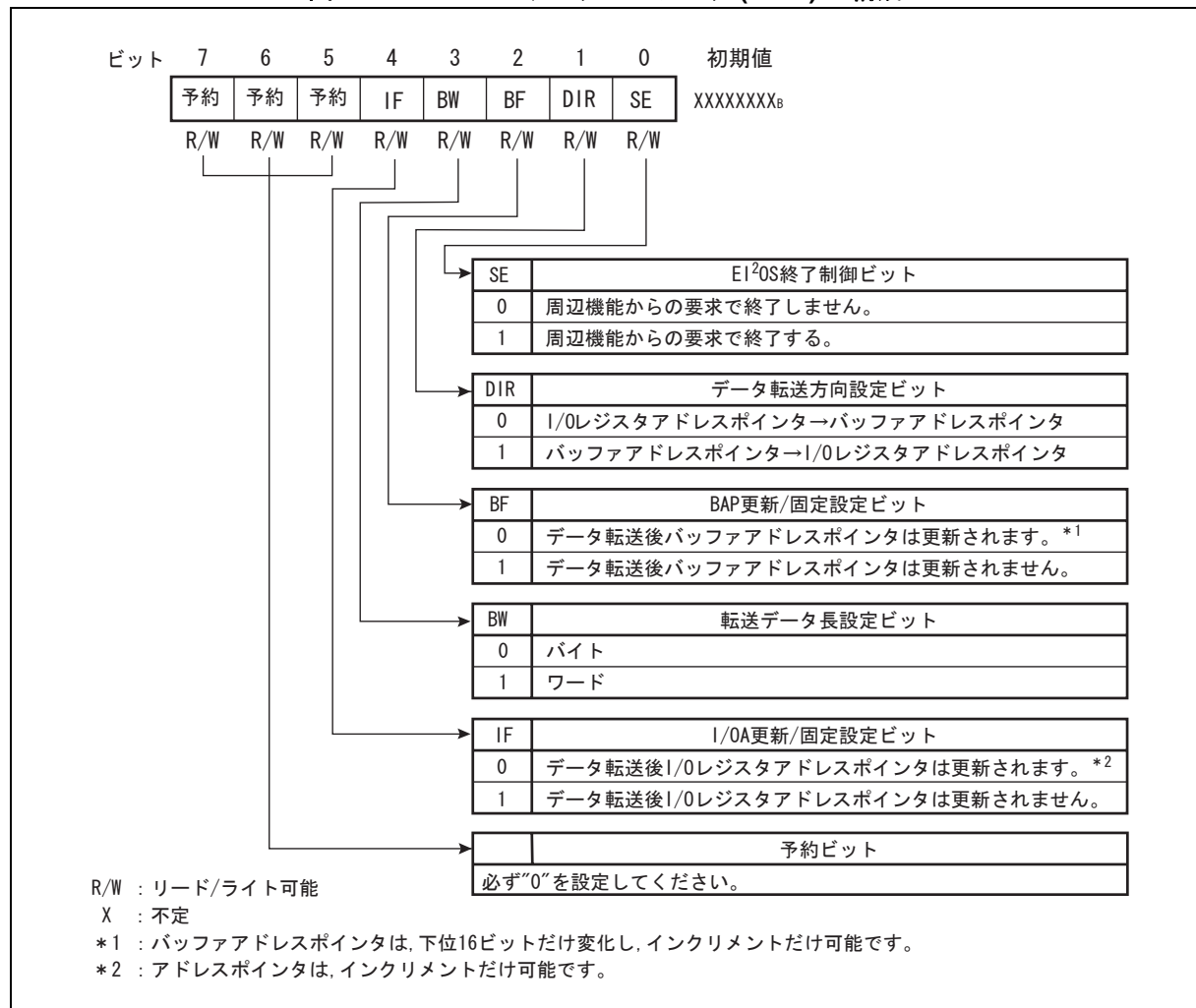
図 3.7-4 I/O レジスタアドレスポインタ (IOA) の構成



■ 拡張インテリジェント I/O サービス (EI²OS) ステータスレジスタ (ISCS)

拡張インテリジェント I/O サービス (EI²OS) ステータスレジスタ (ISCS) は、8 ビット長で、バッファアドレスポインタと I/O レジスタアドレスポインタの更新/固定、転送データ形式 (バイト/ワード) および転送方向を設定します。

図 3.7-5 EI²OS ステータスレジスタ (ISCS) の構成



■ バッファアドレスポインタ (BAP)

バッファアドレスポインタ (BAP) は、24 ビットのレジスタです。EI²OS 動作で、データ転送元のメモリアドレスを設定します。バッファアドレスポインタ (BAP) は、EI²OS の各チャンネルに存在しますので、16M バイトのメモリアドレスと周辺機能 (リソース) アドレスとの間でデータを転送できます。EI²OS ステータスレジスタ (ISCS) の BAP 更新 / 固定設定ビット (BF) に "0" を設定した場合は、下位 16 ビット (BAPM, BAPL) がインクリメントされ、上位 8 ビット (BAPH) はインクリメントされません。

図 3.7-6 バッファアドレスポインタ (BAP) の構成

	bit23	~	bit16	bit15	~	bit8	bit7	~	bit0	初期値
BAP	BAPH			BAPM			BAPL			XXXXXX _H
	(R/W)			(R/W)			(R/W)			
R/W : リード/ライト可能										
X : 不定										

BAPM

(R/W)

BAPL

(R/W)

<参考>

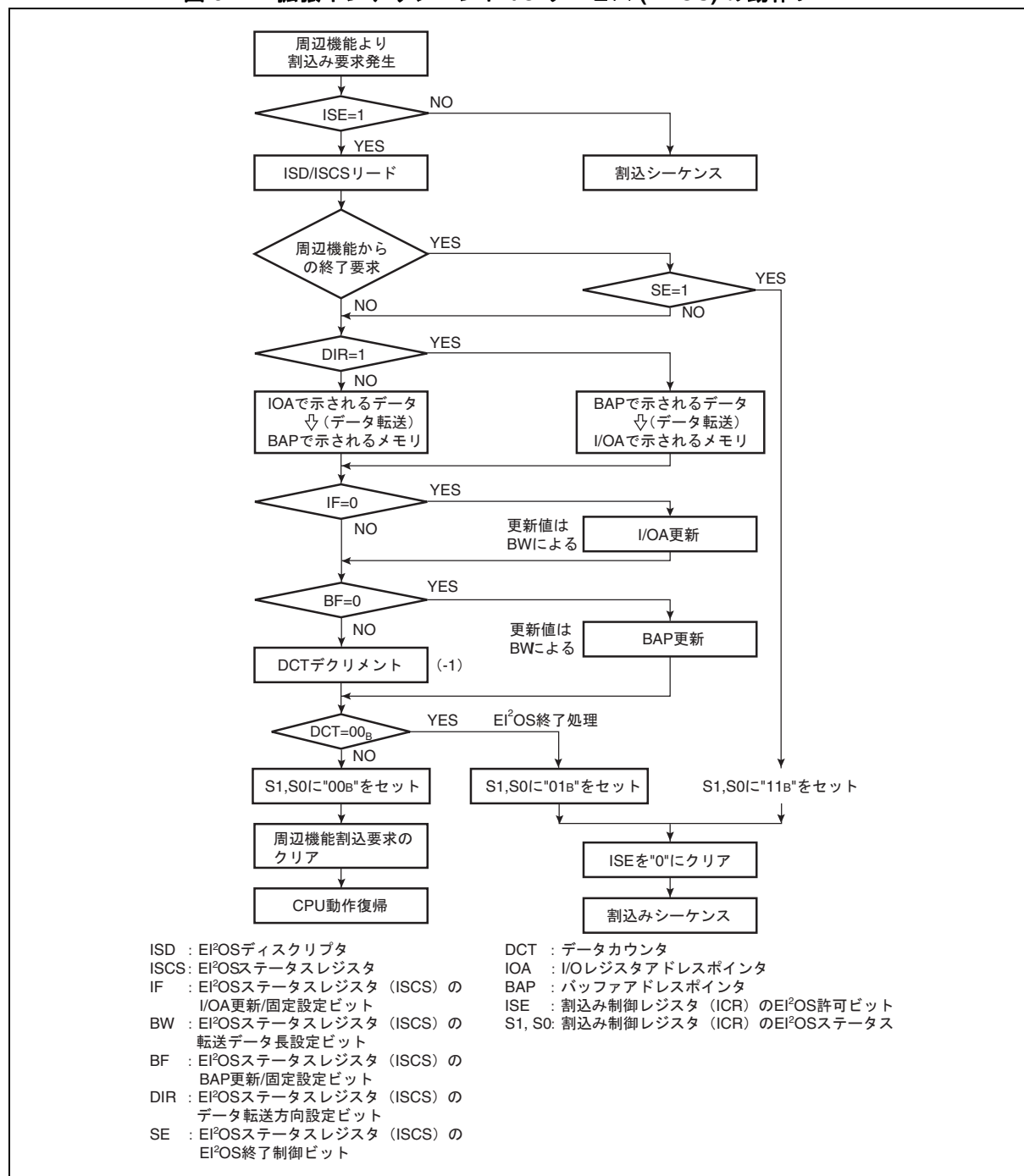
- データカウンタ (DCT) で設定できる最大転送回数は、65,536 回 (64K バイト) です。
- I/O アドレスポインタ (IOA) で設定できる領域は、"000000_H" ~ "00FFFF_H" です。
- バッファアドレスポインタ (BAP) で設定できる領域は、"000000_H" ~ "FFFFFF_H" です。

3.7.3 拡張インテリジェント I/O サービス (EI²OS) の動作

周辺機能 (リソース) から割り込み要求を出力し , 割り込み制御レジスタ (ICR) に EI²OS の起動を事前に設定している場合 , CPU は EI²OS によるデータ転送を行います。EI²OS 処理が終了した場合 , ハードウェア割り込み処理を行います。

■ 拡張インテリジェント I/O サービス (EI²OS) の処理手順

図 3.7-7 拡張インテリジェント I/O サービス (EI²OS) の動作フロー

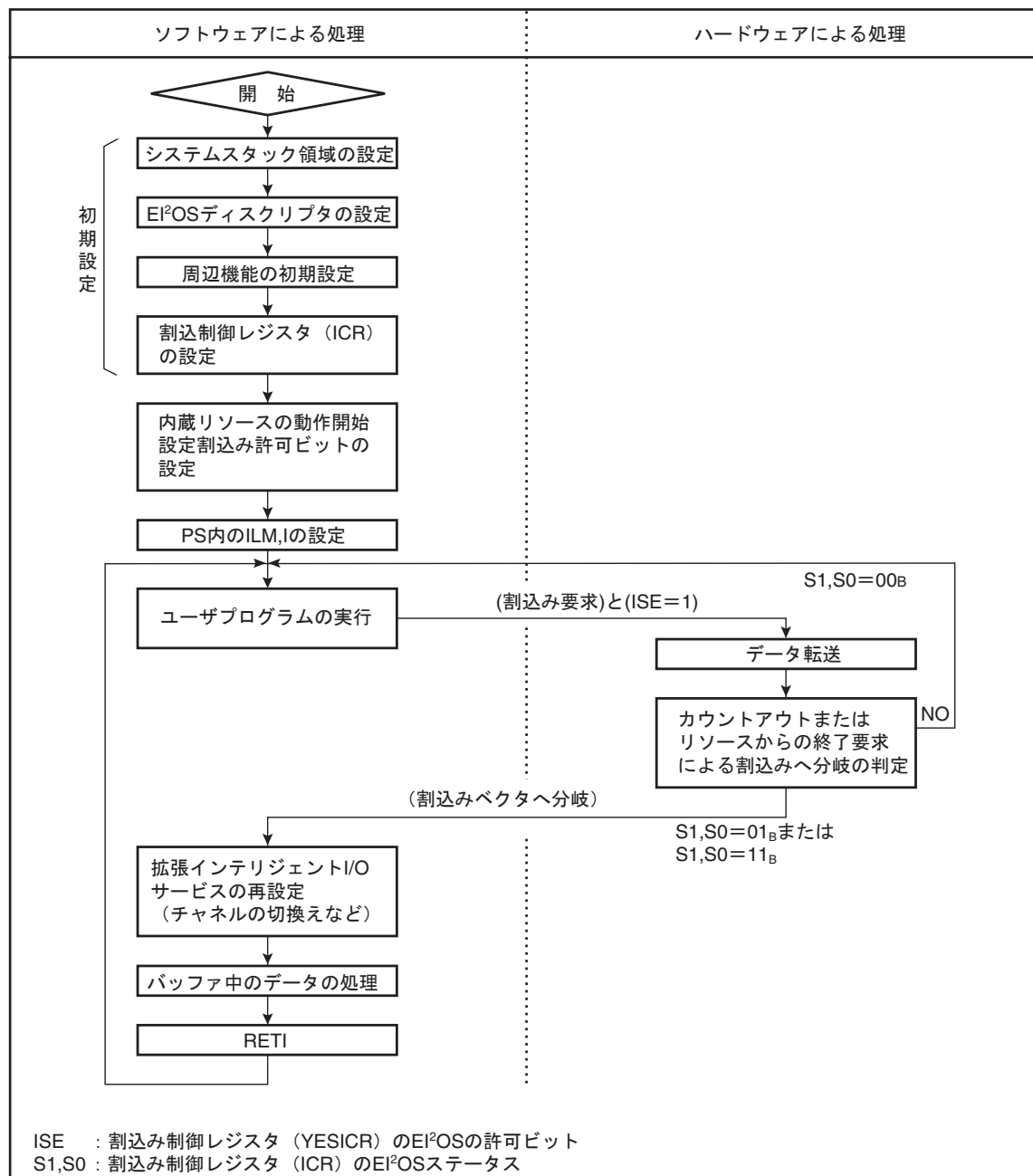


3.7.4 拡張インテリジェント I/O サービス (EI²OS) の設定手順

拡張インテリジェント I/O サービス (EI²OS) の設定は、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 (EI²OS) ディスクリプタ、周辺機能 (リソース) および割り込み制御レジスタ (ICR) で設定します。

■ 拡張インテリジェント I/O サービス (EI²OS) の設定手順

図 3.7-8 拡張インテリジェント I/O サービス (EI²OS) 設定手順



3.7.5 拡張インテリジェント I/O サービス (EI²OS) 処理時間

拡張インテリジェント I/O サービス (EI²OS) の処理に必要な時間は拡張インテリジェント I/O サービスディスクリプタ (ISD) の設定により異なります。

- EI²OS 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) の示すアドレス設定
- バッファアドレスポインタ (BAP) の示すアドレス設定
- 外部アクセス時の外部データバス幅
- 転送データのデータ長

EI²OS によるデータ転送が終了した場合は、ハードウェア割込みが起動されるため、割込みハンドリング時間が加算されます。

■ 拡張インテリジェント I/O サービス (EI²OS) 処理時間 (1 回の転送時間)

● 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 EI²OS 処理時間は、EI²OS ステータスレジスタ (ISCS) の設定によって、表 3.7-2 のようになります。

表 3.7-2 拡張インテリジェント I/O サービス実行時間

EI ² OS 終了制御ビット (SE) の設定		周辺からの終了要求により終了		周辺からの終了要求を無視	
I/OA 更新 / 固定設定ビット (IF) の設定		固定	更新	固定	更新
BAP アドレス更新 / 固定設定ビット (BF) 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 (φ) の 1 クロック周期に相当します。)

データ転送を継続する場合の EI²OS 処理時間は、表 3.7-3 に示すように、EI²OS 実行条件によって補正が必要です。

表 3.7-3 EI²OS 実行時間のデータ転送の補正值

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	奇	B/ 偶	8/ 奇
バッファアドレスポインタ	内部アクセス	B/ 偶	0	+2	+1	+4
		奇	+2	+4	+3	+6
	外部アクセス	B/ 偶	+1	+3	+2	+5
		8/ 奇	+4	+6	+5	+8

B：バイトデータ転送

8：外部バス幅 8 ビット・ワード転送

偶：偶数アドレス・ワード転送

奇：奇数アドレス・ワード転送

● データカウンタ (DCT) のカウント終了時 (最終回のデータ転送時)

EI²OS によるデータ転送が終了した場合、ハードウェア割込みが起動されるため、割込みハンドリング時間が加算されます。カウント終了時の EI²OS の処理時間は、下式で算出されます。式中の Z は、割込みハンドリング時間の補正值です。

$$\text{カウント終了時のEI}^2\text{OS処理時間} = \text{データ転送時のEI}^2\text{OS処理時間} + \underbrace{(21+6 \times Z)}_{\substack{\uparrow \\ \text{割込みハンドリング時間}}} \text{マシンサイクル}$$

割込みハンドリング時間は、スタックポインタが格納しているアドレスによって異なります。

表 3.7-4 割込みハンドリング時間の補正值 (Z)

スタックポインタが指しているアドレス	補正值 (Z)
外部割込みが 8 ビットの場合	+4
外部割込みが偶数アドレスの場合	+1
外部割込みが奇数アドレスの場合	+4
内部割込みが偶数アドレスの場合	0
内部割込みが奇数アドレスの場合	+2

● 周辺機能 (リソース) からの終了要求により終了した場合

周辺機能 (リソース) からの終了要求では、EI²OS によるデータ転送を途中で終了した場合 (ICR : S1, S0=11_B) は、データ転送は行わず、ハードウェア割込みを起動します。EI²OS 処理時間は、下式で算出されます。式中の Z は、割込みハンドリング時間の補正值です (表 3.7-4 を参照)。

$$\text{途中終了した場合のEI}^2\text{OS処理時間} = 36+6 \times Z \text{ マシンサイクル}$$

<参考>

1 マシンサイクルは、マシクロック (φ) の 1 クロック周期に相当します。

3.8 例外処理割込み

F²MC-16LX では、未定義命令が実行された場合、例外処理が行われます。例外処理は、基本的には割込みと同じものであり、命令間で例外事項の発生が検出された場合に、通常処理を中断して行われます。例外処理は予想外の動作を行った結果によって発生する処理なので、デバッグ時や緊急時の復旧ソフトウェアの起動だけに使用してください。

■ 未定義命令の実行による例外処理

● 例外処理の動作

F²MC-16LX では、命令マップで定義していないコードはすべて未定義命令として扱います。未定義命令を実行すると、ソフトウェア割込み命令の "INT#10" と等価な処理を行います。例外処理では、割込みルーチンへ分岐する前に、次の処理が行われます。

- システムスタックへ A, DPR, ADB, DTB, PCB, PC, PS レジスタを退避します。
- コンディションコードレジスタ (CCR) の I フラグを "0" にクリアし、ハードウェア割込みを抑止します。
- コンディションコードレジスタ (CCR) の S フラグを "1" にセットし、システムスタックを有効にします。

スタックへ退避したプログラムカウンタ (PC) の値は、未定義命令が格納されているアドレスです。2 バイト以上の命令コードでは、未定義であることを識別できたコードが格納されているアドレスになります。例外処理ルーチン内で、例外要因の種類を判定する必要がある場合は、退避された PC 値を使用してください。

● 例外処理からの復帰

例外処理から RETI 命令で復帰させると、PC が未定義命令を指しているために、再度、例外処理に入ります。ソフトウェアリセットなどの対策をしてください。

3.9 割り込み処理のスタック動作

割り込みが受け付けられると、割り込み処理に分岐する前に、専用レジスタの内容が自動的にシステムスタックに退避されます。割り込み処理終了時のスタックからの復元も自動的行われます。

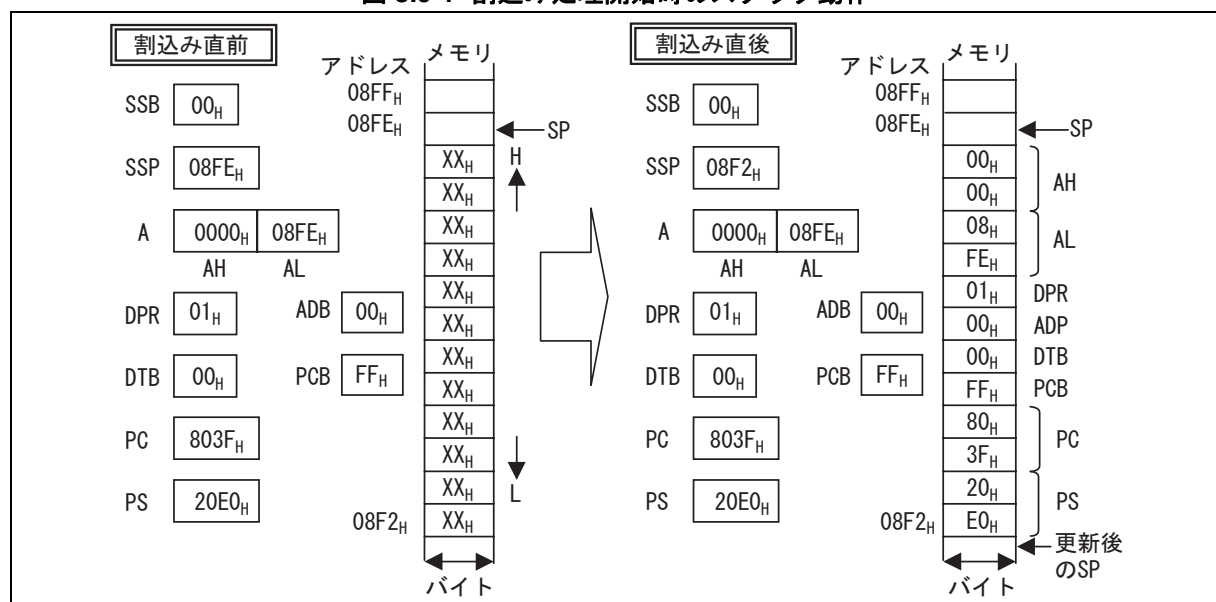
■ 割り込み処理開始時のスタック動作

割り込みが受け付けられると、CPU は現在の専用レジスタ類の内容を、次に示す順番で、自動的にシステムスタックに退避させます。

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

図 3.9-1 に、割り込み処理開始時のスタック動作を示します。

図 3.9-1 割り込み処理開始時のスタック動作



■ 割り込み処理復帰のスタック動作

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、割り込み処理開始時と逆の順番で PS, PC, PCB, DTB, ADB, DPR, A の値がスタックから復帰し、専用レジスタは割り込み開始直前の状態に戻ります。

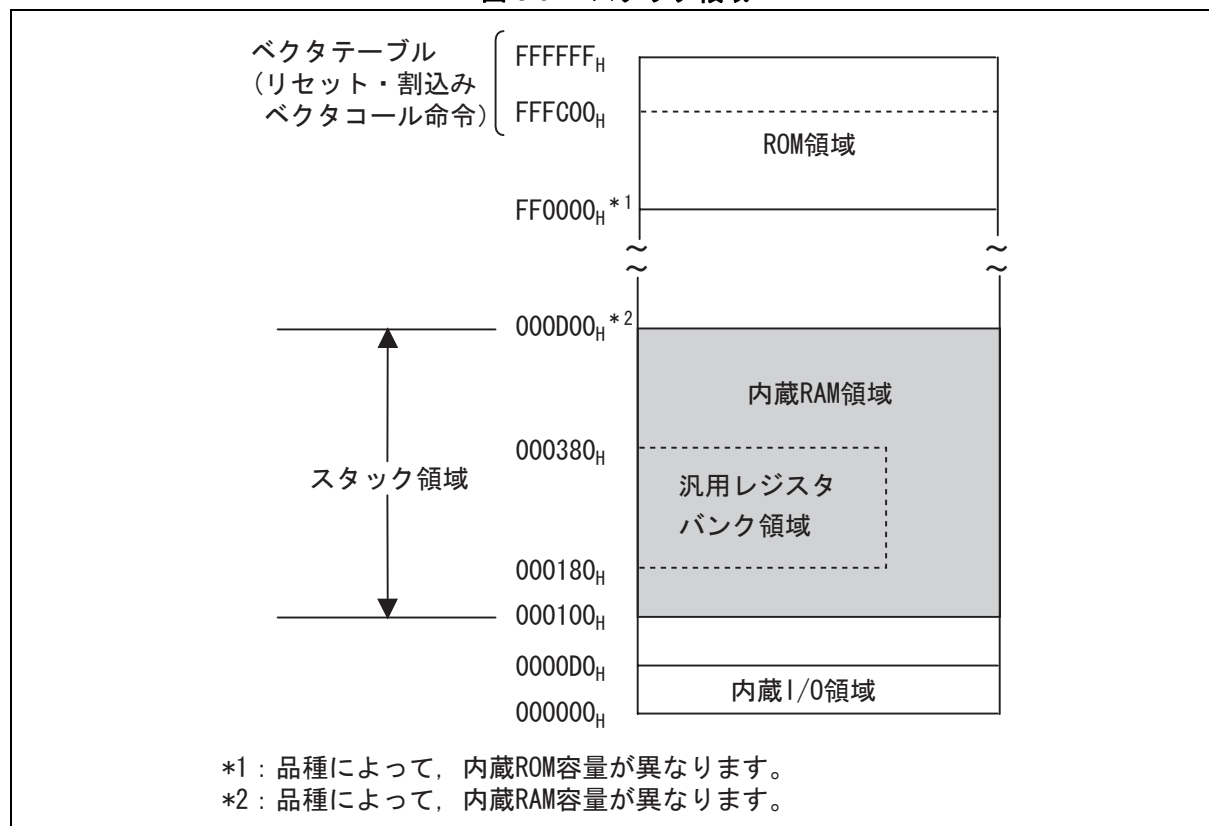
■ スタック領域

● スタック領域の確保

スタック領域は、割り込み処理のほかにサブルーチンコール命令 (CALL) やベクタコール

命令 (CALLV) を実行するときのプログラムカウンタ (PC) の退避 / 復帰や, PUSHW, POPW 命令による一時的なレジスタ類の退避 / 復帰にも使われます。スタック領域は、データ領域とともに RAM 上に確保します。図 3.9-2 にスタック領域を示します。

図 3.9-2 スタック領域



<注意事項>

- スタックポインタ (SSP, USP) にアドレスを設定する場合、原則として偶数アドレスを設定してください。
- システムスタック領域、ユーザスタック領域およびデータ領域は、お互いに重なり合わないよう配置してください。

● システムスタックとユーザスタック

割込み処理では、システムスタック領域を使用します。割込み発生時にユーザスタック領域を使用していても、強制的にシステムスタックに切り換えられます。このため、ユーザスタック領域を主に使用するシステムであっても、システムスタック領域を正しく設定しておく必要があります。特に、スタック空間を分ける必要がなければシステムスタックだけを使用してください。

3.10 割込み処理のプログラム例

割込み処理のプログラム例を示します。

■ 割込み処理のプログラム例

● 処理仕様

外部割込み 0(INT0) を利用した割込みプログラムの一例です。

以下にプログラムのコーディング例を示します。

[コーディング例]

```

DDR1 EQU 000011H ; ポート 1 方向レジスタ
ENIR EQU 028H ; 割込み /DTP 許可レジスタ
EIRR EQU 029H ; 割込み /DTP フラグ
ELVR EQU 02AH ; 要求レベル設定レジスタ
ICR00 EQU 0B0H ; 割込み制御レジスタ 00
STACK SSEG ; スタック
      RW 100
STACK_TRW 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 内 ILM をレベル 7 に設定
      MOV A,#!STACK_T ; システムスタックの設定
      MOV SSB,A
      MOVW A,#STACK_T ; スタックポインタの設定, このときは
      MOVW SP,A ; S フラグ=1 のため SSP にセットされる
      MOV DDR8,#00000000B ; P80/IRQ0 端子を入力に設定
      OR CCR,#40H ; PS 内 CCR の I フラグをセットして割込み許可
      MOV I:ICR00,#00H ; 割込みレベル 0(最高) とする
      MOV I:ELVR,#00000001B ; INT0 を "H" レベル要求とする
      MOV I:EIRR,#00H ; INT0 の割込み要因をクリア
      MOV I:ENIR,#01H ; INT0 の入力許可
      :
LOOP: NOP ; ダミーループ
      NOP
      NOP
      NOP
      BRA LOOP ; 無条件ジャンプ
;----- 割込みプログラム -----
ED_INT1:
      MOV I:EIRR,#00H ; INT0 の新規受付け禁止
      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
      RETI ; 割込みから復帰
CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=OFFH
      ORG OFFDOH ; 割込み #11(OBH) にベクタを設定

```

第3章 割込み

```
                                DSL      ED_INT1
                                ORG      OFFDCH      ; 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
```

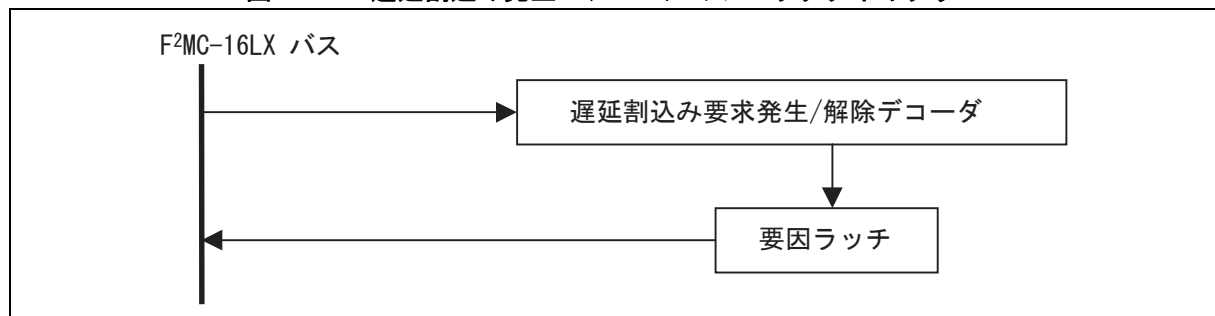
3.11 遅延割り込み発生モジュール

遅延割り込み発生モジュールは、タスク切換え用の割り込みを発生するためのモジュールです。本モジュールを使用することによってソフトウェアで F²MC-16LX CPU に対して割り込み要求の発生 / 取消しを行えます。

■ 遅延割り込み発生モジュールのブロックダイアグラム

図 3.11-1 に、遅延割り込み発生モジュールのブロックダイアグラムを示します。

図 3.11-1 遅延割り込み発生モジュールのブロックダイアグラム



■ 遅延割り込み発生モジュールのレジスタ一覧

遅延割り込み発生モジュール { 遅延割り込み要因発生/解除レジスタ (DIRR: Delayed Interrupt Request Register) } のレジスタ構成は下図のとおりです。

ビット	15	14	13	12	11	10	9	8	初期値
00009F _H	—	—	—	—	—	—	—	R0	-----0 _B
								R/W	
R/W : リード/ライト可能									

遅延割り込み要因発生 / 解除レジスタ (DIRR) は、遅延割り込み要因の発生 / 解除を制御するレジスタです。このレジスタに対して "1" を書き込んだ場合は遅延割り込みの要求を発生させ、"0" を書き込んだ場合は遅延割り込みの要求を解除します。リセット時には要因解除状態になります。予約ビット領域は、将来の拡張を考慮してアクセスする際には、セットビット、クリアビット命令を使用してください。

3.11.1 遅延割込み発生モジュールの動作

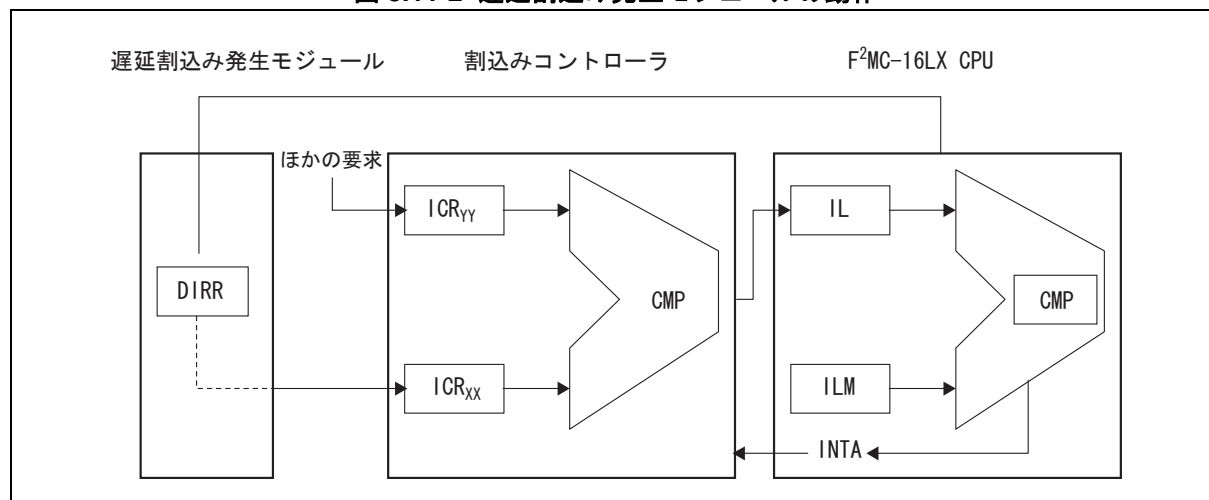
ソフトウェアにより CPU が DIRR の該当ビットに "1" を書き込むと、遅延割込み発生モジュールの中にある要求ラッチがセットされ、割込みコントローラに割込み要求が発生します。

■ 遅延割込み発生モジュールの動作

ソフトウェアにより CPU が DIRR の該当ビットに "1" を書き込むと、遅延割込み発生モジュールの中にある要求ラッチがセットされ、割込みコントローラに割込み要求が発生します。ほかの割込み要求が本割込みより優先順位が低い、またはほかの割込み要求がない場合に、割込みコントローラは F²MC-16LX CPU に対して割込みを発生します。CPU はプロセッサステータス (PS) 内のインタラプトマスクレジスタ (ILM) と割込み要求レベルを比較し、割込み要求レベルが ILM レジスタより高い場合には、現在実行中の命令が終了し次第、ハードウェア割込み処理マイクロプログラムを起動します。この結果、本割込みに対する割込みルーチンが実行されます。割込み処理ルーチン内で DIRR の該当ビットに "0" を書き込むことによって本割込み要因をクリアし、合わせてタスクの切換えを行います。

以上の動作の流れを、図 3.11-2 に示します。

図 3.11-2 遅延割込み発生モジュールの動作



■ 遅延割込み発生モジュールの使用上の注意 (遅延割込み要求ラッチ)

このラッチは、DIRR の該当するビットに "1" を書き込むことによってセットされ、同じビットに "0" を書き込むことによってクリアされます。したがって、割込み処理ルーチン内で、要因をクリアするようにソフトウェアを作成しておかないと、割込み要因処理から復帰した途端に再割込み処理が起動されることになりますので、注意してください。

第4章

リセット

MB90480/485 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すると，CPU は現在実行中の処理を直ちに中断し，リセット解除待ち状態になります。リセット解除後は，リセットベクタで示されたアドレスから処理を開始します。

リセットには，次の4種類の要因があります。

- パワーオンリセットの発生
- ウォッチドッグタイマのオーバフローの発生
- $\overline{\text{RST}}$ 端子からの外部リセット要求の発生
- ソフトウェアリセット要求の発生

■ リセット要因

リセットの要因について表 4.1-1 に示します。

表 4.1-1 リセット要因

リセット	発生要因	マシンのクロック	ウォッチドッグタイマ	発振安定待ち
パワーオン	電源投入時	メインクロック (MCLK)	停止	あり
ウォッチドッグタイマ	ウォッチドッグタイマオーバフロー	メインクロック (MCLK)	停止	なし
外部端子	$\overline{\text{RST}}$ 端子への "L" レベル入力	メインクロック (MCLK)	停止	なし
ソフトウェア	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" を書き込む	メインクロック (MCLK)	停止	なし

メインクロック：発振クロックの2分周クロック

● 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。評価用品 / フラッシュメモリ品の発振安定待ち時間は $2^{18}/\text{HCLK}$ (約 65.54ms: 発振クロック 4MHz 時)，マスク ROM 品の発振安定待ち時間は $2^{17}/\text{HCLK}$ (約 32.77ms: 発振クロック 4MHz 時) 固定です。ただし，MB90F488B/F489B の発振安定待ち時間は $(2^{18}+2^{15})/\text{HCLK}$ (約 73.73ms: 発振クロック 4MHz 時) になります。発振安定待ち時間が経過した後，リセット動作を行います。

● 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の起動後，定められ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" の書き込みがない場合，ウォッチドッグタイマのオーバフローによってリセットを発生させます。ウォッチドッグリセットでは，発振安定待ち時間の経過を待ちません。

● 外部リセット

外部リセットは、外部リセット端子 ($\overline{\text{RST}}$ 端子) に "L" レベルを入力することによってリセットを発生させます。 $\overline{\text{RST}}$ 端子への "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。

外部リセット ($\overline{\text{RST}}$ 端子入力リセット) では、発振安定待ち時間は発生しません。

$\overline{\text{RST}}$ 端子によるリセット要求が発生した場合に限り、書込み動作中 (転送系命令実行中 MOV 命令など) にリセット要因が発生した場合は、命令の終了後にリセット解除待ち状態になります。そのため、書込み中にリセットが入力されても、正常に書込みを終了します。

ただし、ストリング系命令 (MOVS 命令など) は指定したカウンタ分の転送が完了する前に、リセットを受け付けますので、すべてのデータが転送されることは保証されません。また、外部バスアクセス時に RDY 端子によるバスサイクルの延長が 16 マシンサイクル以上続いた場合もリセットを受け付けます。

● 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" を書き込むことによって、内部リセットを発生します。

ソフトウェアリセットでは、発振安定待ち時間は発生しません。

<参考>

クロックの定義

HCLK: 発振クロック (高速発振端子から供給されるクロック)

MCLK: メインクロック (HCLK の 2 分周クロック)

SCLK: サブクロック (低速発振端子から供給されるクロックの 4 分周)

ϕ : マシンクロック (CPU 動作クロック)

1/ ϕ : 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」を参照してください。

<注意事項>

ストップモード、またはサブクロックモードでリセットが発生した場合は、 $2^{17}/\text{HCLK}$ (約 32.77ms: 発振クロック 4MHz 時) の発振安定待ち時間をとります。

クロックモードの詳細は、「5.4 クロックモード」を参照してください。

4.2 リセット要因と発振安定待ち時間

MB90480/485 シリーズには、4 種類のリセット要因があります。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それぞれ異なります。

■ リセット要因と発振安定待ち時間

表 4.2-1 に、リセット要因と発振安定待ち時間について示します。

表 4.2-1 リセット要因と発振安定待ち時間

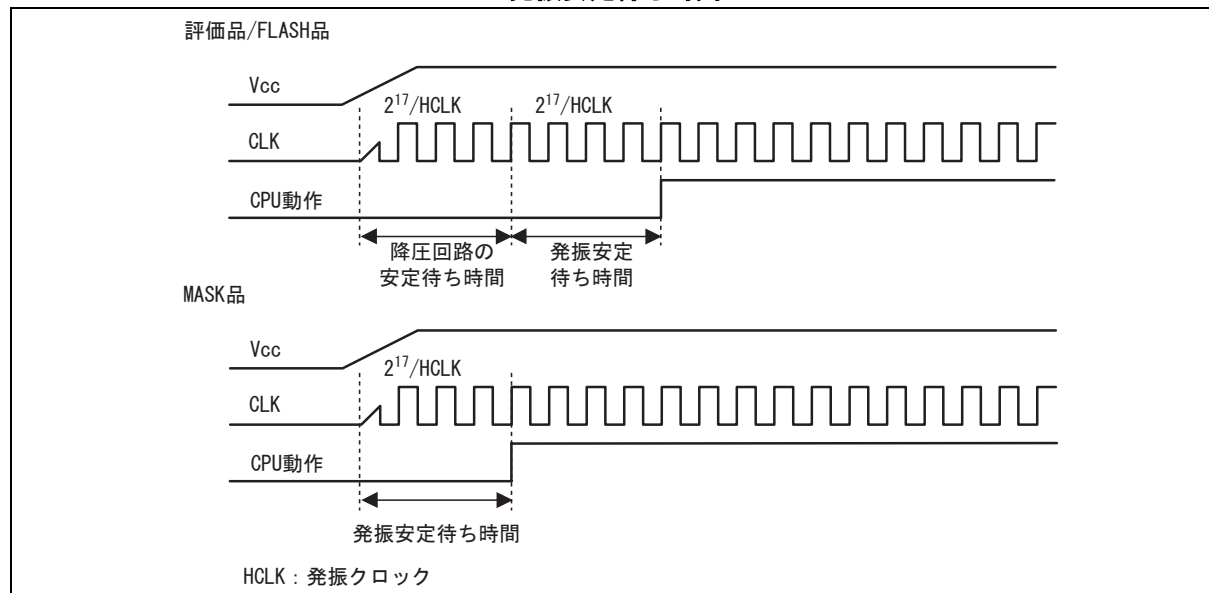
リセット要因	発振安定待ち時間 () 内は発振クロック 4MHz 時の周期
パワーオンリセット	評価品 / フラッシュ品 : $2^{18}/\text{HCLK}$ (約 65.54ms) マスク品 : $2^{17}/\text{HCLK}$ (約 32.77ms)
ウォッチドッグタイマ	なし: ただし, WS1, WS0 ビットは "11" に初期化されます。
$\overline{\text{RST}}$ 端子からの外部リセット	なし: ただし, WS1, WS0 ビットは "11" に初期化されます。
ソフトウェアリセット	なし: ただし, WS1, WS0 ビットは "11" に初期化されます。

HCLK: 発振クロック

WS1, WS0: クロック選択レジスタ (CKSCR) の発振安定待ち時間選択ビット

図 4.2-1 に、パワーオンリセット時の評価用品 / フラッシュメモリ品とマスク ROM 品の発振安定待ち時間について示します。

図 4.2-1 パワーオンリセット時の評価用品 / フラッシュメモリ品とマスク ROM 品の発振安定待ち時間



<注意事項>

セラミックや水晶などの振動子は、発振を開始してから固有の振動数に安定するまで、一般に数ミリ秒から十数ミリ秒の発振安定待ち時間が必要です。そのため、使用する振動子に合わせた値を設定してください。

詳細は、「5.5 発振安定待ち時間」を参照してください。

■ 発振安定待ちリセット状態

電源投入時のリセット，サブクロックモード中のリセットおよびストップモードのリセットに対するリセット動作は、タイムベースタイマによって作られる発振安定待ち時間が経過してからとなります。この場合，外部リセット入力解除されていなければ，外部リセット解除後にリセット動作を行います。

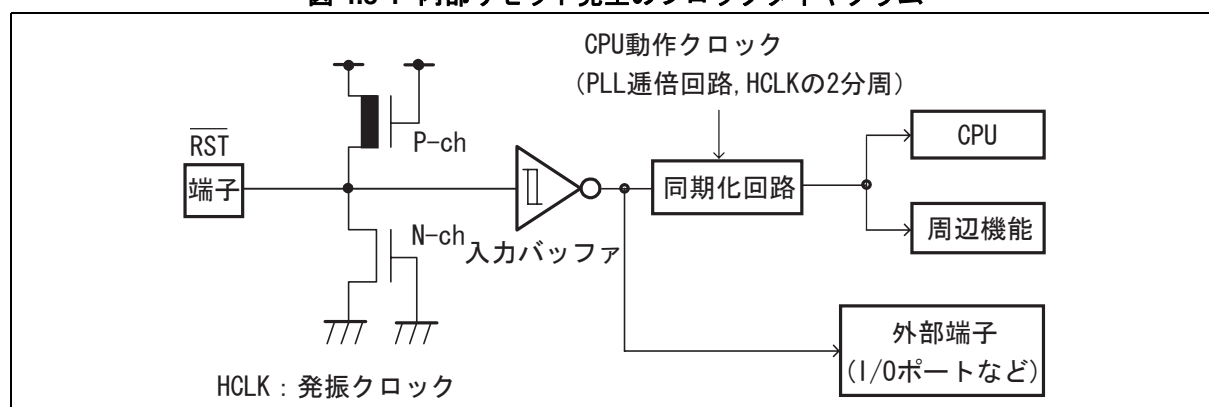
4.3 外部リセット端子

外部リセット端子 ($\overline{\text{RST}}$ 端子) は, リセット入力専用端子で, "L" レベルの入力によって内部リセットを発生します。MB90480/485 シリーズは, CPU 動作クロックに同期してリセットがかかりますが, 外部端子 (I/O ポートなど) のみ非同期でリセット状態に遷移します。

■ 外部リセット端子のブロックダイアグラム

図 4.3-1 に, 内部リセット発生ブロックダイアグラムを示します。

図 4.3-1 内部リセット発生ブロックダイアグラム



<注意事項>

書込み動作中のリセットによるメモリ破壊を防ぐため, $\overline{\text{RST}}$ 端子入力の受付けをメモリが破壊されないサイクルで行います。

また, 内部回路の初期化には, クロックが必要です。特に外部クロックで動作させる場合は, リセット入力時にクロックを入力する必要があります。

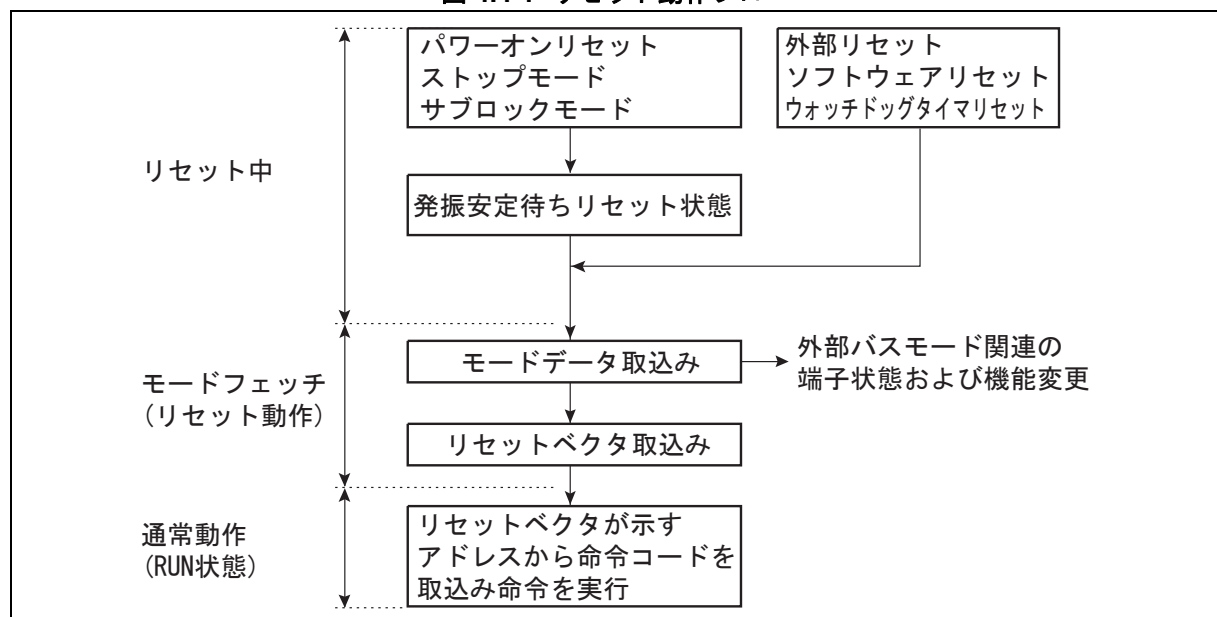
4.4 リセット動作

リセットが解除されると、モード端子の設定によるモードデータとリセットベクタの読出し先が選択され、モードフェッチを行います。このモードフェッチにより、CPUの動作モードとリセット動作終了後の実行開始アドレスが決定されます。電源投入時、またはストップモードおよびサブクロックモードからのリセットによる復帰では、発振安定待ち時間が経過してからモードフェッチを行います。

■ リセット動作の概要

図 4.4-1 に、リセット動作フローを示します。

図 4.4-1 リセット動作フロー



■ モード端子

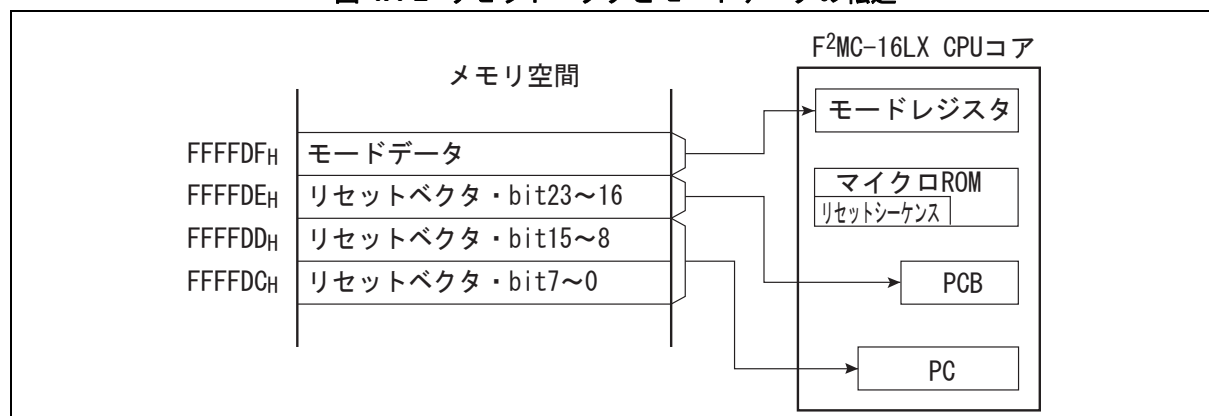
モード端子 (MD2 ～ MD0) は、リセットベクタとモードデータの取込み方法を指定します。リセットベクタとモードデータの取込みは、リセットシーケンスで行います。モード端子の詳細は、「7.2 モード端子 (MD2 ～ MD0)」を参照してください。

■ モードフェッチ

リセットが解除されると、CPU は、リセットベクタとモードデータを CPU コア内の該当レジスタ内にハードウェア転送します。リセットベクタとモードデータは、"FFFFDC_H ～ FFFFDF_H" の4バイトに割り当てられています。CPU は、リセット解除で直ちにこれらのアドレスをバスに出力し、リセットベクタとモードデータを取り込みます。このモードフェッチで、CPU はリセットベクタが指すアドレスから処理を開始します。

図 4.4-2 に、リセットベクタとモードデータの転送を示します。

図 4.4-2 リセットベクタとモードデータの転送



<参考>

リセットベクタとモードデータを内部 ROM, または外部メモリのどちらから読み出すかは、モード端子によって指定します。モード端子で外部ベクタモードを指定すると、外部メモリからリセットベクタとモードデータを読み出します。シングルチップモード、または内部 ROM 外部バスモードで使用する場合は、モード端子を内部ベクタモードに設定してください。

● モードデータ (アドレス :FFFFDF_H)

モードレジスタの内容を変更できるのはリセット動作だけです。モードレジスタの設定はリセット動作以降に有効となります。モードデータの詳細は、「7.3 モードデータ」を参照してください。

● リセットベクタ (アドレス :FFFFDC_H ~ FFFFDE_H)

リセット動作終了後の実行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実行を開始します。

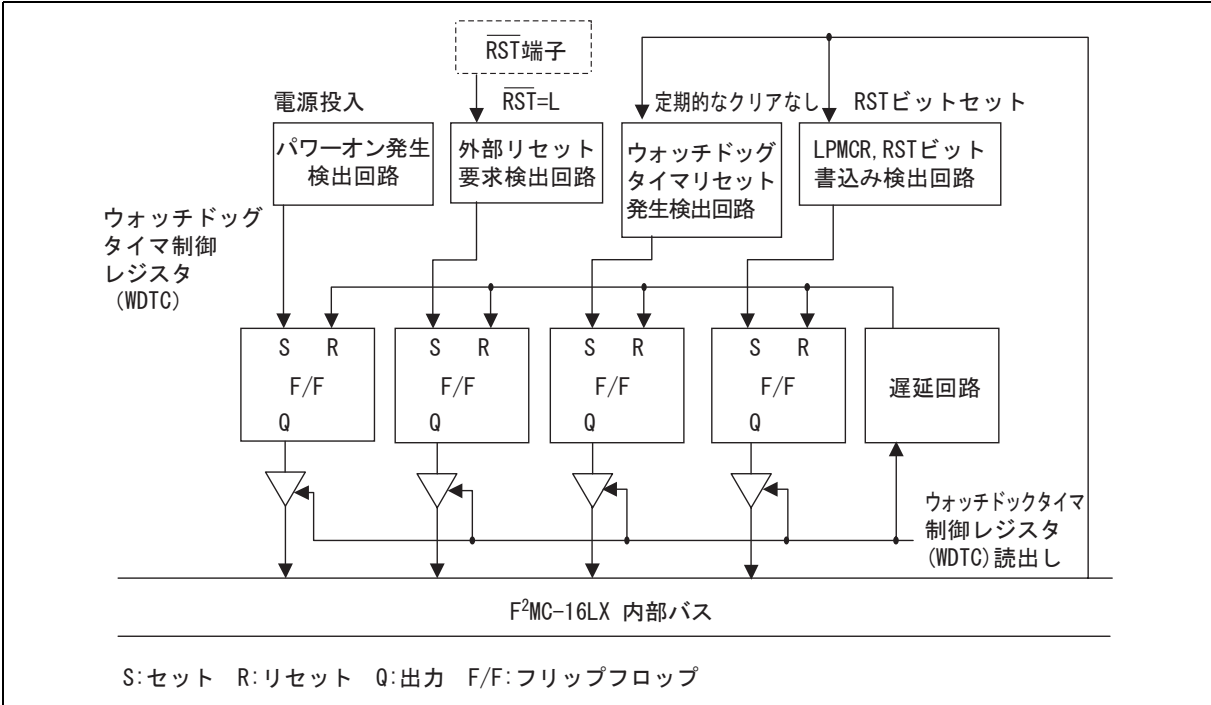
4.5 リセット要因ビット

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) を読み出すことによって識別します。

■ リセット要因ビット

図 4.5-1 に示すように、それぞれのリセット要因には対応したフリップフロップ回路が用意されています。フリップフロップ回路の状態は、ウォッチドッグタイマ制御レジスタ (WDTC) を読み出して確認できます。リセット解除後にリセット発生要因を識別する必要がある場合には、WDTC レジスタの値をソフトウェアで読み出した上で、適切なプログラムへ分岐するようにしてください。

図 4.5-1 リセット要因ビットブロックダイアグラム



■ リセット要因ビットとリセット要因の対応

ウォッチドッグタイマ制御レジスタ (WDTC) のリセット要因ビットの構成を図 4.5-2 に、リセット要因ビットの内容とリセット要因の対応を表 4.5-1 に示します。

詳細は、「10.2 ウォッチドッグタイマ制御レジスタ (WDTC)」を参照してください。

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

ビット	15 - 8	7	6	5	4	3	2	1	0	
0000A8 _H	(TBTC)	PONR	予約	WRST	ERST	SRST	WTE	WT1	WT0	
		X	X	X	X	X	1	1	1	初期値
		R	R	R	R	R	W	W	W	R/W
		R: リードオンリ		W: ライトオンリ						

表 4.5-1 リセット要因ビットの内容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の発生	1	X	X	X
ウォッチドッグタイマのオーバフロー発生	*	1	*	*
端子からの外部リセット要求発生	*	*	1	*
ソフトウェアリセット要求の発生	*	*	*	1

* : 前の状態を保持

X : 不定

■ リセット要因ビットの注意事項

● 複数のリセット要因が発生した場合

リセット要因が複数発生する場合は、WDTC レジスタの対応するそれぞれのリセット要因ビットが "1" にセットされます。例えば、 $\overline{\text{RST}}$ 端子からの外部リセット要求の発生とウォッチドッグタイマのオーバフローが同時に発生した場合、リセット要因ビットのうち ERST ビットと WRST ビットが "1" になります。

● パワーオンリセットの場合

パワーオンリセット時には、リセット要因ビットのうち PONR ビットが "1" にセットされ、PONR ビット以外のリセット要因ビットは不定となります。このため、PONR ビットが "1" の場合は、PONR ビット以外のリセット要因ビットを無視するようにプログラムを作成してください。

● リセット要因ビットのクリア

リセット要因ビットは、WDTC レジスタを読み出した場合のみクリアされます。一度発生したリセット要因に対応するビットは、リセットが発生してもクリアされません ("1" のままです)。

<注意事項>

パワーオンリセットが発生しない条件で電源を立ち上げた場合、WDTC レジスタの値は保証されません。

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。

■ リセット中の端子の状態

リセット中の端子の状態は、モード端子 (MD2 ～ MD0) の設定によって決定されます。

● 内部ベクタモード設定時 (MD2 ～ MD0=011_B)

I/O 端子 (リソース端子) はすべてハイインピーダンスになり、モードデータの読出し先は内部 ROM になります。

リセット中の各端子の状態については、「6.7 スタンバイモード、ホールド、リセット時の端子状態」を参照してください。

■ モードデータ読出し後の端子の状態

モードデータ読出し後の端子の状態は、モードデータ (M1, M0) によって決定されます。

● シングルチップモード選択時 (M1, M0=00_B)

I/O 端子 (リソース端子) はすべてハイインピーダンスになり、モードデータの読出し先は内部 ROM になります。

<注意事項>

リセット要因が発生した場合にハイインピーダンスとなる端子は、その端子に接続した機器が誤動作しないように配慮してください。

第5章

クロック

MB90480/485 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), PLL 出力選択レジスタ (PLLOS)
- 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 や周辺機能の動作クロックである内部クロックの動作を制御します。本書では、クロックの種類に応じて以下のようによびます。

- マシナクロック：内部クロックのことをいいます。
 - マシンサイクル：マシナクロックの1周期をいいます。
 - 発振クロック：高速発振端子から供給されるクロックをいいます。
 - PLL クロック：内部の PLL 発振によるクロックをいいます。
 - サブクロック：低速発振端子から供給されるクロックの4分周をいいます。
-

■ クロックの概要

クロック発生部は発振回路を内蔵しており、外部に振動子を接続することによって、発振クロックおよびサブクロックを発生します。外部で生成したクロックを入力して、発振クロックにすることもできます。また、PLL クロック逡倍回路を内蔵しており、発振クロックの逡倍クロックを4種類発生できます。クロック発生部では、発振安定待ち時間の制御、PLL クロック逡倍制御およびクロックセクタでのクロックの切換えによる、内部クロックの動作制御を行います。

● 発振クロック (HCLK)

高速発振端子に振動子を接続するか、外部クロックを入力して発生させたクロックです。

● サブクロック (SCLK)

時計タイマを動作させるクロックです。また、低速のマシナクロックとしても使用可能です。

低速発振端子に振動子を接続するか、外部クロックを入力して発生させたクロックの4分周です。

● 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と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、内蔵の PLL クロック逡倍回路 (PLL 発振回路) により逡倍したクロックです。4種類のクロックを選択できます。

● マシナクロック (ϕ)

CPU と周辺機能の動作クロックです。このクロックの1周期を、マシンサイクル ($1/\phi$) としています。メインクロック (発振クロックの2分周クロック)、サブクロックと4種類の逡倍クロックの中から1種類を選択できます。

<注意事項>

発振クロックは4.5MHz ~ 25MHz の発振が可能です。ただし PLL 使用時でマシナクロックが20MHz ~ 25MHz での使用は PLLOS レジスタの PLL2 ビットを "1" に設定してください。CPU および周辺機能は最大動作周波数は25MHz です。最大動作周波数を超える逡倍率を指定した場合、デバイスは正常に動作しません。

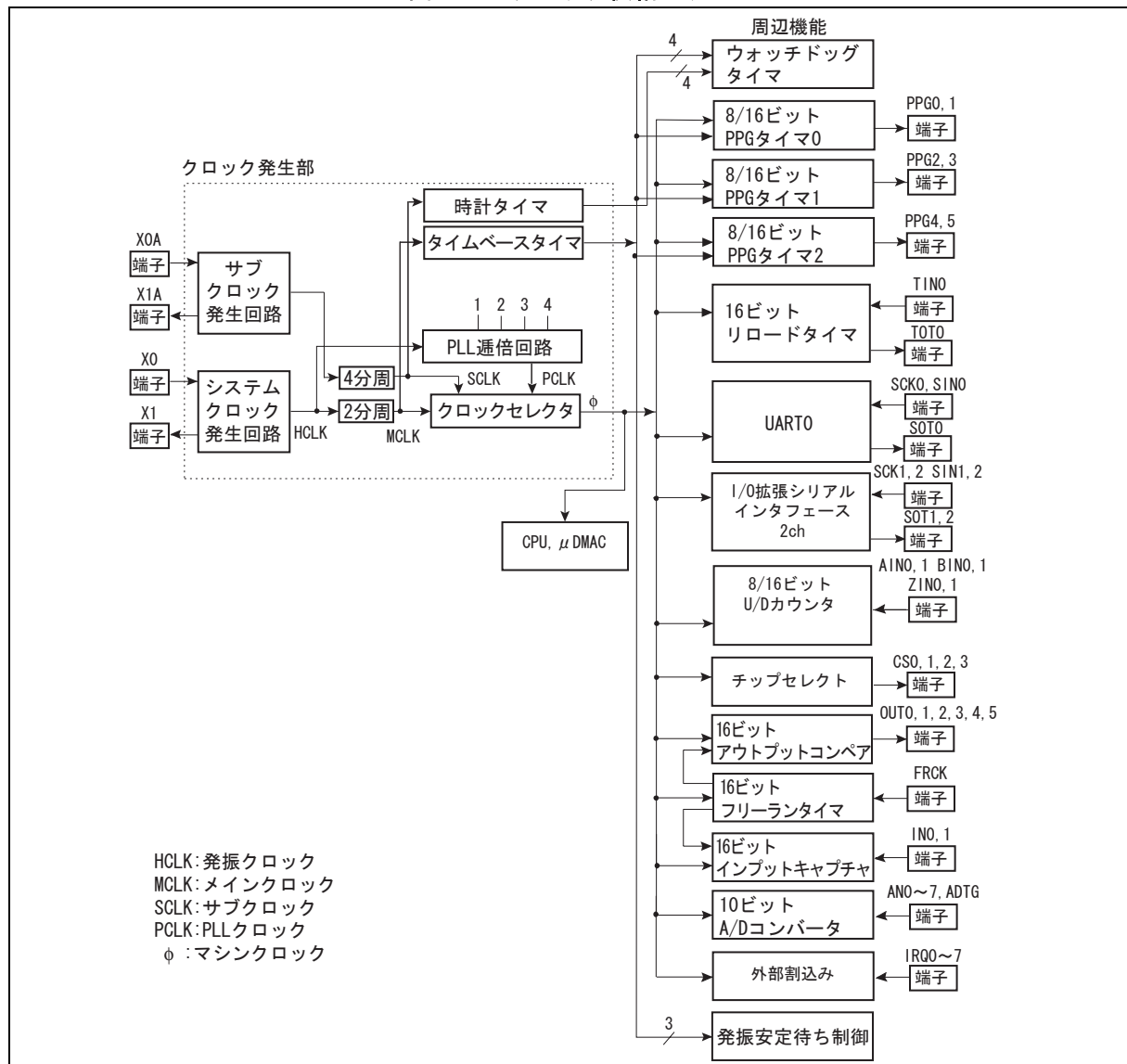
PLL 発振は4.5MHz ~ 25MHz の範囲で発振しますが、PLL 発振範囲は動作電圧および逡倍率によって異なります。

■ クロック供給マップ

CPU や周辺機能の動作クロックとして、クロック発生部で生成したマシナクロックを供給しています。CPU と周辺機能の動作は、メインクロックと PLL クロックの切換え（クロックモード）や PLL クロック通倍率切換えの影響を受けます。一部の周辺機能へは、タイムベースタイマの分周出力が供給されており、各周辺で動作クロックを選択できるようになっています。

図 5.1-1 に、クロック供給マップを示します。

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、次のブロックで構成されています。

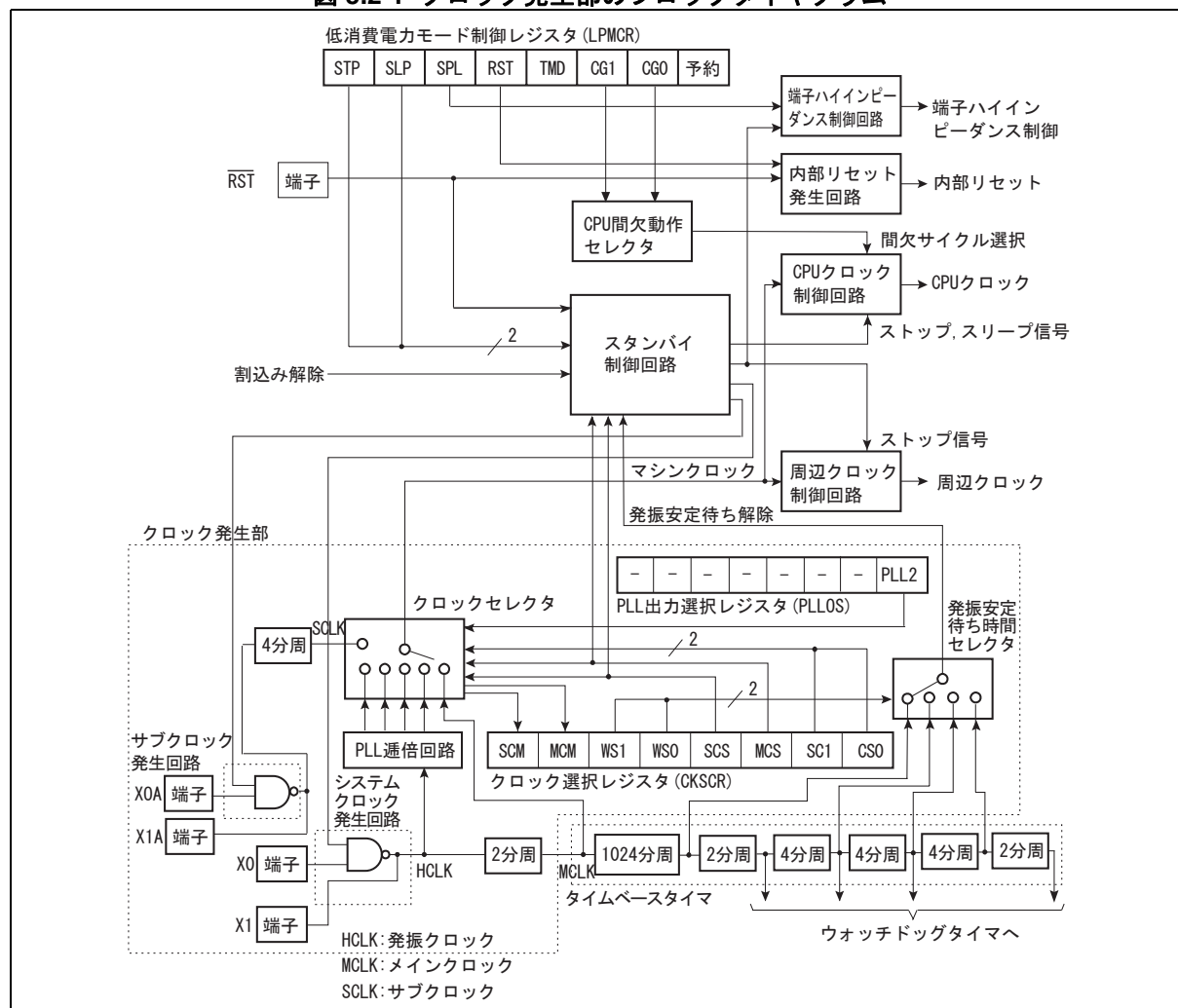
- システムクロック発生回路 / サブクロック発生回路
- PLL 通倍回路
- クロックセクタ
- クロック選択レジスタ (CKSCR), PLL 出力選択レジスタ (PLLOS)
- 発振安定待ち時間セクタ

■ クロック発生部のブロックダイアグラム

図 5.2-1 に、クロック発生部のブロックダイアグラムを示します。

なお、図 5.2-1 には、スタンバイ制御回路、タイムベースタイマの回路も含まれています。

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

高速発振端子に接続した振動子によって、発振クロック (HCLK) を発生します。外部クロックを入力することもできます。

● サブクロック発生回路

低速発振端子に接続した振動子によって、サブクロック (SCLK) を発生します。外部クロックを入力することもできます。

● PLL 通倍回路

PLL 発振で発振クロックを通倍し、CPU 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、サブクロックと、4 種類の PLL クロックから、CPU 系クロック制御回路および周辺系クロック制御回路へ供給されるクロックを選択します。

● クロック選択レジスタ (CKSCR)

発振クロックと PLL クロックの切換え、発振安定待ち時間の選択、PLL クロックの通倍率の選択などを行います。

● PLL 出力選択レジスタ (PLLOS)

PLL を使用してマシナクロックを 20MHz ～ 25MHz で使用する場合、このレジスタで CKSCR レジスタの設定した PLL の通倍を 2 倍にします。

● 発振安定待ち時間セクタ

ストップモードの解除時およびウォッチドッグリセット時の発振クロックの発振安定待ち時間を選択する回路です。4 種類のタイムベースタイマ出力を選択します。

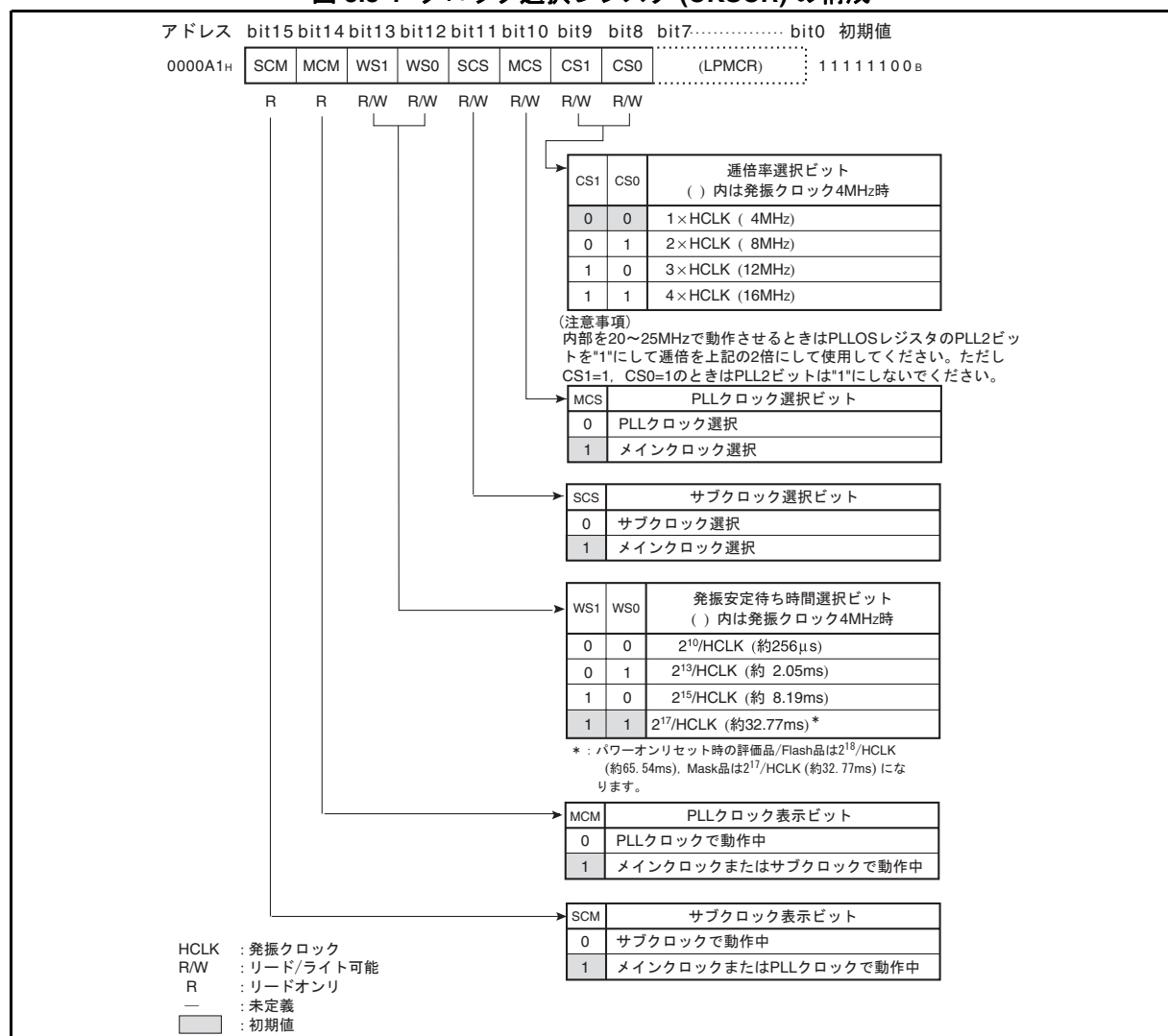
5.3 クロック選択レジスタ (CKSCR), PLL 出力選択レジスタ (PLLOS)

クロック選択レジスタ (CKSCR) は、メインクロック、サブクロックと PLL クロックの切換え、発振安定待ち時間の選択、PLL クロックの通倍率の選択などを行います。また PLL 出力選択レジスタ (PLLOS) は、PLL を使用してマシンのクロックを 20MHz ~ 25MHz にする場合、設定が必要となります。

■ クロック選択レジスタ (CKSCR) の構成

図 5.3-1 にクロック選択レジスタ (CKSCR) の構成を、表 5.3-1 にクロック選択レジスタの各ビットの機能について説明します。

図 5.3-1 クロック選択レジスタ (CKSCR) の構成



<注意事項>

マシンのクロック選択ビット (MCS) は、リセットによってメインクロック選択に初期化されます。

表 5.3-1 クロック選択レジスタ (CKSCR) の各ビットの機能 (1 / 2)

ビット名		機 能
bit15	SCM: サブクロック 表示ビット	<p>マシニングロックとして、メインクロック、またはサブクロックのどちらが選択されているかを表示するビットです。</p> <ul style="list-style-type: none"> このビットが "0" の場合、サブクロックが選択されていることを示し、"1" の場合、メインクロック、または PLL クロックが選択されていることを示します。 SCS=1 で、かつ SCM=0 ならば、メインクロック発振安定待ち時間中であることを示します。 書込みは、動作に影響しません。
bit14	MCM: PLL クロック 表示ビット	<p>マシニングロックとして、メインクロック、または PLL クロックのどちらが選択されているかを表示するビットです。</p> <ul style="list-style-type: none"> このビットが "0" の場合、PLL クロックが選択されていることを示し、"1" の場合、メインクロック、またはサブクロックが選択されていることを示します。 PLL クロック選択ビット (MCS)=0 で、かつ MCM=1 ならば、PLL クロック発振安定待ち時間中であることを示します。 書込みは、動作に影響しません。
bit13, bit12	WS1, WS0: 発振安定待ち 時間選択 ビット	<p>ストップモード解除時、サブクロックモードからメインクロックモードへの移行時およびサブクロックモードから PLL クロックモードへの移行時の発振クロックの発振安定待ち時間を選択します。 すべてのリセット要因で "11_B" に初期化されます。</p> <p>(注意事項)</p> <p>発振安定待ち時間は、使用する振動子に合わせて適切な値を設定する必要があります。「4.2 リセット要因と発振安定待ち時間」を参照してください。"00_B" の設定はメインクロックモード時のみ設定してください。</p> <p>メインクロックモードから PLL クロックモードに切り換えた場合の発振安定待ち時間は、$2^{14}/HCLK$ で固定です。</p> <p>サブクロックモードに切り換えた場合および PLL ストップモードから PLL クロックモードに復帰する場合の発振安定待ち時間は、本ビットに設定された値に従います。PLL クロック発振安定待ち時間は $2^{14}/HCLK$ 以上必要ですので、サブクロックモードから PLL クロックモードに切り換える場合および PLL ストップモードに移行する場合、本ビットには "10_B" または "11_B" を設定してください。</p>
bit11	SCS: サブクロック 選択ビット	<p>マシニングロックとして、メインクロック、またはサブクロックのどちらを選択するかを指定するビットです。</p> <ul style="list-style-type: none"> このビットが "0" のときサブクロックを選択し、"1" のときメインクロックを選択します。 このビットが "1" のときに "0" を書き込むと、サブクロックに同期をとって (約 130 μs)、サブクロックモードに切り換わります。 このビットが "0" のときに "1" を書き込むと、メインクロックの発振安定待ち時間が発生するために、自動的にタイムベースタイマがクリアされます。 <p>すべてのリセット要因で "1" に初期化されます。</p> <p>(注意事項)</p> <p>サブクロック選択時の動作クロックは、サブクロックを使用します (低速発振 32kHz 時、マシニングロックは 8kHz になります)。</p> <p>SCS, MCS がともに "0" のときは、SCS が優先され、サブクロックが選択されます。</p>

表 5.3-1 クロック選択レジスタ (CKSCR) の各ビットの機能 (2 / 2)

ビット名	機 能
bit10 MCS: PLL クロック 選択ビット	<p>マシナクロックとして、メインクロック、または PLL クロックのどちらを選択するかを指定するビットです。</p> <ul style="list-style-type: none"> このビットが "0" の場合には PLL クロックを選択し、"1" の場合にはメインクロックを選択します。 このビットが "1" の場合に "0" を書き込むと、PLL クロックの発振安定待ち時間が発生するために、自動的にタイムベースタイマがクリアされます。さらに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) もクリアされます。 メインクロックモードから PLL クロックモードに切り換えた場合の発振安定待ち時間は、$2^{14}/\text{HCLK}$ で固定です (発振クロック 4MHz 時、発振安定待ち時間は約 4.1ms になります)。サブクロックモードから PLL クロックモードに切り換えた場合の発振安定待ち時間は、発振安定待ち時間選択ビット (CKSCR:WS1, WS0) に設定された値に従います。 メインクロック選択時の動作クロックは、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 4MHz の場合、動作クロックは 2MHz になります)。 <p>すべてのリセット要因で "1" に初期化されます。</p> <p>(注意事項)</p> <p>MCS ビットが "1" の場合に "0" を書き込む場合には、TBTC レジスタの割込み要求許可ビット (TBIE), またはインタラプトレベルマスクレジスタ (ILM) によって、タイムベースタイマ割込みが抑止されている状態で行うようにしてください。</p>
bit9, bit8	<p>PLL クロックの通倍率を選択するビットです。</p> <p>4 種類の通倍率から選択できます。</p> <p>すべてのリセット要因で "00_B" に初期化されます。</p> <p>(注意事項)</p> <p>MCS ビット、または MCM ビットが "0" の場合には、書き込みが抑止されます。いったん、MCS ビットを "1" に設定 (メインクロックモード) 後に、CS1, CS0 ビットを書き換えてください。</p>

HCLK: 発振クロック

■ PLL 出力選択レジスタ (PLLOS) の構成

図 5.3-2 に PLL 出力選択レジスタ (PLLOS) の構成を、表 5.3-2 に PLL 出力選択レジスタの各ビットの機能について説明します。

図 5.3-2 PLL 出力選択レジスタ (PLLOS) の構成

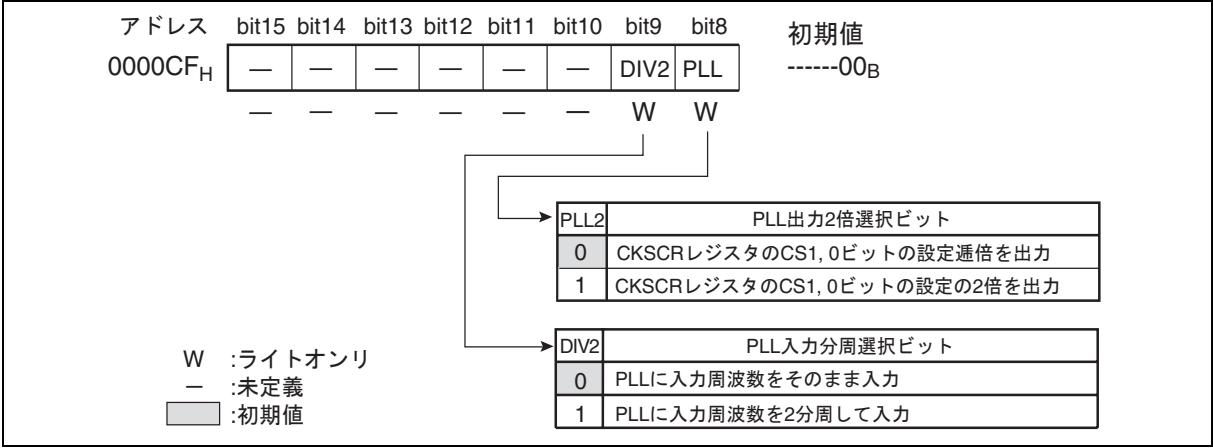


表 5.3-2 PLL 出力選択レジスタ (PLLOS) の各ビットの機能 (1 / 2)

ビット名		機 能
bit15 ～ bit10	未定義ビット	使用しておりません。
bit9	DIV2 : PLL 入力分周 選択ビット	<ul style="list-style-type: none">• PLL への入力クロックを分周するか、そのまま入力するか選択します。• すべてのリセット要因で "0" に初期化されます。• 読出し値は不定です。• このビットは PLL のクロックを使用しているときは変更しないでください。• PLL1, 2, 3, 4 通倍設定で、内部クロックを $20\text{MHz} < f_{\text{CP}} \leq 25\text{MHz}$ で使用する場合、PLLOS レジスタを DIV2 ビット =1, PLL2 ビット =1 に設定してください (ただし 4 通倍時は入力周波数が 6MHz 以上)。 例 原発信周波数 24MHz PLL1 通倍で使用する場合： CKSCR レジスタ CS1 ビット = 0, CS0 ビット = 0 PLLOS レジスタ DIV2 ビット = 1, PLL2 ビット = 1 例 原発信周波数 8MHz PLL3 通倍で使用する場合： CKSCR レジスタ CS1 ビット = 1, CS0 ビット = 0 PLLOS レジスタ DIV2 ビット = 1, PLL2 ビット = 1• PLL2, 4 通倍設定で、内部クロックを $20\text{MHz} < f_{\text{CP}} \leq 25\text{MHz}$ で使用する場合、以下の設定でも対応可能です。 PLL2 通倍 : CKSCR レジスタ CS1 ビット = 0, CS0 ビット = 0 PLLOS レジスタ DIV2 ビット = 0, PLL2 ビット = 1 PLL4 通倍 : CKSCR レジスタ CS1 ビット = 0, CS0 ビット = 1 PLLOS レジスタ DIV2 ビット = 0, PLL2 ビット = 1• PLL6 通倍, PLL8 通倍設定で使用する場合、PLLOS レジスタを DIV2 ビット = 0, PLL2 ビット = 1 に設定してください。 例 原発信周波数 4MHz PLL6 通倍で使用する場合： CKSCR レジスタ CS1 ビット = 1, CS0 ビット = 0 PLLOS レジスタ DIV2 ビット = 0, PLL2 ビット = 1 例 原発信周波数 3MHz PLL8 通倍で使用する場合： CKSCR レジスタ CS1 ビット = 1, CS0 ビット = 1 PLLOS レジスタ DIV2 ビット = 0, PLL2 ビット = 1

表 5.3-2 PLL 出力選択レジスタ (PLLOS) の各ビットの機能 (2 / 2)

ビット名		機 能
bit8	PLL2: PLL 出力 2 倍 選択ビット	<p>PLL 使用時にマシナクロックを 20MHz ～ 25MHz で使用するとき PLL の通倍を 2 倍にするビットです。</p> <ul style="list-style-type: none"> • すべてのリセット要因で "0" に初期化されます。 • 読出し値は不定です。 • このビットは PLL のクロックを使用しているときは変更しないでください。

5.4 クロックモード

クロックモードには、メインクロックモード、PLL クロックモード、サブクロックモードがあります。

■ メインクロックモード、PLL クロックモード、サブクロックモード

● メインクロックモード

メインクロックモードは、CPU および周辺リソースの動作クロックとして発振クロックを2分周したクロックを使用します。

● PLL クロックモード

PLL クロックモードは、CPU および周辺機能の動作クロックとして PLL クロックを使用します。PLL クロックの通倍率は、クロック選択レジスタ (CKSCR) によって選択します。

● サブクロックモード

サブクロックモードは、CPU および周辺リソースの動作クロックとしてサブクロックを使用します。

■ クロックモードの移行

クロックモードは、CKSCR レジスタの PLL クロック選択ビット (MCS) とサブクロック選択ビット (SCS) への書込み動作によって、メインクロックモード、PLL クロックモード、サブクロックモードに移行します。

● メインクロックモードから PLL クロックモードへの移行

メインクロックモードの状態では、CKSCR レジスタの MCS ビットを "1" から "0" に書き換えた場合には、PLL クロックの発振安定待ち時間 ($2^{14}/HCLK$) 後に、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移行

PLL クロックモードの状態では、CKSCR レジスタの MCS ビットを "0" から "1" に書き換えた場合には、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 ~ 8 PLL クロック後) で PLL クロックからメインクロックに切り換わります。

● メインクロックモードからサブクロックモードへの移行

メインクロックモードの状態では、CKSCR レジスタの SCS ビットを "1" から "0" に書き換えた場合には、サブクロックに同期をとって (約 130 μ s)、メインクロックからサブクロックに切り換わります。

● サブクロックモードからメインクロックモードへの移行

サブクロックモードの状態では、CKSCR レジスタの SCS ビットを "0" から "1" に書き換えた場合には、メインクロックの発振安定待ち時間後に、サブクロックからメインクロックに切り換わります。発振安定待ち時間は、CKSCR レジスタの発振安定待ち時間選択ビット (WS1, WS0) によって選択します。

● PLL クロックモードからサブクロックモードへの移行

PLLクロックモードの状態では、クロック選択レジスタ(CKSCR)のサブクロック選択ビット(SCS)を"1"から"0"に書き換えた場合には、PLLクロックからサブクロックに切り換わります。

● サブクロックモードから PLL クロックモードへの移行

サブクロックモードの状態では、CKSCR レジスタの SCS ビットを "0" から "1" に書き換えた場合には、メインクロックの発振安定待ち時間後に、サブクロックから PLL クロックに切り換わります。発振安定待ち時間は、CKSCR レジスタの発振安定待ち時間選択ビット (WS1, WS0) によって選択します。

<注意事項>

- マシナクロックは、CKSCR レジスタの PLL クロック選択ビット (MCS), SCS ビットを書き換えても即座には切り換わりません。マシナクロックに依存するリソースを操作する場合は、CKSCR レジスタの PLL クロック表示ビット (MCM) およびサブクロック表示ビット (SCM) を参照してマシナクロックの切り換えが行われたことを確認した後に、リソースの操作をしてください。
- SCS ビット、MCS ビットがともに "0" の場合は、SCS が優先され、サブクロックモードになります。
- 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の切り換えを行わないでください。切り換えの完了は、クロック選択レジスタ (CKSCR) の MCM ビットおよび SCM 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、他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。

■ PLL クロック逡倍率の選択

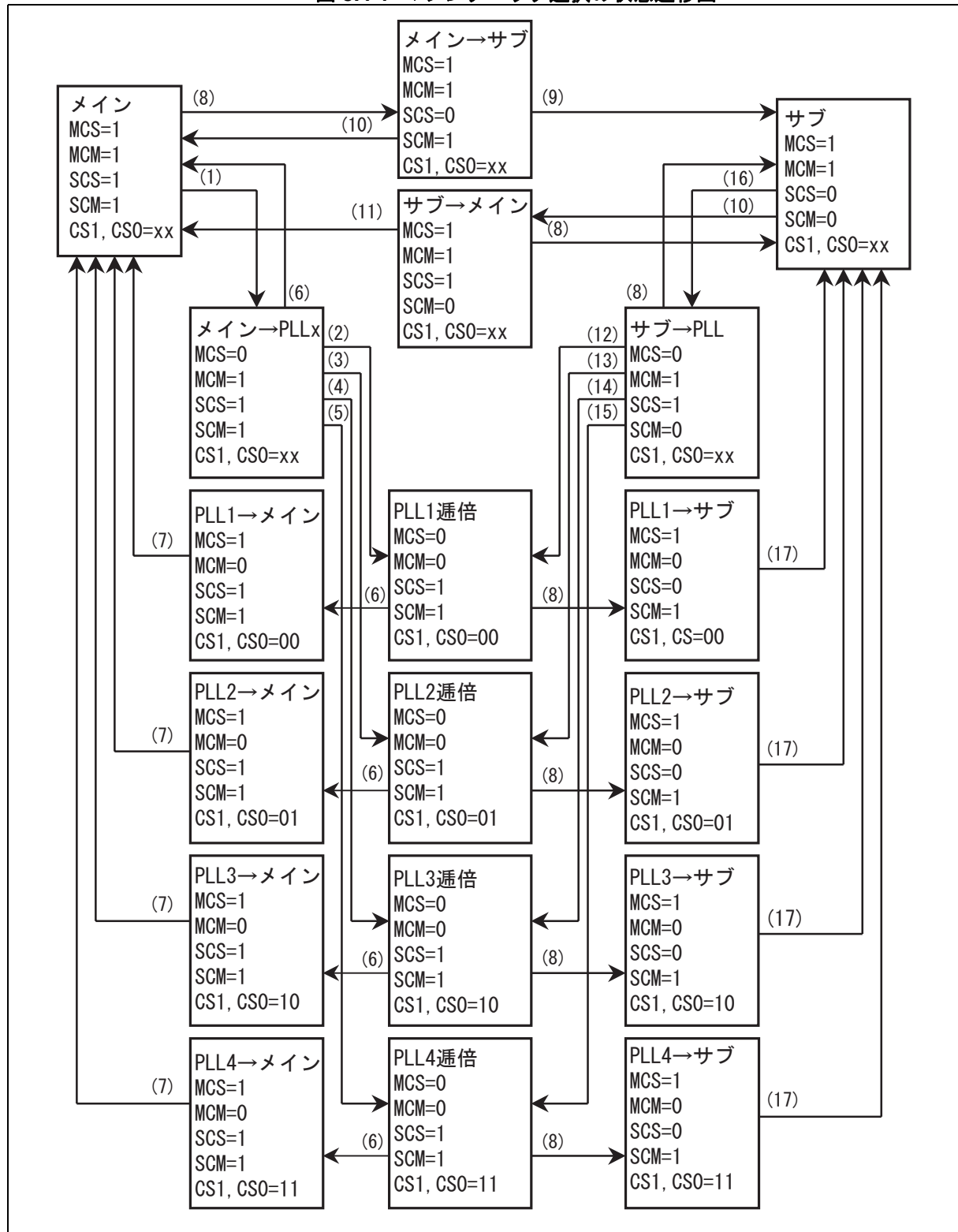
CKSCR レジスタの CS1, CS0 ビットに "00_B" ~ "11_B" を書き込むことによって、1 ~ 4 逡倍の 4 種類の PLL クロック逡倍率を選択できます。

■ マシナクロック

PLL 逡倍回路から出力される PLL クロック、メインクロック、サブクロックが、マシナクロックとなります。このマシナクロックが CPU および周辺機能に供給されます。メインクロック、PLL クロック、サブクロックは、CKSCR レジスタの MCS ビット、または SCS ビットへの書き込みによりいずれかを選択できます。

図 5.4-1 に、マシナクロック選択の状態遷移図を示します。

図 5.4-1 マシンのクロック選択の状態遷移図



(続く)

第5章 クロック

(続き)

- (1) MCS ビット "0" 書込み
- (2) PLL クロック発振安定待ち終了 &CS1, CS0=00
- (3) PLL クロック発振安定待ち終了 &CS1, CS0=01
- (4) PLL クロック発振安定待ち終了 &CS1, CS0=10
- (5) PLL クロック発振安定待ち終了 &CS1, CS0=11
- (6) MCS ビット "1" 書込み (ウォッチドッグリセットを含む)
- (7) PLL クロックとメインクロックの同期タイミング
- (8) SCS ビット "0" 書込み
- (9) サブクロック発振安定待ち時間終了 (最大 $2^{14}/\text{SCLK}$)
- (10) SCS ビット "1" 書込み
- (11) メインクロック発振安定待ち終了
- (12) メインクロック発振安定待ち終了 &CS1, CS0=00
- (13) メインクロック発振安定待ち終了 &CS1, CS0=01
- (14) メインクロック発振安定待ち終了 &CS1, CS0=10
- (15) メインクロック発振安定待ち終了 &CS1, CS0=11
- (16) SCS ビット "1" 書込み, MCS ビット "0" 書込み
- (17) PLL クロックとサブクロックの同期タイミング

MCS: クロック選択レジスタ (CKSCR) の PLL クロック選択ビット

MCM: クロック選択レジスタ (CKSCR) の PLL クロック表示ビット

SCS: クロック選択レジスタ (CKSCR) のサブクロック選択ビット

SCM: クロック選択レジスタ (CKSCR) のサブクロック表示ビット

CS1, CS0: クロック選択レジスタ (CKSCR) の通倍率選択ビット

<注意事項>

マシクロックの初期値はメインクロック (MCS=1, SCS=1) です。

SCS, MCS とともに "0" のときは SCS が優先され, サブクロックが選択されます。

サブクロックモードから PLL クロックモードに切り換える場合は, CKSCR レジスタの発振安定待ち時間選択ビット (WS1, WS0) は "10_B" または "11_B" を設定してください。

5.5 発振安定待ち時間

電源投入時，ストップモード解除時，サブクロックからメインクロックへの切換え時，サブクロックから PLL クロックへの切換え時には，発振クロックの発振が停止しているため，発振開始後，発振安定待ち時間をとる必要があります。また，メインクロックから PLL クロック，メインクロックからサブクロックへの切換え時も，発振安定待ち時間が必要となります。

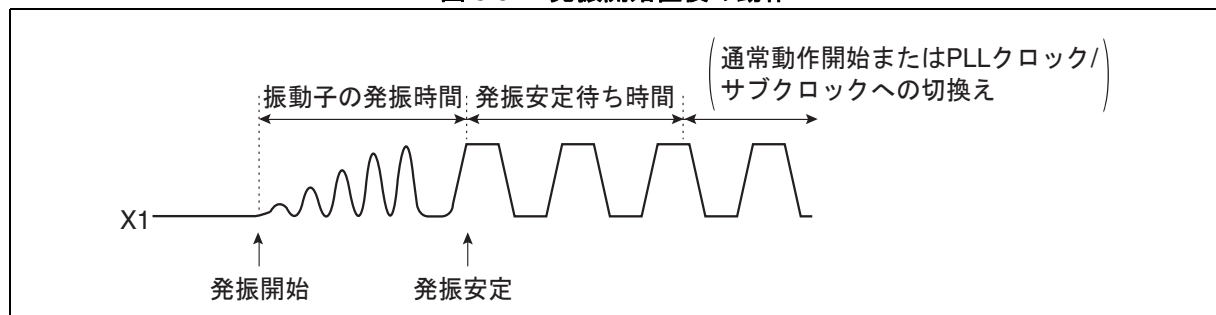
■ 発振安定待ち時間

セラミックや水晶などの振動子は，発振を開始してから固有の振動数（発振周波数）で安定して発振するまでに，一般的に数ミリ秒から数十ミリ秒の時間が必要です。このため，発振開始直後は CPU の動作を禁止し，発振安定待ち時間が経過して十分発振が安定した時点で CPU にクロックを供給するようにしてください。振動子の種類（水晶，セラミックなど）によって発振が安定するまでの時間が異なるため，使用する振動子に合わせて適切な発振安定待ち時間を選択する必要があります。発振安定待ち時間は，クロック選択レジスタ (CKSCR) の設定で選択できます。

メインクロックから PLL クロック，メインクロックからサブクロック，サブクロックからメインクロック，サブクロックから PLL クロックに切り換えた場合，CPU は切換え前のクロックで動作しています。発振安定待ち時間が経過すると，各クロックモードに切り換わります。

図 5.5-1 に，発振開始直後の動作を示します。

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

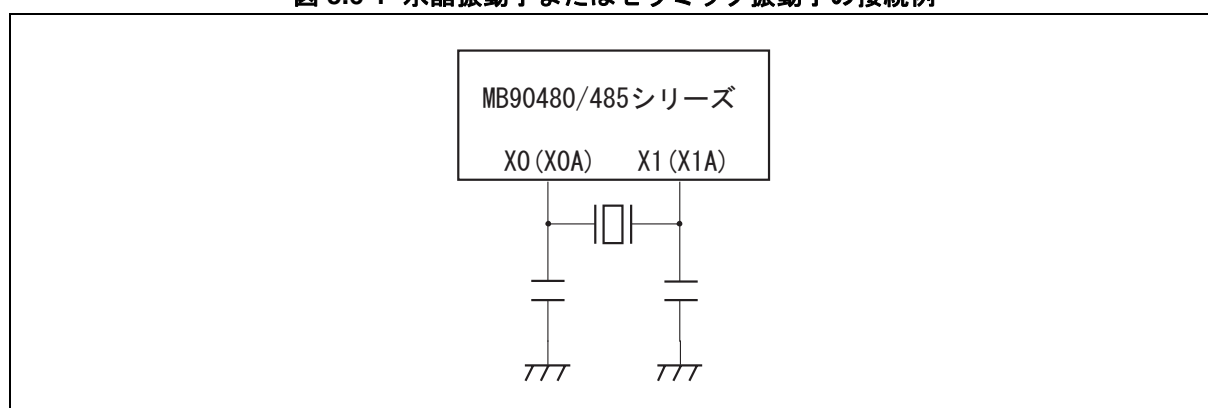
MB90480/485 シリーズは、システムクロック発生回路を内蔵しており、外部に振動子を接続してクロック発生させます。また、外部で生成したクロックを入力することもできます。

■ 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。

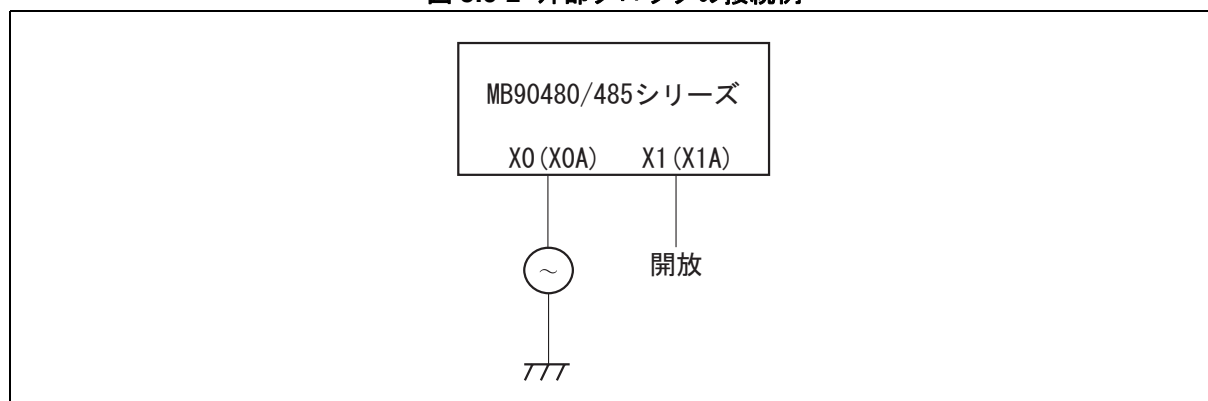
図 5.6-1 水晶振動子またはセラミック振動子の接続例



● 外部クロックの接続例

図 5.6-2 の例で示すように、外部クロックは X0 端子に接続し、X1 端子は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

MB90480/485 シリーズの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 スタンバイモードの状態遷移図
- 6.7 スタンバイモード, ホールド, リセット時の端子状態
- 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

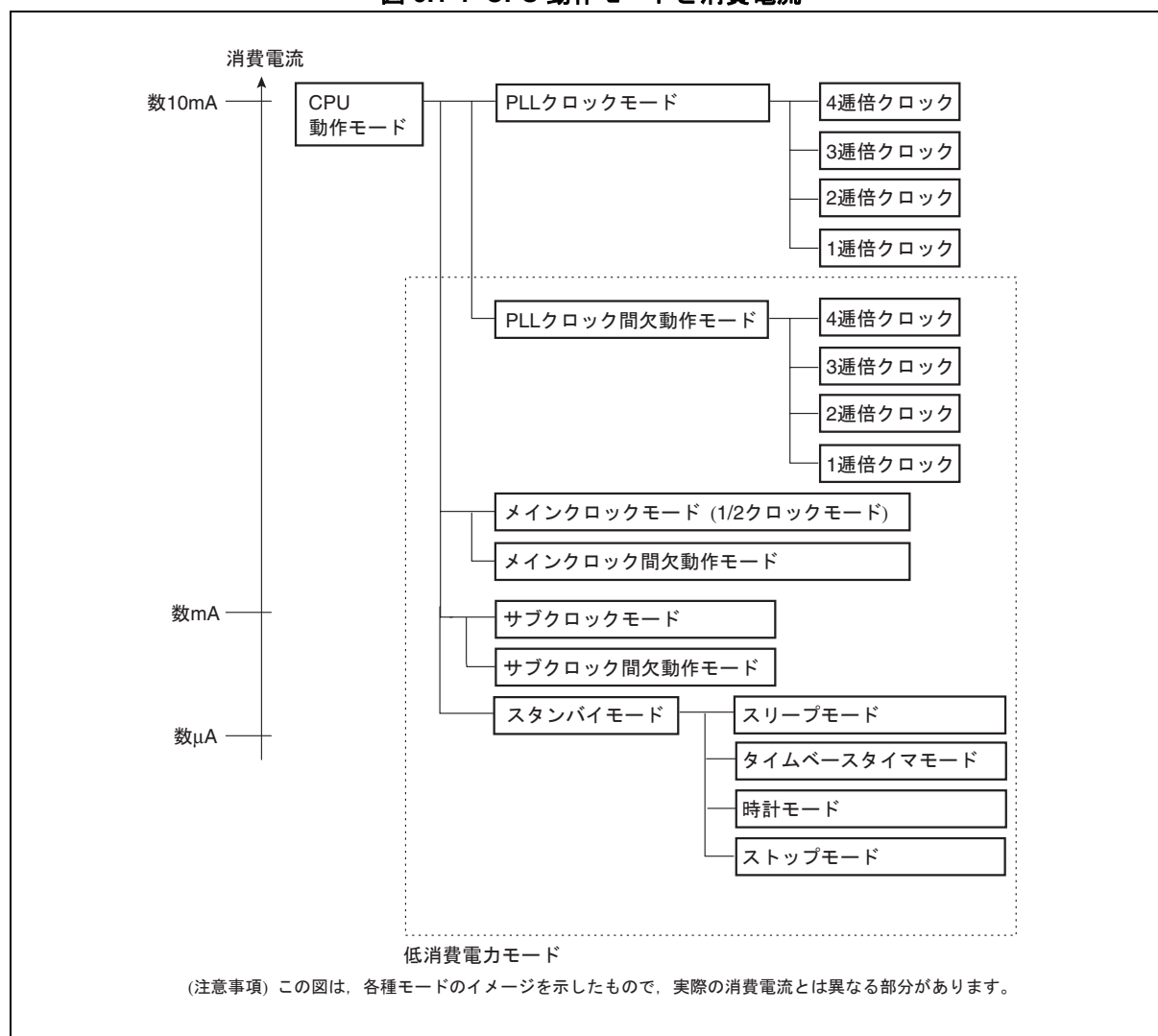
低消費電力モードは、動作クロックの選択とクロックの発振制御によって、消費電力を低減させます。MB90480/485 シリーズでは、以下の低消費電力モードがあります。

- クロックモード (メインクロックモード, サブクロックモード)
- CPU 間欠動作モード (PLL クロック間欠動作モード, メインクロック間欠動作モード, サブ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, タイムベースタイマモード, 時計モード, 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

● メインクロックモード

メインクロックモードは、発振クロック (HCLK) の2分周クロックで、CPUおよび周辺機能を動作させます。メインクロックモード時には、PLL 通倍回路が停止します。

● サブクロックモード

サブクロックモードは、サブクロック (SCLK) で、CPUおよび周辺機能を動作させます。サブクロックモード時には、メインクロックおよび PLL 通倍回路が停止します。

<参考>

クロックモードには、低消費電力モード以外のクロックモードでPLLクロックモードもあります。クロックモードの詳細については、「5.4 クロックモード」を参照してください。

■ CPU 間欠動作モード

CPU 間欠動作モード (PLL クロック間欠動作モード、メインクロック間欠動作モード、サブクロック間欠動作モード) は、周辺機能に高速クロックを供給した状態で、CPUを間欠動作させ、消費電力を低減させます。

■ スタンバイモード

スタンバイモードは、低消費電力制御回路によるCPUへのクロックの供給停止 (スリープモード) や、CPU と周辺機能へのクロックの供給停止 (タイムベースタイマモード)、または発振クロックの停止 (ストップモード) によって消費電力を低減させます。

● スリープモード

スリープモードは、各クロックモード動作中にCPUへの動作クロックを停止させます。CPUは停止し、周辺機能はスリープモード移行前のクロックで動作します。

スリープモードに移行する際のクロックモードによって、メインスリープモード、PLLスリープモード、サブスリープモードに分かれます。

● 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と時計タイマ以外の動作を停止させます。タイムベースタイマと時計タイマ以外の機能が停止します。

● 時計モード

時計モードは、時計タイマのみを動作させます。サブクロックのみ動作し、メインクロックおよび PLL 通倍回路が停止します。

● ストップモード

ストップモードは、原発振を停止させます。すべての機能が停止するため、最も低い消費電力でデータを保持します。

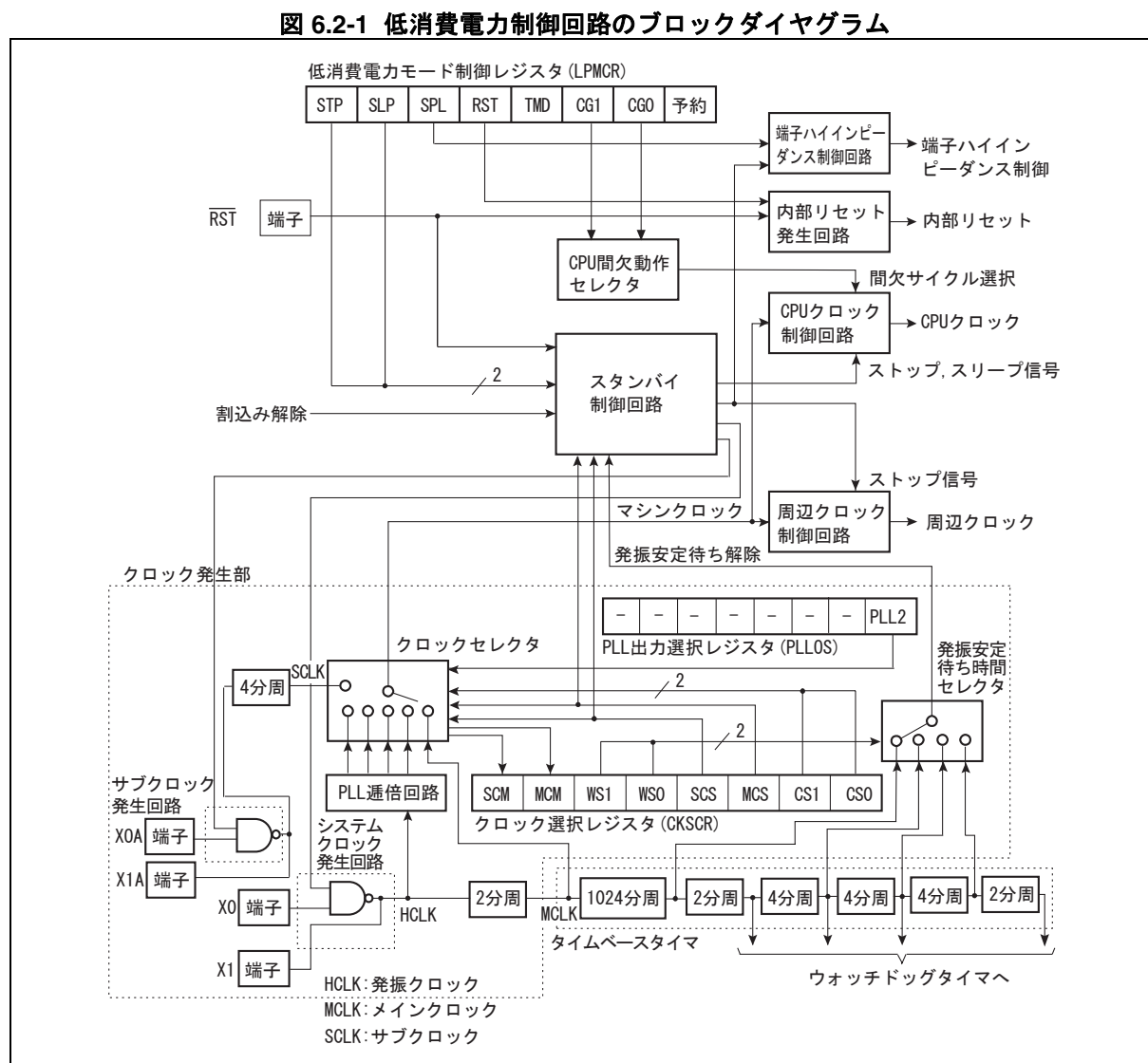
6.2 低消費電力制御回路のブロックダイアグラム

低消費電力制御回路は、以下のブロックで構成されています。

- CPU 間欠動作セクタ
- 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
- 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

図 6.2-1 に、低消費電力制御回路のブロックダイアグラムを示します。



● CPU 間欠動作セクタ

CPU 間欠動作セクタは、CPU 間欠動作モード時の、一時停止のサイクル数を選択します。

● スタンバイ制御回路

スタンバイ制御回路は、CPU クロック制御回路および周辺クロック制御回路を制御し、低消費電力モードへの移行と解除を行います。

● CPU クロック制御回路

CPU クロック制御回路は、CPU に供給するクロックを制御する回路です。

● 周辺クロック制御回路

周辺クロック制御回路は、周辺機能に供給するクロックを制御する回路です。

● 端子ハインピーダンス制御回路

端子ハインピーダンス制御回路は、タイムベースタイマモードおよびストップモード時に、外部端子をハインピーダンスにする回路です。プルアップオプションの選択された端子に対しては、ストップモード時にプルアップ抵抗を無効とします。

● 内部リセット発生回路

内部リセット発生回路は、内部リセット信号を発生させる回路です。

● 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) は、スタンバイモードへの移行 / 解除および CPU 間欠動作機能の設定などを行うレジスタです。

6.3 低消費電力モード制御レジスタ (LPMCR)

低消費電力モードの設定に使用するレジスタを説明します。

■ 低消費電力モード制御レジスタ (LPMCR)

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

図 6.3-1 低消費電力モード制御レジスタ (LPMCR) の構成

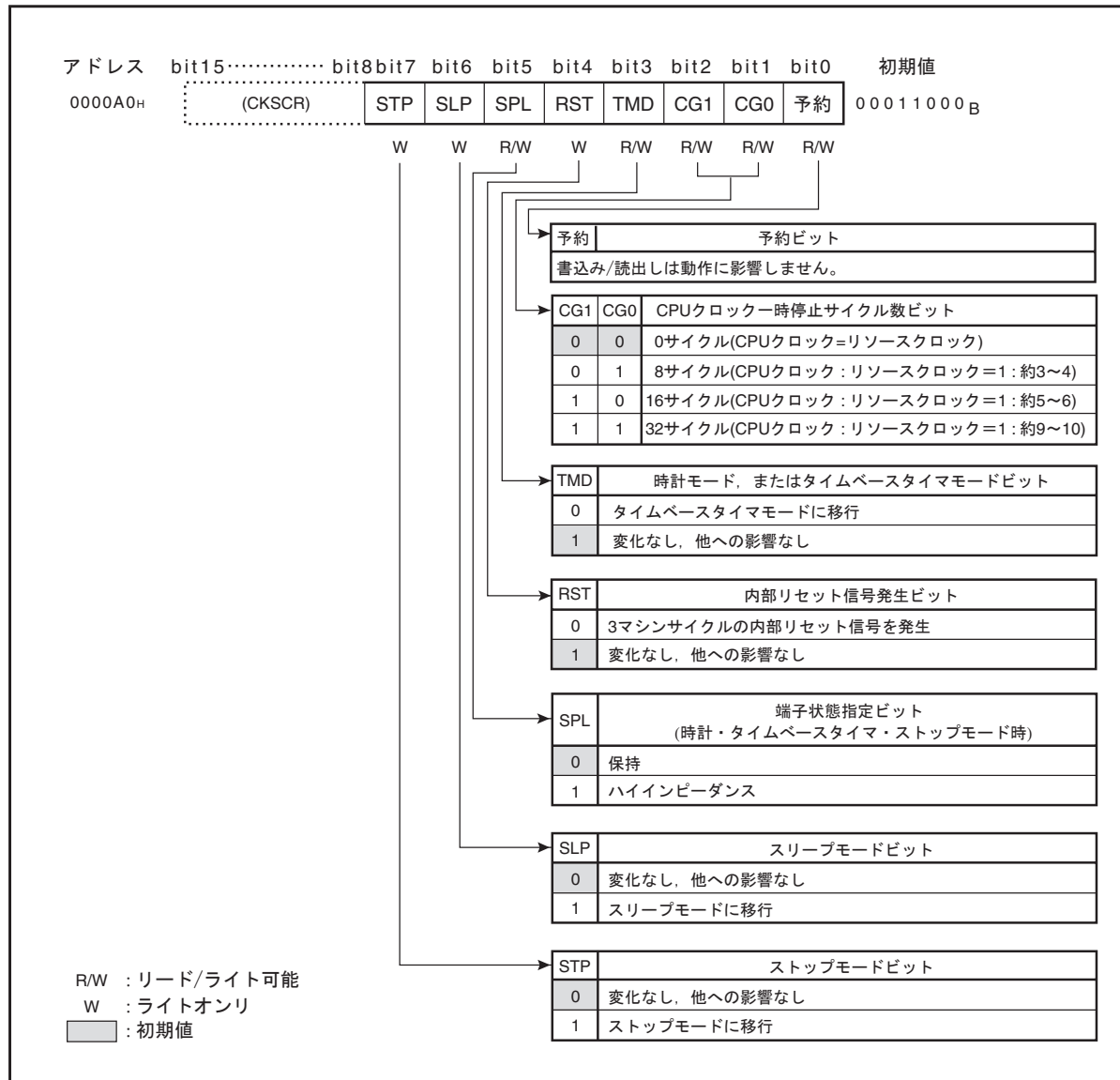


表 6.3-1 低消費電力モード制御レジスタ (LPMCR) の各ビットの機能

ビット名		機 能
bit7	STP: ストップモードビット	ストップモードへの移行を指示するビットです。 ・このビットに "1" を書き込むと、ストップモードへ移行します。 ・このビットに "0" を書き込んでも動作に影響を与えません。 ・リセット、または割込み要求発生で "0" に初期化されます。 ・このビットを読み出した場合は、常に "0" が読み出されます。
bit6	SLP: スリープモードビット	スリープモードへの移行を指示するビットです。 ・このビットに "1" を書き込むと、スリープモードに移行します。 ・このビットに "0" を書き込んでも動作に影響を与えません。 ・リセット、または割込み要求発生で "0" に初期化されます。 ・このビットを読み出した場合は、常に "0" が読み出されます。
bit5	SPL: 端子状態指定ビット (時計、タイムベースタイマ、 ストップモード時)	時計モード、タイムベースタイマモード、ストップモードに移行した場合の外部端子の状態を設定します。 ・このビットが "0" の場合、外部端子のレベルを保持します。 ・このビットが "1" の場合、外部端子をハイインピーダンスにします。 ・リセットで "0" に初期化されます。
bit4	RST: 内部リセット 信号発生ビット	ソフトウェアリセットを発生させるビットです。 ・このビットに "0" を書き込むと、3 マシンサイクルの内部リセット信号を発生します。 ・このビットに "1" を書き込んでも動作に影響を与えません。 ・このビットを読み出した場合は、常に "1" が読み出されます。
bit3	TMD: 時計、タイムベースタイマ モードビット	時計モード、タイムベースタイマモードへの移行を指示するビットです。 ・メインクロックモード、または PLL クロックモード時に、このビットに "0" を書き込むと、タイムベースタイマモードに移行します。 ・サブクロックモード時に、このビットに "0" を書き込むと、時計モードに移行します。 ・リセット、または割込み要求発生で "1" に初期化されます。 ・このビットを読み出した場合は、常に "1" が読み出されます。
bit2, bit1	CG1, CG0: CPU クロック一時停止 サイクル数選択ビット	CPU 間欠動作機能の CPU クロックの一時停止サイクル数を設定するビットです。 ・1 命令ごとに、指定サイクル数 CPU クロック供給を停止します。 ・4 種類のクロック数から選択できます。 ・リセットで "00 _B " に初期化されます。
bit0	予約: 予約ビット	書込み / 読出しは動作に影響しません。

■ 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への書込みで、スタンバイモード(ストップモード, スリープモード, タイムベースタイマモード, 時計モード)への遷移が行われます。この場合に使用する命令は、表 6.3-2 の命令を使用するようにしてください。

表 6.3-2 の命令による低消費電力モード遷移命令の直後には必ず下記 [] 内の命令列を配置してください。

```
MOV LPMCR,#H'xx ;表 6.3-2 の低消費電力モード遷移命令
[
NOP
NOP
JMP $+3 ;次の命令へのジャンプ
MOV A,#H'10 ;任意の命令
]
```

[] 内の命令列以外が配置されるとスタンバイモード解除後の動作は保証されません。

C 言語を使用して低消費電力モード制御レジスタをアクセスする場合は、"6.8 低消費電力モード使用上の注意の「スタンバイモード遷移のための低消費電力モード制御レジスタ (LPMCR) へアクセスする際の注意事項」" を参照してください。

低消費電力モード制御レジスタ (LPMCR) にワード単位で書き込む場合は、偶数アドレスで書き込んでください。奇数アドレスで書き込むと、誤動作の原因となることがあります。

表 6.3-1 に示す以外の機能を制御する場合には、どの命令を使用してもかまいません。

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	

■ STP, SLP, TMD ビットの優先順位

ストップモード要求,スリープモード要求およびタイムベースタイマモード要求を同時に行った場合は、以下の優先順位で要求を処理します。

ストップモード要求>タイムベースタイマモード要求>スリープモード要求

6.4 CPU 間欠動作モード

CPU 間欠動作モードは、外部バスや周辺機能を高速動作させたまま CPU を間欠動作させることで、消費電力を低減させます。

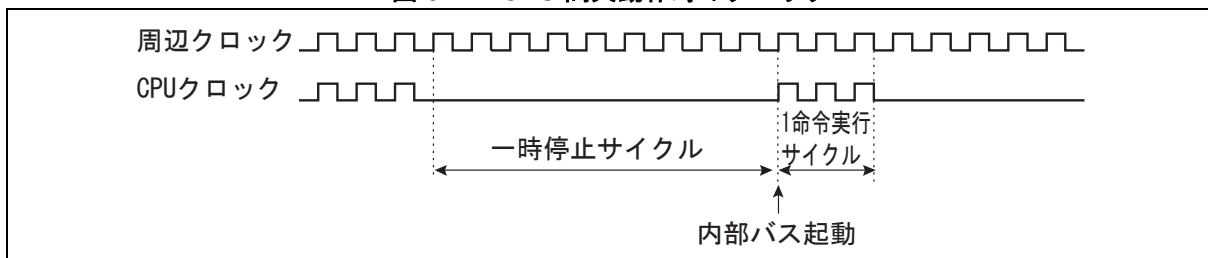
■ CPU 間欠動作モード

CPU 間欠動作モード (PLL クロック間欠動作モード、メインクロック間欠動作モード、サブクロック間欠動作モード) は、レジスタ、内蔵メモリ (ROM, RAM), I/O, 周辺機能および外部バスにアクセスを行う場合に、CPU に供給するクロックを 1 命令実行ごとに一定期間停止させて内部バスサイクルの起動を遅らせます。周辺機能に高速の周辺クロックを供給した状態で、CPU の実行速度を下げることによって低消費電力で処理が行えます。

- 低消費電力モード制御レジスタ (LPMCR) の CPU クロック一時停止サイクル数選択ビット (CG1, CG0) で、CPU に供給するクロックの一時停止サイクル数を選択します。
- 外部バス動作は、周辺機能と同じクロックを使用します。
- CPU 間欠動作モードを使用した場合の命令実行時間の算出は、レジスタ、内蔵メモリ、内蔵周辺機能および外部バスにアクセスを行う場合の命令実行回数に、一時停止サイクル数をかけた補正値を通常の実行時間に加えることで、求められます。

図 6.4-1 に CPU 間欠動作モード時の動作クロックを示します。

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

スタンバイモードは、CPU または周辺機能への動作クロックの供給の停止や、発振クロックの停止によって消費電力を低減させます。スタンバイモードには、スリープ (PLL スリープ、メインスリープ、サブスリープ)、タイムベースタイマ、時計、ストップの各モードがあります。

■ スタンバイモード時の動作状態

表 6.5-1 に、スタンバイモード時の動作状態を示します。

表 6.5-1 スタンバイモード時の動作状態

スタンバイモード		移行条件	メインクロック	サブクロック	マシンクロック	CPU	周辺	端子	解除方法					
スリープモード	PLL スリープモード	SCS=1 MCS=0 SLP=1	動作	動作	動作	停止	動作	動作	リセットまたは割込み					
	メインスリープモード	SCS=1 MCS=1 SLP=1												
	サブスリープモード	SCS=0 SLP=1	停止											
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)	SCS=1 TMD=0	動作		停止		停止	停止 *1		保持				
	タイムベースタイマモード (SPL=1)	SCS=1 TMD=0								Hi-Z				
時計モード	時計モード (SPL=0)	SCS=0 TMD=0	停止					停止		停止	停止 *2	保持		
	時計モード (SPL=1)	SCS=0 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	停止	保持
	ストップモード (SPL=1)	STP=1												Hi-Z

*1: タイムベースタイマ、時計タイマは動作します。

*2: 時計タイマは動作します。

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

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) の時計、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

SCS: クロック選択レジスタ (CKSCR) のマシクロック選択ビット (サブ)

Hi-Z: ハイインピーダンス

RST: 外部リセット端子

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 は停止し、周辺機能はスリープモード移行前のクロックで動作を続行します。

■ 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) に "1", 時計・タイムベースタイマモードビット (TMD) に "1", ストップモードビット (STP) に "0" を書き込むと、スリープモードに移行します。

<注意事項>

LPMCR レジスタの SLP ビットと STP ビットに同時に "1" を書き込むと、STP ビットを優先し、ストップモードへ移行します。

低消費電力モード制御レジスタの SLP ビットに "1", TMD ビットに "0" を同時に書き込むと、TMD ビットを優先し、タイムベースタイマモード、または時計モードに移行します。

● データ保持機能

スリープモード中は、アキュムレータなどの専用レジスタと内部 RAM の内容を保持します。

● ホールド機能

スリープモード中は、外部バスホールド機能は動作し、ホールド要求があるとホールド状態になります。

● 割込み要求発生中の動作

LPMCR レジスタの SLP ビットに "1" を書き込んだ場合に、割込み要求が発生していると、スリープモードには移行しません。そのため、CPU は割込みを受け付けない状態であれば次の命令を実行し、受け付ける状態であれば即座に割込み処理ルーチンへ分岐します。

● 端子状態

スリープモード中は、バス入出力またはバス制御として使用されている端子以外は、直前の状態を保持します。

■ スリープモードの解除

低消費電力制御回路は、リセット入力または割込みの発生によってスリープモードを解除します。

● リセットによる復帰

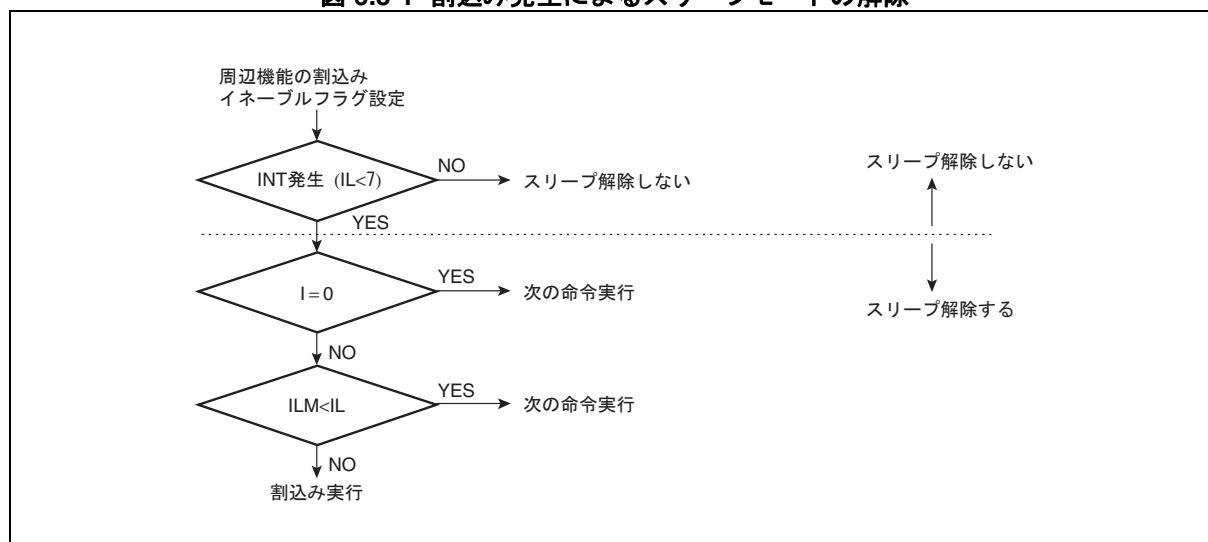
リセットによってメインクロックモードに初期化されます。

● 割込みによる復帰

スリープモード中に、周辺回路などから割込みレベルが7より高い割込み要求が発生した場合は、スリープモードが解除されます。スリープモードの解除後は、通常の割込み処理と同様にコンディションコードレジスタ (CCR) のIフラグ、インタラプトレベルマスクレジスタ (ILM) および割込み制御レジスタ (ICR) の設定によって、割込み要求の判定が行われます。割込みが受け付けられる場合は、CPUは割込み処理を実行します。割込みが受け付けられない場合は、スリープモードを指定した命令の次の命令から処理を続行します。

図 6.5-1 に、割込み発生によるスリープモードの解除を示します。

図 6.5-1 割込み発生によるスリープモードの解除



<注意事項>

割込み処理を実行する場合、通常はスリープモードを指定した命令の次の命令を実行した後に、割込み処理に移行します。ただし、スリープモードへの移行と外部バスホールド要求の受付が同時であった場合、次の命令を実行する前に割込み処理に移行することがあります。

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、タイムベースタイマおよび時計タイマだけを動作させるモードです。タイムベースタイマと時計タイマ以外の周辺機能は停止します。

■ タイムベースタイマモードへの遷移

PLL クロックモードまたはメインクロックモード時 (クロック選択レジスタ (CKSCR) のサブクロック表示ビット (SCM)=1) に、低消費電力モード制御レジスタ (LPMCR) の時計モード、タイムベースタイマモードビット (TMD) に "0" を書き込むと、タイムベースタイマモードに移行します。

● データ保持機能

タイムベースタイマモードでは、アキュムレータなどの専用レジスタと内部 RAM の内容を保持します。

● ホールド機能

タイムベースタイマモード中は、外部バスホールド機能は停止し、ホールド要求が入力されてもその要求を受け付けません。タイムベースタイマモードへの移行中にホールド要求が入力された場合、バスをハイインピーダンスにした状態で、 \overline{HAK} 信号が "L" にならないことがあります。

● 割込み要求発生中の動作

LPMCR レジスタの TMD ビットに "0" を書き込んだ場合、割込み要求が発生していても、タイムベースタイマモードに移行しません。

● 端子状態

タイムベースタイマモード中の外部端子を、直前の状態に保持するか、ハイインピーダンス状態にするかは、LPMCR レジスタの端子状態指定ビット (SPL) で制御できます。

■ タイムベースタイマモードの解除

低消費電力制御回路は、リセット入力または割込み要求の発生によりタイムベースタイマモードを解除します。

● 外部リセットによる復帰

外部リセットによってタイムベースタイマモードが解除された場合は、メインクロックモードに初期化されます。

● 割込みによる復帰

タイムベースタイマモード中に周辺回路などから割込みレベルが 7 より高い割込み要求が発生した場合は (割込み制御レジスタ (ICR) の IL2, IL1, IL0=111_B 以外の場合), 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様にコンディションコードレジスタ (CCR) の I フラグ、インタラプトレベルマスキングレジスタ (ILM) および割込み制御レジスタ (ICR) の設定により割込み要求の判定が行われます。割込みが受け付けられる場合は、割込み処理を実行します。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。

<注意事項>

割込み処理を実行する場合、通常はタイムベースタイマモードを指定した命令の次の命令を実行した後に、割込み処理に移行します。ただし、タイムベースタイマモードへの移行と外部バスホールド要求の受け付けが同時であった場合、次の命令を実行する前に割込み処理に移行することがあります。

6.5.3 時計モード

時計モードは、サブクロックと時計タイマ以外の動作を止めます。

■ 時計モードへの遷移

サブクロックモード時（クロック選択レジスタ (CKSCR) のサブクロック表示ビット (SCS)=0）、低消費電力モード制御レジスタ (LPMCR) の時計モード・タイムベースタイマモードビット (TMD) に "0" を書き込むと、時計モードに移行します。

● データ保持機能

時計モードでは、アキュムレータなどの専用レジスタと内部RAMの内容を保持します。

● ホールド機能

時計モード中は、外部バスホールド機能は停止し、ホールド要求が入力されても、その要求は受け付けません。

<注意事項>

時計モードへの遷移中に、ホールド要求が入力された場合、バスをハイインピーダンスにした状態で、 $\overline{\text{HAK}}$ 信号が "L" にならないことがあります。

● 割込み要求発生中の動作

LPMCR レジスタの TMD ビットに "0" を書き込んだ場合、割込み要求が発生していても時計モードに移行しません。

● 端子状態の設定

時計モード中の外部端子を、直前の状態に保持するか、ハイインピーダンス状態にするかは、LPMCR レジスタの端子状態指定ビット (SPL) で制御できます。

■ 時計モードの解除

低消費電力制御回路は、リセット入力または割込み要求の発生により時計モードを解除します。

● リセットによる復帰

リセット要因による時計モードの解除の場合は、時計モードを解除した上で、発振安定待ちリセット状態になります。発振安定待ち時間経過後、リセットシーケンスは実行されます。

● 割込みによる復帰

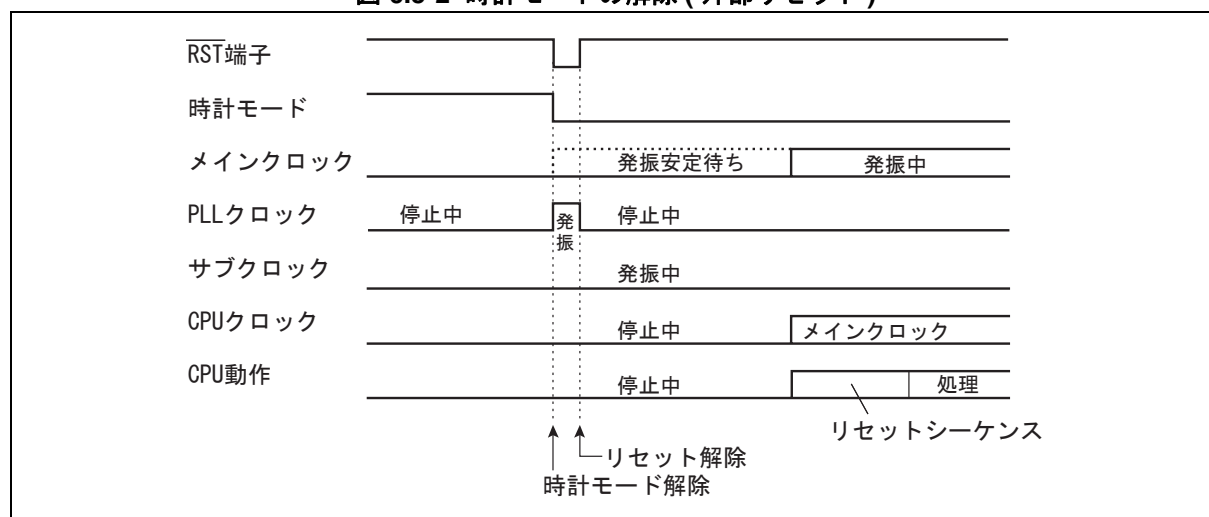
時計モード中に周辺回路などから割込みレベルが 7 より高い割込み要求が発生した場合は、(割込み制御レジスタ (ICR) の IL2, IL1, IL0 が "111_B" 以外)、低消費電力制御回路は時計モードを解除し、即座にサブクロックモードに移行します。サブクロックモードに移行後は、通常の割込み処理と同様にコンディションコードレジスタ (CCR) の I フラグ、インタラプトレベルマスクレジスタ (ILM)、および割込み制御レジスタ (ICR) の設定により割込み要求の判定が行われます。割込みが受け付けられる場合は、割込み処理を実行します。割込みが受け付けられない場合は、時計モードに入る前の次の命令から処理を続行します。

<注意事項>

割込み処理を実行する場合、通常は時計モードを指定した命令の次の命令を実行した後に、割込み処理に移行します。ただし、時計モードへの移行と外部バスホールド要求の受け付けが同時であった場合、次の命令を実行する前に割込み処理に移行することがあります。

図 6.5-2 に、時計モードの解除動作を示します。

図 6.5-2 時計モードの解除 (外部リセット)



6.5.4 ストップモード

ストップモードは、原発振を停止させることにより、すべての機能が停止します。最も低消費電力でデータを保持します。

■ 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) に "1" を書き込むと、ストップモードに移行します。

● データ保持機能

ストップモード中は、アキュムレータなどの専用レジスタと内部 RAM の内容を保持します。

● ホールド機能

ストップモード中は、外部バスホールド機能は停止し、ホールド要求が入力されても、その要求を受け付けません。

ストップモードへの遷移中に、ホールド要求が入力された場合、バスをハイインピーダンスにした状態で、 $\overline{\text{HAK}}$ 信号が "L" にならないことがあります。

● 割込み要求発生中の動作

LPMCR レジスタの STP ビットに "1" を書き込んだ場合に、割込み要求が発生していてもストップモードに移行しません。

● 端子状態の設定

ストップモード中の外部端子を、直前の状態に保持するか、ハイインピーダンス状態にするかは、LPMCR レジスタの端子状態指定ビット (SPL) で指定できます。

■ ストップモードの解除

低消費電力制御回路は、リセット入力または割込みの発生によりストップモードを解除します。ストップモードからの復帰の場合、発振クロック (HCLK) およびサブクロック (SCLK) が停止しているため、メインクロック発振安定待ち時間またはサブクロック発振安定待ち時間を経て、ストップモードが解除されます。

● リセットによる復帰

リセット要因によるストップモード解除の場合は、ストップモードを解除した上で、発振安定待ちリセット状態になります。発振安定待ち時間経過後、リセットシーケンスは実行されます。

● 割込みによる復帰

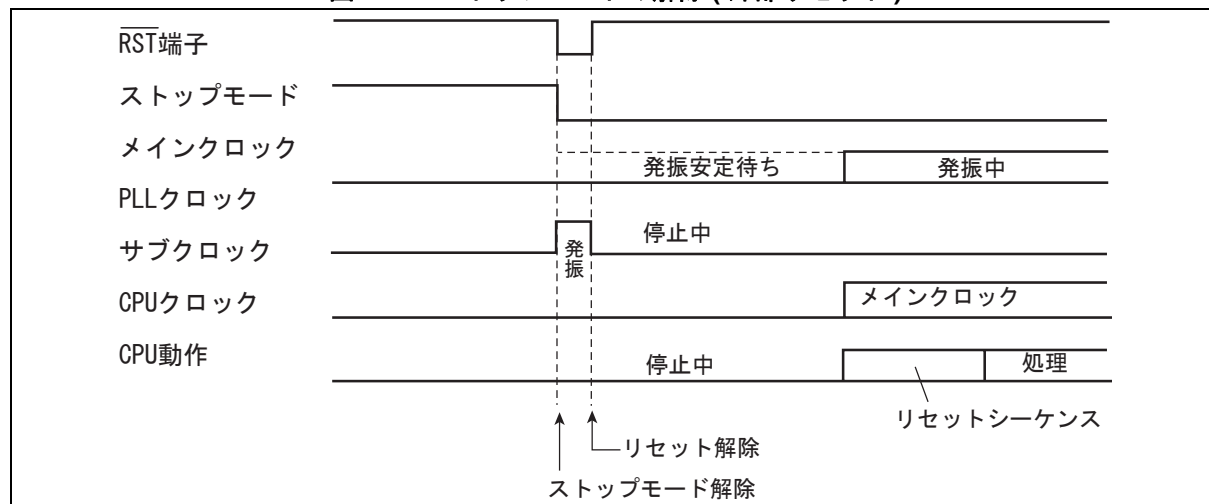
ストップモード中に周辺回路などから割込みレベルが 7 より高い割込み要求が発生した場合 (割込み制御レジスタ (ICR) の IL2, IL1, IL0=111_B 以外の場合)、低消費電力制御回路はストップモードを解除します。ストップモードの解除後は、クロック選択レジスタ (CKSCR) の発振安定待ち時間選択ビット (WS1, WS0) で指定されたメインクロックの発振安定待ち時間が経過した後で、通常の割込み処理と同様にコンディションコードレジスタ (CCR) の I フラグ、インタラプトレベルマスキングレジスタ (ILM) および割込み制御レジスタ (ICR) の設定により割込み要求の判定が行われます。割込みが受け付けられる場合は、割込み処理を実行します。割込みが受け付けられない場合は、ストップモードに入る前の次の命令から処理を続行します。

<注意事項>

- 割込み処理を実行する場合、通常はストップモードを指定した命令の次の命令を実行した後に、割込み処理に移行します。ただし、ストップモードへの移行と外部バスホールド要求の受付けが同時であった場合、次の命令を実行する前に割込み処理に移行することがあります。
- PLL ストップモード中は、メインクロックおよび PLL 通倍回路が停止しているため、PLL ストップモードから復帰する場合は、メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を確保する必要があります。この場合の発振安定待ち時間は、クロック選択レジスタの発振安定待ち時間選択ビット (CKSCR:WS1, WS0) に設定された値に従い、メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を同時にカウントしますので、CKSCR:WS1, WS0 ビットには、発振安定待ち時間の長いほうに合わせて値を設定してください。ただし、PLL クロック発振安定待ち時間は $2^{14}/\text{HCLK}$ 以上必要ですので、CKSCR:WS1, WS0 ビットには "10_B" または "11_B" を設定してください。

図 6.5-3 に、ストップモードの解除動作を示します。

図 6.5-3 ストップモードの解除 (外部リセット)



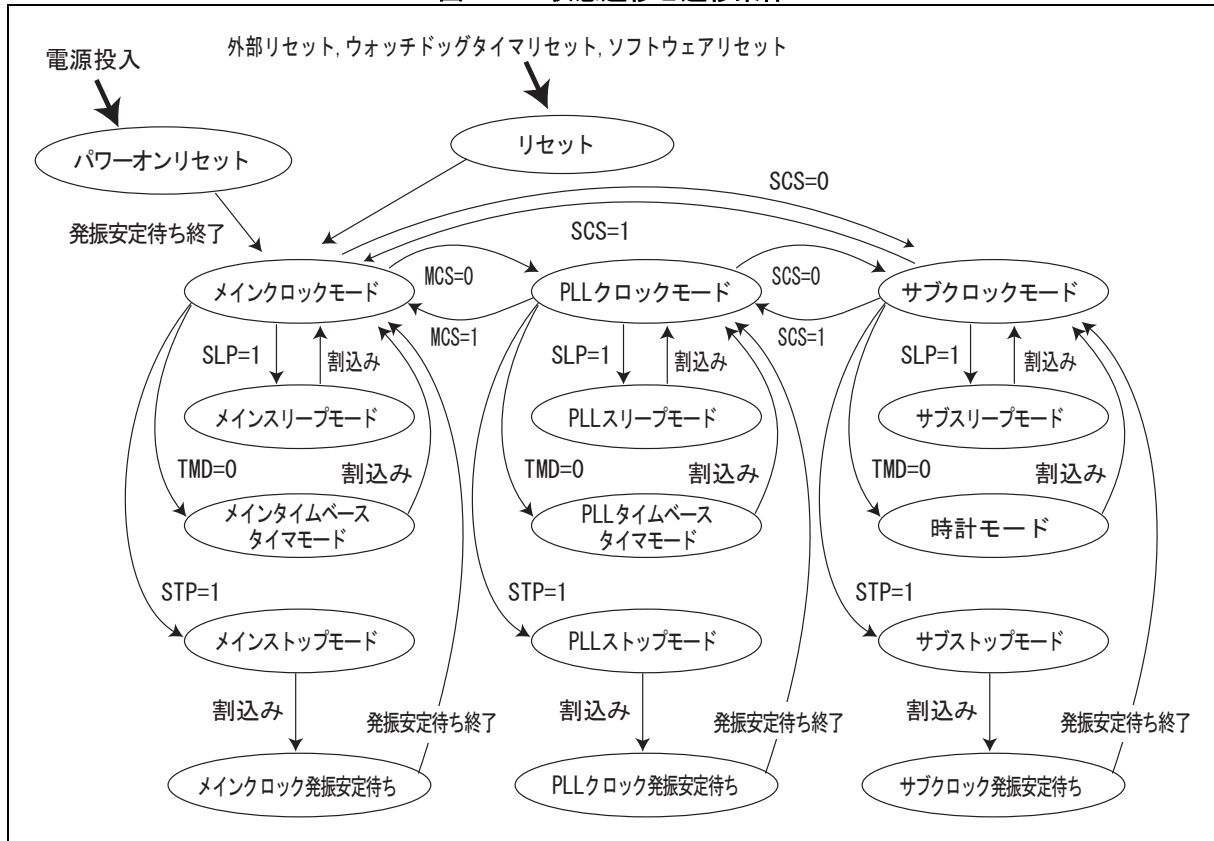
6.6 スタンバイモードの状態遷移図

MB90480/485 シリーズの動作状態の遷移と、遷移条件を示します。

■ 状態遷移図

図 6.6-1 に、MB90480/485 シリーズの動作状態の遷移と、遷移条件を示します。

図 6.6-1 状態遷移と遷移条件



■ 低消費電力モードの動作状態

表 6.6-1 に、低消費電力モードの動作状態を示します。

表 6.6-1 低消費電力モードの動作状態

動作状態	メイン クロック	サブ クロック	PLL クロック	CPU	周辺	時計	タイム ベース タイマ	クロック ソース
PLL クロックモード	動作	動作	動作	動作	動作	動作	動作	PLL クロック
PLL スリープモード				停止				
PLL タイムベースタイマモード					停止	停止	停止	
PLL ストップモード						停止	動作	
PLL クロック発振安定待ち	動作	動作	動作					
メインクロックモード	動作	動作	停止	動作	動作	動作	動作	メイン クロック
メインスリープモード				停止				
メインタイムベースタイマモード					停止	停止	停止	
メインストップモード	停止	動作				動作		
メインクロック発振安定待ち	動作	動作						
サブクロックモード	停止	動作	停止	動作	動作	動作	停止	サブ クロック
サブスリープモード				停止				
時計モード		停止			停止	停止		
サブストップモード					動作	動作		
サブクロック発振安定待ち								
パワーオンリセット	動作	動作	停止	停止	停止	動作	動作	メイン クロック
リセット			動作					

6.7 スタンバイモード, ホールド, リセット時の端子状態

スタンバイモード, ホールド, リセット時の端子の状態を, メモリアクセスモードごとに示します。

■ シングルチップモード時の端子状態

表 6.7-1 に, シングルチップモード時の端子状態を示します。

表 6.7-1 シングルチップモード時の端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット時
		SPL=0	SPL=1		
P07 ~ P00	直前の状態を保持 ^{*1}	入力遮断 / 直前の状態を保持 ^{*1, *2}	入力遮断 / 出力 Hi-Z ^{*2}	本状態はない	入力不可 / 出力 Hi-Z
P17 ~ P10					
P27 ~ P20					
P37 ~ P30					
P47 ~ P40					
P57 ~ P50					
P67 ~ P60					
P77 ~ P70					
P97 ~ P90					
PA3 ~ PA0	入力可能 ^{*3}	入力可能 ^{*3}	入力可能 ^{*3}	本状態はない	入力不可
P87 ~ P80					

*1: 本モードになる直前に出力していた状態をそのまま出力します。または, 入力であれば入力不可を意味します。

入力不可とは, 端子の入力ゲートの動作は許可状態ですが, 内部回路が動作していないので, 端子の内容が内部で受け付けられない状態を意味します。

*2: 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。出力 Hi-Z 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。

*3: 出力状態として使用している場合には, ほかのポートと同じです。入力可能とは, 入力機能が使用可能な状態であることを意味するので, プルアップ/プルダウン, または外部からの入力が必要です。

■ 外部バス 16 ビットデータバスモードおよびマルチプレクス 16 ビット外部バスモードの端子状態

表 6.7-2 に、外部バス 16 ビットデータバスモードおよびマルチプレクス 16 ビット外部バスモードの端子状態を示します。

表 6.7-2 外部バス 16 ビットデータバスモードおよびマルチプレクス 16 ビット外部バスモードの端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット時	リセット解除直後の内 ROM アクセス時	外 ROM アクセス後に内 ROM アクセス時
		SPL=0	SPL=1				
P07 ~ P00 (AD07 ~ AD00)	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z *5	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P17 ~ P10 (AD15 ~ AD08)							
P27 ~ P20 (A23 ~ A16)	出力状態 *1, *2	出力状態 *1, *2		入力不可 / 出力 Hi-Z *2	出力状態 *1	出力状態 *1	直前のアドレスを保持
P57(CLK)	入力不可 / 出力可能 *2, *3	入力不可 / 出力状態 *1, *2		入力不可 / 出力可能 *2, *3	入力不可 / 出力可能 *3	CLK 出力	CLK 出力
P56(RDY)	直前の状態 を保持 *4	入力遮断 / 直前 の状態を保持 *4, *5		入力不可 *2	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P55(HAK)				"L" 出力			
P54(HRQ)				"H" 入力			
P53(WRH)	"H" 出力 *2	"H" 出力 *2		入力不可 / 出力 Hi-Z *2	"H" 出力	"H" 出力	"H" 出力
P52(WRL)							
P51(RD)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z	出力可能 *2	"L" 出力	"L" 出力
P50(ALE)	"L" 出力	"L" 出力					
P37 ~ P30	直前の状態 を保持 *4	入力遮断 / 直前 の状態を保持 *4		直前の状態 を保持 *4	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P47 ~ P40							
P67 ~ P60							
P77 ~ P70							
P97 ~ P91							
PA3 ~ PA0							
P90(CS0)			"H" 出力				
P87 ~ P80	入力可能 *6	入力可能 *6	入力可能 *6	入力不可			

*1: 出力状態とは、端子駆動用トランジスタを駆動許可状態ですが、内部回路の動作が停止状態なので、“H” ないし “L” の固定値を出力することを意味します。内部周辺回路が動作中で、出力機能を使用している場合には、リセット時を除いて出力が変化します。リセット時の出力変化はありません。

*2: 出力ポートとして使用している場合は、直前の値を保持します。

*3: 出力可能とは、端子駆動用トランジスタを駆動状態で、内部回路の動作が許可状態なので、動作内容が端子に現れることを意味します。

*4: 本モードになる直前に出力していた状態をそのまま出力します。または、入力であれば入力不可を意味します。

入力不可とは、端子の入力ゲートの動作は許可状態ですが、内部回路が動作していないので、端子の内容が内部で受け付けられない状態を意味します。

*5: 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。出力 Hi-Z 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。

*6: 出力状態として使用している場合には、ほかのポートと同じです。入力可能とは、入力機能が使用可能な状態であることを意味するので、プルアップ/プルダウン、または外部からの入力が必要です。

■ 外部バス8ビットデータバスモードおよびマルチプレクス8ビット外部バスモードの端子状態

表 6.7-3 に、外部バス8ビットデータバスモードおよびマルチプレクス8ビット外部バスモードの端子状態を示します。

表 6.7-3 外部バス8ビットデータバスモードおよびマルチプレクス8ビット外部バスモードの端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット時	リセット解除直後の内 ROM アクセス時	外 ROM アクセス後に内 ROM アクセス時
		SPL=0	SPL=1				
P07 ～ P00 (AD07 ～ AD00)	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z *5	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P17 ～ P10 (AD15 ～ AD08)	出力状態 *1	出力状態 *1			出力状態 *1	出力状態 *1	出力状態 *1
P27 ～ P20 (A23 ～ A16)	出力状態 *1, *2	出力状態 *1, *2					
P57(CLK)	入力不可 / 出力可能 *2, *3	入力不可 / 出力状態 *1, *2		入力不可 / 出力可能 *2, *3	入力不可 / 出力可能 *3	CLK 出力	CLK 出力
P56(RDY)	直前の状態を保持 *4	入力遮断 / 直前の状態を保持 *4		入力不可 *2	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P55($\overline{\text{HAK}}$)				"L" 出力			
P54(HRQ)				"H" 入力			
P53($\overline{\text{WRH}}$)				直前の状態を保持 *4			
P52($\overline{\text{WRL}}$)	"H" 出力 *4	"H" 出力 *4		入力不可 / 出力 Hi-Z *2	"H" 出力	"H" 出力	"H" 出力
P51($\overline{\text{RD}}$)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z	"L" 出力	"L" 出力	"L" 出力
P50(ALE)	"L" 出力	"L" 出力					
P37 ～ P30	直前の状態を保持 *4	入力遮断 / 直前の状態を保持 *4		直前の状態を保持 *4	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P47 ～ P40							
P67 ～ P60							
P77 ～ P70							
P97 ～ P91							
PA3 ～ PA0							
P90(CS0)							
P87 ～ P80	入力可能 *6	入力可能 *6	入力可能 *6	入力不可			

*1: 出力状態とは、端子駆動用トランジスタを駆動許可状態ですが、内部回路の動作が停止状態なので、"H" ないし "L" の固定値を出力することを意味します。内部周辺回路が動作中で、出力機能を使用している場合には、リセット時を除いて出力が変化します。リセット時の出力変化はありません。

*2: 出力ポートとして使用している場合は、直前の値を保持します。

*3: 出力可能とは、端子駆動用トランジスタを駆動状態で、内部回路の動作が許可状態なので、動作内容が端子に現れることを意味します。

*4: 本モードになる直前に出力していた状態をそのまま出力します。または、入力であれば入力不可を意味します。

入力不可とは、端子の入力ゲートの動作は許可状態ですが、内部回路が動作していないので、端子の内容が内部で受け付けられない状態を意味します。

*5: 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。出力 Hi-Z 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。

*6: 出力状態として使用している場合には、ほかのポートと同じです。入力可能とは、入力機能が使用可能な状態であることを意味するので、プルアップ/プルダウン、または外部からの入力が必要です。

■ 外部バス 16 ビットデータバスモードおよびノンマルチプレクス 16 ビット外部バスモードの端子状態

表 6.7-4 に、外部バス 16 ビットデータバスモードおよびノンマルチプレクス 16 ビット外部バスモードの端子状態を示します。

表 6.7-4 外部バス 16 ビットデータバスモードおよびノンマルチプレクス 16 ビット外部バスモードの端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット時	リセット解除直後の内 ROM アクセス時	外 ROM アクセス後に内 ROM アクセス時
		SPL=0	SPL=1				
P07 ~ P00 (AD07 ~ AD00)	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z *5	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P17 ~ P10 (AD15 ~ AD08)					出力状態 *1	出力状態 *1	出力状態 *1
P37 ~ P30 (A07 ~ A00)							
P47 ~ P40 (A15 ~ A08)							
P27 ~ P20 (A23 ~ A16)	出力状態 *1, *2	出力状態 *1, *2		入力不可 / 出力 Hi-Z *2	入力不可 / 出力可能 *3	CLK 出力	CLK 出力
P57(CLK)	入力不可 / 出力可能 *2, *3	入力不可 / 出力状態 *1, *2		入力不可 / 出力可能 *2, *3			
P56(RDY)	直前の状態 を保持 *4	入力遮断 / 直前 の状態を保持 *4		入力不可 *2			
P55($\overline{\text{HAK}}$)				"L" 出力			
P54(HRQ)				"H" 入力			
P53($\overline{\text{WRH}}$)	"H" 出力 *4	"H" 出力 *4		入力不可 / 出力 Hi-Z *2	"H" 出力	"H" 出力	"H" 出力
P52($\overline{\text{WRL}}$)							
P51($\overline{\text{RD}}$)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z	出力可能 *3	"L" 出力	"L" 出力
P50(ALE)	"L" 出力	"L" 出力					
P67 ~ P60	直前の状態 を保持 *4	入力遮断 / 直前 の状態を保持 *4		直前の状態 を保持 *4	入力不可 / 出力 Hi-Z	出力 Hi-Z/ 入力可能	出力 Hi-Z/ 入力可能
P77 ~ P70							
P97 ~ P91							
PA3 ~ PA0							
P90(CS0)			出力 "H" 出力				
P87 ~ P80	入力可能 *6	入力可能 *6		入力可能 *6	入力不可		

*1: 出力状態とは、端子駆動用トランジスタを駆動許可状態ですが、内部回路の動作が停止状態なので、"H" ないし "L" の固定値を出力することを意味します。内部周辺回路が動作中で、出力機能を使用している場合には、リセット時を除いて出力が変化します。リセット時の出力変化はありません。

*2: 出力ポートとして使用している場合は、直前の値を保持します。

*3: 出力可能とは、端子駆動用トランジスタを駆動状態で、内部回路の動作が許可状態なので、動作内容が端子に現れることを意味します。

*4: 本モードになる直前に出力していた状態をそのまま出力します。または、入力であれば入力不可を意味します。

入力不可とは、端子の入力ゲートの動作は許可状態ですが、内部回路が動作していないので、端子の内容が内部で受け付けられない状態を意味します。

*5: 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。出力 Hi-Z 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。

*6: 出力状態として使用している場合には、ほかのポートと同じです。入力可能とは、入力機能が使用可能な状態であることを意味するので、プルアップ/プルダウン、または外部からの入力が必要です。

■ 外部バス 8 ビットデータバスモードおよびノンマルチプレクス 8 ビット外部バスモードの端子状態

表 6.7-5 に、外部バス 8 ビットデータバスモードおよびノンマルチプレクス 8 ビット外部バスモードの端子状態を示します。

表 6.7-5 外部バス 8 ビットデータバスモードおよびノンマルチプレクス 8 ビット外部バスモード

端子名	スリープ時	ストップ時		ホールド時	リセット時	リセット解除直後の内 ROM アクセス時	外 ROM アクセス後に内 ROM アクセス時		
		SPL=0	SPL=1						
P07 ~ P00 (AD07 ~ AD00)	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z *5	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	出力 Hi-Z / 入力可能	出力 Hi-Z / 入力可能		
P37 ~ P30 (A07 ~ A00)	出力状態 *1	出力状態 *1			出力状態 *1	出力状態 *1	直前のアドレスを保持		
P47 ~ P40 (A15 ~ A08)									
P27 ~ P20 (A23 ~ A16)	出力状態 *1, *2	出力状態 *1, *2		入力不可 / 出力 Hi-Z *2	入力不可 / 出力可能 *3	CLK 出力	CLK 出力		
P57(CLK)	入力不可 / 出力可能 *2, *3	入力不可 / 出力状態 *1, *2		入力不可 / 出力可能 *2, *3					
P56(RDY)	直前の状態を保持 *4	入力遮断 / 直前の状態を保持 *4		入力不可 *2					
P55(HAK)				"L" 出力					
P54(HRQ)				"H" 入力					
P53				直前の状態を保持 *4					
P52(WRL)	"H" 出力 *2	"H" 出力 *2		入力不可 / 出力 Hi-Z *2	"H" 出力	"H" 出力	"H" 出力		
P51(RD)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z					
P50(ALE)	"L" 出力	"L" 出力		出力可能 *3	"L" 出力	"L" 出力	"L" 出力		
P17 ~ P10	直前の状態を保持 *4	入力遮断 / 直前の状態を保持 *4		直前の状態を保持 *4	入力不可 / 出力 Hi-Z	出力 Hi-Z / 入力可能	出力 Hi-Z / 入力可能		
P67 ~ P60									
P77 ~ P70					"H" 出力				
P97 ~ P91									
PA3 ~ PA0									
P90(CS0)			出力可能 *6		出力可能 *6			出力可能 *6	
P87 ~ P80	入力不可								

*1: 出力状態とは、端子駆動用トランジスタを駆動許可状態ですが、内部回路の動作が停止状態なので、"H" ないし "L" の固定値を出力することを意味します。内部周辺回路が動作中で、出力機能を使用している場合には、リセット時を除いて出力が変化します。リセット時の出力変化はありません。

*2: 出力ポートとして使用している場合は、直前の値を保持します。

*3: 出力可能とは、端子駆動用トランジスタを駆動状態で、内部回路の動作が許可状態なので、動作内容が端子に現れることを意味します。

*4: 本モードになる直前に出力していた状態をそのまま出力します。または、入力であれば入力不可を意味します。

入力不可とは、端子の入力ゲートの動作は許可状態ですが、内部回路が動作していないので、端子の内容が内部で受け付けられない状態を意味します。

*5: 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。出力 Hi-Z 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。

*6: 出力状態として使用している場合には、ほかのポートと同じです。入力可能とは、入力機能が使用可能な状態であることを意味するので、プルアップ/プルダウン、または外部からの入力が必要です。

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際には、以下の点にご注意ください。

- スタンバイモードへの移行と割込み
 - スタンバイモードの割込みによる解除
 - ストップモード解除時
 - 発振安定待ち時間
 - スタンバイモード遷移のための低消費電力モード制御レジスタ (LPMCR) へアクセスする際の注意事項
-

■ スタンバイモードへの移行と割込み

周辺機能から CPU に対して割込み要求が発生している場合は、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) とスリープモードビット (SLP) に "1", または時計モード・タイムベースタイマモードビット (TMD) に "0" を設定しても無視されるため、各スタンバイモードへは移行しません (割込み処理後にも、スタンバイモードへの移行はしません)。

CPU が割込み処理中の場合は、その割込み要求フラグビットがクリアされています。ほかに割込み要求がなければ、スタンバイモードへ移行できます。

■ スタンバイモードの割込みによる解除

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、周辺機能などから割込みレベルが 7 より高い割込み要求が発生した場合、スタンバイモードは解除されます (CPU が割込みを受け付けなくてもモードは解除されます)。

割込みによるスタンバイモードの解除後は、通常の割込み動作として、割込み要求に対応する割込みレベル設定ビット (ICR レジスタの IL2, IL1, IL0 ビット) の優先度がインタラプトレベルマスキングレジスタ (ILM) より高く、コンディションコードレジスタ (CCR) の I フラグによって割込みが許可 (I=1) されている場合は、割込み処理ルーチンへ分岐します。割込みが受け付けられない場合は、スタンバイモードを指定した命令の次の命令から動作を再開します。

割込み処理を実行する場合、通常はスタンバイモードを指定した命令の次の命令を実行した後に、割込み処理に移行します。

ただし、スタンバイモードに移行した場合の条件によっては、次の命令を実行する前に割込み処理に移行することがあります。

スタンバイモード復帰直後に割込み処理ルーチンへ分岐させたくない場合は、スタンバイモード設定の前に割込み禁止をするなどの対策が必要です。

■ 発振安定待ち時間

● 発振クロック発振安定待ち時間

ストップモード中は、原発振用の発振器が停止しているため、発振安定待ち時間を確保する必要があります。発振安定待ち時間を、クロック選択レジスタ (CKSCR) の発振安定待ち時間選択ビット (WS1, WS0) で設定します。

CKSCR レジスタの発振安定待ち時間選択ビット (WS1, WS0) に "00_B" を設定する場合は、メインクロック時のみにしてください。

● PLL クロック発振安定待ち時間

メインクロックモード中は、PLL 通倍回路が停止しているため、PLL クロックモードに移行する場合は、PLL クロック発振安定待ち時間を確保する必要があります。PLL クロック発振安定待ち時間中はメインクロックで動作します。

メインクロックモードから PLL クロックモードに切り換えた場合の PLL クロック発振安定待ち時間は、 $2^{14}/\text{HCLK}$ (HCLK: 発振クロック) に固定されています。

サブクロックモード中は、メインクロックおよび PLL 通倍回路が停止しているため、PLL クロックモードに移行する場合は、メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を確保する必要があります。この場合の発振安定待ち時間は、クロック選択レジスタの発振安定待ち時間選択ビット (CKSCR: WS1, WS0) に設定された値に従います。メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を同時にカウントしますので、CKSCR: WS1, WS0 ビットには、発振安定待ち時間の長いほうに合わせて値を設定してください。ただし、PLL クロック発振安定待ち時間は $2^{14}/\text{HCLK}$ 以上必要ですので、CKSCR: WS1, WS0 ビットには "10_B" または "11_B" を設定してください。

PLL ストップモード中は、メインクロックおよび PLL 通倍回路が停止しているため、PLL ストップモードから復帰する場合は、メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を確保する必要があります。この場合の発振安定待ち時間は、クロック選択レジスタの発振安定待ち時間選択ビット (CKSCR: WS1, WS0) に設定された値に従います。メインクロック発振安定待ち時間および PLL クロック発振安定待ち時間を同時にカウントしますので、CKSCR: WS1, WS0 ビットには、発振安定待ち時間の長いほうに合わせて値を設定してください。ただし、PLL クロック発振安定待ち時間は $2^{14}/\text{HCLK}$ 以上必要ですので、CKSCR: WS1, WS0 ビットには "10_B" または "11_B" を設定してください。

■ 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、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しないでください。切換えの完了はクロック選択レジスタ (CKSCR) の MCM ビットおよび SCM ビットを確認してください。切換えが完了する前に、他のクロックモードおよび低消費電力モードへ切り換えた場合、切り換わらない場合があります。

■ スタンバイモード遷移のための低消費電力モード制御レジスタ (LPMCR) へアクセスする際の注意事項

● アセンブラ言語を使用して低消費電力モード制御レジスタ (LPMCR) にアクセスする場合

低消費電力モード制御レジスタ (LPMCR) においてスタンバイモードに移行する設定を行う場合は、表 6.3-2 の命令を使用してください。

表 6.3-2 の命令による低消費電力モード遷移命令の直後には必ず下記 [] 内の命令列を配置してください。

```
MOV LPMCR,#H'xx      ;表 6.3-2 の低消費電力モード遷移命令
[
NOP
NOP
JMP $+3                ;次の命令へのジャンプ
MOV A,#H'10            ;任意の命令
]
```

[] 内の命令列以外が配置されるとスタンバイモード解除後の動作は保証されません。

● C 言語を使用して低消費電力モード制御レジスタ (LPMCR) にアクセスする場合

低消費電力モード制御レジスタ (LPMCR) においてスタンバイモードに移行する設定を行う場合は、以下の (1) から (3) のいずれかの方法でアクセスしてください。

- (1) スタンバイモードに遷移させる命令を関数化し、スタンバイモード遷移命令の後に `__wait_nop()` のビルトイン関数を 2 個挿入してください。関数内で、スタンバイ復帰の割込み以外の割込みが発生する可能性がある場合は、コンパイル時に最適化を実施し、LINK/UNLINK 命令の発生を抑止してください。

例 (時計モードまたはタイムベースタイマモード遷移関数の場合)

```
[
void enter_watch(){
    IO_LPMCR.byte = 0x10; /* LPMCR の TMD ビットに 0 をセット */
    __wait_nop();
    __wait_nop();
}
]
```

- (2) スタンバイモードに遷移させる命令を `__asm` 文で記述し、スタンバイモード遷移命令の後に 2 個の NOP と JMP 命令を挿入してください。

例 (スリープモード遷移の場合)

```
[
__asm(" MOV I:_IO_LPMCR, #H'58"); /* LPMCR の SLP ビットに 1 をセット */
__asm(" NOP");
__asm(" NOP");
__asm(" JMP $+3"); /* 次の命令へのジャンプ */
]
```

- (3) スタンバイモードに遷移させる命令を `#pragma asm ~ #pragma endasm` 間に記述し、スタンバイモード遷移命令の後に 2 個の NOP と JMP 命令を挿入してください。

例 (ストップモード遷移の場合)

```
[
#pragma asm
MOV I:_IO_LPMCR, #H'98 /* LPMCR の STP ビットに 1 をセット */
NOP
NOP
JMP $+3                /* 次の命令へのジャンプ */
#pragma endasm
]
```

第7章

モード設定

モード設定，モード端子，モードデータ，外部メモリアクセスおよび動作について説明します。

- 7.1 モード設定
- 7.2 モード端子 (MD2 ～ MD0)
- 7.3 モードデータ
- 7.4 外部メモリアクセス
- 7.5 モード設定の各モードにおける動作

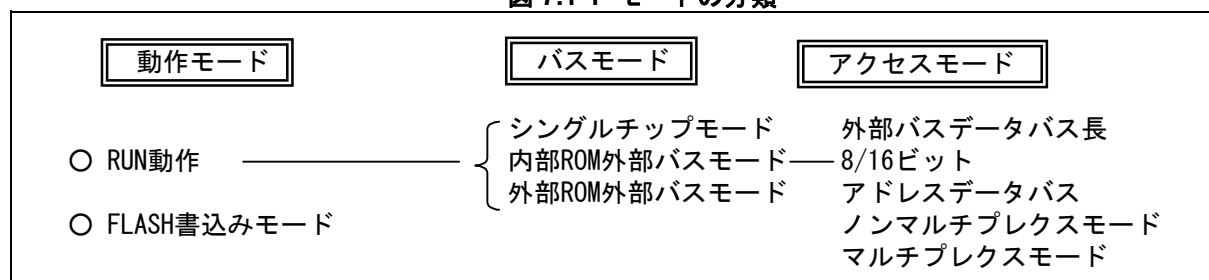
7.1 モード設定

F²MC-16LX には、アクセス方式、アクセス領域について各種のモードがあります。リセット時のモード端子の設定と、モードフェッチされたモードデータによって各モードが設定されます。

■ モード分類

F²MC-16LX では、アクセス方式、アクセス領域について各種のモードがあり、本モジュールでは図 7.1-1 に示す分類になっています。

図 7.1-1 モードの分類



■ 動作モード

動作モードは、デバイスの動作状態を制御します。モード設定用端子 (MDx) とモードデータ内の Mx ビットの内容で指定します。動作モードの選択により、通常動作の起動やフラッシュメモリへの書き込みを行うことができます。

■ バスモード

バスモードは、内部 ROM の動作と外部アクセス機能の動作を制御します。モード設定用端子 (MDx) とモードデータ内の Mx ビットの内容で指定します。モード設定用端子 (MDx) は、リセットベクタおよびモードデータを読み出すときのバスモードを指定します。モードデータ内の Mx ビットは、通常動作時のバスモードを指定します。

■ アクセスモード

アクセスモードは、外部データバス幅を制御します。モード設定用端子 (MDx) とモードデータ内の Sx ビットで指定します。アクセスモードの選択により、外部データバスを 8 ビット長、または 16 ビット長のいずれかを指定します。また、アドレスデータバスのノンマルチプレクスモード、またはマルチプレクスモードのいずれかを指定します。

7.2 モード端子 (MD2 ~ MD0)

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

■ モード端子 (MD2 ~ MD0) の設定

モード端子 (MD2 ~ MD0) では、リセットベクタの読出しを、外部データバスとするか内部データバスとするかの選択および外部データバス選択時のバス幅選択を行います。フラッシュ ROM 内蔵品の場合は、内蔵 ROM にプログラムを書き込むための、フラッシュ ROM 書込みモードの指定もモード端子で行います。

モード端子の設定内容を、表 7.2-1 に示します。

表 7.2-1 モード端子の設定内容

P81	P80	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	外部ベクタモード0	外部	マルチプレクスモード	リセットベクタ 8ビットバス幅アクセス
-	-	0	0	1	外部ベクタモード1	外部	マルチプレクスモード	リセットベクタ 16ビットバス幅アクセス
-	-	0	1	0	外部ベクタモード2	外部	ノンマルチプレクスモード	リセットベクタ 8ビットバス幅アクセス
-	-	0	1	1	内部ベクタモード	内部	モードデータ	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	0	1	1	0	フラッシュシリアル書込み			
-	-	1	1	1	フラッシュライタ書込みモード	-	-	-

(注意事項) MD2 ~ MD0:0= V_{SS} , または 1= V_{CC} としてください。外部ベクタモード2は、データバス幅も8ビットが初期値です。データバス幅を16ビットにする場合は、モードデータをノンマルチプレクス外部データバス16ビットモードに設定するとIOBS, LMBSの領域は16ビットサイズアクセスとなります。HMBSの領域を16ビットサイズアクセスにしたいときは、HMBSを設定し直してください。

7.3 モードデータ

モードデータは, "FFFFDF_H" 番地のメモリ上にあり, リセットシーケンス後の動作を指定します。モードデータは, モードフェッチで CPU に自動的に取り込まれます。

■ モードデータ

モードデータは, "FFFFDF_H" 番地に存在し, リセットシーケンス実行中に CPU コア内のモードレジスタに取り込まれます。CPU は, このモードデータを基にしてメモリアクセスモードを設定します。モードレジスタの内容を変更できるのは, リセットシーケンスだけです。また, モードデータの設定は, リセットシーケンス以降に有効となります。

モードデータの構成を, 下図に示します。



■ 各種モード設定ビット (S1, S0)

S1, S0 ビットは, リセットシーケンス終了後のバスモードおよびアクセスモードを指定するビットです。

表 7.3-1 に, S1, S0 ビットの設定内容を示します。

表 7.3-1 S1, S0 ビットの設定内容

S1	S0	機 能	
0	0	外部データバス 8 ビットモード	アドレスデータバスマルチプレクス
0	1	外部データバス 16 ビットモード	
1	0	外部データバス 8 ビットモード	アドレスデータバスノンマルチプレクス
1	1	外部データバス 16 ビットモード	

■ バスモード設定ビット (M1, M0)

M1, M0 ビットは, リセットシーケンス終了後の動作モードを指定するビットです。

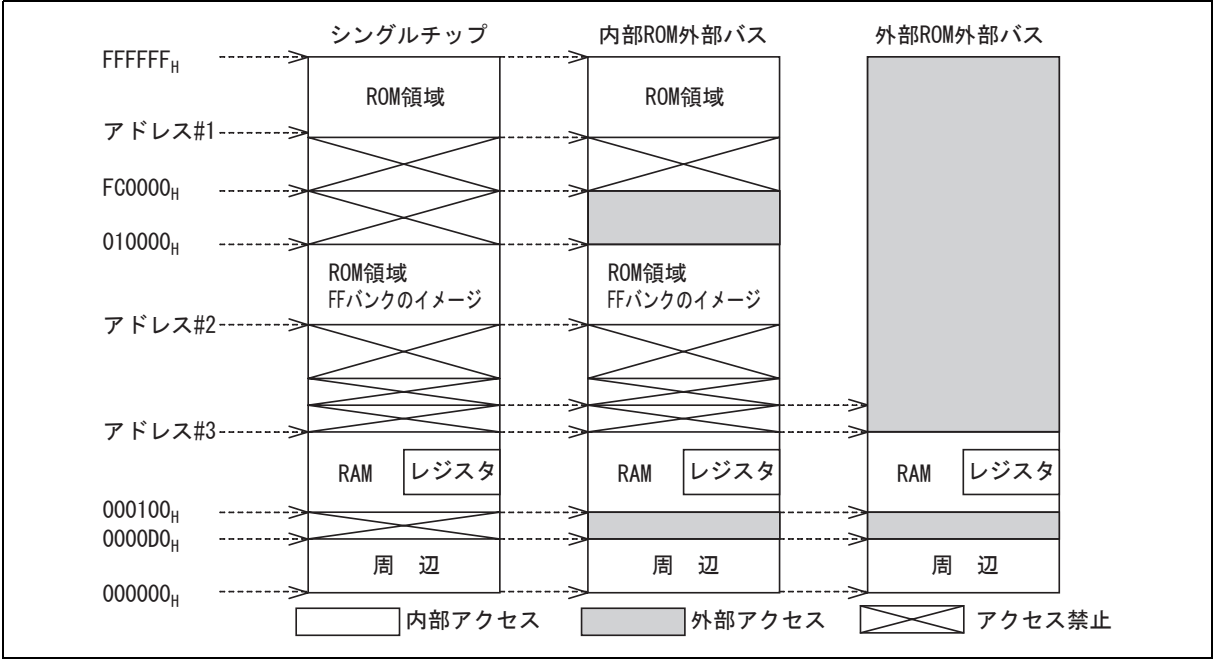
表 7.3-2 に, M1, M0 ビットの設定内容を示します。

表 7.3-2 M1, M0 ビットの設定内容

M1	M0	機 能
0	0	シングルチップモード
0	1	内 ROM 外バスモード
1	0	外 ROM 外バスモード
1	1	(設定禁止)

アクセス領域と物理アドレスの対応を、図 7.3-1 に示します。

図 7.3-1 アクセス領域と物理アドレスの関係



<注意事項>

「アドレス #1」, 「アドレス #2」, 「アドレス #3」は、各品種に依存して決まるアドレスとなります。詳細は、「付録 A メモリマップ」を参照してください。

■ モード端子とモードデータの関係 (推奨例)

表 7.3-3 に、モード端子とモードデータの関係を示します。

表 7.3-3 モード端子とモードデータの関係

モード	MD2	MD1	MD0	M1	M0	S1	S0
シングルチップ	0	1	1	0	0	X	X
内部 ROM 外部バスモード 8 ビット (アドレスデータマルチプレクス)	0	1	1	0	1	0	0
内部 ROM 外部バスモード 16 ビット (アドレスデータマルチプレクス)	0	1	1	0	1	0	1
内部 ROM 外部バスモード 8 ビット (アドレスデータノンマルチプレクス)	0	1	1	0	1	1	0
内部 ROM 外部バスモード 16 ビット (アドレスデータノンマルチプレクス)	0	1	1	0	1	1	1
外部 ROM 外部バスモード 16 ビットバスベクタ 16 ビット幅 (アドレスデータマルチプレクス)	0	0	1	1	0	0	1
外部 ROM 外部バスモード 8 ビット (アドレスデータマルチプレクス)	0	0	0	1	0	0	0
外部 ROM 外部バスモード 8 ビット (アドレスデータノンマルチプレクス)	0	1	0	1	0	1	0

(注意事項) 上位アドレス A23 ~ 16 の出力を抑制した場合には、アクセスできる最大は 64K バイトになります。

■ 各モードにおける外部端子の動作

表 7.3-4 に、ノンマルチプレクスモードおよびマルチプレクスモードにおける各外部端子の動作関係を示します。

表 7.3-4 各モードにおける外部端子の動作関係

	機 能							
	ノンマルチプレクスモード				マルチプレクスモード			
	外部アドレス制御				外部アドレス制御			
	許可 (アドレス)		禁止 (ポート)		許可 (アドレス)		禁止 (ポート)	
	外部バス拡張		外部バス拡張		外部バス拡張		外部バス拡張	
	8 ビット	16 ビット	8 ビット	16 ビット	8 ビット	16 ビット	8 ビット	16 ビット
P07 ~ P00/ D07 ~ D00/ AD07 ~ AD00	D07 ~ D00				AD07 ~ AD00			
P17 ~ P10/ D15 ~ D08/ AD15 ~ AD08	ポート	D15 ~ D08	ポート	D15 ~ D08	A15 ~ A08	AD15 ~ AD08	A15 ~ A08	AD15 ~ AD08
P27 ~ P20	A23 ~ A16		ポート		A23 ~ A16			ポート
P37 ~ P30	A07 ~ A00		A07 ~ A00		ポート			
P47 ~ P40	A15 ~ A08		A15 ~ A08					
ALE	ALE				ALE			
$\overline{\text{RD}}$	$\overline{\text{RD}}$				$\overline{\text{RD}}$			
$\text{P52}/\overline{\text{WRL}}$	$\overline{\text{WRL}}$				$\overline{\text{WRL}}$			
$\text{P53}/\overline{\text{WRH}}$	ポート	$\overline{\text{WRH}}$	ポート	$\overline{\text{WRH}}$	ポート	$\overline{\text{WRH}}$	ポート	$\overline{\text{WRH}}$
P54/HRQ	HRQ				HRQ			
$\text{P55}/\overline{\text{HAK}}$	$\overline{\text{HAK}}$				$\overline{\text{HAK}}$			
P56/RDY	RDY				RDY			
P57/CLK	CLK				CLK			

- ・シングルチップモードは、すべてポートとして使用可能です。
- ・アドレス上位と $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{HAK}}$, HRQ, RDY, CLK は、機能選択によってポートとして使用可能です。
- ・ノンマルチプレクス使用時には、アップダウンカウンタ /SCI2/ μ PG の使用はできません。アドレスとして機能します。

7.4 外部メモリアクセス

外部メモリアクセスのブロックダイアグラム、レジスタの構成 / 機能および外部メモリアクセスの動作について説明します。

■ 外部メモリアクセスの入出力信号端子

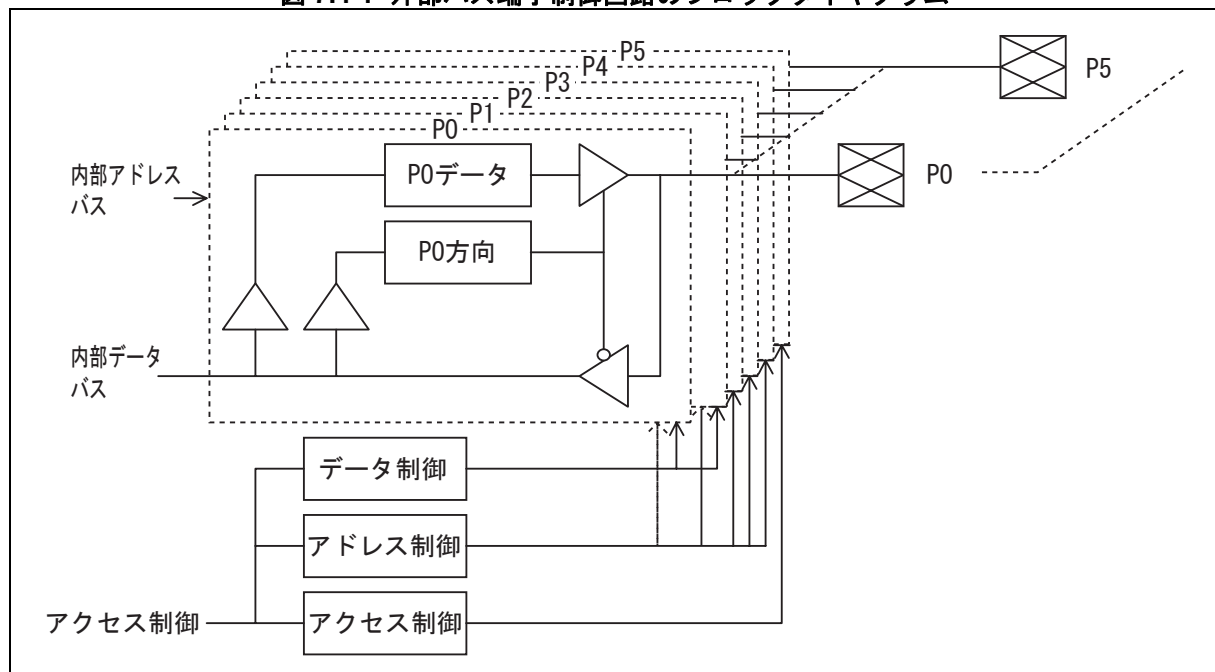
デバイス外部のメモリ / 周辺をアクセスするために、F²MC-16LX は以下のようなアドレス / データ / 制御信号を供給します。

- CLK(P57) : マシンサイクルクロック (KBP) を出力します。
- RDY(P56) : 外部レディ入力端子です。
- $\overline{\text{HAK}}$ (P55) : ホールドアクノリッジ出力端子です。
- HRQ(P54) : ホールドリクエスト入力端子です。
- $\overline{\text{WRH}}$ (P53) : データバス上位 8 ビットの書込み信号です。
- $\overline{\text{WRL}}$ (P52) : データバス下位 8 ビットの書込み信号です。
- $\overline{\text{RD}}$ (P51) : 読出し信号です。
- ALE(P50) : アドレスラッチイネーブル信号です(マルチプレクスモード時に有効)。

■ ブロックダイアグラム

図 7.4-1 に、外部バス端子制御回路のブロックダイヤグラムを示します。

図 7.4-1 外部バス端子制御回路のブロックダイアグラム



■ レジスタ一覧

図 7.4-2 に、外部バス端子制御回路のレジスタ一覧を示します。

図 7.4-2 外部バス端子制御回路のレジスタ一覧

ビット	15	14	13	12	11	10	9	8	自動レディ 機能選択レジスタ (ARSR)
0000A5 _H	IOR1	IOR0	HMR1	HMRO	—	—	LMR1	LMRO	
	(W)	(W)	(W)	(W)	(-)	(-)	(W)	(W)	リード/ライト
	(0)	(0)	(1)	(1)	(-)	(-)	(0)	(0)	初期値
ビット	7	6	5	4	3	2	1	0	外部アドレス 出力制御レジスタ (HACR)
0000A6 _H	E23	E22	E21	E20	E19	E18	E17	E16	
	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	リード/ライト
	(*)	(*)	(*)	(*)	(*)	(*)	(*)	(*)	初期値
ビット	15	14	13	12	11	10	9	8	バス制御 信号選択レジスタ (EPCR)
0000A7 _H	CKE	RYE	HDE	IOBS	HMBS	WRE	LMBS	—	
	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(-)	リード/ライト
	(1)	(0)	(0)	(0)	(*)	(1)	(0)	(-)	初期値

7.4.1 自動レディ機能選択レジスタ (ARSR)

自動レディ機能選択レジスタ (ARSR) の構成および機能について説明します。

■ 自動レディ機能選択レジスタ (ARSR)

自動レディ機能選択レジスタ (ARSR) のビット構成を、下図に示します。

	15	14	13	12	11	10	9	8	
0000A5 _H	IOR1	IOR0	HMR1	HMR0	—	—	LMR1	LMR0	自動レディ 機能選択レジスタ
	(W)	(W)	(W)	(W)	(-)	(-)	(W)	(W)	リード/ライト
	(0)	(0)	(1)	(1)	(-)	(-)	(0)	(0)	初期値

以下に、自動レディ機能選択レジスタ (ARSR) の各ビットの機能を示します。

[bit15, bit14] IOR1, IOR0

これらのビットは、0000D0_H ~ 0000FF_H の領域に対する外部アクセスを行ったときの自動ウェイト機能を選択します。設定内容は、下表のとおりです。

IOR1	IOR0	設 定
0	0	自動ウェイト禁止 [初期値]
0	1	外部アクセス時, 1 マシンサイクルの自動ウェイト
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイト
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイト

[bit13, bit12] HMR1, HMR0

これらのビットは、800000_H ~ FFFFFFFF_H の領域に対する外部アクセスを行ったときの自動ウェイト機能を選択します。設定内容は、下表のとおりです。

HMR1	HMR0	設 定
0	0	自動ウェイト禁止
0	1	外部アクセス時, 1 マシンサイクルの自動ウェイト
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイト
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイト [初期値]

[bit9, bit8] LMR1, LMR0

これらのビットは、002000_H ~ 7FFFFFFF_H の領域に対する外部アクセスを行ったときの自動ウェイト機能を選択します。設定内容は、下表のとおりです。

LMR1	LMR0	設 定
0	0	自動ウェイト禁止 [初期値]
0	1	外部アクセス時, 1 マシンサイクルの自動ウェイト
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイト
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイト

7.4.2 外部アドレス出力制御レジスタ (HACR)

外部アドレス出力制御レジスタの構成および機能について説明します。

■ 外部アドレス出力制御レジスタ (HACR)

外部アドレス出力制御レジスタのビット構成を、下図に示します。

	7	6	5	4	3	2	1	0	外部アドレス 出力制御レジスタ
0000A6 _H	E23	E22	E21	E20	E19	E18	E17	E16	
	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	リード/ライト
	(*)	(*)	(*)	(*)	(*)	(*)	(*)	(*)	初期値

外部アドレス出力制御レジスタは、アドレス (A23 ~ A16) の外部への出力を制御するレジスタです。各ビットがそれぞれアドレス A23 ~ A16 に対応し、各アドレス出力端子を下表のように制御します。

0	対応する端子は、アドレス出力 (AXX) になります。
1	対応する端子は I/O ポート (PXX) になります。

本レジスタは、デバイスがシングルチップモード時にはアクセスすることができません。その場合、本レジスタの値にかかわらず、全ポートが I/O ポートとして機能します。本レジスタの全ビットはすべて書込み専用で、読出し値は "1" となります。また、アドレス出力を選択してアドレスの出力を期待する場合は、DDR を "0" で使用してください。

初期値は内部ベクタモードで起動した場合のみ "1" となります。その他の場合は "0" となります。

<注意事項>

PPG 使用時は、"1" (I/O ポート設定) として使用してください。

7.4.3 バス制御信号選択レジスタ (EPCR)

バス制御信号選択レジスタの構成および機能について説明します。

■ バス制御信号選択レジスタ (EPCR)

バス制御信号選択レジスタは、外部バスモード時のバス動作の制御機能を設定するレジスタです。

バス制御信号選択レジスタのビット構成を、下図に示します。

	15	14	13	12	11	10	9	8	
0000A7 _H	CKE	RYE	HDE	IOBS	HMBS	WRE	LMBS	—	バス制御 信号選択レジスタ
	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(-)	リード/ライト
	(1)	(0)	(0)	(0)	(*)	(1)	(0)	(-)	初期値

このレジスタは、デバイスがシングルチップモード時には、アクセスすることができません。シングルチップモードの場合は、このレジスタの値にかかわらず、全端子が I/O ポートとして機能します。本レジスタの全ビットはすべて書き込み専用で、読出し値は "1" となります。

以下に、バス制御信号選択レジスタの各ビットの機能を説明します。

[bit15] CKE

このビットは、外部クロック (CLK) の出力を制御します。

0	I/O ポート (P57) 動作 (クロック禁止)
1	クロック信号 (CLK) 出力許可 [初期値]

[bit14] RYE

このビットは、外部レディ (RDY) の入力を制御します。

0	I/O ポート (P56) 動作 (外部 RDY 入力禁止) [初期値]
1	外部レディ (RDY) 入力許可

[bit13] HDE

このビットは、ホールド関係端子の入出力許可を指定するビットです。このビットの設定によって、ホールド要求入力 (HRQ) とホールドアクノリッジ出力 ($\overline{\text{HAK}}$) を制御します。

0	I/O ポート (P55, P54) 動作 (ホールド機能入出力禁止) [初期値]
1	ホールド要求 (HRQ) 入力 / ホールドアクノリッジ ($\overline{\text{HAK}}$) 出力許可

[bit12] IOBS

このビットは、外部データバス 16 ビットモード時に、0000D0_H ~ 0000FF_H の領域に対する外部バスをアクセスする場合のバス幅を指定します。

0	16 ビットバス幅アクセス [初期値]
1	8 ビットバス幅アクセス

第7章 モード設定

[bit11] HMBS

このビットは、外部データバス 16 ビットモード時に、800000_H ~ FFFFFFF_H の領域に対する外部バスをアクセスする場合のバス幅を指定します。

0	16 ビットバス幅アクセス [外部ベクタモード 1 の場合の初期値]
1	8 ビットバス幅アクセス [外部ベクタモード 0, 2 の場合の初期値]

[bit10] WRE

このビットは、外部書込み信号 (外部データバス 16 ビットモードの場合には、 $\overline{\text{WRH}}$ / $\overline{\text{WRL}}$ 両端子、外部データバス 8 ビットモードの場合は、 $\overline{\text{WRL}}$ 端子) の出力を制御します。

0	I/O ポート (P53, P52) 動作 (書込み信号出力禁止)
1	書込みストローブ信号 ($\overline{\text{WRH}}$ / $\overline{\text{WRL}}$, または $\overline{\text{WRL}}$ のみ) の出力許可 [初期値]

[bit9] LMBS

このビットは、外部データバス 16 ビットモード時に、002000_H ~ 7FFFFFF_H の領域に対する外部バスをアクセスする場合のバス幅を指定します。

0	16 ビットバス幅アクセス [初期値]
1	8 ビットバス幅アクセス

<注意事項>

RYE, HDE ビットで RDY, HRQ 入力を許可した場合でも、そのポートの I/O ポート機能は有効になります。そのため、DDR5 中のそのポートに対応したビットは必ず "0" (入力モード) にしてください。

7.5 モード設定の各モードにおける動作

モード設定の各モードにおける動作について、タイミングチャートを示して説明します。

■ モードの種類

機能ごとに、以下の項目の動作を説明します。

- 外部メモリアクセス制御信号
 - 外部データバス 8 ビットモード (ノンマルチプレクスモード)
 - 外部データバス 8 ビットモード (マルチプレクスモード)
 - 外部データバス 16 ビットモード (ノンマルチプレクスモード)
 - 外部データバス 16 ビットモード (マルチプレクスモード)
- レディ機能
 - ノンマルチプレクスモード
 - マルチプレクスモード
- ホールド機能
 - ノンマルチプレクスモード
 - マルチプレクスモード

7.5.1 外部メモリアクセス制御信号

外部メモリへのアクセスは、レディ機能を使用しない場合に3サイクルで行います。

■ 外部メモリアクセス制御信号

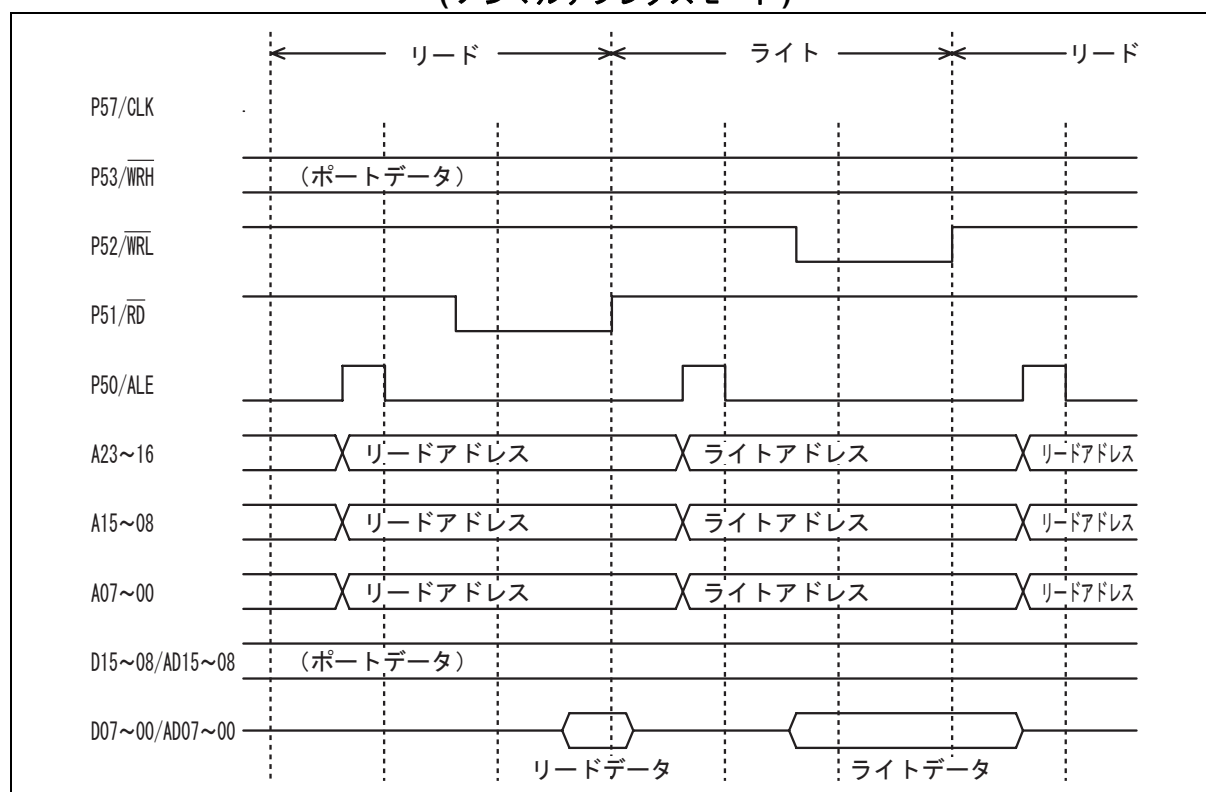
各モードにおける外部アクセスのタイミングチャートを、図 7.5-1 ～図 7.5-4 に示します。外部データバス 16 ビットモードでの 8 ビットバス幅アクセスは、8 ビット幅の周辺チップと 16 ビット幅の周辺チップを混在して外部バスに接続した場合に、8 ビット幅の周辺チップを読出し / 書込みをするための機能です。8 ビットバス幅アクセスは、データバスの下位 8 ビットを使用して実行されるので、8 ビット幅の周辺チップはデータの下位 8 ビットに接続してください。外部データバス 16 ビットモードで 16 ビットバス幅アクセスを行うか、8 ビットバス幅アクセスを行うかは、EPCR の HMBS/LMBS/IOBS ビットを指定することによります。なお、マルチプレクスモード時において、アドレス出力と ALE のアサート出力のみを行い、 $\overline{\text{RD}}/\overline{\text{WRL}}/\overline{\text{WRH}}$ をアサートしないことにより実際には、バス動作を行わない場合があります。

<注意事項>

ALE 信号のみで周辺チップへのアクセスを実行することがないようにしてください。

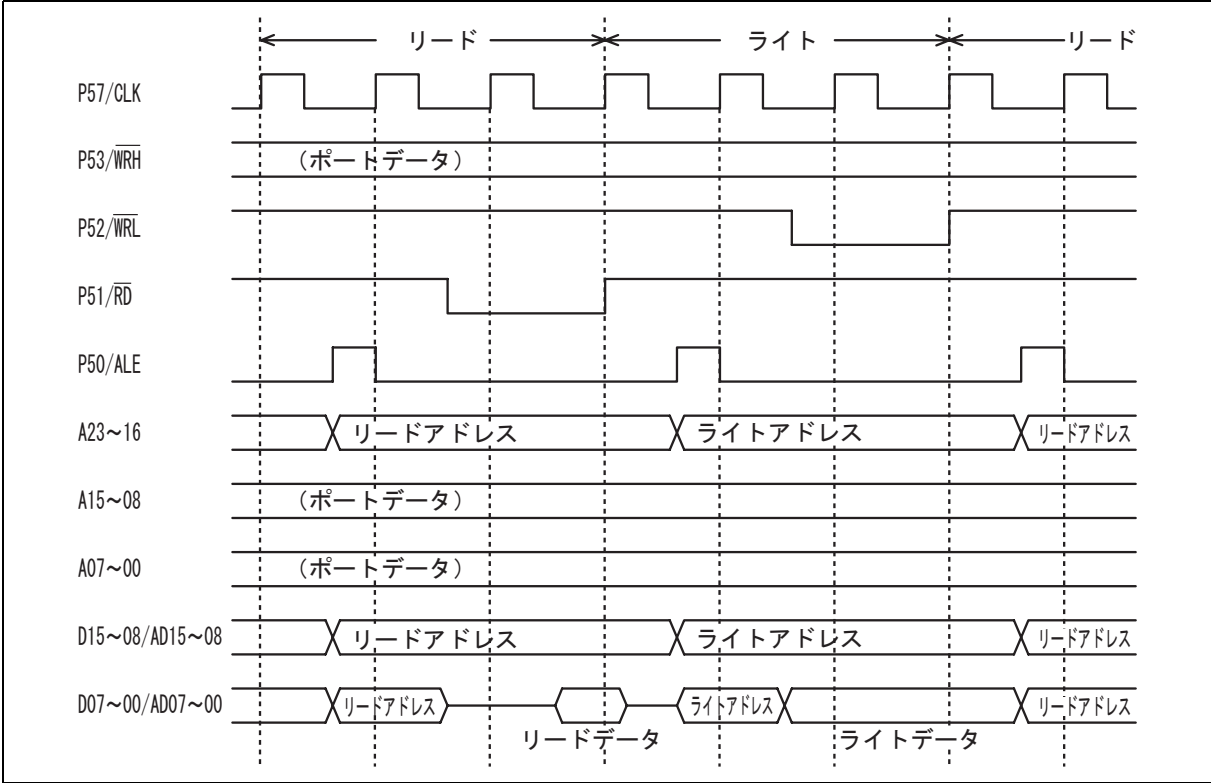
● 外部データバス 8 ビットモード (ノンマルチプレクスモード)

図 7.5-1 外部データバス 8 ビットモードのアクセスタイミングチャート
(ノンマルチプレクスモード)



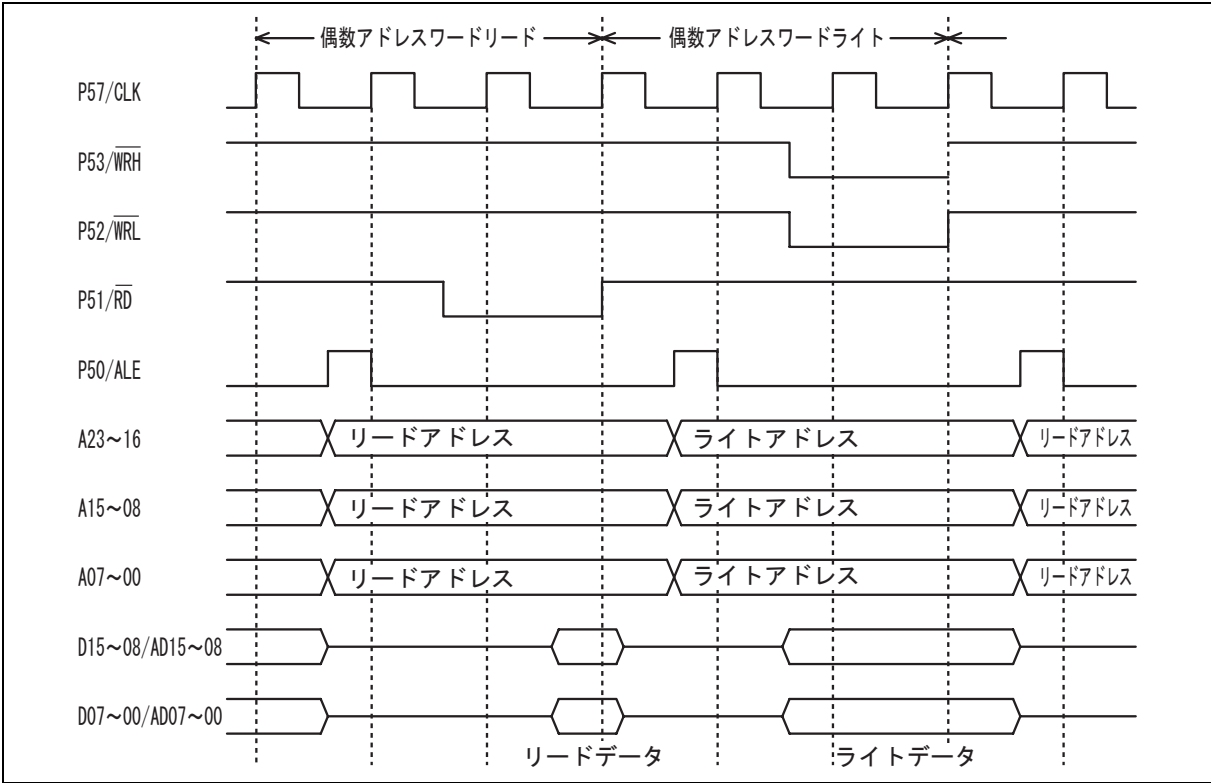
● 外部データバス 8 ビットモード (マルチプレクスモード)

図 7.5-2 外部データバス 8 ビットモードのアクセスタイミングチャート
(マルチプレクスモード)



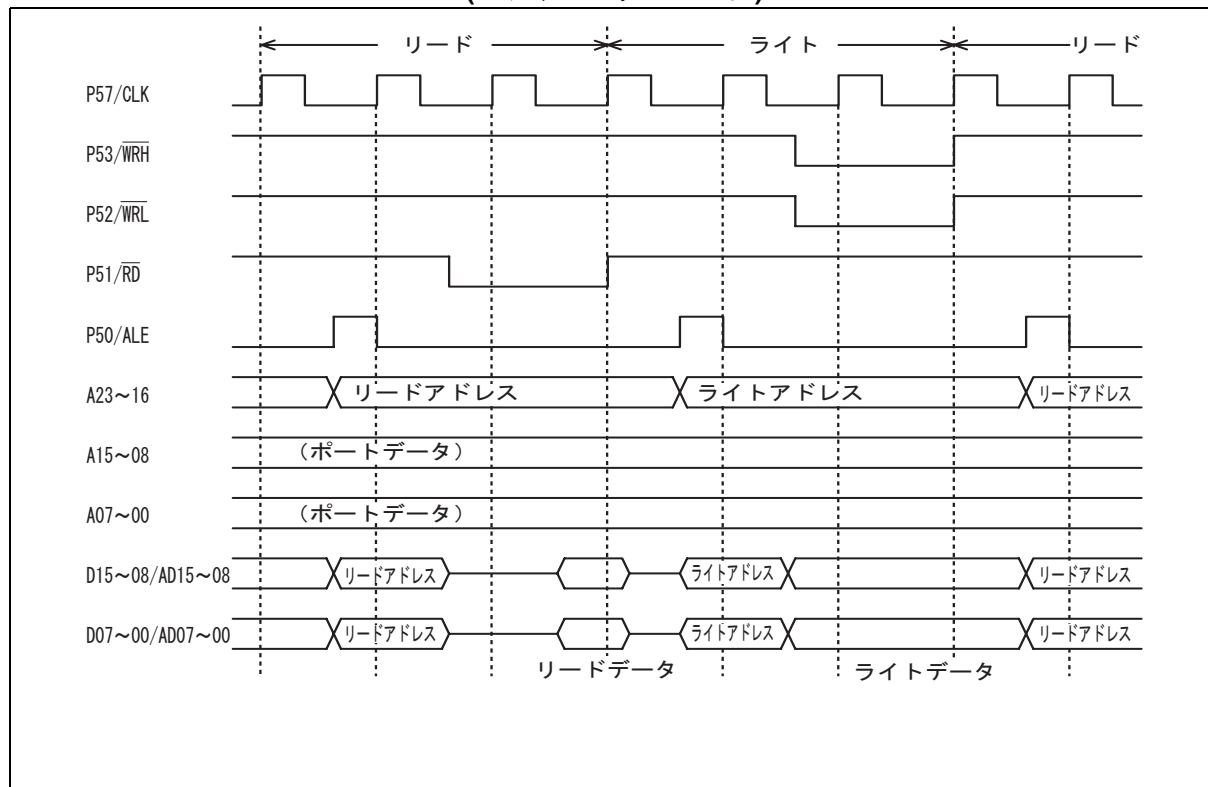
● 外部データバス 16 ビットモード (ノンマルチプレクスモード)

図 7.5-3 外部データバス 16 ビットモードのアクセスタイミングチャート
(ノンマルチプレクスモード)



● 外部データバス 16 ビットモード (マルチプレクスモード)

図 7.5-4 外部データバス 16 ビットモードのアクセスタイミングチャート
(マルチプレクスモード)



7.5.2 レディ機能

P56/RDY 端子, または自動レディ機能選択レジスタ (ARSR) の設定により, 低速のメモリや周辺回路へのアクセスが可能となります。バス制御信号選択レジスタ (EPCR) 中の RYE ビットが "1" に設定されている場合, 外部領域へのアクセス時に P56/RDY 端子に "L" レベルが入力されている期間ウェイトサイクルとなり, アクセスサイクルを延ばすことができます。

■ レディ機能

F²MC-16LX は, 外部メモリ用のオートレディ機能を2種類内蔵しています。オートレディ機能は, アドレス 002000_H ~ 7FFFFFF_H 間に配置した下位アドレス外部領域へのアクセスが発生したとき, アドレス 800000_H ~ FFFFFFF_H 間に配置した上位アドレス外部領域へのアクセスが発生したときに, 外部回路なしで自動的に1~3サイクルのウェイトサイクルを挿入し, アクセスサイクルを延ばすことができる機能です。ARSR 中の LMR1/LMR0 ビット (下位アドレス外部領域) および ARSR 中の HMR1/HMR0 ビット (上位アドレス外部領域) の設定により起動します。

さらに, F²MC-16LX は, 外部 I/O 用にオートレディ機能をメモリ用とは独立に内蔵しています。アドレス 0000D0_H ~ 0000FF_H 間の外部領域へのアクセス時に, 外部回路なしで自動的に1~3サイクルのウェイトサイクルを挿入し, アクセスサイクルを延ばすことができる機能です。ARSR 中の IOR1/IOR0 ビットの設定により起動します。

外部メモリ用, 外部 I/O 用のオートレディどちらも EPCR 中の RYE ビットが "1" に設定されている場合, 上記オートレディによるウェイトサイクル終了後に P56/RDY 端子に "L" レベルが入力されていると, そのままウェイトサイクルを継続します。

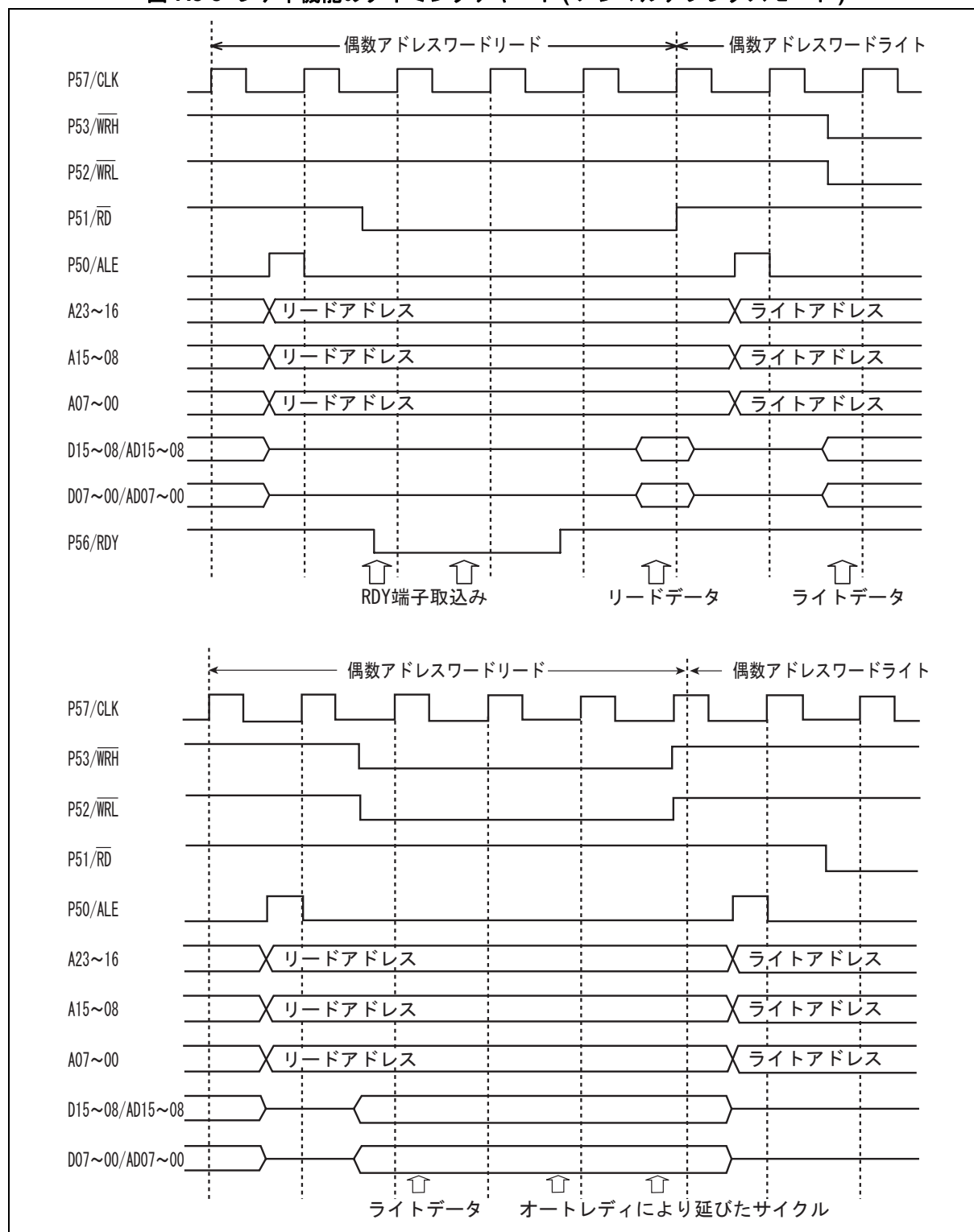
以下に, ノンマルチプレクスモードおよびマルチプレクスモードにおけるレディ機能のタイミングチャートを示します。どちらのモードも上の図がレディ機能を設定しない場合, 下の図がレディ機能を設定した場合です。

<注意事項>

RDY 端子から入力する場合, AC 規格を満たさないとき, 本デバイスは暴走する恐れがありますので, 注意してください。

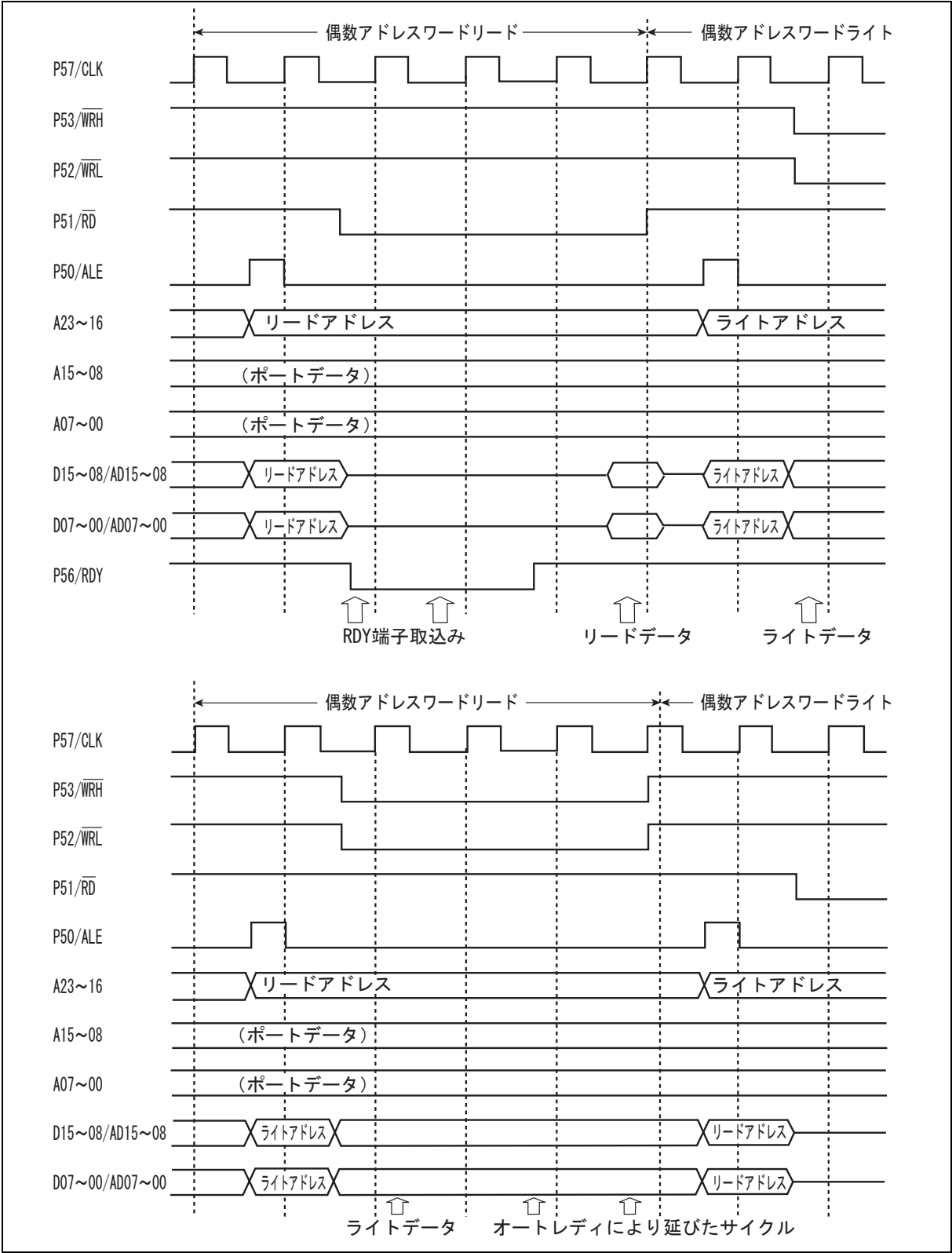
● ノンマルチプレクスモード

図 7.5-5 レディ機能のタイミングチャート(ノンマルチプレクスモード)



● マルチプレクスモード

図 7.5-6 レディ機能のタイミングチャート (マルチプレクスモード)



7.5.3 ホールド機能

ホールド機能の動作を、タイミングチャートを示して説明します。

■ ホールド機能の動作

EPCR 中の HDE ビットが "1" に設定されている場合、P54/HRQ, P55/ $\overline{\text{HAK}}$ 両端子による外部バスのホールド機能が有効となります。P54/HRQ 端子に "H" レベルを入力すると、CPU の命令終了時 (ストリング命令の場合は 1 エLEMENTデータの処理の終了後) にホールド状態となり、P55/ $\overline{\text{HAK}}$ より "L" レベルを出力して以下の端子をハイインピーダンス状態にします。

● ノンマルチプレクスモード

- アドレス出力 :A23 ~ A00
- データ入出力 :D15/AD15 ~ D00/AD00
- バス制御信号 :P51/ $\overline{\text{RD}}$, P52/ $\overline{\text{WRL}}$, P53/ $\overline{\text{WRH}}$

● マルチプレクスモード

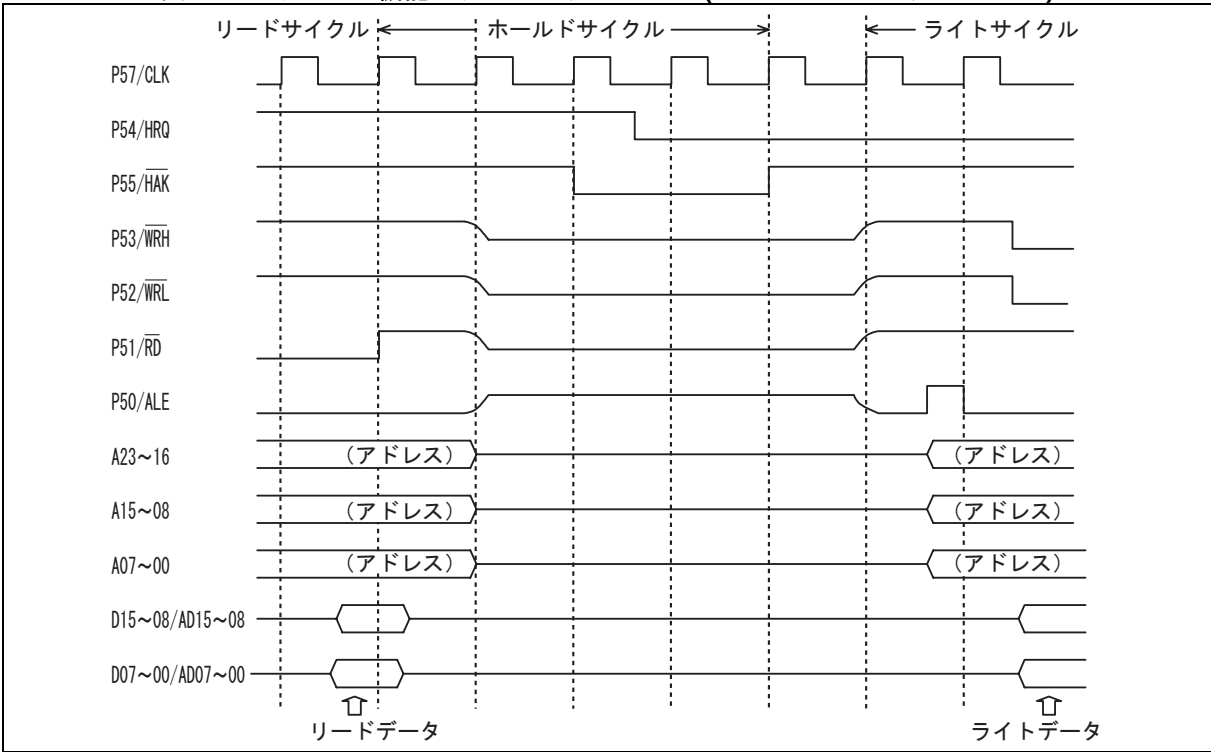
- アドレス出力 :A23 ~ A16
- アドレス出力, データ入出力 :D15/AD15 ~ D00/AD00
- バス制御信号 :P51/ $\overline{\text{RD}}$, P52/ $\overline{\text{WRL}}$, P53/ $\overline{\text{WRH}}$

これにより、デバイス外部回路により外部バスを使用することが可能となります。P54/HRQ 端子に "L" レベルを入力すると、P55/ $\overline{\text{HAK}}$ 端子は "H" レベル出力となり、外部端子状態が復活して CPU は動作を再開します。STOP 状態においては、ホールド要求は受け付けません。

■ ノンマルチプレクスモード

図 7.5-7 に、外部データバス 16 ビットモードにおけるノンマルチプレクスモードのホールド機能のタイミングチャートを示します。

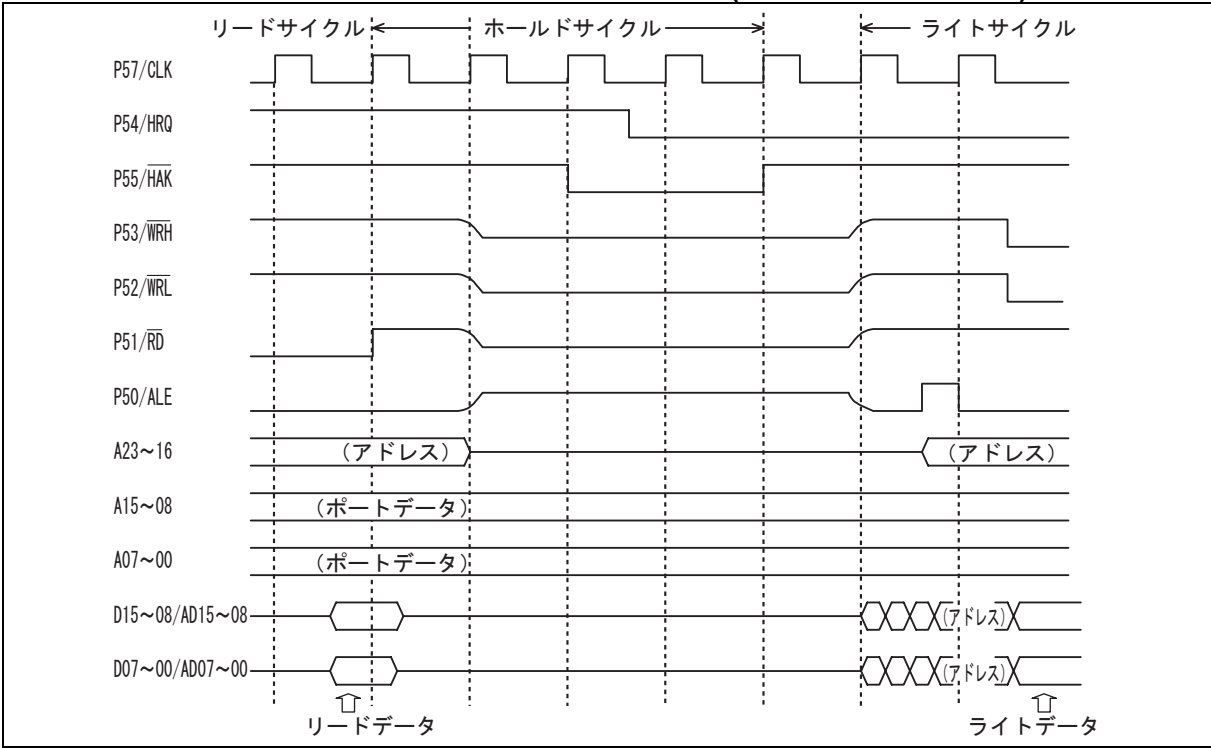
図 7.5-7 ホールド機能のタイミングチャート (ノンマルチプレクスモード)



■ マルチプレクスモード

図 7.5-8 に、外部データバス 16 ビットモードにおけるマルチプレクスモードのホールド機能のタイミングチャートを示します。

図 7.5-8 ホールド機能のタイミングチャート (マルチプレクスモード)



第8章

I/O ポート

I/O ポートの機能およびレジスタについて説明します。

8.1 I/O ポートの機能

8.2 I/O ポートのレジスタ

8.1 I/O ポートの機能

I/O ポートの機能概要を示します。

■ I/O ポートの機能

I/O ポートは、ポートデータレジスタ (PDR) によって、CPU からデータを I/O 端子に出力したり、I/O 端子に入力された信号を CPU に取り込んだりする機能をもっています。また、I/O ポートはポート方向レジスタ (DDR) によって、I/O 端子の入出力の方向をビット単位で任意に設定することができます。

MB90480/485 シリーズには、84 本の入出力端子があります。

8.2 I/O ポートのレジスタ

I/O ポートで使用するレジスタの構成および機能について説明します。

■ I/O ポートのレジスタ

I/O ポートには以下のレジスタがあります。

- ポートデータレジスタ (PDR0 ~ PDRA)
- ポート方向レジスタ (DDR0 ~ DDRA)
- ポート入力抵抗レジスタ (RDR0, RDR1)
- ポート出力端子レジスタ (ODR7, ODR4)
- アナログ入力許可レジスタ (ADER)
- アップダウンタイマ入力許可レジスタ (UDER)

8.2.1 ポートデータレジスタ (PDR0 ~ PDRA)

ポートデータレジスタ (PDR0 ~ PDRA) の構成および機能について説明します。

■ ポートデータレジスタ (PDR0 ~ PDRA)

図 8.2-1 に、ポートデータレジスタ (PDR0 ~ PDRA) の一覧を示します。

図 8.2-1 ポートデータレジスタ (PDR0 ~ PDRA) の一覧

PDR0	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス:000000 _H	P07	P06	P05	P04	P03	P02	P01	P00	不定	R/W *1
PDR1	7	6	5	4	3	2	1	0		
アドレス:000001 _H	P17	P16	P15	P14	P13	P12	P11	P10	不定	R/W *1
PDR2	7	6	5	4	3	2	1	0		
アドレス:000002 _H	P27	P26	P25	P24	P23	P22	P21	P20	不定	R/W *1
PDR3	7	6	5	4	3	2	1	0		
アドレス:000003 _H	P37	P36	P35	P34	P33	P32	P31	P30	不定	R/W *1
PDR4	7	6	5	4	3	2	1	0		
アドレス:000004 _H	P47	P46	P45	P44	P43	P42	P41	P40	不定	R/W *1
PDR5	7	6	5	4	3	2	1	0		
アドレス:000005 _H	P57	P56	P55	P54	P53	P52	P51	P50	不定	R/W *1
PDR6	7	6	5	4	3	2	1	0		
アドレス:000006 _H	P67	P66	P65	P64	P63	P62	P61	P60	不定	R/W *1
PDR7	7	6	5	4	3	2	1	0		
アドレス:000007 _H	P77	P76	P75	P74	P73	P72	P71	P70	不定 *2	R/W *1
PDR8	7	6	5	4	3	2	1	0		
アドレス:000008 _H	P87	P86	P85	P84	P83	P82	P81	P80	不定	R/W *1
PDR9	7	6	5	4	3	2	1	0		
アドレス:000009 _H	P97	P96	P95	P94	P93	P92	P91	P90	不定	R/W *1
PDRA	7	6	5	4	3	2	1	0		
アドレス:00000A _H	-	-	-	-	PA3	PA2	PA1	PA0	不定	R/W *1

*1: 入出力ポートへの R/W アクセスは、メモリへの R/W アクセスとは多少動作が異なります。
 以下のような動作をしますので、注意してください。

- 0: 入力モード
 - 読出し時: 対応する端子のレベルが読み出されます。
 - 書込み時: 出力用のラッチに書き込まれます。
- 1: 出力モード
 - 読出し時: データレジスタラッチの値が読み出されます。
 - 書込み時: 対応する端子に出力されます。

*2: MB90485 シリーズのみ初期値は "11xxxxxx_B" になります。

8.2.2 ポート方向レジスタ (DDR0 ~ DDRA)

ポート方向レジスタの構成および機能について説明します。

■ ポート方向レジスタ (DDR0 ~ DDRA)

図 8.2-2 に、ポート方向レジスタ (DDR0 ~ DDRA) の一覧を示します。

図 8.2-2 ポート方向レジスタ (DDR0 ~ DDRA) の一覧

DDR0	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス:000010 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _B	R/W
DDR1	7	6	5	4	3	2	1	0		
アドレス:000011 _H	D17	D16	D15	D14	D13	D12	D11	D10	00000000 _B	R/W
DDR2	7	6	5	4	3	2	1	0		
アドレス:000012 _H	D27	D26	D25	D24	D23	D22	D21	D20	00000000 _B	R/W
DDR3	7	6	5	4	3	2	1	0		
アドレス:000013 _H	D37	D36	D35	D34	D33	D32	D31	D30	00000000 _B	R/W
DDR4	7	6	5	4	3	2	1	0		
アドレス:000014 _H	D47	D46	D45	D44	D43	D42	D41	D40	00000000 _B	R/W
DDR5	7	6	5	4	3	2	1	0		
アドレス:000015 _H	D57	D56	D55	D54	D53	D52	D51	D50	00000000 _B	R/W
DDR6	7	6	5	4	3	2	1	0		
アドレス:000016 _H	D67	D66	D65	D64	D63	D62	D61	D60	00000000 _B	R/W
DDR7	7	6	5	4	3	2	1	0		
アドレス:000017 _H	D77 ^{*1}	D76 ^{*2}	D75	D74	D73	D72	D71	D70	00000000 _B ^{*3}	R/W
DDR8	7	6	5	4	3	2	1	0		
アドレス:000018 _H	D87	D86	D85	D84	D83	D82	D81	D80	00000000 _B	R/W
DDR9	7	6	5	4	3	2	1	0		
アドレス:000019 _H	D97	D96	D95	D94	D93	D92	D91	D90	00000000 _B	R/W
DDRA	7	6	5	4	3	2	1	0		
アドレス:00001A _H	—	—	—	—	DA3	DA2	DA1	DA0	----0000 _B	R/W

^{*1}: MB90485 シリーズのみ bit7 は "—" になります。
^{*2}: MB90485 シリーズのみ bit6 は "—" になります。
^{*3}: MB90485 シリーズのみ初期値は "xx000000_B" になります。

● 各端子がポートとして機能している場合

各端子がポートとして機能している場合、対応する各端子を以下のように制御します。

- 0: 入力モード
- 1: 出力モード (リセットで "0" になります。)

<注意事項>

- リードモディファイライト系の命令 (ビットセットなどの命令) を使用してこのレジスタをアクセスすると、命令で注目しているビットは所定の値になります。それ以外のビットで入力設定してあるものに対応する出力レジスタの内容は、そのときの端子の入力値に書き換わります。このため、入力として使用していた端子を出力に切り換えるときには、はじめに希望値を PDR に書き込んでから DDR をセットし、そのあとで出力に切り換えるようにしてください。
 - MB90485 シリーズの P77, P76 には、DDR がありません。常にポートとしてデータを有効としますので、P77, P76 を I²C 端子として使用する場合には、PDR の値を "1" にしてください (なお、P77, P76 として使用する場合は、I²C を停止してください)。また、このポートはオープンドレイン出力形式 (P-ch なし) ですので、入力ポートとして使用する場合には出力トランジスタを off するために、PDR の値を "1" にし、かつ、外部出力にはプルアップ抵抗を付加する必要があります。
-

8.2.3 その他のレジスタ

ポートデータレジスタ (PDR0 ~ PDRA), ポート方向レジスタ (DDR0 ~ DDRA) 以外のレジスタの構成および機能について説明します。

■ ポート入力抵抗レジスタ (RDR0, RDR1)

ポート入力抵抗レジスタ (RDR0, RDR1) のビット構成を下図に示します。

RDR0	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス: 00001C _H	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	00000000 _B	R/W
RDR1	7	6	5	4	3	2	1	0		
アドレス: 00001D _H	RD17	RD16	RD15	RD14	RD13	RD12	RD11	RD10	00000000 _B	R/W

ポート入力抵抗レジスタ (RDR0, RDR1) は, 入力モード時にプルアップ抵抗をなしとするか, ありとするかを設定します。

- 0: 入力モード時プルアップ抵抗なし
- 1: 入力モード時プルアップ抵抗あり

このレジスタは, 出力モード時には機能をもちません (プルアップ抵抗なし)。

入出力モードは, ポート方向レジスタ (DDR) の設定で決まります。

ストップ時 (SPL=1) のときはプルアップ抵抗なしになります (ハイインピーダンス)。

外部バスで使用する場合は, この機能は使用禁止です。本レジスタへは書き込まないようにしてください。

■ ポート出力端子レジスタ (ODR7, ODR4)

ポート出力端子レジスタ (ODR7, ODR4) のビット構成を下図に示します。

ODR7	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス: 00001E _H	OD77*1	OD76*2	OD75	OD74	OD73	OD72	OD71	OD70	00000000 _B *3	R/W
ODR4	7	6	5	4	3	2	1	0		
アドレス: 00001B _H	OD47	OD46	OD45	OD44	OD43	OD42	OD41	OD40	00000000 _B	R/W

*1: MB90485 シリーズのみ bit7 は "—" になります。
 *2: MB90485 シリーズのみ bit6 は "—" になります。
 *3: MB90485 シリーズのみ初期値は "XX000000_B" になります。

ポート出力端子レジスタ (ODR7, ODR4) は, 出力モード時のオープンドレイン制御を行います。

- 0: 出力モード時に標準出力ポートになります。
- 1: 出力モード時にオープンドレイン出力ポートになります。

ポート出力端子レジスタ (ODR7, ODR4) は, 入力モード時は機能をもちません (出力 Hi-Z)。

入出力モードは, ポート方向レジスタ (DDR) の設定で決まります。

外部バスで使用する場合は, この機能は使用禁止です。本レジスタへは書き込まないようにしてください。

■ アナログ入力許可レジスタ (ADER)

アナログ入力許可レジスタ (ADER) のビット構成を下图に示します。

ADER	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス: 00001F _H	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	11111111 _B	R/W

アナログ入力許可レジスタ (ADER) は、ポート 6 の各端子を以下のように制御します。

- 0: ポート入出力モードになります。
- 1: アナログ入力モードになります。リセットで "1" になります。

MB90480/485 シリーズでは、各ビットは以下のようになります。

- ADE0:P60/AN0
- ADE1:P61/AN1
- ADE2:P62/AN2
- ADE3:P63/AN3
- ADE4:P64/AN4
- ADE5:P65/AN5
- ADE6:P66/AN6
- ADE7:P67/AN7

■ アップダウンタイム入力許可レジスタ (UDER)

アップダウンタイム入力許可レジスタ (UDER) のビット構成を下图に示します。

UDER	7	6	5	4	3	2	1	0	初期値	アクセス
アドレス: 00000B _H	—	—	UDE5	UDE4	UDE3	UDE2	UDE1	UDE0	XX000000 _B	R/W

アップダウンタイム入力許可レジスタ (UDER) は、ポート 3 の各端子を以下のように制御します。

- 0: ポート入力モードになります。
- 1: アップダウンタイム入力モードになります。リセットで "0" になります。

MB90480/485 シリーズでは、各ビットは以下のようになります。

- UDE0:P30/AIN0
- UDE1:P31/BIN0
- UDE2:P32/ZIN0
- UDE3:P33/AIN1
- UDE4:P34/BIN1
- UDE5:P35/ZIN1

第9章

タイムベースタイマ

タイムベースタイマの概要，構成，制御レジスタ，割込み，動作，使用上の注意およびプログラム例について説明します。

- 9.1 タイムベースタイマの概要
- 9.2 タイムベースタイマの構成
- 9.3 タイムベースタイマ制御レジスタ (TBTC)
- 9.4 タイムベースタイマの割込み
- 9.5 タイムベースタイマの動作
- 9.6 タイムベースタイマ使用上の注意
- 9.7 タイムベースタイマのプログラム例

9.1 タイムベースタイマの概要

タイムベースタイマは、内部カウントクロック（原発振の2分周）に同期してカウントアップする18ビットのフリーランカウンタ（タイムベースタイマカウンタ）です。4種類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があります。また、発振安定待ち時間のタイマ出力やウォッチドッグタイマなどの動作クロックを供給する機能もあります。

■ インターバルタイマ機能

インターバルタイマ機能は、一定時間間隔で繰り返し割込み要求を発生する機能です。

- タイムベースカウンタのインターバルタイマ用ビットがオーバフローすると割込み要求を発生します。
- インターバルタイマ用のビット（インターバル時間）は、次の4種類の中から選択できます。

表 9.1-1 に、タイムベースタイマのインターバル時間を示します。

表 9.1-1 タイムベースタイマのインターバル時間

内部カウントクロック周期	インターバル周期
2/HCLK(0.5μs)	2 ¹² /HCLK(約 1.0ms)
	2 ¹⁴ /HCLK(約 4.1ms)
	2 ¹⁶ /HCLK(約 16.4ms)
	2 ¹⁹ /HCLK(約 131.1ms)

HCLK: 発振クロック

() 内は発振クロック 4MHz 動作時の値です。

■ 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や、一部の周辺機能に対する動作クロックを供給する機能です。表 9.1-2 に、タイムベースタイマから各周辺に供給されるクロックの周期を示します。

表 9.1-2 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	2 ¹³ /HCLK(約 2.0ms)	セラミック振動子用発振安定待ち時間
	2 ¹⁵ /HCLK(約 8.2ms)	水晶振動子用発振安定待ち時間
	2 ¹⁷ /HCLK(約 32.8ms)	
ウォッチドッグタイマ	2 ¹² /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 /HCLK(約 4.1ms)	
	2 ¹⁶ /HCLK(約 16.4ms)	
	2 ¹⁹ /HCLK(約 131.1ms)	

HCLK: 発振クロック

() 内は発振クロック 4MHz 動作時の値です。

発振開始直後は発振周期が不安定なため、発振安定待ち時間は目安の値です。

9.2 タイムベースタイマの構成

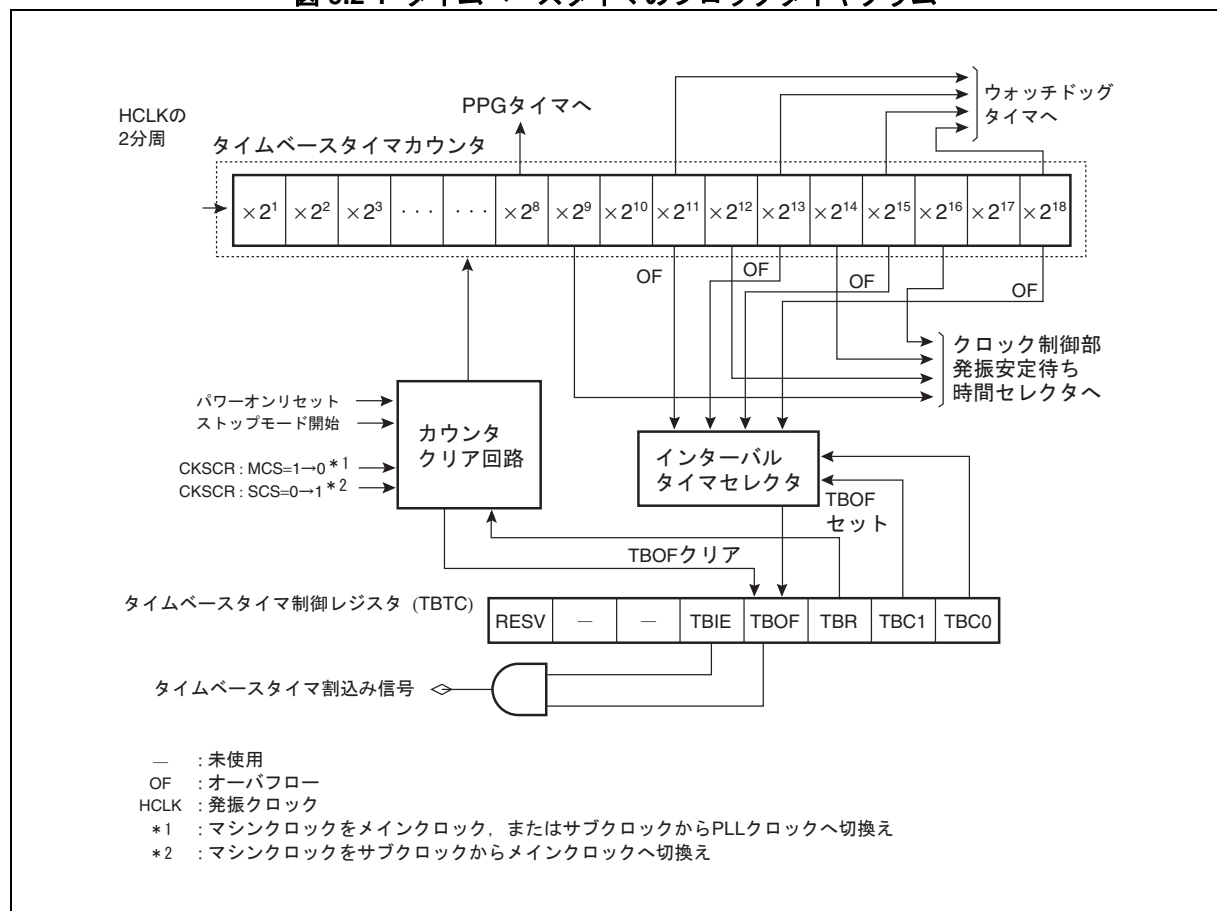
タイムベースタイマは、以下のブロックで構成されています。

- 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レクト
- 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

図 9.2-1 に、タイムベースタイマのブロックダイアグラムを示します。

図 9.2-1 タイムベースタイマのブロックダイアグラム



第9章 タイムベースタイマ

● タイムベースタイマカウンタ

発振クロック (HCLK) の 2 分周クロックをカウントクロックとする, 18 ビットのアップカウンタです。

● カウンタクリア回路

タイムベースタイマのカウンタ値は, 次の要因によってクリアされます。

- 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" の書込み
- パワーオンリセット
- メインストップモード, PLL ストップモードへの移行
- クロックモードの切換え (メインクロックモードから PLL クロックモード, サブクロックから PLL クロックモード, サブクロックモードからメインクロックモード)

● インターバルタイマセレクタ

タイムベースタイマカウンタの 4 種類の出力から 1 つを選択します。選択したビットのオーバフローが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します。

9.3 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選択、カウンタのクリア、割込み制御および状態の確認をするレジスタです。

■ タイムベースタイマ制御レジスタ (TBTC)

図 9.3-1 タイムベースタイマ制御レジスタ (TBTC)

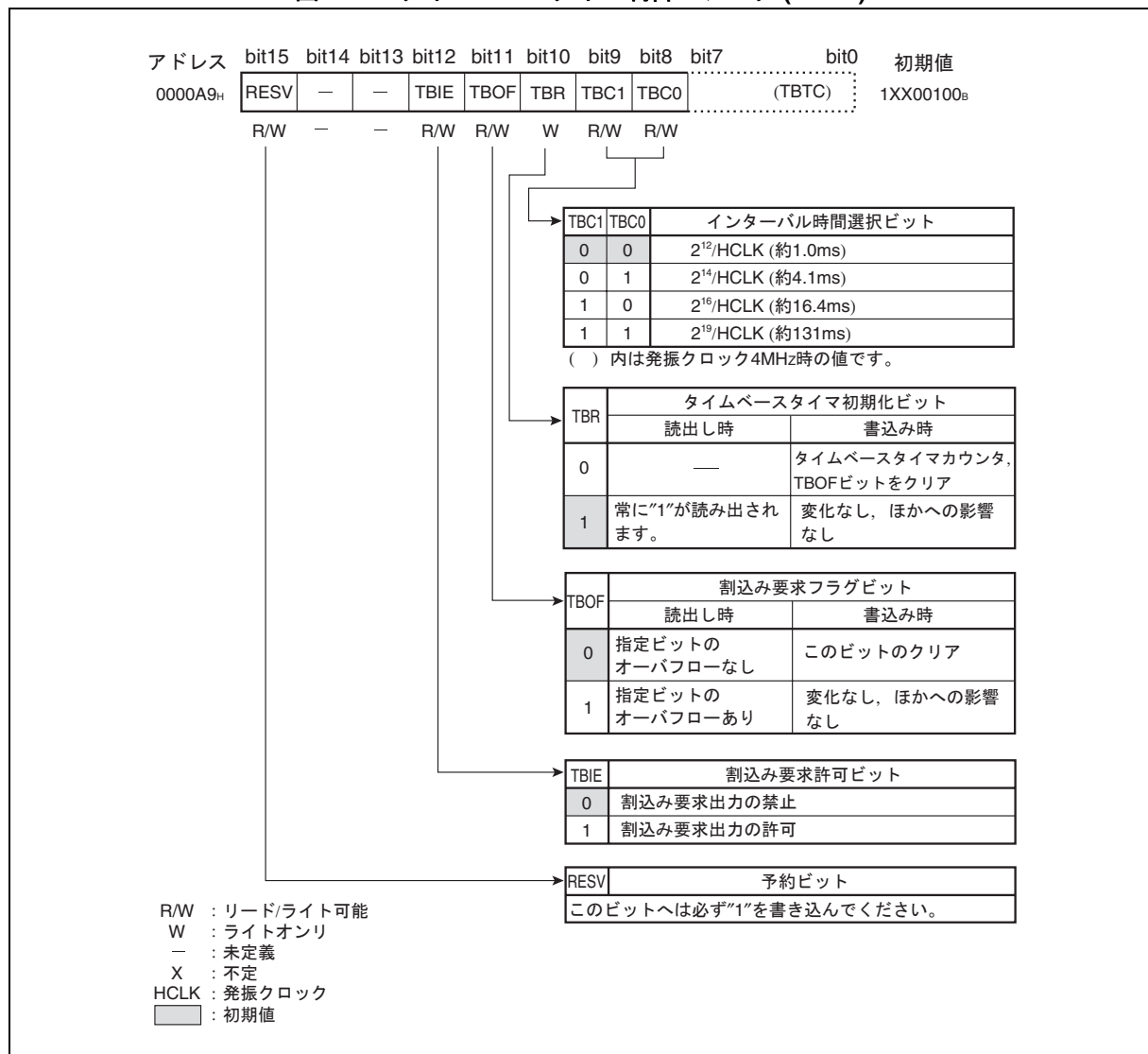


表 9.3-1 タイムベースタイマ制御レジスタ (TBTC) の各ビットの機能

ビット名		機 能
bit15	RESV: 予約ビット	このビットには必ず "1" を書き込んでください。
bit14, bit13	未定義ビット	<ul style="list-style-type: none"> ・ 読出し時の値は不定です。 ・ 書込みは動作に影響を与えません。
bit12	TBIE: 割込み要求 許可ビット	<p>CPU への割込み要求出力の許可 / 禁止をするビットです。</p> <ul style="list-style-type: none"> ・ このビットと割込み要求フラグビット (TBOF) が "1" の場合、割込み要求を出力します。
bit11	TBOF: 割込み要求 フラグビット	<p>タイムベースタイマカウンタ指定ビットがオーバフローすると、"1" にセットされるフラグです。</p> <ul style="list-style-type: none"> ・ このビットと割込み要求許可ビット (TBIE) が "1" の場合、割込み要求を出力します。 ・ 書込み時は "0" でクリアされ、"1" では変化せずほかへの影響はありません。 <p>(注意事項)</p> <ul style="list-style-type: none"> ・ TBOF ビットをクリアする場合には、割込み要求許可ビット (TBIE)、またはプロセッサステータス (PS) のインタラプトレベルマスケジスタ (ILM) の指定で、タイムベースタイマ割込みが禁止されている状態にしてください。 ・ "0" の書込み、メインストップモードへの遷移、PLL ストップモードへの遷移、サブクロックモードからメインクロックモードへの遷移、サブクロックモードから PLL クロックモードへの遷移、メインクロックモードから PLL クロックモードへの遷移およびタイムベースタイマ初期化ビット (TBR) への "0" の書込みおよびリセットにより "0" にクリアされます。
bit10	TBR: タイムベースタイマ初期化ビット	<p>タイムベースタイマカウンタをクリアするビットです。</p> <ul style="list-style-type: none"> ・ このビットに "0" を書き込むとカウンタがクリアされ、さらに TBOF ビットをクリアします。"1" では変化せずほかへの影響はありません。 ・ 読出し値は、常に "1" です
bit9, bit8	TBC1, TBC0: インターバル時間 選択ビット	<p>インターバルタイマの周期を選択するビットです。</p> <ul style="list-style-type: none"> ・ タイムベースタイマカウンタのインターバルタイマ用のビットが指定されます。 ・ 4 種類のインターバル時間が選択できます。

9.4 タイムベースタイマの割込み

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ビットのオーバーフローで割込み要求を発生させることができます (インターバルタイマ機能)。

■ タイムベースタイマの割込み

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用のビットがオーバーフローすると、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) が "1" にセットされます。その場合、割込み要求許可ビット (TBIE) を "1" にして割込み要求を許可していると、CPU に割込み要求が発生します。割込み処理ルーチンでは、TBOF ビットに "0" を書き込んで、割込み要求をクリアしてください。なお、TBOF ビットは、割込み要求許可ビット (TBIE) の値に関係なく、指定したビットがオーバーフローするとセットされます。

<注意事項>

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) をクリアする場合には、割込み要求許可ビット (TBIE)、またはプロセッサステータス (PS) のインタラプトレベルマスキングレジスタ (ILM) の設定で、タイムベースタイマ割込みが禁止されている状態で行ってください。

<参考>

- TBOF ビットが "1" の場合に、TBIE ビットを禁止から許可 (0 → 1) にすると直ちに割込み要求が発生します。
- タイムベースタイマでは、μDMAC は使用できません。

■ タイムベースタイマの割込みと μDMAC

表 9.4-1 に、タイムベースタイマの割込みと μDMAC を示します。

表 9.4-1 タイムベースタイマの割込みと μDMAC

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			μDMAC
	レジスタ名	アドレス	下位	上位	バンク	
#41	ICR15	0000BF _H	FFFF58 _H	FFFF59 _H	FFFF5A _H	×

×: 使用不可

<注意事項>

ICR15 は、タイムベースタイマ割込みと時計タイマ割込みとフラッシュ書込みで共用しています。割込みは 3 つの用途に使用できますが、割込みレベルは同一になります。

9.5 タイムベースタイマの動作

タイムベースタイマには、インターバルタイマ機能、発振安定待ち時間用タイマ機能および一部の周辺機能へのクロック供給機能があります。

■ 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、任意のインターバル時間ごとに割込み要求を発生します。インターバルタイマとして動作させるには、図 9.5-1 の設定が必要です。

図 9.5-1 タイムベースタイマの設定

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit0
0000A9 _H TBTC	RESV	—	—	TBIE	TBOF	TBR	TBC1	TBC0	(WDTC)	
	1	—	—	⊙	0	0	⊙	⊙		

⊙ : 使用ビット
 — : 未使用ビット
 0 : "0"を設定
 1 : "1"を設定

- タイムベースタイマカウンタは、クロックが発振している限り、内部カウントクロック（発振クロックの2分周）に同期して、カウントアップを続けます。
- カウンタがクリアされると "0" からカウントアップを行い、インターバルタイマ用のビットがオーバーフローすると、割込み要求フラグビット (TBOF) を "1" にセットします。この場合、割込み要求出力を許可 (TBIE=1) しているとクリアされた時間を基準にして、選択されたインターバル時間ごとに割込みを発生します。
- インターバル時間は、タイムベースタイマのクリア動作で、設定時間よりも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の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し、発振安定待ち時間用のビットがオーバーフローするまでの時間です。ただし、タイムベースタイマモードから PLL クロックモードまたはメインクロックモード復帰時には、タイムベースタイマカウンタがクリアされないため、カウント途中からの時間になります。表 9.5-1 に、タイムベースカウンタのクリアと発振安定待ち時間について示します。

表 9.5-1 タイムベースタイマカウンタのクリア動作と発振安定待ち時間

動 作	タイムベース タイマカウンタ	TBOF ビット	発振安定待ち時間
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 "0" 書込み	○	○	—
パワーオンリセット	○	○	メインクロック発振安定待ち時間
ウォッチドッグリセット	×	○	
メインストップモードの解除	○	○	
PLL ストップモードの解除	○	○	
サブストップモードの解除	×	×	サブクロック発振安定待ち時間
メインクロックモードから PLL クロックモードへの移行 (MCS=1 → 0)	○	○	PLL クロック発振安定待ち時間
サブクロックモードからメインクロックモードへの移行 (SCM=0 → 1)	○	○	メイン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	×	×	—
スリープモードの解除	×	×	—

○: クリアされる

×: クリアされない

■ 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にクロックを供給しています。

タイムベースカウンタのクリアによって、ウォッチドッグタイマのインターバル時間が設定より長くなることがあります。

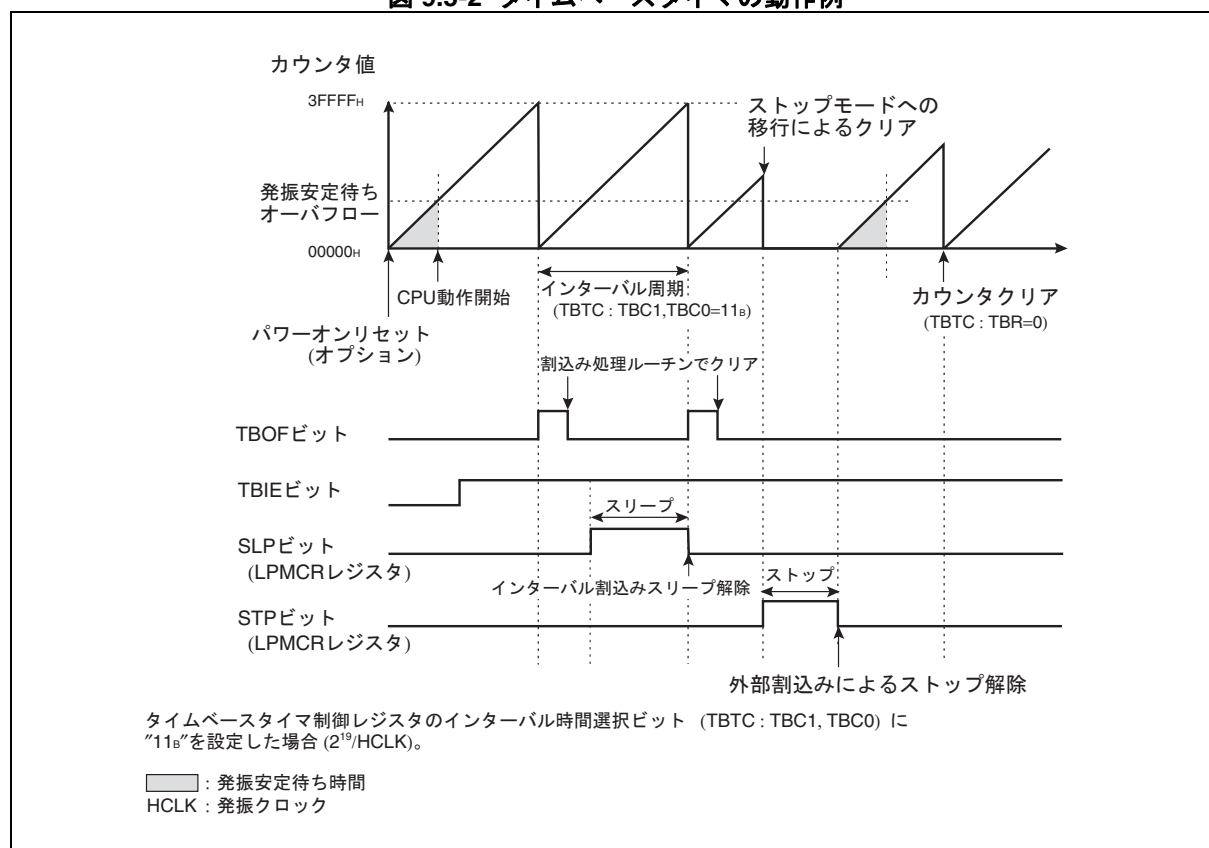
■ タイムベースタイマの動作例

次の状態の動作を図 9.5-2 に示します。

- パワーオンリセットが発生した場合
- インターバルタイマ機能の動作中にスリープモードへ移行した場合
- ストップモードへ移行した場合
- カウンタクリアの要求があった場合

ストップモードへの移行でタイムベースタイマはクリアされ、動作を停止します。ストップモードからの復帰時にタイムベースタイマで発振安定待ち時間をカウントします。

図 9.5-2 タイムベースタイマの動作例



9.6 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のクリアによる周辺機能への影響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 (TBTC) の割込み要求フラグビット (TBOF) をクリアする場合には、割込み要求許可ビット (TBIE), またはプロセッサステータス (PS) のインタラプトレベルマスキングレジスタ (ILM) でタイムベースタイマ割込みが禁止されている状態で行うようにしてください。

● 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のカウンタのクリアで、次の動作は影響を受けます。

- タイムベースタイマでインターバルタイマ機能 (インターバル割込み) を使用している場合、タイムベースタイマのクリアによって、インターバル周期が長くなる場合があります。
- ウォッチドッグタイマを使用している場合、タイムベースタイマのクリアによって、ウォッチドッグタイマのインターバル時間が設定より長くなる場合があります。

● 発振安定待ち時間用タイマとしての使用

電源投入時、メインストップモード中などでは、発振クロックが停止しているため、発振器が動作を開始した後、タイムベースタイマが供給する動作クロックを使用して発振クロックの発振安定待ち時間をとります。高速発振端子に接続する振動子の種類によって、適切な発振安定待ち時間を選択する必要があります。詳細については、「5.5 発振安定待ち時間」を参照してください。

● タイムベースタイマからクロックを供給される周辺機能に対する注意

- メインクロックが停止するモードでは、カウンタはクリアされ、タイムベースタイマは動作を停止します。
- 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カウンタがクリアされると、初期状態からの供給となるため、"H" レベルが短く、または "L" レベルが最大で 1/2 周期長くなる場合があります。
- ウォッチドッグタイマ用のクロックも初期状態からの供給となりますが、ウォッチドッグタイマのカウンタが同時にクリアされるためウォッチドッグタイマは正常な周期で動作します。

9.7 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

● 処理仕様

$2^{12}/\text{HCLK}$ (発振クロック)のインターバル割込みを繰り返し発生します。この場合のインターバル時間は、約 1.0ms(4MHz 動作時)となります。

● コーディング例

```

ICR12 EQU 0000BCH      ; タイムベースタイマ用割込み制御レジスタ
TBTC EQU 0000A9H       ; タイマ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) などは初期化済み
;                               ; とする
;                               ; 割込みディセーブル
AND CCR, #0BFH          ; 割込みレベル 0(最も高い)
MOV I:ICR12, #00H        ; 上位 3 ビットは固定
MOV I:TBTC, #10010000B   ; 割込み許可, TBOF クリア
; カウンタクリア
; インターバル時間  $2^{12}/\text{HCLK}$  選択
; PS 内 ILM をレベル 7 に設定
; 割込みイネーブル
LOOP: MOV ILM, #07H
      OR CCR, #40H
      MOV A, #00H
      MOV A, #01H
      BRA LOOP
;----- 割込みプログラム -----
WARI:
      CLR BIT BOF        ; 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 0FF6CH          ; 割込みベクタを設定
      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10章

ウォッチドッグタイマ

ウォッチドッグタイマの概要，制御レジスタ，構成，動作，使用上の注意およびプログラム例について説明します。

10.1 ウォッチドッグタイマの概要

10.2 ウォッチドッグタイマ制御レジスタ (WDTC)

10.3 ウォッチドッグタイマの構成

10.4 ウォッチドッグタイマの動作

10.5 ウォッチドッグタイマ使用上の注意

10.6 ウォッチドッグタイマのプログラム例

10.1 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、または時計タイマの出力をカウントクロックとする 2 ビットのカウンタです。起動後、一定時間内にクリアされない場合、CPU をリセットします。

■ ウォッチドッグタイマ機能

ウォッチドッグタイマは、プログラム暴走対策用のカウンタです。一度起動すると一定時間内で定期的にクリアし続ける必要があります。プログラムが無限ループに陥るなどして、一定時間以上クリアされない場合、CPU に対してウォッチドッグリセットが発生します。表 10.1-1 に示すように、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制御レジスタ (WDTC) の WT1, WT0 ビットで設定できます。ウォッチドッグタイマがクリアされない場合、最小時間から最大時間の間にウォッチドッグリセットが発生します。この表の最小時間内にカウンタをクリアしてください。クロックソースの出力先は時計タイマ制御レジスタのウォッチドッグクロック選択ビット (WTC:WDSCS) で設定します。

表 10.1-1 ウォッチドッグタイマのインターバル時間

WT1	WT0	WDSCS	SCM	インターバル時間		クロックサイクル数
				最小*	最大*	
0	0	1	1	約 3.58ms	約 4.61ms	$(2^{14} \pm 2^{11})/\text{HCLK}$ サイクル
0	1	1	1	約 14.33ms	約 18.43ms	$(2^{16} \pm 2^{13})/\text{HCLK}$ サイクル
1	0	1	1	約 57.23ms	約 73.73ms	$(2^{18} \pm 2^{15})/\text{HCLK}$ サイクル
1	1	1	1	約 458.75ms	約 589.82ms	$(2^{21} \pm 2^{18})/\text{HCLK}$ サイクル
0	0	上記以外の 組合せ		約 0.457s	約 0.576s	$(2^{12} \pm 2^9)/\text{SCLK}$ サイクル
0	1			約 3.584s	約 4.608s	$(2^{15} \pm 2^{12})/\text{SCLK}$ サイクル
1	0			約 7.168s	約 9.216s	$(2^{16} \pm 2^{13})/\text{SCLK}$ サイクル
1	1			約 14.336s	約 18.432s	$(2^{17} \pm 2^{14})/\text{SCLK}$ サイクル

*: 発振クロック (HCLK) が 4MHz 動作時で、サブクロック (SCLK) 32kHz の 4 分周 (=8kHz) 動作時の値です。

ウォッチドッグタイマインターバル時間の最大、最小および発振クロックサイクル数は、クリアのタイミングで決まります。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の供給クロック) 周期の 3.5 ~ 4.5 倍になります。

ウォッチドッグタイマインターバル時間については、「10.4 ウォッチドッグタイマの動作」を参照してください。

<注意事項>

ウォッチドッグタイマは、タイムベースタイマの桁上り信号をカウントする 2 ビットカウンタによって構成されています。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発生時間が、設定した時間より長くなることがあります。マシクロックとしてサブクロックを使用する場合は、必ず時計タイマ制御レジスタ (WTC) のウォッチドッグタイマクロックソース選択ビット (WDSCS) を "0" に設定して、時計タイマの出力を選択するようにしてください。

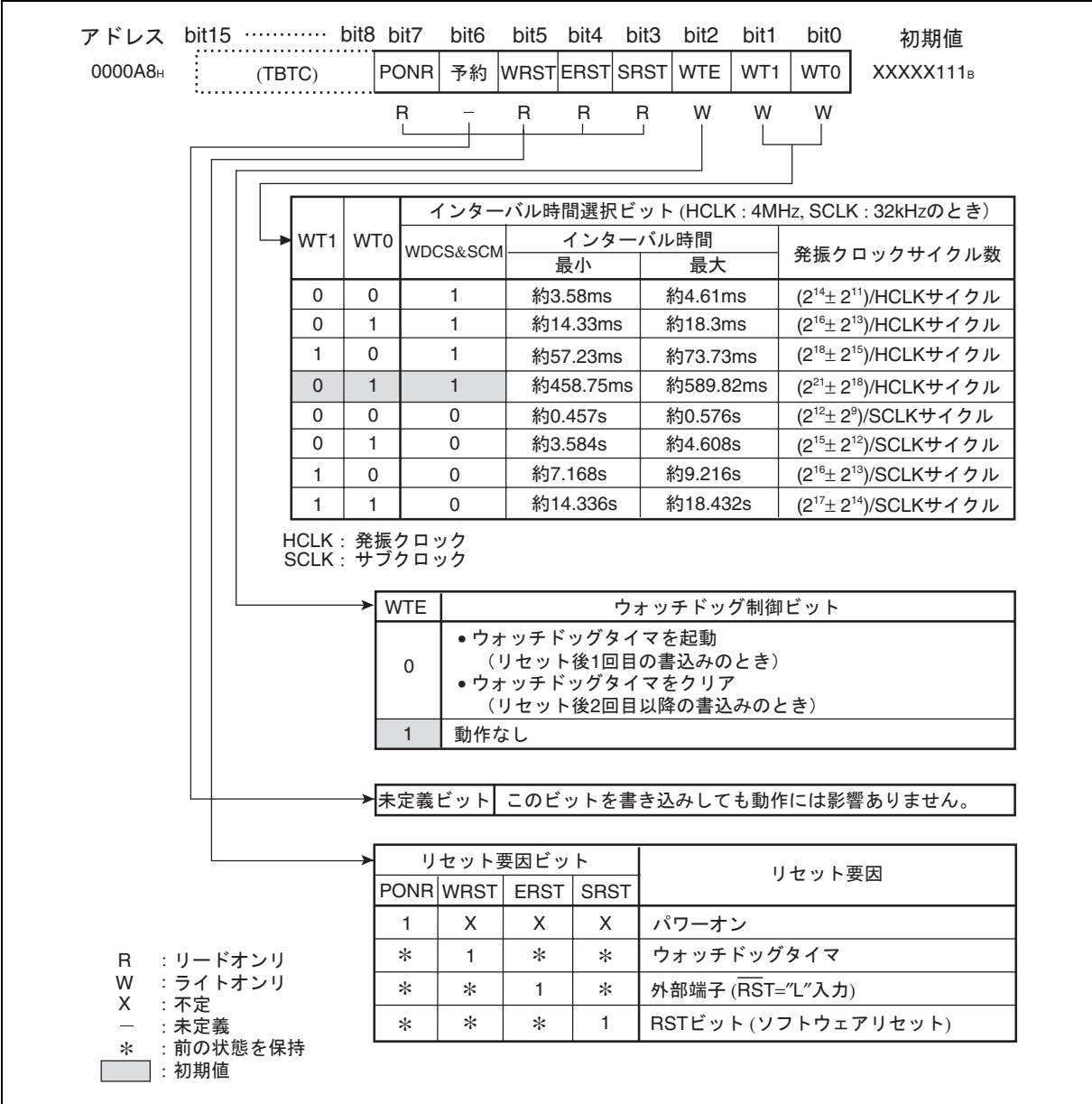
10.2 ウォッチドッグタイマ制御レジスタ (WDTC)

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、クリアおよびリセット要因の表示を行うレジスタです。

■ ウォッチドッグタイマ制御レジスタ (WDTC)

図 10.2-1 にウォッチドッグタイマ制御レジスタ (WDTC) を、表 10.2-1 に WDTC レジスタの各ビットの機能について説明します。

図 10.2-1 ウォッチドッグタイマ制御レジスタ (WDTC)



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の出力値) 周期の 3.5 ~ 4.5 倍となります。詳細は「10.4 ウォッチドッグタイマの動作」を参照してください。

表 10.2-1 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

ビット名			機 能
bit7, bit6, bit5, bit4, bit3	PONR 予約 WRST ERST SRST	リセット要因 ビット	リセット要因を示す読出し専用ビットです。各リセット要因が発生すると、これらのビットが "1" にセットされます。 <ul style="list-style-type: none"> これらのビットは、WDTC レジスタの読出し動作後に、すべて "0" にクリアされます。 パワーオン時は、PONR ビット以外のビットの内容は保証されません。したがって、PONR ビットが "1" の場合は、これ以外のビットの内容を無視するようにしてください。
bit2	WTE	ウォッチドッグ 制御ビット	ウォッチドッグタイマを起動またはクリアします。 <ul style="list-style-type: none"> このビットに "0" を書き込むと、ウォッチドッグタイマを起動 (リセット後 1 回目の書き込み)、または 2 ビットカウンタをクリア (リセット後 2 回目以降の書き込み) します。 "1" の書き込みでは、動作に影響はありません。
bit1, bit0	WT1 WT0	インターバル時 間選択ビット	ウォッチドッグタイマのインターバル時間を選択するビットです。クロックモードとしてサブクロックモードが選択されているか (クロック選択レジスタ (CKSCR) のサブクロック表示ビット (SCM) が "0")、または時計タイマ制御レジスタ (WTC) により、ウォッチドッグタイマのクロックソースが時計タイマになっている場合 (ウォッチドッグタイマクロックソース選択ビット (WDSCS) が "0") と、クロックモードとしてメインクロックモード、または PLL クロックモードが選択されていて、かつ WTC の WDSCS ビットが "1" の場合、インターバル時間は図 10.2-1 のように異なります。 <ul style="list-style-type: none"> ウォッチドッグタイマの起動時のデータのみが有効です。 ウォッチドッグタイマ起動後の書き込みデータは、無視されます。 これらのビットは、書き込み専用です。

10.3 ウォッチドッグタイマの構成

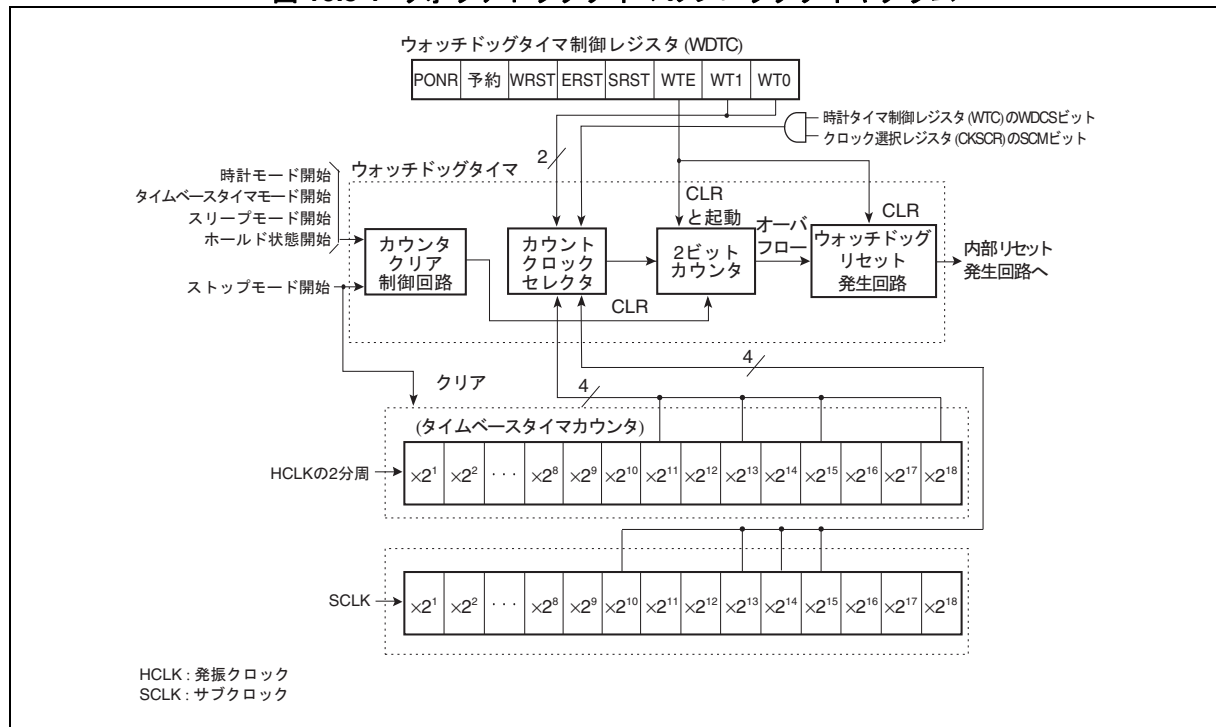
ウォッチドッグタイマは、以下のブロックで構成されています。

- 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発生回路
- 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

図 10.3-1 に、ウォッチドッグタイマのブロックダイアグラムを示します。

図 10.3-1 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

ウォッチドッグタイマのカウントクロックを、4種類のタイムベースタイマ出力と4種類の時計タイマ出力から選択する回路です。これによりウォッチドッグリセットの発生時間が決まります。

● ウォッチドッグカウンタ (2 ビットカウンタ)

タイムベースタイマ出力をカウントクロックとする2ビットのアップカウンタです。

第 10 章 ウォッチドッグタイマ

- 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のオーバフローによって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、カウンタの動作 / 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の起動およびクリアと、リセット発生要因の保持を行います。

10.4 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で、ウォッチドッグリセットが発生します。

■ ウォッチドッグタイマの動作

ウォッチドッグタイマの動作には、図 10.4-1 に示すような設定が必要です。

図 10.4-1 ウォッチドッグタイマの設定

アドレス	bit15	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000A8 _H	WDTC (TBTC)			PONR	予約	WRST	ERST	SRST	WTE	WT1	WT0
									0	◎	◎
◎ : 使用ビット											
0 : "0"を設定											

● ウォッチドッグタイマの起動

ウォッチドッグタイマ制御レジスタ (WDTC) のウォッチドッグ制御ビット (WTE) に、リセット後 1 回目の "0" を書き込むと、ウォッチドッグタイマは起動します。この場合、WDTC レジスタのインターバル時間選択ビット (WT1, WT0) を用いてインターバル時間を同時に指定します。

● ウォッチドッグタイマのクリア

WTE ビットへの、2 回目以降の "0" の書込みで、ウォッチドッグタイマの 2 ビットカウンタをクリアします。インターバル時間内にカウンタがクリアされない場合、カウンタがオーバフローし、ウォッチドッグリセットが発生します。

リセットの発生とスリープモード、ストップモード、タイムベースタイマモード、時計モードへの遷移によって、ウォッチドッグタイマはクリアされます。

<注意事項>

- タイムベースタイマモードおよび時計モードへ遷移したときに、一度、ウォッチドッグカウンタをクリアしますが、クリア後ウォッチドッグカウンタは停止しませんので注意してください。
- 1 系統品でウォッチドッグのクロックとして時計タイマを設定した場合、ウォッチドッグタイマは使用できません。

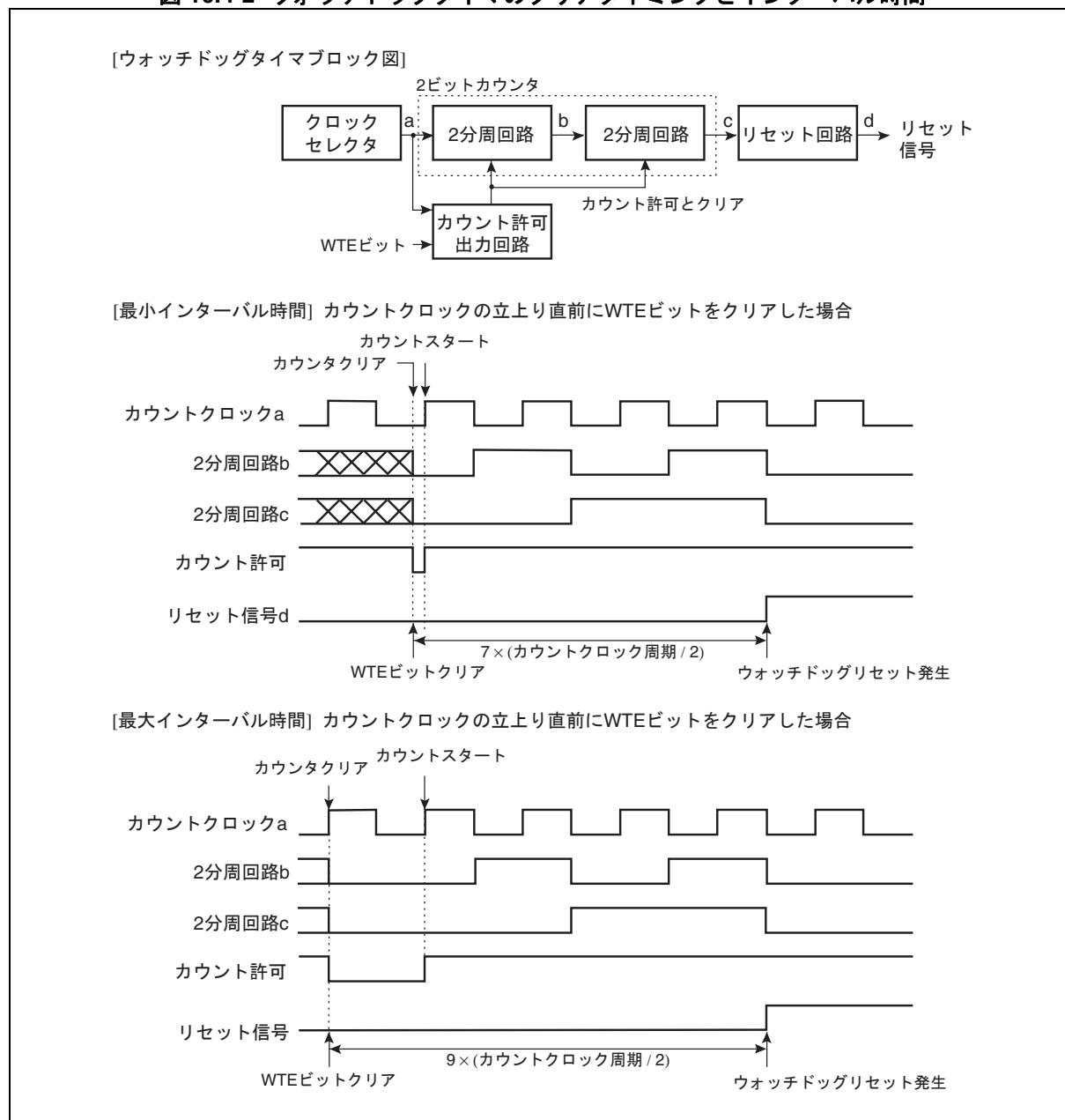
● ウォッチドッグタイマのインターバル時間

図 10.4-2 に、ウォッチドッグタイマのクリアのタイミングとインターバル時間の関係を示します。インターバル時間は、ウォッチドッグタイマをクリアするタイミングによって変化し、カウントクロック周期の 3.5 倍～4.5 倍の時間を要します。

● リセット要因の確認

リセット後 WDTC レジスタのリセット要因ビット (PONR, WRST, ERST, SRST) をチェックすることで、リセット要因が分かります。

図 10.4-2 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



10.5 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時の注意点を示します。

■ 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すべてのリセット要因で停止します。

● インターバル時間

インターバル時間は、タイムベースタイマの桁上り信号をカウントクロックとしているため、タイムベースタイマのクリアによって、ウォッチドッグタイマのインターバル時間が設定より長くなることがあります。

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" の書込み時以外に、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、サブクロックモードからメインクロックモードへの遷移時、サブ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

インターバル時間は、ウォッチドッグタイマの起動と同時に設定してください。起動時以外の書込みデータは無視されます。

● プログラム作成上の注意

メインループの中で、繰り返しウォッチドッグタイマをクリアするようなプログラムを作成する場合、割込み処理を含めたメインループの処理時間が、ウォッチドッグタイマのインターバル時間の最小時間以下となるように設定してください。

● サブクロックモード時の注意

サブクロックモードの場合、必ず時計タイマ制御レジスタ (WTC) のウォッチドッグクロック選択ビット (WDCS) を "0" に設定して、時計タイマの出力を選択してください。

10.6 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

● 処理仕様

- メインプログラムのループの中で毎回ウォッチドッグタイマをクリアします。
- 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 周する必要があります。

● 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) などは初期化済み
; ; ;とする

WDG_START:
MOV WDTC, #00000011B ;ウォッチドッグタイマの起動
; ; ;インターバル時間  $2^{21} \pm 2^{18}$  サイクルを選択
;----- メインループ -----
MAIN: CLRB I:WTE ;ウォッチドッグタイマのクリア
; ; ;定期的に 2 ビットのクリア
; ; ;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
; ; ;より短い時間でループする

CODE ENDS
;----- ベクタ設定 -----
VECT CSEG ABS=0FFH
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11章

時計タイマ

時計タイマの概要，構成，制御レジスタおよび動作について説明します。

- 11.1 時計タイマの概要
- 11.2 時計タイマの構成
- 11.3 時計タイマ制御レジスタ (WTC)
- 11.4 時計タイマの動作

11.1 時計タイマの概要

時計タイマは、サブクロックを使用した 15 ビットのタイマです。インターバル時間ごとに割込みを発生させます。また、設定により、ウォッチドッグタイマのクロックソースとしても使用できます。

■ 時計タイマの機能

時計タイマは 15 ビットのタイマと、インターバル時間ごとの割込みを制御する回路から構成されています。

時計タイマは、クロック選択レジスタ (CKSCR) の PLL クロック選択ビット (MCS) およびサブクロック選択ビット (SCS) と関係なく、サブクロックを使用します。

時計タイマのインターバル時間は表 11.1-1 に示すとおりです。

表 11.1-1 時計タイマのインターバル時間

WTC2	WTC1	WTC0	インターバル時間 [*]
0	0	0	31.25ms
0	0	1	62.5ms
0	1	0	125ms
0	1	1	250ms
1	0	0	500ms
1	0	1	1.000s
1	1	0	2.000s
1	1	1	設定禁止

^{*}: サブクロック 32kHz の 4 分周 (=8kHz)

11.2 時計タイマの構成

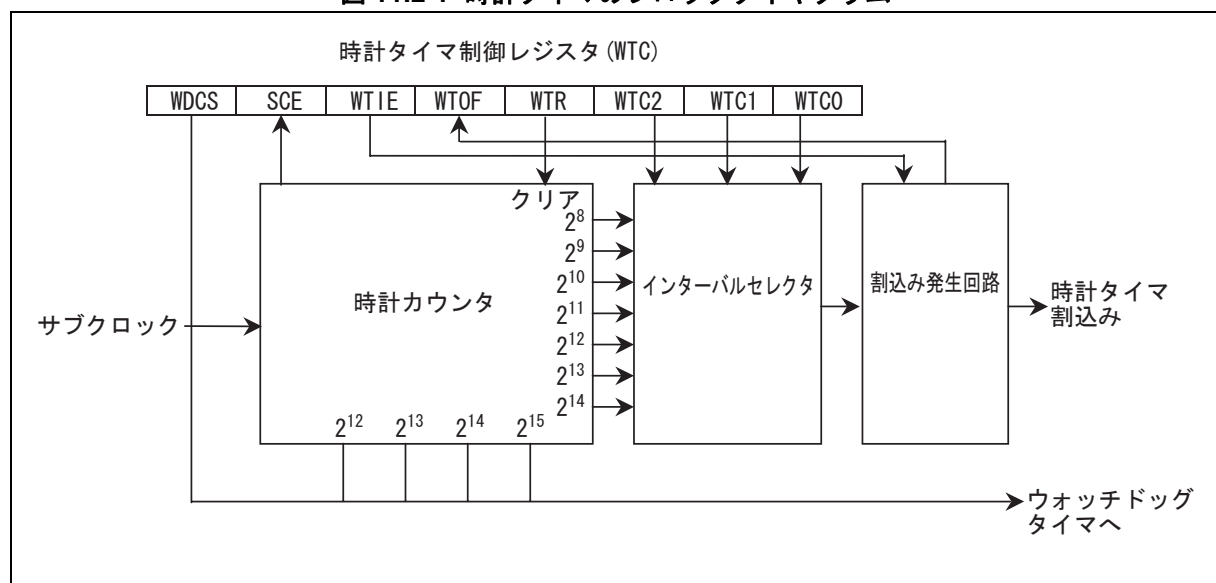
時計タイマは、以下のブロックから構成されています。

- 時計カウンタ
- インターバルセクタ
- 時計タイマ割込み発生回路
- 時計タイマ制御レジスタ (WTC)

■ 時計タイマのブロックダイアグラム

図 11.2-1 に、時計タイマのブロックダイアグラムを示します。

図 11.2-1 時計タイマのブロックダイアグラム



● 時計カウンタ

サブクロックをクロックソースとする 15 ビットのアップカウンタです。

● インターバルセクタ

時計タイマ割込みのインターバル時間を選択するセクタです。

● 割込み発生回路

時計タイマのインターバル割込みを発生させます。

● 時計タイマ制御レジスタ (WTC)

時計タイマの動作、時計タイマ割込みの制御およびウォッチドッグタイマのクロックソースを指定します。

11.3 時計タイマ制御レジスタ (WTC)

時計タイマ制御レジスタ (WTC) は、時計タイマの動作を制御します。また、インターバル割込みの時間を制御します。

■ 時計タイマ制御レジスタ (WTC) の構成

図 11.3-1 に時計タイマ制御レジスタ (WTC) の構成を、表 11.3-1 に時計タイマ制御レジスタ (WTC) の各ビットの機能について説明します。

図 11.3-1 時計タイマ制御レジスタ (WTC) の構成

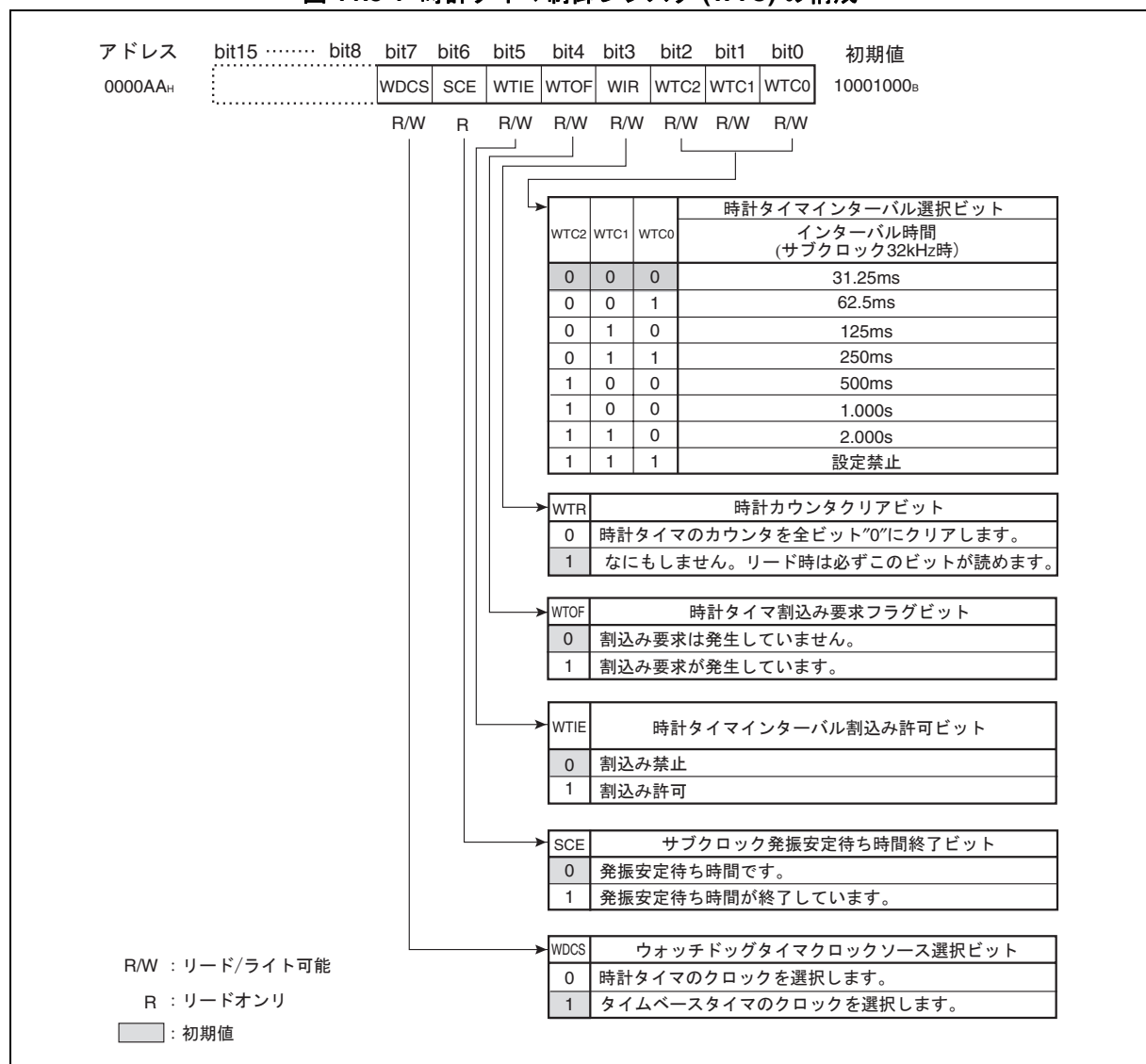


表 11.3-1 時計タイマ制御レジスタ (WTC) の各ビットの機能

ビット名		機 能
bit7	WDCS: ウォッチドッグタイマ クロックソース 選択ビット	ウォッチドッグタイマのクロックソースを選択するビットです。 <ul style="list-style-type: none"> このビットが "0" のとき、時計タイマのクロックを選択し、"1" のとき、タイムベースタイマのクロックを選択します。"1" に設定した状態で、サブクロックモードへ移行した場合は、ウォッチドッグタイマは停止します。 リセットにより "1" に初期化されます。
bit6	SCE: サブクロック発振安定 待ち時間終了ビット	サブクロックの発振安定待ち時間が終了したことを示すビットです。 <ul style="list-style-type: none"> このビットが "0" のとき、発振安定待ち時間であることを示します。 サブクロックの発振安定待ち時間は、2^{14} サブクロックサイクル固定です。 パワーオンリセットおよびストップ時に "0" に初期化します。
bit5	WTIE: 時計タイマインターバ ル割込み許可ビット	時計タイマによるインターバル割込みを許可するビットです。 <ul style="list-style-type: none"> このビットが "1" のとき割込みを許可し、"0" のとき割込みを禁止します。 リセットにより "0" に初期化されます。
bit4	WTOF: 時計タイマ割込み要求 フラグビット	時計タイマの割込み要求が発生していることを示すビットです。 <ul style="list-style-type: none"> WTIE ビットが "1" のときこのビットが "1" になると、割込み要求が発生します。 WTC2 ~ 0 ビットで設定されたインターバルごとに "1" にセットされます。 "0" の書込み、ストップモードへの遷移またはリセットにより "0" にクリアされます。 このビットへの "1" の書込みは意味をもちません。
bit3	WTR: 時計カウンタ クリアビット	時計タイマのカウンタを全ビット "0" にクリアするビットです。 <ul style="list-style-type: none"> このビットに "0" を書き込むと、時計タイマカウンタを "0" にクリアします。 このビットへの "1" 書込みは意味をもちません。 読出し時は、常に "1" が読み出されます。
bit2, bit1, bit0	WTC2, WTC1, WTC0: 時計タイマインターバ ル選択ビット	時計タイマのインターバルを設定するビットです。 <ul style="list-style-type: none"> リセットにより "000_B" に初期化されます。 このビットを変更するときは、WTOF ビットも同時にクリアしてください。

11.4 時計タイマの動作

時計タイマには、時計カウンタ、時計タイマのインターバル割込み機能、ウォッチドッグタイマのクロックソースの指定機能、サブクロックの発振安定待ち機能があります。

■ 時計カウンタ

時計カウンタはサブクロックをカウントする 15 ビットのカウンタにより構成され、サブクロックが入力されている間、常にカウント動作を続けます。

● 時計カウンタのクリア

時計カウンタのクリアは、パワーオンリセット、ストップモードへの遷移および時計タイマ制御レジスタ (WTC) の時計カウンタクリアビット (WTR) への "0" の書込みにより行われます。

<注意事項>

- 時計タイマの出力を使用している、ウォッチドッグタイマとインターバル割込みは、時計カウンタのクリアにより動作に影響を与えます。
 - 時計タイマ制御レジスタ (WTC) の WTR ビットへの "0" の書込みにより時計タイマをクリアする場合は、WTIE ビットに "0" を設定し、時計タイマの割込みを禁止した状態で行ってください。
また、割込みを許可する前に、WTOF フラグへの "0" の書込みによる割込み要求のクリアを行ってください。
-

■ 時計タイマのインターバル割込み機能

時計カウンタの桁上り信号により一定周期で割込みを発生します。

● インターバル時間の指定

WTC レジスタの (WTC2, WTC1, WTC0) ビットにより、インターバル時間を指定できます。

● 時計タイマ割込みの発生

WTC2, 1, 0 ビットで設定されるインターバル時間ごとに時計タイマ割込み要求フラグビット (WTOF) をセットします。このとき、時計タイマインターバル割込み許可ビット (WTIE) が "1" に設定されていて、割込みが許可されていれば、時計タイマ割込みが発生します。

WTOF ビットのセットは、最後に時計タイマがクリアされた時間を基準にして行われます。

ストップモードに遷移すると、時計タイマは、サブクロックの発振安定待ち時間のタイマとして使用されるため、WTOF ビットはモード遷移と同時にクリアされます。

■ ウォッチドッグタイマのクロックソースの指定機能

WTC レジスタのウォッチドッグタイマクロックソース選択ビット (WDCS) により , ウォッチドッグタイマのクロックソースを指定できます。マシンのクロックとしてサブクロックを使用する場合は , 必ず WDCS ビットを "0" に設定して , 時計タイマの出力を選択してください。WDCS ビットを "1" に設定した状態で , サブクロックモードへ移行した場合は , ウォッチドッグタイマは停止します。

■ サブクロックの発振安定時間待ち機能

パワーオンリセット , ストップモードからの復帰時は , 時計タイマは , サブクロックの発振安定時間待ちタイマとして機能します。サブクロックの発振安定待ち時間はサブクロックの 2^{14} サイクルに固定されています。

第12章

16 ビット入出力タイマ

16 ビット入出力タイマの概要，構成，レジスタの構成と機能，割込みおよび動作について説明します。

12.1 16 ビット入出力タイマの概要

12.2 16 ビット入出力タイマの構成

12.3 16 ビット入出力タイマのレジスタの構成と機能

12.4 16 ビット入出力タイマの割込み

12.5 16 ビット入出力タイマの動作

12.6 16 ビット入出力タイマのプログラム例

12.1 16 ビット入出力タイマの概要

16 ビット入出力タイマは、フリーランタイマ 1 本、アウトプットコンペア 6 本、インプットキャプチャ 2 本によって構成されています。フリーランタイマをベースに 6 本の独立した波形出力を得ることができ、入力パルス幅測定、外部クロック周期の測定が可能です。

■ 16 ビット入出力タイマの機能

16 ビット入出力タイマを構成しているフリーランタイマ、アウトプットコンペアおよびインプットキャプチャの機能は、以下のとおりです。

● フリーランタイマ (× 1)

フリーランタイマは 16 ビットのアップカウンタ、コントロールレジスタ、プリスケラから構成されています。

フリーランタイマの出力値は、インプットキャプチャ、アウトプットコンペアの基本時間 (ベースタイマ) として使用します。

- カウント動作のためのクロックは、8 種類から選択できます。
- カウンタオーバフロー割込みを発生できます。
- コンペアクリアレジスタの値と、フリーランタイマの値の一致によるカウンタの初期化が可能です。

● アウトプットコンペア (× 6)

アウトプットコンペアは、6 本の 16 ビットコンペアレジスタ、コンペア出力用ラッチおよびコントロールレジスタから構成されています。フリーランタイマ値とコンペアレジスタ値が一致したとき出力レベルを反転するとともに、割込みを発生できます。

- 6 本のコンペアレジスタを独立して動作させることができます。各コンペアレジスタに対応した出力端子と割込みフラグがあります。
- 2 本のコンペアレジスタをペアにして出力端子を制御できます。
- 出力端子の初期値を設定することができます。

● インプットキャプチャ (× 2)

インプットキャプチャは、独立した 2 本の外部入力端子と対応したキャプチャレジスタ、コントロールレジスタ、エッジ検出回路から構成されています。外部入力端子より入力された信号の任意エッジを検出することにより、フリーランタイマ値をキャプチャレジスタに保持し、同時に割込みを発生します。

- 外部入力信号のエッジを選択可能です。立上りエッジ、立下りエッジまたは両エッジから選択できます。
- 2 本のインプットキャプチャは独立して動作可能です。

割込みは、外部入力信号の有効エッジにより発生可能です。インプットキャプチャは、割込みにより DMA または EI²OS を起動できます。

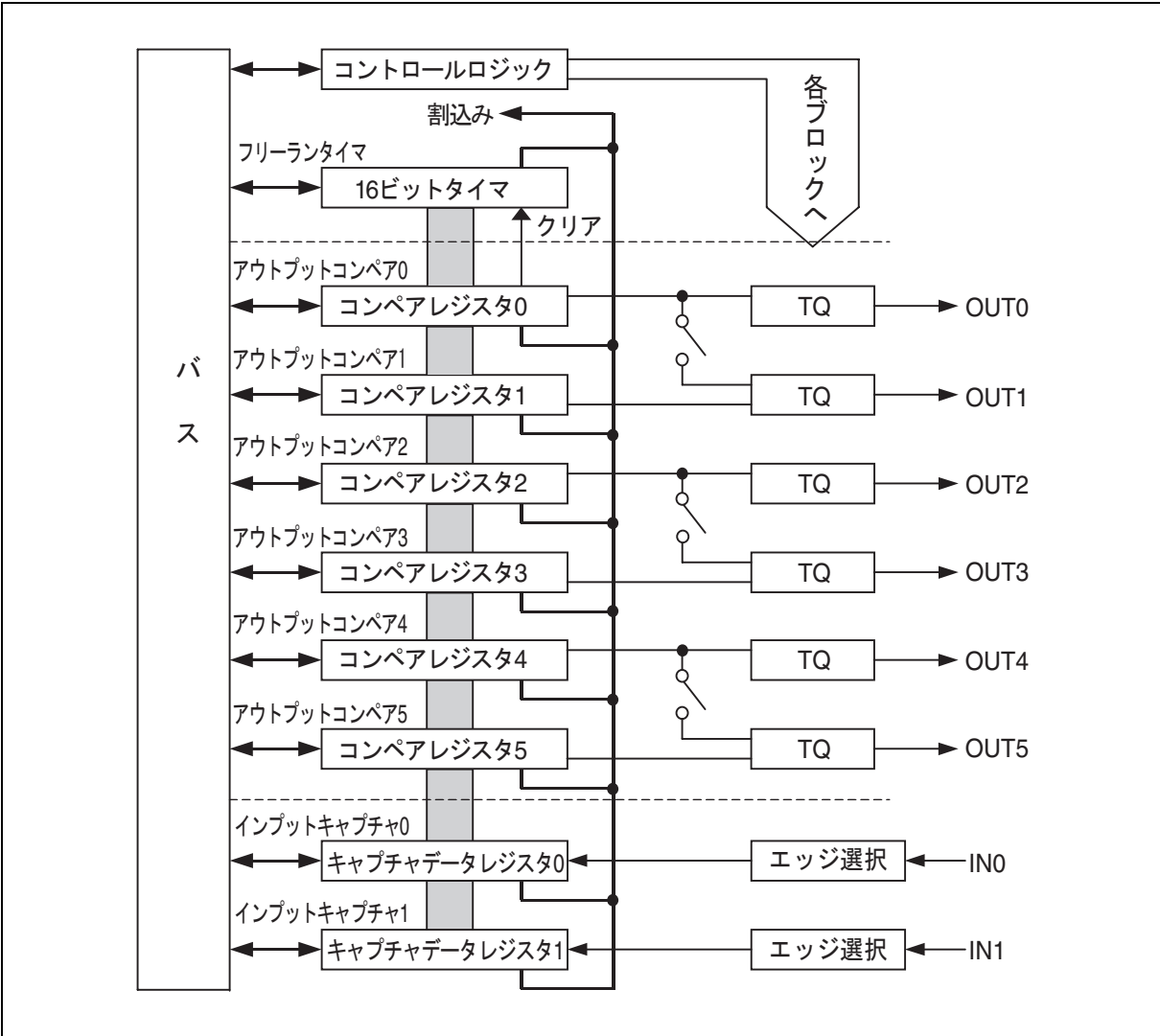
12.2 16 ビット入出力タイマの構成

16 ビット入出力タイマは、フリーランタイマ、アウトプットコンペア、インプットキャプチャの 3 つのモジュールから構成されています。

■ ブロックダイアグラム

図 12.2-1 に、16 ビット入出力タイマのブロックダイアグラムを示します。

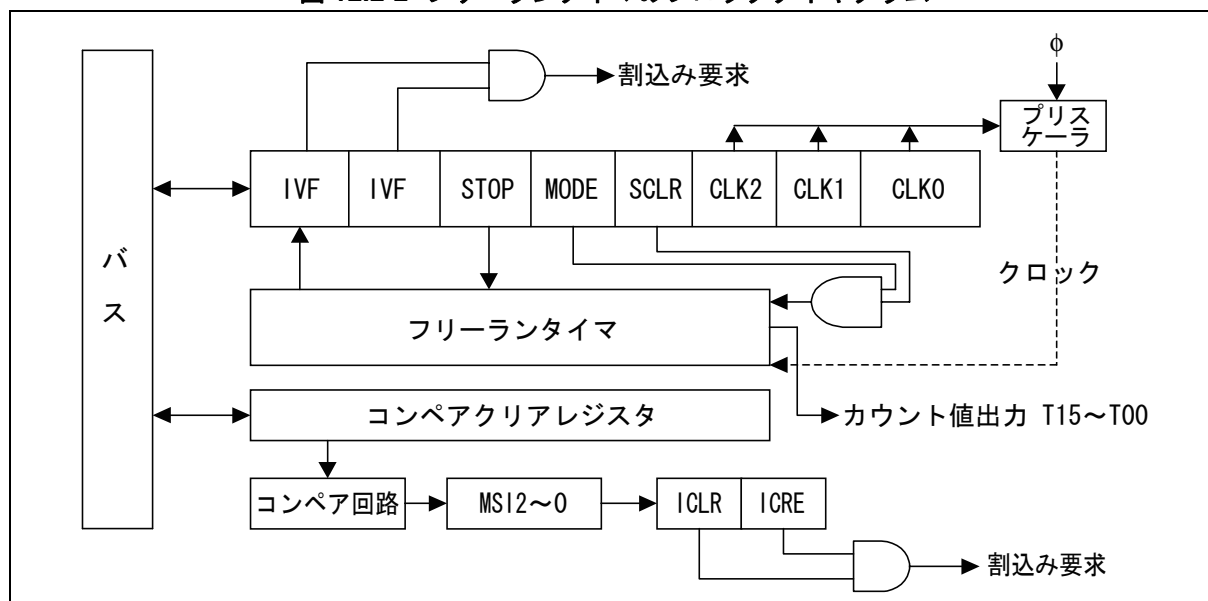
図 12.2-1 16 ビット入出力タイマのブロックダイアグラム



● フリーランタイムのブロックダイアグラム

図 12.2-2 に、フリーランタイマのブロックダイアグラムを示します。

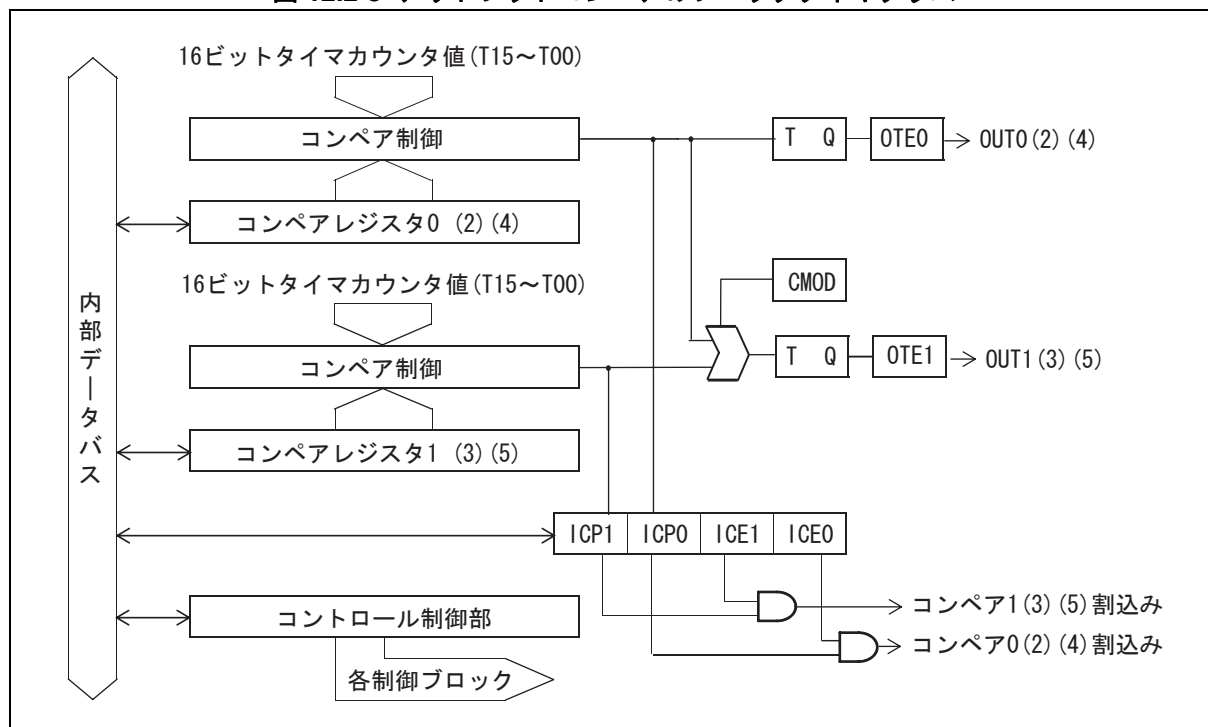
図 12.2-2 フリーランタイマのブロックダイアグラム



● アウトプットコンペアのブロックダイヤグラム

図 12.2-3 に、アウトプットコンペアのブロックダイアグラムを示します。

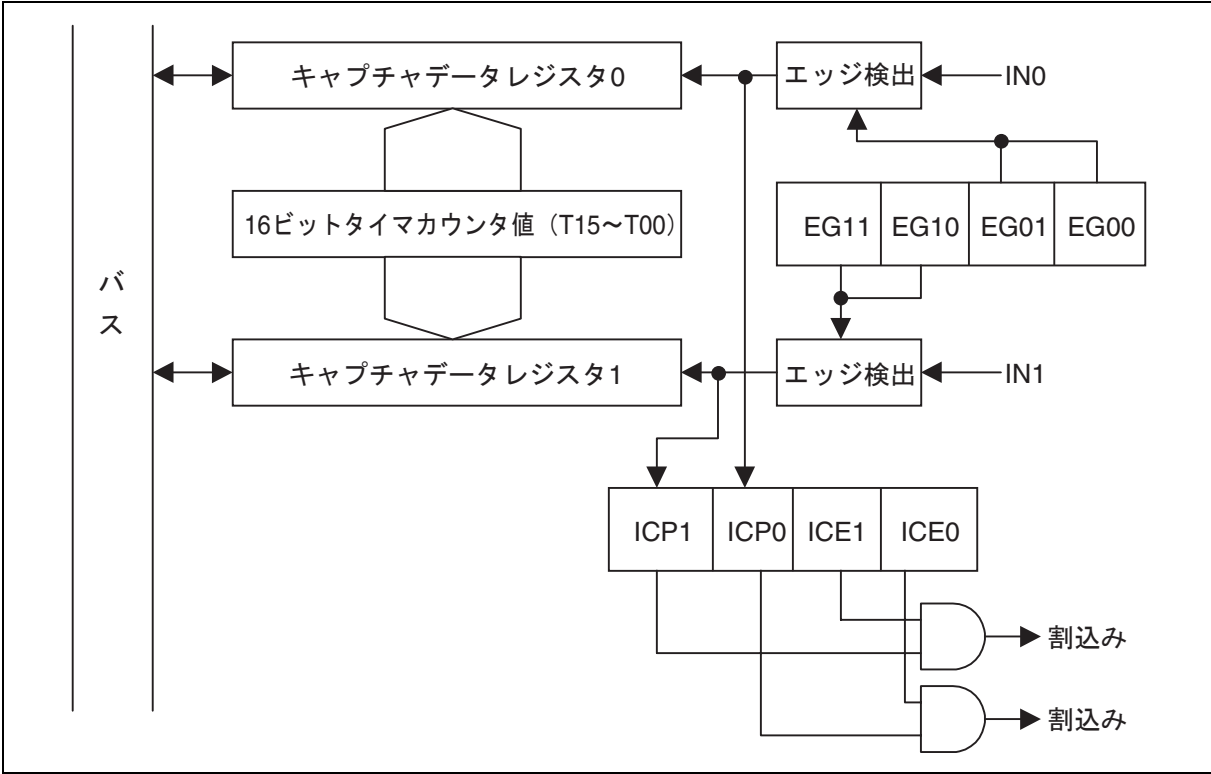
図 12.2-3 アウトプットコンペアのブロックダイアグラム



● インพุットキャプチャのブロックダイアグラム

図 12.2-4 に、インพุットキャプチャのブロックダイアグラムを示します。

図 12.2-4 インพุットキャプチャのブロックダイアグラム



■ 16 ビット入出力タイマに関する端子

16ビット入出力タイマに関する端子は、IN0/IN1 端子と OUT0/OUT1/OUT2/OUT3/OUT4/OUT5 端子があります。IN0/IN1 端子は、汎用入出力ポート (P96/IN0, P97/IN1) と、インพุットキャプチャの入力端子として機能を兼用しており、OUT0/OUT1/OUT2/OUT3/OUT4/OUT5 端子は、汎用入出力ポート (PA0/OUT0, PA1/OUT1, PA2/OUT2, PA3/OUT3, P46/OUT4, P47/OUT5) と、アウトプットコンペア出力端子を兼用しています。

● IN0/IN1 端子として使用する場合の設定

IN0/IN1 端子として使用する場合は、P96/IN0, P97/IN1 端子は、ポート方向レジスタを入力ポート (DDR9 bit15, 14 → "0") に設定してください。

● OUT0/OUT1/OUT2/OUT3/OUT4/OUT5 端子として使用する場合の設定

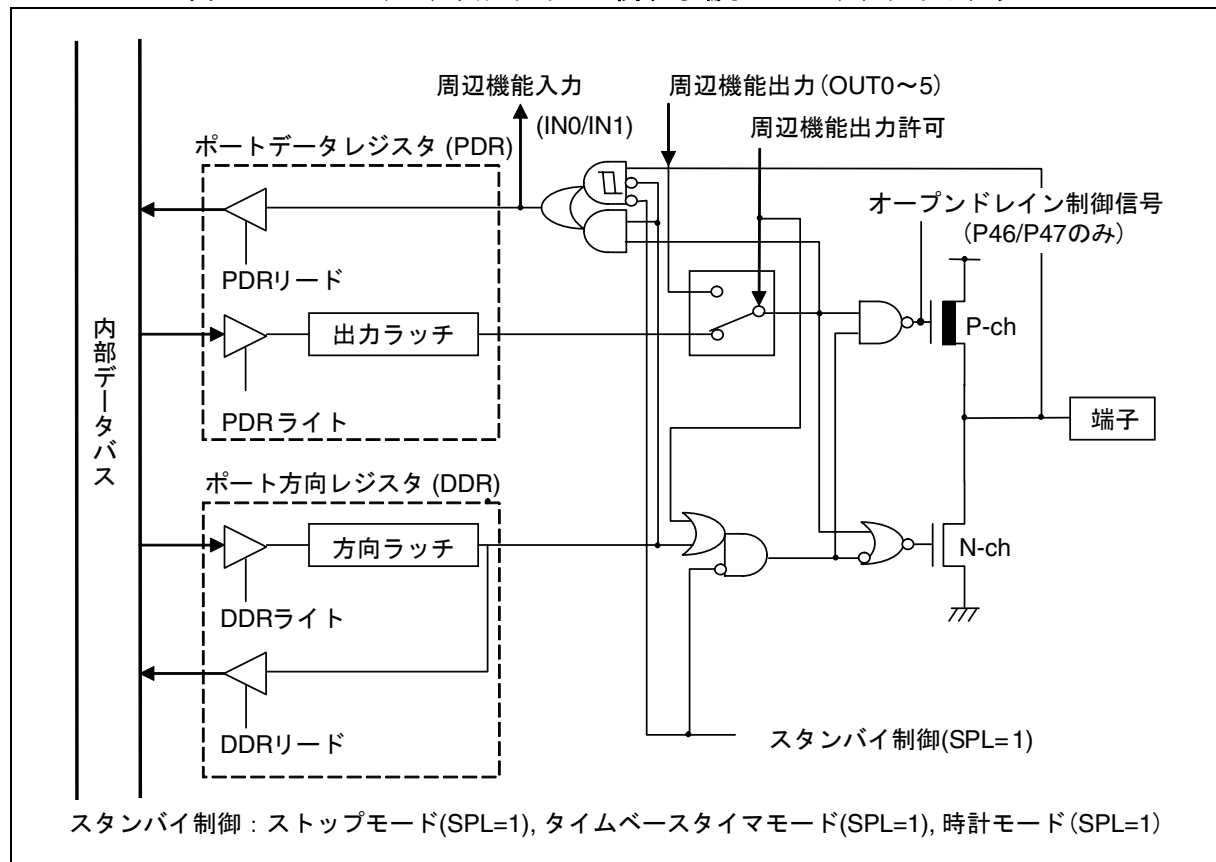
OUT0/OUT1/OUT2/OUT3/OUT4/OUT5 端子を出力として使用する場合は、コントロールレジスタ (OCS01/23/45) をアウトプットコンペア端子出力 (OCS01/23/45 bit10, 11 → "1") に設定してください。

● FRCK 端子として使用する場合の設定

FRCK 端子として使用する場合は、P93/FRCK 端子は、ポート方向レジスタを入力ポート (DDR9 bit11 → "0") に設定してください。

■ 16 ビット入出力タイマに関する端子のブロックダイアグラム

図 12.2-5 16 ビット入出力タイマに関する端子のブロックダイアグラム

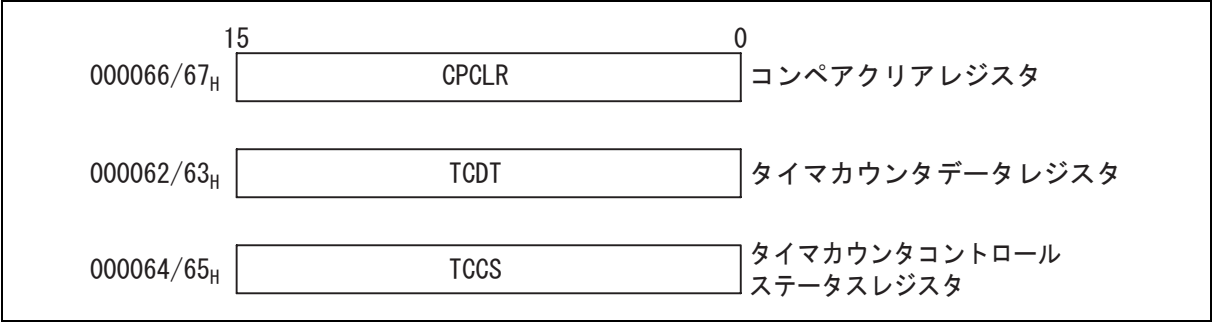


12.3 16 ビット入出力タイマのレジスタの構成と機能

16 ビット入出力タイマのレジスタの構成と機能を示します。

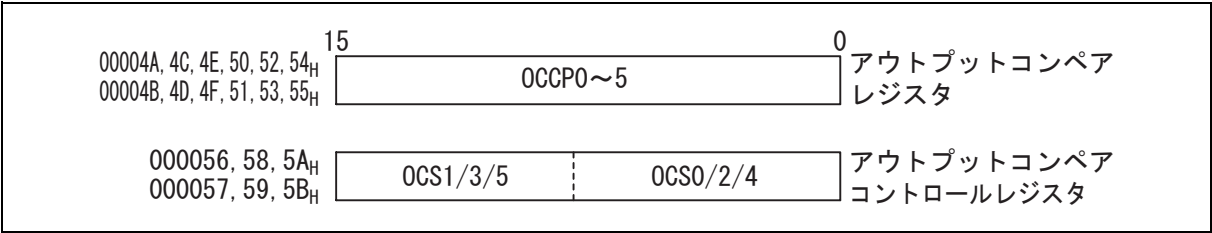
● フリーランタイマ

図 12.3-1 フリーランタイマのレジスタ構成



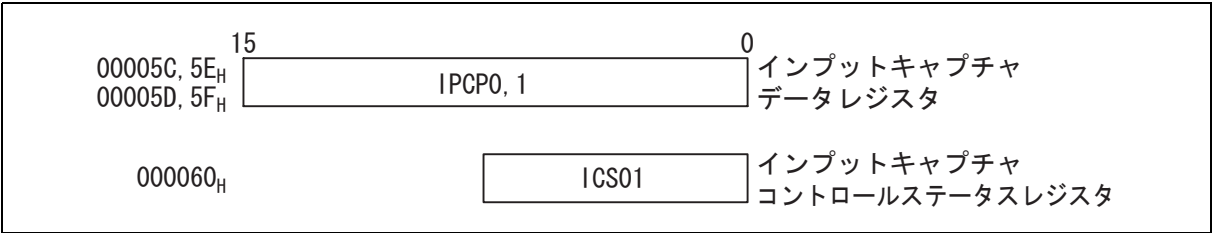
● アウトプットコンペア

図 12.3-2 アウトプットコンペアのレジスタ構成



● インプットキャプチャ

図 12.3-3 インプットキャプチャのレジスタ構成



12.3.1 フリーランタイマ

フリーランタイマのレジスタの構成と機能について説明します。

■ フリーランタイマのレジスタ一覧

図 12.3-4 に、フリーランタイマのレジスタ一覧を示します。

図 12.3-4 フリーランタイマのレジスタ一覧

	15	14	13	12	11	10	9	8	CPCLR
000067 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	コンペアクリアレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値XXXXXXXX _B
	7	6	5	4	3	2	1	0	CPCLR
000066 _H	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	コンペアクリアレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値XXXXXXXX _B
	15	14	13	12	11	10	9	8	TCDT
000063 _H	T15	T14	T13	T12	T11	T10	T09	T08	タイマカウンタ データレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
	7	6	5	4	3	2	1	0	TCDT
000062 _H	T07	T06	T05	T04	T03	T02	T01	T00	タイマカウンタ データレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
	15	14	13	12	11	10	9	8	TCCS
000065 _H	ECKE	—	—	MS12	MS11	MS10	ICLR	ICRE	タイマカウンタ コントロールステータスレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値0—00000 _B
	7	6	5	4	3	2	1	0	TCCS
000064 _H	IVF	IVFE	STOP	MODE	SCLR	CLK2	CLK1	CLK0	タイマカウンタ コントロールステータスレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B

■ コンペアクリアレジスタ (CPCLR)

コンペアクリアレジスタ (CPCLR) のビット構成を、下図に示します。

図 12.3-5 コンペアクリアレジスタ (CPCLR) のビット構成

	15	14	13	12	11	10	9	8	CPCLR
000067 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	コンペアクリアレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値XXXXXXXX _B
	7	6	5	4	3	2	1	0	CPCLR
000066 _H	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	コンペアクリアレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値XXXXXXXX _B

コンペアクリアレジスタ (CPCLR) は、フリーランタイマと比較を行うための 16 ビット長のコンペアレジスタです。本レジスタは、初期値不定です。動作を許可する場合は、値を設定してから割込み動作を許可してください。また、本レジスタはワードアクセスしてください。

タイマカウンタコントロールステータスレジスタ (TCCS) の MODE ビットに "1" を設定している場合は、本レジスタ値とフリーランタイマ値が一致したとき、フリーランタイマ値を "0000_H" に初期化します。また、本レジスタ値とフリーランタイマ値が一致したとき、コンペアクリア割込みフラグをセットします。コンペアクリア割込みフラグが

"1" のとき、割込み動作が許可されている場合は、CPU に対して割込み要求が行われます。

■ タイマカウンタデータレジスタ (TCDT)

タイマカウンタデータレジスタ (TCDT) のビット構成を、下図に示します。

図 12.3-6 タイマカウンタデータレジスタ (TCDT) のビット構成

000063 _H	15	14	13	12	11	10	9	8	TCDT
	T15	T14	T13	T12	T11	T10	T09	T08	タイマカウンタ データレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
000062 _H	7	6	5	4	3	2	1	0	TCDT
	T07	T06	T05	T04	T03	T02	T01	T00	タイマカウンタ データレジスタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B

タイマカウンタデータレジスタ (TCDT) は 16 ビットのアップカウンタで、フリーランタイマのカウント値を読み出すことができるレジスタです。カウント値はリセット時に "0000" にクリアされます。このレジスタに書き込むことによって、タイマ値を設定できますが、必ず停止 (STOP=1) 状態で書き込んでください。

このレジスタは、ワードアクセスしてください。フリーランタイマの初期化は、次の要因で行われます。

- リセットによる初期化
- タイマカウンタコントロールステータスレジスタ (TCCS) のクリアビット (SCLR) による初期化
- コンペアクリアレジスタ (CPCLR) とタイマカウンタ値の一致による初期化 (TCCS:MODE = 1)

■ タイマカウンタコントロールステータスレジスタ (TCCS)

タイマカウンタコントロールステータスレジスタ (TCCS) のビット構成を、下図に示します。

図 12.3-7 タイマカウンタコントロールステータスレジスタ (TCCS) のビット構成

000065 _H	15	14	13	12	11	10	9	8	TCCS
	ECKE	—	—	MS12	MS11	MS10	ICLR	ICRE	タイマカウンタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	コントロールステータスレジスタ
000064 _H	7	6	5	4	3	2	1	0	TCCS
	IVF	IVFE	STOP	MODE	SCLR	CLK2	CLK1	CLK0	タイマカウンタ
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	コントロールステータスレジスタ

以下に、タイマカウンタコントロールステータスレジスタ (TCCS) の各ビットの機能を説明します。

[bit15] ECKE

このビットは、フリーランタイマのカウンタクロックソースが内部か外部を選択します。このビットに書き込んだ後すぐにクロックは変更されますので、アウトプットコンペア、インプットキャプチャが停止状態のときに変更してください。

0	内部クロックソースを選択 (初期値)
1	外部端子 (FRCK) より入力されたクロックを選択

[bit14, bit13] 未使用ビット

これらのビットは、未使用ビットです。

[bit12, bit11, bit10] MSI2, MSI1, MSIO

これらのビットは、コンペアクリア割込みをマスクする回数を設定します。3 ビットのリロードカウンタで構成され、カウンタ値が "000" になるたびにカウンタ値をリロードします。また、本レジスタへの書き込み時にもカウンタ値をロードします。マスク回数=設定回数となります(例:2回マスクし3回目で割込み処理をする場合, "010" に設定)。ただし, "000" を設定すると, 割込み要因は, マスクされません。

[bit9] ICLR

このビットは、コンペアクリアの割込み要求フラグです。コンペアクリアレジスタ値とフリーランタイマ値が比較結果一致したときにこのビットが "1" にセットされます。割込み要求許可ビット (bit8 の ICRE ビット) がセットされていると、割込みが発生します。本ビットは, "0" を書き込むことによりクリアされます。"1" の書き込みは意味を持ちません。リードモディファイライト系命令では, 常に "1" が読み出されます。

0	割込み要求なし (初期値)
1	割込み要求あり

[bit8] ICRE

このビットは、コンペアクリアの割込み許可ビットです。このビットが "1" のとき、割込みフラグ (bit9 の ICLR) が "1" にセットされていると割込みが発生します。

0	割込み禁止 (初期値)
1	割込み許可

[bit7] IVF

このビットは、フリーランタイマの割込み要求フラグです。

フリーランタイマがオーバフローを起こしたとき、またはモード設定状態でコンペアクリアレジスタとフリーランタイマの比較結果が一致してカウンタクリアされたとき、IVF ビットが "1" にセットされます。割込み要求許可ビット (bit5 の IVFE) がセットされていると、割込みが発生します。本ビットは "0" に設定した場合クリアされます。"1" に設定した場合は意味を持ちません。リードモディファイライト系命令では, "1" が読み出されます。

0	割込み要求なし (初期値)
1	割込み要求あり

[bit6] IVFE

このビットは、フリーランタイマの割込み許可ビットです。このビットが "1" のとき、書込みフラグ (bit5 の IVF) が "1" にセットされると割込みが発生します。

0	割込み禁止 (初期値)
1	割込み許可

[bit5] STOP

このビットは、フリーランタイマのカウントの許可または禁止を設定します。このビットは "1" を書き込んだ時にタイマのカウントを停止し、"0" を書き込んだ時にタイマのカウントを開始します。

0	カウント許可 (動作) (初期値)
1	カウント禁止 (停止)

なお、フリーランタイマのカウントが停止すると、アウトプットコンペア動作も停止します。

[bit4] MODE

このビットは、フリーランタイマの初期化条件を設定します。

"0" のときは、リセットとクリアビット (bit3 の SCLR) でカウンタ値を初期化します。

"1" のときは、リセットとクリアビット (bit3 の SCLR) のほかにフリーランタイマとコンペアクリアレジスタ (CPCLR) の値との一致により、カウンタ値を初期化します。

0	リセット、クリアビットによる初期化 (初期値)
1	リセット、クリアビット、コンペアクリアレジスタによる初期化

なお、カウンタ値の初期化はカウント値の変化点で行われます。

[bit3] SCLR

このビットは、動作中のフリーランタイマ値を "0000" に初期化します。

"1" を書き込んだ時にカウンタ値を "0000" に初期化します。"0" を書き込んでも意味を持ちません。読出し値は、常に "0" です。カウンタ値の初期化は、カウンタの変化点に同期して行われます。

0	意味を持ちません。 (初期値)
1	カウンタ値を "0000" に初期化します。

なお、タイマ停止中に初期化する場合は、データレジスタに "0000" を書き込んでください。

<注意事項>

"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。

[bit2, bit1, bit0] CLK2, CLK1, CLK0

これらのビットは、フリーランタイマのカウントクロックを選択します。このビットに書き込み後すぐにクロックは変更されますので、アウトプットコンペア、インプットキャプチャが停止状態のときに変更してください。

CLK2	CLK1	CLK0	カウントクロック	$\phi=20\text{MHz}$	$\phi=16\text{MHz}$	$\phi=8\text{MHz}$	$\phi=4\text{MHz}$	$\phi=1\text{MHz}$
0	0	0	ϕ	50ns	62.5ns	0.125 μs	0.25 μs	1.0 μs
0	0	1	$\phi/2$	100ns	0.125 μs	0.25 μs	0.5 μs	2.0 μs
0	1	0	$\phi/4$	0.2 μs	0.25 μs	0.5 μs	1.0 μs	4.0 μs
0	1	1	$\phi/8$	0.4 μs	0.5 μs	1.0 μs	2.0 μs	8.0 μs
1	0	0	$\phi/16$	0.8 μs	1.0 μs	2.0 μs	4.0 μs	16.0 μs
1	0	1	$\phi/32$	1.6 μs	2.0 μs	4.0 μs	8.0 μs	32.0 μs
1	1	0	$\phi/64$	3.2 μs	4.0 μs	8.0 μs	16.0 μs	64.0 μs
1	1	1	$\phi/128$	6.4 μs	8.0 μs	16.0 μs	32.0 μs	128.0 μs

12.3.2 アウトプットコンペア

アウトプットコンペアのレジスタの構成と機能について説明します。

■ アウトプットコンペアのレジスター一覧

図 12.3-8 に、アウトプットコンペアのレジスタ一覧を示します。

図 12.3-8 アウトプットコンペアのレジスタ一覧

ch. 0	00004B _H	15	14	13	12	11	10	9	8	OC0P0~5
ch. 1	00004D _H									
ch. 2	00004F _H	C15	C14	C13	C12	C11	C10	C09	C08	アウトプットコンペアレジスタ
ch. 3	000051 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 4	000053 _H									
ch. 5	000055 _H									
ch. 0	00004A _H	7	6	5	4	3	2	1	0	OC0P0~5
ch. 1	00004C _H									
ch. 2	00004E _H	C07	C06	C05	C04	C03	C02	C01	C00	アウトプットコンペアレジスタ
ch. 3	000050 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 4	000052 _H									
ch. 5	000054 _H									
ch. 0, ch. 1	000057 _H	15	14	13	12	11	10	9	8	OC0S1/23/45
ch. 2, ch. 3	000059 _H	—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	アウトプットコンペアコントロールレジスタ
ch. 4, ch. 5	00005B _H	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値 ———00000 _B
ch. 0, ch. 1	000056 _H	7	6	5	4	3	2	1	0	OC0S1/23/45
ch. 2, ch. 3	000058 _H	ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0	アウトプットコンペアコントロールレジスタ
ch. 4, ch. 5	00005A _H	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(R/W)	(R/W)	初期値 0000—00 _B

＜注意事項＞

アウトプットコンペアレジスタを書き換える場合は、コンペア割込みのルーチン内で行うか、コンペア動作禁止の状態で行い、比較結果一致と書込みが同時に発生しないようにしてください。

■ アウトプットコンペアレジスタ (OCCP0 ~ 5)

アウトプットコンペアレジスタ (OCCP0 ~ 5) のビット構成を、下図に示します。

図 12.3-9 アウトプットコンペアレジスタ (OCCP0 ~ 5) のビット構成

ch. 0	00004B _H									
ch. 1	00004D _H	15	14	13	12	11	10	9	8	OCCP0~5
ch. 2	00004F _H	C15	C14	C13	C12	C11	C10	C09	C08	Aウトビットコンペアレジスタ
ch. 3	000051 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 4	000053 _H									
ch. 5	000055 _H									
ch. 0	00004A _H									
ch. 1	00004C _H	7	6	5	4	3	2	1	0	OCCP0~5
ch. 2	00004E _H	C07	C06	C05	C04	C03	C02	C01	C00	Aウトビットコンペアレジスタ
ch. 3	000050 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 4	000052 _H									
ch. 5	000054 _H									

アウトプットコンペアレジスタ (OCCP0 ~ 5) は、フリーランタイマと比較を行うための 16 ビット長のコンペアレジスタです。リセットにより初期化されます。本レジスタは、ワードアクセスしてください。本レジスタ値とフリーランタイマ値が一致したとき、コンペア信号が発生してアウトプットコンペア割込みフラグがセットされます。また、出力許可をしている場合は、コンペアレジスタ値に対応した出力レベルが反転出力されます。

■ アウトプットコンペアコントロールレジスタ (OCS01/23/45)

アウトプットコンペアコントロールレジスタ (OCS01/23/45) のビット構成を、下図に示します。

図 12.3-10 アウトプットコンペアコントロールレジスタ (OCS01/23/45) のビット構成

ch. 0, ch. 1	000057 _H	15	14	13	12	11	10	9	8	OCS01/23/45
ch. 2, ch. 3	000059 _H	—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	アウトプットコンペアコントロールレジスタ
ch. 4, ch. 5	00005B _H	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値 —00000 _B

ch. 0, ch. 1	000056 _H	7	6	5	4	3	2	1	0	OCS01/23/45
ch. 2, ch. 3	000058 _H	ICP1	ICP0	ICE1	ICE0	—	—	GST1	GST0	アウトプットコンペアコントロールレジスタ
ch. 4, ch. 5	00005A _H	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(R/W)	(R/W)	初期値 0000—00 _B

以下に、アウトプットコンペアコントロールレジスタ (OCS01/23/45) の各ビットの機能を説明します。

[bit15, bit14, bit13] 未使用ビット

これらのビットは、未使用ビットです。これらのビットには、必ず "0" を書き込んでください。

[bit12] CMOD

このビットは、端子出力を許可した場合 (OTE1=1 OR OTE0=1) の比較結果一致における端子出力レベル反転動作モードを切り換えます。

- CMOD=0 のとき (初期値) には、コンペアレジスタ値に対応したレベルを反転します。
 - OUT0/2/4: コンペアレジスタ 0/2/4 の一致によりレベルを反転します。
 - OUT1/3/5: コンペアレジスタ 1/3/5 の一致によりレベルを反転します。
- CMOD=1 のときには、コンペアレジスタ 0(2/4) は、CMOD=0 時と同じく出力レベルを反転します。コンペアレジスタ 1(3/5) に対応した端子 OUT1(OUT3/OUT5) の出力レベルは、コンペアレジスタ 0(2/4) の一致とコンペアレジスタ 1(3/5) の一致の両方で出力レベルを反転します。なおコンペアレジスタ 0(2/4) と 1(3/5) が同値のときは、コンペアレジスタ 1 本のときと同じ動作をします。
 - OUT0/2/4: コンペアレジスタ 0/2/4 の一致によりレベルを反転します。
 - OUT1/3/5: コンペアレジスタ 0/2/4 と 1/3/5 の一致によりレベルを反転します。

[bit11, bit10] OTE1, OTE0

これらのビットは、アウトプットコンペアの端子出力を許可します。これらのビットは、初期値 "0" です。

0	汎用ポートとして動作します。 (初期値)
1	アウトプットコンペア端子出力として動作します。

- OTE1: アウトプットコンペア 1/3/5 に対応
- OTE0: アウトプットコンペア 0/2/4 に対応

[bit9, bit8] OTD1, OTD0

これらのビットは、アウトプットコンペアの端子出力を許可した場合の端子出力レベルを変更するときに使用します。コンペア端子出力の初期値は "0" です。書込み時は、コンペア動作を停止してから行ってください。読出し時は、アウトプットコンペア端子の出力値が読み出されます。

0	コンペア端子出力を "0" にします。 (初期値)
1	コンペア端子出力を "1" にします。

- OTD1: アウトプットコンペア 1/3/5 に対応
- OTD0: アウトプットコンペア 0/2/4 に対応

[bit7, bit6] ICP1, ICP0

これらのビットは、アウトプットコンペアの割込みフラグです。コンペアレジスタとフリーランタイム値が一致した場合に "1" にセットされます。割込み要求ビット (ICE1, ICE0) が許可されているときにこのビットがセットされると、アウトプットコンペア割込みが発生します。本ビットは、"0" の書込みによりクリアされ、"1" の書込みは意味を持ちません。リードモディファイ系では、"1" が読み出されます。

0	比較結果一致なし (初期値)
1	比較結果一致あり

- ICP1: アウトプットコンペア 1/3/5 に対応
- ICP0: アウトプットコンペア 0/2/4 に対応

[bit5, bit4] ICE1, ICE0

これらのビットは、アウトプットコンペアの割込み許可ビットです。このビットが "1" のときに割込みフラグ (ICP1, ICP0) がセットされると、アウトプットコンペア割込みが発生します。

0	アウトプットコンペア割込み禁止 (初期値)
1	アウトプットコンペア割込み許可

- ICE1: アウトプットコンペア 1/3/5 に対応
- ICE0: アウトプットコンペア 0/2/4 に対応

[bit3, bit2] 未使用ビット

これらのビットは、未使用ビットです。

[bit1, bit0] CST1, CST0

これらのビットは、コンペアレジスタとフリーランタイムとの一致動作を許可するビットです。

0	コンペア動作禁止 (初期値)
1	コンペア動作許可

- CST1: アウトプットコンペア 1/3/5 に対応
- CST0: アウトプットコンペア 0/2/4 に対応

コンペア動作を許可する前に、コンペアレジスタ値を設定してください。

<注意事項>

アウトプットコンペアは、フリーランタイムのクロックと同期させているため、フリーランタイムを停止させるとコンペア動作も停止します。

12.3.3 インプットキャプチャ

インプットキャプチャのレジスタの構成と機能について説明します。

■ インプットキャプチャのレジスタ一覧

図 12.3-11 に、インプットキャプチャのレジスタ一覧を示します。

図 12.3-11 インプットキャプチャのレジスタ一覧

ch. 0	00005D _H	15	14	13	12	11	10	9	8	IPCP0, 1
ch. 1	00005F _H	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	インプットキャプチャデータレジスタ
		(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	初期値XXXXXXXX _B
ch. 0	00005C _H	7	6	5	4	3	2	1	0	IPCP0, 1
ch. 1	00005E _H	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	インプットキャプチャデータレジスタ
		(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	初期値XXXXXXXX _B
	000060 _H	7	6	5	4	3	2	1	0	ICS01
		ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	インプットキャプチャ
		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	コントロールステータスレジスタ
										初期値00000000 _B

■ インプットキャプチャデータレジスタ (IPCP0, 1)

インプットキャプチャデータレジスタ (IPCP0, 1) のビット構成を、下図に示します。

図 12.3-12 インプットキャプチャデータレジスタ (IPCP0, 1) のビット構成

ch. 0	00005D _H	15	14	13	12	11	10	9	8	IPCP0, 1
ch. 1	00005F _H	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	インプットキャプチャデータレジスタ
		(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	初期値XXXXXXXX _B
ch. 0	00005C _H	7	6	5	4	3	2	1	0	IPCP0, 1
ch. 1	00005E _H	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	インプットキャプチャデータレジスタ
		(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	初期値XXXXXXXX _B

インプットキャプチャデータレジスタ (IPCP0, 1) は、対応した外部端子入力波形の有効エッジを検出したときにフリーランタイム値を保持するレジスタです。

このレジスタは、ワードアクセスをしてください。このレジスタに書き込むことはできません。

■ インพุットキャプチャコントロールステータスレジスタ (ICS01)

インพุットキャプチャコントロールステータスレジスタ (ICS01) のビット構成を、下図に示します。

図 12.3-13 インพุットキャプチャコントロールステータスレジスタ (ICS01) のビット構成

	7	6	5	4	3	2	1	0	ICS01
000060 _H	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	インพุットキャプチャ コントロール ステータスレジスタ 初期値00000000 _B
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	

以下に、インพุットキャプチャコントロールステータスレジスタ (ICS01) の各ビットの機能を説明します。

[bit7, bit6] ICP1, ICP0

これらのビットは、インพุットキャプチャ割込みフラグです。外部入力端子の有効エッジを検出すると、このビットを "1" にセットします。割込み許可ビット (ICE1, ICE0) がセットされていると、有効エッジを検出することにより割込みを発生します。

本ビットは "0" 書込みによりクリアされます。"1" の書込みは意味を持ちません。リードモディファイライト系命令では、"1" が読み出されます。

0	有効エッジ検出なし (初期値)
1	有効エッジ検出あり

- ICP1: インพุットキャプチャ 1 に対応
- ICP0: インพุットキャプチャ 0 に対応

[bit5, bit4] ICE1, ICE0

これらのビットは、インพุットキャプチャ割込み許可ビットです。このビットが "1" のときに割込みフラグ (ICP1, ICP0) がセットされると、インพุットキャプチャ割込みが発生します。

0	割込み禁止 (初期値)
1	割込み許可

- ICE1: インพุットキャプチャ 1 に対応
- ICE0: インพุットキャプチャ 0 に対応

[bit3, bit2, bit1, bit0] EG11, EG10, EG01, EG00

これらのビットは、外部入力の有効エッジ極性を指定します。インพุットキャプチャ動作許可も兼用しています。

EG11/EG01	EG10/EG00	エッジ検出極性
0	0	エッジ検出なし (停止状態) (初期値)
0	1	立上りエッジ検出
1	0	立下りエッジ検出
1	1	両エッジ検出

- EG11/EG10: インพุットキャプチャ 1 に対応
- EG01/EG00: インพุットキャプチャ 0 に対応

12.4 16 ビット入出力タイマの割込み

16 ビット入出力タイマの割込みは、下記の 3 つの場合に発生します。

- ・フリーランタイマのカウンタ値がオーバーフローした場合
- ・インプットキャプチャ入力端子へトリガエッジを入力した場合
- ・アウトプットコンペアの一致検出した場合

インプットキャプチャとアウトプットコンペアの割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができます。

■ 16 ビット入出力タイマの割込み

16 ビット入出力タイマの割込み制御ビットと割込み要因を、表 12.4-1 に示します。

表 12.4-1 16 ビット入出力タイマの割込み

	タイマカウンタ オーバーフロー割込み	インプットキャプチャ 割込み	アウトプットコンペア 割込み
割込み要求フラグ	TCCS: IVF (bit7)	ICS01: ICP1 (bit7) ch.1 ICS01: ICP0 (bit6) ch.0	OCS01/23/45: ICP1 (bit7) ch.1, ch.3, ch.5 OCS01/23/45: ICP0 (bit6) ch.0, ch.2, ch.4
割込み要求出力許可ビット	TCCS: IVFE (bit6)	ICS01: ICE1 (bit5) ch.1 ICS01: ICE0 (bit4) ch.0	OCS01/23/45: ICE1 (bit5) ch.1, ch.3, ch.5 OCS01/23/45: ICE0 (bit4) ch.0, ch.2, ch.4
割込み発生要因	16 ビットフリーラン タイマのカウンタ オーバーフロー	インプットキャプチャ 入力端子への有効エッ ジの入力	アウトプットコンペア レジスタ値とカウンタ値の一致

ICS01: ICP0/ICE0 がインプットキャプチャ端子 (IN0) に対応

ICS01: ICP1/ICE1 がインプットキャプチャ端子 (IN1) に対応

OCS01/23/45: ICP0/ICE0 がアウトプットコンペア端子 (OUT0/OUT2/OUT4) に対応

OCS01/23/45: ICP1/ICE1 がアウトプットコンペア端子 (OUT1/OUT3/OUT5) に対応

● タイマカウンタオーバーフロー割込み

タイマカウンタオーバーフロー割込み要求フラグがセットされる場合

次の場合に、タイマカウンタコントロールステータスレジスタのタイマカウンタオーバーフロー発生フラグがセットされます (TCCS: IVF=1)。

- ・フリーランタイマのカウントアップでオーバーフロー ("FFFF_H" → "0000_H") が発生した場合
- ・コンペアクリアレジスタによる初期化を許可に設定して (TCCS: MODE=1)、フリーランタイマの設定値とコンペアクリアレジスタの値が一致した場合

タイマカウンタオーバーフロー割込み要求が発生する場合

タイマカウンタオーバーフロー割込み要求を許可に設定していた場合は (TCCS: IVFE=1)、タイマカウンタオーバーフロー発生フラグが "1" にセットされると (TCCS: IVF=1)、割込み要求が発生します。

● インพุットキャプチャ割込み

インพุットキャプチャ端子で、設定した有効エッジ (ICS: EG) が検出された場合の割込み動作を次に示します。

- 検出された時点のフリーランタイムのカウント値がインพุットキャプチャレジスタに格納されます。
- コントロールステータスレジスタの有効エッジ検出フラグが "1" にセットされます (ICS: ICP=1)。
- インพุットキャプチャ割込み要求の出力を許可に設定している場合は (ICS: ICE=1), 割込み要求が発生します。

● アウトพุットコンペア割込み

フリーランタイムのカウント値とコンペアレジスタ設定値の一致が検出された場合の割込み発生動作を次に示します。

- コントロールレジスタのアウトพุットコンペア一致フラグが "1" にセットされます (OCS: IOP=1)。
- アウトพุットコンペア割込み要求を許可に設定している場合は (OCS: IOE=1), 割込み要求が発生します。

■ 16 ビット入出力タイマの割込みと DMA 転送および EI²OS

表 12.4-2 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 12.4-2 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS のクリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
インพุットキャプチャ (ch.0) 取込み *	○	5	#26	FFFF94 _H	ICR07	0000B7 _H
インพุットキャプチャ (ch.1) 取込み	○	6	#27	FFFF90 _H	ICR08	0000B8 _H
アウトพุットコンペア (ch.0) 一致	○	8	#28	FFFF8C _H		
アウトพุットコンペア (ch.1) 一致	○	9	#29	FFFF88 _H	ICR09	0000B9 _H
アウトพุットコンペア (ch.2) 一致	○	10	#30	FFFF84 _H		
アウトพุットコンペア (ch.3) 一致	○	×	#31	FFFF80 _H	ICR10	0000BA _H
アウトพุットコンペア (ch.4) 一致	○	×	#32	FFFF7C _H		
アウトพุットコンペア (ch.5) 一致 *	○	×	#33	FFFF78 _H	ICR11	0000BB _H
16 ビットフリーランタイムオーバーフロー, * 16 ビットリロードタイムアンダフロー	○	12	#35	FFFF70 _H	ICR12	0000BC _H

× : 割込み要求フラグはクリアされません。

○ : 割込み要求フラグはクリアされます。

* : この割込み要因は、ほかの周辺機能の割込み要因と割込み番号を共有しています。

詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグをクリアします。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送 , および EI²OS 機能への対応

インプットキャプチャとフリーランタイマは DMA 転送機能 , および EI²OS 機能に対応しています。アウトプットコンペアは EI²OS 機能と ch.0 ~ ch.2 のみが DMA 転送機能に対応しています。DMA または EI²OS 機能を使用する場合には , 割込み制御レジスタ (ICR) を共有するほかの割込みを , 禁止に設定する必要があります。

12.5 16 ビット入出力タイマの動作

16 ビット入出力タイマの動作およびタイミングについて説明します。

■ 16 ビット入出力タイマの動作, タイミング

16 ビット入出力タイマの動作, タイミングとして以下の項目について説明します。

- フリーランタイマの動作
- アウトプットコンペアの動作
- インプットキャプチャの動作
- フリーランタイマのタイミング
 - カウントタイミング
 - クリアタイミング
- アウトプットコンペアのタイミング
 - 割込みタイミング
 - 出力端子の変化タイミング
- インプットキャプチャのタイミング
 - 入力信号に対するキャプチャタイミング

12.5.1 フリーランタイマの動作

フリーランタイマの動作およびタイミングについて説明します。

■ フリーランタイマの動作

フリーランタイマは、リセット解除後にカウンタ値 "0000" よりカウントを開始します。このカウンタ値が、アウトプットコンペアとインプットキャプチャの基準時間となります。カウンタ値は、次の条件でクリアされます。

- オーバフローが発生したとき
- アウトプットコンペア 0 値とコンペア一致したとき (モード設定が必要)
- TCCS レジスタの SCLR ビットに "1" を書き込んだとき
- TCDT レジスタに "0000" を書き込んだとき
- リセット時

割込みはオーバフローが発生したとき、またはコンペアレジスタ 0 値とフリーランタイマのカウンタ値の比較結果が一致したとき発生します。比較結果一致割込みには、モード設定が必要です。

図 12.5-1 にオーバフローによるカウンタクリアのタイミングチャートを、図 12.5-2 に比較結果一致によるカウンタクリアのタイミングチャートを示します。

図 12.5-1 オーバフローによるカウンタクリアのタイミングチャート

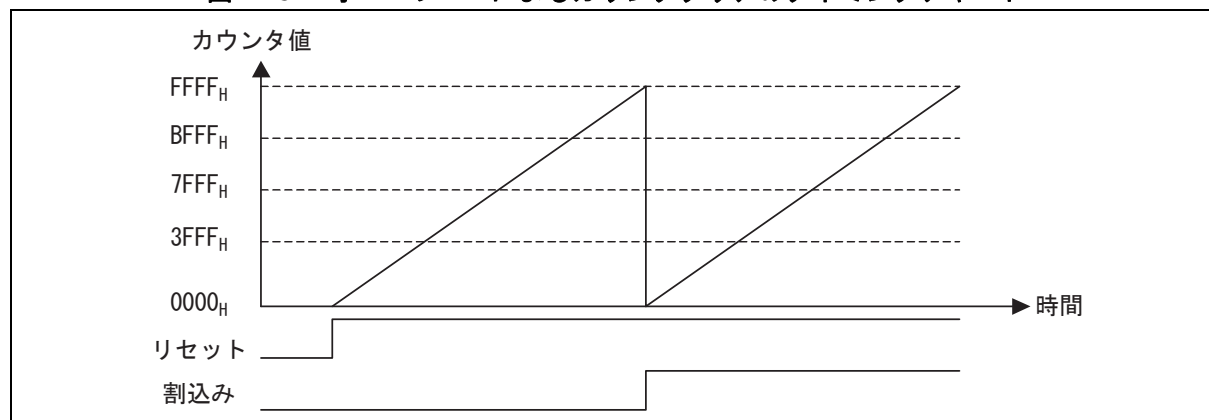
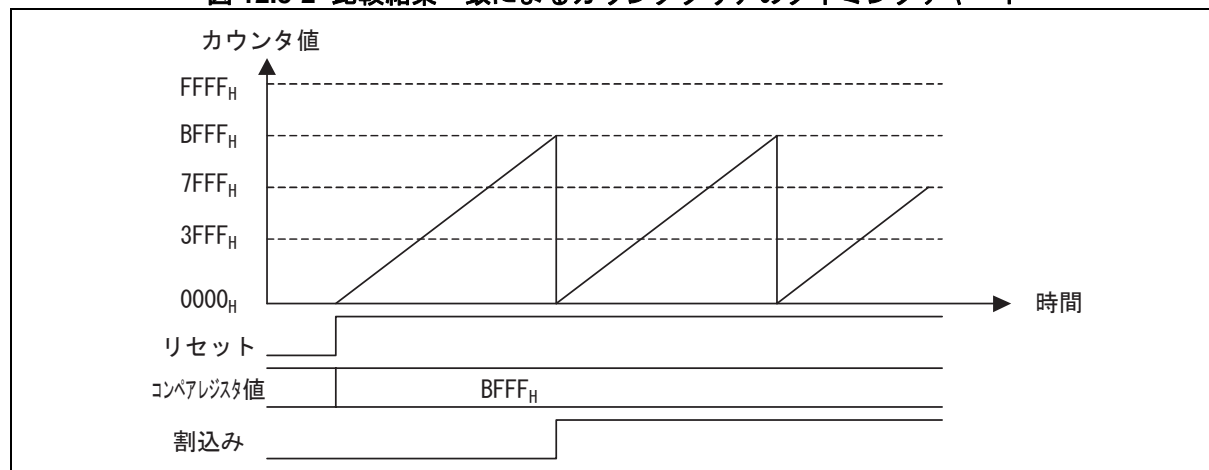


図 12.5-2 比較結果一致によるカウンタクリアのタイミングチャート



12.5.2 アウトプットコンペアの動作

アウトプットコンペアは、設定されたコンペアレジスタ値とフリーランタイムとの値を比較して、一致していた場合に割込み要求を発生するとともに、出力レベルを反転します。

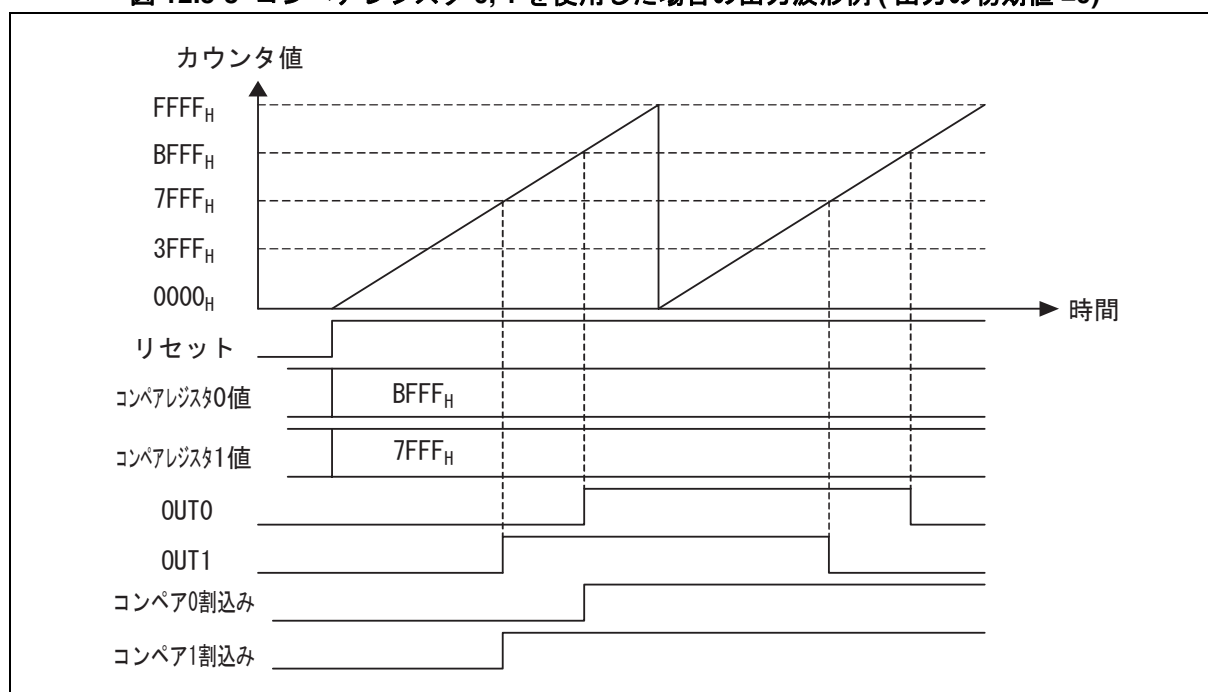
■ 出力波形例

以下に、出力波形例を示します。

● コンペアレジスタ 0, 1 を使用した場合の出力波形例

出力の初期値を "0" とした場合の出力波形例を、図 12.5-3 に示します。

図 12.5-3 コンペアレジスタ 0, 1 を使用した場合の出力波形例 (出力の初期値 = 0)

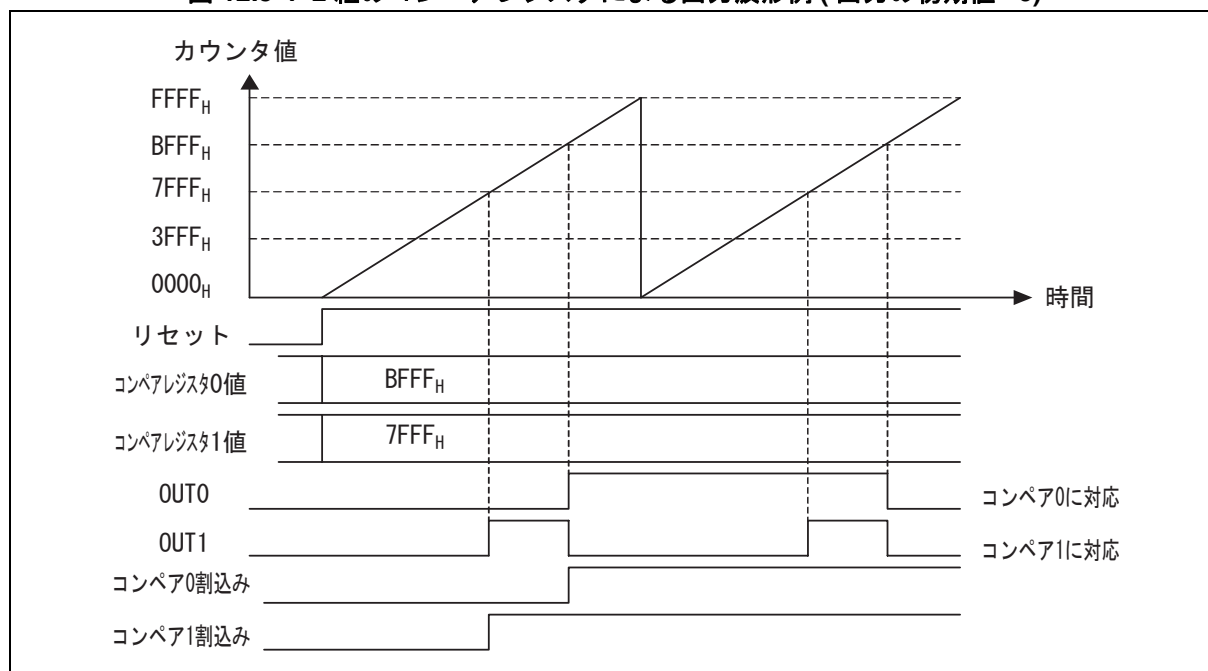


CMOD=1 のとき、2 組のコンペアレジスタを使って出力レベルを変えることもできます。

● 2 組のコンペアレジスタによる出力波形例

出力の初期値を "0" とした場合の出力波形例を、図 12.5-4 に示します。

図 12.5-4 2 組のコンペアレジスタによる出力波形例 (出力の初期値 =0)



<注意事項>

コンペアレジスタを書き換える場合はコンペア割込みのルーチン内で行うか、コンペア動作禁止の状態で行い、比較結果一致と書込みが同時に発生しないようにしてください。

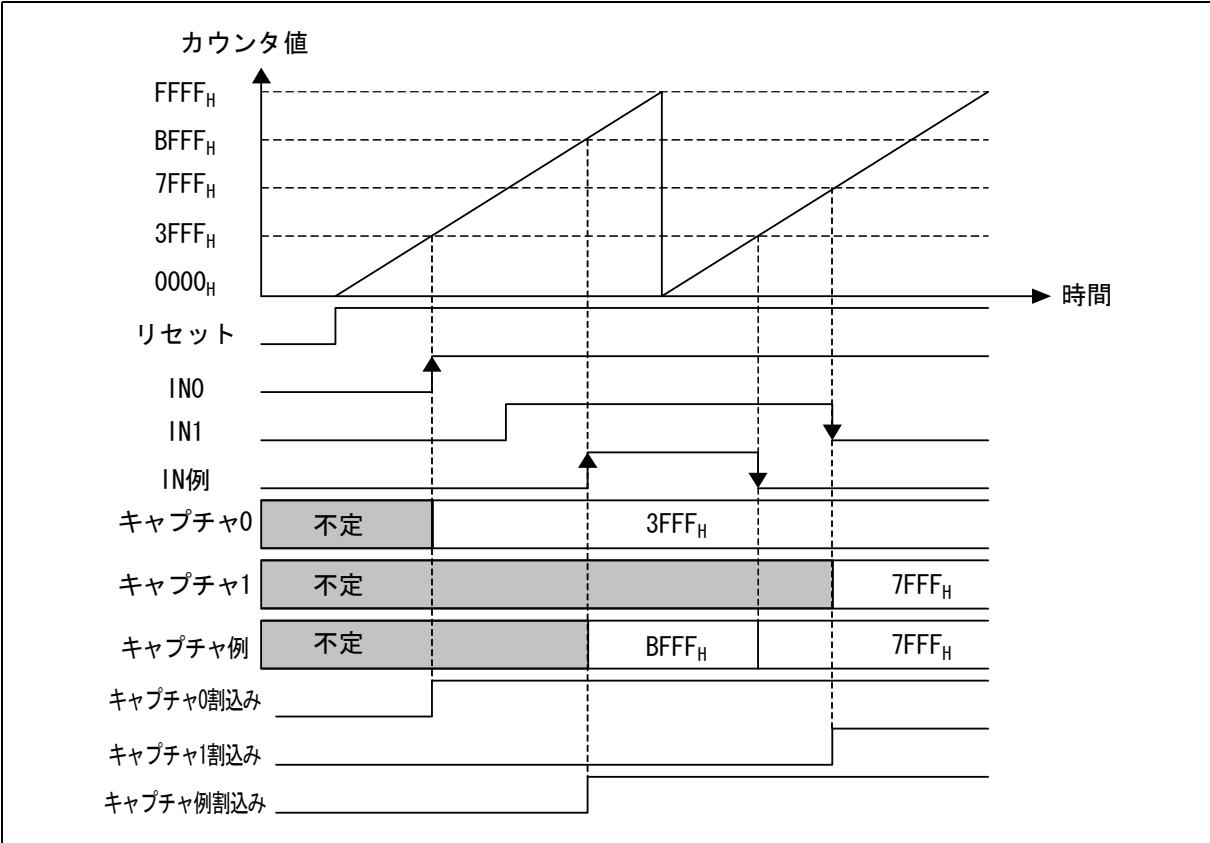
12.5.3 インプットキャプチャの動作

インプットキャプチャは，設定された有効エッジを検出するとフリーランタイマの値をキャプチャレジスタに取り込んで割込み要求を発生します。

■ インプットキャプチャの取込みタイミング例

図 12.5-5 に，インプットキャプチャの取込みタイミング例を示します。

図 12.5-5 インプットキャプチャの取込みタイミング例



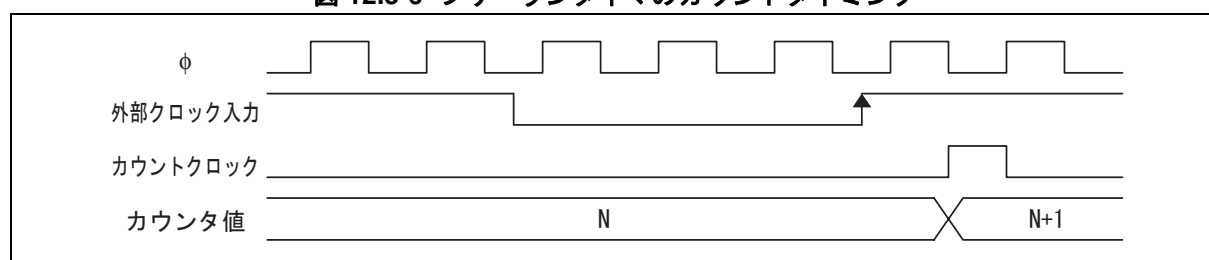
12.5.4 フリーランタイマのタイミング

フリーランタイマは，入力されたクロック（内部，または外部クロック）によりカウントアップされます。外部クロック選択時は，立上リエッジでカウントされます。

■ フリーランタイマのカウントタイミング

図 12.5-6 に，フリーランタイマのカウントタイミングを示します。

図 12.5-6 フリーランタイマのカウントタイミング

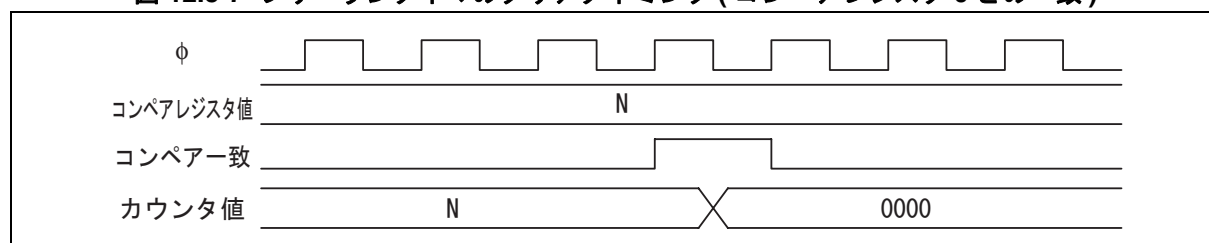


カウンタのクリアはリセット，ソフトウェアによるクリア，およびコンペアレジスタ 0 とフリーランタイマとの一致で行うことができます。リセットとソフトウェアでのカウンタクリアは，クリア発生とともに行われます。コンペアレジスタ 0 との一致によるカウンタクリアは，カウントタイミングに同期して行われます。

■ フリーランタイマのクリアタイミング (コンペアレジスタ 0 との一致)

図 12.5-7 に，コンペアレジスタ 0 との一致によるフリーランタイマのクリアタイミングを示します。

図 12.5-7 フリーランタイマのクリアタイミング (コンペアレジスタ 0 との一致)

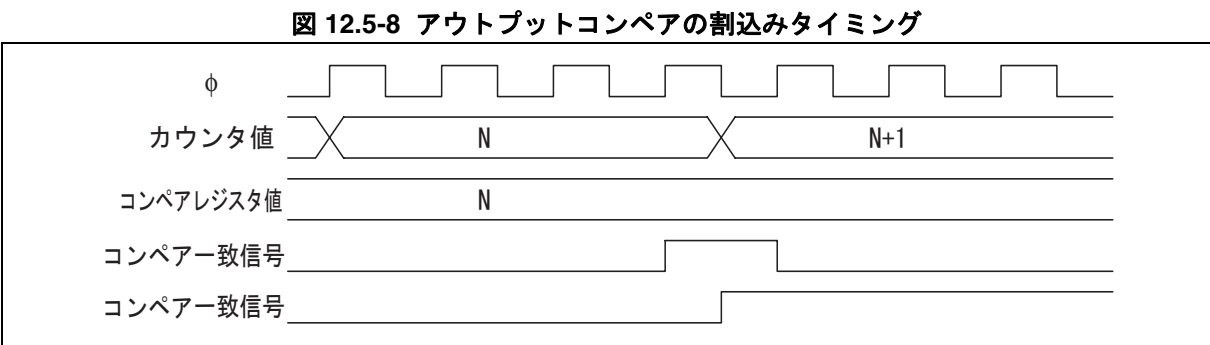


12.5.5 アウトプットコンペアのタイミング

アウトプットコンペアは、フリーランタイムと設定したコンペアレジスタの値が一致したときにコンペアマッチ信号が発生して出力値を反転するとともに割込み要求が発生します。コンペアマッチ時の出力反転タイミングはカウンタタイミングに同期して行われます。

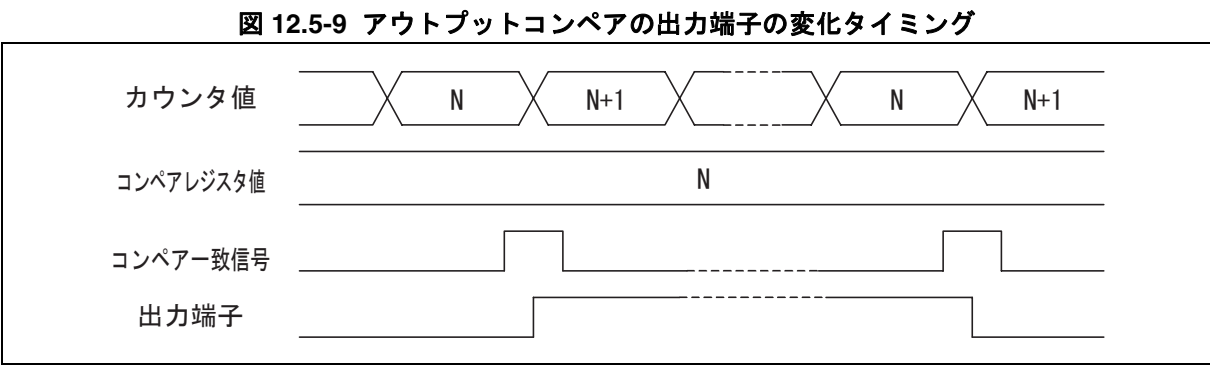
■ 割込みタイミング

図 12.5-8 に、アウトプットコンペアの割込みタイミングを示します。



■ 出力端子の変化タイミング

図 12.5-9 に、アウトプットコンペアの出力端子の変化タイミングを示します。



<注意事項>

コンペアレジスタを書き換える場合は、コンペア割込みのルーチン内で行うか、コンペア動作禁止の状態で行い、比較結果一致と書込みが同時に発生しないようにしてください。

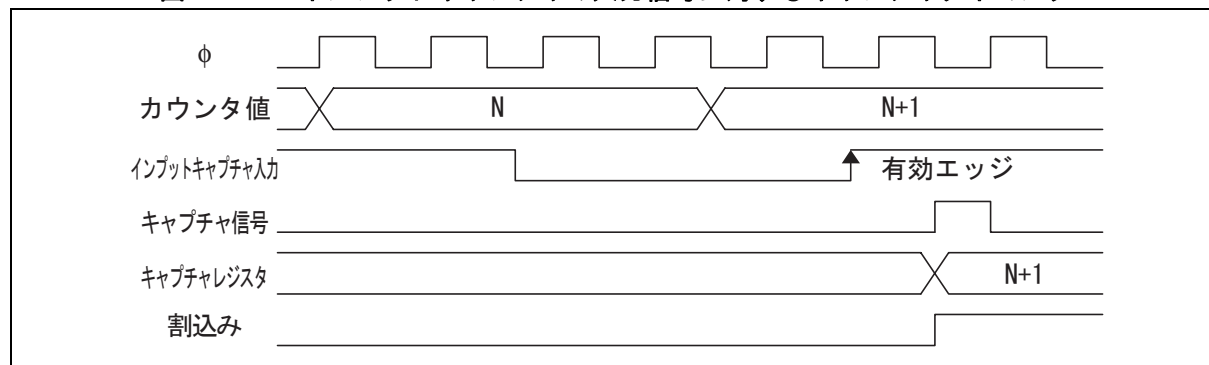
12.5.6 インプットキャプチャのタイミング

インプットキャプチャの入力信号に対するキャプチャタイミングを示します。

■ 入力信号に対するキャプチャタイミング

図 12.5-10 に、インプットキャプチャの入力信号に対するキャプチャタイミングを示します。

図 12.5-10 インプットキャプチャの入力信号に対するキャプチャタイミング



12.6 16 ビット入出力タイマのプログラム例

16 ビット入出力タイマのプログラム例を示します。

■ フリーランタイマのプログラム例

<p>設定手順例</p> <p>フリーランタイマ、クロック = $\phi/2^4$、割込み処理でオーバーフロー回数をカウント。</p> <p><初期設定></p> <ul style="list-style-type: none">フリーランタイマの制御 <table><tr><td>制御レジスタの設定</td><td>TCCS</td></tr><tr><td>クロック選択 >></td><td>.ECKE</td></tr><tr><td>割込み要求フラグ >></td><td>.IVF</td></tr><tr><td>割込み要求許可 >></td><td>.IVFE</td></tr><tr><td>カウント動作 >></td><td>.STOP</td></tr><tr><td>タイマの初期化条件 >></td><td>.MODE</td></tr><tr><td>TCDT クリア >></td><td>.SCLR</td></tr><tr><td>カウントクロック >></td><td>.CLK2-0</td></tr><tr><td>タイマデータ値の設定</td><td>TCDT</td></tr></table> <p>・割込み関連</p> <table><tr><td>割込みレベルの設定</td><td>ICR12</td></tr><tr><td>I フラグの設定</td><td>(CCR)</td></tr></table> <p>・変数の設定</p> <p><起動></p> <ul style="list-style-type: none">フリーランタイマ ch.0 起動 <table><tr><td colspan="2">レジスタ名 . ビット名</td></tr><tr><td>カウント動作起動</td><td>TCCS .STOP</td></tr></table> <p><割込み></p> <ul style="list-style-type: none">割込み処理 <table><tr><td colspan="2">レジスタ名 . ビット名</td></tr><tr><td>割込み要求フラグのクリア</td><td>TCCS .IVF</td></tr><tr><td colspan="2">(任意の処理)</td></tr><tr><td colspan="2">変数のカウント</td></tr></table> <p><割込みベクタ></p> <ul style="list-style-type: none">ベクタテーブルの設定 <p>(注意事項)</p> <p>事前にクロック関連の設定および、<code>__set_il(数値)</code> の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	制御レジスタの設定	TCCS	クロック選択 >>	.ECKE	割込み要求フラグ >>	.IVF	割込み要求許可 >>	.IVFE	カウント動作 >>	.STOP	タイマの初期化条件 >>	.MODE	TCDT クリア >>	.SCLR	カウントクロック >>	.CLK2-0	タイマデータ値の設定	TCDT	割込みレベルの設定	ICR12	I フラグの設定	(CCR)	レジスタ名 . ビット名		カウント動作起動	TCCS .STOP	レジスタ名 . ビット名		割込み要求フラグのクリア	TCCS .IVF	(任意の処理)		変数のカウント		<p>プログラム例</p> <pre>void FREE_RUN_TIMER_sample(void) { FREERUN_initial(); FREERUN_start(); } void FREERUN_initial(void) { IO_TCCS.word = 0x006C; /* 設定値 = 0000_0000_0110_1100 */ /* bit15 = 0 ECKE 内部クロックソース */ /* bit7 = 0 IVF 割込み要求フラグ */ /* bit6 = 1 IVFE 割込み許可 */ /* bit5 = 1 STOP カウント禁止 */ /* bit4 = 0 MODE リセット、クリアビットによる初期化 */ /* bit3 = 1 SCLR フリーランタイマ値の初期化 */ /* bit2-0 = 100 CLK2-0 カウントクロック $\phi/16$ */ IO_TCDT = 0x0000; /* タイマデータ値の初期化 */ IO_ICR12.byte = 0x00; /* フリーランタイマ 割込みレベル設定 (値は任意) */ __EI(); /* 割込み許可 */ count = 0; } void FREERUN_start(void) { IO_TCCS.bit.STOP = 0; /* bit5 = 0 STOP カウント許可 */ } __interrupt void FREE_RUN_TIMER_int(void) { IO_TCCS.bit.IVF = 0; /* bit7 = 0 IVF オーバフローフラグのクリア */ count++; } #pragma intvect FREE_RUN_TIMER_int 35</pre> <p>(注意事項)</p> <p>レジスタの記述形式については、「F²MC-16LX ファミリ MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
制御レジスタの設定	TCCS																																		
クロック選択 >>	.ECKE																																		
割込み要求フラグ >>	.IVF																																		
割込み要求許可 >>	.IVFE																																		
カウント動作 >>	.STOP																																		
タイマの初期化条件 >>	.MODE																																		
TCDT クリア >>	.SCLR																																		
カウントクロック >>	.CLK2-0																																		
タイマデータ値の設定	TCDT																																		
割込みレベルの設定	ICR12																																		
I フラグの設定	(CCR)																																		
レジスタ名 . ビット名																																			
カウント動作起動	TCCS .STOP																																		
レジスタ名 . ビット名																																			
割込み要求フラグのクリア	TCCS .IVF																																		
(任意の処理)																																			
変数のカウント																																			

■ プログラム例以外の設定方法

● 内部クロックの種類と選択方法

8 種の内部クロックがありクロック選択ビット (TCCS.ECKE) と , カウントクロックビット (TCCS.CLK[2:0]) で設定します。

内部クロック	設定		カウント周期	
	クロック選択ビット (ECKE)	カウントクロックビット (CLK[2:0])	$\phi = 20\text{MHz}$	$\phi = 16\text{MHz}$
ϕ を選択するには	"0" にする	"000" にする	50ns	62.5ns
$\phi/2$ を選択するには	"0" にする	"001" にする	100ns	0.125 μs
$\phi/4$ を選択するには	"0" にする	"010" にする	0.2 μs	0.25 μs
$\phi/8$ を選択するには	"0" にする	"011" にする	0.4 μs	0.5 μs
$\phi/16$ を選択するには	"0" にする	"100" にする	0.8 μs	1.0 μs
$\phi/32$ を選択するには	"0" にする	"101" にする	1.6 μs	2.0 μs
$\phi/64$ を選択するには	"0" にする	"110" にする	3.2 μs	4.0 μs
$\phi/128$ を選択するには	"0" にする	"111" にする	6.4 μs	8.0 μs

● 外部クロックの選択方法

クロック選択ビット (TCCS.ECKE) と , データ方向ビット , ポート機能ビットで設定します。

設定		端子	カウント周期
クロック選択ビット (ECKE) を "1" にする	データ方向ビット (DDR9.P93) を "0" にする	FRCK	2 ϕ 以上

● フリーランタイムのカウント動作を許可 / 禁止する方法

カウント動作ビット (TCCS.STOP) で設定します。

動作	カウント動作ビット (STOP)
フリーランタイムのカウント動作を許可させるには	"0" にする
フリーランタイムのカウント動作を禁止させるには	"1" にする

● フリーランタイムをクリアする方法

フリーランタイムをクリアするには以下の方法があります。

- クリアビット (TCCS.SCLR) で設定

動作	クリアビット (SCLR)
フリーランタイムをクリアするには	"1" を書き込む

- フリーランタイム値とコンペアクリアレジスタ値との一致 (タイマ初期化条件ビット (TCCS.MODE) で設定)

動作	タイマ初期化条件ビット (MODE)
比較結果一致でフリーランタイムをクリアするには	"1" にする

- リセット
リセット (外部リセット, ウォッチドッグリセット, ソフトウェアリセット) によりフリーランタイマがクリアします。
- タイマカウンタデータレジスタ (TCDT) へ "0000_H" を設定
フリーランタイマの動作停止中にタイマカウンタデータレジスタ (TCDT) へ "0000_H" を書き込むと, カウント値が "0000_H" にクリアします。

● 割込み関連レジスタ

割込みレベル, ベクタの関係は下表のとおりです。
割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

割込みベクタ	割込みレベル設定レジスタ
#35 アドレス: FFFF70 _H	割込みレベルレジスタ (ICR12) アドレス: 0000BC _H

割込み要求フラグ (TCCS.IVF) は自動的にクリアしませんので, 割込み処理から復帰する前にソフトウェアにてクリアしてください (IVF ビットに "0" を書き込む)。

● 割込みの種類

割込みは 1 種類です。フリーランタイマのオーバフローで発生します。

● 割込みを許可する方法

割込みの許可 / 禁止は, 割込み要求許可ビット (TCCS.IVFE) で設定します。

制御	割込み要求許可ビット (IVFE)
割込み禁止	"0" にする
割込み許可	"1" にする

割込み要求のクリアは, 割込み要求ビット (TCCS.IVF) で設定します。

制御	割込要求ビット (IVF)
割込み要求クリア	"0" を書き込む

● フリーランタイマ動作を停止する方法

カウント動作ビット (TCCS.STOP) で設定します。

動作	カウント動作ビット (STOP)
フリーランタイマのカウント動作を停止させるには	"1" にする

■ アウトプットコンペアのプログラム例

設定手順例

2 チャネル独立出力
コンペア動作 (7FFF, BFFFF)
割込み発生 コンペアクリアなし

<初期設定>
・フリーランタイマの制御

制御レジスタの設定	TCCS
クロック選択 >>	.ECKE
割込み要求フラグ >>	.IVF
割込み要求許可 >>	.IVFE
カウント動作 >>	.STOP
タイマの初期化条件 >>	.MODE
TCDT クリア >>	.SCLR
カウントクロック >>	.CLK2-0
タイマデータ値の設定	TCDT

・アウトプットコンペアの制御

レジスタ名 ビット名

制御レジスタの設定	OCS01
端子出力レベル反転動作 >>	.CMOD
端子出力許可	.OTE1, OTE0
端子出力レベル指定 >>	.OTD1, OTD0
割込み要求フラグ >>	.ICP1, ICP0
割込み要求許可 >>	.ICE1, ICE0
動作許可設定 >>	.CST1, CST0

コンペア値 ch.0 の設定	OCCP0
コンペア値 ch.1 の設定	OCCP1

・割込み関連

割込みレベルの設定	ICR08
割込みレベルの設定	ICR09
I フラグの設定	(CCR)

<起動>
・アウトプットコンペア起動

レジスタ名 .ビット名

割込み制御	OCS01	.ICE1,ICE0
コンペア動作起動	OCS01	.CST1,CST0

・フリーランタイマ起動

レジスタ名 .ビット名

カウント動作起動	TCCS	.STOP
----------	------	-------

プログラム例

void OUTPUT01_sample(void)

{

freerun_initial();

OUTPUT01_initial();

OUTPUT01_start();

freerun_start();

}

void freerun_initial(void)

{

IO_TCCS.word = 0x0020; /* 設定値 =0000_0000_0010_0000 */

/* bit15 = 0 ECKE 内部クロックソース */

/* bit7 = 0 IVF 割込み要求フラグ */

/* bit6 = 0 IVFE 割込み禁止 */

/* bit5 = 1 STOP カウント禁止 */

/* bit4 = 0 MODE リセット, クリアビットによる初期化 */

/* bit3 = 0 SCLR フリーランタイマ値の初期化 */

/* bit2-0 = 000 CLK2-0 カウントクロック $\phi/4=32\text{MHz}/4$ */

/* タイマデータ値の初期化 */

IO_TCDT = 0x0000;

}

void OUTPUT01_initial(void)

{

IO_OCS01.word = 0x0C00; /* 設定値 =0000_1100_0000_0000 */

/* bit15-13 = 000 未定義ビット */

/* bit12 = 0 CMOD ch.0, ch.1 レベル反転 */

/* bit11-10 = 11 OTE1, OTE0 端子出力許可 */

/* bit9-8 = 00 OTD1, OTD0 コンペア端子出力 0 */

/* bit7-6 = 00 ICP1,ICP0 アウトプットコンペアフラグのクリア */

/* bit5-4 = 00 ICE1, ICE0 アウトプットコンペア割込み禁止 */

/* bit3-2 = 00 未定義ビット */

/* bit1-0 = 00 CST1, CST0 コンペア動作禁止 */

IO_OCCP0 = BFFF;

/* コンペアレジスタ ch.0 の設定 */

IO_OCCP1 = 7FFF;

/* コンペアレジスタ ch.1 の設定 */

IO_ICR08.byte = 0x00;

/* アウトプットコンペア ch.0 割込みレベル設定
 (値は任意) */

IO_ICR09.byte = 0x00;

/* アウトプットコンペア ch.1 割込みレベル設定
 (値は任意) */

__EI();

/* 割込み許可 */

}

void OUTPUT01_start(void)

{

IO_OCS01.word = 0x0C30; /* bit5-4 = 11 ICE1, ICE0 アウトプットコンペア割
 込み許可 */

IO_OCS01.word = 0x0C33; /* bit1-0 = 11 CST1, CST0 コンペア動作許可 */

}

void freerun_start(void)

{

IO_TCCS.bit.STOP = 0;

/* bit4 = 0 STOP カウント許可 */

}

(続く)

(続き)

<p>< 割込み ></p> <ul style="list-style-type: none">・ 割込み処理 <div><div>レジスタ名 . ビット名</div><table><tr><td>割込み要求フラグのクリア</td><td>OCS01 . ICP0</td></tr><tr><td>(任意の処理)</td><td></td></tr><tr><td>*****</td><td></td></tr><tr><td>割込み要求フラグのクリア</td><td>OCS01 . ICP1</td></tr><tr><td>(任意の処理)</td><td></td></tr><tr><td>*****</td><td></td></tr></table></div> <p>< 割込みベクタ ></p> <ul style="list-style-type: none">・ ベクタテーブルの設定 <p>(注意事項)</p> <p>事前にクロック関連の設定および, __set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	割込み要求フラグのクリア	OCS01 . ICP0	(任意の処理)		*****		割込み要求フラグのクリア	OCS01 . ICP1	(任意の処理)		*****		<pre>__interrupt void OUTPUT0_int(void) { IO_OCS01.bit.ICP0 = 0; /* bit6 = 0 ICP0 割込みフラグのクリア */ } __interrupt void OUTPUT1_int(void) { IO_OCS01.bit.ICP1 = 0; /* bit7 = 0 ICP1 割込みフラグのクリア */ } #pragma intvect OUTPUT0_int 28 #pragma intvect OUTPUT1_int 29</pre> <p>(注意事項)</p> <p>レジスタの記述形式については, 「F²MC-16LX ファミリ MB90480/485 シリーズ用サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
割込み要求フラグのクリア	OCS01 . ICP0												
(任意の処理)													

割込み要求フラグのクリア	OCS01 . ICP1												
(任意の処理)													

■ プログラム例以外の設定方法

● コンペア値を設定する方法

コンペアレジスタ OCCP0 ~ OCCP5 にコンペア値を書き込みます。

● コンペアモードを設定する方法 (OUT1, OUT2, OUT3, OUT4, OUT5 出力にて有効)

コンペアモードビット (OCS01.CMOD, OCS23.CMOD, OCS45.CMOD) で行います。

動作	コンペアモードビット
フリーランタイマとコンペアレジスタ 1 のみの比較結果一致により OUT1 出力を反転させるには	(OCS01.CMOD) ビットを "0" にする
フリーランタイマとコンペアレジスタ 3 のみの比較結果一致により OUT3 出力を反転させるには	(OCS23.CMOD) ビットを "0" にする
フリーランタイマとコンペアレジスタ 5 のみの比較結果一致により OUT5 出力を反転させるには	(OCS45.CMOD) ビットを "0" にする
フリーランタイマとコンペアレジスタ 0, およびフリーランタイマとコンペアレジスタ 1, との比較結果一致により OUT1 出力を反転させるには	(OCS01.CMOD) ビットを "1" にする
フリーランタイマとコンペアレジスタ 2, およびフリーランタイマとコンペアレジスタ 3, との比較結果一致により OUT3 出力を反転させるには	(OCS23.CMOD) ビットを "1" にする
フリーランタイマとコンペアレジスタ 4, およびフリーランタイマとコンペアレジスタ 5, との比較結果一致により OUT5 出力を反転させるには	(OCS45.CMOD) ビットを "1" にする

CMOD ビットとは関係なく, 以下の出力がされます。

- OUT0 出力 は, フリーランタイマとコンペアレジスタ 0 のみの比較結果一致により出力を反転します。
- OUT2 出力 は, フリーランタイマとコンペアレジスタ 2 のみの比較結果一致により出力を反転します。

- OUT4 出力は、フリーランタイマとコンペアレジスタ 4 のみの比較結果一致により出力を反転します。

● コンペア動作を許可 / 禁止する方法

コンペア動作許可ビット (OCS01.CST[1:0], OCS23.CST[1:0], OCS45.CST[1:0]) で行います。

動作	コンペア	コンペア動作許可ビット
コンペア動作を停止 (禁止) するには	コンペア 0	(OCS01.CST0) を "0" にする
	コンペア 1	(OCS01.CST1) を "0" にする
	コンペア 2	(OCS23.CST0) を "0" にする
	コンペア 3	(OCS23.CST1) を "0" にする
	コンペア 4	(OCS45.CST0) を "0" にする
	コンペア 5	(OCS45.CST1) を "0" にする
コンペア動作を許可するには	コンペア 0	(OCS01.CST0) を "1" にする
	コンペア 1	(OCS01.CST1) を "1" にする
	コンペア 2	(OCS23.CST0) を "1" にする
	コンペア 3	(OCS23.CST1) を "1" にする
	コンペア 4	(OCS45.CST0) を "1" にする
	コンペア 5	(OCS45.CST1) を "1" にする

● コンペア端子出力の初期レベルを設定する方法

コンペア端子出力指定ビット (OCS01.OTD[1:0], OCS23.OTD[1:0], OCS45.OTD[1:0]) で設定します。

動作	コンペア端子出力指定ビット
コンペア 0 端子を "L" するには	(OCS01.OTD0) を "0" にする
コンペア 0 端子を "H" するには	(OCS01.OTD0) を "1" にする
コンペア 1 端子を "L" するには	(OCS01.OTD1) を "0" にする
コンペア 1 端子を "H" するには	(OCS01.OTD1) を "1" にする
コンペア 2 端子を "L" するには	(OCS23.OTD0) を "0" にする
コンペア 2 端子を "H" するには	(OCS23.OTD0) を "1" にする
コンペア 3 端子を "L" するには	(OCS23.OTD1) を "0" にする
コンペア 3 端子を "H" するには	(OCS23.OTD1) を "1" にする
コンペア 4 端子を "L" するには	(OCS45.OTD1) を "0" にする
コンペア 4 端子を "H" するには	(OCS45.OTD1) を "1" にする
コンペア 5 端子を "L" するには	(OCS45.OTD1) を "0" にする
コンペア 5 端子を "H" するには	(OCS45.OTD1) を "1" にする

● コンペア端子 OUT0-OUT5 を出力に設定する方法

ポート機能レジスタ (OCS01.OTE[1:0], OCS23.OTE[1:0], OCS45.OTE[1:0]) で設定します。

動作	ポート機能ビット
コンペア 0 端子 (OUT0) を出力に設定するには	(OCS01.OTE0) を "1" にする
コンペア 1 端子 (OUT1) を出力に設定するには	(OCS01.OTE1) を "1" にする
コンペア 2 端子 (OUT2) を出力に設定するには	(OCS23.OTE0) を "1" にする
コンペア 3 端子 (OUT3) を出力に設定するには	(OCS23.OTE1) を "1" にする
コンペア 4 端子 (OUT4) を出力に設定するには	(OCS45.OTE0) を "1" にする
コンペア 5 端子 (OUT5) を出力に設定するには	(OCS45.OTE1) を "1" にする

● フリーランタイムをクリアする方法

クリアビット (TCCS.SCLR) で設定します。

動作	クリアビット (SCLR)
フリーランタイムをクリアするには	"1" を書き込む

他にも方法については「12.3.1 フリーランタイム」を参照してください。

● コンペア動作許可する方法

コンペア動作許可ビット (OCS01.CST[1:0], OCS23.CST[1:0], OCS45.CST[1:0]) で行います。

● フリーランタイム値とをコンペアレジスタ値比較して一致したときにフリーランタイムをクリアさせる方法

タイマ初期化条件ビット (TCCS.MODE) で設定します。

動作	タイマ初期化条件ビット (MODE)
コンペア 0 一致でフリーランタイム 0 をクリアするには	(TCCS.MODE) を "1" にする

● 割込み関連レジスタ

チャネル、割込みレベル、ベクタの関係は下表のとおりです。

割込みレベル、割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

チャネル	割込みベクタ	割込みレベル設定レジスタ
アウトプット コンペア 0	#28 アドレス : FFFF8C _H	割込みレベルレジスタ (ICR08) アドレス : 0000B8 _H
アウトプット コンペア 1	#29 アドレス : FFFF88 _H	割込みレベルレジスタ (ICR09) アドレス : 0000B9 _H
アウトプット コンペア 2	#30 アドレス : FFFF84 _H	割込みレベルレジスタ (ICR09) アドレス : 0000B9 _H
アウトプット コンペア 3	#31 アドレス : FFFF80 _H	割込みレベルレジスタ (ICR10) アドレス : 0000BA _H
アウトプット コンペア 4	#32 アドレス : FFFF7C _H	割込みレベルレジスタ (ICR10) アドレス : 0000BA _H
アウトプット コンペア 5	#33 アドレス : FFFF78 _H	割込みレベルレジスタ (ICR11) アドレス : 0000BB _H

割込み要求フラグ (OCS01.ICP[1:0], OCS23.ICP[1:0], OCS45.ICP[1:0]) は自動的にクリアしませんので、割込み処理から復帰する前にソフトウェアにて ICP[1:0] ビットに "0" を書き込みクリアしてください。

● 割込みの種類

割込みは 1 種類のみです。比較結果一致で発生します。

● 割込みを許可する方法

割込み許可の設定は、割込み要求許可ビット (OCS01.ICE[1:0], OCS23.ICE[1:0], OCS45.ICE[1:0]) で行います。

制御	割込み要求許可ビット (ICE0, ICE1)
割込み禁止	"0" にする
割込み許可	"1" にする

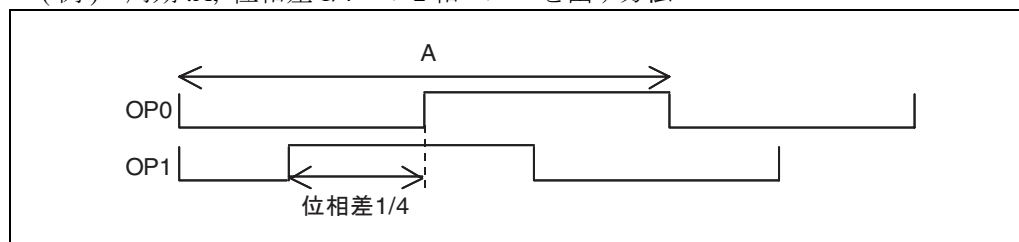
割込み要求のクリアは、割込み要求ビット (OCS01.ICP[1:0], OCS23.ICP[1:0], OCS45.ICP[1:0]) で行います。

制御	割込要求ビット (ICP0, ICP1)
割込み要求クリア	"0" を書き込む

● コンペア値の算出方法

- トグル出力パルス

(例) 周期:A, 位相差 1/4 の 2 相パルスを出す方法



式: コンペア 0 の値 = $(A/2) / \text{カウントクロック}$

コンペア 1 の値 = $(A/4) / \text{カウントクロック}$

(カウントクロック: フリーランタイムで設定した時間)

<注意事項>

コンペア 0 の一致でフリーランタイム 0 をクリアする設定

(TCCS0.MODE=1) と CMOD=0 の設定が必要です。

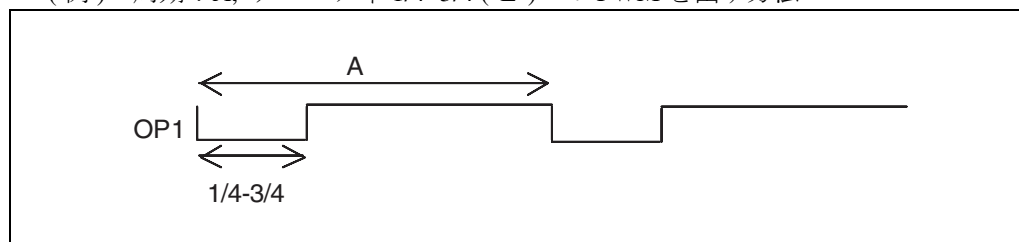
計算例: A=1024μs, カウントクロック=125ns

コンペア 0 の値 = $(1024000 / 2) / 125 - 1 = 4095 = \text{FFF}_H$

コンペア 1 の値 = $(1024000 / 4) / 125 - 1 = 1023 = \text{7FF}_H$

- PWM 出力

(例) 周期:A, デューティ 1/4~3/4 ("L") の PWM を出す方法



式: コンペア 0 の値 = $A / \text{カウントクロック}$

コンペア 1 の値 = $(A/4) / \text{カウントクロック}$ (デューティ 1/4 のとき)

$(A \times 3/4) / \text{カウントクロック}$ (デューティ 3/4 のとき)

(カウントクロック: フリーランタイムで設定した時間)

<注意事項>

コンペア 0 の一致でフリーランタイム 0 をクリアする設定

(TCCS0.MODE=1) と CMOD=1 の設定が必要です。

計算例: A=1024μs, カウントクロック=125ns

コンペア 0 の値 = $1024000 / 125 - 1 = 8191 = \text{1FFF}_H$

コンペア 1 の値 = $(1024000 / 4) / 125 - 1 = 1023 = \text{7FF}_H$ (デューティ 1/4 のとき)

$(1024000 \times 3/4) / 125 - 1 = 1023 = \text{BFF}_H$ (デューティ 3/4 のとき)

■ インプットキャプチャのプログラム例

設定手順例

IN0 に入力するパルスの立ち上りを検出してフリーランタイマの値を記録する。これを 2 回繰り返してトリガからトリガまでの時間を計測する。ただし、キャプチャ値の読出しと計算処理は割込み処理とする。

< 初期設定 >

- ・フリーランタイマの制御

制御レジスタの設定	TCCS
クロック選択 >>	.ECKE
割込み要求フラグ >>	.IVF
割込み要求許可 >>	.IVFE
カウント動作 >>	.STOP
タイマの初期化条件 >>	.MODE
TCDT クリア >>	.SCLR
カウントクロック >>	.CLK2-0
タイマデータ値の設定	TCDT

- ・ポート

レジスタ名 . ビット名	
ポートの IN0 入力設定	DDR9 .P96

- ・インプットキャプチャの制御

制御レジスタの設定	ICS01
割込み要求フラグ >>	.ICP1, ICP0
割込み要求許可 >>	.ICE1, ICE0
ch.1 有効エッジ極性選択 >>	.EG11, EG10
ch.0 有効エッジ極性選択 >>	.EG01, EG00

- ・割込み関連

割込みレベルの設定	ICR07
I フラグの設定	(CCR)

- ・変数の設定

< 起動 >

- ・インプットキャプチャ ch.0 起動

レジスタ名 . ビット名	
割込み制御	ICS01 .ICE0

- ・フリーランタイマ起動

レジスタ名 . ビット名	
カウント動作起動	TCCS .STOP

< 割込み >

- ・割込み処理

レジスタ名 . ビット名	
割込み要求フラグのクリア	ICS01 .ICP0
(任意の処理)	

< 割込みベクタ >

- ・ベクタテーブルの設定

(注意事項)

事前にクロック関連の設定および、`__set_il(数値)` の設定が必要です。クロックおよび割込みの章をご参照ください。

プログラム例

```
void INPUT0_sample(void)
```

```
{
    freerun_initial();
    INPUT0_initial();
    INPUT0_start();
    freerun_start();
}
```

```
void freerun_initial(void)
```

```
{
    IO_TCCS.word = 0x20; /* 設定値 =0000_0000_0010_0000 */
    /* bit15 = 0 ECKE 内部クロックソース */
    /* bit7 = 0 IVF 割込み要求フラグ */
    /* bit6 = 0 IVFE 割込み禁止 */
    /* bit5 = 1 STOP カウント禁止 */
    /* bit4 = 0 MODE リセット, クリアビットによる初期化 */
    /* bit3 = 0 SCLR フリーランタイマ値の初期化 */
    /* bit2-0 = 000 CLK2-0 カウントクロック φ */
    /* タイマデータ値の初期化 */
    IO_TCDT = 0x0000;
}
```

```
void INPUT0_initial(void)
```

```
{
    IO_DDR9.byte = 0x00; /* DDR9 IN0(P96) 入力 */
    IO_ICS01.byte = 0x01; /* 設定値 =0000_0001 */
    /* bit7-6 = 00 ICP1, 0 有効エッジフラグクリア */
    /* bit5-4 = 00 ICE1, 0 割込み禁止 */
    /* bit3-2 = 00 EG11, EG10 ch.1 エッジ検出なし */
    /* bit1-0 = 01 EG01, EG00 ch.0 立ち上がりエッジ検出 */
    IO_ICR07.byte = 0x10; /* インプットキャプチャ ch.0 割込みレベル設定
    ( 値は任意 ) */
    __EI(); /* 割込み許可 */
    count = 0;
}
```

```
void INPUT0_start(void)
```

```
{
    IO_ICS01.bit.ICE0 = 1; /* bit4 = 1 ICE0 ch.0 割込み許可 */
}
void freerun_start(void)
```

```
{
    IO_TCCS.bit.STOP = 0; /* bit5 = 0 STOP カウント許可 */
}
__interrupt void INPUT0_int(void)
```

```
{
    IO_ICS01.bit.ICP0 = 0; /* bit6 = 0 ICP0 有効エッジ検出フラグのクリア */
    if(count==0)
        Data1 = IO_IPCP0; /* フリーランタイマ値を記録する。(1 回目) */
    else if(count==1) {
        Data2 = IO_IPCP0; /* フリーランタイマ値を記録する。(2 回目) */
        cycle = (data2-data1)*125; /* 時間を計測する。 */
    }
    count++;
}
```

```
#pragma intvect INPUT0_int 26
```

(注意事項)

レジスタの記述形式については、「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」を参照ください。

■ プログラム例以外の設定方法

● 外部入力の有効エッジ極性の種類と選択方法

有効エッジ極性は、立上り、立下り、両エッジの 3 種類です。

外部入力の有効エッジ極性ビット (ICS01.EG[01:00], ICS01.EG[11:10]) で設定します。

動作	外部入力の有効エッジ極性ビット (EG[01:00], EG[11:10])
立上りエッジを選択するには	"00" を選択する
立下りエッジを選択するには	"10" を選択する
両エッジを選択するには	"11" を選択する

● 外部入力端子 (IN0, IN1) に設定する方法

データ方向ビット (DDR9.P96, DDR9.P97) で行います。

動作	データ方向ビット (P96, P97)
外部入力端子 (IN0, IN1) に設定するには	"0" にする

● 割込み関連レジスタ

チャンネル、割込みレベル、ベクタの関係は下表のとおりです。

割込みレベル、割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

チャンネル	割込みベクタ	割込みレベル設定レジスタ
インプット キャプチャ 0	#26 アドレス : FFFF94 _H	割込みレベルレジスタ (ICR07) アドレス : 0000B7 _H
インプット キャプチャ 1	#27 アドレス : FFFF90 _H	割込みレベルレジスタ (ICR08) アドレス : 0000B8 _H

割込み要求フラグ (ICS01.ICP0, ICS01.ICP1) は自動的にクリアしません。割込み処理から復帰する前にソフトウェアにてインプットキャプチャ割込み要求フラグ (ICP1, ICP0) に "0" を書き込みクリアしてください。

● 割込みの種類

割込みは 1 種類のみです。入力信号のエッジ検出で発生します。

● 割込みを許可する方法

割込み許可は、割込み要求許可ビット (ICS01.ICE0, ICS01.CE1) で設定します。

	割込み要求許可ビット (ICE0, CE1)
割込み禁止	"0" にする
割込み許可	"1" にする

割込み要求のクリアは、割込み要求ビット (ICS01.ICP0, ICS01.ICP1) で設定します。

	割込要求ビット (ICP0, ICP1)
割込み要求クリア	"0" を書き込む

● 入力信号のパルス幅を測定する方法

- "H" 幅測定：

エッジ検出に両エッジを指定します。

最初に立上りエッジを検出し次に立下りエッジを検出します。

パルス幅 = { 立下り時に記録した値 (インプットキャプチャレジスタ値) +
 "10000_H" × オーバフロー回数 -
 立上り時に記録した値 (インプットキャプチャレジスタ値) } ×
 フリーランタイマのカウントクロック幅

例：立下り時に記録した値 = 2320_H, 立上り時に記録した値 = A635_H,

オーバフロー回数 = 1, カウントクロック = 125ns

==> パルス幅 = (2320_H + 10000_H - A635_H) × 125ns = 3997.375μs

- 周期測定：

エッジ検出に立上り (または立下り) を指定します。

エッジを 2 回検出

周期 = { 2 回目に記録した値 (インプットキャプチャレジスタ値) +

"10000_H" × オーバフロー回数 -

1 回目に記録した値 (インプットキャプチャレジスタ値) } ×

フリーランタイマのカウントクロック幅

第13章

8/16 ビットアップダウン カウンタ / タイマ

8/16 ビットアップダウンカウンタ / タイマの概要，構成，レジスタの構成と機能，割込みおよび動作について説明します。

- 13.1 8/16 ビットアップダウンカウンタ / タイマの概要
- 13.2 8/16 ビットアップダウンカウンタ / タイマの構成
- 13.3 8/16 ビットアップダウンカウンタ / タイマのレジスタの構成と機能
- 13.4 8/16 ビットアップダウンカウンタ / タイマの割込み
- 13.5 8/16 ビットアップダウンカウンタ / タイマの動作
- 13.6 8/16 ビットアップダウンカウンタ / タイマのプログラム例

13.1 8/16 ビットアップダウンカウンタ / タイマの概要

8/16 ビットアップダウンカウンタ / タイマは、イベント入力端子 6 本、8 ビットアップ / ダウンカウントレジスタ × 2 本、8 ビットリロード / コンペアレジスタ × 2 本と、それらの制御回路から構成されています。

■ 8/16 ビットアップダウンカウンタ / タイマの主な機能

- 8 ビットカウントレジスタにより、0 ～ 255 の範囲でカウントできます。
(16 ビット × 1 動作モードでは、0 ～ 65535 の範囲でカウント可能です。)
- カウントクロックの選択による 4 種のカウントモードがあります。
 - タイマモード
 - アップ / ダウンカウントモード
 - 位相差ダウンカウントモード (2 倍)
 - 位相差ダウンカウントモード (8 倍)
- タイマモード時には、カウントクロックとして 2 種類の内部クロックから選択してください。
 - 2 分周 (内部マシンサイクル $f=25\text{MHz}$ 時 80ns)
 - 8 分周 (内部マシンサイクル $f=25\text{MHz}$ 時 320ns)
- アップ / ダウンカウントモード時には、外部端子入力信号の検出エッジを選択できます。
 - 立下りエッジ検出
 - 立上りエッジ検出
 - 立下り / 立上りの両エッジ検出
 - エッジ検出禁止
- 位相差カウントモードは、モータなどのエンコーダのカウントに適しています。エンコーダの A 相、B 相、Z 相出力をそれぞれ入力することによって、高精度で回転角度、回転数などのカウントを容易に行えます。
- ZIN 端子は、2 種の機能を選択できます。
 - カウンタクリア機能
 - ゲート機能
- コンペア機能およびリロード機能を有し、それぞれの機能のみで、または組み合わせても使用可能です。両機能を起動することによって、任意幅でのアップダウンカウントを行えます。
 - コンペア機能 (コンペア時に割込み出力)
 - コンペア機能 (コンペア時に割込み出力およびカウンタクリア)
 - リロード機能 (アンダフロー時に割込み出力およびリロード)
 - コンペア / リロード機能 (コンペア時に割込み出力およびカウンタクリア / アンダフロー時に割込み出力およびリロード)
 - コンペア / リロード禁止

- コンペア時, リロード (アンダフロー) 時およびオーバフロー時の割込み発生を, それぞれ個別に制御できます。
- カウント方向フラグにより, 直前のカウント方向を識別できます。
- カウント方向が変わった場合に割込みを発生します。

13.2 8/16 ビットアップダウンカウンタ / タイマの構成

8 ビットアップダウンカウンタ / タイマは、2 チャンネルあり 1 チャンネルごとに、イベント入力端子 3 本、8 ビットアップ / ダウンカウント × 1 本、8 ビットリロード / コンペアレジスタ × 1 本から構成されています。また、8 ビットアップダウンカウンタ / タイマ 2 チャンネルを 1 チャンネルとして 16 ビットアップダウンカウンタ / タイマとして使用可能です。(16 ビットアップダウンカウンタ / タイマとして使用する場合、0 チャンネルのレジスタが有効になります。)

■ 8/16 ビットアップダウンカウンタ / タイマのブロックダイアグラム

図 13.2-1 および図 13.2-2 に、8/16 ビットアップダウンカウンタ / タイマのブロックダイアグラムを示します。

図 13.2-1 8/16 ビットアップダウンカウンタ / タイマのブロックダイアグラム (ch.0)

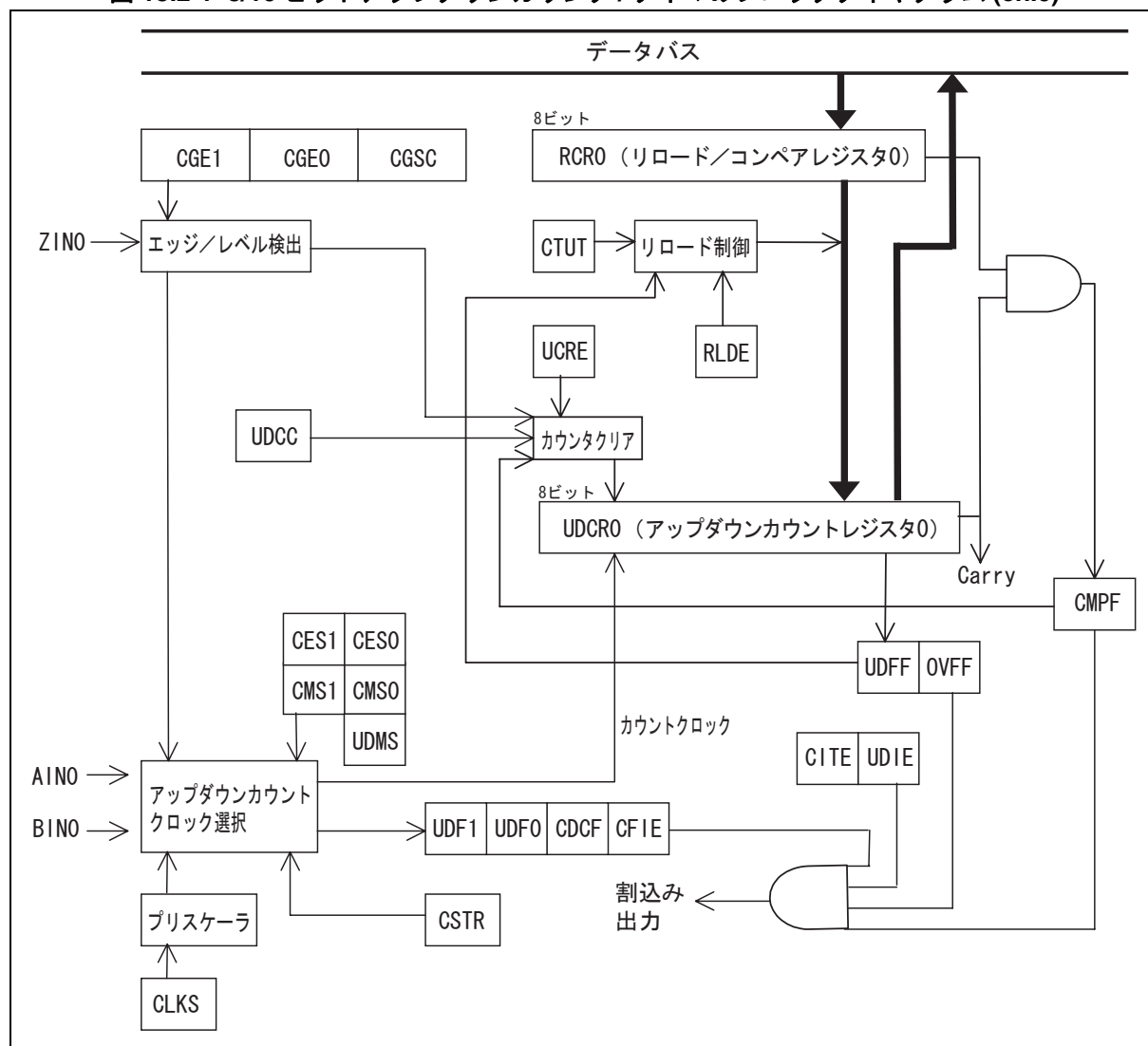
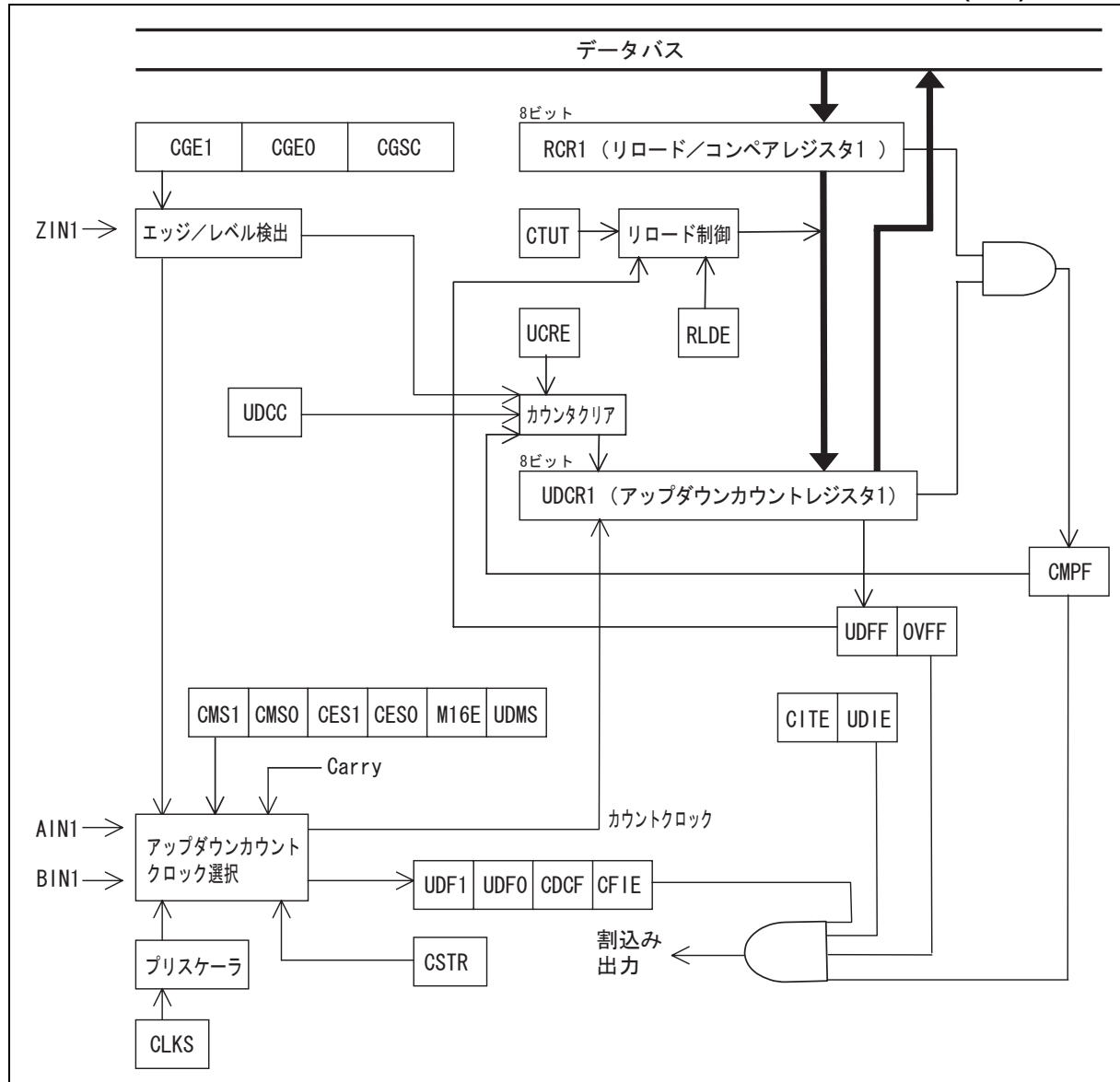


図 13.2-2 8/16 ビットアップダウンカウンタ / タイマのブロックダイアグラム (ch.1)



■ 8/16 ビットアップダウンカウンタ / タイマに関する端子

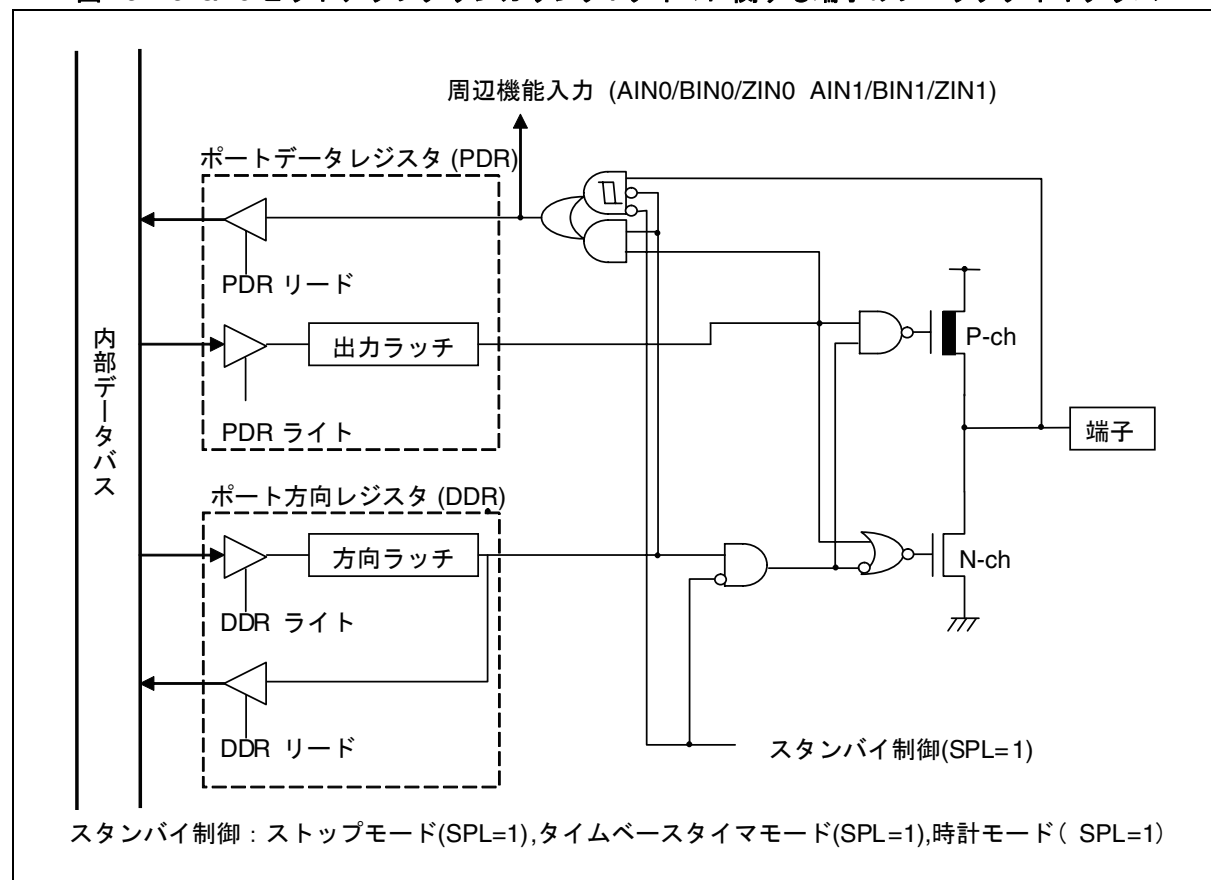
8/16 ビットアップダウンカウンタ / タイマに関する端子は、AIN0/BIN0/ZIN0, AIN1/BIN1/ZIN1 端子があります。AIN0/BIN0/ZIN0 端子は、汎用入出力ポート (AIN0/P30, BIN0/P31, ZIN0/P32) と、AIN1/BIN1/ZIN1 端子は、汎用入出力ポート (AIN1/P33, BIN1/P34, ZIN1/P35) と、アップダウンカウンタ / タイマの入力端子として機能を兼用しています。

● AIN0/BIN0/ZIN0, AIN1/BIN1/ZIN1 端子として使用する場合の設定

AIN/BIN/ZIN 入力端子として使用の場合は、AIN0/P30, BIN0/P31, ZIN0/P32 端子と AIN1/P33, BIN1/P34, ZIN1/P35 端子は、ポート方向レジスタにより入力ポート (DDR3 bit8, 9, 10, 11, 12, 13 → "0") に設定してください。

■ 8/16 ビットアップダウンカウンタ / タイマに関する端子のブロックダイアグラム

図 13.2-3 8/16 ビットアップダウンカウンタ / タイマに関する端子のブロックダイアグラム



13.3 8/16 ビットアップダウンカウンタ / タイマのレジスタの構成と機能

8/16 ビットアップダウンカウンタ / タイマのレジスタの構成と機能について説明します。

■ 8/16 ビットアップダウンカウンタ / タイマのレジスタ一覧

図 13.3-1 に、8/16 ビットアップダウンカウンタ / タイマのレジスタ一覧を示します。

図 13.3-1 8/16 ビットアップダウンカウンタ / タイマのレジスタ一覧

		15	8 7								0							
		UDCR1								UDCR 0								
		RCR1								RCR 0								
		予約領域								CSR 0								
		CCR0								CCRL 0								
		予約領域								CSR 1								
		CCR1								CCRL 1								
		← 8ビット →								← 8ビット →								
ch. 0	UDCR0		7	6	5	4	3	2	1	0	初期値							
	アドレス:	000068 _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _B							
			R	R	R	R	R	R	R	R								
ch. 1	UDCR1		15	14	13	12	11	10	9	8	初期値							
	アドレス:	000069 _H	D17	D16	D15	D14	D13	D12	D11	D10	00000000 _B							
			R	R	R	R	R	R	R	R								
ch. 0	RCR0		7	6	5	4	3	2	1	0	初期値							
	アドレス:	00006A _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _B							
			W	W	W	W	W	W	W	W								
ch. 1	RCR1		15	14	13	12	11	10	9	8	初期値							
	アドレス:	00006B _H	D17	D16	D15	D14	D13	D12	D11	D10	00000000 _B							
			W	W	W	W	W	W	W	W								
ch. 0	CSR0		7	6	5	4	3	2	1	0	初期値							
	アドレス:	000072 _H	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	00000000 _B							
ch. 1	CSR1		アドレス:	000074 _H	R/W	R/W	R/W	R/W	R/W	R	R							
ch. 0	CCRL0		7	6	5	4	3	2	1	0	初期値							
	アドレス:	00006C _H	UDMS	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	0X00X000 _B							
ch. 1	CCRL1		アドレス:	000070 _H	R/W	W	R/W	R/W	W	R/W	R/W	R/W						
ch. 0	CCR0		15	14	13	12	11	10	9	8	初期値							
	アドレス:	00006D _H	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	00000000 _B							
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W								
ch. 1	CCR1		15	14	13	12	11	10	9	8	初期値							
	アドレス:	000071 _H	—	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	-00000000 _B							
			-	R/W	R/W	R/W	R/W	R/W	R/W	R/W								

13.3.1 カウンタコントロールレジスタ (ch.0) 上位 (CCRHO)

カウンタコントロールレジスタ (ch.0) 上位 (CCRHO) の構成および機能について説明します。

■ カウンタコントロールレジスタ (ch.0) 上位 (CCRHO)

カウンタコントロールレジスタ (ch.0) 上位 (CCRHO) のビット構成を、下図に示します。

図 13.3-2 カウンタコントロールレジスタ (ch.0) 上位 (CCRHO) のビット構成

ビット		15	14	13	12	11	10	9	8	初期値
CCRHO		M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	00000000 _B
ch. 0 アドレス: 00006D _H		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

以下に、カウンタコントロールレジスタ (ch.0) 上位 (CCRHO) の各ビットの機能を説明します。

[bit15] M16E(16 ビットモード許可)

このビットは、8 ビット×2 チャンネル / 16 ビット×1 チャンネル動作モードを選択 (切換え) します。

M16E	16 ビットモード許可設定
0	8 ビット×2 チャンネル動作モード (初期値)
1	16 ビット×1 チャンネル動作モード

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

[bit14] CDCF(カウント方向転換フラグ)

このビットは、カウント方向が変わった場合にセットされるフラグです。カウント起動中にカウント方向がアップ→ダウンまたはダウン→アップに変わった場合にセットされます。

初期化 ("0" を書き込む) のみ可能です。

リードモディファイライト系命令の読み込み時は常に "1" が読み込まれます。

CDCF	方向転換検出
0	方向転換は行われていない (初期値)
1	方向転換が 1 回以上行われた

[bit13] CFIE(カウント方向転換割込みイネーブル)

このビットは、CDCF がセットされた場合に、CPU に対して割込み出力を制御します。このビットを "1" にセットした場合は、カウント起動中に一度でもカウント方向が変わった場合に割込みが発生します。

CFIE	方向転換割込み出力
0	方向転換割込み出力禁止 (初期値)
1	方向転換割込み出力許可

[bit12] CLKS(内蔵プリスケアラ選択)

このビットは、タイマモード選択時に、内蔵プリスケアラの周波数を選択します。
タイマモードでのみ有効であり、タイマモードではダウンカウントのみとなります。

CLKS	選択内部クロック
0	2 マシンサイクル (初期値)
1	8 マシンサイクル

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

[bit11, bit10] CMS1, CMS0(カウントモード選択)

これらのビットは、カウントモードを選択します。

CMS1	CMS0	カウントモード
0	0	タイマモード [ダウンカウント] (初期値)
0	1	アップ / ダウンカウントモード
1	0	位相差カウントモード 2 通倍
1	1	位相差カウントモード 4 通倍

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

[bit9, bit8] CES1, CES0(カウントクロックエッジ選択)

これらのビットは、アップ / ダウンカウントモードにおいて、外部端子 AIN および BIN の検出するエッジを選択します。

アップ / ダウンカウントモード以外では、この設定は無効です。

CES1	CES0	選択エッジ
0	0	エッジ検出禁止 (初期値)
0	1	立下りエッジ検出
1	0	立上りエッジ検出
1	1	立上り / 立下り両エッジ検出

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

13.3.2 カウンタコントロールレジスタ (ch.1) 上位 (CCR1H)

カウンタコントロールレジスタ (ch.1) 上位 (CCR1H) の構成および機能について説明します。

■ カウンタコントロールレジスタ (ch.1) 上位 (CCR1H)

カウンタコントロールレジスタ (ch.1) 上位 (CCR1H) のビット構成を、下図に示します。

図 13.3-3 カウンタコントロールレジスタ (ch.1) 上位 (CCR1H) のビット構成

ビット		15	14	13	12	11	10	9	8	初期値
CCR1H	ch. 1 アドレス: 000071 _H	—	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	-0000000 _B
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	

以下に、カウンタコントロールレジスタ (ch.1) 上位 (CCR1H) の各ビットの機能を説明します。

[bit14] CDCF(カウント方向転換フラグ)

このビットは、カウント方向が変わった場合にセットされるフラグです。カウント起動中にカウント方向がアップ→ダウンまたはダウン→アップに変わった場合にセットされます。

初期化 ("0" を書き込む) のみ可能です。

リードモディファイライト系命令の読み込み時は常に "1" が読み込まれます。

CDCF	方向転換検出
0	方向転換は行われていない (初期値)
1	方向転換が 1 回以上行われた

[bit13] CFIE(カウント方向転換割込みイネーブル)

このビットは、CDCF がセットされた場合に、CPU に対して割込み出力を制御します。このビットを "1" にセットした場合は、カウント起動中に一度でもカウント方向が変わった場合に割込みが発生します。

CFIE	方向転換割込み出力
0	方向転換割込み出力禁止 (初期値)
1	方向転換割込み出力許可

[bit12] CLKS(内蔵プリスケラ選択)

このビットは、タイマモード選択時に、内蔵プリスケラの周波数を選択します。

タイマモードでのみ有効であり、タイマモードではダウンカウントのみとなります。

CLKS	選択内部クロック
0	2 マシンサイクル (初期値)
1	8 マシンサイクル

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

[bit11, bit10] CMS1, CMS0(カウントモード選択)

これらのビットは, カウントモードを選択します。

CMS1	CMS0	カウントモード
0	0	タイマモード [ダウンカウント] (初期値)
0	1	アップ / ダウンカウントモード
1	0	位相差カウントモード 2 通倍
1	1	位相差カウントモード 4 通倍

起動後にこのビットを書き換えた場合は, カウント値の保証ができません。

[bit9, bit8] CES1, CES0(カウントクロックエッジ選択)

これらのビットは, アップ / ダウンカウントモードにおいて, 外部端子 AIN および BIN の検出するエッジを選択します。

アップ / ダウンカウントモード以外では, この設定は無効です。

CES1	CES0	選択エッジ
0	0	エッジ検出禁止 (初期値)
0	1	立下りエッジ検出
1	0	立上りエッジ検出
1	1	立上り / 立下り両エッジ検出

起動後にこのビットを書き換えた場合は, カウント値の保証ができません。

13.3.3 カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)

カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1) の構成および機能について説明します。

■ カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)

カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1) のビット構成を、下図に示します。

図 13.3-4 カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1) のビット構成

		ビット								初期値
CCRL0	ch. 0	7	6	5	4	3	2	1	0	
アドレス:	00006C _H	UDMS	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	0X00X000 _B
CCRL1	ch. 1	7	6	5	4	3	2	1	0	
アドレス:	000070 _H	R/W	W	R/W	R/W	W	R/W	R/W	R/W	

以下に、カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1) の各ビットの機能を示します。

[bit7] UDMS(アップダウンモード選択)

このビットは、位相差カウンタモード2 通倍モードにおいて、BIN 端子の立下りエッジで検出する場合のアップダウンを制御します。

リセット時は "0" に初期化されます。読出しおよび書込みが可能です。

UDMS	動 作
0	BIN 端子の立下りエッジで検出した AIN 端子の値が "H" のとき、ダウンカウント (初期値) BIN 端子の立下りエッジで検出した AIN 端子の値が "L" のとき、アップカウント (初期値)
1	BIN 端子の立下りエッジで検出した AIN 端子の値が "L" のとき、ダウンカウント BIN 端子の立下りエッジで検出した AIN 端子の値が "H" のとき、アップカウント

起動後にこのビットを書き換えた場合は、カウント値の保証ができません。

[bit6] CTUT(カウンタ書込み)

このビットは、RCR から UDCR へのデータ転送を制御します。

このビットに "1" を書き込むと、RCR から UDCR にデータが転送されます。

"0" 書込みは無効です。

[bit5] UCRE(UDCR クリアイネーブル)

このビットは、コンペアによる UDCR のクリアを制御します。

コンペア発生によるクリア以外の UDCR クリア機能 (ZIN 端子によるものなど) には影響しません。

UCRE	コンペアによるカウンタのクリア
0	カウンタクリア禁止 (初期値)
1	カウンタクリア許可

[bit4] RLDE(リロードイネーブル)

このビットは、リロード機能の起動を制御します。リロード機能起動時に UDCR がアンダフローを発生した場合に、RCR の値を UDCR に転送します。

RLDE	リロード機能
0	リロード機能禁止 (初期値)
1	リロード機能許可

[bit3] UDCC(UDCR クリア)

このビットは、UDCR をクリアします。このビットに "0" を書き込むと、UDCR が "0000_H" にクリアされます。

"1" の書込みは無効です。

[bit2] CGSC(カウンタクリア / ゲート選択)

このビットは、外部端子 ZIN の機能を選択します。

CGSC	ZIN の機能
0	カウンタクリア機能 (初期値)
1	ゲート機能

[bit1, bit0] CGE1, CGE0(カウンタクリア / ゲートエッジ選択)

これらのビットは、外部端子 ZIN の検出エッジ / レベルを選択します。

CGE1	CGE0	カウンタクリア機能選択時	ゲート機能選択時
0	0	エッジ検出禁止 (初期値)	レベル検出禁止 (カウントディセーブル)
0	1	立下りエッジ	"L" レベル
1	0	立上りエッジ	"H" レベル
1	1	設定禁止	設定禁止

起動後にこのビットを書き換えた場合は、カウンタ値の保証ができません。

13.3.4 カウンタステータスレジスタ 0/1(CSR0/1)

カウンタステータスレジスタ 0/1(CSR0/1) の構成および機能について説明します。

■ カウンタステータスレジスタ 0/1(CSR0/1)

カウンタステータスレジスタ 0/1(CSR0/1) のビット構成を、下図に示します。

図 13.3-5 カウンタステータスレジスタ 0/1(CSR0/1) のビット構成

ビット		7	6	5	4	3	2	1	0	初期値
CSR0 ch. 0	アドレス: 000072 _H	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	00000000 _B
CSR1 ch. 1	アドレス: 000074 _H	R/W	R/W	R/W	R/W	R/W	R/W	R	R	

以下に、カウンタステータスレジスタ 0/1(CSR0/1) の各ビットの機能を説明します。

[bit7] CSTR(カウント起動)

このビットは、UDCR のカウント動作の起動 / 停止を制御します。

CSTR	カウント動作の起動 / 停止
0	カウント動作停止 (初期値)
1	カウント動作起動

[bit6] CITE(コンペア割込み出力制御)

このビットは、CMPF がセットされた (コンペアが発生した) 場合に、CPU への割込み出力の許可 / 禁止を制御します。

CITE	コンペア割込み出力の許可 / 禁止
0	コンペア割込み出力の禁止 (初期値)
1	コンペア割込み出力の許可

[bit5] UDIE(オーバフロー / アンダフロー割込み出力制御)

このビットは、OVFF/UDFF がセットされた (オーバフロー / アンダフローが発生した) 場合に、CPU への割込み出力の許可 / 禁止を制御します。

UDIE	オーバフロー / アンダフロー割込み出力の許可 / 禁止
0	オーバフロー / アンダフロー割込み出力の禁止 (初期値)
1	オーバフロー / アンダフロー割込み出力の許可

[bit4] CMPF(コンペア検出フラグ)

このビットは、UDCR の値と RCR の値の比較結果が等しくなったことを示すフラグです。

初期化 ("0" を書き込む) のみ可能です。

リードモディファイライト系命令の読み込み時は常に "1" が読み込まれます。

CMPF	コンペア検出比較の一致 / 不一致
0	比較結果が一致していない (初期値)
1	比較結果が一致した

[bit3] OVFF(オーバフロー検出フラグ)

このビットは、オーバフローの発生を示すフラグです。

初期化 ("0" を書き込む) のみ可能です。

リードモディファイライト系命令の読み込み時は常に "1" が読み込まれます。

OVFF	オーバフロー発生の有無
0	オーバフローなし (初期値)
1	オーバフローあり

[bit2] UDFF(アンダフロー検出フラグ)

このビットは、アンダフローの発生を示すフラグです。

初期化 ("0" を書き込む) のみ可能です。

リードモディファイライト系命令の読み込み時は常に "1" が読み込まれます。

UDFF	アンダフロー発生の有無
0	アンダフローなし (初期値)
1	アンダフローあり

[bit1, bit0] UDF1, UDF0(アップダウンフラグ)

これらのビットは、直前のカウント動作 (アップ / ダウン) を示すビットです。

読出しのみ可能で、書込みはできません。

UDF1	UDF0	検出エッジ
0	0	入力なし (初期値)
0	1	ダウンカウント
1	0	アップカウント
1	1	アップ / ダウン同時発生

13.3.5 アップダウンカウントレジスタ (ch.0/ch.1)(UDCR0/1)

アップダウンカウントレジスタ (ch.0/ch.1)(UDCR0/1) の構成および機能について説明します。

■ アップダウンカウントレジスタ (ch.0/ch.1)(UDCR0/1)

アップダウンカウントレジスタ (ch.0/ch.1) (UDCR0/1) のビット構成を、下図に示します。

図 13.3-6 アップダウンカウントレジスタ (ch.0/ch.1)(UDCR0/1) のビット構成

UDCR1										初期値	
ch. 1	アドレス: 000069 _H	ビット	15	14	13	12	11	10	9	8	00000000 _B
			D17	D16	D15	D14	D13	D12	D11	D10	
			R	R	R	R	R	R	R	R	
UDCR0										初期値	
ch. 0	アドレス: 000068 _H	ビット	7	6	5	4	3	2	1	0	00000000 _B
			D07	D06	D05	D04	D03	D02	D01	D00	
			R	R	R	R	R	R	R	R	

このレジスタは、8 ビットのカウンタレジスタです。内部プリスケアラ、または AIN 端子と BIN 端子の入力によってアップ / ダウンカウント動作を行います。

また 16 ビットカウンタモードでは、16 ビットのカウンタレジスタとして動作します。この場合、上位 8 ビット側の制御レジスタの設定値は、動作上無効になります。

このレジスタは、直接には書き込めません。このレジスタに書き込む場合は、RCR を介して行う必要があります。このレジスタに書き込みたい値をまず、RCR に書き込み、その後 CCRL の CTUT ビットに "1" を書き込むことによって、RCR の値が本レジスタに転送されます (ソフトウェアによるリロード)。

本レジスタは、ワードアクセスで読み出してください。

13.3.6 リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1)

リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1) の構成および機能について説明します。

■ リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1)

リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1) のビット構成を、下図に示します。

図 13.3-7 リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1) のビット構成

RCR1	15	14	13	12	11	10	9	8	初期値
ch. 1 アドレス: 00006B _H	D17	D16	D15	D14	D13	D12	D11	D10	00000000 _B
	W	W	W	W	W	W	W	W	
RCR0	7	6	5	4	3	2	1	0	初期値
ch. 0 アドレス: 00006A _H	D07	D06	D05	D04	D03	D02	D01	D00	00000000 _B
	W	W	W	W	W	W	W	W	

リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1) は、リロード値およびコンペア値を設定します。リロード値とコンペア値は同一であり、リロード機能およびコンペア機能を起動することによって 00_H ~ RCR の値 (16 ビット動作モード : 0000_H ~ RCR 値) の間でアップ / ダウンカウントが可能になります。

このレジスタは、書込みのみ可能で、読出しはできません。CCR0/1 の CTUT ビットに "1" を書き込むことによって、このレジスタの値を UDCR に転送する (ソフトウェアによるリロード) ことができます。

このレジスタには、ワードアクセスで書き込んでください。

13.4 8/16 ビットアップダウンカウンタ / タイマの割込み

8/16 ビットアップダウンカウンタ / タイマの割込みは、カウント起動中に 1 度でもカウント方向が変わった場合と、比較結果一致検出した場合と、オーバフロー / アンダフローが発生した場合に発生します。

8/16 ビットアップダウンカウンタ / タイマの割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) は起動できません。

■ 8/16 ビットアップダウンカウンタ / タイマの割込み

8/16 ビットアップダウンカウンタ / タイマの割込制御ビットと割込要因を、表 13.4-1 に示します。

表 13.4-1 8/16 ビットアップダウンカウンタ / タイマの割込み

	カウント方向検出割込み	オーバフロー / アンダフロー割込み	カウンタコンペアー一致割込み
割込み要求フラグ	CCR0: CDCF (bit14) ch.0 CCR1: CDCF (bit14) ch.1	CSR0: OVFF (bit3) ch.0 UDFF (bit2) CSR1: OVFF (bit3) ch.1 UDFF (bit2)	CSR0: CMPF (bit4) ch.0 CSR1: CMPF (bit4) ch.1
割込み要求出力許可ビット	CCR0: CFIE (bit13) ch.0 CCR1: CFIE (bit13) ch.1	CSR0: UDIE (bit5) ch.0 CSR1: UDIE (bit5) ch.1	CSR0: CITE (bit6) ch.0 CSR1: CITE (bit6) ch.1
割込み発生要因	アップダウンカウンタ 方向検出	オーバフロー / アンダ フロー検出	アップダウンカウンタ値 とリロード / コンペアー レジスタ値の一致

CCR0/OCR0 がアップダウンカウンタ端子 (AIN0/BIN0/ZIN0) に対応
CCR1/OCR1 がアップダウンカウンタ端子 (AIN1/BIN1/ZIN1) に対応

● カウント方向転換割込み

カウント方向転換割込みについて発生動作を次に示します。

- ・ カウンタコントロールレジスタ (CCR0/1) bit14:CDCF フラグが "1" にセットされます。
- ・ 割込み要求 (CCR0/1) bit13:CFIE が許可 ("1") されている場合で、カウント起動中に 1 度でもカウント方向が変わった時に割込みが発生します。

● オーバフロー / アンダフロー割込み

オーバフロー / アンダフロー割込みについて発生動作を次に示します。

- ・ カウンタステータスレジスタ (CSR0/1) bit5:UDIE フラグを "1" にセットします。
- ・ カウンタステータスレジスタ (CSR0/1) bit3:OVFF または bit2:UDFF 結果が "1" なら、割込み要求が発生します。

● カウンタコンペアー一致割込み

コンペアー割込みについて発生動作を次に示します。

- ・ カウンタステータスレジスタ (CSR0/1) bit6:CITE フラグを "1" にセットします。
- ・ カウンタステータスレジスタ (CSR0/1) bit4:CMPE で UDCR 値と RCR 値の比較結果が一致した場合、割込み要求が発生します。

■ 8/16 ビットアップダウンカウンタ / タイマの割込みと DMA 転送および EI²OS

表 13.4-2 に、ソフトウェア割込みを除く割込み要因と割込みベクタ、割込み制御レジスタの関係を示します。

表 13.4-2 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS のク リ ア	μDMAC チャネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
8/16 ビットアップダウンカウンタ / タイマ * (ch.0, ch.1) コンペア / アンダフロー / オーバフロー / アップダウン反転	○	×	#25	FFFF98 _H	ICR07	0000B7 _H

×：割込み要求フラグはクリアされません。

○：割込み要求フラグはクリアされます。

*：この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。

詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

8/16 ビットアップダウンカウンタ / タイマは EI²OS 機能のみに対応し、DMA 転送機能には対応していません。EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

13.5 8/16 ビットアップダウンカウンタ / タイマの動作

8/16 ビットアップダウンカウンタ / タイマの各種カウントモードおよびリロード / コンペア機能の動作について説明します。

■ カウントモードの選択

8/16 ビットアップダウンカウンタ / タイマは、4 種のカウントモードを持っています。これらのカウントモードは、CCRH の CMS1, CMS0 で選択します。

表 13.5-1 カウントモードの選択

CMS1	CMS0	カウントモード
0	0	タイマモード (ダウンカウント)
0	1	アップ / ダウンカウントモード
1	0	位相差カウントモード 2 通倍
1	1	位相差カウントモード 4 通倍

● タイマモード (ダウンカウント)

タイマモードでは、内部プリスケアラの出力をダウンカウントします。内蔵プリスケアラについては、CCRH の CLKS によって 2 マシンサイクル / 8 マシンサイクルの選択が可能です。

● アップ / ダウンカウントモード

アップ / ダウンカウントモードでは、外部端子 AIN および BIN の入力をカウントすることでアップ / ダウンカウントを行います。AIN 端子の入力はアップカウントを、BIN 端子の入力はダウンカウントをそれぞれ制御します。

AIN 端子、BIN 端子の入力はエッジ検出用入力であり、CCRH の CES1, CES0 によって検出エッジの選択ができます。

表 13.5-2 検出エッジの選択

CES1	CES0	検出エッジ
0	0	エッジ検出禁止
0	1	立下りエッジ検出
1	0	立上りエッジ検出
1	1	立下り / 立上り両エッジ検出

● 位相差カウントモード (2 通倍 / 4 通倍)

位相差カウントモードでは、エンコーダの出力信号 A 相、B 相の位相差をカウントするため、AIN 端子の入力エッジ検出時に BIN 端子の入力レベルを検出し、BIN 端子の入力エッジ検出時には AIN 端子の入力レベルを検出し、カウントを行います。

2 通倍 / 4 通倍モードでは、AIN 端子入力と BIN 端子入力の位相差について、AIN 端子の方が早い場合にはアップカウントを、BIN 端子の方が早い場合にはダウンカウントを行います。

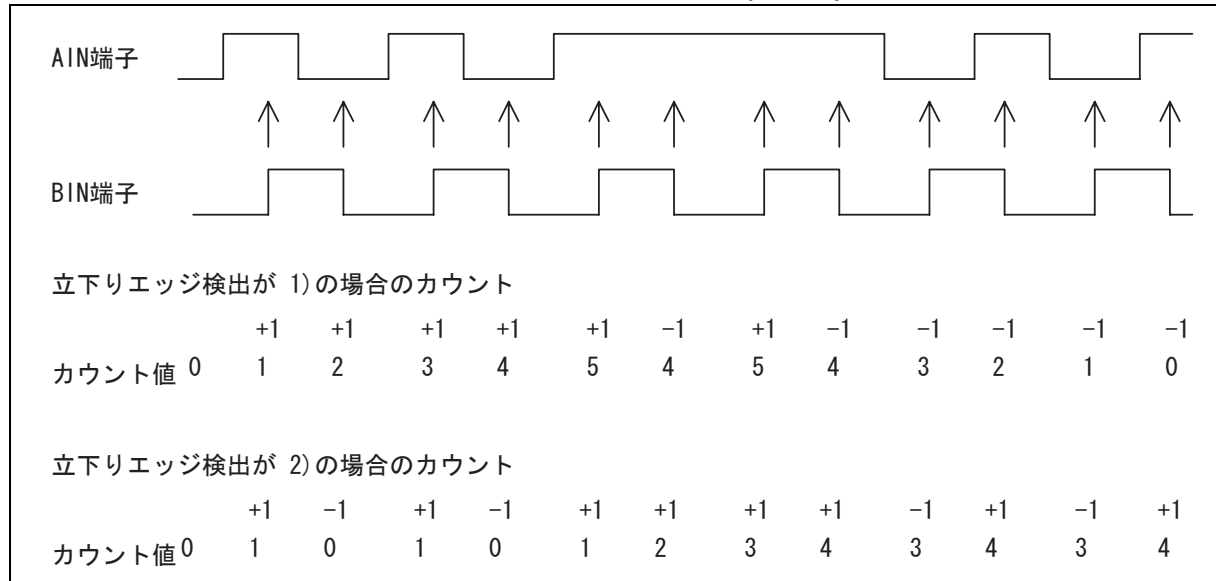
2 通倍モードでは、BIN 端子の立上り / 立下り両方のエッジのタイミングで AIN 端子の値を検出することによってカウントを行います。この場合のカウント動作は、以下のとおりです。

- BIN 端子の立上りエッジで検出した AIN 端子の値が "H" のとき、アップカウント

- BIN 端子の立上りエッジで検出した AIN 端子の値が "L" のとき , ダウンカウント
BIN 端子の立下りエッジで検出する AIN 端子の値は , 以下に示す 1), 2) の状態の 2 種類
から選択できます。

- 1) BIN 端子の立下りエッジで検出した AIN 端子の値が "H" の場合 , ダウンカウント
BIN 端子の立下りエッジで検出した AIN 端子の値が "L" の場合 , アップカウント
- 2) BIN 端子の立下りエッジで検出した AIN 端子の値が "L" の場合 , ダウンカウント
BIN 端子の立下りエッジで検出した AIN 端子の値が "H" の場合 , アップカウント

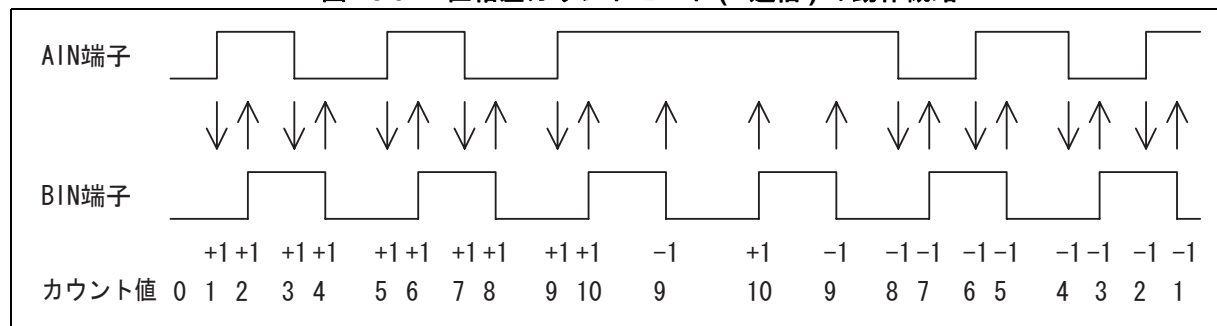
図 13.5-1 位相差カウントモード (2 通倍) の動作概略



4 通倍モードでは , BIN 端子の立上り / 立下り両方のエッジのタイミングで AIN 端子の値を検出し , また AIN 端子の立上り / 立下り両方のエッジのタイミングで BIN 端子の値を検出することでカウントを行います。この場合のカウント動作は , 以下のとおりです。

- BIN 端子の立上りエッジで検出した AIN 端子の値が "H" の場合 , アップカウント
- BIN 端子の立上りエッジで検出した AIN 端子の値が "L" の場合 , ダウンカウント
- BIN 端子の立下りエッジで検出した AIN 端子の値が "H" の場合 , ダウンカウント
- BIN 端子の立下りエッジで検出した AIN 端子の値が "L" の場合 , アップカウント
- AIN 端子の立上りエッジで検出した BIN 端子の値が "H" の場合 , ダウンカウント
- AIN 端子の立上りエッジで検出した BIN 端子の値が "L" の場合 , アップカウント
- AIN 端子の立下りエッジで検出した BIN 端子の値が "H" の場合 , アップカウント
- AIN 端子の立下りエッジで検出した BIN 端子の値が "L" の場合 , ダウンカウント

図 13.5-2 位相差カウントモード (4 通倍) の動作概略



エンコーダ出力のカウントの際に、位相と端子の関係を以下のように入力することによって、回転角度、回転数のカウントおよび回転方向が高精度で検出可能となります。

- A 相を AIN 端子に入力する。
- B 相を BIN 端子に入力する。
- Z 相を ZIN 端子に入力する。

なお、このカウントモードを選択したときは、CCR_H の CES1/0、CCRL の CGE1/0 による検出エッジの選択は無効となります。

13.5.1 リロード / コンペア機能

8/16 ビットアップダウンカウンタ / タイマには、リロード機能およびコンペア機能があります。この 2 つの機能は組み合わせて処理することもできます。

■ リロード / コンペア機能の選択

リロード / コンペア機能の選択例を、表 13.5-3 に示します。

表 13.5-3 リロード / コンペア機能の選択例

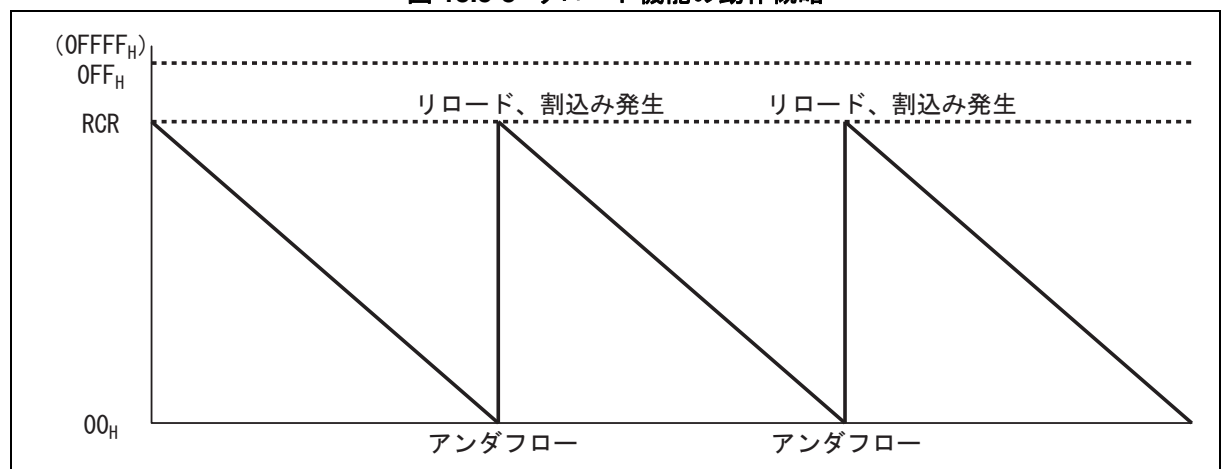
RLDE, UCRE	リロード / コンペア機能
00 _B	リロード / コンペア禁止 (初期値)
01 _B	コンペア許可
10 _B	リロード許可
11 _B	リロード / コンペア許可

■ リロード機能

リロード機能の起動時は、アンダフローが発生した次のダウンカウントクロックのタイミングで、RCR の値を UDCR に転送します。このとき、UDFF がセットされ、割込み要求が発生します。

なお、ダウンカウントをしないモードでは、この機能の起動は無効となります。

図 13.5-3 リロード機能の動作概略

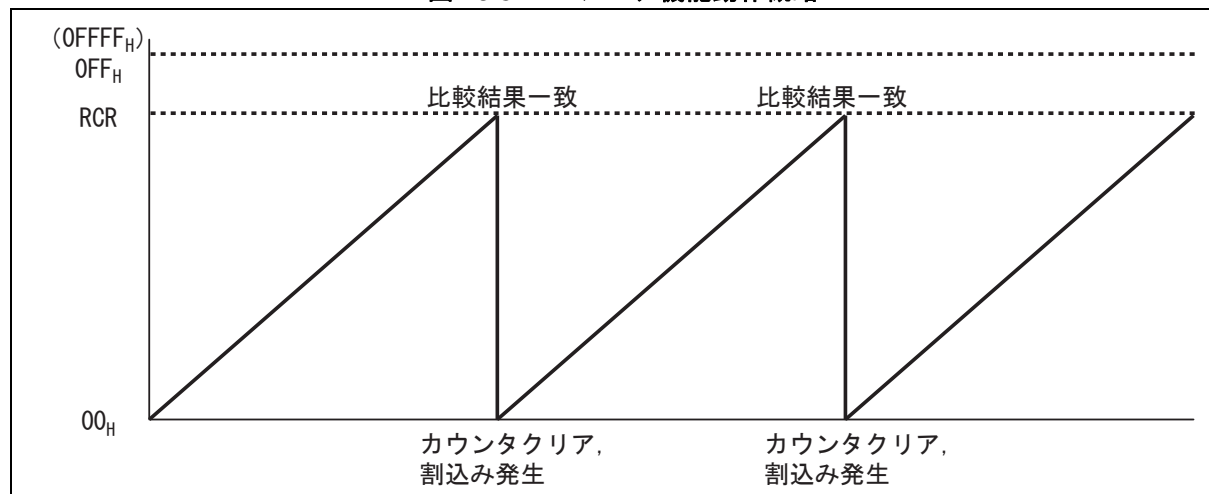


■ コンペア機能

コンペア機能は、タイマモード以外のすべてのモードで使用できます。コンペア機能の起動時は、RCR と UDCR の値が一致した場合に、CMPF がセットされるとともに割込み要求が発生します。また、コンペアクリア機能の起動時は、その次のアップカウントクロックのタイミングで UDCR をクリアします。

なお、アップカウントを行わないモードでは、この機能の起動は無効となります。

図 13.5-4 コンペア機能動作概略

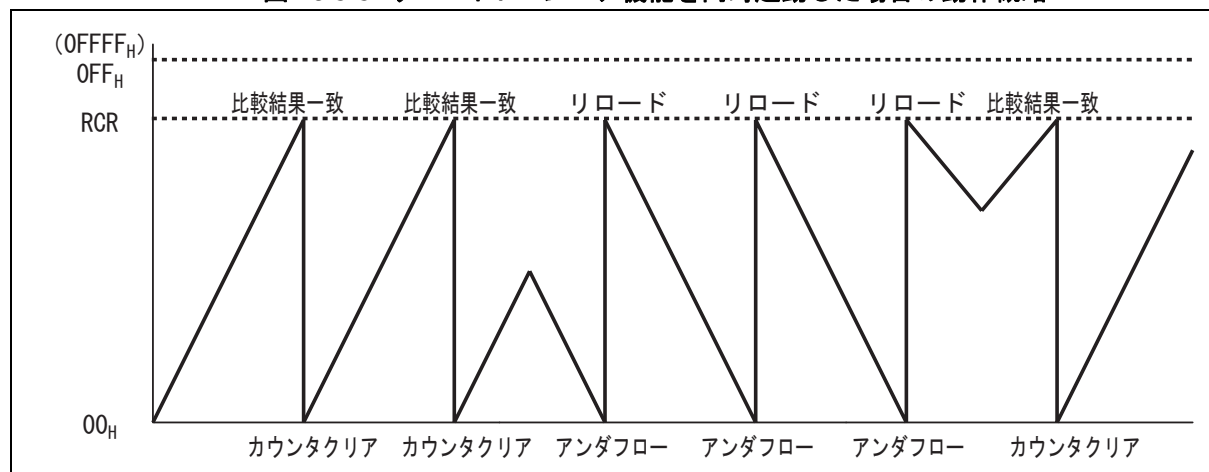


■ リロード/コンペア機能任意幅でのアップ/ダウンカウント

リロード/コンペア機能の起動時は、任意幅でのアップ/ダウンカウントが可能です。

リロード機能により、アンダフローが発生した場合に RCR の値を UDCR に転送します。また、コンペア機能により、RCR と UDCR の値が一致した場合に UDCR をクリアします。この両機能を利用して、00_H ~ RCR の値の間でアップ/ダウンカウントを行います。

図 13.5-5 リロード/コンペア機能を同時起動した場合の動作概略



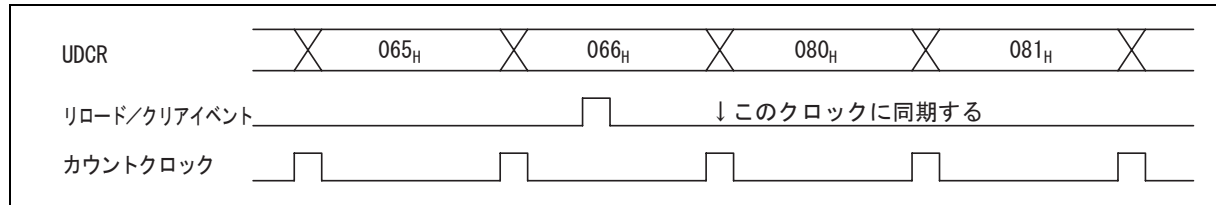
比較結果一致時、またはリロード (アンダフロー) 時に CPU に割込みを発生させることができます。また、これらの割込み出力のイネーブルは個別に制御できます。

UDCR に対しリロードおよびクリアを行う場合のタイミングは、カウント起動中と停止中では異なります。

● カウント動作中にリロード、またはクリアのイベントが発生した場合

UDCR の更新は、すべてカウントクロックに同期して行われます。図 13.5-6 は、080_H をリロードした場合を示します。

図 13.5-6 通常の動作カウント

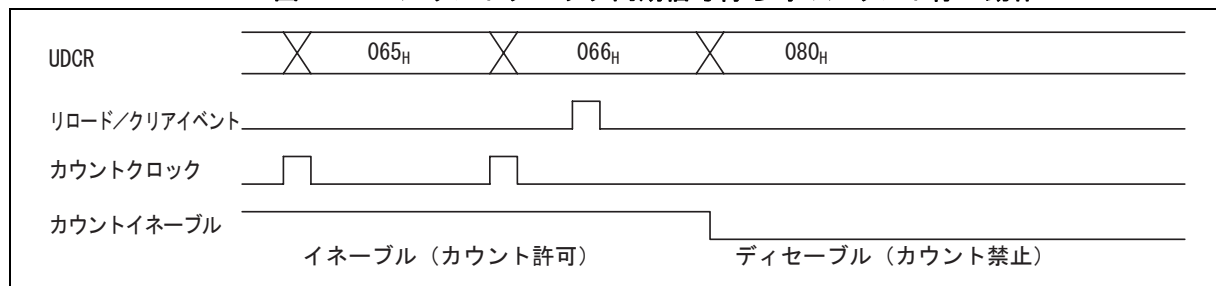


● カウント動作停止直前にリロード、またはクリアのイベントが発生した場合

カウントクロック同期待ち (同期させるためのカウント入力を持っている状態) のままカウントを停止させた場合は、停止した時点で UDCR へのリロード、または UDCR のクリアが行われます。

図 13.5-7 は、080_H をリロードした場合を示します。

図 13.5-7 カウントクロック同期信号待ち時のカウント停止動作

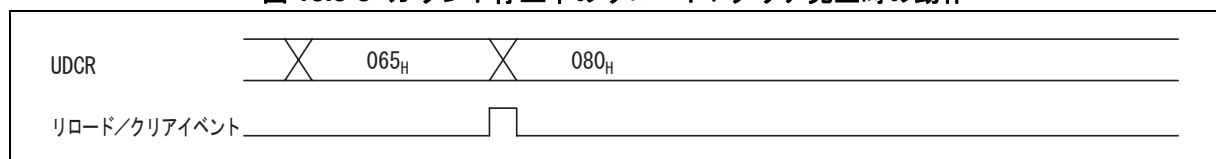


● カウント停止中にリロードおよびクリアのイベントが発生した場合

UDCR の更新は、イベント発生時点で行われます。

図 13.5-8 は、080_H をリロードした場合を示します。

図 13.5-8 カウント停止中のリロード / クリア発生時の動作



● 比較結果一致によるカウンタクリアの場合

UDCR と RCR の値が一致し、さらにアップカウントが行われたときにカウンタクリアが行われます。UDCR と RCR の値が一致した場合でも、その後ダウンカウントが行われたり、カウント停止になった場合は、カウンタクリアは行われません。

カウンタクリアについてはリセット入力以外のすべてのイベントで、またリロードもすべてのイベントで、上記タイミングに従います。

クリアイベントとリロードイベントが同時期に発生した場合は、クリアイベントが優先となります。

13.5.2 アップダウンカウントレジスタ (UDCR) へのデータの書込み

UDCR へデータバスからは直接データを書き込むことはできません。ここでは、UDCR に任意のデータを書き込む場合の手順について説明します。

■ UDCR へのデータの書込み

UDCR へデータを書き込むには、以下の手順で書き込みます。

- 1) UDCR に書き込むデータを、まず RCR に書き込みます。
- 2) CCRH の CTUT に "1" を書き込むことによって、RCR から UDCR にデータが転送されます。

■ カウンタのクリア

カウンタのクリアは、UDCR への "0000_H" 書込み以外に下記の方法があります。

- ・ リセット入力によるクリア (初期化)
- ・ ZIN 端子からのエッジ入力によるクリア
- ・ CCRL の UDCC に "0" を書き込むことによるクリア
- ・ コンペア機能によるクリア

これらのクリアは、カウント起動 / 停止にかかわらず行うことができます。

■ カウントクリア / ゲート機能

ZIN 端子は、CCRH の CGSC によってカウントクリア機能、またはゲート機能のいずれかとして使用することができます。

カウントクリア機能の起動時は、ZIN 端子からのエッジ入力によりカウンタをクリアします。ZIN 端子入力信号のどのエッジでカウントをクリアするかを、CCRL の CGE1, CGE0 で選択します。この機能により、エンコーダの Z 相出力を本端子に入力すると、エンコーダのカウントの起点で UDCR をクリアすることが可能となります。

ゲート機能の起動時は、ZIN 端子からのレベル入力によりカウントのイネーブル / ディセーブルを行います。ZIN 端子入力信号のどのレベルでイネーブルとするかは、CCRL の CGE1, CGE0 で選択します。

この機能は、全カウントモードで使用できます。

表 13.5-4 ZIN 端子機能の選択

CGSC	ZIN 端子の機能	CGE1, CGE0	カウンタクリア機能	ゲート機能
0	カウンタクリア機能	00 _B	検出禁止	検出禁止
1	ゲート機能	01 _B	立上りエッジ	"L" レベル
		10 _B	立下りエッジ	"H" レベル

■ カウント方向フラグ, カウント方向転換フラグ

カウント方向フラグ (UDF1, UDF0) は, アップ / ダウンカウント時に, 直前のカウントがアップカウントであったか, ダウンカウントであったかを示します。AIN, BIN 両端子の入力から生成されたカウントクロックから判断して, カウントのたびにフラグを書き換えます。モータの制御などで, 現在の回転方向を知りたい場合など, このフラグをみることによって回転方向を判別できます。

この機能は, 全カウントモードで使用できます。

表 13.5-5 カウント方向フラグ

UDF1, UDF0	カウント方向
01 _B	ダウンカウント
10 _B	アップカウント
11 _B	アップ / ダウン同時発生 (カウント動作は行われない)

カウント方向転換フラグ (CDCF) は, カウント方向がアップ→ダウンで変わった場合にセットされます。このフラグがセットされるとともに, CPU に対して割込みが発生します。この割込みと, カウント方向フラグ (UDF1, UDF0) を参照することによって, カウント方向がどちらからどちらへ変化したかが分かります。ただし, 方向転換の期間が短く, 連続して発生した場合などは, 方向転換後のフラグの示す方向が元に戻って同一方向となり, 正しいカウント方向転換検出ができない場合がありますので注意が必要です。

表 13.5-6 カウント方向転換フラグ

CDCF	カウント方向転換検出
0	方向転換なし
1	方向転換あり (1 回以上)

13.6 8/16 ビットアップダウンカウンタ / タイマのプログラム例

8/16 ビットアップダウンカウンタ / タイマのプログラム例を示します。

■ 8/16 ビットアップダウンカウンタ / タイマのプログラム例

<p>設定手順例</p> <p>16 ビットモードタイマモード (ダウンカウント) カウントクロック = 8 分周 カウント値をリロードさせインターバルタイマ割込みを発生させる。 割込み要因 = アンダフロー</p> <p><初期設定> ・アップダウンカウンタ ch.0 の制御</p> <table border="1"> <thead> <tr> <th>制御レジスタの設定</th><th>CCR0</th></tr> </thead> <tbody> <tr> <td>16 ビットモード許可設定 >></td><td>.M16E</td></tr> <tr> <td>カウント方向転換 >></td><td>.CDCF</td></tr> <tr> <td>カウント方向転換割込み許可 >></td><td>.CFIE</td></tr> <tr> <td>内蔵プリスケラ選択 >></td><td>.CLKS</td></tr> <tr> <td>カウントモード選択 >></td><td>.CMS1-0</td></tr> <tr> <td>カウントクロックエッジ選択 >></td><td>.CES1-0</td></tr> <tr> <td>アップダウンモード選択 >></td><td>.UDMS</td></tr> <tr> <td>カウンタライト >></td><td>.CTUT</td></tr> <tr> <td>UDCR クリア許可 >></td><td>.UCRE</td></tr> <tr> <td>リロード機能許可 >></td><td>.RLDE</td></tr> <tr> <td>UDCR クリア >></td><td>.UDCC</td></tr> <tr> <td>カウンタクリア / ゲート選択 >></td><td>.CGSC</td></tr> <tr> <td>カウンタクリア / ゲートエッジ選択 >></td><td>.CGE1-0</td></tr> </tbody> </table> <p>・リロード値 / コンペア値設定</p> <table border="1"> <thead> <tr> <th>リロード値設定</th><th>RCR0</th></tr> </thead> <tbody> <tr> <td></td><td>RCR1</td></tr> </tbody> </table> <p>・割込み関連</p> <table border="1"> <thead> <tr> <th>UD カウンタ 0 割込みレベルの設定</th><th>ICR07</th></tr> </thead> <tbody> <tr> <th>I フラグの設定</th><th>(CCR)</th></tr> </tbody> </table> <p><起動> ・アップダウンカウンタ ch.0 起動</p> <p style="text-align: center;">レジスタ名 ビット名</p> <table border="1"> <tbody> <tr> <td>アンダフロー割込み制御</td><td>CSR0.UDIE</td></tr> <tr> <td>RCR から UDCR ヘデータ転送</td><td>CCR0.CTUT</td></tr> <tr> <td>カウント動作起動</td><td>CSR0.CSTR</td></tr> </tbody> </table> <p><割込み> ・割込み処理</p> <table border="1"> <thead> <tr> <th colspan="2">アンダフロー検出フラグのチェック</th></tr> </thead> <tbody> <tr> <td>割込み要求フラグクリア</td><td>CSR0.UDFF</td></tr> <tr> <td colspan="2">(任意の処理)</td></tr> </tbody> </table> <p><割込みベクタ> ・ベクタテーブルの設定</p> <p>(注意事項) 事前にクロック関連の設定および、__set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	制御レジスタの設定	CCR0	16 ビットモード許可設定 >>	.M16E	カウント方向転換 >>	.CDCF	カウント方向転換割込み許可 >>	.CFIE	内蔵プリスケラ選択 >>	.CLKS	カウントモード選択 >>	.CMS1-0	カウントクロックエッジ選択 >>	.CES1-0	アップダウンモード選択 >>	.UDMS	カウンタライト >>	.CTUT	UDCR クリア許可 >>	.UCRE	リロード機能許可 >>	.RLDE	UDCR クリア >>	.UDCC	カウンタクリア / ゲート選択 >>	.CGSC	カウンタクリア / ゲートエッジ選択 >>	.CGE1-0	リロード値設定	RCR0		RCR1	UD カウンタ 0 割込みレベルの設定	ICR07	I フラグの設定	(CCR)	アンダフロー割込み制御	CSR0.UDIE	RCR から UDCR ヘデータ転送	CCR0.CTUT	カウント動作起動	CSR0.CSTR	アンダフロー検出フラグのチェック		割込み要求フラグクリア	CSR0.UDFF	(任意の処理)		<p>プログラム例</p> <pre> void UD0_sample_1(void) { UD0_initial(); UD0_start(); } void UD0_initial(void) { IO_CCR0.word = 0x9018; /* 設定値 = 1001_0000_0001_1000 */ /* bit15 = 1 M16E 16bit × 1ch 動作モード */ /* bit14 = 0 CDCF カウント方向転換フラグクリア */ /* bit13 = 0 CFIE 方向転換割込み禁止 */ /* bit12 = 1 CLKS 8 マシンサイクル */ /* bit11-10 = 00 CMS1, 0 タイマモード */ /* bit9-8 = 00 CES1, 0 エッジ検出禁止 */ /* bit7 = 0 UMDS アップダウンモード選択 */ /* bit6 = 0 CTUT カウンタライト (無効) */ /* bit5 = 0 UCRE カウンタクリア禁止 */ /* bit4 = 1 RLDE リロード機能許可 */ /* bit3 = 1 UDCC UDCR クリア (無効) */ /* bit2 = 0 CGSC ZIN カウンタクリア機能 */ /* bit1-0 = 00 CGE1, 0 ZIN エッジ検出禁止 */ IO_RCR0 = 0xff; /* 16 ビットモード リロード値設定 (値は任意) */ IO_RCR1 = 0xff; IO_ICR07.byte = 0x10; /* 割込みレベル設定 (値は任意) */ __EI(); /* 割込み許可 */ } void UD0_start(void) { IO_CSR0.bit.UDIE = 1; /* bit5 = 1 UDIE アンダフロー割込み許可 */ IO_CCR0.bit.CTUT = 1; /* bit6 = 1 CTUT カウンタライト */ IO_CSR0.bit.CSTR = 1; /* bit7 = 1 CSTR カウント動作起動 */ } __interrupt void UD0_int(void) { if(IO_CSR0.bit.UDFF) { IO_CSR0.bit.UDFF = 0; /* bit2 = 0 UDFF アンダフロー検出フラグクリア */ } } #pragma intvect UD0_int 25 </pre> <p>(注意事項) レジスタの記述形式については、「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
制御レジスタの設定	CCR0																																																
16 ビットモード許可設定 >>	.M16E																																																
カウント方向転換 >>	.CDCF																																																
カウント方向転換割込み許可 >>	.CFIE																																																
内蔵プリスケラ選択 >>	.CLKS																																																
カウントモード選択 >>	.CMS1-0																																																
カウントクロックエッジ選択 >>	.CES1-0																																																
アップダウンモード選択 >>	.UDMS																																																
カウンタライト >>	.CTUT																																																
UDCR クリア許可 >>	.UCRE																																																
リロード機能許可 >>	.RLDE																																																
UDCR クリア >>	.UDCC																																																
カウンタクリア / ゲート選択 >>	.CGSC																																																
カウンタクリア / ゲートエッジ選択 >>	.CGE1-0																																																
リロード値設定	RCR0																																																
	RCR1																																																
UD カウンタ 0 割込みレベルの設定	ICR07																																																
I フラグの設定	(CCR)																																																
アンダフロー割込み制御	CSR0.UDIE																																																
RCR から UDCR ヘデータ転送	CCR0.CTUT																																																
カウント動作起動	CSR0.CSTR																																																
アンダフロー検出フラグのチェック																																																	
割込み要求フラグクリア	CSR0.UDFF																																																
(任意の処理)																																																	

■ プログラム例以外の設定方法

● 動作を 8 ビットか 16 ビットか選択する方法

16 ビットモード許可設定ビット (CCR0.M16E) で設定します。

アップダウンカウンタのビット長	16 ビットモード許可設定ビット (M16E)
8 ビットに設定するには	"0" にする
16 ビットに設定するには	"1" にする

● カウントモードの種類と設定方法

4 種のカウントモードがあります。

カウントモードは、カウントモード選択ビット (CCR0.CMS[1:0], CCR1.CMS[1:0]) で設定します。

カウントモード	カウントモード選択ビット (CMS[1:0])
タイマモードにするには	"00" にする
アップダウンカウントモードにするには	"01" にする
位相差カウントモード(2 通倍)にするには	"10" にする
位相差カウントモード(4 通倍)にするには	"11" にする

● タイマモード動作時のカウントソースを選択する方法

内部プリスケアラ選択ビット (CCR0.CLKS, CCR1.CLKS) で設定します。

タイマモード動作時のカウントソース	内部プリスケアラ選択ビット (CLKS)
ϕ を 2 分周したクロックにするには	"0" にする
ϕ を 8 分周したクロックにするには	"1" にする

● アップダウンカウンタ動作時に入力する信号 (AIN, BIN) を検出するときのエッジを選択する方法

カウントクロックエッジ選択ビット (CCR0.CES[1:0], CCR1.CES[1:0]) で設定します。

カウント件検出エッジ	カウントクロックエッジ選択ビット (CES[1:0])
検出禁止を設定するには	"00" にする
立下り検出を設定するには	"01" にする
立下り検出を設定するには	"10" にする
両エッジ検出を設定するには	"11" にする

● アップダウンカウンタに値を設定する方法

リロード / コンペアレジスタ (RCR) に値を書いた後で, カウンタライトビット (CCR0.CTUT, CCR1.CTUT) に "1" を書き込むことでアップダウンカウンタに値を設定できます。

● アップダウンカウンタのアップカウント値とコンペア値(RCR[0:1])が一致し, さらにアップカウントした時にアップダウンカウンタのクリアを許可する方法

アップダウンカウンタクリア許可ビット (CCR0.UCRE, CCR1.UCRE) で設定します。

動作	アップダウンカウンタクリア許可ビット (UCRE)
アップダウンカウンタのクリアを禁止するには	"0" にする
アップダウンカウンタのクリアを許可するには	"1" にする

● アップダウンカウンタのアンダフロー発生時に, リロード値(RCR[1:0])をアップダウンカウンタにリロードすることを許可する方法

リロードイネーブルビット (CCR0.RLDE, CCR1.RLDE) で設定します。

動作	リロードイネーブルビット (RLDE)
アップダウンカウンタにリロード値 (RCR) をリロードすることを禁止するには	"0" にする
アップダウンカウンタにリロード値 (RCR) をリロードすることを許可するには	"1" にする

● アップダウンカウンタをクリアする方法

アップダウンカウンタは以下の 5 つの方法でクリアすることができます。

- アップダウンカウンタクリアビット (CCR0.UDCC, CCR1.UDCC) への "0" 書込み
- ZIN 端子へのエッジ入力
- コンペア値とアップダウンカウンタのアップカウント値が一致したとき
- フルカウントからのカウントアップ動作
- リセット入力 (外部リセット, ウォッチドッグリセット, ソフトウェアリセット)

● ZIN 端子でアップダウンカウンタをクリアする方法

カウンタクリアゲートビット (CCR0.CGSC, CCR1.CGSC) とカウンタクリアゲートエッジ選択ビット (CCR0.CGE[1:0], CCR1.CGE[1:0]) で設定します (すべてのカウントモードで有効)。

ZIN 端子入力に対して	カウンタクリアゲートビット (CGSC)	カウンタクリアゲートエッジ選択ビット (CGE[1:0])
エッジ検出禁止 (クリアなし) にさせるには	"0" にする	"00" にする
立下りエッジでアップダウンカウンタをクリアするには	"0" にする	"01" にする
立上りエッジでアップダウンカウンタをクリアするには	"0" にする	"10" にする

GCE[1:0]=11_B は設定禁止です。

● アップダウンのカウント動作を ZIN 端子で制御する方法

カウンタクリアゲートビット (CCR0.CGSC, CCR1.CGSC) とカウンタクリアゲートエッジ選択ビット (CCR0.CGE[1:0], CCR1.CGE[1:0]) で設定します (すべてのカウントモードで有効)。

ZIN 端子入力に対して	カウンタクリアゲートビット (CGSC)	カウンタクリアゲートエッジ選択ビット (CGE[1:0])
レベル検出禁止 (カウント禁止状態) にさせるには	"1" にする	"00" にする
"L" レベルでアップダウンカウント動作 "H" レベルでアップダウンカウント停止させるには	"1" にする	"01" にする
"L" レベルでアップダウンカウント停止 "H" レベルでアップダウンカウント動作させるには	"1" にする	"10" にする

GCE[1:0]=11_B は設定禁止です。

● アップダウンカウンタのカウンタ動作を許可 / 禁止する方法

カウンタ起動ビット (CSR0.CSTR, CSR1.CSTR) で設定します。

動作	カウンタ起動ビット (CSR0.CSTR, CSR1.CSTR)
アップダウンカウンタのカウンタ動作を禁止するには	"0" にする
アップダウンカウンタのカウンタ動作を許可するには (起動するには)	"1" にする

カウンタの開始は、カウンタモードにより異なります。

タイマモード → 内部クロックでカウンタ開始

アップダウンカウンタモード → AIN 端子, BIN 端子のエッジを検出してカウン
タ開始

位相差カウンタモード → AIN 端子, BIN 端子の位相差を検出してカウン
タ開始

ただし, ZIN端子のゲート機能を選択時はカウンタ動作許可レベルを検出している必要
があります。

● 直前のカウンタ方向を知る方法 (現在の回転方向を知る方法)

アップダウンフラグ (CSR0.UDF[1:0], CSR1.UDF[1:0]) で設定します。

設定	アップダウンフラグ (UDF[1:0])
"00 _B " のとき	リセット後カウンタなし
"01 _B " のとき	ダウンカウンタ
"10 _B " のとき	アップカウンタ
"11 _B " のとき	アップとダウンが同時発生 (アップもダウンもカウンタしていません)

このフラグは割込みとは関係ありませんので、割込み処理する場合はカウンタ方向転
換フラグ (CCR0.CDCF, CCR1.CDCF) を利用ください。

● 方向転換を知る方法

カウンタ方向転換フラグ (CCR0.CDCF, CCR1.CDCF) で設定します。

設定	カウンタ方向転換フラグ (CDCF)
"0" のとき	フラグクリア後方向転換は行われていない
"1" のとき	フラグクリア後方向転換が行われている (1 回以上)

● 比較結果一致が発生したことを知る方法

コンペア検出フラグ (CSR0.CMPF, CSR1.CMPF) で設定します。

設定	コンペア検出フラグ (CMPF)
"0" のとき	アップダウンカウンタのカウント値とコンペア値が一致していない
"1" のとき	アップダウンカウンタのカウント値とコンペア値が一致している

- 動作がアップ/ダウン/値の設定/値のリロードなどに関係なく比較結果一致したらフラグが "1" になります。

● オーバフロー, アンダフローが発生したことを知る方法

オーバフロー検出フラグ (CSR0.OVFF, CSR1.OVFF), アンダフロー検出フラグ (CSR0.UDFF, CSR1.UDFF) で設定します。

(OVFF)=1 のとき	アップダウンカウンタがオーバフローを発生した
(UDFF)=1 のとき	アップダウンカウンタがアンダフローを発生した

● リロード値, コンペア値を設定する方法

リロード/コンペアレジスタ RCR0, RCR1 に値を設定します (コンペア値とリロード値は同じ値となります)。

● 割込み関連レジスタ

アップダウンカウンタ番号, 割込みレベル, ベクタの関係は下表のとおりです。
割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

要因	割込みベクタ	割込みレベル設定レジスタ
アップダウンカウンタ	#25 アドレス: FFFF98 _H	割込みレベルレジスタ (ICR07) アドレス: 0000B7 _H

割込み要求フラグ

カウント方向転換	: (CCR0.CDCF), (CCR1.CDCF)
コンペア検出	: (CSR0.CMPF), (CSR1.CMPF)
オーバフロー	: (CSR0.OVFF), (CSR1.OVFF)
アンダフロー	: (CSR0.UDFF), (CSR1.UDFF)

上記割込み要求フラグは自動的にクリアしません。割込み処理から復帰する前にソフトウェアにて割込み要求フラグに "0" を書き込んでください。

● 割込みの種類と選択方法

割込み要因には、以下の 3 つがあります。

カウント方向転換, 比較結果一致, オーバ / アンダフロー

割込みは上記 3 つの割込み要因の OR で発生します。割込み要因の選択は割込み要求許可ビットで設定します。

● 割込みを許可 (選択) / 禁止 / クリアする方法

割込みの許可 (選択) / 禁止は、割込み要求許可ビットの

カウント方向転換割込み要求許可ビット : (CCR0.CFIE), (CCR1.CFIE)

コンペア割込み要求許可ビット : (CSR0.CITE), (CSR1.CITE)

オーバフロー / アンダフロー割込み要求許可ビット : (CSR0.UDIE), (CSR1.UDIE)

で設定します。

制御	割込み要求許可ビット (CFIE, CITE, UDIE)
割込み要求を禁止するには	"0" にする
割込み要求を許可するには	"1" にする

割込み要求のクリアは、割込み要求ビットの

カウント方向転換 : (CCR0.CDCF), (CCR1.CDCF)

コンペア検出 : (CSR0.CMPF), (CSR1.CMPF)

オーバフロー : (CSR0.OVFF), (CSR1.OVFF)

アンダフロー : (CSR0.UDFF), (CSR1.UDFF)

で設定します。

制御	割込要求ビット (CDCF, CMPF, OVFF, UDFF)
割込み要求をクリアするには	"0" を書き込む

第14章

16 ビットリロードタイマ

16 ビットリロードタイマの概要，レジスタの構成と機能，割込みおよび動作について説明します。

14.1 16 ビットリロードタイマの概要

14.2 16 ビットリロードタイマのレジスタの構成と機能

14.3 16 ビットリロードタイマの割込み

14.4 16 ビットリロードタイマの動作

14.5 16 ビットリロードタイマのプログラム例

14.1 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は、以下の機能があります。

- クロックモードは、内部クロックモードおよびイベントカウントモードから選択できます。
- カウント動作は、16 ビットタイマレジスタ (TMR) のアンダフローが発生した場合は、カウント動作を停止するワンショットモードまたはカウント設定値をリロードしてカウント動作を継続するリロードモードから選択できます。
- 16 ビットタイマレジスタ (TMR) のアンダフローが発生した場合は、割込みを発生させることにより、インターバルタイマとして利用できます。

■ 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下記に示します。

クロックモード	カウント動作	16 ビットリロードタイマ動作
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
	ワンショットモード	外部トリガ動作 外部ゲート入力動作
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

カウントクロックとして 3 種類の内部クロックから 1 種類が選択できます。

● ソフトウェアトリガ動作

タイマ制御ステータスレジスタ (TMCSR) の TRG ビットに "1" を書き込むと、カウンタ動作を開始します。TRG ビットによるトリガ入力は、外部トリガ動作時および外部ゲート入力動作時にも有効です。

● 外部トリガ動作

選択されたエッジ (立上り / 立下り / 両エッジ) が TIN0 端子に入力されたとき、カウント動作を開始します。

● 外部ゲート入力動作

選択された信号 ("L", または "H") が TIN0 に入力されている間、カウント動作を続行します。

■ イベントカウントモード (外部クロックモード)

選択されたエッジ (立上り / 立下り / 両エッジ) が TIN0 端子に入力されると、そのエッジでカウントダウンする機能です。

一定周期の外部クロックを使用した場合は、インターバルタイマとしても使用できます。

■ カウンタ動作のモード

● リロードモード

カウントダウンで、アンダフロー ("0000_H" → "FFFF_H") が発生すると、カウント設定値がリロードされてカウント動作が継続されます。アンダフローで、割込み要求を発生させられるため、インターバルタイマとして使用できます。また、アンダフローごとに、反転するトグル波形を TOT0 端子から出力することもできます。

カウントクロック	カウントクロック周期	インターバル時間
内部クロック	$\phi/2^1(80\text{ns})$	80ns ~ 5.243ms
	$\phi/2^3(0.32\mu\text{s})$	0.32 μs ~ 20.972ms
	$\phi/2^5(1.28\mu\text{s})$	1.28 μs ~ 83.886ms
外部クロック	$\phi/2^3(0.32\mu\text{s})$	0.32 μs 以上

(注意事項) ・ マシンサイクル (ϕ) = 25MHz 例) $25\text{MHz}/2^1 = 12.5\text{MHz} = 80\text{ns}$

・ インターバル時間の最大値は 0000 ~ FFFF

例) $\phi/2^1(80\text{ns}) \times 65536 \div 5.243\text{ms}$

● ワンショットモード

カウントダウンで、アンダフロー ("0000_H" → "FFFF_H") が発生すると、カウント動作を停止します。

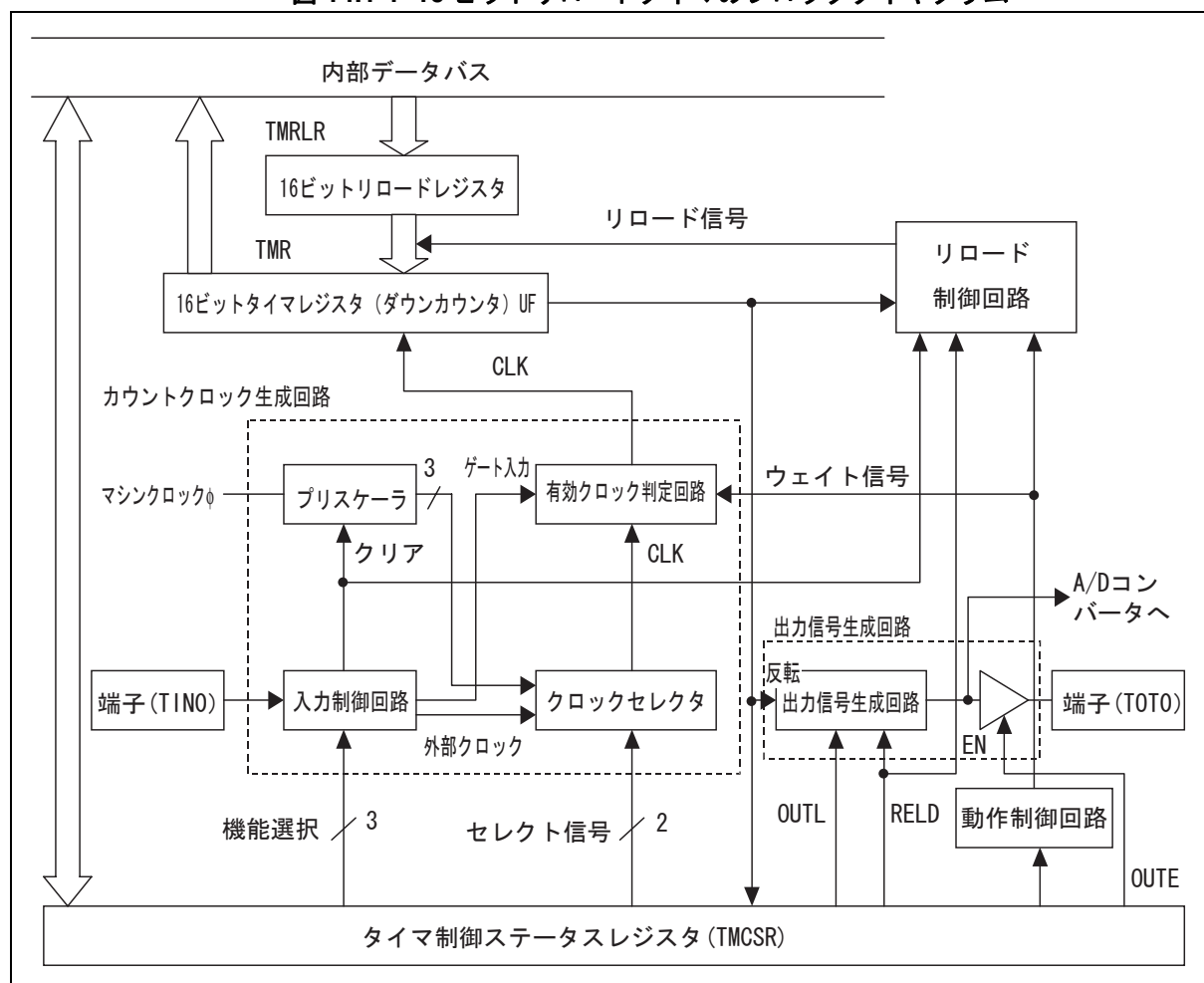
アンダフローで割込みを発生します。また、カウントの動作中は、カウント中を示す短波形を TOT0 端子から出力できます。

< 参考 >

16 ビットリロードタイマは、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ブロックダイアグラム

図 14.1-1 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に関する端子

16 ビットリロードタイマに関する端子は、TIN0 端子と TOT0 端子があります。TIN0 端子は、汎用入出力ポート (P73/TIN0) と、16 ビットリロードタイマの入力端子として機能を兼用しており、TOT0 端子は、汎用入出力ポート (P74/TOT0) と、16 ビットリロードタイマの出力端子として機能を兼用しています。

● TIN0 端子として使用する場合の設定

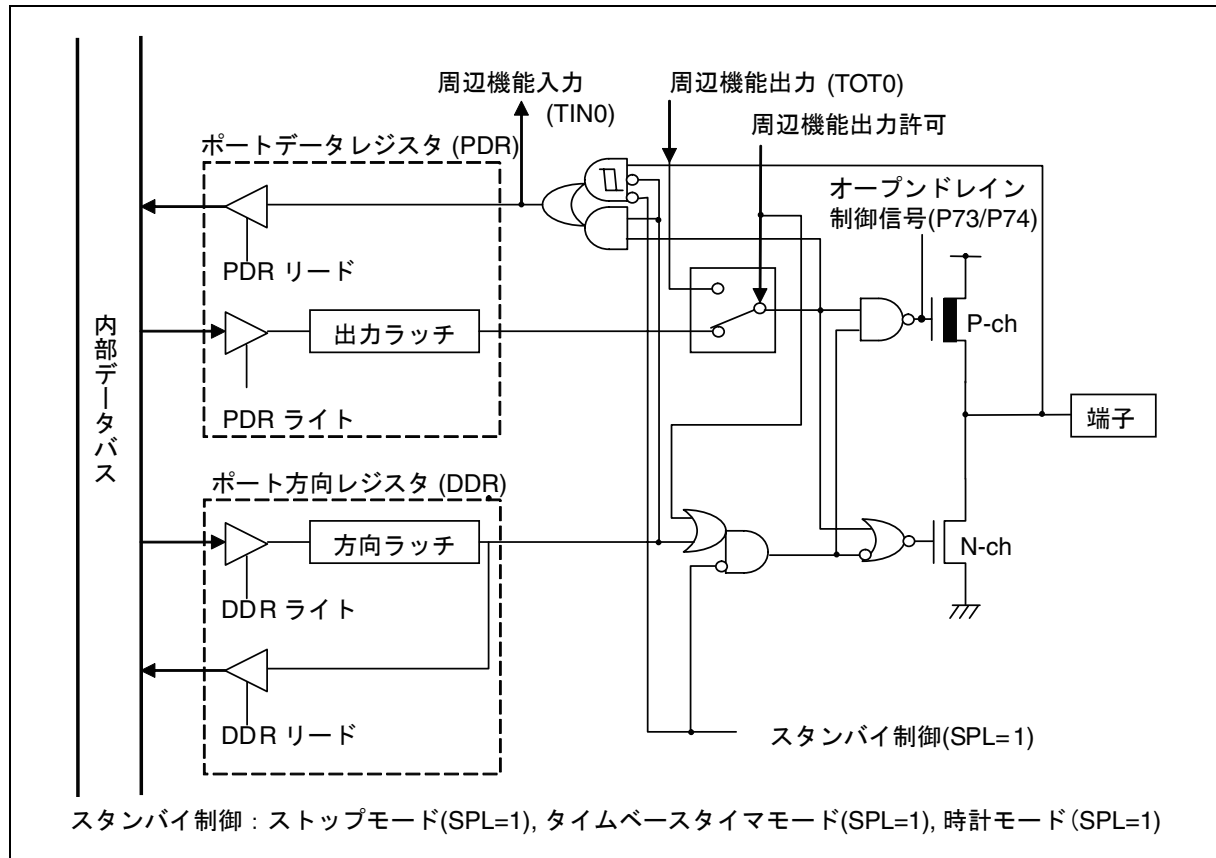
TIN0 端子を入力として使用する場合は、P73/TIN0 端子は、ポート方向レジスタにより入力ポート (DDR7 bit12 → "0") に設定してください。

● TOT0 端子として使用する場合の設定

TOT0 端子を出力として使用する場合は、タイマ制御ステータスレジスタ (TMCSR) を出力許可 (OUTE bit6 → "1") に設定してください。

■ 16 ビットリロードタイマに関する端子のブロックダイアグラム

図 14.1-2 16 ビットリロードタイマに関する端子のブロックダイアグラム



14.2 16 ビットリロードタイマのレジスタの構成と機能

16 ビットリロードタイマのレジスタの構成と機能について説明します。

■ 16 ビットリロードタイマのレジスタ構成

図 14.2-1 に、16 ビットリロードタイマのレジスタ一覧を示します。

図 14.2-1 16 ビットリロードタイマのレジスタ一覧

0000CB _H	15	14	13	12	11	10	9	8	TMCSR タイマ制御ステータスレジスタ (上位) リード/ライト 初期値
	—	—	—	—	CSL1	CSL0	MOD2	MOD1	
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(0)	初期値
0000CA _H	7	6	5	4	3	2	1	0	TMCSR タイマ制御ステータスレジスタ (下位) リード/ライト 初期値
	MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	初期値
0000CD _H	15	14	13	12	11	10	9	8	TMR/TMRLR 16ビットタイマレジスタ/ 16ビットリロードレジスタ (上位) リード/ライト 初期値
	D15	D14	D13	D12	D11	D10	D09	D08	
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値
0000CC _H	7	6	5	4	3	2	1	0	TMR/TMRLR 16ビットタイマレジスタ/ 16ビットリロードレジスタ (下位) リード/ライト 初期値
	D07	D06	D05	D04	D03	D02	D01	D00	
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値

14.2.1 タイマ制御ステータスレジスタ (TMCSR)

タイマ制御ステータスレジスタ (TMCSR) の構成と機能について説明します。

■ タイマ制御ステータスレジスタ (TMCSR)

タイマ制御ステータスレジスタ (TMCSR) は、16 ビットリロードタイマの動作モードおよび割込みを制御します。CNTE=0 のとき、UF/CNTE/TRG 以外のビットをモディファイします。

下図に、タイマ制御ステータスレジスタ (TMCSR) のビット構成を示します。

0000CB _H	15	14	13	12	11	10	9	8	TMCSR
	—	—	—	—	CSL1	CSL0	MOD2	MOD1	タイマ制御ステータスレジスタ (上位)
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(0)	初期値
0000CA _H	7	6	5	4	3	2	1	0	TMCSR
	MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG	タイマ制御ステータスレジスタ (下位)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	初期値

以下に、タイマ制御ステータスレジスタ (TMCSR) の各ビットの機能を説明します。

[bit11, bit10] CSL1, CSL0(クロック選択)

これらのビットは、カウントクロック選択ビットで、クロックソースを選択します。

CSL1	CSL0	クロックソース (マシンクロック $\phi=16\text{MHz}$ 時)
0	0	$\phi/2^1(0.125\mu\text{s})$ (初期値)
0	1	$\phi/2^3(0.5\mu\text{s})$
1	0	$\phi/2^5(2.0\mu\text{s})$
1	1	イベントカウントモード

[bit9, bit8, bit7] MOD2, MOD1, MOD0

これらのビットは、動作モードと入出力端子の機能を設定します。MOD2=0 のとき、入力端子はトリガとして機能します。アクティブエッジが入力端子に入力されカウント動作が進行すると、リロードレジスタの内容がカウンタにロードされます。MOD2=1 のときは、タイマはゲートカウンタモードで動作し、入力端子はゲート入力として機能します。このモードでは、カウンタはアクティブレベルが入力端子に入力されている間だけ、カウントします。

MOD2～MOD0 ビットの組合せによって、内部クロックモードおよびイベントカウントモードを表 14.2-1 と表 14.2-2 に示すモードから選択できます。

表 14.2-1 内部クロックモード (CLS1, 0=00, 01 または 10)

MOD2	MOD1	MOD0	入力端子機能	アクティブエッジ, またはレベル	初期値
0	0	0	トリガ無効	-	
0	0	1	トリガ入力	立上りエッジ	
0	1	0		立下りエッジ	
0	1	1		両エッジ	
1	X	0	ゲート入力	"L" レベル	
1	X	1		"H" レベル	

表 14.2-2 イベントカウントモード (CLS1, 0=11)

MOD2	MOD1	MOD0	入力端子機能	アクティブエッジ, またはレベル	初期値
X	0	0	トリガ無効	-	
X	0	1	トリガ入力	立上りエッジ	
X	1	0		立下りエッジ	
X	1	1		両エッジ	

[bit6] OUTE(出力許可)

このビットは、出力許可を制御します。

TOT 端子は、このビットが "0" のとき汎用ポートとして、"1" のときタイマ出力端子として機能します。リロードモードでは、出力波形はトグル波形となります。ワンショットモードでは、TOT 端子は、カウントが進行中であることを示す短形波を出力します。

OUTE	機能
0	汎用ポート (初期値)
1	タイマ出力

[bit5] OUTL(出力レベルの設定)

このビットは、TOT 端子の出力レベルを設定するビットです。OUTL が、"0" のときと "1" のときでは出力端子レベルは逆になります。

OUTL	ワンショットモード時 (RELD=0)	リロードモード時 (RELD=1)	初期値
0	カウント中 "H" の短形波	0	
1	カウント中 "L" の短形波	1	
X	1	0	
X	1	1	

[bit4] RELD(リロード動作許可)

このビットは、リロード動作を許可します。RELD が "1" のときは、タイマはリロードモード動作で動作します。このモードでは、タイマは、リロードレジスタの内容をカウンタにロードし、アンダフローが発生してもカウント動作を継続します。RELD が "0" のときは、タイマはワンショットモードで動作します。このモードでは、アンダフローが発生するとカウンタ動作を停止します。

RELD	機能
0	ワンショットモード (初期値)
1	リロードモード

[bit3] INTE(タイマ割込み要求許可)

このビットは、タイマ割込み要求を許可するビットです。INTE=0 のときは、UF が "1" になっても、割込み要求が発生しません。

INTE	機能
0	割込み要求出力禁止 (初期値)
1	割込み要求出力許可

<注意事項>

リロードタイマアンダフロー割込みを許可 (TMCSR レジスタの INTE ビット = 1) から禁止 (TMCSR レジスタの INTE ビット = 0) にする場合は、割込み制御レジスタ (ICR12) の IL2 ~ IL0 ビット = "111" として割込みを禁止にしてから INTE ビットに "0" を書き込んでください。

[bit2] UF(タイマ割込み要求フラグ)

このビットは、タイマ割込み要求フラグです。アンダフローが発生すると UF は "1" に設定されます。"0" を書き込むか、または μ DMAC によってクリアされます。本ビットに "1" を書き込んでも意味をもちません。リードモディファイライト命令での読出し値は "1" です。

UF	読出し時	書込み時
0	カウンタのアンダフローなし (初期値)	このビットのクリア (初期値)
1	カウンタのアンダフローあり	変化なし (ほかへの影響なし)

<注意事項>

16 ビットリロードタイマの割込みルーチン内で、ILM (インタラプトレベルマスクレジスタ) 値を変更する場合には、UF ビット (タイマ割込み要求フラグ) を "0" にクリアした後に行ってください。

[bit1] CNTE(タイマカウンタ許可)

このビットは、タイマカウンタを許可します。

CNTE	機能
0	カウンタ停止 (初期値)
1	カウンタ許可 (起動トリガ待ち)

[bit0] TRG(ソフトウェアトリガ)

このビットは、ソフトウェアトリガビットです。TRG=1 の書込みにより、ソフトウェアトリガが適用されタイマのリロードレジスタの内容をカウンタにロードし、カウントが開始されます。"0" を書き込んでも意味をもちません。読出しは常に "0" です。CNTE=1 のときのみ、動作モードにかかわらず常に有効になります。

TRG	機能
0	変化なし (初期値)
1	カウント動作開始

14.2.2 16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR)

16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR) の構成と機能について説明します。

■ 16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR)

下図に、16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR) のビット構成を示します。

	15	14	13	12	11	10	9	8	
0000CD _H	D15	D14	D13	D12	D11	D10	D09	D08	TMR/TMRLR (上位)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値
	7	6	5	4	3	2	1	0	
0000CC _H	D07	D06	D05	D04	D03	D02	D01	D00	TMR/TMRLR (下位)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値

■ 16 ビットタイマレジスタ (TMR)

16 ビットタイマレジスタは、16 ビットダウンカウンタのカウント値を読み出します。カウンタ動作を許可 (TMCSR の CNTE=1) し、カウントを開始すると、16 ビットリロードレジスタに書き込まれた値がこのレジスタにロードされ、カウントダウンが開始されます。カウント停止状態 (TMCSR の CNTE=0) では、このレジスタの値が保持されます。

<注意事項>

このレジスタの読出しはカウント動作中にできますが、必ずワード転送命令 (MOVW A 003AH など) を使用してください。

16 ビットタイマレジスタ (TMR) は、機能的には読出し専用ですが、書込み専用の 16 ビットリロードレジスタ (TMRLR) と同一アドレスに配置されています。したがって、書込みをしても TMR の値は影響を受けませんが、TMRLR に対して書込みが行われます。

■ 16 ビットリロードレジスタ (TMRLR)

16 ビットリロードレジスタは、カウント動作を禁止 (TMCSR の CNTE=0) している状態で、カウンタの初期値を設定します。カウンタ動作を許可 (TMCSR の CNTE=1) し、カウンタを起動すると、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。このレジスタに設定した値は、リロードモードのとき、アンダフローの発生でカウンタにリロードされ、カウントダウンを続行します。ワンショットモードのときは、アンダフローの発生後、カウンタは FFFF_H で停止します。

<注意事項>

このレジスタへの書込みは、カウンタ停止 (TMCSR の CNTE=0) 状態で行ってください。また、必ずワード転送命令 (MOVW A 003AH など) で書き込んでください。

16 ビットリロードレジスタ (TMRLR) は、機能的には書込み専用ですが、読出し専用の 16 ビットタイマレジスタ (TMR) と同一アドレスに配置されています。したがって、読出し値は TMR の値となるため、INC/DEC 命令などのリードモディファイライト動作をする命令は、使用できません。

14.3 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は、カウンタのアンダフロー検出した場合に発生します。

カウンタのアンダフロー割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができます。

■ 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 14.3-1 に示します。

表 14.3-1 16 ビットリロードタイマの割込み

リロードタイマ	アンダフロー割込み
割込み要求フラグ	TMCSR: UF (bit2)
割込み要求出力許可ビット	TMCSR: INTE (bit3)
割込み発生要因	16 ビットリロードタイマのアンダフロー

16 ビットタイマレジスタ (TMR) のカウント中に TMR 値が "0000" から "FFFF" にカウントダウンされると、アンダフローが発生します。アンダフローが発生すると、タイマ制御ステータスレジスタ (TMCSR) のアンダフロー発生フラグ (UF=1) が設定されます。アンダフロー割込みが許可されている場合は (INTE=1), 割込み要求が発生します。

■ 16 ビットリロードタイマの割込みと DMA 転送および EI²OS

表 14.3-2 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 14.3-2 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
16 ビットフリーランタイムオーバーフロー、 16 ビットリロードタイマアンダフロー	○	12	#35	FFFF70 _H	ICR12	0000BC _H

○：割込み要求フラグはクリアされます。

*：この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。

詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に2つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

16 ビットリロードタイマは DMA 転送機能、および EI²OS 機能に対応しています。

DMA または EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

14.4 16 ビットリロードタイマの動作

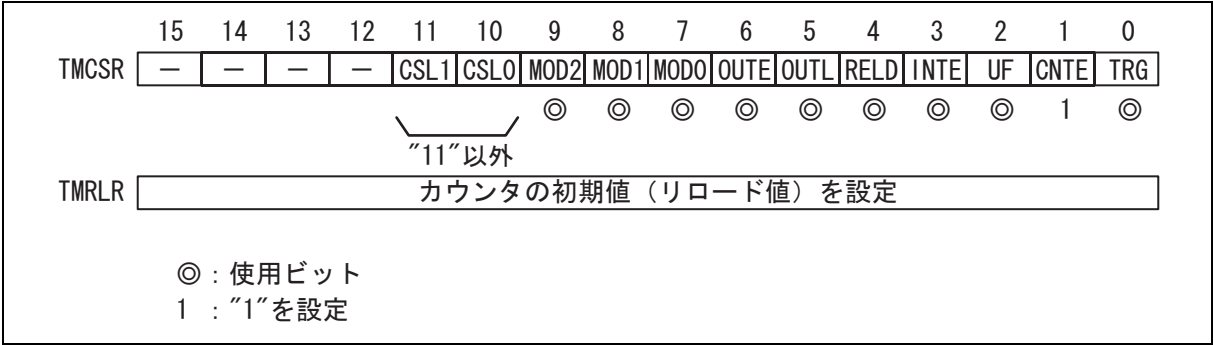
16 ビットリロードタイマの設定について説明します。

■ 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、図 14.4-1 に示す設定が必要です。

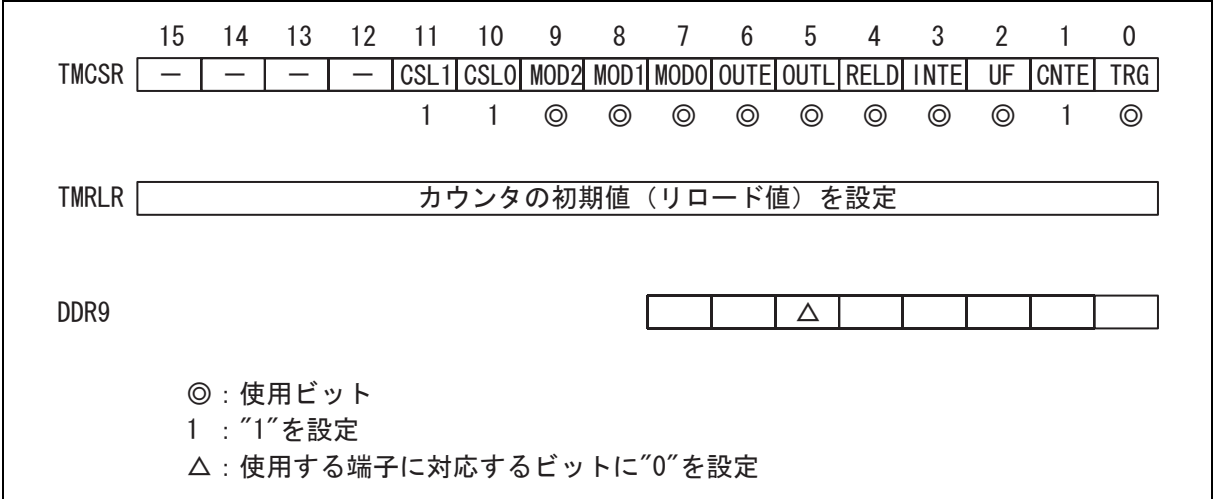
図 14.4-1 内部クロックモードの設定



● イベントカウントモードの設定

イベントカウントモードとして動作させるには、図 14.4-2 に示す設定が必要です。

図 14.4-2 イベントカウントモードの設定

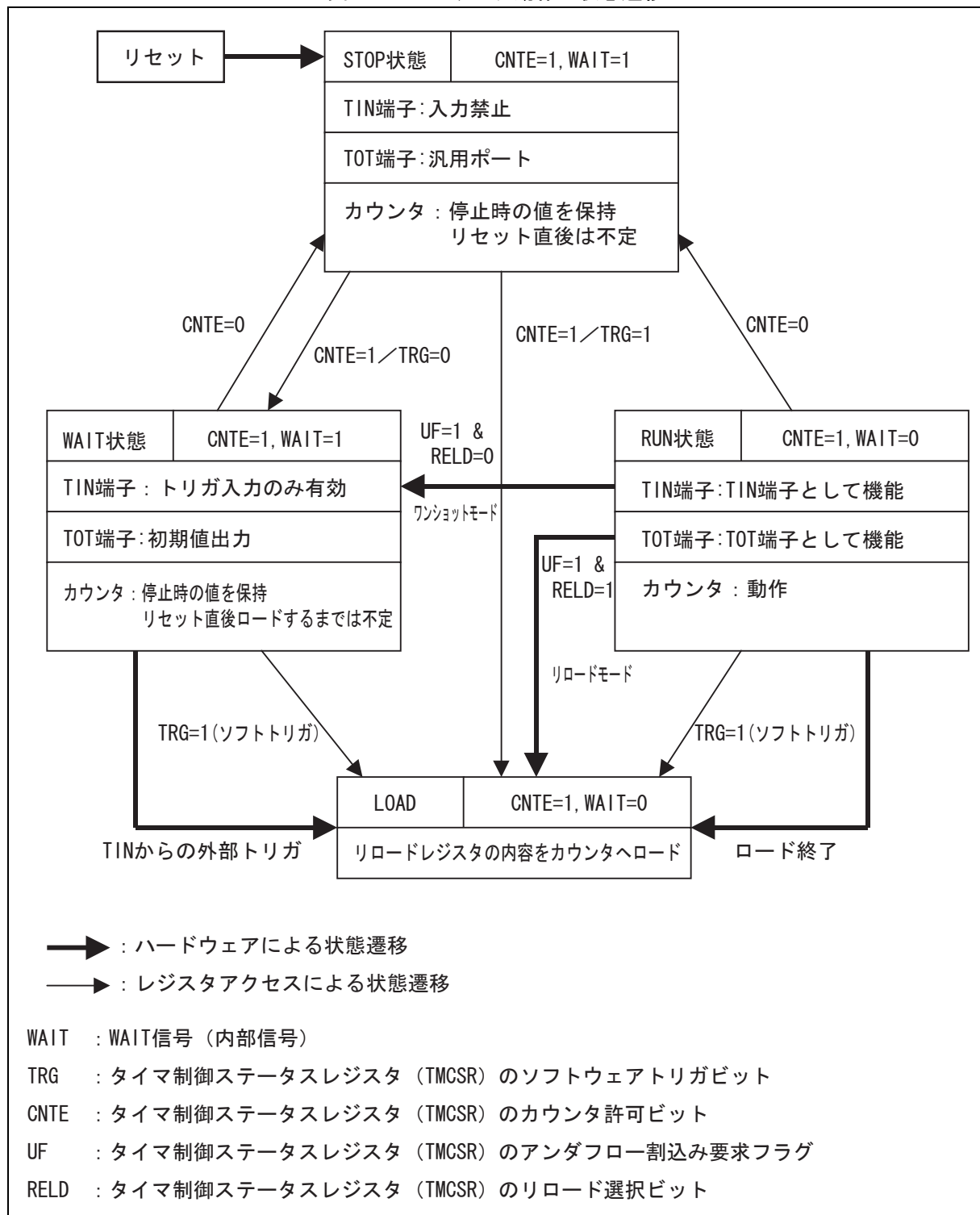


14.4.1 カウント動作の状態遷移

カウント動作の状態遷移を示します。

■ カウント動作の状態遷移

図 14.4-3 カウンタ動作の状態遷移



14.4.2 内部クロックモード (リロードモード) の動作

内部カウントクロックに同期して、16 ビットカウンタをカウントダウンし、カウンタのアンダフローで CPU に割込み要求を発生します。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

カウント動作を許可 (TMCSR の CNTE=1) し、ソフトウェアトリガビット (TMCSR の TRG) または外部トリガでタイマを起動すると、16 ビットリロードレジスタ (TMRLR) の値をカウンタにロードして、カウンタ動作を開始します。

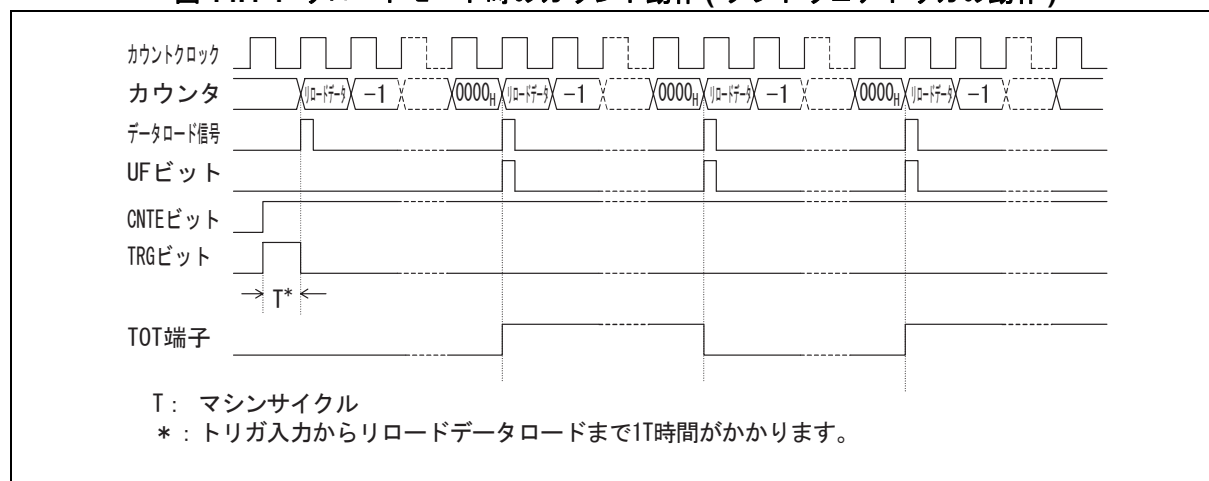
カウンタ許可ビットとソフトウェアトリガビットを同時に "1" にセットすると、カウンタ許可と同時にカウントを開始します。カウンタ値がアンダフロー ("0000_H" → "FFFF_H") すると、16 ビットリロードレジスタ (TMRLR) の値をカウンタにロードして、カウント動作を継続します。このとき、アンダフロー割込み要求フラグビット (UF) が "1" にセットされ、割込み要求許可ビット (INTE) が "1" である場合は割込み要求を発生します。また、TOT 端子からは、アンダフローごとに反転するトグル波形を出力します。

● ソフトウェアトリガの動作

タイマ制御ステータスレジスタ (TMCSR) の TRG ビットに "1" を書き込むと、カウンタが起動されます。

図 14.4-4 にリロード時のソフトウェアトリガの動作を示します。

図 14.4-4 リロードモード時のカウント動作 (ソフトウェアトリガの動作)

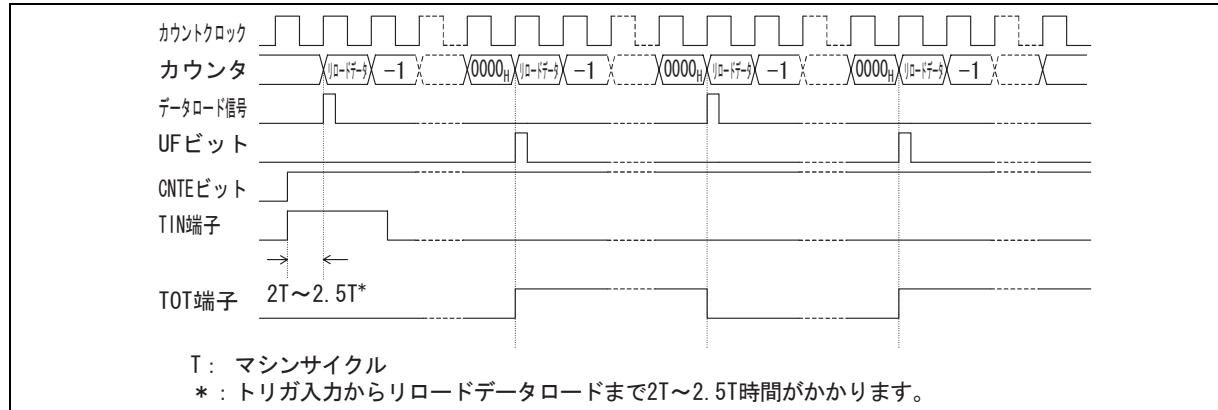


● 外部トリガの動作

有効エッジ (立上り , 立下り , 両エッジ選択可) が TIN 端子に入力されると , カウンタが起動されます。

図 14.4-5 にリロードモード時の外部トリガの動作を示します。

図 14.4-5 リロードモード時のカウント動作 (外部トリガの動作)



<注意事項>

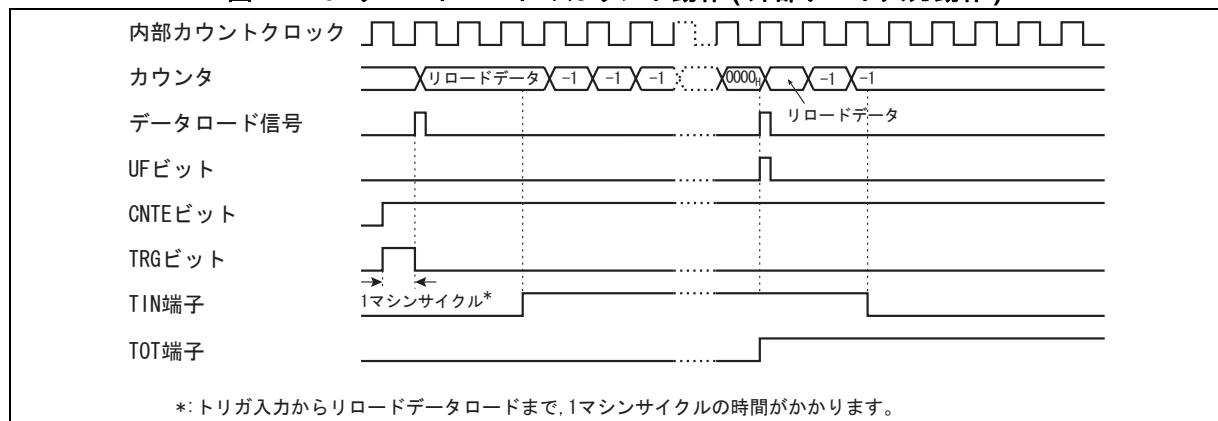
TIN 端子へ入力するトリガパルス幅は , $2T$ (T: マシンサイクル) 以上としてください。

● 外部ゲート入力の動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に "1" を設定している場合 , ソフトウェアトリガビット (TRG) に "1" を設定すると , カウント動作を開始します。

動作モード設定ビット (MOD2, MOD1, MOD0) で設定されているゲート入力の有効レベル ("L", "H" から設定可能) が TIN 端子に入力されている間 , カウント動作を行います。

図 14.4-6 リロードモードのカウント動作 (外部ゲート入力動作)



<注意事項>

TIN 端子へ入力するトリガパルス幅は , $2T$ (T: マシンサイクル) 以上としてください。

14.4.3 内部クロックモード (ワンショットモード) の動作

内部カウントクロックに同期して, 16 ビットカウンタをカウントダウンし, カウンタのアンダフローで CPU に割込み要求を発生します。また, TOT 端子から, カウンタ中を示す矩形波を出力します。

■ 内部クロックモード (ワンショットモード) の動作

カウント動作を許可 (TMCSR の CNTE=1) し, ソフトウェアトリガビット (TMCSR の TRG) または外部トリガでタイマを起動すると, カウンタ動作を開始します。

カウンタ許可ビットとソフトウェアトリガビットを同時に "1" にセットすると, カウンタ許可と同時にカウントを開始します。カウンタ値がアンダフロー ("0000_H" → "FFFF_H") すると, "FFFF_H" の状態でカウンタを停止します。このとき, アンダフロー割込み要求フラグビット (UF) が "1" にセットされ, 割込み要求許可ビット (INTE) が "1" である場合は割込み要求が発生します。

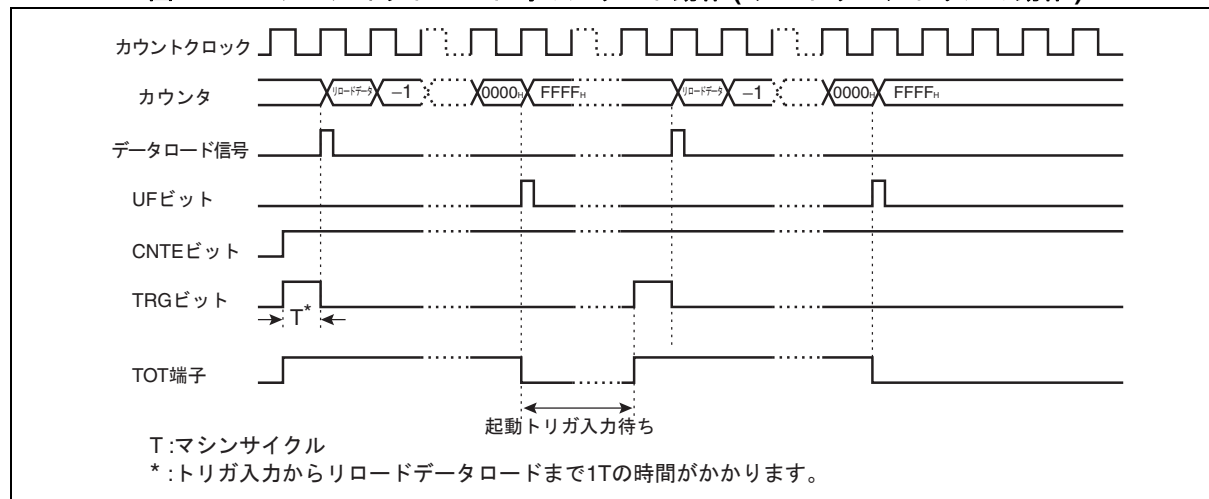
また, TOT 端子からはカウンタ中を示す矩形波を出力できます。

● ソフトウェアトリガの動作

タイマ制御ステータスレジスタ (TMCSR) の TRG ビットに "1" を書き込むことで, カウンタが起動されます。

図 14.4-7 に, ワンショットモード時のソフトウェアトリガ動作を示します。

図 14.4-7 ワンショットモード時のカウント動作 (ソフトウェアトリガの動作)

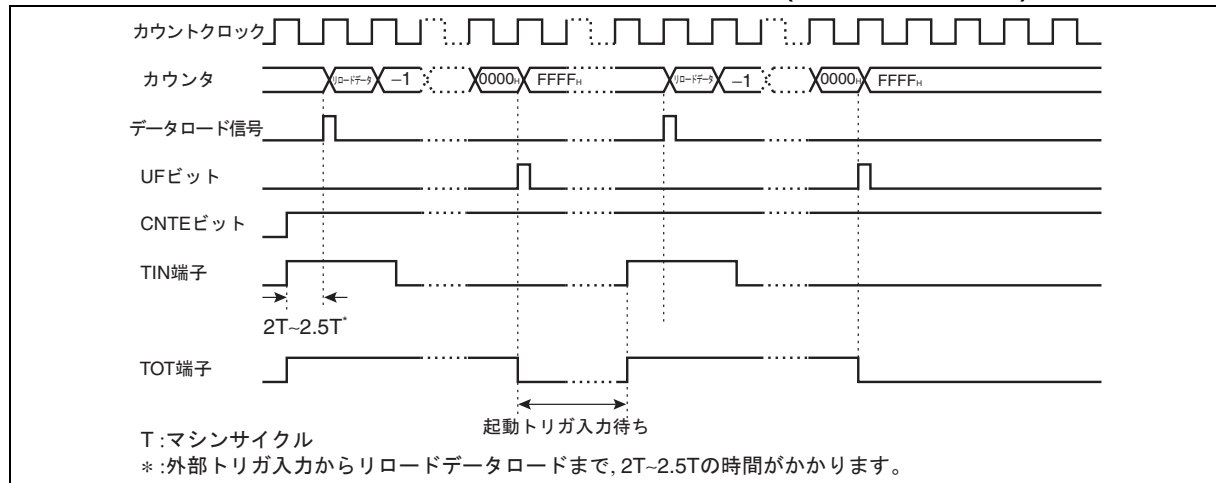


● 外部トリガの動作

有効エッジ (立上り , 立下り , 両エッジ選択可) が TIN 端子に入力されると , カウンタが起動されます。

図 14.4-8 に , ワンショットモード時の外部トリガの動作を示します。

図 14.4-8 ワンショットモード時のカウント動作 (外部トリガの動作)



<注意事項>

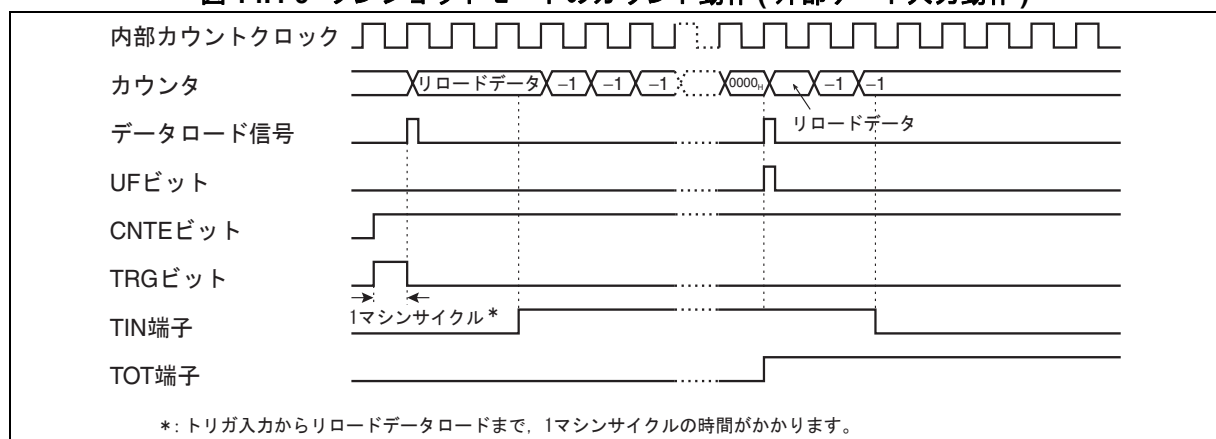
TIN 端子へ入力するトリガパルス幅は , $2T$ (T: マシンサイクル) 以上としてください。

● 外部ゲート入力の動作

有効レベル ("H" レベル , "L" レベル選択可) が TIN 端子に入力されている間 , カウンタ動作を行います。

図 14.4-9 に , ワンショットモード時のゲート入力動作を示します。

図 14.4-9 ワンショットモードのカウント動作 (外部ゲート入力動作)



<注意事項>

TIN端子へ入力するゲート入力のパルス幅は, $2T$ (T: マシンサイクル) 以上としてください。

14.4.4 イベントカウントモード

TIN 端子からの入力エッジをカウントして、16 ビットカウンタをカウントダウンし、カウンタのアンダフローで CPU に割込み要求を発生します。また、TOT 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の動作

カウント動作を許可 (TMCSR の CNTE=1) し、カウンタを起動 (TMCSR の TRG=1) すると、16 ビットリロードレジスタ (TMRLR) の値がカウンタにロードされ、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、両エッジ選択可) を検出するたびにカウントダウンします。

カウンタ許可ビットとソフトウェアトリガビットを同時に "1" にセットすると、カウンタ許可と同時にカウントを開始します。

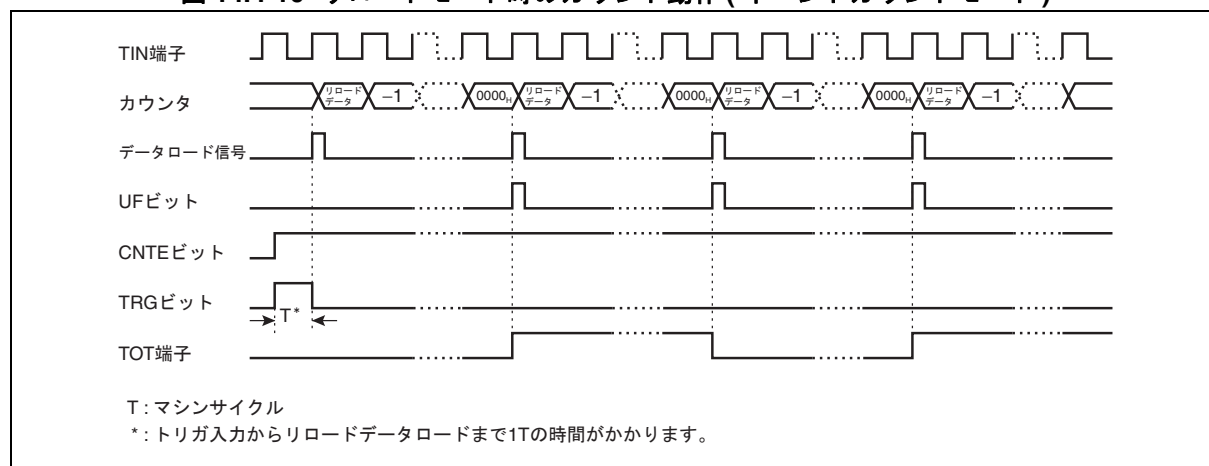
● リロードモード時の動作

カウンタ値がアンダフロー ("0000_H" → "FFFF_H") すると、16 ビットリロードレジスタ (TMRLR) の値をカウンタにロードして、カウント動作を継続させます。このとき、アンダフロー割込み要求フラグビット (UF) を "1" にセットし、割込み要求許可ビット (TMCSR の INTE) が "1" である場合は割込み要求を発生します。

また、TOT 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。

図 14.4-10 に、リロードモードのカウント動作を示します。

図 14.4-10 リロードモード時のカウント動作 (イベントカウントモード)



<注意事項>

TIN 端子へ入力されるクロックの "H" 幅および "L" 幅は、4T(T: マシンサイクル) 以上としてください。

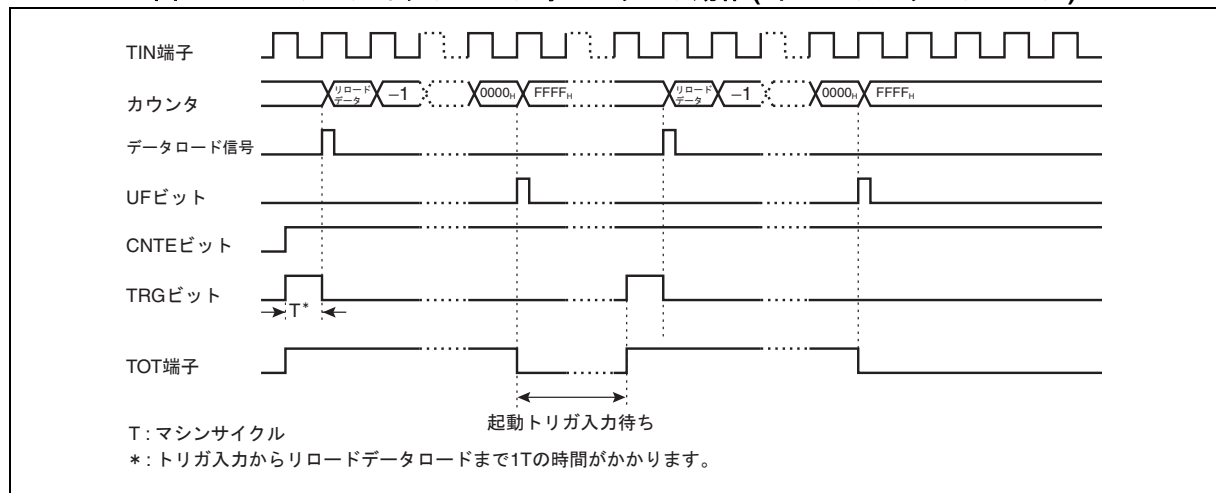
● ワンショットモード時の動作

カウンタ値がアンダフロー ("0000_H" → "FFFF_H") すると, "FFFF_H" の状態でカウンタが停止します。このとき, アンダフロー要求フラグビット (UF) が "1" にセットされ, 割込み要求許可ビット (INTE) が "1" である場合は割込み要求を発生します。

また, TOT 端子からは, カウント中を示す矩形波を出力できます。

図 14.4-11 に, ワンショットモードのカウンタ動作を示します。

図 14.4-11 ワンショットモード時のカウンタ動作 (イベントカウンタモード)



<注意事項>

TIN 端子へ入力されるクロックの "H" 幅および "L" 幅は, 4T(T: マシンサイクル) 以上としてください。

14.5 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プログラム例を示します。

■ 16 ビットリロードタイマのプログラム例

<p>設定手順例</p> <p>TOT0 からのパルス出力させるソフトトリガ, duty1/2, 通常極性</p> <p>< 初期設定 ></p> <ul style="list-style-type: none"> ・ 設定 <table border="1"> <thead> <tr> <th>レジスタ名</th> <th>ビット名</th> </tr> </thead> <tbody> <tr> <td>リロード値設定</td> <td>TMRLR</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ・ リロードタイマ 0 の制御 <table border="1"> <thead> <tr> <th>制御レジスタの設定</th> <th>TMCSR</th> </tr> </thead> <tbody> <tr> <td>クロックソース選択 >></td> <td>.CSL1, CSL0</td> </tr> <tr> <td>トリガ選択 >></td> <td>.MOD</td> </tr> <tr> <td>出力許可 >></td> <td>.OUTE</td> </tr> <tr> <td>出力レベル選択 >></td> <td>.OUTL</td> </tr> <tr> <td>動作モード選択 >></td> <td>.RELD</td> </tr> <tr> <td>割込み禁止 >></td> <td>.INTE</td> </tr> <tr> <td>割込みフラグクリア >></td> <td>.UF</td> </tr> <tr> <td>カウント停止 >></td> <td>.CNTE</td> </tr> <tr> <td>ソフトトリガ (未処理) >></td> <td>.TRG</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ・ 割込み関連 <table border="1"> <tbody> <tr> <td>リロードタイマ割込みレベル</td> <td>ICR12</td> </tr> <tr> <td>I フラグの設定</td> <td>(CCR)</td> </tr> </tbody> </table> <p>< 起動 ></p> <ul style="list-style-type: none"> ・ リロードタイマ起動 <table border="1"> <thead> <tr> <th>レジスタ名</th> <th>ビット名</th> </tr> </thead> <tbody> <tr> <td>PPG 起動</td> <td>TMCSR</td> </tr> <tr> <td>割込みフラグクリア >></td> <td>UF</td> </tr> <tr> <td>割込み許可 >></td> <td>.INTE</td> </tr> <tr> <td>カウント許可 >></td> <td>.CNTE</td> </tr> <tr> <td>ソフトトリガ (起動) >></td> <td>.TRG</td> </tr> </tbody> </table> <p>< 割込み ></p> <ul style="list-style-type: none"> ・ 割込み処理 <table border="1"> <thead> <tr> <th>(任意の処理)</th> <th></th> </tr> </thead> <tbody> <tr> <td>割込み要求フラグクリア</td> <td>TMCSR.UF</td> </tr> </tbody> </table> <p>< 割込みベクタ ></p> <ul style="list-style-type: none"> ・ ベクタテーブルの設定 <p>(注意事項)</p> <p>事前にクロック関連の設定および, __set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	レジスタ名	ビット名	リロード値設定	TMRLR	制御レジスタの設定	TMCSR	クロックソース選択 >>	.CSL1, CSL0	トリガ選択 >>	.MOD	出力許可 >>	.OUTE	出力レベル選択 >>	.OUTL	動作モード選択 >>	.RELD	割込み禁止 >>	.INTE	割込みフラグクリア >>	.UF	カウント停止 >>	.CNTE	ソフトトリガ (未処理) >>	.TRG	リロードタイマ割込みレベル	ICR12	I フラグの設定	(CCR)	レジスタ名	ビット名	PPG 起動	TMCSR	割込みフラグクリア >>	UF	割込み許可 >>	.INTE	カウント許可 >>	.CNTE	ソフトトリガ (起動) >>	.TRG	(任意の処理)		割込み要求フラグクリア	TMCSR.UF	<p>プログラム例</p> <pre> void RT_sample(void) { RT_initial(); RT_start(); } void RT_initial(void) { IO_TMRLR = 0xAA; /* リロード値は任意の値を入れてください */ IO_TMCSR.word = 0x0050; /* 設定値 = 0000_0000_0101_0000 */ /* bit15-12=0000 未定義ビット */ /* bit11-10=00 CSL1, CSL0 内部クロック $\phi/2$ */ /* bit9-7=000 MOD ソフトウェアトリガ */ /* bit6=1 OUTE 出力許可 */ /* bit5=0 OUTL 外部出力レベル Low */ /* bit4=1 RELD リロード許可 */ /* bit3=0 INTE 割込禁止 */ /* bit2=0 UF 割込要求フラグのクリア */ /* bit1=0 CNTE カウント停止 */ /* bit0=0 TRG ソフトウェアトリガ */ IO_ICR12.byte = 0x10; /* 割込みレベルを設定 */ __EI(); /* 割込み許可 */ } void RT_start(void) { IO_TMCSR.word = 0x005B; /* bit3=1 INTE 割込許可 */ /* bit2=0 UF 割込要求フラグのクリア */ /* bit1=1 CNTE カウント開始 */ /* bit0=1 TRG ソフトウェアトリガ */ } __interrupt void RT_int(void) /* アンダフロー発生時に割込みがかかる */ { IO_TMCSR.bit.UF = 0; /* bit2=0 UF 割込要求フラグのクリア */ /* 任意の処理動作 */ } #pragma intvect RT_int 35 </pre> <p>(注意事項)</p> <p>レジスタの記述形式については「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
レジスタ名	ビット名																																												
リロード値設定	TMRLR																																												
制御レジスタの設定	TMCSR																																												
クロックソース選択 >>	.CSL1, CSL0																																												
トリガ選択 >>	.MOD																																												
出力許可 >>	.OUTE																																												
出力レベル選択 >>	.OUTL																																												
動作モード選択 >>	.RELD																																												
割込み禁止 >>	.INTE																																												
割込みフラグクリア >>	.UF																																												
カウント停止 >>	.CNTE																																												
ソフトトリガ (未処理) >>	.TRG																																												
リロードタイマ割込みレベル	ICR12																																												
I フラグの設定	(CCR)																																												
レジスタ名	ビット名																																												
PPG 起動	TMCSR																																												
割込みフラグクリア >>	UF																																												
割込み許可 >>	.INTE																																												
カウント許可 >>	.CNTE																																												
ソフトトリガ (起動) >>	.TRG																																												
(任意の処理)																																													
割込み要求フラグクリア	TMCSR.UF																																												

■ プログラム例以外の設定方法

● リロード値の設定 (書換え) 方法

リロード値は 16 ビットリロードレジスタ TMRLR に設定します。

以下に設定する値の計算式を示します。

<式>

$$\text{TMRLR レジスタ値} = \{ \text{リロード間隔} / \text{カウントクロック} \} - 1$$

<設定可能な範囲>

$$\text{TMRLR レジスタ値} = 0 \sim \text{FFFF}_{\text{H}}(65535)$$

● カウントクロックの種類と選択方法

カウントクロックは下表の4種類から, カウントクロック選択ビット(TMCSR.CSL[1:0])で選択します。

カウント クロック	カウントクロック選 択ビット		カウントクロック例		
	CSL1	CSL0	$\phi = 32\text{MHz}$ 時	$\phi = 16\text{MHz}$ 時	$\phi = 8\text{MHz}$ 時
$\phi/2$	0	0	62.5ns	125ns	250ns
$\phi/8$	0	1	250ns	500ns	1.0 μs
$\phi/32$	1	0	1.0 μs	2.0 μs	4.0 μs
外部イベント	1	1	パルス幅: $2/\phi$ min		

● リロードタイマのカウント動作を許可 / 停止する方法

タイマのカウントイネーブルビット (TMCSR.CNTE) で設定します。

制御内容	動作許可ビット (CNTE)
リロードタイマを停止するには	"0" にする
リロードタイマのカウント動作を許可するには	"1" にする

停止した状態からは再開できません。許可は起動前か起動と同時にしてください。

● リロードタイマのモード (リロード / ワンショット) を設定する方法

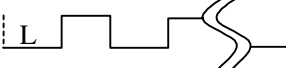
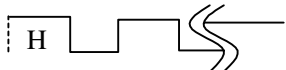

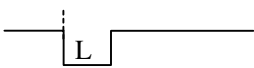
モード選択ビット (TMCSR.RELD) で設定します。

動作モード	モード選択ビット (RELD)
ワンショットにするには	"0" にする
リロードにするには	"1" にする

● 出力レベルを反転させる方法

出力レベルを下表に示します。

タイマ出力レベルビット (TMCSR.OUTL) で設定します。

出力レベル	タイマ出力レベルビット (OUTL)
リロードモード, 初期値 "L" レベル出力 	"0" にする
リロードモード, 初期値 "H" レベル出力 (反転) 	"1" にする
ワンショットモード, カウント中 "H" レベル出力 	"0" にする
ワンショットモード, カウント中 "L" レベル出力 (反転) 	"1" にする

● 起動方法

- 内部クロックモードの起動トリガは 4 種あります。トリガ選択ビット (TMCSR.MOD[2:0]) で設定します。

トリガ	トリガ指定ビット (MOD[2:0])
ソフトウェアトリガ (TRG ビットのセット)	"000 _B " にする
TIN0 端子からの外部トリガ (立上りエッジ)	"001 _B " にする
TIN0 端子からの外部トリガ (立下りエッジ)	"010 _B " にする
TIN0 端子からの外部トリガ (両エッジ)	"011 _B " にする

- イベントカウントモードの起動はソフトウェアで行えます。タイマ制御ステータスレジスタ (TMCSR) のカウンタ許可ビット (CNTE) とソフトウェアトリガビット (TRG) を同時に "1" に設定します。

● イベントカウントモード時の有効エッジの種類と選択方法

設定は、トリガ選択ビット (TMCSR.MOD[1:0]) で行います。

有効エッジは、3 種あります。

有効エッジ	トリガ選択ビット (MOD1, MOD0)
立上りエッジ	"01 _B " にする
立下りエッジ	"10 _B " にする
両エッジ	"11 _B " にする

MOD2 の設定は "0", "1" にかかわらず意味を持ちません。

● TIN 端子を外部イベント入力端子、または外部トリガ入力端子にする方法

データ方向指定ビット (DDR7.P73) に "0" を書き込みます。

端子	制御ビット	
TIN0 端子	ポート 7 方向レジスタ (DDR7)	データ方向指定ビット (P73)

● 起動トリガの発生方法

• ソフトトリガの発生方法

設定はソフトウェアトリガビット (TMCSR.TRG) で行います。

ソフトウェアトリガビット (TGR) に "1" を書き込むとトリガが発生します。

動作の許可と起動を同時に行うためには、カウント許可ビット (TMCSR.CNTE) とソフトウェアトリガビット (TMCSR.TRG) を同時に設定します。

• 外部トリガの発生方法

各リロードタイマに対応するトリガ端子に、トリガ選択ビットで指定したエッジを入力すると、トリガが発生します。

タイマ	トリガ端子
リロードタイマ	TIN0

● 割込みに関連するレジスタ

リロードタイマ番号と割込みレベル、ベクタ、制御レジスタ等との関係は下表のとおりです。

割込みレベル、割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

	割込みベクタ	割込みレベル設定ビット
リロードタイマ	#35 アドレス: FFFF70 _H	割込み制御レジスタ 12 (ICR12) アドレス: 0000BC _H

割込み要求フラグ (TMCSR.UF) は自動的にクリアされませんので、割込み処理から復帰する前に UF ビットに "0" を書き込みクリアしてください。

● 割込みを許可する方法

割込み要求の許可、割込み要求フラグ

割込み許可の設定は、割込み要求許可ビット (TMCSR.INTE) で行います。

	割込み要求許可ビット (INTE)
割込み要求禁止にするには	"0" にする
割込み要求許可にするには	"1" にする

割込み要求のクリアは、割込み要求ビット (TMCSR.UF) で行います。

	タイマ割込み要求ビット (UF)
割込み要求禁止にするには	"0" にする

<注意事項>

リロードタイマアンダフロー割込みを許可 (TMCSR レジスタの INTE ビット = 1) から禁止 (TMCSR レジスタの INTE ビット = 0) にする場合は、割込み制御レジスタ (ICR12) の IL2 ~ IL0 ビット = "111" として割込みを禁止にしてから INTE ビットに "0" を書き込んでください。

● リロードタイマを停止する方法

リロードタイマ停止ビットの設定で行います。

制御内容	動作許可ビット (CNTE)
リロードタイマを停止するには	"0" にする

第15章

8/16 ビット PPG タイマ

8/16 ビット PPG タイマの概要，レジスタの構成と機能，割込みおよび動作について説明します。

15.1 8/16 ビット PPG タイマの概要

15.2 8/16 ビット PPG タイマの構成

15.3 8/16 ビット PPG タイマのレジスタの構成と機能

15.4 8/16 ビット PPG タイマの割込み

15.5 8/16 ビット PPG タイマの動作

15.6 8/16 ビット PPG タイマのプログラム例

15.1 8/16 ビット PPG タイマの概要

8/16 ビット PPG タイマは、任意の周期およびデューティ比のパルス出力が可能な 6 チャンネルのリロードタイマモジュールです。

ハードウェアとして、8 ビットダウンカウンタ 6 個、8 ビットリロードタイマ 12 個、16 ビット制御レジスタ 3 個、外部パルス出力端子 6 本、割込み出力 6 本を有しています。

なお、MB90480/485 シリーズには、8 ビット PPG として 6 チャンネルあり、このうち PPG0+PPG1, PPG2+PPG3, PPG4+PPG5 の組合せで 16 ビット PPG(3 チャンネル)としても使用できます。

■ 8/16 ビット PPG タイマの機能

- 8 ビット PPG 出力 6 チャンネル独立動作モード

6 チャンネルの独立した PPG 出力動作が可能です。

- 16 ビット PPG 出力動作モード

3 チャンネルの 16 ビットの PPG 出力動作が可能です。

PPG0+PPG1, PPG2+PPG3, PPG4+PPG5 の組合せで使用します。

- 8+8 ビット PPG 出力動作モード

PPG0(PPG2/PPG4) の出力を PPG1(PPG3/PPG5) のクロックに入力することによって、任意周期の 8 ビット PPG 出力が可能です。

- PPG 出力動作

任意周期およびデューティ比のパルス波を出力します。

外付け回路を使用することにより D/A コンバータとしても使用できます。

15.2 8/16 ビット PPG タイマの構成

8/16 ビット PPG タイマの ch.0/ch.2/ch.4 および ch.1/ch.3/ch.5 の構成を示します。

■ 8/16 ビット PPG タイマのブロックダイアグラム

図 15.2-1 に ch.0/ch.2/ch.4 のブロックダイアグラムを、図 15.2-2 に ch.1/ch.3/ch.5 のブロックダイアグラムを示します。

図 15.2-1 8/16 ビット PPG タイマのブロックダイアグラム (ch.0/ch.2/ch.4)

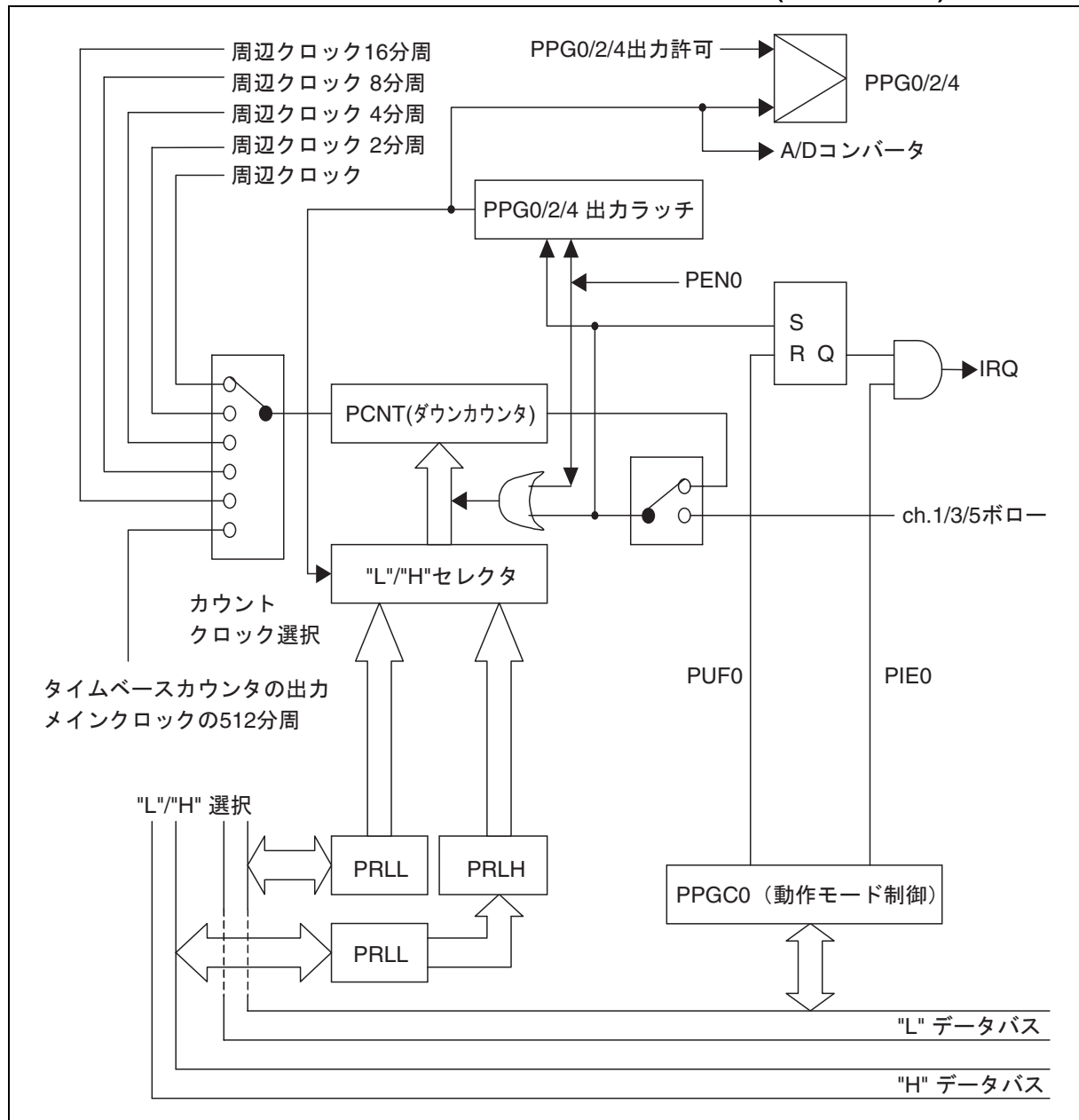
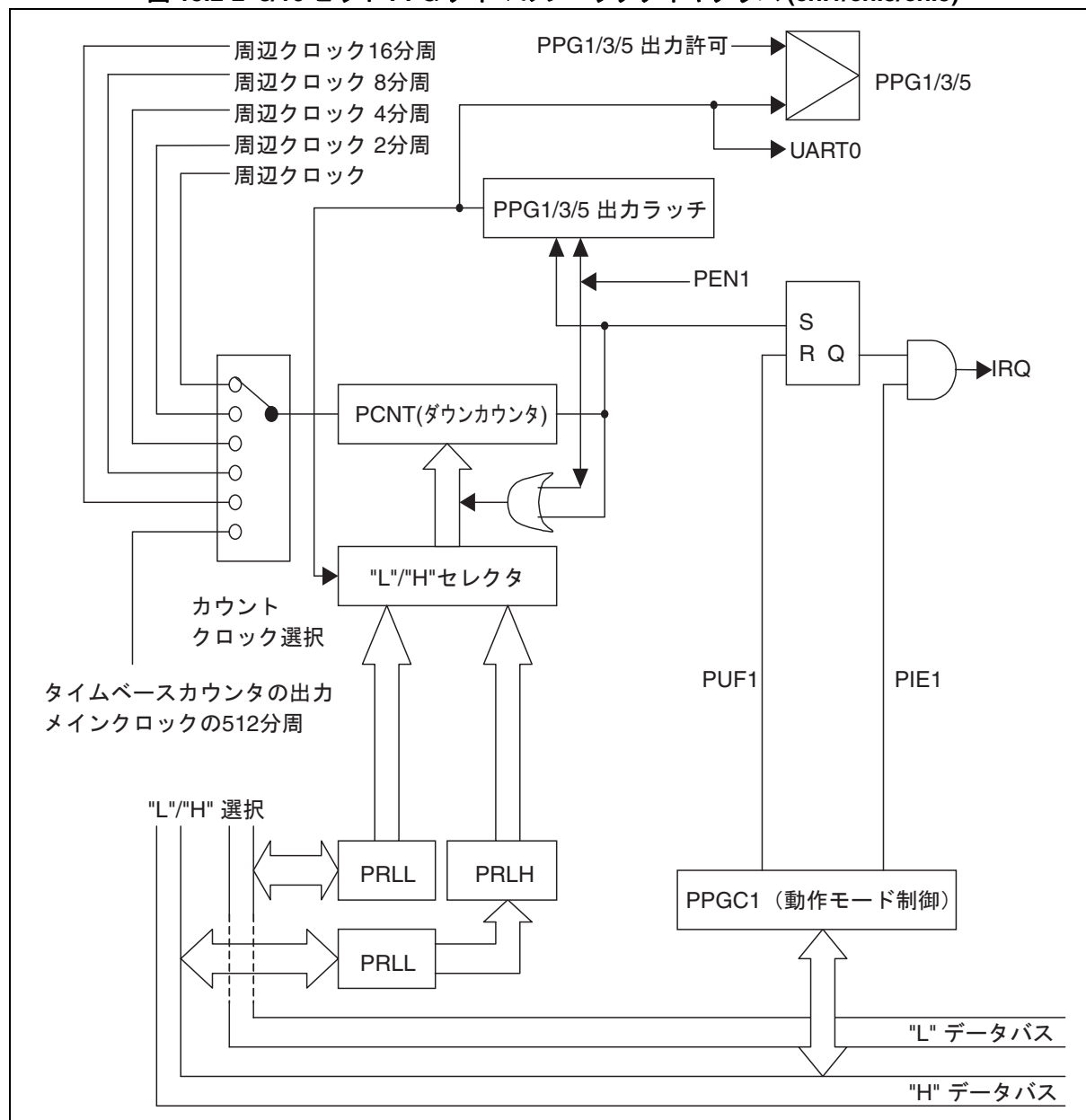


図 15.2-2 8/16 ビット PPG タイマのブロックダイヤグラム (ch.1/ch.3/ch.5)



■ 8/16 ビット PPG タイマに関する端子

8/16 ビット PPG タイマに関する端子は、PPG0/PPG1/PPG2/PPG3/PPG4/PPG5 端子があります。これらの端子は、汎用入出力ポート (P24/PPG0, P25/PPG1, P26/PPG2, P27/PPG3, P94/PPG4, P95/PPG5) と、PPG タイマの出力端子として機能を兼用しています。

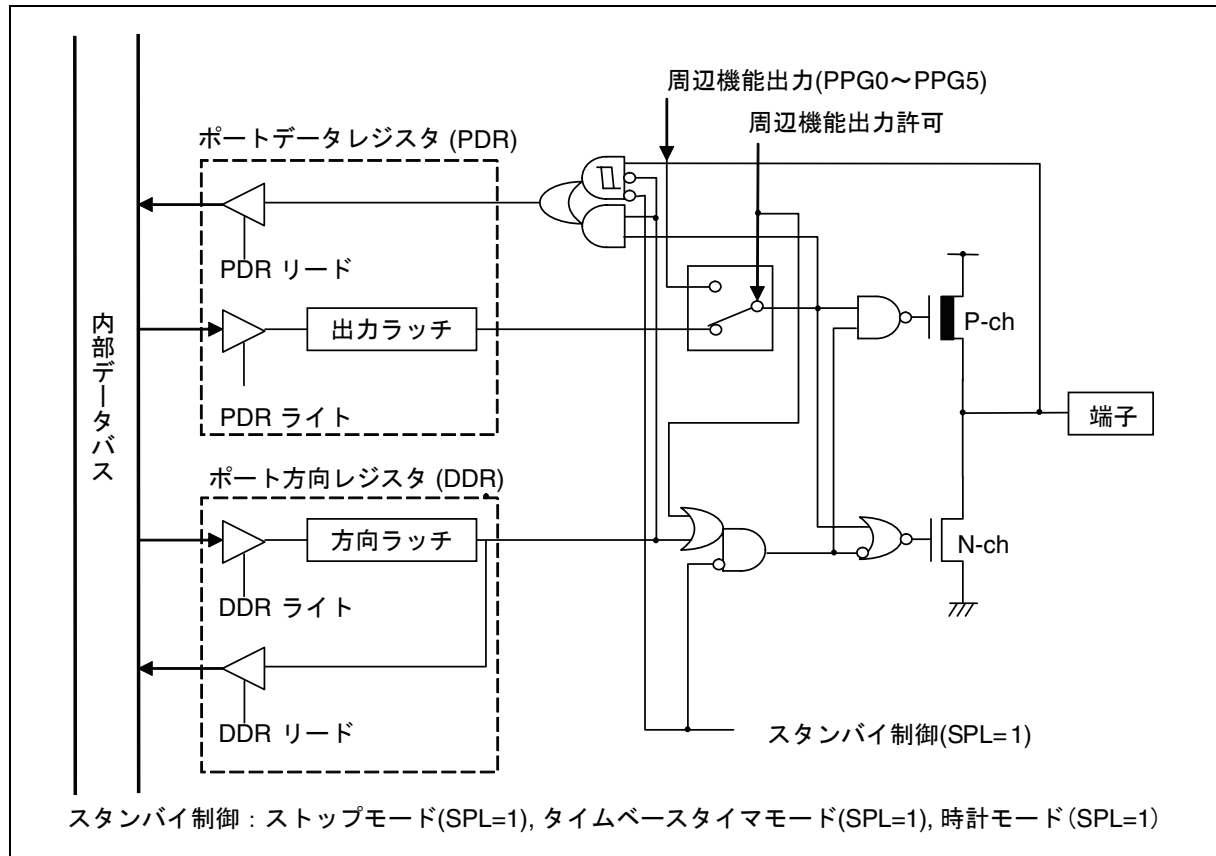
● PPG0/PPG1/PPG2/PPG3/PPG4/PPG5 端子として使用する場合の設定

PPG0/PPG1/PPG2/PPG3/PPG4/PPG5 端子を出力として使用する場合は、ポート方向レジスタ (DDR) の値に関係なく自動的に出力端子となります。

PPG0/PPG1/PPG2/PPG3 端子を使用する場合は、外部アドレス出力制御レジスタ (HACR) の E20 から E23 ビットを "1" (I/O ポート設定) としてください。

■ 8/16 ビット PPG タイマに関する端子のブロックダイアグラム

図 15.2-3 8/16 ビット PPG タイマに関する端子のブロックダイアグラム



15.3 8/16 ビット PPG タイマのレジスタの構成と機能

8/16 ビット PPG タイマで使用するレジスタの構成と機能について説明します。

■ 8/16 ビット PPG タイマのレジスタ一覧

図 15.3-1 に、8/16 ビット PPG タイマのレジスタ一覧を示します。

図 15.3-1 8/16 ビット PPG タイマのレジスタ一覧

ch. 0	00003A _H	7	6	5	4	3	2	1	0	PPGC0/2/4
ch. 2	00003C _H	PEN0	—	PE00	PIE0	PUF0	—	—	予約	動作モード制御レジスタ
ch. 4	00003E _H	(R/W)	(—)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)	リード/ライト
		(0)	(X)	(0)	(0)	(0)	(X)	(X)	(1)	初期値
ch. 1	00003B _H	15	14	13	12	11	10	9	8	PPGC1/3/5
ch. 3	00003D _H	PEN1	—	PE10	PIE1	PUF1	MD1	MD0	予約	動作モード制御レジスタ
ch. 5	00003F _H	(R/W)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	リード/ライト
		(0)	(X)	(0)	(0)	(0)	(0)	(0)	(1)	初期値
ch. 0, ch. 1	000040 _H	7	6	5	4	3	2	1	0	PPG01/PPG23/PPG45
ch. 2, ch. 3	000042 _H	PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	予約	予約	出力制御レジスタ
ch. 4, ch. 5	000044 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	初期値
ch. 0	00002E _H	7	6	5	4	3	2	1	0	PRL0~5
ch. 1	000030 _H	D07	D06	D05	D04	D03	D02	D01	D00	リロードレジスタL
ch. 2	000032 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
ch. 3	000034 _H	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値
ch. 4	000036 _H									
ch. 5	000038 _H									
ch. 0	00002F _H	15	14	13	12	11	10	9	8	PRL0~5
ch. 1	000031 _H	D15	D14	D13	D12	D11	D10	D09	D08	リロードレジスタH
ch. 2	000033 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
ch. 3	000035 _H	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値
ch. 4	000037 _H									
ch. 5	000039 _H									

15.3.1 PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)

PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4) の構成と機能について説明します。

■ PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)

PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4) は、ch.0/ch.2/ch.4 の動作モード選択、端子出力制御、カウントクロック選択およびトリガ制御を行います。

下図に、PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4) のビット構成を示します。

ch. 0	00003A _H	7	6	5	4	3	2	1	0	PPGC0/2/4
ch. 2	00003C _H	PEN0	—	PE00	PIE0	PUF0	—	—	予約	動作モード制御レジスタ
ch. 4	00003E _H	(R/W)	(—)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)	リード/ライト
		(0)	(X)	(0)	(0)	(0)	(X)	(X)	(1)	初期値

以下に、PPG0/2/4動作モード制御レジスタ(PPGC0/2/4)の各ビットの機能を説明します。

[bit7] PEN0:ppg Enable(動作許可)

このビットは、PPG の動作モードを選択します。

PEN0	動作状態
0	動作停止 ("L" レベル出力保持)
1	PPG 動作イネーブル

- このビットに "1" を書き込むことによって、PPG はカウントを開始します。
- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit5] PE00:ppg output Enable 00(PPG0/2/4 出力端子許可)

このビットは、パルス出力外部端子 PPG0/2/4 へのパルス出力の禁止 / 許可を制御します。

PE00	動作状態
0	汎用ポート端子 (パルス出力禁止)
1	PPG0/2/4 パルス出力 (パルス出力許可)

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit4] PIE0:ppg Interrupt Enable(PPG の割込み許可)

このビットは、PPG の割込みの禁止 / 許可を制御します。

PIE0	動作状態
0	割込み禁止
1	割込み許可

- このビットが "1" のとき、PUF0 が "1" になると割込み要求が発生します。
このビットが "0" のときは、割込みは発生しません。
- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit3] PUF0:ppg Underflow Flag(PPG カウンタアンダフロー)

このビットは、PPG のカウンタアンダフローの検出結果を表示します。

PUF0	動作状態
0	PPG カウンタアンダフローを検出していません。
1	PPG カウンタアンダフローを検出しました。

8 ビット PPG6 チャンネルモード (PPG0, 1/PPG2, 3/PPG4, 5) および 8 ビットプリスケアラ +8 ビット PPG モード時には、ch.0, ch.2, ch.4 のカウンタの値が 00_H → FF_H になったときのアンダフローにより "1" にセットされます。16 ビット PPG3 チャンネルモード (PPG0, 1/PPG2, 3/PPG4, 5) 時には、ch.1, ch.3, ch.5 / ch.0, ch.2, ch.4 のカウンタの値が 0000_H → FFFF_H になったときのアンダフローにより "1" にセットされます。"0" の書込みにより "0" になります。このビットへの "1" の書込みは意味がありません。リードモディファイライト系命令の読出しに対しては、"1" が読み出されます。

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit0] 予約ビット

このビットは、予約ビットです。PPGC0/2/4 を設定するときは、必ず "1" に設定してください。

15.3.2 PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)

PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5) の構成および機能について説明します。

■ PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)

PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5) は、ch.1/ch.3/ch.5 の動作モード選択、端子出力制御、カウントクロック選択およびトリガの制御を行います。

下図に、PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5) のビット構成を示します。

ch. 1	00003B _H	15	14	13	12	11	10	9	8	PPGC1/3/5
ch. 3	00003D _H	PEN1	—	PE10	PIE1	PUF1	MD1	MD0	予約	動作モード制御レジスタ
ch. 5	00003F _H	(R/W)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	リード/ライト
		(0)	(X)	(0)	(0)	(0)	(0)	(0)	(1)	初期値

以下に、PPG1/3/5動作モード制御レジスタ(PPGC1/3/5)の各ビットの機能を説明します。

[bit15] PEN1:ppg Enable(動作許可)

このビットは、PPG の動作モードを選択します。

PEN1	動作状態
0	動作停止 ("L" レベル出力保持)
1	PPG 動作イネーブル

- このビットに "1" を書き込むことによって、PPG はカウントを開始します。
- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit13] PE10:ppg output Enable 10(PPG1/3/5 出力端子許可)

このビットは、パルス出力外部端子 PPG1/3/5 へのパルス出力の禁止 / 許可を制御します。

PE10	動作状態
0	汎用ポート端子 (パルス出力禁止)
1	PPG1/3/5 パルス出力 (パルス出力許可)

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit12] PIE1:ppg Interrupt Enable(PPG の割込み許可)

このビットは、PPG の割込みの禁止 / 許可を制御します。

PIE1	動作状態
0	割込み禁止
1	割込み許可

このビットが "1" のとき PUF0 が "1" になると、割込み要求が発生します。このビットが "0" のときは、割込みは発生しません。

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit11] PUF1:ppg Underflow Flag(PPG カウンタアンダフロー)

このビットは、PPG のカウンタアンダフローの検出結果を表示します。

PUF1	動作状態
0	PPG カウンタアンダフローを検出していません。
1	PPG カウンタアンダフローを検出しました。

8 ビット PPG6 チャンネルモード (PPG0, 1/PPG2, 3/PPG4, 5) および 8 ビットプリスケアラ +8 ビット PPG モード時には、ch.1, ch.3, ch.5 のカウンタの値が 00_H → FF_H になったときのアンダフローにより "1" にセットされます。16 ビット PPG3 チャンネルモード (PPG0, 1/PPG2, 3/PPG4, 5) 時には、ch.1, ch.3, ch.5 / ch.0, ch.2, ch.4 のカウンタの値が 0000_H → FFFF_H になったときのアンダフローにより "1" にセットされます。"0" の書込みにより "0" になります。このビットへの "1" の書込みは意味がありません。リードモディファイライト系命令系の読出しに対しては、"1" が読み出されます。

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit10, bit9] MD1, 0:ppg count Mode(動作モード選択)

これらのビットは、PPG タイマの動作モードを選択します。

MD1	MD0	動作モード
0	0	8 ビット PPG2 チャンネル独立モード (× 3 あり)
0	1	8 ビットプリスケアラ +8 ビット PPG1 チャンネル
1	0	予約 (設定禁止)
1	1	16 ビット PPG1 チャンネルモード (× 3 あり)

- このビットは、リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。

<注意事項>

- このビットを "10" に設定しないでください。
- このビットを "01" に設定する場合は、PPGC0 の PEN0 ビット /PPGC1 の PEN1 ビットを "01" に設定しないでください。また、PEN0 ビットおよび PEN1 ビットを同時に "11" または "00" にセットすることを推奨します。
- このビットを "11" に設定する場合は、PPGC0/PPGC1 をワード転送で書き換え、PEN0/PEN1 を同時に "11", または "00" にセットしてください。

[bit8] 予約ビット

このビットは、予約ビットです。PPGC1/3/5 を設定するときは、必ず "1" に設定してください。

15.3.3 PPG0 ～ 5 出力制御レジスタ (PPG01/PPG23/PPG45)

PPG0 ～ 5 出力制御レジスタ (PPG01/PPG23/PPG45) の構成および機能について説明します。

■ PPG0 ～ 5 出力制御レジスタ (PPG01/PPG23/PPG45)

下図に、PPG0 ～ 5 出力制御レジスタ (PPG01/PPG23/PPG45) のビット構成を示します。

ch. 0, ch. 1	000040 _H	7	6	5	4	3	2	1	0	PPG01/PPG23/PPG45
ch. 2, ch. 3	000042 _H	PCS2	PCS1	PCS0	PCM2	PCM1	PCMO	予約	予約	出力制御レジスタ
ch. 4, ch. 5	000044 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	初期値

以下に、PPG0 ～ 5 出力制御レジスタ (PPG01/PPG23/PPG45) の各ビットの機能を説明します。

[bit7, bit6, bit5] PCS2 ～ PCS0:ppg Count Select(カウントクロック選択)

これらのビットは、ch.1, ch.3, ch.5 のダウンカウンタの動作クロックを選択します。

PCS2	PCS1	PCS0	動作モード
0	0	0	周辺クロック (62.5ns マシンクロック 16MHz 時)
0	0	1	周辺クロック /2(125ns マシンクロック 16MHz 時)
0	1	0	周辺クロック /4(250ns マシンクロック 16MHz 時)
0	1	1	周辺クロック /8(500ns マシンクロック 16MHz 時)
1	0	0	周辺クロック /16(1μs マシンクロック 16MHz 時)
1	1	1	タイムベースカウンタからの入力クロック ($2^9 \times 250\text{ns} = 128\mu\text{s}$ 原発振 4MHz 時)

- このビットは、リセットによって "000" に初期化されます。
- 読出し / 書込みが可能です。

<注意事項>

8 ビットプリスケラ +8 ビット PPG モードおよび 16 ビット PPG モード時には、ch.1, ch.3, ch.5 の PPG は ch.0, ch.2, ch.4 からカウントクロックを受けて動作するため、PCS2 ～ 0 ビットの指定は無効になります。

[bit4, bit3, bit2] PCM2 ~ PCM0:ppg Count Mode(カウントクロック選択)

これらのビットは , ch.0, ch.2, ch.4 のダウンカウンタの動作クロックを選択します。

PCM2	PCM1	PCM0	動作モード
0	0	0	周辺クロック (62.5ns マシンクロック 16MHz 時)
0	0	1	周辺クロック /2(125ns マシンクロック 16MHz 時)
0	1	0	周辺クロック /4(250ns マシンクロック 16MHz 時)
0	1	1	周辺クロック /8(500ns マシンクロック 16MHz 時)
1	0	0	周辺クロック /16(1μs マシンクロック 16MHz 時)
1	1	1	タイムベースカウンタからの入力クロック ($2^9 \times 250\text{ns} = 128\mu\text{s}$ 原発振 4MHz 時)

- このビットは , リセットによって "000" に初期化されます。
- 読出し / 書込みが可能です。

[bit1, bit0] 予約ビット

このビットは , 予約ビットです。PPG01/23/45 を設定するときは , 必ず "00" に設定してください。

15.3.4 リロードレジスタ (PRL0 ~ 5, PRLH0 ~ 5)

リロードレジスタ (PRL0 ~ 5, PRLH0 ~ 5) の構成および機能について説明します。

■ リロードレジスタ (PRL0 ~ 5, PRLH0 ~ 5)

下図に、リロードレジスタ (PRLLO ~ 5, PRLHO ~ 5) のビット構成を示します。

[illegible]

リロードレジスタ (PRLLO ~ 5, PRLH0 ~ 5) は、ダウncカウンタ (PCNT) へのリロード値を保持する、各 8 ビットのレジスタです。それぞれ下表に示す役割を持っています。

レジスタ名	機能
PRLl	"L" 側リロード値保持
PRLH	"H" 側リロード値保持

- どちらのレジスタも、読出し/書込みが可能です。

＜注意事項＞

8ビットプリスケラ+8ビットPPGモードで使用する場合には、ch.0, ch.2, ch.4のPRLとPRLHには、同じ値に設定することを推奨します。

15.4 8/16 ビット PPG タイマの割込み

8/16 ビット PPG タイマの割込みは、PPG カウンタアンダフローを検出した場合に発生します。

PPG カウンタアンダフローの割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) は起動できません。

■ 8/16 ビット PPG タイマの割込み

8/16 ビット PPG タイマの割込み制御ビットと割込み要因を、表 15.4-1 に示します。

表 15.4-1 8/16 ビット入出力タイマの割込み

	PPG0/2/4 オーバフロー割込み	PPG1/3/5 オーバフロー割込み
割込み要求フラグ	PPG0:PUF0 (bit3) ch.0 PPG2:PUF0 (bit3) ch.2 PPG4:PUF0 (bit3) ch.4	PPG1:PUF1 (bit3) ch.1 PPG3:PUF1 (bit3) ch.3 PPG5:PUF1 (bit3) ch.5
割込み要求出力許可ビット	PPG0:PUE0 (bit4) ch.0 PPG2:PUE0 (bit4) ch.2 PPG4:PUE0 (bit4) ch.4	PPG1:PUE1 (bit4) ch.1 PPG3:PUE1 (bit4) ch.3 PPG5:PUE1 (bit4) ch.5
割込み発生要因	PPG0/2/4 ダウンカウンタのオーバフロー	PPG1/3/5 ダウンカウンタのオーバフロー

■ PPG カウンタアンダフロー割込み

● 8 ビット PPG と 8+8 ビット出力動作モードの場合

- 8 ビット PPG6 チャンネル独立動作モードまたは 8+8 ビット PPG 出力動作モードでは、それぞれ独立して割込みを発生できます。
- PPG のダウンカウンタが "00" から "FF" にカウントダウンすると、アンダフローが発生します。アンダフローが発生すると、アンダフローしたチャンネル側のアンダフロービットが設定されます (PUF0=1, PUF1=1)。

● 16 ビット PPG 出力動作モードの場合

- 16 ビット PPG 出力動作モードでは、PPG0+PPG1/PPG2+PPG3/PPG4+PPG5 ダウンカウンタが "0000" から "FFFF" にダウンカウントすると、アンダフローが発生します。アンダフローが発生すると、2 チャンネルのアンダフロー発生ビットが同時に設定されます (PIF0=1, PIF1=1)。
- 2 チャンネルの割込み要求のうち、どちらかが許可されている場合は (PIE0=0+PIE1=1, PIE0=1+PIE1=0), アンダフローが発生すると割込みが発生します。
- 割込み要求が重複して発生しないように、事前に 2 チャンネルのアンダフロー割込み許可ビットのうち、どちらかを禁止にしておいてください (PIE0=0+PIE1=1 PIE0=1+PIE1=0)。
- 2 チャンネルのアンダフロー発生フラグが設定された場合は (PUF0=1 かつ PUF1=1), 2 チャンネル同時クリアしてください。

■ 8/16 ビット PPG タイマの割込みと DMA 転送および EI²OS

表 15.4-2 に、ソフトウェア割込みを除く割込み要因と割込みベクタ割込み制御レジスタの関係を示します。

表 15.4-2 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
PPG0/PPG1 カウンタボロー *	×	×	#22	FFFFA4 _H	ICR05	0000B5 _H
PPG2/PPG3 カウンタボロー	×	×	#23	FFFFA0 _H	ICR06	0000B6 _H
PPG4/PPG5 カウンタボロー	×	×	#24	FFF9C _H		

×：割込み要求フラグはクリアされません。

*：この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。
詳細につきましては、表 3.2-2 を参照してください。

（注意事項） 同一割込み番号に2つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

8/16 ビット PPG タイマは、DMA 転送機能、および EI²OS 機能に対応していません。

15.5 8/16 ビット PPG タイマの動作

8/16 ビット PPG タイマには、8 ビット長の PPG ユニットが 6 チャンネル (PPG0, 1/PPG2, 3/PPG4, 5) あります。それぞれを独立モードで動作させる以外にも、直結 (PPG0+PPG1/PPG2+PPG3/PPG4+PPG5) 動作をさせることによって、8 ビットプリスケアラ +8 ビット PPG モード、または 16 ビット PPG モードで動作させることもできます。計 3 種類のモードで動作を行えます。

■ 8/16 ビット PPG タイマの動作概略

8 ビット長の各 PPG ユニットは、それぞれ 8 ビット長のリロードレジスタ "L" 側と "H" 側の 2 本を持っています (PRL, PRLH)。

このレジスタに書き込まれた値が、8 ビットダウンカウンタ (PCNT) の "L" 側 / "H" 側交互にリロードされてカウントクロックごとにダウンカウントされます。カウントのボロー発生によるリロード時に、出力端子の値を反転させます。この動作によって、端子出力はリロードレジスタ値に対応した "L" 幅 / "H" 幅をもつパルス出力となります。

動作開始 / 再スタートは、レジスタのビット書込みによります。

リロード動作とパルス出力の関係を下表に示します。

リロード動作	端子出力変化
PRLH → PCNT	PPG0/1[0 → 1] 立上り

また、PPGC0 レジスタのビット 4 (PIE0) が "1" のとき、および PPGC1 レジスタのビット 12 (PIE1) が "1" のときは、それぞれのカウンタの 00 → FF 変化時のボロー (16 ビット PPG モードの場合は、0000 → FFFF 変化時のボロー) によって割込み要求が出力されます。

■ 動作モード

8/16 ビット PPG タイマは、2 チャンネル独立モード、8 ビットプリスケアラ +8 ビット PPG モードまたは 16 ビット PPG モードの計 3 種類の動作モードがあります (MB90480/485 シリーズはモードごとに×3 チャンネルあります)。

2 チャンネル独立モードは、8 ビット PPG として 2 チャンネル独立に動作させるモードです。PPG0 端子には、ch.0 の PPG 出力が接続され、PPG1 端子には ch.1 の PPG 出力が接続されます (PPG2 ~ 5 はそれぞれ ch.2 ~ ch.5 に対応)。

8 ビットプリスケアラ +8 ビット PPG モードは、ch.0(ch.2/ch.4) を 8 ビットプリスケアラとして動作させ、ch.1(ch.3/ch.5) を ch.0(ch.2/ch.4) のボロー出力でカウントすることによって、任意周期の 8 ビット PPG 波形を出力できるようにする動作モードです。PPG0(PPG2/4) 端子には、ch.0(ch.2/ch.4) のプリスケアラ出力が接続され、PPG1 端子には、ch.1(ch.3/ch.5) の PPG の出力が接続されます。

16 ビット PPG1 チャンネルモード (MB90480/485 シリーズは×3 チャンネルあります) は、ch.0 と ch.1 を直結させて (ch.2, ch.3 / ch.4, ch.5 直結) 16 ビットの PPG として動作させる動作モードです。PPG0, 1 には、両方ともに 16 ビット PPG 出力が接続されます。

■ PPG 出力動作

8/16 ビット PPG タイマは, ch.0(ch.2/ch.4) の PPG については, PPGC0 レジスタのビット 7(PEN0) を, ch.1(ch.3/ch.5) の PPG については, PPGC1 レジスタのビット 15(PEN1) を "1" にセットすることによって起動され, カウント動作を開始します。カウント動作を開始した後は, PPGC0 レジスタのビット 7(PEN0), または PPGC1 レジスタのビット 15(PEN1) に "0" を書き込むことによってカウント動作が停止します。停止した後, パルス出力は "L" レベルを保持します。

8 ビットプリスケアラ +8 ビット PPG モードのときで, かつ ch.0(ch.2/ch.4) の停止状態では, ch.1(ch.3/ch.5) を動作状態に設定しないでください。

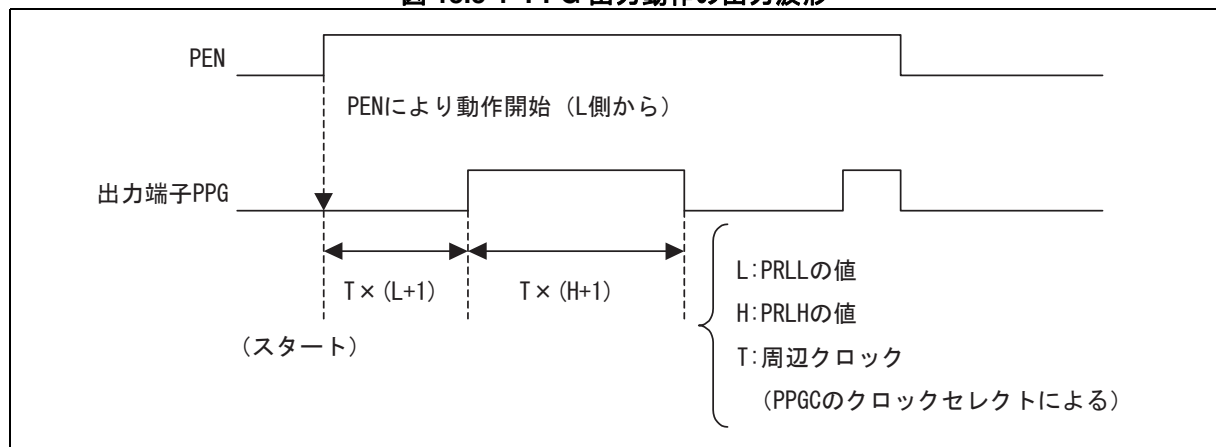
16 ビット PPG モードのときには, PPGC0 レジスタのビット 7(PEN0) と PPGC1 レジスタのビット 15(PEN1) は, 同時に開始 / 停止を制御してください。

以下に, PPG 出力動作について説明します。

PPG 動作時は, 任意周期任意デューティ比 (パルス波の "H" レベル期間と "L" レベル期間の比) のパルス波出力を連続して出力します。PPG は, パルス波出力を開始し, 動作停止が設定されるまで停止しません。

図 15.5-1 に, PPG 出力動作の出力波形を示します。

図 15.5-1 PPG 出力動作の出力波形



■ リロード値とパルス幅の関係

リロードレジスタに書かれた値に +1 した値に, カウントクロックの周期を掛けた値が, 出力されるパルスの幅となります。

8 ビット PPG 動作時のリロードレジスタ値は 00_H のとき, また 16 ビット PPG 動作時のリロードレジスタ値は 0000_H のときには, カウントクロック 1 周期分のパルス幅を持つこととなりますので注意してください。また, 8 ビット PPG 動作時のリロードレジスタ値が FF_H のときは, カウントクロック 256 周期分のパルス幅をもつことになり, 16 ビット PPG 動作時のリロードレジスタ値が $FFFF_H$ のときは, カウントクロック 65536 周期分のパルス幅をもつこととなりますので注意してください。パルス幅の計算式を以下に示します。

$$P_L = T \times (L+1)$$

$$P_H = T \times (H+1)$$

P_L : "L" パルスの幅

P_H : "H" パルスの幅

T: 入力クロック周期

L: PRLl 値

H: PRLH 値

■ カウントクロックの選択

8/16 ビット PPG タイマの動作に使用するカウントクロックは、周辺クロックおよびタイムベースカウンタの入力を使用しています。6 種類のカウントクロック入力から選択できます。

PPG01/23/45 レジスタの bit4 ~ bit2(PCM2 ~ PCM0) で ch.0(ch.2/ch.4) のクロックを、PPG01/23/45 レジスタの bit7 ~ bit5(PCS2 ~ PCS0) で ch.1(ch.3/ch.5) のクロックを選択します。

クロックの選択は、マシクロックから (マシクロック 1/16 ~ 1 倍) またはタイムベースカウンタから選択できます。

<注意事項>

- 8 ビットプリスケアラ +8 ビット PPG モードおよび 16 ビット PPG モードのときは、PPGC1 レジスタの bit14(PCS1) の値は無効となります。
 - タイムベースタイマの入力を使用した場合、トリガ、または停止後の最初のカウントサイクルがずれる可能性があります。また、8/16 ビット PPG タイマが動作中にタイムベースカウンタの初期化を行うとサイクルがずれることがあります。
 - 8 ビットプリスケアラ +8 ビット PPG モードにおいて、ch.0/ch.2/ch.4 が動作状態で ch.1/ch.3/ch.5 が停止状態のときに ch.1(ch.3/ch.5) を起動すると、最初のカウントサイクルがずれる可能性があります。
-

■ パルスの端子出力制御

8/16 ビット PPG タイマによって生成されたパルスは、外部端子 (PPG0～5) から出力します。外部端子からパルスを出すには、各端子に対応するビットに "1" を書き込みます。PPG0/2/4 端子に対しては PPGC0 の bit5(PE0) を、PPG1/3/5 に対しては PPGC1 の bit3(PE1) を使用します。このビットに "0" が書き込まれているとき (初期値) は、パルスは外部端子から出力されずに、汎用ポートとして機能します。

16 ビット PPG モードでは、PPG0～5 からは同じ波形が出力されるため、どちらの外部端子を有効にしても同じ出力を得られます。

8 ビットプリスケアラ+8 ビット PPG モードでは、PPG0/2/4 からは 8 ビットプリスケアラのトグル波形が出力され、PPG1/3/5 からは 8 ビット PPG の波形が出力されます。

このモードにおける出力波形を図 15.5-2 に示します。

図 15.5-2 8 ビットプリスケアラ+8 ビット PPG モード出力動作の波形

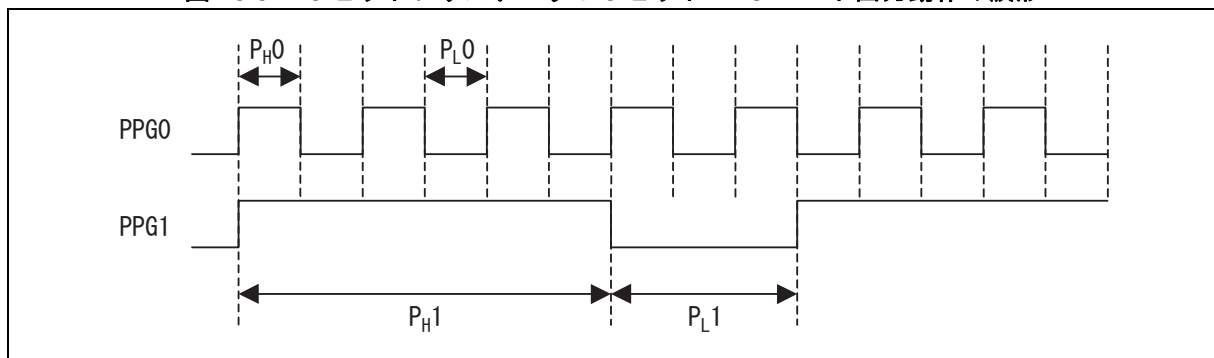


図 15.5-2 におけるパルス幅は、以下の式によって計算することができます。

$$P_{L0} = T \times (L0+1)$$

$$P_{H0} = T \times (L0+1)$$

$$P_{L1} = T \times (L0+1) \times (L1+1)$$

$$P_{H1} = T \times (L0+1) \times (H1+1)$$

L0: ch.0 の PRLl の値および ch.1 の PRLH の値

L1: ch.1 の PRLl の値

H1: ch.1 の PRLH の値

T: 入力クロックサイクル

P_{H0}: PPG0 の "H" パルスの幅

P_{L0}: PPG0 の "L" パルスの幅

P_{H1}: PPG1 の "H" パルスの幅

P_{L1}: PPG1 の "L" パルスの幅

<注意事項>

ch.0 の PRLl と ch.1 の PRLH には同じ値を設定してください。

■ 8/16 ビット PPG タイマの割込み

8/16 ビット PPG タイマの割込みは、リロード値がカウントアウトしてボローが発生したときにアクティブになります。8 ビット PPG2 チャネルモード、または 8 ビットプリスケアラ +9 ビット PPG モード (MB90480/485 シリーズにはそれぞれ×3 チャネル分あります) のときは、それぞれのボローによってそれぞれの割込み要求が行われます。しかし、16 ビット PPG モードでは、16 ビットのカウンタのボローによって PUG0 と PUG1 が同時に設定されます。このため、割込み要因を 1 本化するために、PIE0、または PIE1 のどちらか一方のみを許可します。また、割込み要因のクリアも PUF0 と PUF1 により同時に行います。

■ ハードウェアコンポーネントの初期値

8/16 ビット PPG タイマのハードウェアコンポーネントは、リセット時に次の値に初期化されます。

<レジスタ>	PPG0	→	0X000001
	PPG1	→	00000001
	PPG01	→	XXXXXXX00
<パルス出力>	PPG0	→	"L"
	PPG1	→	"L"
	PE0	→	PPG0 出力禁止
	PE1	→	PPG1 出力禁止
<割込み要求>	IRQ0	→	"L"
	IRQ1	→	"L"

上記以外のハードウェアコンポーネントは初期化されません。

■ リロードレジスタへの書込みタイミング

16 ビット PPG モード以外のモードでは、リロードレジスタ PRL_L, PRL_H へのデータ書込みはワード転送命令を使用することを推奨します。データ項目をバイト転送命令 2 回でレジスタに書き込んだ場合は、タイミングによっては予想しないパルス幅の出力が発生する可能性があります。

図 15.5-3 に、リロードレジスタへの書込みタイミングを示します。

図 15.5-3 リロードレジスタへの書込みタイミングチャート

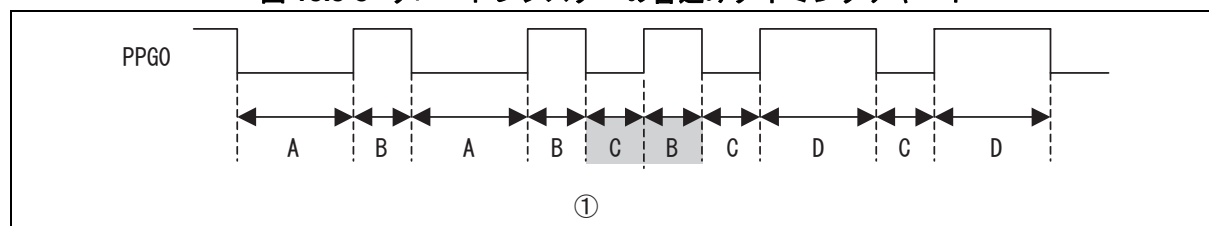
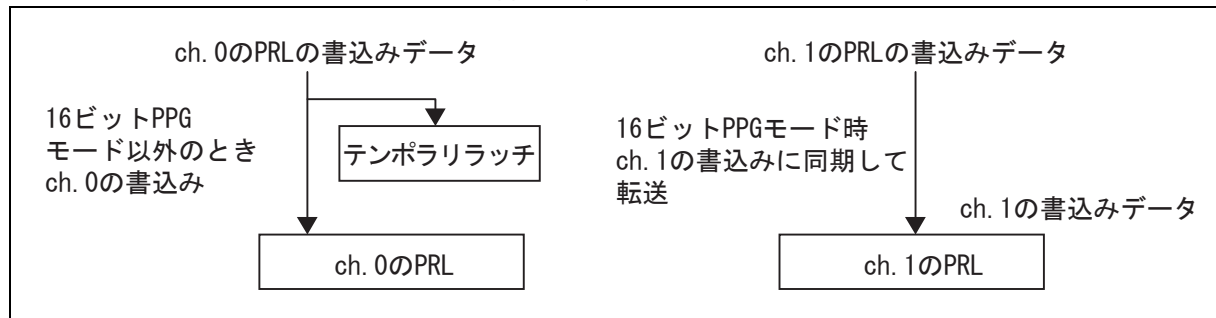


図 15.5-3 において、のタイミングの前で PRL を A から C に書き換えて、①のあとに PRLH の値を B から D に書き換えたとすると、①のタイミングにおける PRL の値は、 $PRL = C$ 、 $PRLH = B$ であるため、1 回だけ "L" 側のカウンタ数 C、"H" 側のカウンタ数 B のパルスが発生します。同様に 16 ビット PPG モードで ch.0/ch.2/ch.4 と ch.1/ch.3/ch.5 の PRL にデータを書き込むには、ロングワード転送命令を使用するか、または ch.0 → ch.1(ch.2 → ch.3 / ch.4 → ch.5) の PRL に順にワード転送命令を使用します。このモードのときは、データは、ch.0/ch.2/ch.4 から PRL へ一時的に書き込まれ、その後 ch.1/ch.3/ch.5 の PRL に書き込んだ時点で、実際に ch.0 の PRL へ書き込まれます。

16ビット PPG モード以外では、ch.0/ch.2/ch.4 と ch.1/ch.3/ch.5 の書込みは独立に行われます。

図 15.5-4 に、PRL 書き込み動作のブロックダイアグラムを示します。

図 15.5-4 PRL 書き込み動作のブロックダイアグラム



15.6 8/16 ビット PPG タイマのプログラム例

8/16 ビット PPG タイマのプログラム例を示します。

■ 8/16 ビット PPG タイマのプログラム例

<p>設定手順例</p> <p>インターバル割込みを発生させ PPG 出力させる。 (PPG1 端子から PPG 出力, ソフトトリガ)</p> <p><初期設定></p> <ul style="list-style-type: none"> PPG01 の制御 <table border="1"> <thead> <tr> <th></th> <th>レジスタ名 . ビット名</th> </tr> </thead> <tbody> <tr> <td>制御レジスタの設定 カウントクロック選択 >></td> <td>PPG01 .PCS2-0</td> </tr> <tr> <td>制御レジスタの設定 パルス出力許可 >></td> <td>PPGC01 .PE10</td> </tr> <tr> <td>割込み許可 >></td> <td>.PIE1</td> </tr> <tr> <td>動作モード選択 >></td> <td>.MD1-0</td> </tr> </tbody> </table> <ul style="list-style-type: none"> デューティ設定 <table border="1"> <tbody> <tr> <td>PPG01 のデューティ設定</td> <td>PRL0</td> </tr> </tbody> </table> <ul style="list-style-type: none"> 割込み関連 <table border="1"> <tbody> <tr> <td>PPG01 割込みレベルの設定</td> <td>ICR05</td> </tr> <tr> <td>I フラグの設定</td> <td>(CCR)</td> </tr> </tbody> </table> <p><起動></p> <ul style="list-style-type: none"> PPG01 の起動 <table border="1"> <thead> <tr> <th></th> <th>レジスタ名 . ビット名</th> </tr> </thead> <tbody> <tr> <td>割込み許可</td> <td>PPGC01.PIE1</td> </tr> <tr> <td>PPG01 起動</td> <td>PPGC01.PEN1</td> </tr> </tbody> </table> <p><割込み></p> <ul style="list-style-type: none"> 割込み処理 <table border="1"> <tbody> <tr> <td>任意の処理</td> <td></td> </tr> <tr> <td>割込み要求フラグクリア</td> <td>PPGC01.PUF1</td> </tr> </tbody> </table> <p><割込みベクタ></p> <ul style="list-style-type: none"> ベクタテーブルの設定 <p><その他></p> <p>(注意事項) 事前にクロック関連の設定および、__set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>		レジスタ名 . ビット名	制御レジスタの設定 カウントクロック選択 >>	PPG01 .PCS2-0	制御レジスタの設定 パルス出力許可 >>	PPGC01 .PE10	割込み許可 >>	.PIE1	動作モード選択 >>	.MD1-0	PPG01 のデューティ設定	PRL0	PPG01 割込みレベルの設定	ICR05	I フラグの設定	(CCR)		レジスタ名 . ビット名	割込み許可	PPGC01.PIE1	PPG01 起動	PPGC01.PEN1	任意の処理		割込み要求フラグクリア	PPGC01.PUF1	<p>プログラム例</p> <pre> void PPG_sample(void) { PPG01_initial(); PPG01_start(); } void PPG01_initial(void) { /* *1 IO_PPG01.bit.PCS = 1; /* bit7-5 = 0 PPG1 カウントクロック選択 */ /* *2 IO_PPGC01.word = 0x3101; /* bit15 = 0 PPG 動作停止 */ /* bit13 = 1 PPG1/3/5 パルス出力許可 */ /* bit12 = 1 PPG 割込み許可 */ /* bit11 = 0 PUF1 割込み要求フラグクリア */ /* bit10-9 = 00 8 ビット PPG 2ch */ /* bit8 = 1 予約ビット */ /* bit0 = 1 予約ビット */ /* *3 IO_PRL0.word = 0x7f7f; /* PPGduty 比 設定 */ IO_ICR05.byte = 0x10; /* 割込みレベル (値は任意) */ __EI(); /* 割込み許可 */ } void PPG01_start(void) { IO_PPGC01.bit.PIE1 = 1; /* bit12 = 1 PIE1 割込み要求許可 */ IO_PPGC01.bit.PEN1 = 1; /* bit15 = 1 PEN1 PPG 動作開始 */ } __interrupt void PPG01_int(void) { /* 任意の処理 */ IO_PPGC01.bit.PUF1 = 0; /* bit11 = 0 PUF1 割込み要求フラグ */ } #pragma intvect PPG01_int 22 </pre> <p>(注意事項) レジスタの記述形式については、「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
	レジスタ名 . ビット名																										
制御レジスタの設定 カウントクロック選択 >>	PPG01 .PCS2-0																										
制御レジスタの設定 パルス出力許可 >>	PPGC01 .PE10																										
割込み許可 >>	.PIE1																										
動作モード選択 >>	.MD1-0																										
PPG01 のデューティ設定	PRL0																										
PPG01 割込みレベルの設定	ICR05																										
I フラグの設定	(CCR)																										
	レジスタ名 . ビット名																										
割込み許可	PPGC01.PIE1																										
PPG01 起動	PPGC01.PEN1																										
任意の処理																											
割込み要求フラグクリア	PPGC01.PUF1																										

*1: io_PPG01 は PPG1 レジスタ + PPG0 レジスタを表します。

*2: io_PPGC01 は PPGC1 レジスタ + PPGC0 レジスタを表します。

*3: io_PRL0 は PRL0 レジスタ + PRLH0 レジスタを表します。

■ プログラム例以外の設定方法

● PPG 動作を許可 / 停止する方法

PPG 動作許可ビット (PPG01/PPG23/PPG45.PEN0or1) で設定します。

制御	PPG 動作許可ビット (PEN0or1)
PPG 動作を停止させるには	"0" にする
PPG 動作を許可するには	"1" にする

PPG 動作許可は PPG を起動する前に行ってください。

● PPG の動作モードを設定する方法

モード選択ビット (PPGC01.MD[1:0]/PPGC23.MD[1:0]/PPGC45.MD[1:0]) で設定します。

● カウントクロックの種類と選択方法

ch.1, ch.3, ch.5 は, カウントクロック選択ビット (PPG01.PCS[2:0]/PPG23.PCS[2:0]/PPG45.PCS[2:0]) で選択できます。

ch.0, ch.2, ch.4 は, カウントクロック選択ビット (PPG01.PCM[2:0]/PPG23.PCM[2:0]/PPG45.PCM[2:0]) で選択できます。

● 割込み関連レジスタ

チャンネル, 割込みレベル, ベクタの関係は下表のとおりです。

割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

チャンネル	割込みベクタ	割込みレベル設定レジスタ
PPG0	#22 アドレス : FFFFA4 _H	割込み制御レジスタ 05 (ICR05) アドレス : 0000B5 _H
PPG1		
PPG2	#23 アドレス : FFFFA0 _H	割込み制御レジスタ 06 (ICR06) アドレス : 0000B6 _H
PPG3		
PPG4	#24 アドレス : FFFF9C _H	割込み制御レジスタ 06 (ICR06) アドレス : 0000B6 _H
PPG5		

割込み要求フラグ (PPG01/PPG23/PPG45.PUF0 or 1) は自動的にクリアしませんので, 割込み処理から復帰する前にソフトウェアにてクリアしてください (PUF0 or 1 ビットに "0" を書き込む)。

● 割込みの種類

割込みは 1 種類です。PPG カウンタのアンダフローで発生します。

● 割込みを許可 / 禁止 / クリアする方法

割込みの許可 / 禁止は、割込み要求許可ビット (PPG01/PPG23/PPG45.PIE0or1) で設定します。

制御内容	割込み要求許可ビット (PIE0or1)
割込み要求を禁止するには	"0" にする
割込み要求を許可するには	"1" にする

割込み要求のクリアは、割込み要求ビット (PPG01/PPG23/PPG45.PUF0or1) で設定します。

制御内容	割込要求ビット (PUF0or1)
割込み要求をクリアするには	"0" を書き込む

第16章

DTP/ 外部割込み

DTP/ 外部割込みの概要，レジスタの構成と機能，動作および使用上の注意について説明します。

16.1 DTP/ 外部割込みの概要

16.2 DTP/ 外部割込みのレジスタの構成と機能

16.3 DTP/ 外部割込み

16.4 DTP/ 外部割込みの動作

16.5 DTP/ 外部割込み使用上の注意

16.6 DTP/ 外部割込みのプログラム例

16.1 DTP/ 外部割込みの概要

DTP(Data Transfer Peripheral) は、デバイス外部に存在する周辺装置と F²MC-16LX CPU との間において、外部周辺装置が発生する DMA 要求または割込み要求を受け取り、これを F²MC-16LX CPU に伝えて μ DMAC もしくは EI²OS または割込み処理を起動させるための周辺制御部です。

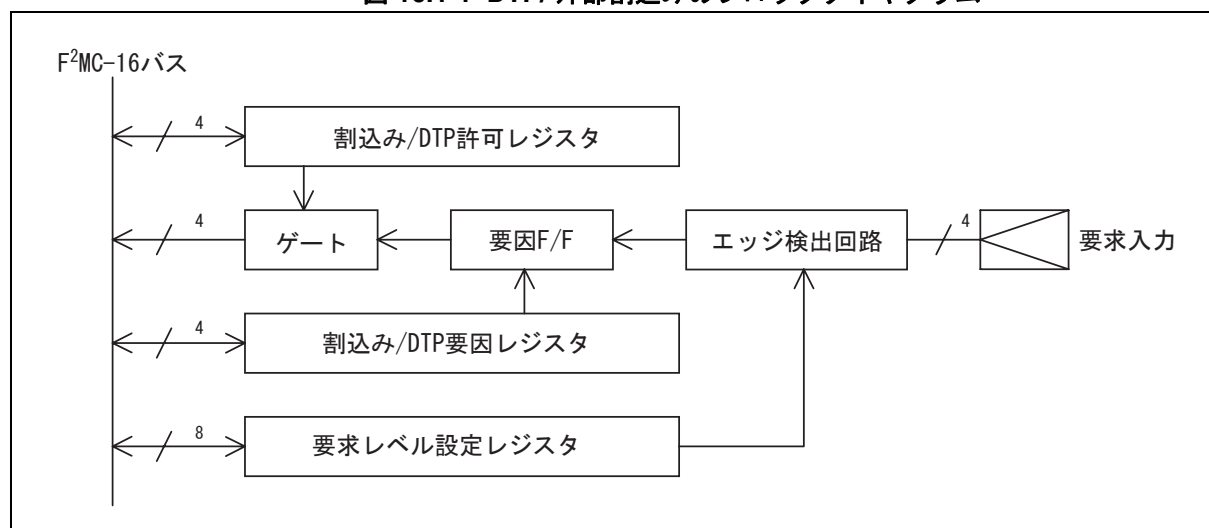
■ DTP/ 外部割込みの概要

要求レベルとして、 μ DMAC もしくは EI²OS の場合は "H" と "L" の 2 種類が、外部割込み要求の場合は "H", "L" のほか、立上りエッジと立下りエッジの計 4 種類が選択できます。

■ DTP/ 外部割込みのブロックダイアグラム

図 16.1-1 に、DTP/ 外部割込みのブロックダイアグラムを示します。

図 16.1-1 DTP/ 外部割込みのブロックダイアグラム



■ DTP/ 外部割込みに関する端子

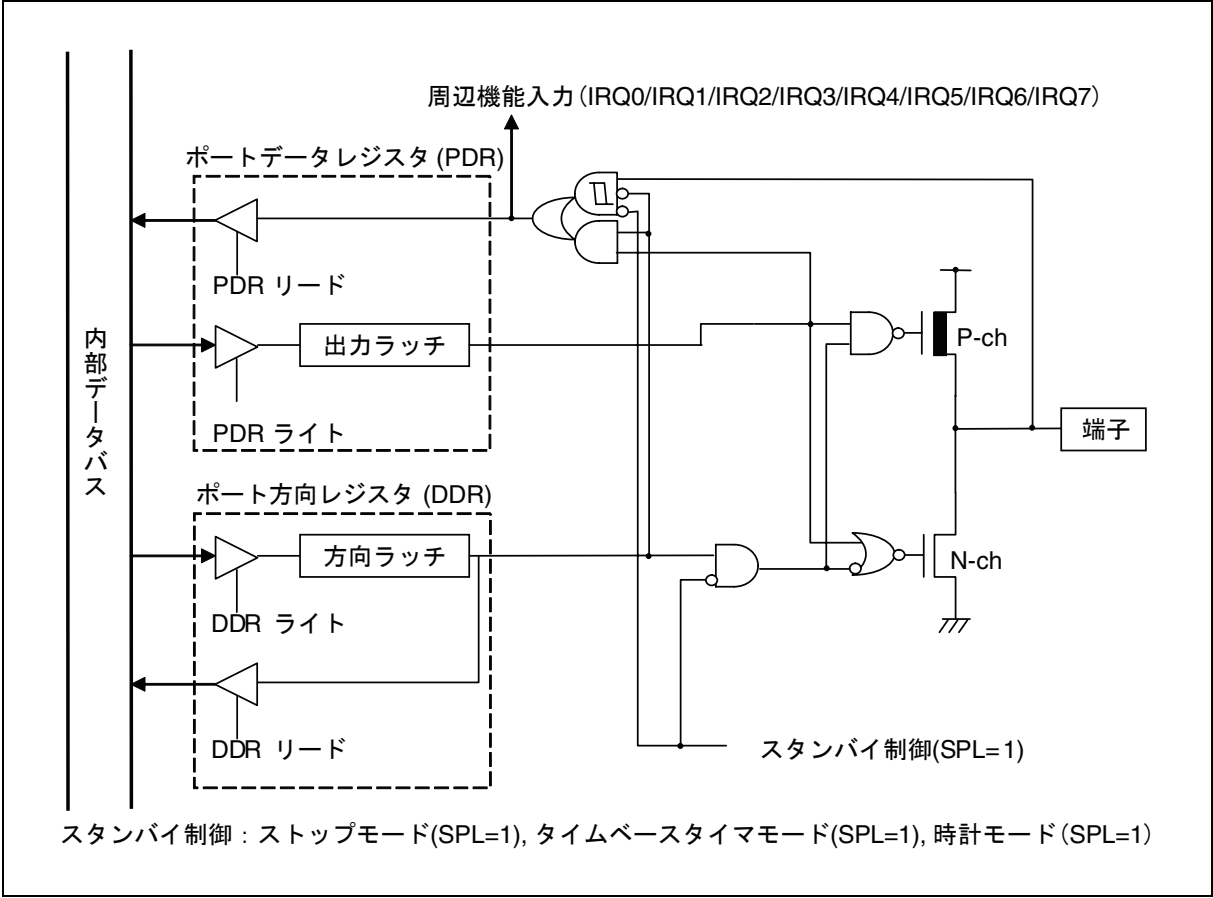
外部割込み端子に関する端子は、IRQ0/IRQ1/IRQ2/IRQ3/IRQ4/IRQ5/IRQ6/IRQ7端子があり入力ポートとして機能します。IRQ0/IRQ1/IRQ2/IRQ3/IRQ4/IRQ5/IRQ6/IRQ7 端子は、汎用入出力ポート (P80/IRQ0, P81/IRQ1, P82/IRQ2, P83/IRQ3, P84/IRQ4, P85/IRQ5, P86/IRQ6, P87/IRQ7) と外部割込み入力端子として機能を兼用しています。

● IRQ0/IRQ1/IRQ2/IRQ3/IRQ4/IRQ5/IRQ6/IRQ7 端子として使用する場合の設定

P80/IRQ0, P81/IRQ1, P82/IRQ2, P83/IRQ3, P84/IRQ4, P85/IRQ5, P86/IRQ6, P87/IRQ7 端子を入力端子として使う場合は、ポート方向レジスタを入力ポート (DDR8 bit0 ~ bit7 → "0") に設定してください。

■ DTP/ 外部割込みに関する端子のブロックダイヤグラム

図 16.1-2 DTP/ 外部割込みに関する端子のブロックダイヤグラム



16.2 DTP/ 外部割込みのレジスタの構成と機能

DTP/ 外部割込みに関するレジスタの構成と機能について説明します。

■ DTP/ 外部割込みのレジスタ

図 16.2-1 に、DTP/ 外部割込みのレジスタ一覧を示します。

図 16.2-1 DTP/ 外部割込みのレジスタ一覧

ビット	7	6	5	4	3	2	1	0	
アドレス:00000C _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	割込み/DTP許可レジスタ (ENIR)
ビット	15	14	13	12	11	10	9	8	
アドレス:00000D _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	割込み/DTP要因レジスタ (EIRR)
ビット	7	6	5	4	3	2	1	0	
アドレス:00000E _H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	要求レベル設定レジスタ (ELVR)
ビット	15	14	13	12	11	10	9	8	
アドレス:00000F _H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	要求レベル設定レジスタ (ELVR)

■ 割込み /DTP 許可レジスタ (ENIR:ENable Interrupt Request Register)

下図に、割込み /DTP 許可レジスタ (ENIR) のビット構成を示します。

ENIR	7	6	5	4	3	2	1	0	初期値
アドレス:00000C _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

割込み /DTP 許可レジスタ (ENIR) は、外部割込み /DTP チャンネルに対して外部割込み /DTP 要求を許可または禁止します。

割込み /DTP 許可ビット (ENIR:EN) に "1" が設定され、かつ、割込み /DTP 要求フラグビット (EIRR:EN) に "1" がセットされた場合、対応する割込み /DTP 端子の割込み要求が発生します。スタンバイモード時は入力を遮断しません。

<注意事項>

DTP/ 外部割込みを許可 (ENIR: EN=1) する直前に対応する DTP/ 外部割込み要因ビット (EIRR: ER) をクリアしてください。

■ 割込み /DTP 要因レジスタ (EIRR: External Interrupt Request Register)

下図に、割込み /DTP 要因レジスタ (EIRR) のビット構成を示します。

EIRR	15	14	13	12	11	10	9	8	初期値
アドレス: 00000D _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

割込み /DTP 要因レジスタ (EIRR) は、外部割込み端子に、検出レベル設定レジスタの検出条件選択ビット (ELVR: LB, LA) 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" にセットされます。

"1" がセットされた場合：割込み /DTP 要求許可ビット (ENIR: EN) を "1" に設定している場合は、対応する割込み /DTP チャンネルの割込み要求が発生します。

"0" に設定した場合：本レジスタはクリアされます。

"1" に設定した場合：影響しません。

<注意事項>

- ・ リードモディファイライト系命令の読出し時には "1" が読み出されます。複数の外部割込み要求出力が許可 (ENIR: EN7 ~ EN0=1) されている場合、CPU が割込みを受け付けたビット (EN7 ~ EN0 の "1" にセットされているビット) だけを "0" にクリアするようにしてください。それ以外のビットを無条件にクリアすることは避けてください。
- ・ DTP/ 外部割込み要因ビット (EIRR: ER) の値は、対応する DTP/ 外部割込み許可ビット (ENIR: EN) が "1" に設定されている時のみ有効です。
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/ 外部割込み要因ビット (EIRR: ER) をクリアしてください。

■ 要求レベル設定レジスタ (ELVR: External Level Register)

下図に、要求レベル設定レジスタ (ELVR) のビット構成を示します。

	7	6	5	4	3	2	1	0	初期値
アドレス: 00000E _H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	15	14	13	12	11	10	9	8	初期値
アドレス: 00000F _H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

要求レベル設定レジスタ (ELVR) は、要求検出レベルを選択するレジスタです。1 つの端子あたり 2 ビットが割り当てられていて、表 16.2-1 に示す対応となっています。要求入力がレベルの場合、クリアしても入力がアクティブならば再びセットされます。

表 16.2-1 ELVR 割当て (LA0 ~ LA7, LB0 ~ LB7)

LBx	LAx	動 作
0	0	"L" レベルで要求あり
0	1	"H" レベルで要求あり
1	0	立上りエッジで要求あり
1	1	立下りエッジで要求あり

16.3 DTP/ 外部割込み

DTP/ 外部割込みに関する割込みは、入力端子へのエッジ入力もしくはレベル入力を検出した場合に発生します。

DTP/ 外部割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができます。

■ DTP/ 外部割込み

DTP/ 外部割込み制御ビットと割込み要因を下記の表に示します。

	外部割込み ICR の ISE=0 の場合	DTP 割込み ICR の ISE=1 の場合
割込み要求フラグ	EIRR:ER (bit8 ~ bit15)	EIRR:ER (bit8 ~ bit15)
割込み要求出力許可ビット	ENIR:EN (bit0 ~ bit7)	ENIR:EN (bit0 ~ bit7)
割込み発生要因	外部割込み検出	外部割込み検出

設定の手順

DTP/ 外部割込みを使用する場合は、次の手順で各レジスタの設定を行ってください。

1. 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を "0" に設定します (ENIR:EN)。
2. 使用する DTP/外部割込みチャンネルに対応する検出条件選択ビットで、検出するエッジまたは、レベルを設定します (ELVR:LA/LB)。
3. 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を "0" に設定します (EIRR:ER)。
4. 対応する割込み要求許可ビットを "1" に設定します (ENIR:EN)。

<注意事項>

- DTP/ 外部割込みのレジスタ設定は、外部割込み要求を禁止に (ENIR:EN=0) に設定してください。
- DTP/ 外部割込みを許可 (ENIR:EN=1) にする場合は、先に対応する DTP/ 外部割込み要求フラグビットをクリア (EIRR:ER=0) してください。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の EI²OS 許可ビット (ICR:ISE) の設定もしくは、DMA イネーブルレジスタ (DER:EN) の設定によって決定されます。

ISE ビットを "1" に設定した場合は、拡張インテリジェントサービス (EI²OS) が、EN ビットを "1" に設定した場合は、DMA 転送が許可されます。

ISE ビットと EN ビットを "0" に設定した場合は、EI²OS および DMA 転送は禁止され、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り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り当てられている場合、そのうちの 1 つで EI²OS を起動しようとすると、他の割込み要求は使用できません。

■ DTP/ 外部割込みと DMA 転送および EI²OS

表 16.3-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 16.3-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
INT0 (IRQ0)	○	0	#11	FFFFD0 _H	ICR00	0000B0 _H
INT1 (IRQ1)	○	×	#12	FFFFCC _H		
INT2 (IRQ2)	○	×	#13	FFFFC8 _H	ICR01	0000B1 _H
INT3 (IRQ3)	○	×	#14	FFFFC4 _H		
INT4 (IRQ4)	○	×	#15	FFFFC0 _H	ICR02	0000B2 _H
INT5 (IRQ5)	○	×	#16	FFFFBC _H		
INT6 (IRQ6)	○	×	#17	FFFFB8 _H	ICR03	0000B3 _H
INT7 (IRQ7)	○	×	#18	FFFFB4 _H		

×：割込み要求フラグはクリアされません。

○：割込み要求フラグはクリアされます。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

DTP/ 外部割込みは EI²OS 機能と ch.0 のみが DMA 転送機能に対応しています。DMA または EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有するほかの割込みを禁止に設定する必要があります。

16.4 DTP/ 外部割込みの動作

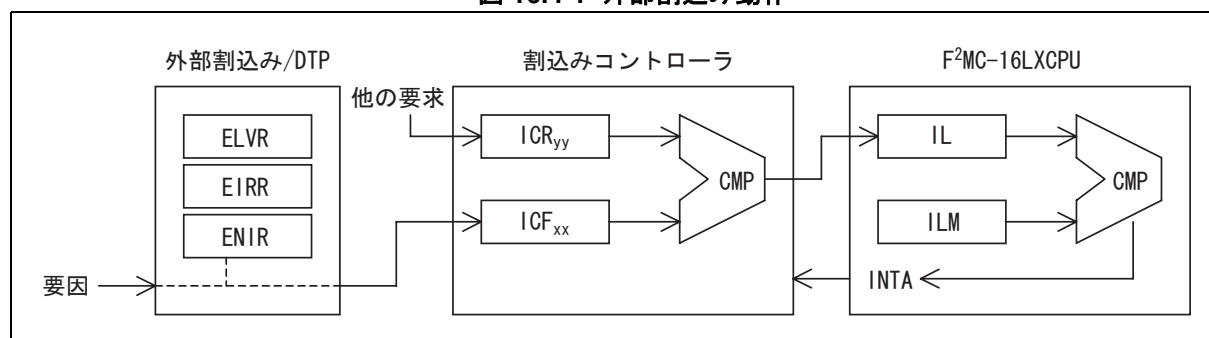
DTP/ 外部割込みの動作について説明します。

■ 外部割込みの動作

外部割込み要求の設定の後，対応する端子に ELVR レジスタで設定された要求が入力されると，本リソースは割込みコントローラに対して割込み要求信号を発生します。割込みコントローラ内で同時発生した割込みの優先順位を識別した結果，本リソースからの割込みが最も優先順位が高かったときに，割込みコントローラは F²MC-16LX CPU に対して割込み要求を発生します。F²MC-16LX 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レジスタ (ILM) と割込み要求を比較します。要求レベルが ILM ビットより高かったときに現在実行中の命令が終了し次第，ハードウェア割込み処理マイクロプログラムを起動します。

図 16.4-1 に，外部割込み動作のフローを示します。

図 16.4-1 外部割込み動作



割込み処理マイクロプログラムでは割込みベクタ領域の読出しと，割込みコントローラへの割込みアクリッジを発生し，ベクタから生成したマクロ命令のジャンプ先アドレスをプログラムカウンタへ転送した上で，ユーザの割込み処理プログラムを実行します。

■ DTP の動作

μDMAC 初期化のためユーザプログラム内では μDMAC を起動するにあたって、μDMAC ディスクリプタ内の I/O アドレスポインタに 000000_H から 0000FF_H に割り付けられているレジスタのアドレスを設定し、バッファアドレスポインタにメモリバッファの先頭アドレスを設定します。

DTP の動作シーケンスは外部割込みの場合と同じです。μDMAC が起動されると、アドレスリングされている外部周辺装置に読出し信号、または書込み信号が送られ、本チップとの間で転送が行われます。外部周辺装置側では、その転送が行われてから 3 マシンサイクル以内に本チップに対する割込み要求を取り下げする必要があります。転送が終了するとディスクリプタの更新などが行われ、その後割込み要因をクリアする信号を割込みコントローラに発生させます。割込み要因をクリアする信号を受け取った本リソースは、要因を保持しているフリップフロップをクリアして、端子からの次の要求に備えます。

図 16.4-2 に、DTP 動作終了時の外部割込み要求を取り下げるタイミングを示します。

また、図 16.4-3 に、外部周辺装置とのインタフェース例を示します。

図 16.4-2 DTP 動作終了時の外部割込み要求の取下げタイミング

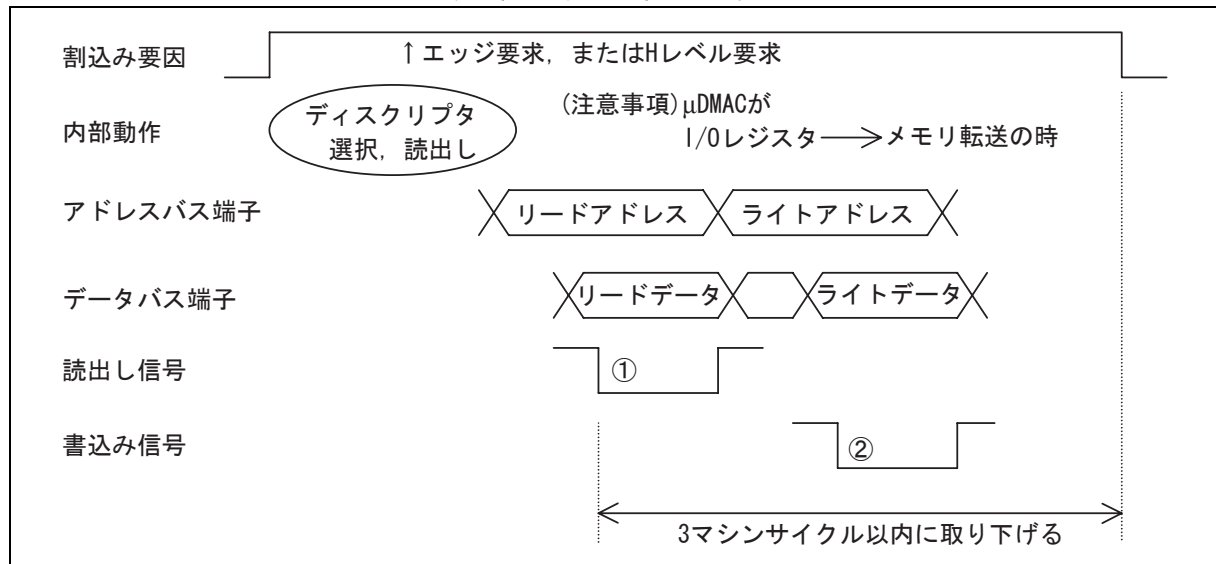
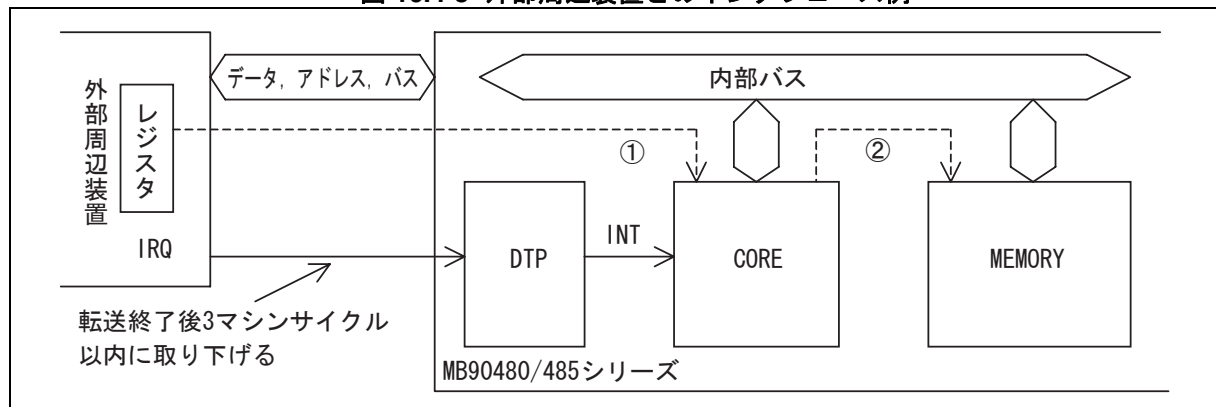


図 16.4-3 外部周辺装置とのインタフェース例



16.5 DTP/ 外部割込み使用上の注意

DTP/ 外部割込みを使用する上での注意事項について説明します。

■ 外部に接続する周辺装置の条件

DTP がサポートできる外部周辺装置は、転送が行われたことにより自動的に要求をクリアできるものでなければなりません。また、転送動作を開始してから 3 マシンサイクル以内で転送要求を取り下げようになっていないと、本リソースは次の転送要求が発生したものとして処理を行います。

■ DTP/ 外部割込みの動作手順

DTP/ 外部割込み内に存在するレジスタの設定は、次の手順で行ってください。

- (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。
- (2) 許可レジスタ (ENIR) の対象となるビットを禁止状態にする。
- (3) 要求レベル設定レジスタ (ELVR) の対象となるビットを設定する。
- (4) 要因レジスタ (EIRR) の対象となるビットをクリアする。
- (5) 許可レジスタ (ENIR) の対象となるビットを許可状態にする。

(4) と (5) はワード指定による同時書込みが可能です。

本リソース内のレジスタを設定するときには、必ず許可レジスタを禁止状態に設定しておかなくてはなりません。また、許可レジスタを許可状態にする前に必ず要因レジスタをクリアしておく必要があります。これは、レジスタ設定時や割込み許可状態時に誤って割込み要因が発生してしまうことを避けるためです。

■ 外部割込み要求レベル

- 要求レベルがエッジ要求の場合、エッジがあったことを検出するためには、パルス幅は最小 3 マシンサイクル必要です。
- 要求入力レベルがレベル設定の場合、外部から要求入力が入ってその後取り下げられても、内部に要因保持回路が存在するので、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り下げるには、要因保持回路をクリアする必要があります。

図 16.5-1 レベル設定時の要因保持回路のクリア

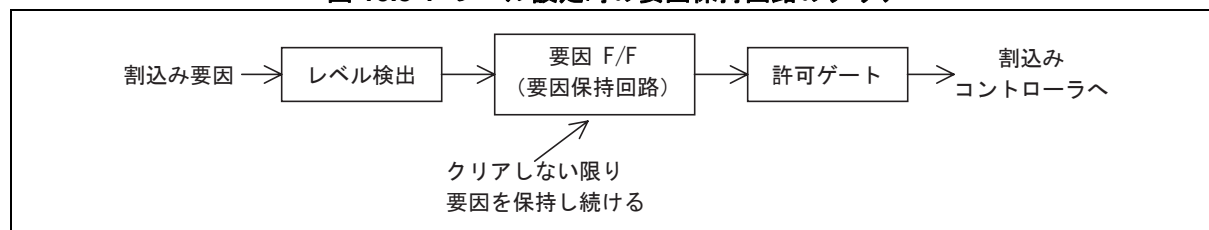
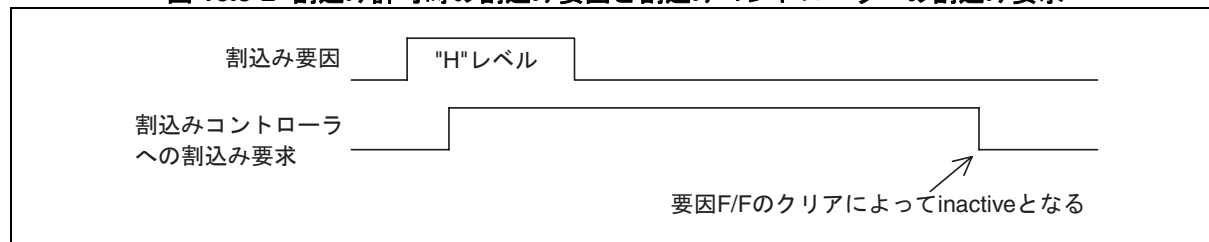


図 16.5-2 割込み許可時の割込み要因と割込みコントローラへの割込み要求



16.6 DTP/ 外部割込みのプログラム例

DTP/ 外部割込みのプログラム例を示します。

■ DTP/ 外部割込みのプログラム例

<p>設定手順例</p> <p>INT0 より入力した信号の立上りエッジで外部割込みを発生させる。</p> <p>< 初期設定 ></p> <ul style="list-style-type: none">・ポート レジスタ名、ビット名 <table><tr><td>INT0 ポートの入力選択</td><td>DDR8 .P80</td></tr></table> <ul style="list-style-type: none">・ INT0 制御 レジスタ名、ビット名 <table><tr><td>外部割込み検出の選択</td><td>ELVR</td></tr><tr><td></td><td>LB7, LA7-LB1, LA1 LB0, LA0</td></tr></table> <ul style="list-style-type: none">・ 割込み関連 1 <table><tr><td>INT0 割込みレベルの設定</td><td>ICR00</td></tr><tr><td>I フラグの設定</td><td>(CCR)</td></tr></table> <ul style="list-style-type: none">・ 割込み関連 2 <table><tr><td>INT0 割込み要因</td><td>EIRR. ER0</td></tr><tr><td>INT0 割込み許可</td><td>ENIR. EN0</td></tr></table> <p>< 割込み ></p> <ul style="list-style-type: none">・ 変換値の読出し レジスタ名、ビット名 <table><tr><td>割込み要求フラグのクリア</td><td>EIRR. ER0</td></tr></table> <ul style="list-style-type: none">・ 任意の処理 <p>< 割込みベクタ ></p> <ul style="list-style-type: none">・ ベクタテーブルの設定 <p>(注意事項)</p> <p>事前にクロック関連の設定および、__set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	INT0 ポートの入力選択	DDR8 .P80	外部割込み検出の選択	ELVR		LB7, LA7-LB1, LA1 LB0, LA0	INT0 割込みレベルの設定	ICR00	I フラグの設定	(CCR)	INT0 割込み要因	EIRR. ER0	INT0 割込み許可	ENIR. EN0	割込み要求フラグのクリア	EIRR. ER0	<p>プログラム例</p> <pre>void EX_INT_sample_1() { EX_INT0_initial(); } void EX_INT0_initial(void) { IO_DDR8.bit.D80= 0; /* INT0 入力 */ IO_ELVR.word= 0x0001; /* 設定値 : 00000001 (bit) */ /* bit7-2= "000000" */ /* bit1-0= "01" H レベル検出 */ IO_ICR00 = 0x00 /* 値は任意 */ __EI(); /* 割込み許可 */ IO_EIRR.bit.ER0= 0; /* ER0 割込みフラグクリア */ IO_ENIR.bit.EN0= 0; /* EN0 割込み許可 */ } __interrupt void INT0_int(void) /* */ { IO_EIRR.bit.ER0= 0; /* ER0 割込みフラグクリア */ } #pragma intvect INT0_int 11 (注意事項) レジスタの記述形式については、「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</pre>
INT0 ポートの入力選択	DDR8 .P80																
外部割込み検出の選択	ELVR																
	LB7, LA7-LB1, LA1 LB0, LA0																
INT0 割込みレベルの設定	ICR00																
I フラグの設定	(CCR)																
INT0 割込み要因	EIRR. ER0																
INT0 割込み許可	ENIR. EN0																
割込み要求フラグのクリア	EIRR. ER0																

■ プログラム例以外の設定方法

● 検出レベルの種類と設定方法

検出レベルの種類は 4 種類 ("L" レベル , "H" レベル , 立上り , 立下り) です。

検出レベルビット (ELVR. LBx, LAx) x=0 ~ 7 で設定します。

動作モード	検出レベルビット (LBx, LAx) x=0 ~ 7
"L" レベル検出にするには	"00" にする
"H" レベル検出にするには	"01" にする
立上り検出にするには	"10" にする
立下り検出にするには	"11" にする

● IRQ 端子を入力にする方法

ポート 8 方向レジスタ (DDR8) で設定します。

動作	方向ビット (P80 ~ P87)	設定
IRQ0 端子入力にするには	DDR8. P80	"0" にする
IRQ1 端子入力にするには	DDR8. P81	"0" にする
IRQ2 端子入力にするには	DDR8. P82	"0" にする
IRQ3 端子入力にするには	DDR8. P83	"0" にする
IRQ4 端子入力にするには	DDR8. P84	"0" にする
IRQ5 端子入力にするには	DDR8. P85	"0" にする
IRQ6 端子入力にするには	DDR8. P86	"0" にする
IRQ7 端子入力にするには	DDR8. P87	"0" にする

● 割込み関連レジスタ

外部割込み端子, 割込みレベル, ベクタの関係は下表のとおりです。

割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

外部割込み端子	割込みベクタ	割込みレベル設定ビット
IRQ0	#11 アドレス: FFFFD0 _H	割込み制御レジスタ 00 (ICR00) アドレス: 0000B0 _H
IRQ1	#12 アドレス: FFFFCC _H	割込み制御レジスタ 00 (ICR00) アドレス: 0000B0 _H
IRQ2	#13 アドレス: FFFFC8 _H	割込み制御レジスタ 01 (ICR01) アドレス: 0000B1 _H
IRQ3	#14 アドレス: FFFFC4 _H	割込み制御レジスタ 01 (ICR01) アドレス: 0000B1 _H
IRQ4	#15 アドレス: FFFFC0 _H	割込み制御レジスタ 02 (ICR02) アドレス: 0000B2 _H
IRQ5	#16 アドレス: FFFFBC _H	割込み制御レジスタ 02 (ICR02) アドレス: 0000B2 _H
IRQ6	#17 アドレス: FFFF8 _H	割込み制御レジスタ 03 (ICR03) アドレス: 0000B3 _H
IRQ7	#18 アドレス: FFFF4 _H	割込み制御レジスタ 03 (ICR03) アドレス: 0000B3 _H

● 割込みの種類

割込み要因は外部割込みのみです。選択のビットはありません。

● 割込みを許可 / 禁止 / クリアする方法

割込みの許可 / 禁止は、割込み許可ビット (ENIR.ENx x=0 ~ 7) で設定します。

制御内容	割込み許可ビット (ENx x=0 ~ 7)
割込み要求を禁止するには	"0" にする
割込み要求を許可するには	"1" にする

割込み要求のクリアは、割込み要求ビット (EIRR.ERx x=0 ~ 7) で設定します。

制御内容	割込み要求ビット (ERx x=0 ~ 7)
割込み要求をクリアするには	"0" を書き込む

第17章

8/10 ビット A/D コンバータ

8/10 ビット A/D コンバータの概要，レジスタの構成と機能，動作，変換データ保護機能および使用上の注意について説明します。

- 17.1 8/10 ビット A/D コンバータの概要
- 17.2 8/10 ビット A/D コンバータの構成
- 17.3 8/10 ビット A/D コンバータのレジスタの構成と機能
- 17.4 8/10 ビット A/D コンバータの割込み
- 17.5 8/10 ビット A/D コンバータの動作
- 17.6 8/10 ビット A/D コンバータの変換データ保護機能
- 17.7 8/10 ビット A/D コンバータの使用上の注意
- 17.8 8/10 ビット A/D コンバータのプログラム例

17.1 8/10 ビット A/D コンバータの概要

8/10 ビット A/D コンバータの特長およびブロックダイアグラムを示します。

■ 8/10 ビット A/D コンバータの特長

8/10 ビット A/D コンバータには、以下の特長があります。

- 変換時間：
 - 1 チャネルあたり最小 3.68 μ s(92 マシンサイクル / マシンクロック 25MHz 時 / サンプルング時間を含む)
- サンプルング時間：
 - 1 チャネルあたり最小 1.92 μ s(48 マシンサイクル / マシンクロック 25MHz 時)
- サンプル & ホールド回路付き RC 型逐次比較変換方式を採用
- 8 ビットまたは 10 ビットの分解能選択可能
- アナログ入力は 8 チャネルからプログラムで選択可能
 - シングル変換モード : 1 チャネルを選択変換
 - スキャン変換モード : 連続した複数のチャネルを変換。最大 8 チャネルプログラム可能
 - 連続変換モード : 指定チャネルを繰り返し変換
 - 停止変換モード : 1 チャネルを変換したら一時停止して次の起動が掛かるまで待機 (変換開始の同期が可能)

A/D 変換終了時には、CPU に対して A/D 変換終了の割込み要求を発生します。この割込み発生で μ DMAC を起動することで、A/D 変換結果データをメモリに転送できますので連続処理に適しています。

起動要因は、ソフトウェア、外部トリガ (立下りエッジ)、タイマ (立上りエッジ) のいずれかから選択できます。

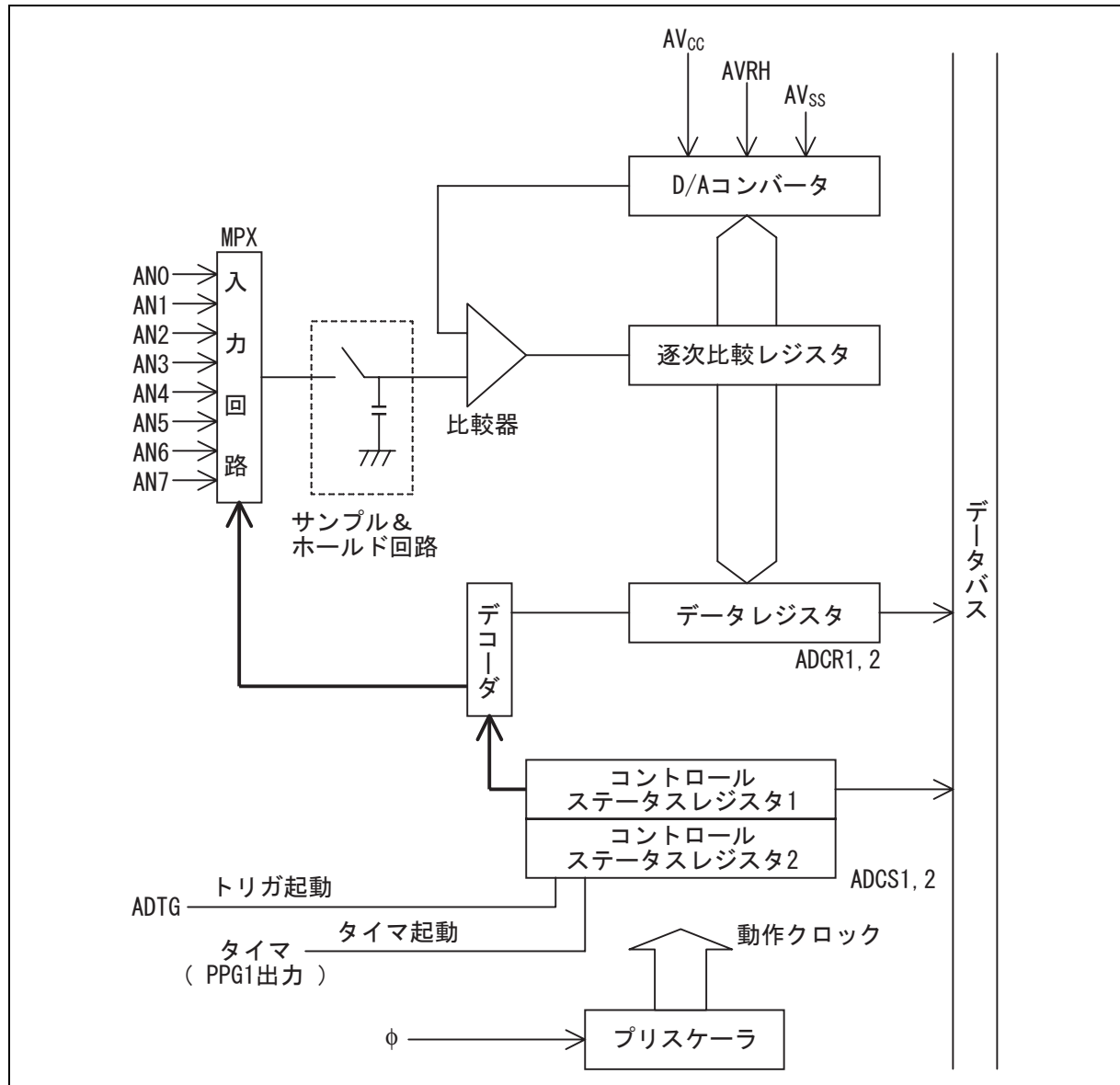
17.2 8/10 ビット A/D コンバータの構成

8/10 ビット A/D コンバータのブロックダイアグラムと構成を示します。

■ 8/10 ビット A/D コンバータのブロックダイアグラム

図 17.2-1 に、8/10 ビット A/D コンバータのブロックダイアグラムを示します。

図 17.2-1 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 コンバータに関する端子

8/10 ビット A/D コンバータに関する端子には、アナログ入力 AN0/AN1/AN2/AN3/AN4/AN5/AN6/AN7 端子と入力トリガ ADTG 端子があります。汎用入出力ポート (P60/AN0, P61/AN1, P62/AN2, P63/AN3, P64/AN4, P65/AN5, P66/AN6, P67/AN7) は、A/D のアナログ入力端子として機能を兼用しており、汎用入出力ポート (P93/ADTG) は、A/D のトリガ入力として機能を兼用しています。

● AN0/AN1/AN2/AN3/AN4/AN5/AN6/AN7 端子として使用する場合の設定

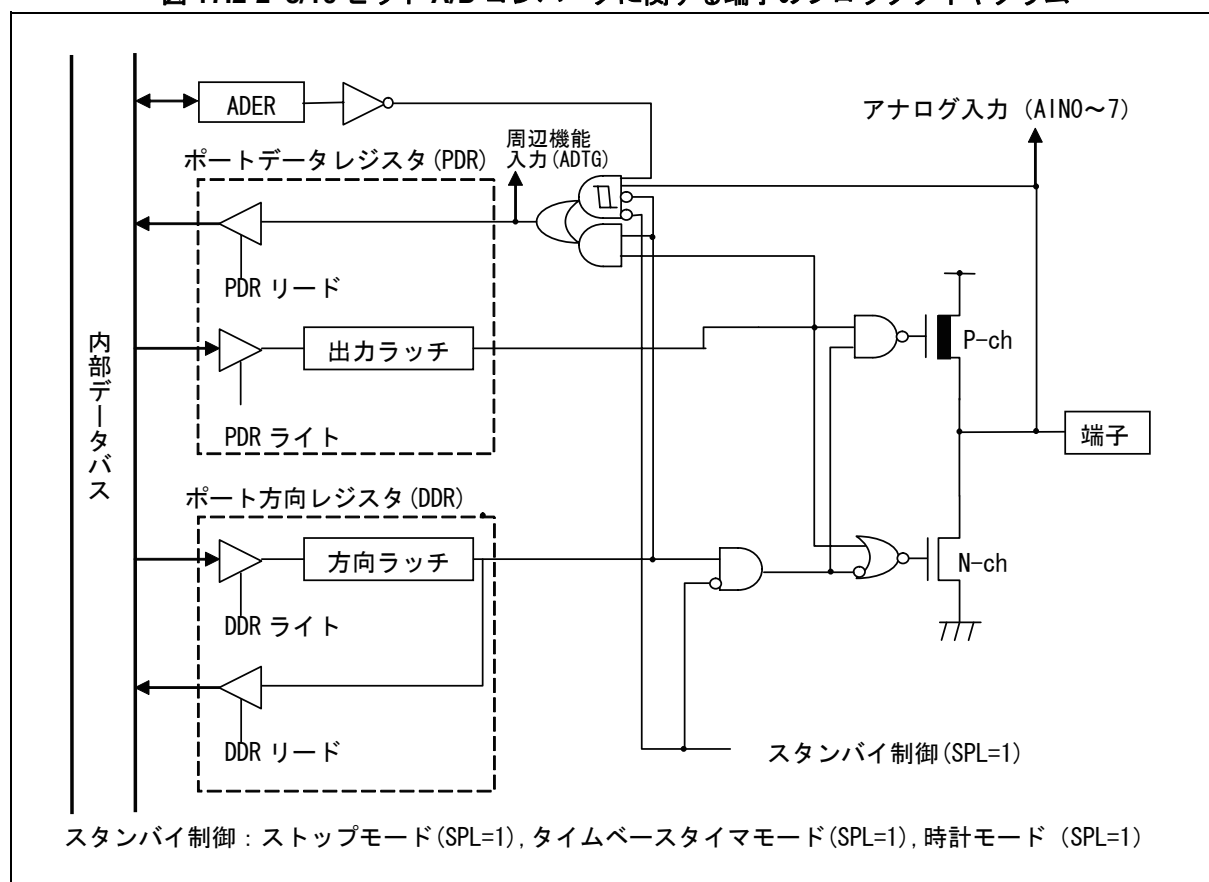
アナログ入力として使用する場合は、ポート方向レジスタ (DDR6: bit8,7,6,5,4,3,2,1,0 → "0")とアナログ許可レジスタ (ADER:bit15,14,13,12,11,10,9,8 → "1")を設定してください。

● ADTG 端子として使用する場合の設定

A/D コンバータの外部トリガとして使用する場合、P93/ADTG 端子は、ポート方向レジスタにより入力ポート (DDR9 bit11 → "0") に設定してください。

■ 8/10 ビット A/D コンバータに関する端子のブロックダイアグラム

図 17.2-2 8/10 ビット A/D コンバータに関する端子のブロックダイアグラム



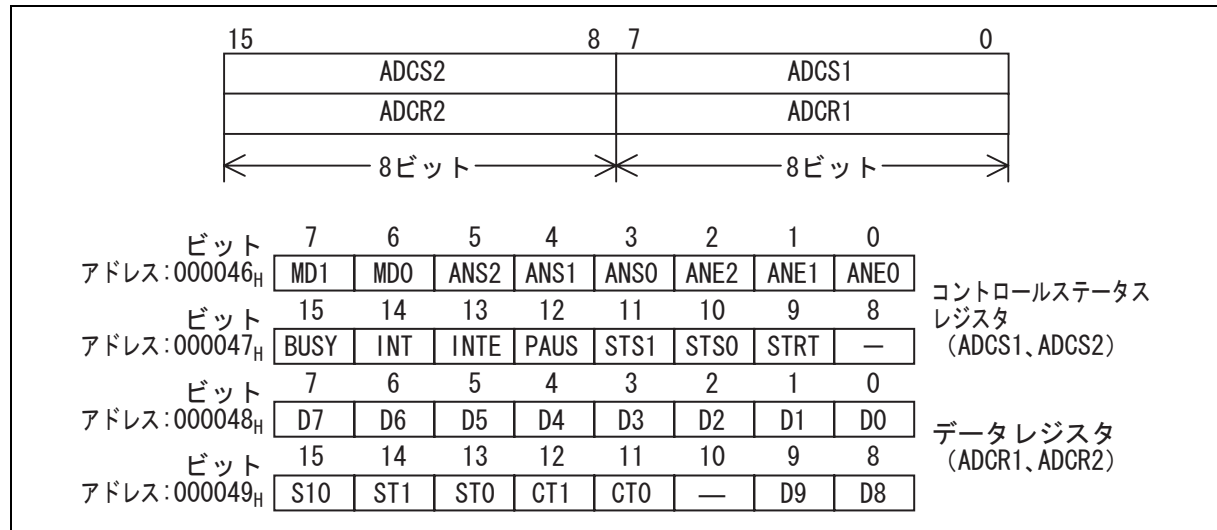
17.3 8/10 ビット A/D コンバータのレジスタの構成と機能

8/10 ビット A/D コンバータに関するレジスタの構成と機能について示します。

■ 8/10 ビット A/D コンバータのレジスタ一覧

図 17.3-1 に、8/10 ビット A/D コンバータのレジスタ一覧を示します。

図 17.3-1 8/10 ビット A/D コンバータのレジスタ一覧



17.3.1 コントロールステータスレジスタ 1(ADCS1)

コントロールステータスレジスタ 1(ADCS1) は, A/D コンバータの制御およびステータス表示を行います。

■ コントロールステータスレジスタ 1(ADCS1)

下図に, コントロールステータスレジスタ 1(ADCS1) のビット構成を示します。

ビット	7	6	5	4	3	2	1	0	
ADCS1 アドレス:000046 _H	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	
	0	0	0	0	0	0	0	0	←初期値
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	←ビット属性

<注意事項>

コントロールステータスレジスタ 1(ADCS1) は, A/D 変換動作中に書き換えしないでください。

以下に, コントロールステータスレジスタ 1(ADCS1) の各ビットの機能を説明します。

[bit7, bit6] MD1, MD0:A/D converter MoDe set(動作モード)

これらのビットは, 動作モードを設定します。下表に選択できる動作モードを示します。

MD1 MD0	動作モード
00 _B	単発モード, 動作中の再起動はすべて可能
01 _B	単発モード, 動作中の再起動不可能
10 _B	連続モード, 動作中の再起動不可能
11 _B	停止モード, 動作中の再起動不可能

各モードの機能は, 以下のとおりです。

- 単発モード: ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまで A/D 変換を連続して行い, 1 回変換が終了すると停止します。
- 連続モード: ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまで A/D 変換を繰り返し行います。
- 停止モード: ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまで 1 チャンネルごとに A/D 変換を行い, 一時停止します。変換再開は, 起動要因発生によって行われます。

これらのビットは, リセット時に "00_B" に初期化されます。

各動作モードでは, 次の動作をします。

- 連続モード, 停止モードで A/D 変換を起動すると, BUSY ビットで停止するまで変換動作を続けます。
- 各動作モードは, BUSY ビットに "0" を書き込むことによって停止します。
- 単発, 連続, 停止の各モードにおける「再起動の不可能」は, タイマ, 外部トリガ, ソフトウェアすべての起動要因に対して適用されます。

17.3.2 コントロールステータスレジスタ 2(ADCS2)

コントロールステータスレジスタ 2(ADCS2) は、A/D コンバータの制御およびステータス表示を行います。

■ コントロールステータスレジスタ 2(ADCS2)

下図に、コントロールステータスレジスタ 2(ADCS2) のビット構成を示します。

ビット	15	14	13	12	11	10	9	8	
ADCS2 アドレス:000047 _H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	予約	
	0	0	0	0	0	0	0	0	←初期値
	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W	←ビット属性

以下に、コントロールステータスレジスタ 2(ADCS2) の各ビットの機能を説明します。

[bit15] BUSY:busy flag and stop

- ・ 読出し時：A/D コンバータ動作表示用のビットです。A/D 変換の起動によってセットされ、終了でクリアされます。
- ・ 書込み時：A/D 動作中にこのビットに "0" を書き込むと、強制的に動作が停止されます。連続モード時および停止モード時の強制停止に利用します。動作表示用のビットに "1" を書き込むことはできません。RMW 系命令では "1" が読み出されます。単発モードでは A/D 変換終了によってクリアされます。連続、停止モードでは "0" 書込みによって停止するまでクリアはされません。リセット時に "0" に初期化されます。

<注意事項>

強制停止とソフトウェア起動を同時には行わないでください (BUSY=0, STRT=1)。

[bit14] INT:interrupt

このビットは、データ表示ビットです。変換データが ADCR に書き込まれるとセットされます。

bit5(INTE) が "1" のときにこのビットがセットされると、割込み要求が発生します。また、μDMAC 起動が許可されている場合は μDMAC が起動されます。"1" の書込みは、意味を持ちません。クリアは、"0" 書込みと μDMAC 割込みクリア信号で行われます。

リセット時に "0" に初期化されます。

「17.6 8/10 ビット A/D コンバータの変換データ保護機能」の変換データ保護機能の使用上のご注意もお読みください。

<注意事項>

このビットの "0" 書込みによるクリアは、A/D 停止中に行ってください。

[bit13] INTE:interrupt enable

このビットは、変換終了による割込みの許可 / 不許可を指定します。

- 0: 割込みの不許可
- 1: 割込みの許可

μDMACを使用するときは、このビットをセットしてください。μDMACは、割込み要求発生で起動します。

リセット時に "0" に初期化されます。

「17.6 8/10 ビット A/D コンバータの変換データ保護機能」の変換データ保護機能の使用上のご注意もお読みください。

[bit12] PAUS:A/D converter pause

このビットは、A/D 変換動作が一時的に停止した場合にセットされます。

A/D 変換結果を格納するレジスタが 1 つのため、連続で変換した場合、変換結果を μDMAC で転送しなければ前のデータが壊れてしまいます。これを保護するため、データレジスタの内容を μDMAC で転送していない場合は、次の変換データが格納されません。この間、A/D 変換動作は停止します。μDMAC で転送を終了すると、A/D は変換を再開します。

このレジスタは、μDMAC を使用したときのみ有効です。

「17.6 8/10 ビット A/D コンバータの変換データ保護機能」の変換データ保護機能の使用上のご注意もお読みください。

リセット時に "0" に初期化されます。

[bit11, bit10] STS1, STS0:start source select

これらのビットは、A/D 起動要因を選択します。

STS1	STS0	機能
0	0	ソフトウェア起動
0	1	外部端子トリガでの起動とソフトウェア起動
1	0	タイマでの起動とソフトウェア起動
1	1	外部端子トリガ、タイマでの起動とソフトウェア起動

起動が兼用になっているモードでは、最初に発生した要因で起動します。

起動要因は書換えと同時に変わりますので、A/D 動作中に書き換えるときは注意してください。

- 外部端子トリガの場合は、立下りエッジを検出します。
- 外部トリガ入力レベルが "L" のときにこのビットを書き換えて外部トリガ起動を選択すると、A/D が起動する場合があります。
- タイマ選択時は、PPG1 の出力が選択されます。

リセット時 "00_B" に初期化されます。

<注意事項>

A/D コンバータを外部トリガ、または内部タイマを使って起動する場合、外部トリガおよび内部タイマの入力値は、インアクティブ側の状態で設定してください。

STS1, 0 の設定時は、ADTG=1 入力、内部タイマ (PPG1)=0 出力の状態で行ってください。

[bit9] STRT:start

8/10 ビット A/D コンバータをソフトウェアで起動します。

- このビットに "1" を書き込むことによって A/D を起動します。
- 再起動をかけるときは、再び書き込んでください。
- 停止モード時は、動作機能上再起動はかかりません。
- リセット時に "0" に初期化されます。
- バイト/ワード命令では "1" が読み出されます。
- 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。

<注意事項>

強制停止とソフト起動を同時には行わないでください (BUSY=0, STRT=1)。

[bit8] 予約

書込み時は、"0" を書き込んでください。

17.3.3 データレジスタ (ADCR2, ADCR1)

データレジスタ (ADCR2, ADCR1) の構成および機能について説明します。

■ データレジスタ (ADCR2, ADCR1)

下図に、データレジスタ (ADCR2, ADCR1) のビット構成を示します。

ADCR1 アドレス: 000048 _H	ビット	7	6	5	4	3	2	1	0	
		D7	D6	D5	D4	D3	D2	D1	D0	
		X	X	X	X	X	X	X	X	←初期値
ADCR2 アドレス: 000049 _H	ビット	15	14	13	12	11	10	9	8	
		S10	ST1	ST0	CT1	CT0	—	D9	D8	
		0	0	0	0	0	X	X	X	←初期値
		W	W	W	W	W	R	R	R	←ビット属性

データレジスタ (ADCR2, ADCR1) は変換結果格納レジスタであり、変換結果であるディジタル値が格納されます。ADCR2 には、変換値の上位 2 ビットが、ADCR1 には下位 8 ビットが格納されます。このレジスタの値は、1 回の変換終了時ごとに更新されます。通常は、最終変換値が格納されています。S10 ビットの手換えは、必ず変換動作前の A/D 動作停止状態でするようにしてください。変換後に手換えをしたとき、ADCR の内容は不定です。ADCR レジスタの読出しは 10 ビットモードを指定したときは、必ずワード命令を使用してください。また、ADCR レジスタには、変換データ保護機能があります（「17.6 8/10 ビット A/D コンバータの変換データ保護機能」を参照）。

A/D 動作中にこのレジスタにデータを書き込まないようにしてください。S10 ビットが "0" のときは、変換結果が 10 ビットで、"1" のときは 8 ビットで得られます。

ST1	ST0	サンプリング時間設定ビット
0	0	20 マシンサイクル (0.8μs@25MHz)
0	1	32 マシンサイクル (1.28μs@25MHz)
1	0	48 マシンサイクル (1.92μs@25MHz)
1	1	128 マシンサイクル (5.12μs@25MHz)

CT1	CT0	コンパリア時間設定ビット
0	0	44 マシンサイクル (1.76μs@25MHz)
0	1	66 マシンサイクル (2.64μs@25MHz)
1	0	88 マシンサイクル (3.52μs@25MHz)
1	1	176 マシンサイクル (7.04μs@25MHz)

（注意事項） 25MHz 動作時に ST1, ST0=00 または 01 を設定すると、正常にアナログ電圧を取り込めない場合があります。

17.4 8/10 ビット A/D コンバータの割込み

8/10 ビット A/D コンバータでは、A/D 変換が終了して変換結果が A/D データレジスタ (ADCR) に格納されると、割込み要求が発生します。また、DMA 転送および拡張インテリジェント I/O サービス (EI2OS) を起動することができます。

■ 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下記の表に示します。

割込み要求フラグ	ADCS:INT (bit14)
割込み要求出力許可ビット	ADCS:INTE (bit13)
割込み発生要因	A/D 変換結果が A/D データレジスタ (ADCR) に格納された

■ A/D コンバータの割込み

アナログ入力電圧の A/D 変換が終了して、A/D 変換結果がデータレジスタ (ADCR) に格納されると、A/D コントロールステータスレジスタの割込み要求フラグビット (ADCS:INT) に "1" が設定されます。割込み要求の出力が許可されている場合 (ADCS:INTE=1) に、割込み要求フラグビットが設定されると (ADCS:INT=1) 割込み要求が発生します。

■ 8/10 ビット A/D コンバータの割込みと DMA 転送および EI²OS

表 17.4-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 17.4-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS のクリア	μDMAC チャンネル番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
A/D コンバータ *	○	15	#40	FFFF5C _H	ICR14	0000BE _H

○: 割込み要求フラグはクリアされます。

* : この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

A/D コンバータ割込みは DMA 転送機能、および EI²OS 機能に対応しています。DMA または EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

17.5 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、逐次比較方式で動作し、10 ビットの分解能をもっています。8/10 ビット A/D コンバータには、変換結果格納用のレジスタが 1 つ (10 ビット) しかないため、1 回の変換が終了した時点でデータレジスタ (ADCR1, 2) が更新されます。このため、単独では連続変換処理には適しませんので、F²MC-16LX の μ DMAC 機能を使って変換データをメモリに転送しながら変換することを推奨します。

■ 動作モード

● 単発モード

このモードは、ANS ビットと ANE ビットで設定されたアナログ入力を順に変換し、ANE ビットで設定された終了チャンネルまで変換が終わると、A/D は動作を停止します。開始チャンネルと終了チャンネルが同じ場合 (ANS=ANE) は、1 チャンネル変換動作になります。

例：

ANS=000_B, ANE=011_B

開始 → AN0 → AN1 → AN2 → AN3 → 終了

ANS=010_B, ANE=010_B

開始 → AN2 → 終了

<注意事項>

A/D 変換の再起動と、終了が同時に発生した場合、再起動は行われずに A/D 変換は終了し、データレジスタ (ADCR1/0) には 300_H が格納されます。したがって、A/D 変換再起動と、終了が同時に発生しないように再起動をかけてください。

● 連続モード

このモードは、ANS ビットと ANE ビットで設定されたアナログ入力を順に変換していき、ANE ビットで設定された終了チャンネルまで変換が終わると、ANS のアナログ入力に戻り A/D 変換動作を続けます。

開始チャンネルと終了チャンネルが同じ場合 (ANS=ANE) は、1 チャンネル変換を続けます。

例：

ANS=000_B, ANE=011_B

開始 → AN0 → AN1 → AN2 → AN3 → AN0 繰り返し

ANS=010_B, ANE=010_B

開始 → AN2 → AN2 → AN2 繰り返し

連続モードで変換させると、BUSY ビットに "0" を書き込むまで変換動作を繰り返し、続けます。

BUSY ビットに "0" を書き込むと、強制動作停止となります。

強制動作停止した場合は、変換レジスタには変換完了した前のデータが格納されています。

<注意事項>

連続モードを強制動作停止させると、変換データは強制動作停止前のデータを保持します。

● 停止モード

このモードは、ANS ビットと ANE ビットで設定されたアナログ入力を順に変換していきますが、1 チャンネル変換するたびに変換動作を一時停止します。一時停止を解除するには、もう一度 A/D 起動をかける必要があります。

ANE ビットで設定された終了チャンネルまで変換が終わると、ANS で設定されたチャンネルのアナログ入力に戻り A/D 変換動作を続けます。

開始チャンネルと終了チャンネルが同じ場合 (ANS=ANE) は、1 チャンネルを変換します。

例：

ANS=000_B, ANE011_B

開始 → AN0 → 停止 → 起動 → AN1 → 停止 → 起動 →

→ AN2 → 停止 → 起動 → AN3 → 停止 → 起動 → AN0 繰り返し

ANS=010_B, ANE=010_B

開始 → AN2 → 停止 → 起動 → AN2 → 停止 → 起動 → AN2 繰り返し

このときの A/D 起動要因は、STS1, 0 で設定されたものだけです。

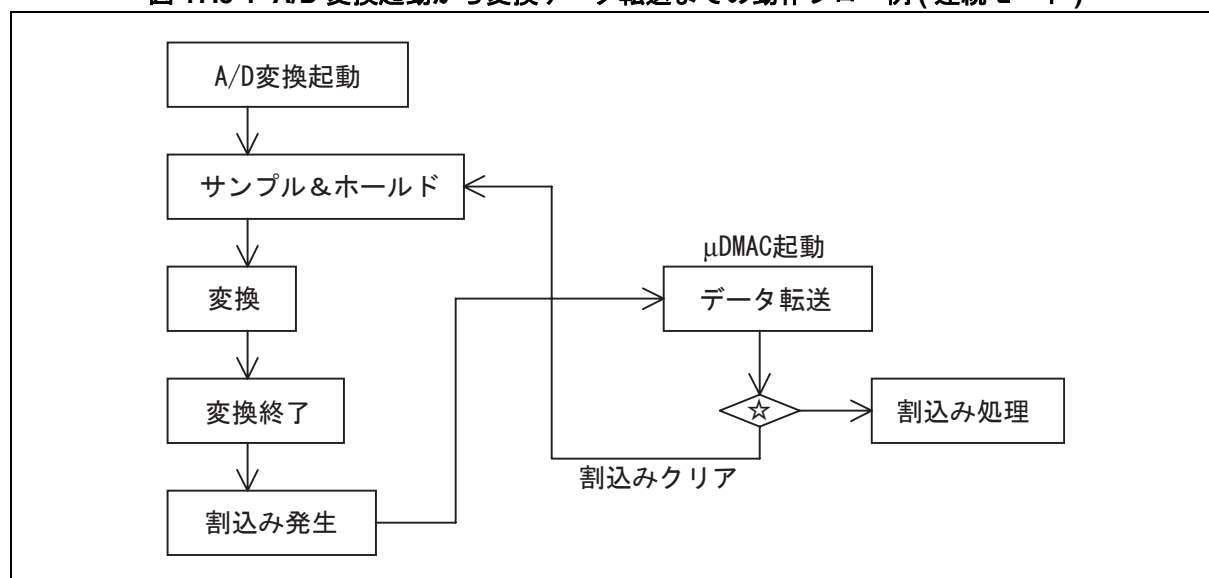
このモードを使用することによって、変換開始の同期をかけることが可能となります。

■ μDMAC を使った変換動作

A/D 変換起動から変換データ転送までの動作フロー例 (連続モード) を、図 17.5-1 に示します。

図中の☆印は、μDMAC の設定で決まります。

図 17.5-1 A/D 変換起動から変換データ転送までの動作フロー例 (連続モード)



17.5.1 単発モードにおける μ DMAC の起動例

単発モードにおける μ DMAC の起動例を示します。

■ 単発モードにおける μ DMAC の起動例

以下の条件における起動例を示します。

- アナログ入力, AN1 ~ AN3 まで変換して終了
- 変換データを 200_H ~ 206_H 番地に順に転送
- ソフトウェアで起動
- 割込みレベルを最高に設定

項目設定	プログラム例	動作
μ DMAC の設定	MOV ICR14,#00 _H	最高割込み設定, 割込み許可を設定
	MOV BAPL,#00 _H	変換データの転送番地
	MOV BAPM,#02 _H	
	MOV BAPH,#00 _H	
	MOV DMACS,#18 _H	DMA コントロールステータスレジスタの設定 (ワードデータ転送, 転送後転送番地をインクリメントする)
	MOV IOA,#48 _H	A/D コンバータの結果をレジスタに設定
	MOV DCT,#03 _H	転送回数の設定 (3 回転送)
	MOVW DERL,#8000 _H	μ DMAC イネーブルレジスタ設定 (EN15)
A/D コンバータの設定	MOV ADCS1,#0B _H	単発モード, 開始チャンネル AN1, 終了チャンネル AN3
	MOV ADCS2,#A2 _H	ソフトウェア起動, A/D 変換開始
μ DMAC 終了シーケンス	WBTC io ADCS2:7	A/D 変換終了判定
	MOV ADCS2,#000 _H	リソースの割込みクリア
	MOVW DSRL,#0000 _H	μ DMAC ステータスレジスタのクリア
	RETI	割込みからの復帰

ICR14 : 割込み制御レジスタ

BAPL : バッファアドレスポインタ下位

BAPM : バッファアドレスポインタ中位

BAPH : バッファアドレスポインタ上位

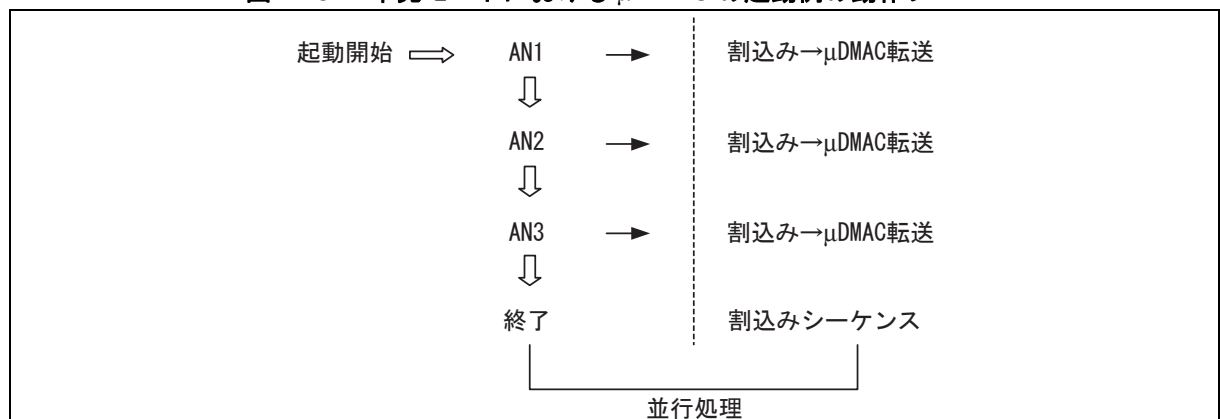
ISCS : ステータスレジスタ

IOA : アドレスレジスタ

DCT : データカウンタ

図 17.5-2 に, 起動例の動作フローを示します。

図 17.5-2 単発モードにおける μ DMAC の起動例の動作フロー



17.5.2 連続モードにおける μ DMAC の起動例

連続モードにおける μ DMAC の起動例を示します。

■ 連続モードにおける μ DMAC の起動例

以下の条件における起動例を示します。

- アナログ入力 (AN3 ~ AN5) を変換して、各チャネルの変換データを 2 つ取得
- 変換データを 600_H ~ 60C_H 番地に順に転送
- 外部エッジで起動
- 割込みレベルを最高に設定

項目設定	プログラム例	動作
μ DMAC の設定	MOV ICR14,#00 _H	最高割込み設定, 割込み許可を設定
	MOV BAPL,#00 _H	変換データの転送番地
	MOV BAPM,#06 _H	
	MOV BAPH,#00 _H	
	MOV DMACS,#18 _H	DMA コントロールステータスレジスタの設定 (ワードデータ転送, 転送後転送番地をインクリメントする)
	MOV IOA,#48 _H	A/D コンバータの結果をレジスタに設定
	MOV DCT,#06 _H	転送回数の設定 (6 回転送)
	MOVW DERL,#8000 _H	μ DMAC イネーブルレジスタ設定 (EN15)
A/D コンバータの設定	MOV ADCS1,#9D _H	単発モード, 開始チャネル AN3, 終了チャネル AN5
	MOV ADCS2,#A4 _H	外部エッジ起動, A/D 変換開始
μ DMAC 終了シーケンス	WBTC io ADCS2:7	A/D 変換終了判定
	MOV ADCS2,#000 _H	リソースの割込みクリア
	MOVW DSRL,#0000 _H	μ DMAC ステータスレジスタのクリア
	RETI	割込みからの復帰

ICR14 : 割込み制御レジスタ

BAPL : バッファアドレスポインタ下位

BAPM : バッファアドレスポインタ中位

BAPH : バッファアドレスポインタ上位

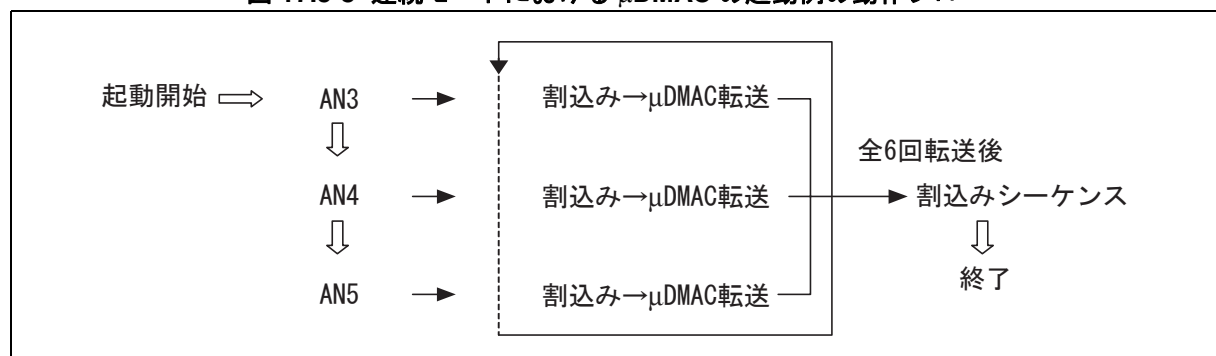
ISCS : ステータスレジスタ

IOA : アドレスレジスタ

DCT : データカウンタ

図 17.5-3 に、起動例の動作フローを示します。

図 17.5-3 連続モードにおける μ DMAC の起動例の動作フロー



17.5.3 停止モードにおける μ DMAC の起動例

停止モードにおける μ DMAC の起動例を示します。

■ 停止モードにおける μ DMAC の起動例

以下の条件における起動例を示します。

- アナログ入力 AN3 を一定期間おいて 12 回変換
- 変換データを 600_H ~ 618_H 番地に順に転送
- 外部エッジで起動
- 割込みレベルを最高に設定

項目設定	プログラム例	動作
μ DMAC の設定	MOV ICR14,#00 _H	最高割込み設定, 割込み許可を設定
	MOV BAPL,#00 _H	変換データの転送番地
	MOV BAPM,#06 _H	
	MOV BAPH,#00 _H	
	MOV DMACS,#18 _H	DMA コントロールステータスレジスタの設定 (ワードデータ転送, 転送後転送番地をインクリメントする)
	MOV IOA,#48 _H	A/D コンバータの結果をレジスタに設定
	MOV DCT,#0C _H	転送回数の設定 (12 回転送)
	MOVW DERL,#8000 _H	μ DMAC イネーブルレジスタ設定 (EN15)
A/D コンバータの設定	MOV ADCS1,#DB _H	連続モード, 開始チャネル AN3, 終了チャネル AN3 (1 チャネル変換)
	MOV ADCS2,#A4 _H	外部エッジ起動, A/D 変換開始
μ DMAC 終了シーケンス	WBTC io ADCS2:7	A/D 変換終了判定
	MOV ADCS2,#000 _H	リソースの割込みクリア
	MOVW DSRL,#0000 _H	μ DMAC ステータスレジスタのクリア
	RETI	割込みからの復帰

ICR14 : 割込み制御レジスタ

BAPL : バッファアドレスポインタ下位

BAPM : バッファアドレスポインタ中位

BAPH : バッファアドレスポインタ上位

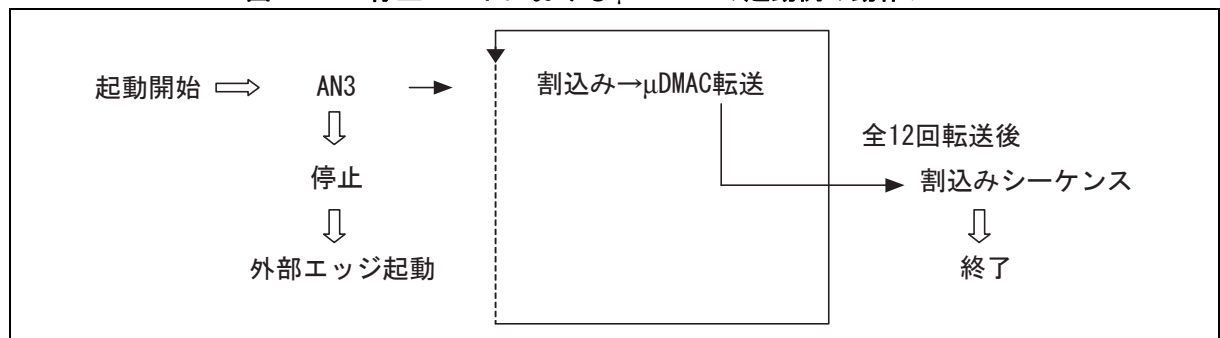
ISCS : ステータスレジスタ

IOA : アドレスレジスタ

DCT : データカウンタ

図 17.5-4 に, 起動例の動作フローを示します。

図 17.5-4 停止モードにおける μ DMAC の起動例の動作フロー



17.6 8/10 ビット A/D コンバータの変換データ保護機能

8/10 ビット A/D コンバータは、変換データ保護機能を持ち、 μ DMAC を使って連続変換と複数のデータを確保できます。

■ 変換データ保護機能

変換データレジスタは 1 つのため、連続で A/D 変換をすると 1 回の変換が終了した時点で変換データが格納されて前のデータが失われます。これを保護するために、この 8/10 ビット A/D コンバータは、変換が終了しても前データが μ DMAC を使ってメモリへ転送されていなかった場合に、変換データをレジスタへ格納せず、A/D 変換を一時停止する機能を持っています。

μ DMAC で変換データをメモリへ転送された後に一時停止が解除されます。

前データが転送されていれば、一時停止することなく連続して A/D 変換が行われます。

■ 変換データ保護機能の使用上の注意

この機能は、ADCS2 の INT, INTE ビットに関係しています。

データ保護機能は割込み許可 (INTE=1) 状態でしか動作しません。

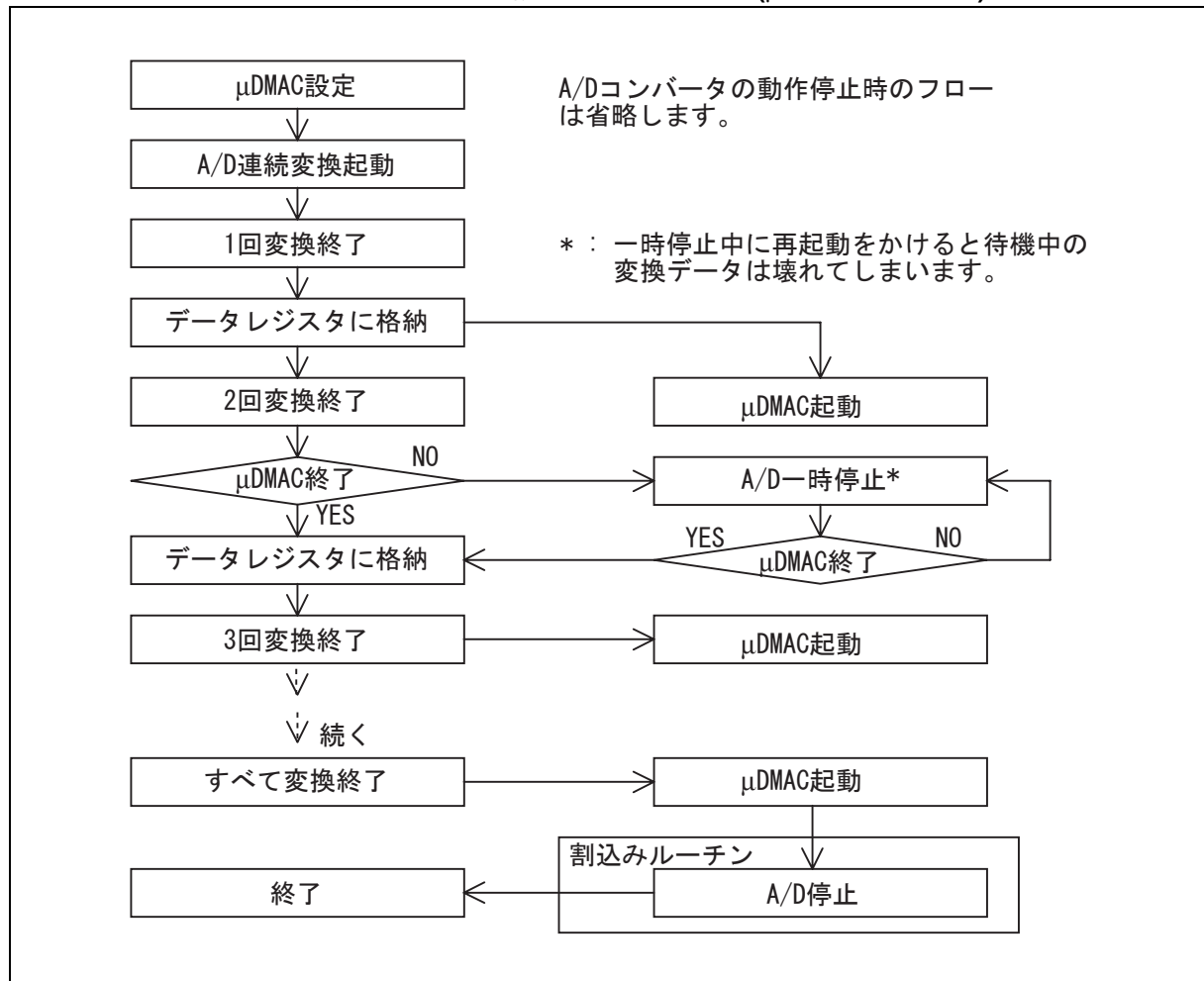
割込み禁止 (INTE=0) の場合、この機能は動作せず、連続で A/D 変換した場合に次々と変換データがレジスタに格納されて、旧データは失われます。また、割込み許可 (INTE=1) 状態で μ DMAC を使わない場合、INT ビットはクリアされないためデータ保護機能が働き、8/10 ビット A/D コンバータは変換を一時停止状態にします。この場合は、割込みシーケンスで INT ビットをクリアすると、停止状態が解除されます。

DMA 動作中で A/D 変換が一時停止をしているときに割込みを禁止にすると、A/D 変換が動作し転送する前に変換データレジスタの内容が変わることがあります。また、一時停止中に再起動をかけると、待機データが壊れます。

■ 変換データ保護機能の動作フロー (μDMAC を使用時)

図 17.6-1 に、変換データ保護機能の動作フローを示します。

図 17.6-1 変換データ保護機能の動作フロー (μDMAC を使用時)



17.7 8/10 ビット A/D コンバータの使用上の注意

8/10 ビット A/D コンバータの使用上の注意について説明します。

■ 外部トリガ / 内部タイマで起動する場合の注意

A/D コンバータを外部トリガ, または内部タイマを使って起動する場合, 外部トリガおよび内部タイマの入力値は, インアクティブの状態を設定してください。

STS1, 0 の設定時は, ADTG=1 入力, 内部タイマ (PPG1)=0 出力の状態で行ってください。

■ アナログ入力端子の扱い

アナログ入力に使用する端子は, 必ずその対応する ADER のビットを "1" に設定してください。

ビット	7	6	5	4	3	2	1	0	初期値
アドレス: 00001F _H	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	11111111 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

ポート 6 の各端子を以下のように制御してください。

- 0: ポート入出力モード
- 1: アナログ入力モード

リセットで "1" になります。

● A/D 変換の再起動について

A/D 変換の再起動と, 終了が同時に発生した場合, 再起動は行われずに A/D 変換は終了し, データレジスタ (ADCR1/0) には 300_H が格納されます。したがって, A/D 変換再起動と, 終了が同時に発生しないように再起動をかけてください。

17.8 8/10 ビット A/D コンバータのプログラム例

8/10 ビット A/D コンバータのプログラム例を示します。

■ 8/10 ビット A/D コンバータのプログラム例

<p>設定手順例</p> <p>AN0 より入力したレベルを A/D 変換する例 (シングル変換, ソフトトリガ) を以下に示します。</p> <p>< 初期設定 ></p> <ul style="list-style-type: none"> ・ AN0 入力許可 レジスタ名 ビット名 <table border="1" style="width: 100%;"> <tr> <td>ポートの入力設定</td><td>DDR6. P60</td></tr> <tr> <td>AN0 の A/D 入力許可</td><td>ADER.ADE0</td></tr> </table> <ul style="list-style-type: none"> ・ A/D0 変換時間 <table border="1" style="width: 100%;"> <tr> <td>変換時間の設定</td><td>ADCR2</td></tr> <tr> <td>サンプリング時間設定 >></td><td>.ST1-0</td></tr> <tr> <td>コンベア時間設定 >></td><td>.CT1-0</td></tr> </table> <ul style="list-style-type: none"> ・ A/D0 制御 レジスタ名 ビット名 <table border="1" style="width: 100%;"> <tr> <td>コントロールステータスレジスタ 1</td><td>ADCS1</td></tr> <tr> <td>動作モードの選択 >></td><td>.MD1-0</td></tr> <tr> <td>開始チャネルの設定 >></td><td>.ANS2-0</td></tr> <tr> <td>終了チャネルの設定 >></td><td>.ANE2-0</td></tr> </table> <table border="1" style="width: 100%;"> <tr> <td>コントロールステータスレジスタ 2</td><td>ADCS2</td></tr> <tr> <td>割込み要求フラグのクリア >></td><td>.BUSY</td></tr> <tr> <td>割込み禁止 >></td><td>.INT</td></tr> <tr> <td></td><td>.INTE</td></tr> <tr> <td></td><td>.PAUS</td></tr> <tr> <td>起動トリガ選択 >></td><td>.STS1-0</td></tr> <tr> <td></td><td>.STRT</td></tr> <tr> <td></td><td>.予約ビット</td></tr> </table> <ul style="list-style-type: none"> ・ 割込み関連 <table border="1" style="width: 100%;"> <tr> <td>A/D 割込みレベルの設定</td><td>ICR14</td></tr> <tr> <td>I フラグの設定</td><td>(CCR)</td></tr> </table> <p>< A/D 起動 ></p> <ul style="list-style-type: none"> ・ AN0 起動 レジスタ名 ビット名 <table border="1" style="width: 100%;"> <tr> <td>A/D0 割込み許可</td><td>ADCS2</td></tr> <tr> <td></td><td>.INT</td></tr> <tr> <td></td><td>.INTE</td></tr> </table> <table border="1" style="width: 100%;"> <tr> <td>A/D0 のソフト起動</td><td>ADCS2</td></tr> <tr> <td></td><td>.STRT</td></tr> </table>	ポートの入力設定	DDR6. P60	AN0 の A/D 入力許可	ADER.ADE0	変換時間の設定	ADCR2	サンプリング時間設定 >>	.ST1-0	コンベア時間設定 >>	.CT1-0	コントロールステータスレジスタ 1	ADCS1	動作モードの選択 >>	.MD1-0	開始チャネルの設定 >>	.ANS2-0	終了チャネルの設定 >>	.ANE2-0	コントロールステータスレジスタ 2	ADCS2	割込み要求フラグのクリア >>	.BUSY	割込み禁止 >>	.INT		.INTE		.PAUS	起動トリガ選択 >>	.STS1-0		.STRT		.予約ビット	A/D 割込みレベルの設定	ICR14	I フラグの設定	(CCR)	A/D0 割込み許可	ADCS2		.INT		.INTE	A/D0 のソフト起動	ADCS2		.STRT	<p>プログラム例</p> <pre> void AD_sample() { AD0_INITIAL(); AD0_ch.0_start(); } void AD0_INITIAL(void) { IO_DDR6.bit.D60 = 0; /* DDR6 AN0(P60) 入力 */ IO_ADER.bit.ADE0 = 1; /* AN0 のみ A/D 入力 */ IO_ADCR2.byte = 0x70; /* 値は推奨値 */ /* 11 */ /* 10 */ IO_ADCS1.byte= 0x00; /* 設定値 : 00000000 (bit) */ /* bit7-6=00: 単発モード */ /* bit5-3=000: AN0 */ /* bit2-0=000: AN0 */ IO_ADCS2.byte= 0x00; /* 設定値 : 00000000 (bit) */ /* bit15=0: (影響なし) */ /* bit14=0: 割込み要求クリア */ /* bit13=0: 割込み禁止 */ /* bit12=0: */ /* bit11-10=00: ソフトトリガ */ /* bit9=0: */ /* bit8=0: "0" 書き込み */ IO_ICR14.byte= 0x00; /* 値は任意 */ __EI(); /* 割込み許可 */ } void AD0_ch.0_start(void) { IO_ADCS2.byte= 0x20; /* bit14=0: AD0 割込みフラグクリア */ /* bit13=1: AD0 割込み許可 */ IO_ADCS2.byte= 0xA2; /* bit9=1: ソフト起動 */ } </pre>
ポートの入力設定	DDR6. P60																																																
AN0 の A/D 入力許可	ADER.ADE0																																																
変換時間の設定	ADCR2																																																
サンプリング時間設定 >>	.ST1-0																																																
コンベア時間設定 >>	.CT1-0																																																
コントロールステータスレジスタ 1	ADCS1																																																
動作モードの選択 >>	.MD1-0																																																
開始チャネルの設定 >>	.ANS2-0																																																
終了チャネルの設定 >>	.ANE2-0																																																
コントロールステータスレジスタ 2	ADCS2																																																
割込み要求フラグのクリア >>	.BUSY																																																
割込み禁止 >>	.INT																																																
	.INTE																																																
	.PAUS																																																
起動トリガ選択 >>	.STS1-0																																																
	.STRT																																																
	.予約ビット																																																
A/D 割込みレベルの設定	ICR14																																																
I フラグの設定	(CCR)																																																
A/D0 割込み許可	ADCS2																																																
	.INT																																																
	.INTE																																																
A/D0 のソフト起動	ADCS2																																																
	.STRT																																																

(続く)

(続き)

< 割込み >	
・ 変換値の読出し	

■ プログラム例以外の設定方法

● 変換モードの種類と設定方法

変換には, 以下の 3 つのモードがあります。

- ・ 1 回だけ変換して終了する単発変換モード
- ・ 指定したチャネル分を変換する連続変換モード
- ・ 指定したチャネル分を変換後, 一時停止する停止モード

変換モードは, 変換モード選択ビット (ADCS1.MD[1:0]) で設定します。

動作モード	変換モード選択ビット (MD[1:0])
単発変換モードにするには	"00 _B ", "01 _B " にする
連続変換モードにするには	"10 _B " にする
停止モードにするには	"11 _B " にする

● ビット長を指定する方法

変換結果の格納ビット数 (ADCR.S10) で設定します。

動作モード	変換結果の格納ビット数 (S10)
ADCR1, 2 レジスタに 10 ビットで格納するには	"0" にする
ADCR1 レジスタに 8 ビットで格納するには	"1" にする

● チャネルを選択する方法

変換を開始するチャネルは A/D 変換開始チャネルビット (ADCS1.ANS[2:0]) にて指定します。

変換を終了するチャネルは A/D 変換終了チャネルビット (ADCS1.ANE[2:0]) にて指定します。

● アナログ端子入力を有効にする方法

アナログ入力許可レジスタ ADER で設定します。

動作	制御ビット	設定
AN0 端子を入力にするには	(ADER.ADE0)	"1" にする
AN1 端子を入力にするには	(ADER.ADE1)	"1" にする
AN2 端子を入力にするには	(ADER.ADE2)	"1" にする
AN3 端子を入力にするには	(ADER.ADE3)	"1" にする
AN4 端子を入力にするには	(ADER.ADE4)	"1" にする
AN5 端子を入力にするには	(ADER.ADE5)	"1" にする
AN6 端子を入力にするには	(ADER.ADE6)	"1" にする
AN7 端子を入力にするには	(ADER.ADE7)	"1" にする

● A/D コンバータの起動方法を選択する方法

起動トリガには、以下の 3 種類があります。

- ソフトトリガ
- リロードタイマの立上り信号
- 外部トリガ入力の立下り信号

起動トリガは、起動要因選択ビット (ADCS2.STS[1:0]) で設定します。

A/D 起動要因	起動要因選択ビット (STS[1:0])
ソフトトリガを指定するには	"00" にする
外部トリガ/ソフトトリガを指定するには	"01" にする
リロードタイマ/ソフトトリガを指定するには	"10" にする
外部トリガ/リロードタイマ/ソフトトリガを指定するには	"11" にする

A/D コンバータは、選択した要因の中で最初にきた要因で起動します。

● A/D コンバータを起動する方法

- ソフトトリガの発生方法

ソフトトリガは A/D 変換ソフトトリガビット (ADCS2.STRT) で設定します

動作	A/D 変換ソフトトリガビット (STRT)
ソフトトリガを発生させるには	"1" を書き込む

- リロードタイマ 0, リロードタイマ 1 での起動方法

リロードタイマの設定と起動が必要です。詳細は「第 14 章 16 ビットリロードタイマ」をご参照ください。

リロードタイマのアンダフローによって、リロードタイマの出力信号が立上りになると、起動トリガが発生します。

- 外部トリガによる起動方法

外部トリガは外部トリガ入力端子 ADTG で設定します

外部トリガ入力端子の設定は、データ方向ビット (DDR9.P93) で設定します。

動作	設定
ADTG 端子をトリガ入力にするには	データ方向レジスタの (DDR9.P93) ビットを "0" にする

● 変換終了を確認する方法

変換終了を確認する方法は 2 つあります。

- A/D 変換終了割込み要求ビット (ADCS2.INT) で確認する方法

(INT)	意味
読み出した値が "0" の場合	A/D 変換終了割込み要求なし
読み出した値が "1" の場合	A/D 変換終了割込み要求あり

- 動作確認ビット (ADCS2.BUSY) で確認する方法

(BUSY)	設定
読み出した値が "0" の場合	A/D 変換終了 (停止中)
読み出した値が "1" の場合	A/D 変換中

● 変換値を読み出す方法

変換値は、データレジスタ ADCR1, 2 から読み出すことができます。

● A/D 変換動作を強制的に停止させる方法

強制停止ビット (ADCS2.BUSY) で設定します。

動作	強制停止ビット (BUSY)
A/D 変換動作を強制的に停止させるには	"0" を書き込む

強制停止ビット (BUSY) に、"1" を書き込んでも、A/D の動作に影響はありません。

● 割込み関連レジスタ

割込みレベル、ベクタの関係は下表のとおりです。

割込みレベル、割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

割込みベクタ	割込みレベル設定ビット
#40 アドレス : FFFF5C _H	割込み制御レジスタ 14 (ICR14) アドレス : 0000BE _H

● 割込みの種類

割込み要因は A/D 変換終了のみです。選択のビットはありません。

● 割込みを許可 / 禁止 / クリアする方法

割込み許可は、割込み要求許可ビット (ADCS2.INTE) で設定します。

制御内容	割込み要求許可ビット (INTE)
割込み要求を禁止するには	"0" にする
割込み要求を許可するには	"1" にする

割込み要求のクリアは、割込み要求ビット (ADCS2.INT) で設定します。

制御内容	割込み要求ビット (INT)
割込み要求をクリアするには	"0" を書き込む または、A/D を起動する

第18章

拡張 I/O シリアル インタフェース

拡張 I/O シリアルインタフェースの概要，構成，レジスタの構成と機能，割込みおよび動作について説明します。

- 18.1 拡張 I/O シリアルインタフェースの概要
- 18.2 拡張 I/O シリアルインタフェースの構成
- 18.3 拡張 I/O シリアルインタフェースのレジスタの構成と機能
- 18.4 拡張 I/O シリアルインタフェースの割込み
- 18.5 拡張 I/O シリアルインタフェースの動作
- 18.6 拡張 I/O シリアルインタフェースのプログラム例

18.1 拡張 I/O シリアルインタフェースの概要

拡張 I/O シリアルインタフェースは、8 ビット×1 チャネル構成のクロック同期式によるデータ転送を行うシリアル I/O インタフェースです。また、データ転送において、LSB ファースト /MSB ファーストの選択ができます。

■ 拡張 I/O シリアルインタフェースの概要

拡張 I/O シリアルインタフェースには、以下の 2 種類の動作モードがあります。

- 内部シフトクロックモード：内部クロックに同期してデータ転送を行います。
- 外部シフトクロックモード：外部端子 (SCK) から入力されるクロックに同期してデータ転送を行います。このモードで外部端子 (SCK) を共有している汎用ポートを操作することにより、CPU の命令による転送動作も可能です。

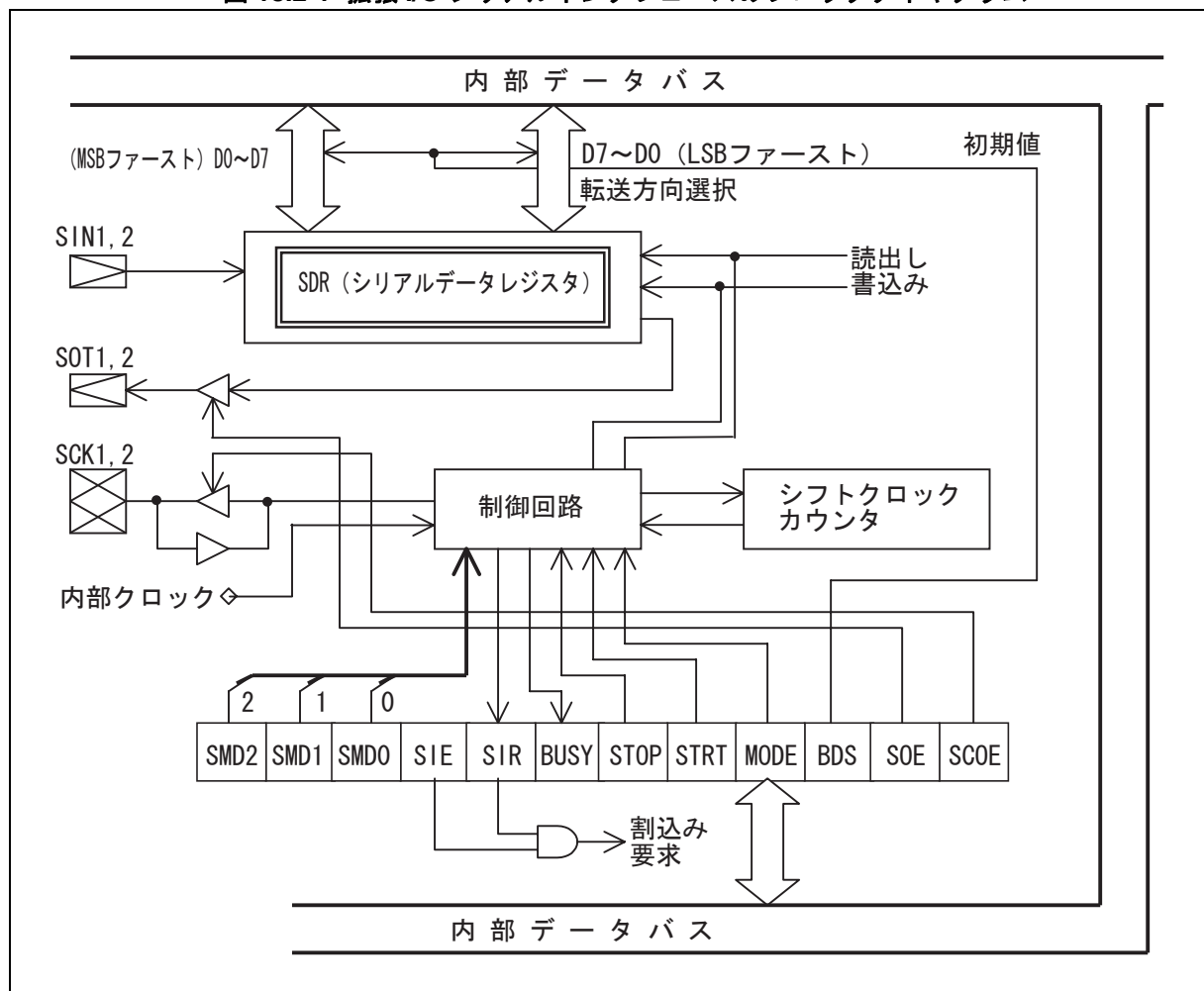
18.2 拡張 I/O シリアルインタフェースの構成

拡張 I/O シリアルインタフェースは、シリアルモードコントロールステータスレジスタ、シリアルデータレジスタから構成されています。

■ 拡張 I/O シリアルインタフェースのブロックダイアグラム

図 18.2-1 に、拡張 I/O シリアルインタフェースのブロックダイアグラムを示します。

図 18.2-1 拡張 I/O シリアルインタフェースのブロックダイアグラム



■ 拡張 I/O シリアルインタフェースに関する端子

拡張 I/O シリアルインタフェースに関する端子は、SIN1/SOT1/SCK1、SIN2/SOT2/SCK2 端子の 2 チャンネルがあります。SIN1/2 端子がシリアル入力ポート、SOT1/2 端子がシリアル出力ポート、SCK1/2 端子が外部クロック入力ポートとして機能します。SIN1、SCK1、SIN2、SCK2 端子は、汎用入出力ポート (P90/SIN1、P92/SCK1、P40/SIN2、P42/SCK2) と、拡張 I/O シリアルインタフェースの入力端子として機能を兼用しており、SOT1、SOT2 端子は、汎用入出力ポート (P91/SOT1、P41/SOT2) と、拡張 I/O シリアルインタフェースの出力端子を兼用しています。

● SIN1/SCK1/SIN2/SCK2 端子として使用する場合の設定

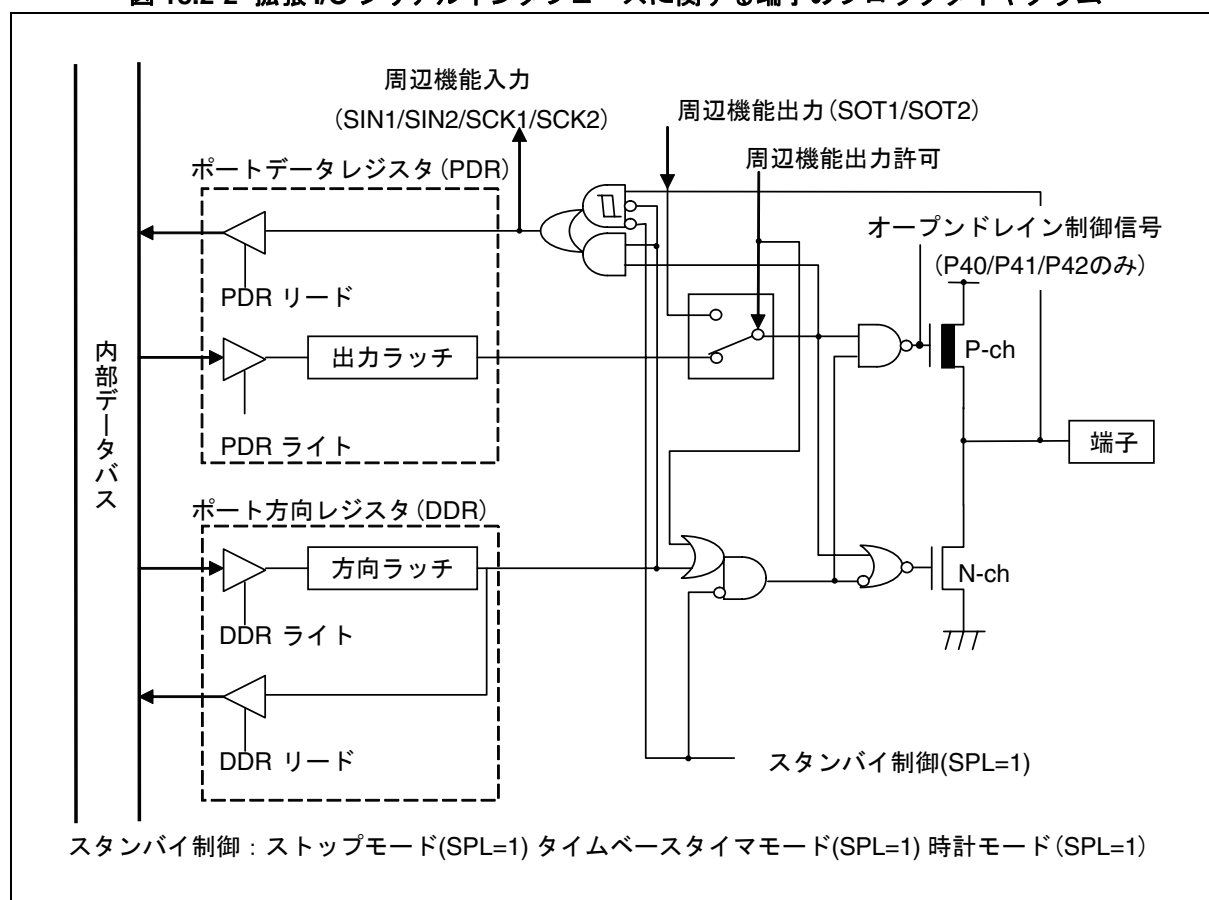
拡張 I/O シリアルインタフェースで SIN1/SCK1/SIN2/SCK2 を入力として使用する場合は、P90/SIN1, P92/SCK1, P40/SIN2, P42/SCK2 端子は、ポート方向レジスタにより入力ポート (DDR9 bit8, bit10 → "0" DDR4 bit0, bit2 → "0") に設定してください。

● SOT1/SOT2 端子として使用する場合の設定

拡張 I/O シリアルインタフェースで SOT1/SOT2 をデータ出力端子として使用する場合は、シリアルモードコントロールステータスレジスタ 0/1(SMCS0/1) をシリアル出力許可 (SOE bit1 → "1") に設定してください。

■ 拡張 I/O シリアルインタフェースに関する端子のブロックダイアグラム

図 18.2-2 拡張 I/O シリアルインタフェースに関する端子のブロックダイアグラム



18.3 拡張 I/O シリアルインタフェースのレジスタの構成と機能

拡張 I/O シリアルインタフェースで使用するレジスタの構成と機能について説明します。

■ 拡張 I/O シリアルインタフェースのレジスタ一覧

図 18.3-1 に、拡張 I/O シリアルインタフェースのレジスタ一覧を示します。

図 18.3-1 拡張 I/O シリアルインタフェースのレジスタ一覧

ch. 0	アドレス: 000027 _H	15	14	13	12	11	10	9	8	シリアルモードコントロール ステータスレジスタ0/1 (SMCS0/SMCS1)
ch. 1	00002B _H	SMD2	SMD1	SMD0	SIE	SIR	BUSY	STOP	STRT	
ch. 0	アドレス: 000026 _H	7	6	5	4	3	2	1	0	シリアルデータレジスタ0/1 (SDR0/SDR1)
ch. 1	00002A _H	—	—	—	—	MODE	BDS	SOE	SCOE	
ch. 0	アドレス: 000028 _H	7	6	5	4	3	2	1	0	通信プリスケラコントロール レジスタ0/1 (SDCR0/SDCR1) リード/ライト
ch. 1	00002C _H	D7	D6	D5	D4	D3	D2	D1	D0	
ch. 0	アドレス: 000029 _H	15	14	13	12	11	10	9	8	
ch. 1	00002D _H	MD	—	—	—	DIV3	DIV2	DIV1	DIV0	
		(R/W)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	

18.3.1 シリアルモードコントロールステータスレジスタ 0/1 (SMCS0/SMCS1)

シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1) の構成と機能について説明します。

■ シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1)

シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1) は、シリアル I/O の転送動作モードを制御するレジスタです。

下図に、シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1) のビット構成を示します。

SMCS0/SMCS1		15	14	13	12	11	10	9	8	初期値
ch. 0	アドレス: 000027 _H	SMD2	SMD1	SMD0	SIE	SIR	BUSY	STOP	STRT	00000010 _B
ch. 1	00002B _H	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	

SMCS0/SMCS1		7	6	5	4	3	2	1	0	初期値
ch. 0	アドレス: 000026 _H	—	—	—	—	MODE	BDS	SOE	SCOE	----0000 _B
ch. 1	00002A _H					R/W	R/W	R/W	R/W	

以下に、シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1) の各ビットの機能について説明します。

[bit15, bit14, bit13] SMD2, SMD1, SMD0: Serial Shift Clock Mode(シフトクロック選択)

これらのビットは、シリアルシフトクロックモードを選択します。

シリアルシフトクロックモードの設定内容を、表 18.3-1 に示します。

表 18.3-1 シリアルシフトクロックモードの設定内容

シリアルシフトクロックモードの選択

SMD2	SMD1	SMD0	φ=16MHz div=8	φ=8MHz div=4	φ=4MHz div=4	分周値
0	0	0	1MHz	1MHz	500kHz	2
0	0	1	500kHz	500kHz	250kHz	4
0	1	0	125kHz	125kHz	62.5kHz	16
0	1	1	62.5kHz	62.5kHz	31.2kHz	32
1	0	0	31.2kHz	31.2kHz	15.6kHz	64
1	0	1	外部シフトクロックモード			
1	1	0	reserved			
1	1	1	reserved			

通信プリスケアラ（SDCR）の設定

Div	(マシンクロック)					推奨マシン サイクル
	MD	DIV3	DIV2	DIV1	DIV0	
1	1	0	0	0	0	2MHz
2	1	0	0	0	1	4MHz
3	1	0	0	1	0	6MHz
4	1	0	0	1	1	8MHz
5	1	0	1	0	0	10MHz
6	1	0	1	0	1	12MHz
7	1	0	1	1	0	14MHz
8	1	0	1	1	1	16MHz

リセットにより"000"に初期化されます。このビットは、転送中の書換えは禁止です。シフトクロックは、内部シフトクロック 5 種類と、外部シフトクロックが選択できます。SMD2, SMD1, SMD0=110_B, 111_B はリザーブですので設定しないでください。クロック選択で SCOE=0 とし、SCK1, 2 端子を共有しているポートを操作することによって、命令ごとにシフト動作させることもできます。

[bit12] SIE:Serial I/O Interrupt Enable(シリアル I/O 割込み許可)

このビットは、シリアル I/O の割込み要求を、下表に示すように制御します。

0	シリアル I/O 割込み禁止 (初期値)
1	シリアル I/O 割込み許可

- ・ リセットにより "0" に初期化されます。
- ・ このビットは、読出し / 書込みが可能です。

[bit11] SIR:Serial I/O Interrupt Request(シリアル I/O 割込み要求)

このビットは、シリアルデータの転送が終了すると"1"にセットされ、割込み許可時 (SIE=1) にこのビットが"1"になると、CPU へ割込み要求が発生します。クリア条件は、MODE ビットの設定によって異なります。

- ・ MODE ビットが"0"のとき、SIR ビットへの"0"書込みによってクリアされます。
- ・ MODE ビットが"1"のとき、SDR の読出し、または書込み動作によってクリアされます。
- ・ MODE ビットの値にかかわらずリセット、または STOP ビットへの"1"書込み動作によってクリアされます。
- ・ このビットに"1"を書き込んでも意味がありません。
- ・ リードモディファイライト系命令の読出し時は、常に、"1"が読み出されます。

[bit10] BUSY(転送状態表示)

このビットは、シリアル転送が実行中か否かを示すビットです。

BUSY	動作
0	停止、またはシリアルデータレジスタ R/W 待機状態 (初期値)
1	シリアル転送状態

- ・ リセットにより "0" に初期化されます。
- ・ このビットは、読出しのみ可能です。

[bit9] STOP(ストップビット)

このビットは、シリアル転送を強制的に中断させるビットです。このビットを"1"にすると、停止状態になります。

STOP	動作
0	通常動作
1	STOP=1 による転送停止 (初期値)

- ・ リセットにより "1" に初期化されます。
- ・ このビットは、読出し / 書込みが可能です。

[bit8] STRT:Start(スタートビット)

このビットは、シリアル転送を起動するためのビットです。停止状態で "1" を書き込むことによって転送が開始されます。シリアル転送動作中およびシリアルシフトレジスタ R/W 待機中の "1" の書込みは無視されます。

- "0" の書込みは意味がありません。
- 読出し時には、常に "0" が読み出されます。

[bit3] MODE(シリアルモード選択)

このビットは、停止状態からの起動条件を選択します。ただし、動作中の書換えは禁止です。

MODE	動作
0	STRT=1 にすることにより起動します。 (初期値)
1	シリアルデータレジスタの読出し / 書込みにより起動します。

- リセットによって "0" に初期化されます。
- 読出し / 書込みが可能です。
- μ DMA を起動する際には、"1" にしておいてください。

[bit2] BDS:Bit Direction Select(転送方向選択)

このビットは、シリアルデータ入出力時に、最下位ビット側から先に転送するか(LSB ファースト)、最上位ビット側から先に転送するか(MSB ファースト)を選択します。

0	LSB ファースト (初期値)
1	MSB ファースト

- リセットによって "0" に初期化されます。
- このビットは、読出し / 書込み可能です。

<注意事項>

転送方向の選択は、SDR にデータを書き込む前に、設定しておいてください。

[bit1] SOE:Serial Out Enable(シリアル出力許可)

このビットは、シリアル I/O 用出力外部端子 (SOT1, 2) の出力を制御します。

0	汎用ポート端子 (初期値)
1	シリアルデータ出力

- リセットによって "0" に初期化されます。
- このビットは、読出し / 書込みが可能です。

[bit0] SCOE:SCK1 Output Enable(シフトクロック出力許可)

このビットは、シフトクロック用入出力外部端子 (SCK1, 2) の出力を制御します。

0	汎用ポート端子, 命令ごと転送時 (初期値)
1	シフトクロック出力端子

外部シフトクロックモードで命令ごとに転送を行うときは、"0" に設定してください。

- リセットによって、"0" に初期化されます。
- このビットは、読出し / 書込みが可能です。

18.3.2 シリアルデータレジスタ 0/1(SDR0/SDR1)

シリアルデータレジスタ 0/1(SDR0/SDR1) の構成と機能について説明します。

■ シリアルデータレジスタ 0/1(SDR0/SDR1)

下図に、シリアルデータレジスタ 0/1(SDR0/SDR1) のビット構成を示します。

SDR0/SDR1		7	6	5	4	3	2	1	0	初期値
ch. 0	アドレス: 000028 _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
ch. 1	00002C _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	(不定)

シリアルデータレジスタ 0/1(SDR0/SDR1) は、シリアル I/O の転送データを保持するシリアルデータレジスタです。

転送中の SDR への書込みおよび読出しは禁止です。

18.3.3 通信プリスケアラコントロールレジスタ 0/1 (SDCR0/SDCR1)

通信プリスケアラコントロールレジスタ 0/1(SDCR0/SDCR1) の構成と機能について説明します。

■ 通信プリスケアラコントロールレジスタ 0/1(SDCR0/SDCR1)

下図に、通信プリスケアラコントロールレジスタ 0/1(SDCR0/SDCR1) のビット構成を示します。

SDCR0/SDCR1	7	6	5	4	3	2	1	0	初期値
ch. 0 アドレス: 000029 _H	MD	—	—	—	DIV3	DIV2	DIV1	DIV0	0—0000 _B
ch. 1 アドレス: 00002D _H	R/W	—	—	—	R/W	R/W	R/W	R/W	

以下に、通信プリスケアラコントロールレジスタ 0/1(SDCR0/SDCR1) の各ビットの機能を説明します。

[bit15] MD: Machine clock divide moDe select

このビットは、通信プリスケアラの動作を許可するビットです。

0	通信プリスケアラは停止します。
1	通信プリスケアラは動作します。

[bit11, bit10, bit9, bit8] DIV3, DIV2, DIV1, DIV0: DIVide3 ~ 0

これらのビットは、マシンのクロックの分周比を決定します

DIV3 ~ DIV0	分周比
0000 _B	1 分周
0001 _B	2 分周
0010 _B	3 分周
0011 _B	4 分周
0100 _B	5 分周
0101 _B	6 分周
0110 _B	7 分周
0111 _B	8 分周

<注意事項>

分周比を変えた場合には、クロックの安定時間として 2 分周分の時間をおいてから通信を行ってください。

18.4 拡張 I/O シリアルインタフェースの割込み

拡張 I/O シリアルインタフェースの割込みは、データの転送終了時に発生します。拡張 I/O シリアルインタフェースの割込みでは、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができます。

■ 拡張 I/O シリアルインタフェースの割込み

拡張 I/O シリアルインタフェースの割込み制御ビットと割込み要因を下記の表に示します。

	シリアル I/O 割込み
割込み要求フラグ	SMCS0:SIR (bit11) ch.0 SMCS1:SIR (bit11) ch.1
割込み要求出力許可ビット	SMCS0:SIE (bit12) ch.0 SMCS1:SIE (bit12) ch.1
割込み発生要因	シリアルデータの転送終了

■ 拡張 I/O シリアルインタフェースに関する割込み要因

拡張 I/O シリアルインタフェースの割込みはデータの転送終了時に発生します。シリアルモードコントロールステータスレジスタ (SMCS) の SIE (bit12) フラグがセットされ、SIE(bit11): 割込み許可が "1" の場合に、CPU へ割込み要求を行います。

■ 拡張 I/O シリアルインタフェースの割込みと DMA 転送および EI²OS

表 18.4-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 18.4-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
SIO1	○	13	#37	FFFF68 _H	ICR13	0000BD _H
SIO2	○	14	#38	FFFF64 _H		

○：割込み要求フラグはクリアされます。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

拡張 I/O シリアルインタフェースは DMA 転送機能、および EI²OS 機能に対応しています。DMA または EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

18.5 拡張 I/O シリアルインタフェースの動作

拡張 I/O シリアルインタフェースは、シリアルモードコントロールステータスレジスタ (SMCS) およびシリアルデータレジスタ (SDR) によって構成され、8 ビットのシリアルデータの入出力に使用します。

本節では、拡張 I/O シリアルインタフェースの動作について説明します。

■ 拡張 I/O シリアルインタフェースの動作概要

シリアルデータの入出力は、入力、出力それぞれが以下のように動作します。

● シリアルデータの入力

シリアルシフトクロック (外部クロック、内部クロック) の立上りに同期してシリアル入力端子 (SIN1 端子) から SDR (シリアルデータレジスタ) に入力されます。

シフトの方向 (MSB からの転送、または LSB からの転送) は、SMCS (シリアルモードコントロールステータスレジスタ) の方向指定ビット (BDS) により指定します。

転送が終了すると、シリアルモードコントロールステータスレジスタ (SMCS) の MODE ビットによって停止状態、またはデータレジスタ R/W 待機状態に入ります。各々の状態から転送状態に移るには、それぞれ次の設定を行います。

- 停止状態から復帰するときは、STOP ビットに "0"、STRT ビットに "1" を書き込んでください (STOP と STRT は同時設定が可能です)。
- 待機状態から復帰するときは、データレジスタの読出し、または書込みをしてください。

● シリアルデータの出力

シフトレジスタの内容がシリアルシフトクロック (外部クロック、内部クロック) の立下りに同期してシリアル出力端子 (SOT1 端子) から出力されます。

18.5.1 シフトクロックモード

シフトクロックには、内部シフトクロックモードおよび外部シフトクロックモードの 2 種類があり、SMCS の設定で指定します。モードの切換えは、シリアル I/O が停止した状態で行ってください。停止状態は、BUSY ビットを読み出すことによって確認できます。

■ 内部シフトクロックモード

内部クロックを使用し、同期タイミング出力用としてデューティ比 50% のシフトクロックが SCK 端子より出力されます。

データが、1 クロックごとに 1 ビット転送されます。

転送の速度は、次式によって算出できます。

$$\text{転送速度 (S)} = \frac{A}{\text{内部クロックマシンサイクル (A)}}$$

A は、SMCS の SMD ビットで選択される以下の分周比です。

$$(\phi \div \text{div})/2, (\phi \div \text{div})/2^2, (\phi \div \text{div})/2^4, (\phi \div \text{div})/2^5, (\phi \div \text{div})/2^6$$

■ 外部シフトクロックモード

SCK 端子から入力される外部シフトクロックに同期して、データが 1 クロックごとに 1 ビット転送されます。

転送速度は、DC から 1/(8 マシンサイクル) まで可能です。例えば、1 マシンサイクル = 62.5ns のときは、2MHz まで可能となります。

命令ごとに転送することもできます。また、以下の設定をすることによっても実現できます。

- 外部シフトクロックモードを選択し、SMCS の SCOE ビットを "0" に設定します。
- SCK 端子を共有しているポートの方向レジスタに "1" を書込み、ポートを出力モードに設定します。

以上の設定をした後、ポートデータレジスタ (PDR) に "1", "0" を書き込むと、SCK 端子に出力されるポートの値を外部クロックとして取り込み、転送動作を行います。シフトクロックは、"H" から開始させてください。

<注意事項>

シリアル I/O 動作中の SMCS, SDR への書込みは禁止です。

18.5.2 シリアル I/O の動作状態

シリアル I/O の動作状態として、STOP 状態、停止状態、SDR の R/W 待機状態および転送状態の 4 種類の状態があります。

■ シリアル I/O の動作状態

● STOP 状態

RESET 時、または SMCS の STOP ビットに "1" が書き込まれたときにシフトカウンタは初期化され、SIR=0 となります。

STOP 状態からの復帰は、STOP=0, STRT=1(同時設定可)とすることによって行われます。STOP ビットは STRT ビットよりも優先度の高いビットですので、STOP=1 のとき STRT=1 としても転送動作は行われません。

● 停止状態

MODE ビットが "0" のとき、転送が終了したことによって SMCS が BUSY=0, SIR=1 となり、カウンタが初期化されて停止状態に入ります。停止状態からの復帰は、STRT=1 により行われ、転送動作が再開されます。

● シリアルデータレジスタ R/W 待機状態

SMCS の MODE ビットが "1" のとき、シリアル転送が終了すると、BUSY=0, SIR=1 となりシリアルデータレジスタ R/W 待機状態に入ります。割込み許可レジスタが許可状態であれば、本ブロックより割込み信号が出力されます。

R/W 待機状態からの復帰は、シリアルデータレジスタが読み出されたり、書き込まれたりすると、BUSY=1 となり転送動作が再開されます。

● 転送状態

BUSY=1 はシリアル転送を行っている状態を示しています。MODE ビットにより、それぞれ停止状態および R/W 待機状態に遷移します。

図 18.5-1 に各状態の動作遷移図を示し、図 18.5-2 にシリアルデータレジスタの読出し / 書き込みの概念図を示します。

図 18.5-1 I/O 拡張シリアルインタフェースの動作遷移図

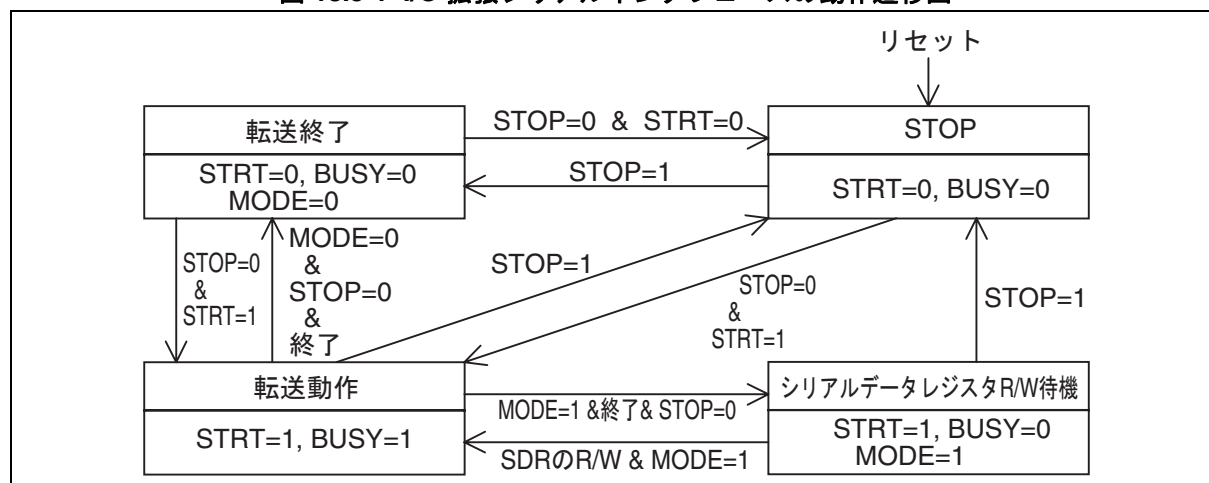
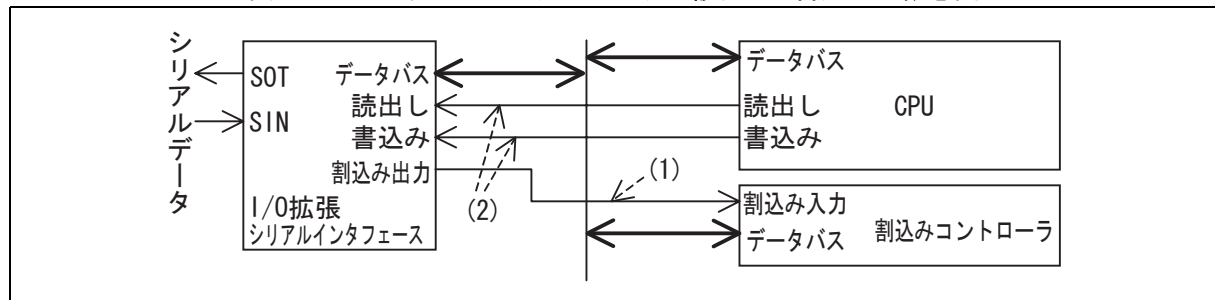


図 18.5-2 シリアルデータレジスタの読出し / 書込みの概念図



- (1) **MODE=1** のとき, シフトクロックカウンタによって転送終了し, **SIR=1** となって読出し / 書込み待機状態に入ります。 **SIE** ビットが "1" であれば, 割込み信号を発生します。ただし, **SIE** がインアクティブのときや **STOP** ビットへ "1" を書き込むことによる転送中断のときは, 割込み信号は発生しません。
- (2) シリアルデータレジスタが読出し / 書込みされると, 割込み要求はクリアされ, シリアル転送が開始されます。

18.5.3 シフト動作のスタート/ストップタイミングと入出力のタイミング

シフト動作のスタート/ストップタイミングと入出力のタイミングについて説明します。

■ シフト動作のスタート/ストップタイミングと入出力のタイミング

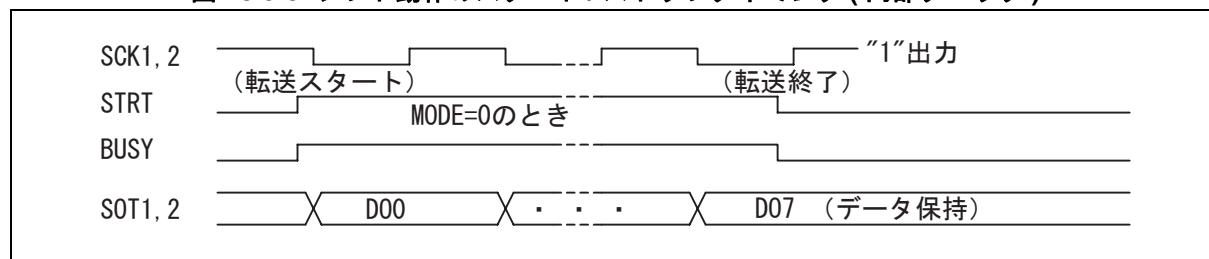
- スタート
SMCS の STOP ビットを "0", STRT ビット "1" に設定します。
- ストップ
転送終了によって停止する場合と, STOP=1 によって停止する場合があります。
 - STOP=1 による停止 :MODE ビットにかかわらず SIR=0 のまま停止します。
 - 転送終了による停止 :MODE ビットにかかわらず, SIR=1 となり停止します。

BUSY ビットは, MODE ビットにかかわらずシリアル転送状態の場合 "1" となり, 停止状態, または R/W 待機状態の場合 "0" となります。転送状態を確認したいときは, このビットを読み出してください。

以下に, 各モードにおける転送動作のタイミングチャートを示します。図中の D07 ~ D00 は, 出力データを示します。

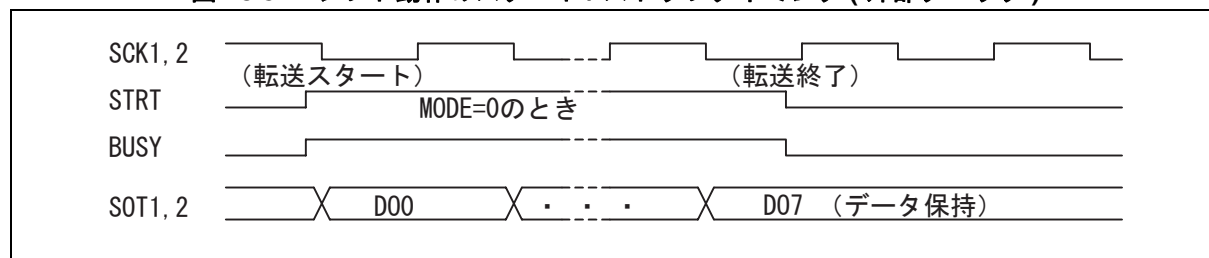
● 内部シフトクロックモード (LSB ファースト)

図 18.5-3 シフト動作のスタート/ストップタイミング (内部クロック)



● 外部シフトクロックモード (LSB ファースト)

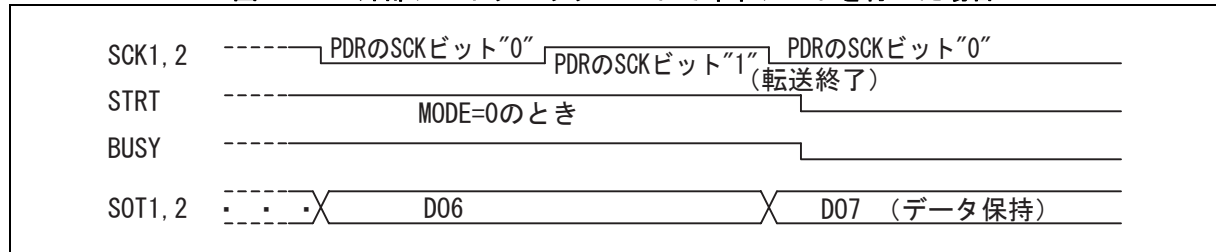
図 18.5-4 シフト動作のスタート/ストップタイミング (外部クロック)



● 外部シフトクロックモードで命令シフトを行った場合 (LSB ファースト)

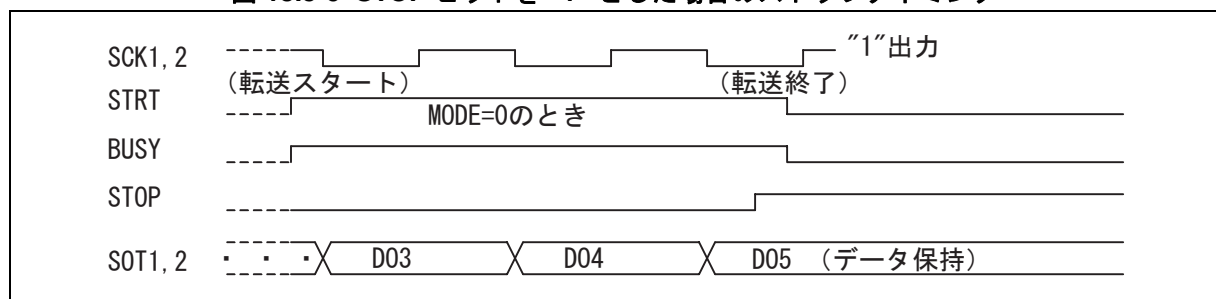
命令シフトでは, PDR の SCK に対応するビットに "1" が書き込まれたとき "H" が出力され, "0" が書き込まれたとき "L" が出力されます (ただし, 外部シフトクロックモード選択で, SCOE=0 のとき)。

図 18.5-5 外部シフトクロックモードで命令シフトを行った場合



● STOP=1 による停止 (LSB ファースト, 内部クロック時)

図 18.5-6 STOP ビットを "1" とした場合のストップタイミング



■ シリアルデータ転送中の動作

シリアルデータの転送中は, シフトクロックの立下りでシリアル出力端子 (SOT2) からのデータが出力され, 立上りでシリアル入力端子 (SIN) のデータが入力されます。

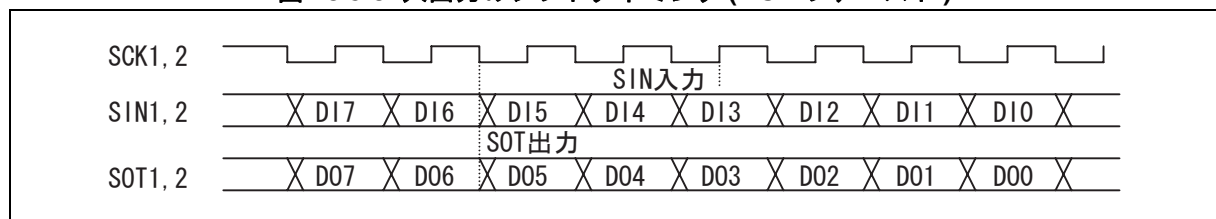
● LSB ファースト (BDS ビットが "0" のとき)

図 18.5-7 入出力のシフトタイミング (LSB ファースト)



● MSB ファースト (BDS ビットが "1" のとき)

図 18.5-8 入出力のシフトタイミング (MSB ファースト)



18.5.4 割込み機能

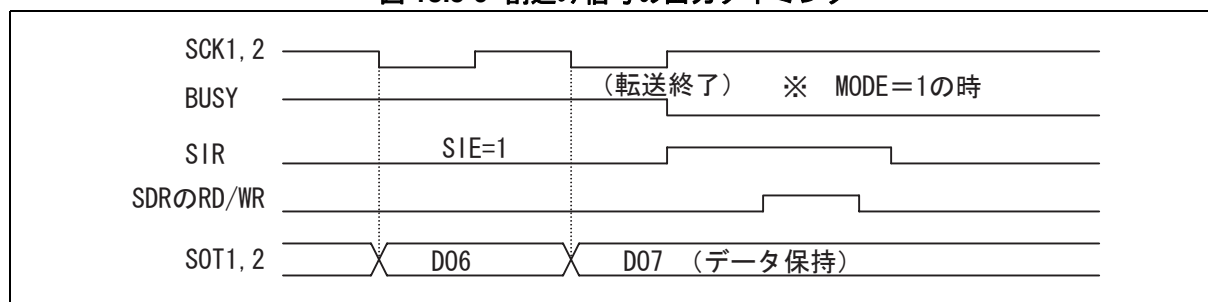
拡張 I/O シリアルインタフェースは、CPU に対し割込み要求を発生します。

■ 拡張 I/O シリアルインタフェースの割込み機能

データの転送終了時に割込みフラグである SIR ビットがセットされ、割込みを許可する SMCS の SIE ビットが "1" のとき、CPU へ割込み要求が出力されます。

図 18.5-9 に、割込み信号の出力タイミングを示します。

図 18.5-9 割込み信号の出力タイミング



18.6 拡張 I/O シリアルインタフェースのプログラム例

拡張 I/O シリアルインタフェースのプログラム例を示します。

■ 拡張 I/O シリアルインタフェースのプログラム例

<p>設定手順例</p> <p>1byte のデータを sio1(ch.0) で送信する。</p> <p>< 初期設定 ></p> <p>・ SIO の制御</p> <table> <tr> <th>レジスタ名</th> <th>ビット名</th> </tr> <tr> <td>モードレジスタの設定</td> <td>SMCS0</td> </tr> <tr> <td>シフトクロックの選択 ></td> <td>.SMD2-0</td> </tr> <tr> <td>割込み許可の設定 ></td> <td>.SIE</td> </tr> <tr> <td>割込み要求の設定 ></td> <td>.SIR</td> </tr> <tr> <td>転送状態表示 ></td> <td>.BUSY</td> </tr> <tr> <td>SOT1,2 端子の設定 ></td> <td>.SOE</td> </tr> <tr> <td>SCK1,2 端子の設定 ></td> <td>.SCOE</td> </tr> <tr> <td>制御レジスタの設定</td> <td>SDCR0</td> </tr> <tr> <td>通信プリスケアラの設定 ></td> <td>.MD</td> </tr> <tr> <td>分周比の設定 >></td> <td>.DIV3-0</td> </tr> </table> <p>・ 割込み関連</p> <table> <tr> <td>割込みレベルの設定</td> <td>ICR13</td> </tr> <tr> <td>I フラグの設定</td> <td>(CCR)</td> </tr> </table> <p>< 起動 ></p> <p>・ SIO 起動</p> <table> <tr> <th>レジスタ名</th> <th>ビット名</th> </tr> <tr> <td>任意のデータを送信</td> <td>SDR0</td> </tr> <tr> <td>SIO 動作起動</td> <td>SMCS0 .STRT</td> </tr> </table> <p>< 割込み ></p> <p>・ 送信割込み処理</p> <table> <tr> <th>レジスタ名</th> <th>ビット名</th> </tr> <tr> <td>割込みフラグ初期化</td> <td>SMCS0 .SIR</td> </tr> </table> <p>< 割込みベクタ ></p> <p>・ ベクタテーブルの設定</p> <p>(注意事項)</p> <p>事前にクロック関連の設定および, __set_il(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	レジスタ名	ビット名	モードレジスタの設定	SMCS0	シフトクロックの選択 >	.SMD2-0	割込み許可の設定 >	.SIE	割込み要求の設定 >	.SIR	転送状態表示 >	.BUSY	SOT1,2 端子の設定 >	.SOE	SCK1,2 端子の設定 >	.SCOE	制御レジスタの設定	SDCR0	通信プリスケアラの設定 >	.MD	分周比の設定 >>	.DIV3-0	割込みレベルの設定	ICR13	I フラグの設定	(CCR)	レジスタ名	ビット名	任意のデータを送信	SDR0	SIO 動作起動	SMCS0 .STRT	レジスタ名	ビット名	割込みフラグ初期化	SMCS0 .SIR	<p>プログラム例</p> <pre> void sio_sample(void) { sio_initial(); sio_start(); } void sio_initial(void) { IO_SMCS0.word = 0x3003; /* 設定値 =0011_0000_0000_0011 */ /* bit15-13 = 001 4 分周 */ /* bit12 = 1 割込み許可 */ /* bit11 = 0 割込み要求クリア */ /* bit10 = 0 R/W 待機状態 */ /* bit1 = 1 SOE シリアル出力許可 */ /* bit0 = 1 SCOE シフトクロック出力許可 */ IO_SDCR0.byte = 0x83; /* 設定値 =1000_0011 */ /* bit15 = 1 通信プリスケアラ許可 */ /* bit11-8 = 11 4 分周 */ IO_ICR13.byte = 0x10; /* SIO 送信完了割込みレベル設定 (値は任意) */ __EI(); /* 割込み許可 */ } void sio_start(void) { IO_SDR0 = 0xaa; /* 任意のデータ値を送信する。 */ IO_SMCS0.bit.STRT = 1; /* bit1 = 1 SIO 動作許可 */ } __interrupt void sio_int(void) { IO_SMCS0.bit.SIR = 0; /* bit0 = 0 SIR 割込みフラグ初期化 */ } #pragma intvect sio_int 37 </pre> <p>(注意事項)</p> <p>レジスタの記述形式については, 「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
レジスタ名	ビット名																																				
モードレジスタの設定	SMCS0																																				
シフトクロックの選択 >	.SMD2-0																																				
割込み許可の設定 >	.SIE																																				
割込み要求の設定 >	.SIR																																				
転送状態表示 >	.BUSY																																				
SOT1,2 端子の設定 >	.SOE																																				
SCK1,2 端子の設定 >	.SCOE																																				
制御レジスタの設定	SDCR0																																				
通信プリスケアラの設定 >	.MD																																				
分周比の設定 >>	.DIV3-0																																				
割込みレベルの設定	ICR13																																				
I フラグの設定	(CCR)																																				
レジスタ名	ビット名																																				
任意のデータを送信	SDR0																																				
SIO 動作起動	SMCS0 .STRT																																				
レジスタ名	ビット名																																				
割込みフラグ初期化	SMCS0 .SIR																																				

■ プログラム例以外の設定方法

● 動作クロックの種類と選択方法

動作クロックには内蔵タイマと外部クロックの 2 種類があります。

シフトクロック選択ビット (SMCS0.SMD[2:0], SMCS1.SMD[2:0]) で設定します。

制御内容	シフトクロック選択ビット (SMD[2:0])
内蔵タイマを選択するには	"000 _B " ~ "100 _B " する
外部クロックを選択するには	"101 _B " にする

● SCK 端子, SIN 端子, SOT 端子を制御する方法

SIO1, SIO2 で設定します。

動作	SIO1	SIO2
SCK 端子を入力にするには	DDR9.P92 = 0	DDR4.P42 = 0
SCK 端子を出力にするには	SMCS0.SCOE = 1	SMCS1.SCOE = 1
SIN 端子入力にするには	DDR9.P90 = 0	DDR4.P40 = 0
SOT 端子を出力にするには	SMCS0.SOE = 1	SMCS1.SOE = 1

● SIO の動作を許可 / 停止する方法

スタートビット (SMCS0.STRT, SMCS1.STRT) で設定します。

制御内容	スタートビット (STRT)
シリアル転送起動	"1" にする

ストップビット (SMCS0.STOP, SMCS1.STOP) で設定します。

制御内容	ストップビット (STOP)
シリアル転送停止	"1" にする

● 転送方向の設定方法

設定方向選択ビット (SMCS0.BDS, SMCS1.BDS) で設定します。

転送方向はどの動作モードでも, LSB/MSB の選択が可能です。

制御内容	設定方向選択ビット (BDS)
LSB 転送 (最下位ビットから) するには	"0" にする
MSB 転送 (最上位ビットから) するには	"1" にする

● 割込み関連レジスタ

SIO 番号, 割込みレベル, ベクタの関係は下表のとおりです。

割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

	割込みベクタ	割込みレベル設定ビット
SIO1	#37 アドレス: FFFF68 _H	割込みレベルレジスタ (ICR13) アドレス: 0000BD _H
SIO2	#38 アドレス: FFFF64 _H	割込みレベルレジスタ (ICR13) アドレス: 0000BD _H

● 割込みの種類

割込みは 1 種類です。シリアルデータの転送終了で発生します。

● 割込みを許可 / 禁止 / クリアする方法

割込みの許可 / 禁止の設定は, 割込み要求許可ビット (SMCS0.SIE, SMCS1.SIE) で設定します。

制御内容	割込み要求許可ビット (SIE)
割込み要求を禁止するには	"0" にする
割込み要求を許可するには	"1" にする

割込み要求のクリアは, 割込み要求フラグ (SMCS0.SIR, SMCS1.SIR) で設定します。

制御内容	割込み要求フラグ (SIR)
割込み要求をクリアするには	"0" にする

第19章

UART

UART の概要，構成，レジスタの構成と機能，割り込み，動作，使用上の注意およびプログラム例について説明します。

19.1 UART の概要

19.2 UART の構成

19.3 UART のレジスタの構成と機能

19.4 UART の割り込み

19.5 UART の動作

19.6 UART の使用上の注意

19.7 UART のプログラム例

19.1 UART の概要

UART は、非同期 (調歩同期) 通信または CLK 同期通信を行うためのシリアル I/O ポートです。

■ UART の特長

UART は、以下の特長をもっています。

- 全二重ダブルバッファ搭載
- 非同期 (調歩同期), CLK 同期通信 (スタートビット / ストップビットなし) が可能
- マルチプロセッサモードのサポート
- 専用ボーレートジェネレータ内蔵
 - 非同期 : 76923/38461/19230/9615/500k/250kbps
 - CLK 同期 : 16M/8M/4M/2M/1M/500kbps
- 外部クロックによる自由なボーレートの設定が可能
- PPG1 から供給される内部クロックを利用可能
- データ長 : 7 ビット長 (非同期ノーマルモードのみ) / 8 ビット長
- マスタ / スレーブ型通信機能 (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
- エラー検出機能 (パリティ , フレーミング , オーバラン)
- 転送信号 : NRZ 符号
- DMA のサポート (受信 / 送信対応)

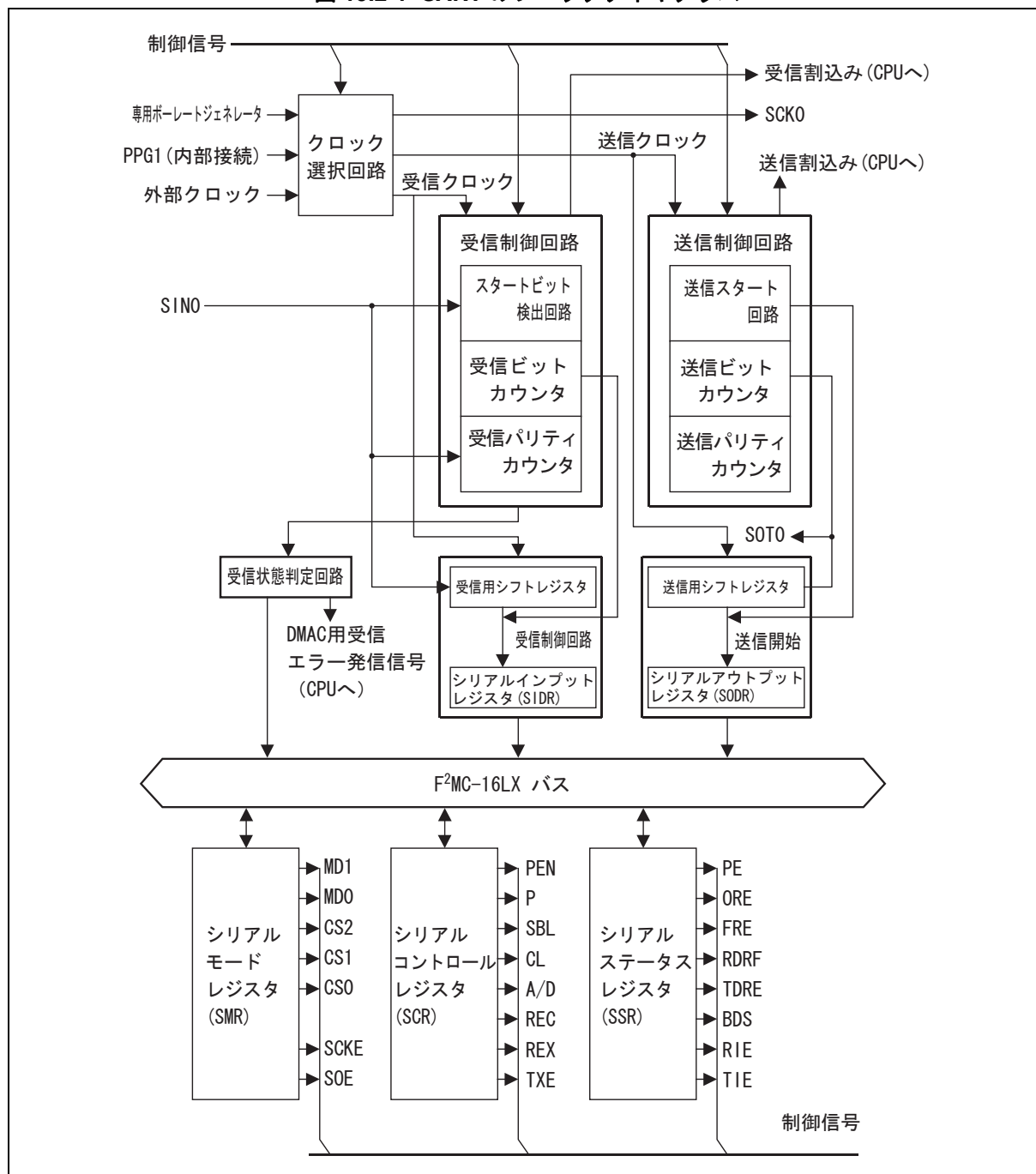
19.2 UART の構成

UART は、シリアルモードレジスタ、シリアルコントロールレジスタ、シリアルステータスレジスタと通信プリスケアラコントロールレジスタ、シリアルインプット/アウトプットレジスタで構成されます。

■ UART のブロックダイアグラム

図 19.2-1 に、UART のブロックダイアグラムを示します。

図 19.2-1 UART のブロックダイアグラム



■ UART に関する端子

UART に関する端子は, SIN0/SOT0/SCK0 端子があります。SIN0 端子がシリアル入力ポート, SOT0 端子がシリアル出力ポート, SCK0 端子が外部クロック入力ポートとして機能します。SIN0, SCK0 端子は, 汎用入出力ポート (P70/SIN0, P72/SCK0) と, UART の入力端子として機能を兼用しており, SOT0 端子は, 汎用入出力ポート (P71/SOT0) と, UART の出力端子を兼用しています。

● SIN0/SCK0 端子として使用する場合の設定

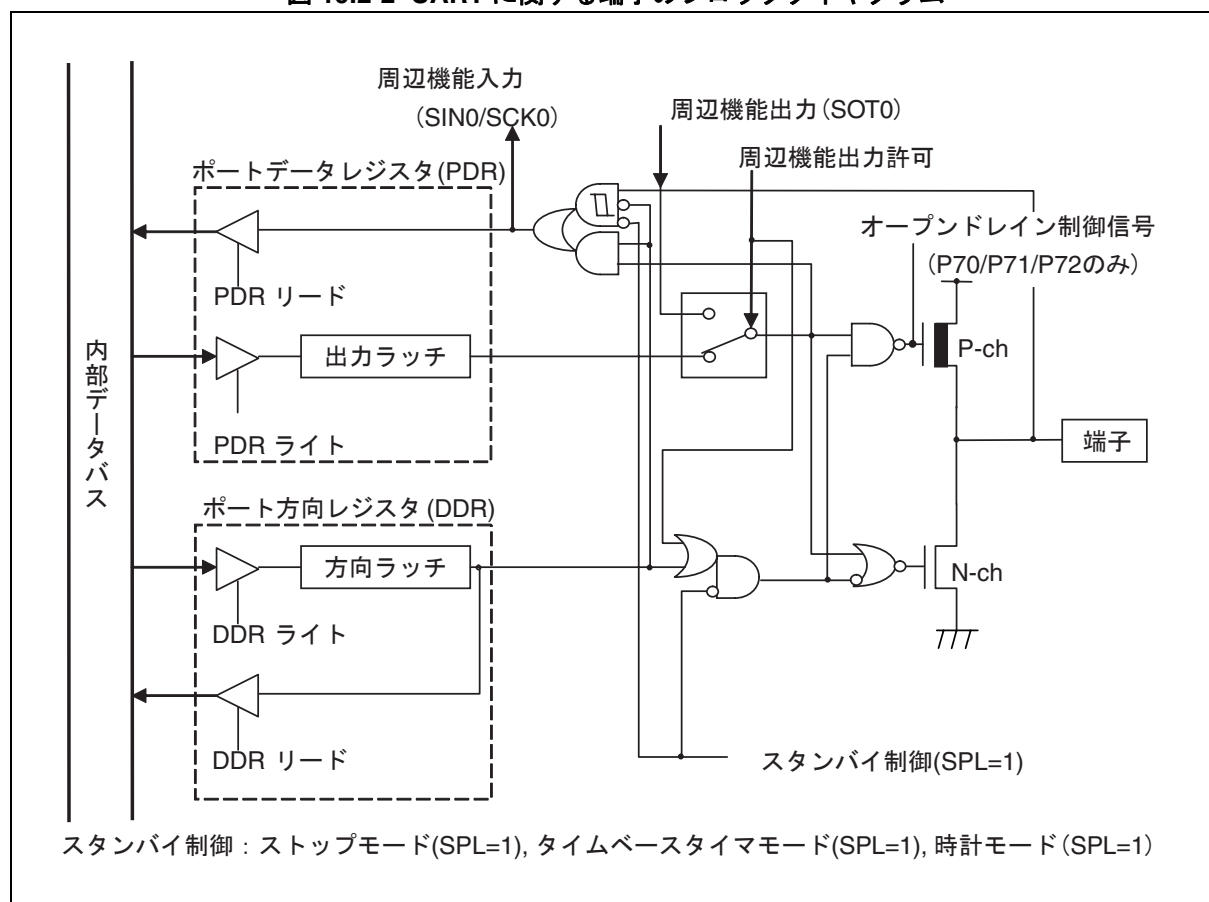
UART で SIN0/SCK0 を入力端子として使用する場合は, P70/SIN0, P72/SCK0 端子は, ポート方向レジスタにより入力ポート (DDR7 bit8, 10 → "0") に設定してください。

● SOT0 端子として使用する場合の設定

UART で SOT0 をデータ出力端子として使用する場合は, シリアルモードレジスタ (SMR) をシリアルデータ出力 (SOE bit0 → "1") に設定してください。

■ UART に関する端子のブロックダイアグラム

図 19.2-2 UART に関する端子のブロックダイアグラム



19.3 UART のレジスタの構成と機能

UART で使用するレジスタの構成および機能について説明します。

■ UART のレジスタ一覧

図 19.3-1 に、UART のレジスタ一覧を示します。

図 19.3-1 UART のレジスタ一覧

15	8				7	0				
SCR					SMR					(R/W)
SSR					SIDR (R) / SODR (W)					(R/W)
CDCR					—					(R/W)
← 8ビット →					← 8ビット →					

7	6	5	4	3	2	1	0		
000020 _H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	シリアルモードレジスタ (SMR)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(0)	(X)	(0)	(0)	初期値

15	14	13	12	11	10	9	8		
000021 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	シリアルコントロールレジスタ (SCR)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	初期値

7	6	5	4	3	2	1	0		
000022 _H	D7	D6	D5	D4	D3	D2	D1	D0	シリアルインプットレジスタ (SIDR) / シリアルアウトプットレジスタ (SODR)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	初期値

15	14	13	12	11	10	9	8		
000023 _H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	シリアルステータスレジスタ (SSR)
	(R)	(R)	(R)	(R)	(R)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	初期値

15	14	13	12	11	10	9	8		
000025 _H	MD	SRST	—	—	DIV3	DIV2	DIV1	DIV0	通信プリスケアラコントロールレジスタ (CDCR)
	(R/W)	(R/W)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(—)	(—)	(0)	(0)	(0)	(0)	初期値

<注意事項>

UART のレジスタを設定する場合、通信モードの設定は、動作停止中に行ってください。
動作中に設定すると、モード設定時に送受信したデータは保証されません。

19.3.1 シリアルモードレジスタ (SMR)

シリアルモードレジスタ (SMR) の構成および機能について説明します。

■ シリアルモードレジスタ (SMR)

下図に、シリアルモードレジスタ (SMR) のビット構成を示します。

	7	6	5	4	3	2	1	0	
000020 _H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	シリアルモードレジスタ (SMR)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(0)	(X)	(0)	(0)	初期値

以下に、シリアルモードレジスタ (SMR) の各ビットの機能を説明します。

[bit7, bit6] MD1, MD0: Mode Select

これらのビットは、UART 動作モードを選択します。

モード	MD1	MD0	動作モード
0	0	0	非同期 (調歩同期) ノーマルモード
1	0	1	非同期 (調歩同期) マルチプロセッサモード
2	1	0	CLK 同期モード
-	1	1	設定禁止

モード 1 の非同期 (調歩同期) マルチプロセッサモードとは、1 台のホスト CPU に数台のスレーブ CPU が接続される場合の使用方法です。本リソースでは、受信データのデータ形式を判別できません。したがって、マルチプロセッサモードのマスタのみをサポートします。また、パリティチェック機能は使用できませんので、SCR レジスタの PEN は "0" に設定してください。

[bit5, bit4, bit3] CS2, CS1, CS0: Clock Select

これらのビットは、ボーレートクロックソースを選択します。専用ボーレートジェネレータを選択した場合には、同時にボーレートも決定されます。

CS2 ~ CS0	クロック入力
000 _B ~ 101 _B	専用ボーレートジェネレータ
110 _B	内部クロック
111 _B	外部クロック

- 内部クロックを選択した場合、MB90480/485 シリーズでは、PPG1 が選択されます。
- 同期転送時に専用ボーレートジェネレータを使用する場合、以下の設定をしないでください。
 - 1) CS2 ~ CS0 = 000_B
 - 2) CS2 ~ CS0 = 001_B, DIV3 ~ DIV0 = 0000_B

[bit2] 予約

このビットは、予約ビットです。

[bit1] SCKE:SCLK Enable

このビットは、CLK 同期モード (モード 2) で通信する場合、SCK0 をクロック入力端子にするかクロック出力端子として使うかを指定します。CLK 非同期モード時、または外部クロックモード時では、"0" に設定してください。

- 0: クロック入力端子として機能します。
- 1: クロック出力端子として機能します。

<注意事項>

クロック入力端子として使用する場合は、外部クロックソースが選択されている必要があります。

[bit0] SOE:Serial Output Enable

このビットは、汎用 I/O ポート端子と兼用されている外部端子 (SOT0) を、シリアル出力端子として使うか、I/O ポート端子として使うかを指定します。

- 0: 汎用 I/O ポート端子として機能します。
- 1: シリアルデータ出力端子 (SOT0) として機能します。

19.3.2 シリアルコントロールレジスタ (SCR)

シリアルコントロールレジスタ (SCR) の構成および機能について説明します。

■ シリアルコントロールレジスタ (SCR)

下図に、シリアルコントロールレジスタ (SCR) のビット構成を示します。

	15	14	13	12	11	10	9	8	
000021 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	シリアルコントロールレジスタ (SCR)
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	初期値

以下に、シリアルコントロールレジスタ (SCR) の各ビットの機能を説明します。

[bit15] PEN:Parity ENable

このビットは、シリアルデータに対して、パリティビットの付加 (送信時) および検出 (受信時) をするかどうかを選択します。

0	パリティなし
1	パリティあり

<注意事項>

パリティを付加できるのは、非同期 (調歩同期) 通信モードのノーマルモード (モード 0) のみです。マルチプロセッサモード (モード 1) および CLK 同期通信 (モード 2) では、パリティを付加できません。

[bit14] P:Parity

このビットは、パリティを付加してデータ通信を行うときの偶数 / 奇数パリティを指定します

0	偶数パリティ
1	奇数パリティ

[bit13] SBL:Stop Bit Length

このビットは、非同期 (調歩同期) 通信を行うときのフレームエンドマークである、ストップビットのビット長を指定します。

0	1 ストップビット
1	2 ストップビット

[bit12] CL:Chracter Length

このビットは、送受信する 1 フレームのデータ長を指定します。

0	7 ビットデータ
1	8 ビットデータ

<注意事項>

7 ビットデータを扱えるのは、非同期（調歩同期）通信のうちノーマルモード（モード 0）のみです。マルチプロセッサモード（モード 1）および CLK 同期モード（モード 2）では、8 ビットデータを指定してください。

[bit11] A/D:Address/Data

このビットは、非同期（調歩同期）通信のマルチプロセッサモード（モード 1）において、送受信するフレームのデータ形式を指定します。

0	データフレーム
1	アドレスフレーム

[bit10] REC:Receiver Error Clear

"0"書込みで、SSRレジスタのエラーフラグ(PE,ORE, FRE)をクリアします。"1"書込みは無効であり、読出し時は、常に "1" となります。

[bit9] RXE:Receiver Enable

このビットは、UART の受信状態を制御します。

0	受信動作を禁止します。
1	受信動作を許可します。

受信中（受信シフトレジスタにデータが入力されているとき）に受信動作を禁止した場合には、そのフレームの受信が完了し、受信データバッファから SDR レジスタに受信データをストアした時点で受信動作が禁止されます。

[bit8] TXE:Transmitter Enable

このビットは、UART の送信状態を制御します。

0	送信動作を禁止します。
1	送信動作を許可します。

送信中（送信レジスタからデータが出力されているとき）に送信動作を禁止した場合には、シリアルアウトプットレジスタ (SODR) に、送信データがなくなった後に送信動作が禁止されます。送信動作は、シリアルアウトプットレジスタ (SODR) に値を書き込むと、内部のシリアルクロックに同期して動作が開始されます。また、送信動作の禁止 (TXE=0) は、TDRE フラグが "0" のときは、無効になります。

19.3.3 シリアルインプット / アウトプットレジスタ (SIDR/ SODR)

シリアルインプット / アウトプットレジスタ (SIDR/SODR) の構成と機能について説明します。

■ シリアルインプット / アウトプットレジスタ (SIDR/SODR)

下図に、シリアルインプット / アウトプットレジスタ (SIDR/SODR) のビット構成を示します。

	7	6	5	4	3	2	1	0	シリアルインプットレジスタ (SIDR) / シリアルアウトプットレジスタ (SODR)
000022 _H	D7	D6	D5	D4	D3	D2	D1	D0	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)	

シリアルインプット / アウトプットレジスタ (SIDR/SODR) は、データ要素が 7 ビット長の場合は、上位 1 ビット (D7) が無効となります。SODR レジスタにデータ要素を書き込むには、SSR レジスタの TDRE に必ず "1" を書き込んでください。

<注意事項>

このアドレスへのデータ要素の書込みは、SODR レジスタへのデータ要素書込みを意味します。このアドレスの読出しは、SIDR レジスタの読出しを意味します。

19.3.4 シリアルステータスレジスタ (SSR)

シリアルステータスレジスタ (SSR) の構成および機能について説明します。

■ シリアルステータスレジスタ (SSR)

下図に、シリアルステータスレジスタ (SSR) のビット構成を示します。

	15	14	13	12	11	10	9	8	
000023 _H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	シリアルステータスレジスタ (SSR)
	(R)	(R)	(R)	(R)	(R)	(R/W)	(R/W)	(R/W)	リード/ライト
	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	初期値

SSR は、UART の状態を表すフラグです。

以下に、シリアルステータスレジスタ (SSR) の各ビットの機能を説明します。

[bit15] PE:Parity Error

このビットは、受信時にパリティエラーが発生したときにセットされる、割込み要求フラグです。一度セットされたフラグをクリアするには、SCR レジスタの REC ビット (bit10) に "0" を書き込みます。このビットがセットされた場合には、SIDR のデータは、無効となります。

0	パリティエラーなし
1	パリティエラーが発生

[bit14] ORE:Over Run Error

このビットは、受信時にオーバランエラーが発生したときにセットされる、割込み要求フラグです。一度セットされたフラグをクリアするには、SCR レジスタの REC ビット (bit10) に "0" を書き込みます。このビットがセットされた場合には、SIDR のデータは、無効となります。

0	オーバランエラーなし
1	オーバランエラーが発生

[bit13] FRE:FRaming Error

このビットは、受信時にフレーミングエラーが発生したときにセットされる、割込み要求フラグです。一度セットされたフラグをクリアするには、SCR レジスタの REC ビット (bit10) に "0" を書き込みます。このビットがセットされた場合には、SIDR のデータは、無効となります。

0	フレーミングエラーなし
1	フレーミングエラーが発生

[bit12] RDRF:Receiver Data Register Full

このビットは、SIDR レジスタに受信データがあることを示す、割込み要求フラグです。SIDR レジスタに受信データがロードされるとセットされ、SIDR レジスタを読み出すと自動的にクリアされます。

0	受信データなし
1	受信データあり

[bit11] TDRE:Transmitter Data Register Empty

このビットは、SODR レジスタに送信データを書き込めることを示す、割込み要求フラグです。SODR レジスタに送信データを書き込むとクリアされます。書き込んだデータが送信用シフトレジスタにロードされて、転送が開始されると再びセットされ、次の送信データを書き込めることを表します。

0	送信データの書き込み禁止
1	送信データの書き込み許可

[bit10] BDS:Bit Direction Select

このビットは、転送方向選択を制御します。

0	シリアルデータを最下位ビット側から転送します (LSB ファースト)。
1	シリアルデータを最上位ビット側から転送します (MSB ファースト)。

<注意事項>

シリアルデータレジスタ (SDR) からの読出し、書込み時にデータの上位側と下位側を入れ替えるために、SDR にデータを書き込んだ後、このビットを書き換えると、そのデータは無効になります。

[bit9] RIE:Receiver Interrupt Enable

このビットは、受信割込みを制御します。

0	割込みを禁止します。
1	割込みを許可します。

受信割込み要因は、PE, ORE, FRE によるエラーのほか、RDRF による正常受信があります。

[bit8] TIE:Transmitter Interrupt Enable

このビットは、送信割込みを制御します。

0	割込みを禁止します。
1	割込みを許可します。

<注意事項>

送信中に送信動作を禁止したときには、シリアルアウトプットレジスタ (SODR) にデータがなくなった後に送信動作を停止します。"0" の書込みは、SODR レジスタにデータを書き込んだ後は、一定の期間を置いて行ってください。一定の期間とは、クロック非同期転送モード時は、ボーレートの 1/16 の時間です。クロック同期転送モード時は、ボーレートの時間です。

19.3.5 通信プリスケアラコントロールレジスタ (CDCR)

通信プリスケアラコントロールレジスタ (CDCR) の構成および機能について説明します。

■ 通信プリスケアラコントロールレジスタ (CDCR)

下図に、通信プリスケアラコントロールレジスタ (CDCR) のビット構成を示します。

000025 _H	15	14	13	12	11	10	9	8	通信プリスケアラコントロールレジスタ (CDCR) リード/ライト 初期値
	MD	SRST	—	—	DIV3	DIV2	DIV1	DIV0	
	(R/W)	(R/W)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	
	(0)	(0)	(—)	(—)	(0)	(0)	(0)	(0)	

CDCR レジスタは、マシンのクロックの分周を制御します。UART の動作クロックは、マシンのクロックを分周することによって得られます。この通信プリスケアラによって、種々のマシンのクロックに対して一定のボーレートを得ることが可能です。

[bit15] MD:Machine clock divide moDe select

このビットは、通信プリスケアラの動作許可ビットです。

0	通信プリスケアラは停止します。
1	通信プリスケアラは動作します。

[bit14] SRST:Set ReSeT

このビットは、UART のすべての動作をリセットします。また、すべてのデータ / レジスタを初期化します。

0	初期値 (なにもしません。)
1	強制リセット

<注意事項>

このビットをセットすると、UART のすべてのデータ / レジスタ値が強制クリアされます。初期値状態に戻るため、すべてのデータ / レジスタを再設定してください。転送中のデータも、保存されたデータも、再度設定されるまで無効となります。

[bit11, bit10, bit9, bit8] DIV3, DIV2, DIV1, DIV0

これらのビットは、マシクロックの分周比を決定します。

DIV3 ~ DIV0	分周比
0000 _B	1 分周
0001 _B	2 分周
0010 _B	3 分周
0011 _B	4 分周
0100 _B	5 分周
0101 _B	6 分周
0110 _B	7 分周
0111 _B	8 分周

＜注意事項＞

- 分周比を変えた場合には、クロックの安定時間として2分周分の時間をおいてから通信してください。
- 同期転送時に専用ボーレートジェネレータを使用する場合，以下の設定をしないでください。
 - 1) CS2 ~ CS0 = 000_B
 - 2) CS2 ~ CS0 = 001_B, DIV3 ~ DIV0 = 0000_B

19.4 UART の割込み

UART 割込みには、受信割込みと送信割込みがあります。

UART の割込みによって、DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができます。

■ UART の割込み

UART の割込み制御ビットと割込み要因を下記の表に示します。

	UART 受信割込み	UART 送信割込み
割込み要求フラグ	データ受信完了 SSR:RDRF (bit12) フレーミングエラー SSR:FRE (bit13) オーバランエラー SSR:ORE (bit14) パリティエラー SSR:PE (bit15)	SSR:TDRE (bit11)
割込み要求出力許可ビット	SSR:RIE (bit9)	SSR:TIE (bit8)
割込み発生要因	UART 受信時	UART 送信時

■ UART に関する割込み要因

割込みは UART の受信時と送信時に発生します。

以下に示す要因で割込み要求が発生します。

- ・受信データがシリアルインプットレジスタ (SIDR) 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アウトプットレジスタ (SODR)から送信用シフトレジスタへ転送された場合

■ UART の割込みと DMA 転送および EI²OS

表 19.4-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 19.4-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
UART 送信完了 *	○	11	#34	FFFF74 _H	ICR11	0000BB _H
UART 受信完了 *	◎	7	#36	FFFF6C _H	ICR12	0000BC _H

○：割込み要求フラグはクリアされます。

◎：割込み要求フラグはクリアされます (ストップ要求あり)。

*：この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

UART は DMA 転送機能、および EI²OS 機能に対応しています。DMA または EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

19.5 UART の動作

UART の動作について説明します。

■ 動作モード

UART は、下表に示す動作モードを持ち、SMR レジスタおよび SCR レジスタに値を設定することによってモードを切り換えることができます。

モード	パリティ	データ長	動作モード	ストップビット長
0	あり / なし	7	非同期 (調歩同期) ノーマルモード	1 ビットまたは 2 ビット ^{*2}
	あり / なし	8		
1	なし	8+1 ^{*1}	非同期 (調歩同期) マルチプロセッサモード	
2	なし	8	CLK 同期モード	なし

^{*1}: +1 は、通信制御用に使用されるアドレス / データ選択ビット (A/D) です。

^{*2}: 受信時のストップビットは、1 ビットのみ検出可

非同期 (調歩同期) モードでのストップビット長については、送信動作のみ指定が可能です。受信動作については、常に 1 ビット長となります。非同期 (調歩同期) モード以外ではストップビット長は、設定しないでください。

UART 動作モード 1 は、マスタ / スレーブ型接続時のマスタ時だけに使用されます。

● CPU 間接続方式

1 対 1 接続 (ノーマルモード) とマスタ / スレーブ型接続 (マルチプロセッサモード) のどちらかを選択できます。どちらの方式でもデータ長、パリティの有無、同期方式などは、すべての CPU で統一しておく必要があり、動作モードを次のように選択します。

- 1 対 1 接続 (ノーマルモード) では、2 つの CPU で動作モード 0, 2 のいずれかの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 0 を、同期式では動作モード 2 を選択してください。
- マスタ / スレーブ型接続 (マルチプロセッサモード) では、動作モード 1 を使用します。動作モード 1 を選択し、マスタとして使用してください。なお、この接続ではパリティなしを選択してください。

■ UART のクロック選択

● 専用ボーレートジェネレータ

- 非同期ボーレート = $\phi / (\text{プリスケアラ分周比}) / (\text{非同期転送クロック分周比})$
 - 同期ボーレート = $\phi / (\text{プリスケアラ分周比}) / (\text{同期転送クロック分周比})$
- ϕ : マシンクロック

- プリスケアラによる分周比 (非同期 / 同期共通) : 表 19.5-1 に示します。

表 19.5-1 プリスケアラによる分周比

MD	DIV3 ~ DIV0	DIV
0	ϕ	停止
1	0000 _B	1
1	0001 _B	2
1	0010 _B	3
1	0011 _B	4
1	0100 _B	5
1	0101 _B	6
1	0110 _B	7
1	0111 _B	8

- 同期転送クロック分周比 : 表 19.5-2 に示します。

表 19.5-2 同期転送クロック分周比

CS2	CS1	CS0	CLK 同期	算出式	SCK0
0	0	1	4M	$(\phi / \text{DIV}) / 2$	$(\phi / \text{DIV}) / 2$
0	1	0	2M	$(\phi / \text{DIV}) / 4$	$(\phi / \text{DIV}) / 4$
0	1	1	1M	$(\phi / \text{DIV}) / 8$	$(\phi / \text{DIV}) / 8$
1	0	0	500K	$(\phi / \text{DIV}) / 16$	$(\phi / \text{DIV}) / 16$
1	0	1	250K	$(\phi / \text{DIV}) / 32$	$(\phi / \text{DIV}) / 32$

ϕ : マシンクロック (内部周波数 $f=16\text{MHz}$), DIV=2 で算出

<注意事項>

同期転送時に以下の設定をしないでください。

- 1) CS2 ~ CS0 = 000_B
- 2) CS2 ~ CS0 = 001_B, DIV3 ~ DIV0 = 0000_B

- 非同期転送クロック分周比 : 表 19.5-3 に示します。

表 19.5-3 非同期転送クロック分周比

CS2	CS1	CS0	非 CLK 同期	算出式	SCK0
0	0	0	76923	$(\phi / \text{DIV}) / (8 \times 13 \times 2)$	$(\phi / \text{DIV}) / (13 \times 2)$
0	0	1	38461	$(\phi / \text{DIV}) / (8 \times 13 \times 4)$	$(\phi / \text{DIV}) / (13 \times 4)$
0	1	0	19230	$(\phi / \text{DIV}) / (8 \times 13 \times 8)$	$(\phi / \text{DIV}) / (13 \times 8)$
0	1	1	9615	$(\phi / \text{DIV}) / (8 \times 13 \times 16)$	$(\phi / \text{DIV}) / (13 \times 16)$
1	0	0	500K	$(\phi / \text{DIV}) / (8 \times 2 \times 2)$	$(\phi / \text{DIV}) / 2$
1	0	1	250K	$(\phi / \text{DIV}) / (8 \times 2 \times 4)$	$(\phi / \text{DIV}) / 4$

ϕ : マシンクロック (内部周波数 $f=16\text{MHz}$), DIV=1 で算出

● 内部タイマ

CS2 ～ 0 を "110" に設定して , 内部タイマ (PPG1) を選択した場合のボーレートは , 次式によって計算できます。

非同期 (調歩同期) : $(\phi/N)/(16 \times 2 \times (n+1))$

CLK 同期 : $(\phi/N)/(2 \times (n+1))$

N: タイマ (PPG1) のカウントクロックソース

n: タイマ (PPG1) のリロード値

マシンのクロックが 7.3728MHz の場合のボーレートとリロード値の関係を表 19.5-4 に示します。

表 19.5-4 ボーレートとリロード値の関係 (マシンのクロック =7.3728MHz)

ボーレート	リロード値			
	クロック非同期 (調歩同期)		クロック同期	
	N=2 ¹ (マシンのクロック の 2 分周)	N=2 ³ (マシンのクロックの 8 分周)	N=2 ¹ (マシンのクロックの 2 分周)	N=2 ³ (マシンのクロックの 8 分周)
38400	2	-	47	11
19200	5	-	95	23
9600	11	2	191	47
4800	23	5	383	95
2400	47	11	767	191
1200	95	23	1535	383
600	191	47	3071	767
300	383	95	6143	1535

"-" は設定禁止です。

<注意事項>

クロック同期時 , 以下の設定をしないでください。

N = 1, n = 0

● 外部クロック

CS2 ～ CS0 を "111" に設定した場合のボーレートは , 次式によって計算できます。

非同期 (調歩同期) : $f/16$

CLK 同期 : f

ただし , f の最大はマシンのクロックの 1/2 まで , f' の最大はマシンのクロックの 1/8 までです。

19.5.1 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 (ノーマルモード) または動作モード 1 (マルチプロセッサモード) で使用する場合は、転送方式は非同期となります。

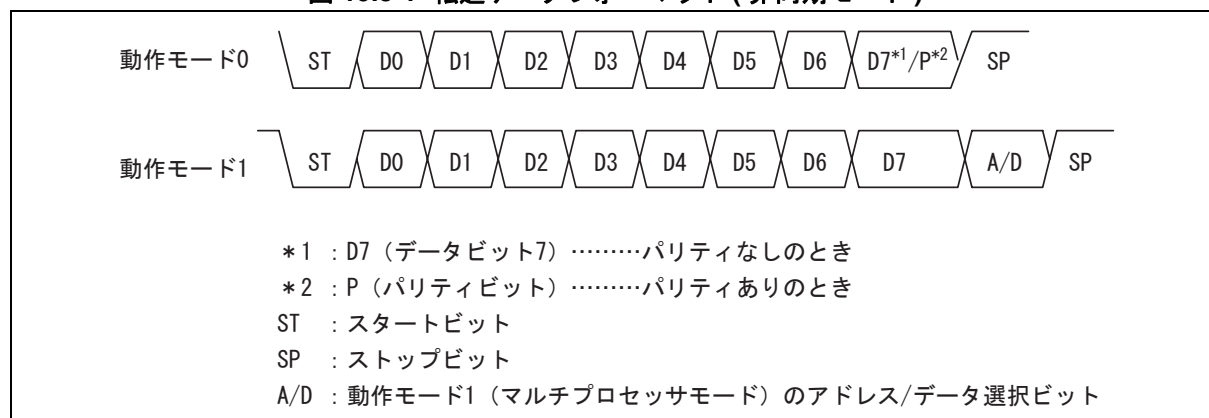
■ 非同期モード (動作モード 0, 1) 時の動作

● 転送データフォーマット

転送データは、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転送が 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 のノーマルモードでは、データ長は 7 ビットまたは 8 ビットに設定が可能です。
- 動作モード 1 では、データは 8 ビット長固定でパリティは付加されず、そのかわり A/D (アドレス/データ選択ビット) が付加されています。

図 19.5-1 転送データフォーマット (非同期モード)



● 送信動作

送信データエンプティフラグビット (SSR:TDRE) が "1" のとき、アウトプットデータレジスタ (SODR) に送信データを書き込みます。ここで送信動作が許可 (SCR:TXE=1) されていると、送信が行われます。送信データが送信用シフトレジスタに転送され、送信が開始されると、TDRE フラグが再び "1" にセットされて、次の送信データをセットできるようになります。

このとき、送信割込み要求が許可 (SSR:TIE=1) されていれば、送信割込み要求を出力して SODR に送信データをセットするように要求します。TDRE フラグは、SODR に送信データが書き込まれると "0" にクリアされます。

● 受信動作

受信動作が許可 (SCR:RXE=1) されていると、常に受信動作が行われます。スタートビットを検出すると、シリアルコントロールレジスタ (SCR) で決められたデータフォーマットに従って、1 フレームデータの受信が行われます。1 フレームの受信が終わると、エラーが発生したときにはエラーフラグがセットされた後、受信データフルフラグビット (SSR:RDRF) が "1" にセットされます。このとき、受信割込み要求が許可 (SSR:RIE=1) されていれば、受信割込み要求を出力します。シリアルステータスレジスタ (SSR) の各フラグを調べ、正常受信であればシリアルインプットレジスタ (SIDR) を読み出して、エラーが発生していた場合はエラー処理をするようにしてください。

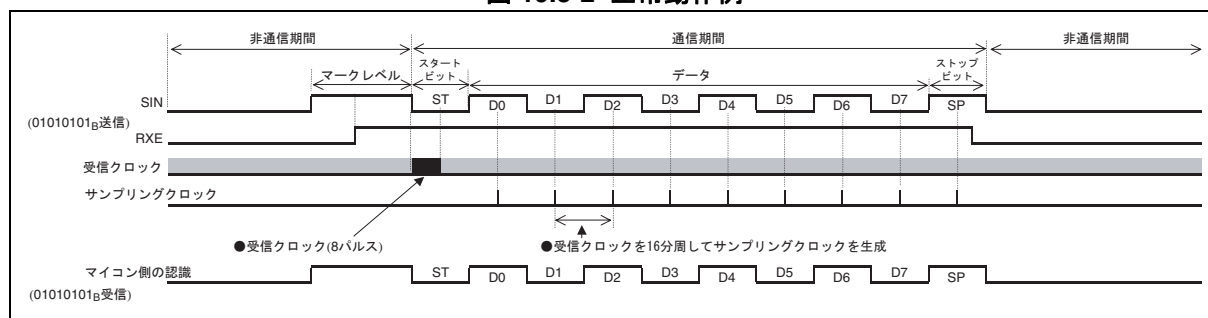
RDRF フラグは、SIDR から受信データを読み出すと "0" にクリアされます。

● 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。
- 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。

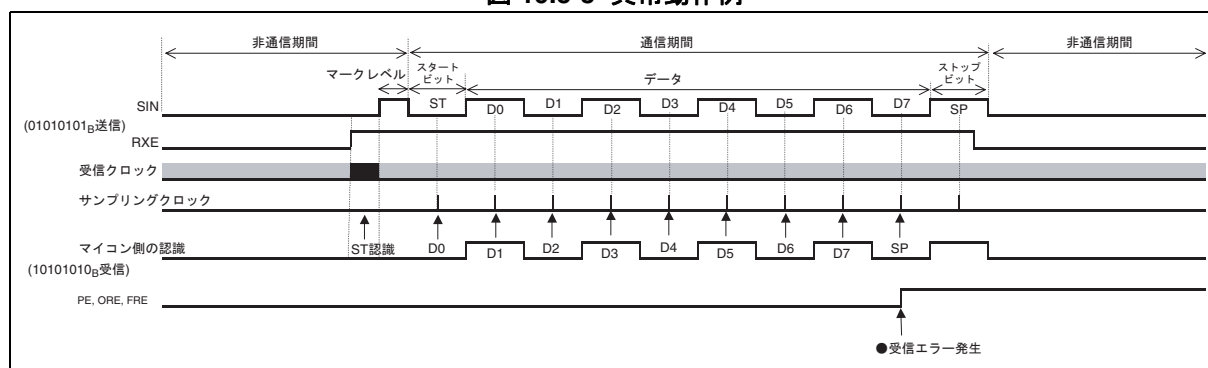
図 19.5-2 正常動作例



以下のようなタイミングで受信許可に設定した場合は、マイコン側で入力データ (SIN) が正しく確認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") にした場合の動作例

図 19.5-3 異常動作例



● ストップビット

送信時には1ビットまたは2ビットを選択できます。ただし、受信側では、常に最初の1ビットだけを判別します。

● エラー検出

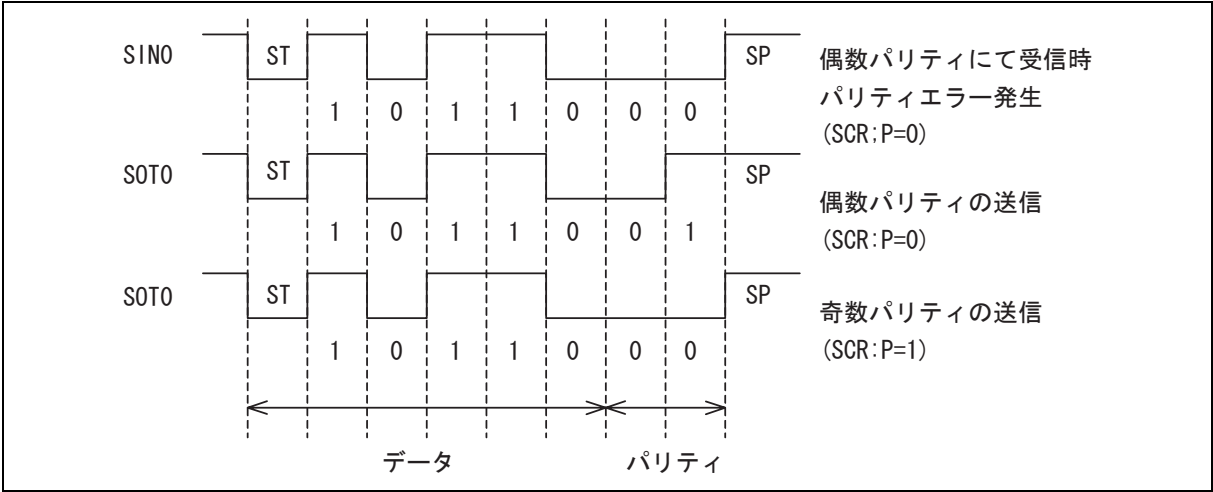
- モード 0: パリティエラー、オーバランエラー、フレーミングエラーが検出できます。
- モード 1: オーバランエラー、フレーミングエラーが検出でき、パリティエラーは検出できません。

● パリティ

パリティは、動作モード 0(非同期, ノーマルモード) のときだけに使用できます。シリアルコントロールレジスタ (SCR) の PEN ビットでパリティの有無を、P ビットで偶数/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) および動作モード 2 (CLK 同期モード) では、パリティの使用はできません。パリティ有効時の送受信データのデータフォーマットを、図 19.5-4 に示します。

図中の ST はスタートビット, SP はストップビットです。

図 19.5-4 パリティ使用時の転送データフォーマット



<注意事項>

動作モード 1, 2 では、パリティを使用できません。

19.5.2 同期モード (動作モード 2) 時の動作

UART 動作モード 2 (CLK 同期モード) で使用するときには、転送方式はクロック同期となります。

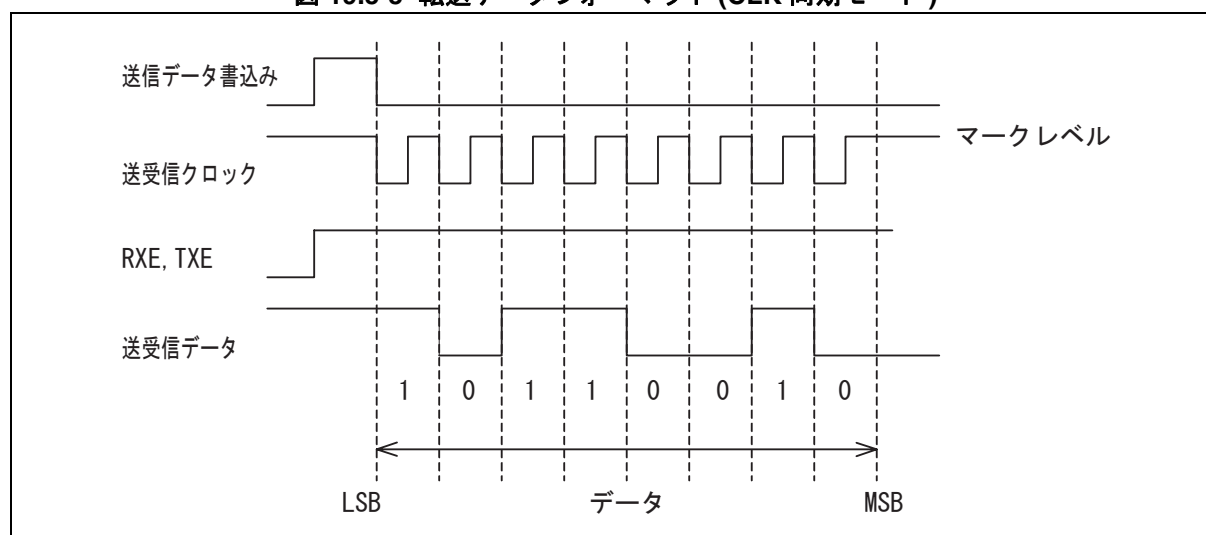
■ CLK 同期モード (動作モード 2) 時の動作

● 転送データフォーマット

同期モードでは、8 ビットデータを LSB ファーストで転送します。スタートビット / ストップビットは付加されません。

図 19.5-5 に、同期モード時のデータフォーマットを示します。

図 19.5-5 転送データフォーマット (CLK 同期モード)



● クロック供給

クロック同期 (I/O 拡張シリアル) 方式では、送受信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 (専用ボーレートジェネレータ、または内部タイマ) を選択しているときは、データを送信するとデータ受信用同期クロックが自動生成されます。

外部クロックを選択しているときは、送信側 UART のシリアルアウトプットレジスタ (SODR) にデータがあることが必要です。SSR の TDRE=0 であることを確かめた後に、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、送信開始前と終了後は、必ずマークレベル ("H") にする必要があります。

● エラー検出

オーバランエラーだけ検出可能で、パリティエラー、フレーミングエラーは検出できません。

● 初期化

以下に、同期モードを使用するときの各制御レジスタの設定値を示します。

[シリアルモードレジスタ (SMR)]

- MD1, MD0: "10"
- CS2, CS1, CS0: クロックセレクタのクロックソースを指定
- SCKE: 専用ボーレートジェネレータ、または内部クロック時は "1", 外部クロック時は "0"
- SOE: 送信するとき "1", 受信だけのとき "0"

[シリアルコントロールレジスタ (SCR)]

- PEN: "0"
- P, SBL, A/D: これらのビットは意味を持ちません。
- CL: "1"(8 ビットデータ)
- REC: "0"(初期化するために、エラーフラグをクリア)
- RXE, TXE: 少なくともどちらか一方が "1"

[シリアルステータスレジスタ (SSR)]

- RIE: 割込みを使用するときは "1", 割込みを使用しないときは "0"
- TIE: "0"

● 通信開始

シリアルアウトプットレジスタ (SODR) への書込みで通信を開始します。受信する場合でも通信を開始するときは、必ず仮のデータを SODR に書き込む必要がありますので、注意してください。

● 通信終了

1 フレームのデータ送受信が終了すると、シリアルステータスレジスタ (SSR) の RDRF フラグが "1" にセットされます。受信時は、オーバランエラーフラグビット (SSR:ORE) をチェックし、通信が正常に行われたかどうかの判断をしてください。

19.5.3 双方向通信機能 (ノーマルモード)

動作モード 0, 2 で , 1 対 1 接続の通常シリアル双方向通信をすることができます。
同期方式は , 動作モード 0 の場合は非同期 , 動作モード 2 の場合は同期となります。

■ 双方向通信時における各レジスタの設定

双方向通信を行う場合は , 各レジスタを図 19.5-6 に示すように設定する必要があります。

図 19.5-6 双方向通信時における各レジスタの設定

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR/SMR	PEN	P	SBL	CL	A/D	REC	RXE	TXE	MD1	MDO	CS2	CS1	CS0	BCH	SCKE	SOE
モード0	◎	◎	◎	◎	×	0	◎	◎	0	0	◎	◎	◎	◎	◎	◎
モード2	0	×	×	1	×	0	◎	◎	1	0	◎	◎	◎	◎	◎	◎

SSR, SIDR/SODR	PE	ORE	FRE	TDRF	—	R1E	T1E	変換データを設定 (書き込み時) / 受信データを保持 (読出し時)								
モード0	◎	◎	◎	◎	—	◎	◎									
モード2	×	◎	×	◎	—	◎	◎									

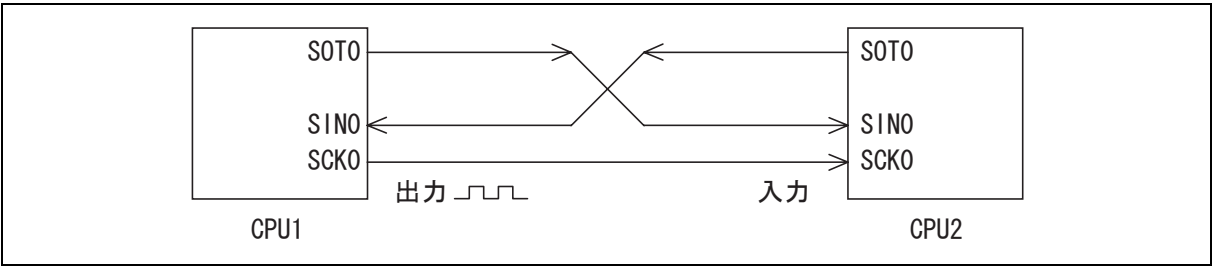
DDR7*	P77	P76	P75	P74	P73	P72	P71	P70
						△		△

* : UART0を使用する場合は、DDR7：ビット0，2を設定してください。
 ◎ : 使用ビット
 × : 未使用ビット
 1 : “1”を設定
 0 : “0”を設定
 △ : 端子入力を使用する場合“0”を設定

■ 双方向通信時の CPU 間接続

図 19.5-7 に , 双方向通信時の CPU 間接続を示します。

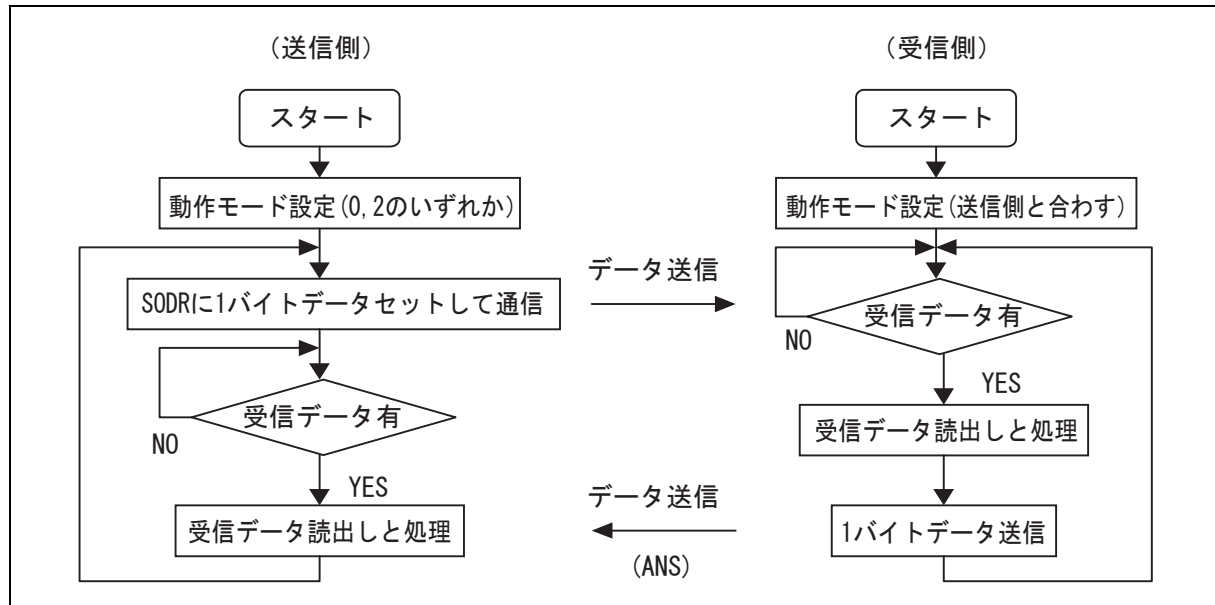
図 19.5-7 双方向通信時の CPU 間接続



■ 双方向通信機能の通信手順

通信は任意のタイミングで、送信データが準備できたときに送信側から開始します。
受信側で送信データを受け取ると、定期的にANS(本例では、1バイトごと)を返します。
図 19.5-8 に、双方向通信機能の通信手順を示します。

図 19.5-8 双方向通信機能の通信手順



19.5.4 マスタ / スレーブ型通信機能 (マルチプロセッサモード)

UART は , 複数 CPU のマスタ / スレーブ型接続による通信が可能であり , 動作モード 1 を使用します。ただし , UART は , マスタとしてだけ使用可能です。

■ マスタ / スレーブ通信時の各レジスタの設定

マスタ / スレーブ通信を行う場合は , 各レジスタを図 19.5-9 に示すように設定する必要があります。

図 19.5-9 マスタ / スレーブ通信時の各レジスタの設定

ビット	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR/SMR	PEN	P	SBL	CL	A/D	REC	RXE	TXE	MD1	MD0	CS2	CS1	CS0	BCH	SCKE	SOE
モード1	0	×	◎	1	◎	0	◎	◎	0	1	◎	◎	◎	◎	0	◎

SSR,
SIDR/SODR

PE	ORE	FRE	TDRF	—	RIE	TIE	変換データを設定（書き込み時）／受信データを保持（読出し時）										
モード1	×	◎	◎	◎	—	◎	◎										

DDR7*

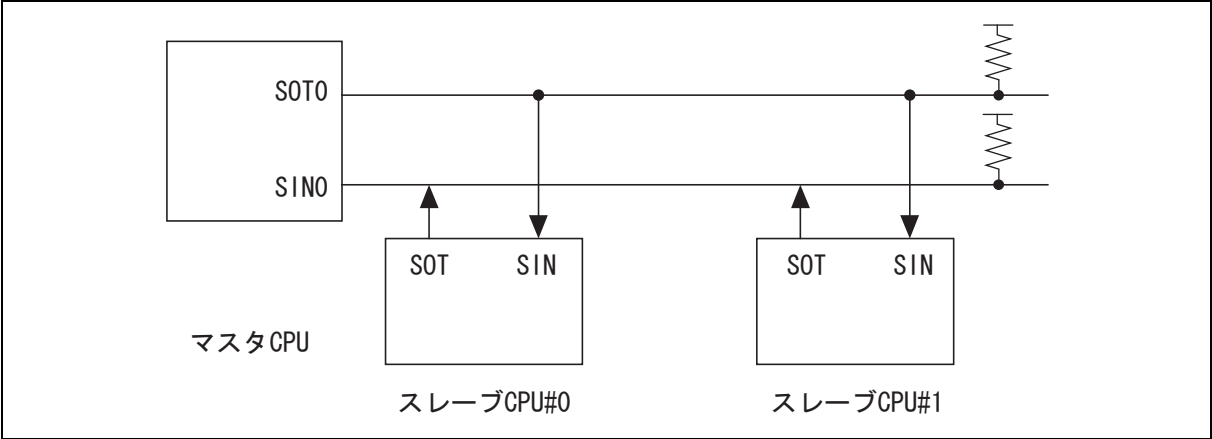
P77	P76	P75	P74	P73	P72	P71	P70
					△		△

* : UART0を使用する場合は、DDR7：ビット0，2を設定してください。
◎ : 使用ビット
× : 未使用ビット
1 : “1”を設定
0 : “0”を設定
△ : 端子入力を使用する場合“0”を設定

■ マスタ / スレーブ通信時の CPU 間接続

図 19.5-10 に , マスタ / スレーブ通信時の CPU 間接続を示します。

図 19.5-10 マスタ / スレーブ通信時の CPU 間接続



■ 機能選択

表 19.5-5 に、マスタ / スレーブ通信時の通信方式の設定内容を示します。

表 19.5-5 マスタ / スレーブ通信時の機能選択

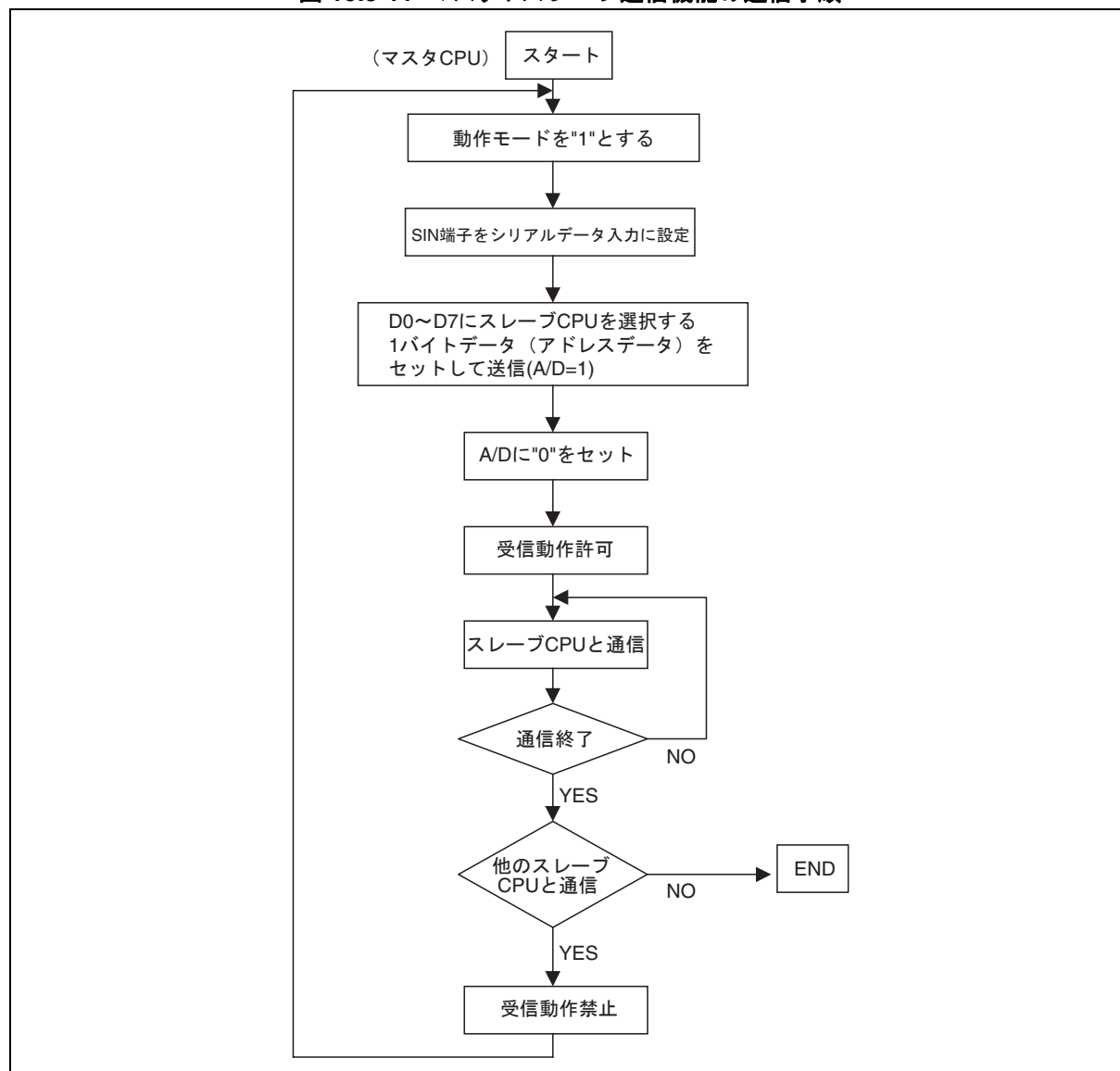
	動作モード		データ	パリティ	同期方式	ストップビット
	マスタ CPU	スレーブ CPU				
アドレス送受信	モード 1	-	A/D=1+8 ビットアドレス	なし	非同期	1 ビットまたは 2 ビット
データ送受信			A/D=0+8 ビットアドレス			

■ マスタ / スレーブ通信機能の通信手順

通信は、マスタ CPU 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 ビットを "1" としたデータで、通信先となるスレーブ CPU を選択します。各スレーブ CPU はプログラムでアドレスデータを判断し、割り当てられたアドレスと一致するとき、マスタ CPU との通信（通常データ）を行います。

図 19.5-11 に、マスタ / スレーブ通信機能の通信手順を示します。

図 19.5-11 マスタ / スレーブ通信機能の通信手順



19.6 UART の使用上の注意

UART を使用する場合は、次の点に注意してください。

■ UART の使用上の注意

● 動作許可

送信、受信のそれぞれに対して、シリアルコントロールレジスタ (SCR) 内に TXE(送信), RXE(受信) の動作許可ビットがあります。

デフォルト (初期値) では、送受信ともに動作禁止です。必要に応じて動作を許可してください。

● 通信モードの設定

通信モードは、停止中に設定してください。送受信中に通信モード設定をすると、その送受信データは保証されません。

● 同期モード

UART のクロック同期モード (動作モード 2) は、クロック制御 (I/O 拡張シリアル) 方式を採用しています。スタートビットおよびストップビットはデータに付加されません。

● 送信データエンプティフラグビット

送信データエンプティフラグビット (SSR の TDRE) は、デフォルト (初期値) が "1" (送信データなし、送信データ書込み許可) であるため、送信割込み要求を許可 (SSR の TIE=1) すると、直ち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、TIE フラグを "1" にしてください。

● 同期転送時のクロック設定

- 同期転送時に専用ボーレートジェネレータを使用する場合、以下の設定をしないでください。

1) CS2 ~ CS0 = 000_B

2) CS2 ~ CS0 = 001_B, DIV3 ~ DIV0 = 0000_B

- 同期転送時に内部タイマ (PPG1) を使用する場合、以下の設定をしないでください。

N = 1, n = 0

19.7 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

<p>設定手順例</p> <p>1byte のデータを UART で送信する。 非同期ノーマルモード</p> <p><初期設定></p> <ul style="list-style-type: none"> ・割込み関連 <table border="1"> <tr> <td>割込みレベルの設定</td><td>ICR11</td></tr> <tr> <td>I フラグの設定</td><td>(CCR)</td></tr> </table> <ul style="list-style-type: none"> ・UART の制御 <table border="1"> <tr> <td>レジスタ名、ビット名</td><td></td></tr> <tr> <td>モードレジスタの設定</td><td>SMR</td></tr> <tr> <td>動作モードの設定</td><td>.MD1, MD0</td></tr> <tr> <td>動作クロックの設定</td><td>.CS2-0</td></tr> <tr> <td>SCK0 端子の設定</td><td>.SCKE</td></tr> <tr> <td>SOT0 端子の設定</td><td>.SOE</td></tr> <tr> <td>制御レジスタの設定</td><td>SCR</td></tr> <tr> <td>パリティ有無の設定</td><td>.PEN</td></tr> <tr> <td>偶数/奇数パリティの設定</td><td>.P</td></tr> <tr> <td>ストップビット長の設定</td><td>.SBL</td></tr> <tr> <td>1 フレームデータ長の設定</td><td>.CL</td></tr> <tr> <td>フレームデータ形式の設定</td><td>.A/D</td></tr> <tr> <td>エラーフラグ</td><td>.REC</td></tr> <tr> <td>受信動作許可設定</td><td>.RXE</td></tr> <tr> <td>送信動作許可設定</td><td>.TXE</td></tr> <tr> <td>割込み制御</td><td>SSR .TIE</td></tr> <tr> <td>送信データの設定</td><td>SIDR</td></tr> </table> <p><起動></p> <ul style="list-style-type: none"> ・UART 起動 <table border="1"> <tr> <td>レジスタ名、ビット名</td><td></td></tr> <tr> <td>UART0 送信動作起動</td><td>SCR0 .TXE</td></tr> </table> <p><割込み></p> <ul style="list-style-type: none"> ・送信割込み処理 <table border="1"> <tr> <td>レジスタ名、ビット名</td><td></td></tr> <tr> <td>任意のデータを送信</td><td>SIDR</td></tr> </table> <p><割込みベクタ></p> <ul style="list-style-type: none"> ・ベクタテーブルの設定 <p>(注意事項) 事前にクロック関連の設定および、__set_irq(数値) の設定が必要です。クロックおよび割込みの章をご参照ください。</p>	割込みレベルの設定	ICR11	I フラグの設定	(CCR)	レジスタ名、ビット名		モードレジスタの設定	SMR	動作モードの設定	.MD1, MD0	動作クロックの設定	.CS2-0	SCK0 端子の設定	.SCKE	SOT0 端子の設定	.SOE	制御レジスタの設定	SCR	パリティ有無の設定	.PEN	偶数/奇数パリティの設定	.P	ストップビット長の設定	.SBL	1 フレームデータ長の設定	.CL	フレームデータ形式の設定	.A/D	エラーフラグ	.REC	受信動作許可設定	.RXE	送信動作許可設定	.TXE	割込み制御	SSR .TIE	送信データの設定	SIDR	レジスタ名、ビット名		UART0 送信動作起動	SCR0 .TXE	レジスタ名、ビット名		任意のデータを送信	SIDR	<p>プログラム例</p> <pre> void Asynch_uart_sample(void) { INT_initial(); Asynch_uart_initial(); Uart_start(); } void INT_initial(void) { IO_ICR11.byte = 0x10; /* UART 送信完了割込みレベル設定 (値は任意) */ __EI(); /* 割込み許可 */ } void Asynch_uart_initial(void) { IO_SMR.byte = 0x19; /* 設定値 = 0001_1001 */ ; /* bit7-6 = 00 MD1, MD0 非同期ノーマルモード */ /* bit5-3 = 011 CS2-0 専用ボーレートジェネレータ */ /* bit2 = 0 未定義ビット */ /* bit1 = 0 SCKE クロック入力端子 */ /* bit0 = 1 SOE 外部端子 */ IO_SCR.byte = 0x10; /* 設定値 = 0001_0000 */ /* bit15 = 0 PEN パリティなし */ /* bit14 = 0 P 偶数パリティ */ /* bit13 = 0 SBL 1 ストップビット */ /* bit12 = 1 CL 8 ビットデータ */ /* bit11 = 0 A/D データフレーム */ /* bit10 = 0 REC エラーフラグクリア */ /* bit9 = 0 RXE 受信動作禁止 */ /* bit8 = 0 TXE 送信動作禁止 */ /* bit8 = 1 TIE 送信割込み許可 */ IO_SSR.bit.TIE = 1; IO_SIDR = 0x0aa; /* 任意のデータ値を送信する。 */ } void Uart_start(void) { IO_SCR.bit.TXE = 1; /* bit8 = 1 TXE 送信動作許可 */ } __interrupt void uart_tx_int(void) { IO_SIDR = 0x0aa; /* 任意のデータ値を送信する。 */ } #pragma intvect uart_tx_int 34 </pre> <p>(注意事項) レジスタの記述形式については、「F²MC-16LX ファミリー MB90480/485 シリーズ用 サンプル I/O レジスタファイル 使用手引書」をご参照ください。</p>
割込みレベルの設定	ICR11																																														
I フラグの設定	(CCR)																																														
レジスタ名、ビット名																																															
モードレジスタの設定	SMR																																														
動作モードの設定	.MD1, MD0																																														
動作クロックの設定	.CS2-0																																														
SCK0 端子の設定	.SCKE																																														
SOT0 端子の設定	.SOE																																														
制御レジスタの設定	SCR																																														
パリティ有無の設定	.PEN																																														
偶数/奇数パリティの設定	.P																																														
ストップビット長の設定	.SBL																																														
1 フレームデータ長の設定	.CL																																														
フレームデータ形式の設定	.A/D																																														
エラーフラグ	.REC																																														
受信動作許可設定	.RXE																																														
送信動作許可設定	.TXE																																														
割込み制御	SSR .TIE																																														
送信データの設定	SIDR																																														
レジスタ名、ビット名																																															
UART0 送信動作起動	SCR0 .TXE																																														
レジスタ名、ビット名																																															
任意のデータを送信	SIDR																																														

■ プログラム例以外の設定方法

● 設定できる組合せ

組合せは以下のとおりです。

動作モード (MD[1:0])		データ長 (CL)	パリティ (PEN)	パリティ 選択 (P)	データ形式 (A/D)	STOP ビット長 選択 (SBL) *		エラーフラグの有無		
								オーバラン (ORE)	フレーミング (FRE)	パリティ (PE)
0	非同期 - ノーマルモード (00)	7 ビット (0)	なし (0)	—	—	1 ビット (0)	2 ビット (1)	○	○	○
			あり (1)	偶 (0)		1 ビット (0)	2 ビット (1)			
				奇 (1)		1 ビット (0)	2 ビット (1)			
		8 ビット (1)	なし (0)	—		1 ビット (0)	2 ビット (1)			
			あり (1)	偶 (0)		1 ビット (0)	2 ビット (1)			
				奇 (1)		1 ビット (0)	2 ビット (1)			
1	非同期 - マルチプロセッサ モード (01)	8 ビット (1)	なし (0)	—	アドレス (1)	1 ビット (0)	2 ビット (1)	○	○	—
					アドレス (0)	1 ビット (0)	2 ビット (1)			
2	CLK 同期モード (10)	8 ビット (1)	なし (0)	—	—	—		○	—	—

※: STOP ビットの選択は送信時のみ有効です。受信時は 1 ビット目のみの検出 (2 ビット目は無視) となります。

● 各動作モードを選択する方法

動作モードビット (SMR.MOD[1:0]) で設定します。

動作モード		動作モードビット (MOD[1:0])
モード 0	非同期 - ノーマルモード	"00 _B " にする
モード 1	非同期 - マルチプロセッサモード	"01 _B " にする
モード 2	CLK 同期モード	"10 _B " にする
—		"11 _B " は設定禁止

● 動作クロックの種類と選択方法

動作クロック選択ビット (SMR.CS[2:0]) で設定します。

クロック入力	動作クロック選択ビット (CS[2:0])
専用ポーレートジェネレータ	"000 _B " ~ "101 _B " にする
内蔵クロックを選択するには	"110 _B " にする
外部クロックを選択するには	"111 _B " にする

● SCK 端子, SIN 端子, SOT 端子を制御する方法

下記のように設定します。

	UART レジスタ
SCK 端子を入力にするには	DDR7.P72 = 0 SMR.SCKE = 0
SCK 端子を出力にするには	SMR.SCKE = 1
SIN 端子を入力にするには	DDR7.P70 = 0
SOT 端子を出力にするには	SMR.SOE = 1

● UART の動作を許可 / 停止する方法

受信動作制御ビット (SCR.RXE) で設定します。

制御内容	受信動作制御ビット (RXE)
受信動作禁止 (停止)	"0" にする
受信動作許可	"1" にする

送信動作制御ビット (SCR.TXE) で設定します。

制御内容	送信動作制御ビット (TXE)
送信動作禁止 (停止)	"0" にする
送信動作許可	"1" にする

● パリティを設定する方法

パリティ設定ビット (SCR.PEN), パリティ選択ビット (SCR.P) で設定します。

動作	パリティ設定, 選択ビット (PEN, P)
パリティなしにするには	"00 _B " または "01 _B " にする
偶数パリティにするには	"10 _B " にする
奇数パリティにするには	"11 _B " にする

● データ長を設定する方法

データ長選択ビット (SCR.CL) で設定します。

動作	データ長選択ビット (CL)
7 ビット長にするには	"0" にする
8 ビット長にするには	"1" にする

● STOP ビット長を選択する方法

STOP ビット長選択ビット (SCR.SBL) で設定します。

動作	STOP ビット長選択ビット (SBL)
STOP ビットを 1 ビット長にするには	"0" にする
STOP ビットを 2 ビット長にするには	"1" にする

● エラーフラグをクリアする方法

エラーフラグクリアビット (SCR.REC) で設定します。

制御内容	エラーフラグクリアビット (REC)
エラーフラグ (PE, OFE, PRE) をクリアするには	"0" を書き込む

● 転送方向の設定方法

設定方向選択ビット (SSR.BDS) で設定します。

転送方向はどの動作モードでも, LSB/MSB の選択が可能です。

制御内容	設定方向選択ビット (BDS)
LSB 転送 (最下位ビットから) には	"0" にする
MSB 転送 (最上位ビットから) には	"1" にする

● 受信完了フラグをクリアする方法

下記の方法で行います。

制御内容	シリアルインプットレジスタ (SIDR)
受信完了フラグをクリアするには	SIDR レジスタを読み出す

初回の SIDR レジスタの読出しは, 受信開始になります。

● エラーフラグの種類と意味

エラーフラグは 3 種類あり以下の意味を持っています。

エラーフラグ	意味
パリティエラー (PE)	受信した値 (数値) に誤りがあります。
オーバランエラー (ORE)	受信データを読み出し終える前に, 次のデータが届きました。
フレーミングエラー (FRE)	受信したデータの形式に誤りがあります。

● 受信データの格納レジスタ

シリアルインプットレジスタ (SIDR) に受信データが格納されます。

● 送信データを書込みタイミングのステータス確認方法

送信バッファエンプティフラグ (SSR.TDRE) で確認できます。

● 送信データの書込みレジスタ

送信データはシリアルアウトプットレジスタ (SODR) に書き込みます。

● 送信バッファエンプティフラグをクリアする方法

下記の方法で行います。

制御内容	シリアルアウトプットレジスタ (SODR)
送信バッファエンプティフラグをクリアするには	SODR レジスタに書き込む

初回の SODR レジスタの書込みは、送信開始になります。

● データ形式 (アドレス / データ) を選択する方法

データ長選択ビット (SCR.A/D) で設定します。

動作	データ長選択ビット (A/D)
bit-8 をデータ ("L" レベル) にするには	"0" にする
bit-8 をアドレス ("H" レベル) にするには	"1" にする

送信時のみ有効です。受信時 A/D ビットを無視します。

● 受信 / 送信を開始させる方法

下記の方法で行います。

- モード 0/1, 送信 :
 1. 送信動作を許可にする
 2. シリアルアウトプットレジスタ (SODR) ヘデータを書き込む
(=送信開始)
- モード 0/1, 受信 :
 1. 受信を許可にする
 2. シリアルインプットレジスタ (SIDR) を読む
(ダミーリード=受信開始)
- モード 2, 送信 / 受信 :
 1. 送信を許可にする (受信許可省略可)
 2. シリアルアウトプットレジスタ (SODR) ヘデータを書き込む
(=送受信開始)
(受信動作だけでもシリアルアウトプットレジスタ (SODR) への書込みが、必要となります)

● 動作の停止と状態

- モード 0/1, 送信 : 送信動作を禁止にすると、送信バッファが空になったあとで、最後の送信データを送信し、STOP ビットを送信し終えたところで動作を停止します。
- モード 0/1, 受信 : 受信動作を禁止にすると、受信中のデータを受信完了後 (STOP ビット受信後)、受信データをシフトからレジスタに転送したあとで動作を停止します。
- モード 2, 送受信 : 受信動作と送信動作の両方を禁止にすると、送/受信しているデータの送/受信を完了し、最後に受信データをシフトからレジスタに転送した後で動作を停止します。

● 動作完了を確認する方法

下記の方法で行います。

- モード 0/1, 送信：次の送信データを書き込んだあとに, SSR レジスタエンプティフラグをチェック
(次の送信データをレジスタからシフトに転送し, シリアルアウトプットレジスタのエンプティフラグが "1" になることで確認可能。)
- モード 0/1, 受信：レジスタフルフラグをチェック
(シリアルインプットレジスタのフルフラグが "1" になることで確認可能。)
- モード 2, 送受信：レジスタフルフラグをチェック
(シリアルインプットレジスタのフルフラグが "1" になることで確認可能。)

● ボーレートを設定する方法

「19.5 UART の動作」をご参照ください。

● 割込み関連レジスタ

UART 割込みベクタと UART 割込みレベルの設定レジスタの関係は下表のとおりです。
割込みレベル, 割込みベクタの詳細については「第 3 章 割込み」をご参照ください。

	割込みベクタ	割込みレベル設定ビット
UART 受信	#36 アドレス : FFFF6C _H	割込み制御レジスタ 12 (ICR12) アドレス : 0000BC _H
UART 送信	#34 アドレス : FFFF74 _H	割込み制御レジスタ 11 (ICR11) アドレス : 0000BB _H

● 割込みの種類

割込み要因は, 受信側が 4 種類, 送信側が 1 種類です。

UART 受信	受信完了 (シリアルインプットレジスタフル), パリティエラー, オーバーランエラー, フレーミングエラーのうち最初に発生した要因によって, 割込み要求が発生します。
UART 送信	送信バッファエンプティにて割込み要求が発生します。

● 割込みを許可 / 禁止 / クリアする方法

割込み許可 / 禁止の設定は, 割込み要求許可ビット (SSR.RIE, SSR.TIE) で設定します。

	UART 受信	UART 送信
	割込み要求許可ビット (RIE)	割込み要求許可ビット (TIE)
割込み要求を禁止するには	"0" にする	
割込み要求を許可するには	"1" にする	

第 19 章 UART

割込み要求のクリアは、下記の方法で行います。

	UART 受信	UART 送信
割込み要求をクリアするには	受信完了フラグ (RDRF) はシリアル インプットレジスタ SISR を読み込 むことで "0" になります。	送信バッファエンプティフ ラグ (TDRE) はシリアルアウ トプットレジスタ (SODR) に データを書き込むことで "0" になります。
	エラーフラグ (PE, ORE, FRE) はエ ラーフラグクリアビット (REC) に "0" を書き込むことで "0" になります。	—

第20章

チップセレクト機能

チップセレクト機能の概要，構成，レジスタの構成と機能および動作について説明します。

20.1 チップセレクト機能の概要

20.2 チップセレクト機能の構成

20.3 チップセレクト機能のレジスタの構成と機能

20.4 チップセレクト機能の動作

20.1 チップセレクト機能の概要

チップセレクト機能は、外部へのメモリ接続を容易にするためのチップセレクト信号を発生するモジュールです。4 本のチップセレクト出力端子を持ち、ハードウェアで設定された領域を各出力の設定レジスタに設定し、外部アドレスに対するアクセスを検出すると、セレクト信号を端子から出力します。

■ チップセレクト機能の概要

チップセレクト機能には、各端子出力にそれぞれに対して設定用の 2 本の 8 ビットレジスタが用意されています。一方のレジスタ (CARx) で、比較するアドレスの上位 8 ビットを指定することにより、64K バイト単位の領域を検出します。また、もう一方のレジスタ (CMRx) で、比較するビットをマスクすることにより、検出する領域を 64K バイト以上に設定することができます。

なお、外部バスホールド時に、CS の出力は、ハイインピーダンス状態になります。

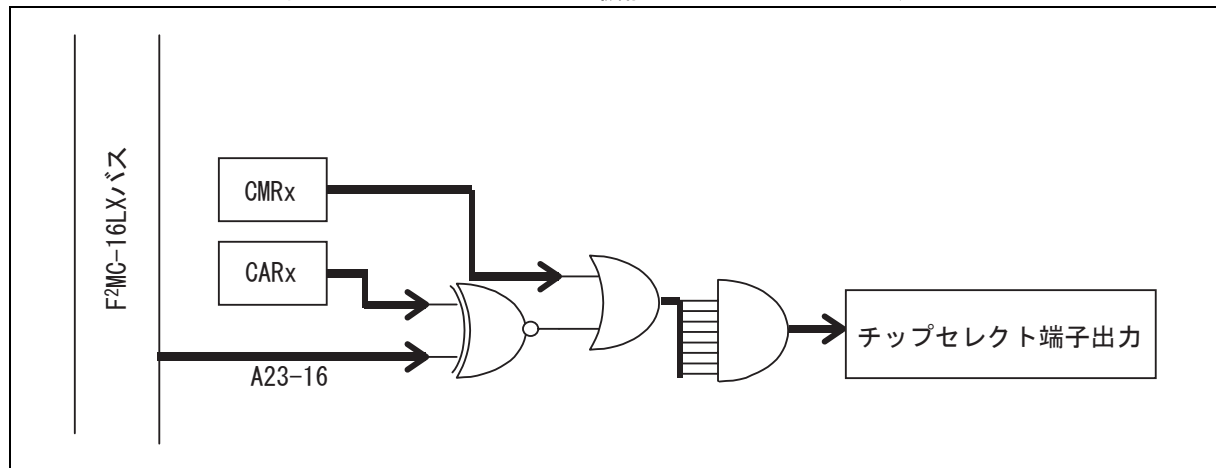
20.2 チップセレクト機能の構成

チップセレクト機能に関するブロックダイアグラムと関係する端子について示します。

■ チップセレクト機能のブロックダイアグラム

図 20.2-1 に、チップセレクト機能のブロックダイアグラムを示します。

図 20.2-1 チップセレクト機能のブロックダイアグラム



■ チップセレクト機能に関する端子

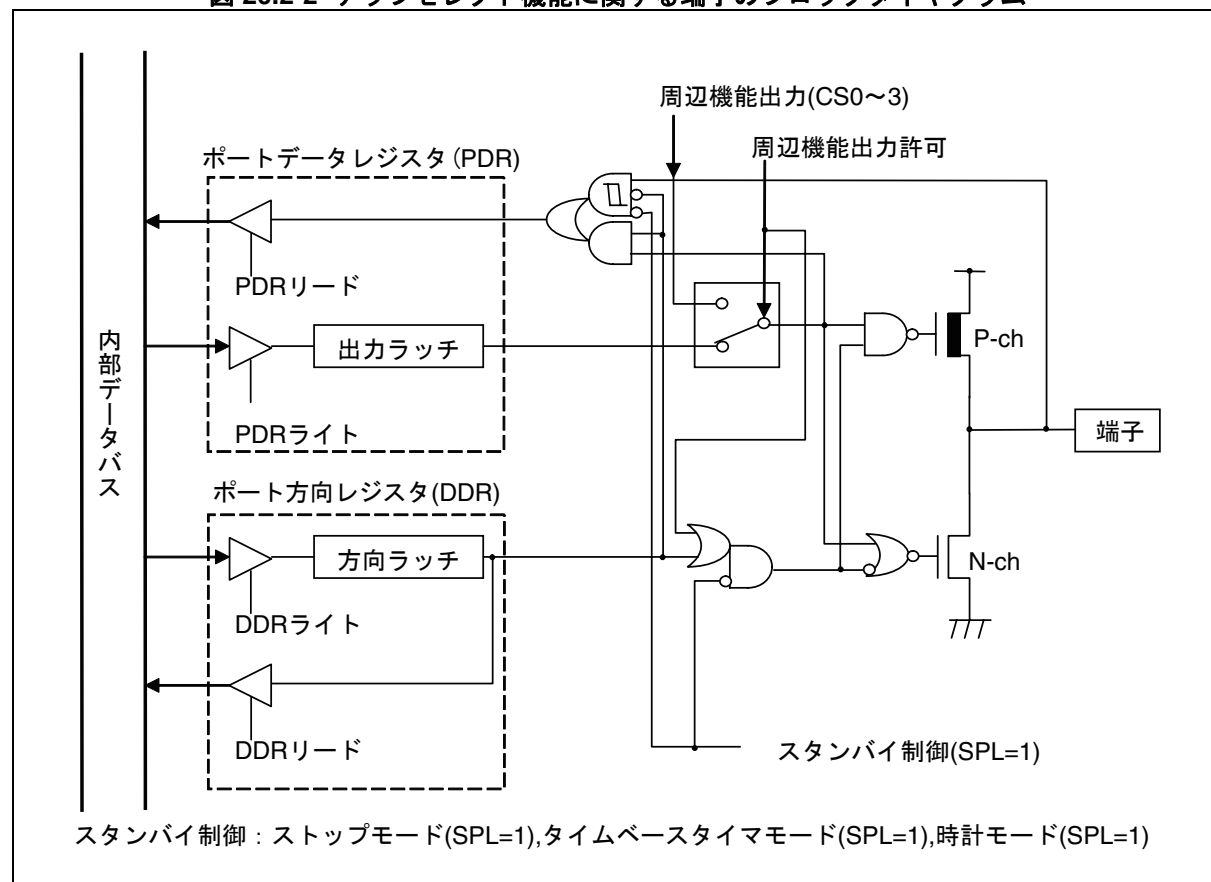
チップセレクト機能に関する端子は、4 本の CS0/CS1/CS2/CS3 出力端子があります。CS0/CS1/CS2/CS3 端子は、汎用入出力ポート (P90/CS0, P91/CS1, P92/CS2, P93/CS3) とチップセレクト機能の出力端子を兼用しています。

● CS0/CS1/CS2/CS3 端子として使用する場合の設定

チップセレクト機能で CS0/CS1/CS2/CS3 を出力として使用する場合は、チップセレクトコントロールレジスタ (CSCR) を出力許可 (OPL[3:0] → "1") に設定してください。また、この端子に割り付けられたほかのリソースは、チップセレクト機能を使用している場合は、使用できません。

■ チップセレクト機能に関する端子のブロックダイアグラム

図 20.2-2 チップセレクト機能に関する端子のブロックダイアグラム



20.3 チップセレクト機能のレジスタの構成と機能

チップセレクト機能に関連するレジスタについて示します。

■ チップセレクト機能のレジスタ一覧

図 20.3-1 に、チップセレクト機能のレジスタ一覧を示します。

図 20.3-1 チップセレクト機能のレジスタ一覧

15	8 7				0				
	CAR0				CMR0				(R/W)
	CAR1				CMR1				(R/W)
	CAR2				CMR2				(R/W)
	CAR3				CMR3				(R/W)
	CALR				CSCR				(R/W)

0000C0 _H	7	6	5	4	3	2	1	0	CMRx
0000C2 _H	M7	M6	M5	M4	M3	M2	M1	M0	チップセレクト領域MASKレジスタ
0000C4 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
0000C6 _H	(0)	(0)	(0)	(0)	(1)	(1)	(1)	(1)	初期値

0000C1 _H	15	14	13	12	11	10	9	8	CARx
0000C3 _H	A7	A6	A5	A4	A3	A2	A1	A0	チップセレクト領域レジスタ
0000C5 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	リード/ライト
0000C7 _H	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	初期値

	7	6	5	4	3	2	1	0	CSCR
0000C8 _H	—	—	—	—	OPL3	OPL2	OPL1	OPL0	チップセレクトコントロールレジスタ
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(*)	初期値

	15	14	13	12	11	10	9	8	CALR
0000C9 _H	—	—	—	—	ACTL3	ACTL2	ACTL1	ACTL0	チップセレクトアクティブレベルレジスタ
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(0)	初期値

20.3.1 チップセレクト領域 MASK レジスタ (CMRx)

チップセレクト領域 MASK レジスタ (CMRx) の構成と機能について説明します。

■ チップセレクト領域 MASK レジスタ (CMRx)

下図に , チップセレクト領域 MASK レジスタ (CMRx) のビット構成を示します。

0000C0 _H	7	6	5	4	3	2	1	0	CMRx
0000C2 _H	M7	M6	M5	M4	M3	M2	M1	M0	チップセレクト領域MASKレジスタ
0000C4 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
0000C6 _H	(0)	(0)	(0)	(0)	(1)	(1)	(1)	(1)	初期値
[bit7-0] M7-M0									

[bit7 ~ bit0] M7 ~ M0

これらのビットは , チップセレクト端子のアドレスデコード領域を設定します。"1" を書き込むことによって , そのビットはマスクされます。

128K バイト以上の領域を指定する場合に使用します。

<注意事項>

すべてのビットをマスクした場合は , 外部アクセス可能なすべての領域で CS 端子がアクティブになります。

20.3.2 チップセレクト領域レジスタ (CARx)

チップセレクト領域レジスタ (CARx) の構成と機能について説明します。

■ チップセレクト領域レジスタ (CARx)

下図に、チップセレクト領域レジスタ (CARx) のビット構成を示します。

0000C1 _H	15	14	13	12	11	10	9	8	CARx
0000C3 _H	A7	A6	A5	A4	A3	A2	A1	A0	チップセレクト領域レジスタ
0000C5 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	リード/ライト
0000C7 _H	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	初期値
[bit7-0] A7-A0									

[bit7 ~ bit0] A7 ~ A0

これらのビットは、チップセレクト端子のアドレスデコード領域を設定します。アドレスの上位 8 ビットを指定し、64K バイト単位の領域を指定することができます。

<注意事項>

CPU が内部アクセス (内蔵 RAM, 内蔵 ROM, I/O) 中では、CS 端子はアクティブになりません。

20.3.3 チップセレクトコントロールレジスタ (CSCR)

チップセレクトコントロールレジスタ (CSCR) の構成と機能について説明します。

■ チップセレクトコントロールレジスタ (CSCR)

下図に、チップセレクトコントロールレジスタ (CSCR) のビット構成を示します。

	7	6	5	4	3	2	1	0	CSCR
0000C8 _H	—	—	—	—	OPL3	OPL2	OPL1	OPL0	チップセレクトコントロールレジスタ
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(*)	初期値
* : このビットの初期値は, "1" または "0" です。モード端子 (MD2, MD1, MD0 端子) によって異なります。									

[bit7 ~ bit4] 未使用ビット

これらのビットは、未使用ビットです。読出し時は、値は不定です。

[bit3 ~ bit0] OPL3 ~ OPL0

これらのビットは、CS3 ~ CS0 の各出力を外部端子に出力するか、否かを設定するビットです。

以下のように動作が設定されます。

- "0":CS3 ~ CS0 の各端子からのデコード出力を禁止します。
- "1":CS3 ~ CS0 の各端子からのデコード出力を許可します。

<注意事項>

- OPL0 の初期値は、外部ベクタモード時は、"1" になります。内部ベクタモード時は、"0" になります。
- CS3 ~ CS0 端子の出力を許可する場合は、すべての設定を行った後、最後に OPL3 ~ OPL0 を出力許可に設定してください。
- 途中で設定を変更する場合は、必ず、OPL3 ~ OPL0 を出力禁止にしてから行ってください。

20.3.4 チップセレクトアクティブレベルレジスタ (CALR)

チップセレクトアクティブレベルレジスタ (CALR) の構成と機能について説明します。

■ チップセレクトアクティブレベルレジスタ (CALR)

下図に、チップセレクトアクティブレベルレジスタ (CALR) のビット構成を示します。

	15	14	13	12	11	10	9	8	CALR
0000C9 _H	—	—	—	—	ACTL3	ACTL2	ACTL1	ACTL0	チップセレクトアクティブレベルレジスタ
	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
	(—)	(—)	(—)	(—)	(0)	(0)	(0)	(0)	初期値

[bit15 ~ bit12] 未使用ビット

これらのビットは、未使用ビットです。読出し時の値は不定です。

[bit11 ~ bit8] ACTL3 ~ ACTL0

これらのビットは、CS3 ~ CS0 の各端子のアクティブレベルを設定します。

以下のように動作が設定されます。

- "0":CS3 ~ CS0 の各端子は、デコード時に "L" を出力します。
- "1":CS3 ~ CS0 の各端子は、デコード時に "H" を出力します。

<注意事項>

- アクティブレベルを変更するときは、必ずチップセレクトコントロールレジスタで、出力を禁止にしてから変更してください。
- ワードでの書込みは、禁止です。必ずバイトで書き込んでください。これは、書込みと同時に出力許可ができることと、アクティブレベルの変更を防ぐためです。

20.4 チップセレクト機能の動作

チップセレクト機能の動作について説明します。

■ 動作概要

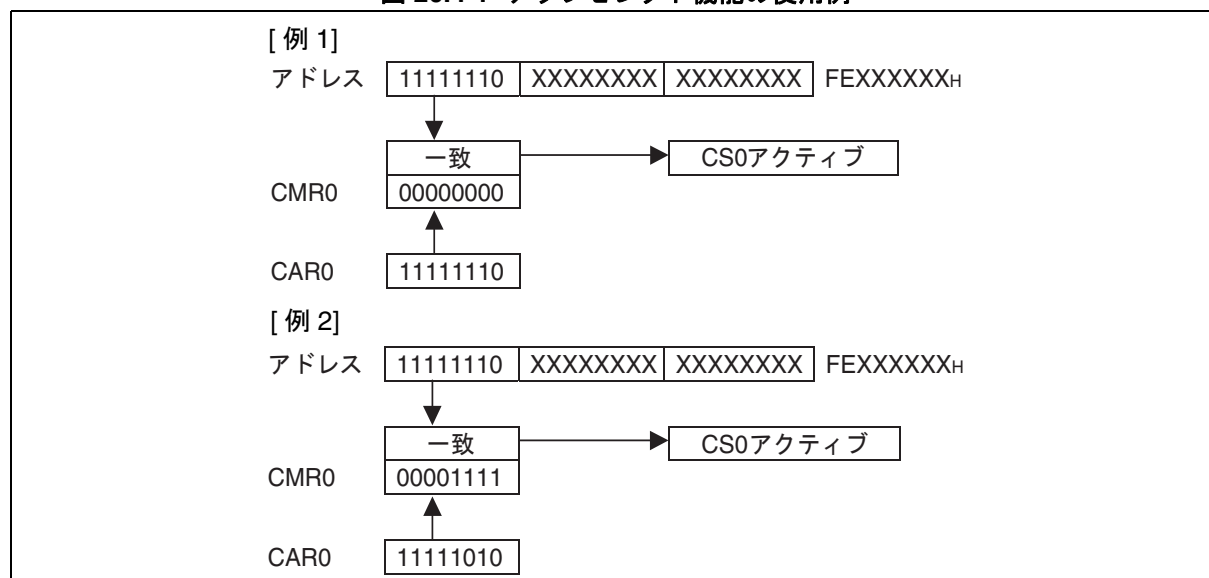
CPU がプログラム、またはデータにアクセスする際のアドレスの上位 8 ビットと CAR0/1/2/3 の一致を検出することによって、チップセレクトをアクティブにします。また、CMR0/1/2/3 に "1" が設定されたビットは無視しますので、64K バイト～16M バイトのデコードが可能です。

なお、CPU が内部アクセス (I/O、内蔵 RAM、内蔵 ROM) 中では、CS 端子はアクティブになりません。

■ チップセレクト機能の使用例

図 20.4-1 に、チップセレクト機能の使用例を示します。

図 20.4-1 チップセレクト機能の使用例



■ チップセレクト機能の使用上の注意点

- CS0 の端子は、外部ベクタモードの場合、リセットベクタを読むために、必ずアクティブになります。アドレス F00000_H～FFFFFF_H (1M バイトの空間：初期値)において、リセット直後にデコード信号を出力しますので、本端子は必ずプログラム ROM 用端子としてご使用ください。その場合の CS0 端子のアクティブレベルは "L" となり、リセット中は "H" が出力されます。また、内部ベクタモード時は、ほかの CS3～1 端子と同様、汎用ポートになりますので、設定後、CS0 の端子は出力状態に切り換えてください。
- 必ず、チップセレクト領域レジスタ、チップセレクト領域 MASK レジスタ、チップセレクトアクティブレベルレジスタを設定してから、出力許可を行ってください。
- チップセレクト出力は、P90～P93 端子と共用のため、この端子に割り付けられたリソースを利用する場合には、使用できませんので注意してください。
- 外部バス使用時にホールド状態にした場合、出力がディセーブルになり、ハイインピーダンス状態になりますので、兼用する汎用ポートは、必ず入力にしておいてください。
- スリープ、ストップ時に、CS 端子はアクティブになりません。
- 内蔵 DMA アクセス中には、チップセレクト機能は使用できません。

第21章

アドレス一致検出機能

アドレス一致検出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 命令に置き換え、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

- プログラムアドレス検出レジスタ (PADR0, PADR1) 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

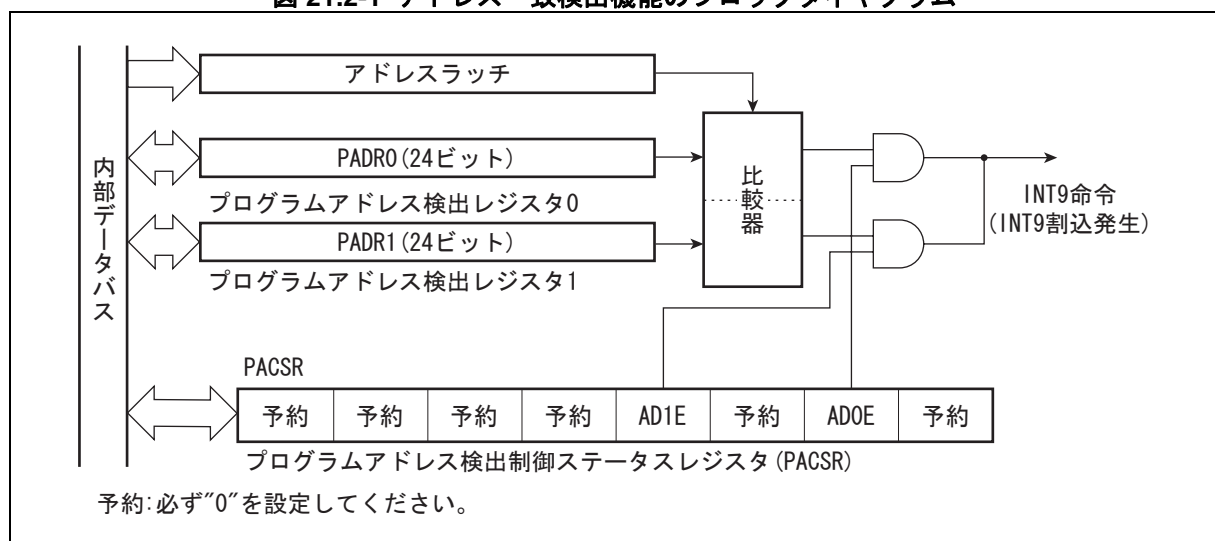
アドレス一致検出モジュールは、以下のブロックで構成されています。

- 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

図 21.2-1 に、アドレス一致検出機能のブロックダイアグラムを示します。

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 (PADR0, PADR1)
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。

<注意事項>

プログラムアドレス検出レジスタは、1FF0_H～1FF5_Hにレジスタをもっているため、RAM 領域と重なりますので、本機能をご使用の場合は、RAM アクセスを行わないようにしてください。

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初期値の一覧

図 21.3-1 アドレス一致検出機能のレジスタと初期値の一覧

プログラムアドレス検出制御ステータス レジスタ (PACSR) アドレス 009E _H	ビット	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 (PADR0): 上位 アドレス 1FF2 _H	ビット	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 (PADR0): 中位 アドレス 1FF1 _H	ビット	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 (PADR0): 下位 アドレス 1FF0 _H	ビット	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 (PADR1): 上位 アドレス 1FF5 _H	ビット	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 (PADR1): 中位 アドレス 1FF4 _H	ビット	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 (PADR1): 下位 アドレス 1FF3 _H	ビット	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)

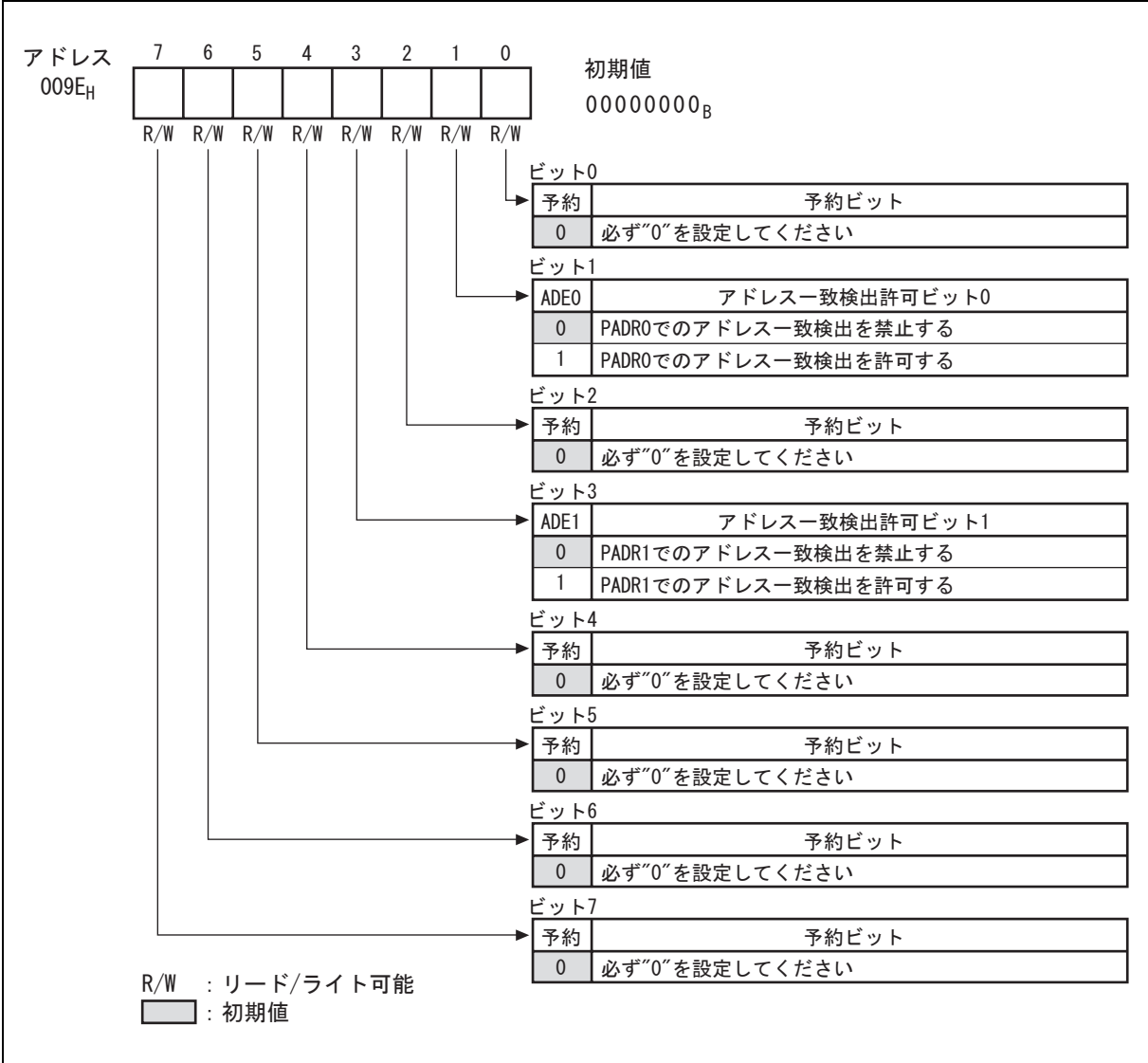


表 21.3-1 プログラムアドレス検出制御ステータスレジスタ (PACSR) の機能

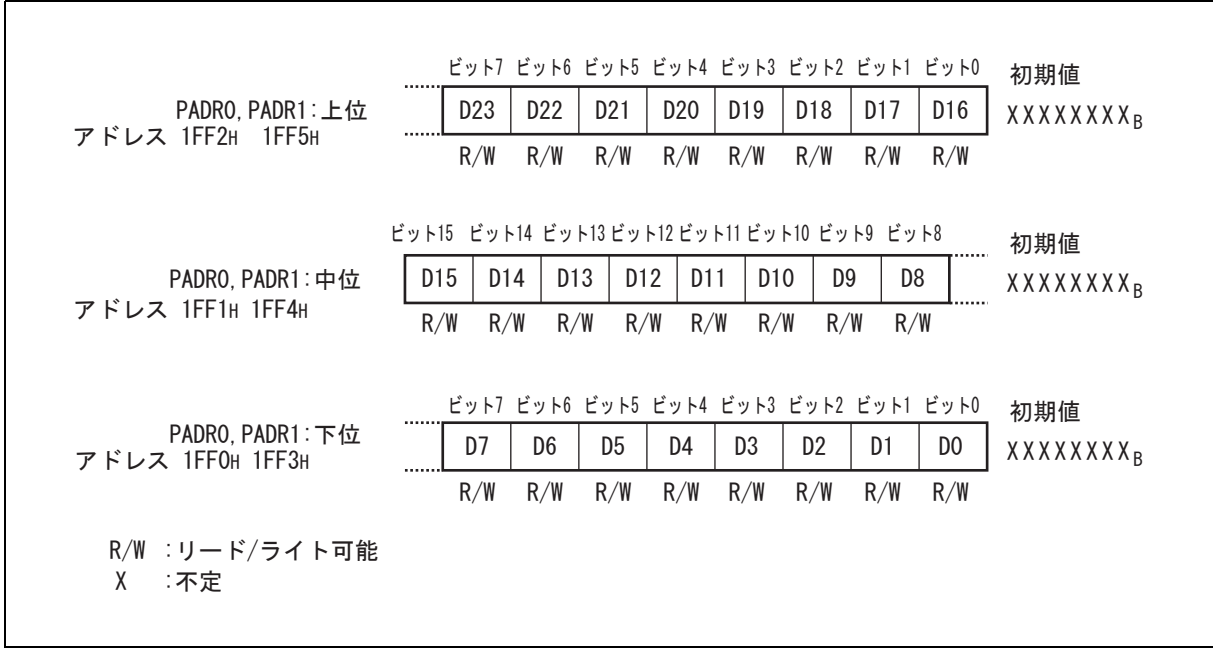
ビット名		機能
bit7 ～ bit4	予約 : 予約ビット	必ず "0" を設定してください。
bit3	AD1E: アドレス一致 検出許可ビット 1	プログラムアドレス検出レジスタ 1 (PADR1) とのアドレス一致検出動作を許可または禁止します。 このビットに "0" を設定した場合 : アドレス一致検出動作を禁止します。 このビットに "1" を設定した場合 : アドレス一致検出動作を許可します。 アドレス一致検出動作を許可すると (AD1E=1), アドレスラッチの値とプログラムアドレス検出レジスタ 1 (PADR1) の値が一致した場合は, 直ちに INT9 命令が実行されます。
bit2	列 : 列ビット	必ず "0" を設定してください。
bit1	AD01: アドレス一致 検出許可ビット 0	プログラムアドレス検出レジスタ 0 (PADR0) とのアドレス一致検出動作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 : アドレス一致検出動作を禁止します。 このビットに "1" を設定した場合 : アドレス一致検出動作を許可します。 アドレス一致検出動作を許可すると (AD0E=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 (PADR0) の値が一致した場合は, 直ちに INT9 命令が実行されます。
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 (PADR0, PADR1)

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 命令に置き換えられ、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 (PADR0, PADR1)

図 21.3-3 プログラムアドレス検出レジスタ (PADR0, PADR1)



■ 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は2本 (PADR0, PADR1) 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(PADR0)	PACSR:AD0E	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(PADR1)	PACSR:AD1E	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 (PADR0, PADR1) には, INT9 命令と置き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 命令と置き換える命令コードの先頭アドレスの設定

				検出アドレスに設定 (上位:FF _H , 中位:00 _H , 下位:1F _H)	
FF001C :	A8	00	00	MOVW	RW0, #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 (PADR0, PADR1) 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 命令に置き換えられず, 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_H" (INT9 命令コード) に置き換わってしまい, 誤作動を起こす原因になる場合がありますので注意してください。

プログラムアドレス検出レジスタ (PADR0, PADR1) 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 AD0E: bit1=0 または AD1E: bit3=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。

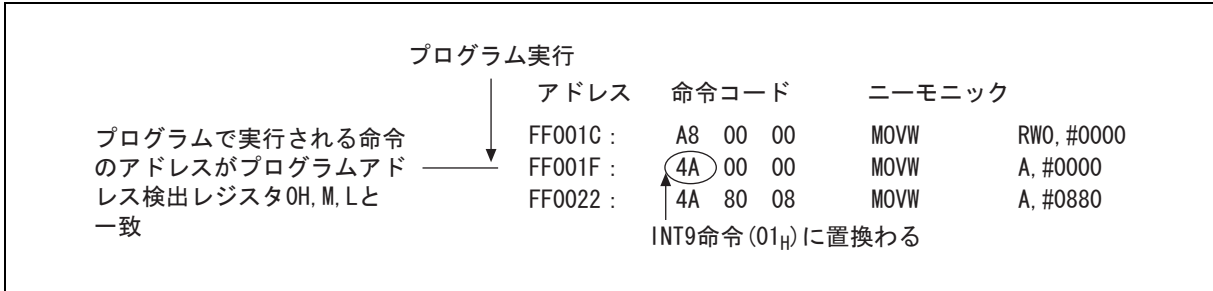
21.4 アドレス一致検出機能の動作説明

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 (PADR0, PADR1) 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_H) 命令に置き換え、割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

図 21.4-1 に、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方法

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(PADR0) 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 AD0E: bit1=0)。プログラムアドレス検出レジスタ 0(PADR0) 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(PADR0) の上位に "FF_H", 中位に "00_H" 下位に "1F_H" を設定してください。

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(PADR0) 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 AD0E: bit1=1)。

■ プログラムの実行

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_H") に置き換わります。

INT9 命令が実行されると、INT9 割込みが発生し、割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

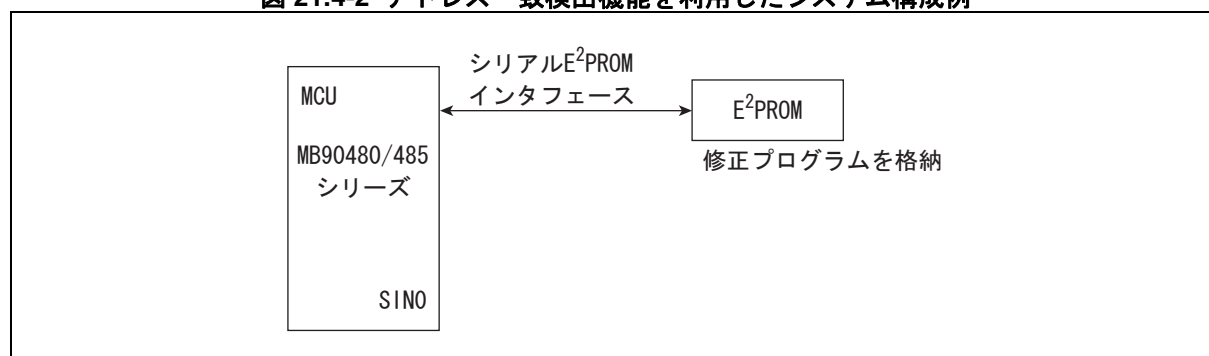
アドレス一致検出機能を利用した，プログラム修正のパッチ処理の例を示します。

■ システム構成と E²PROM のメモリ構成

● システム構成

図 21.4-2 に，アドレス一致検出機能を利用したシステム構成例を示します。

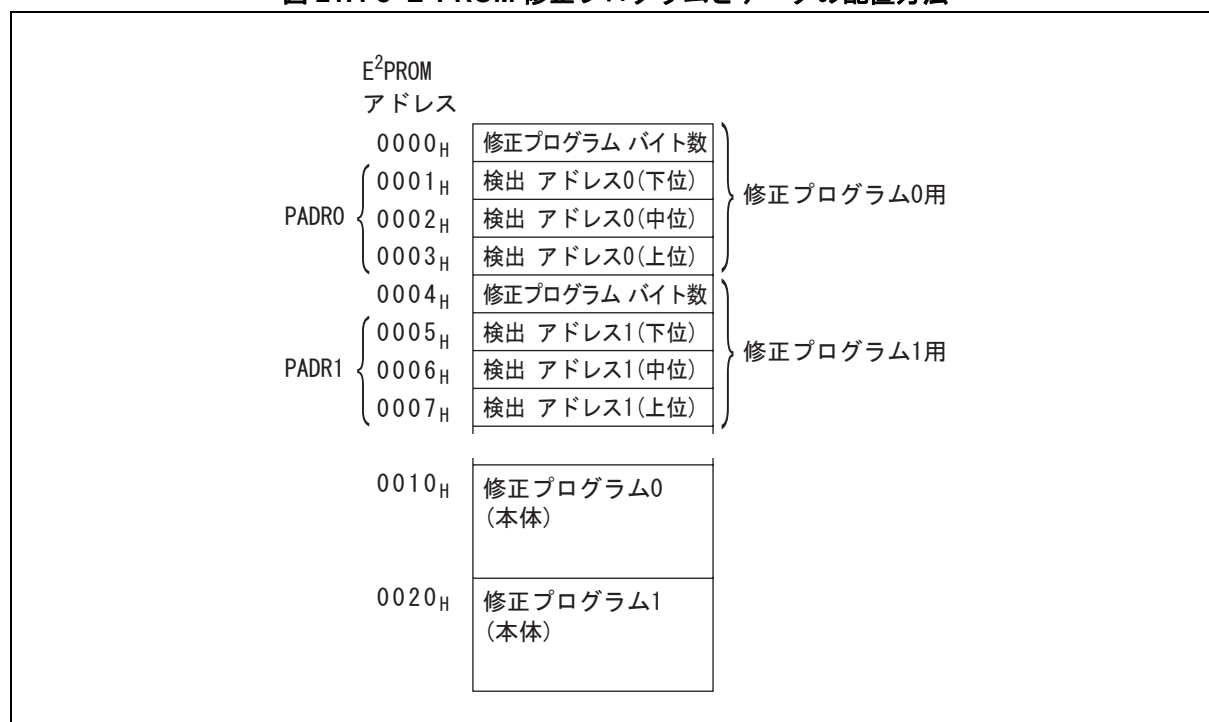
図 21.4-2 アドレス一致検出機能を利用したシステム構成例



■ E²PROM のメモリマップ

修正プログラムを E²PROM 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 E²PROM 修正プログラムとデータの配置方法」に示します。

図 21.4-3 E²PROM 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_H" 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 命令に置き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 (PADR0, PADR1) 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

E²PROM のデータはすべて "00_H" 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 (MB90480/485 シリーズ) へ , E²PROM 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 (MB90480/485 シリーズ) 側では , 外部から受け取った修正プログラムの情報を E²PROM へ格納します。

- リセットシーケンス

リセット後 , MCU (MB90480/485 シリーズ) は , E²PROM 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_H" 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 (PADR0, PADR1) 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 (MB90480/485 シリーズ) 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 AD0E: bit1=1, AD1E: bit3=1)。

- INT9 割込み処理

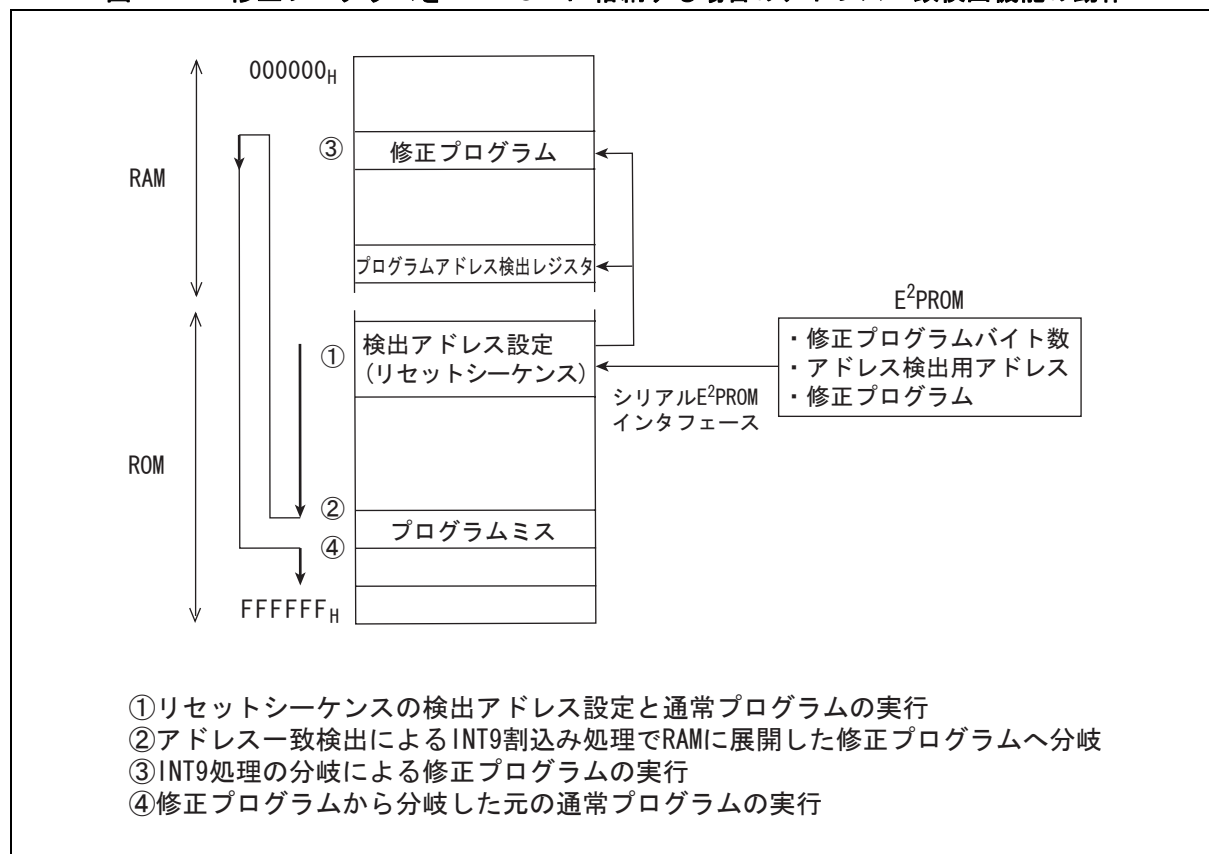
INT9 命令によって割込み処理が実行されます。MB90480/485 シリーズには , アドレス一致検出による割込み要求フラグがありません。したがって , プログラムカウンタのスタック情報が破棄されると検出アドレスを確認できません。検出アドレスを確認する場合は , 割込み処理ルーチンでスタックされているプログラムカウンタの値を確認してください。

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 E²PROM に格納する場合のアドレス一致検出機能の動作

図 21.4-4 に、修正プログラムを E²PROM に格納する場合のアドレス一致検出機能の動作を示します。

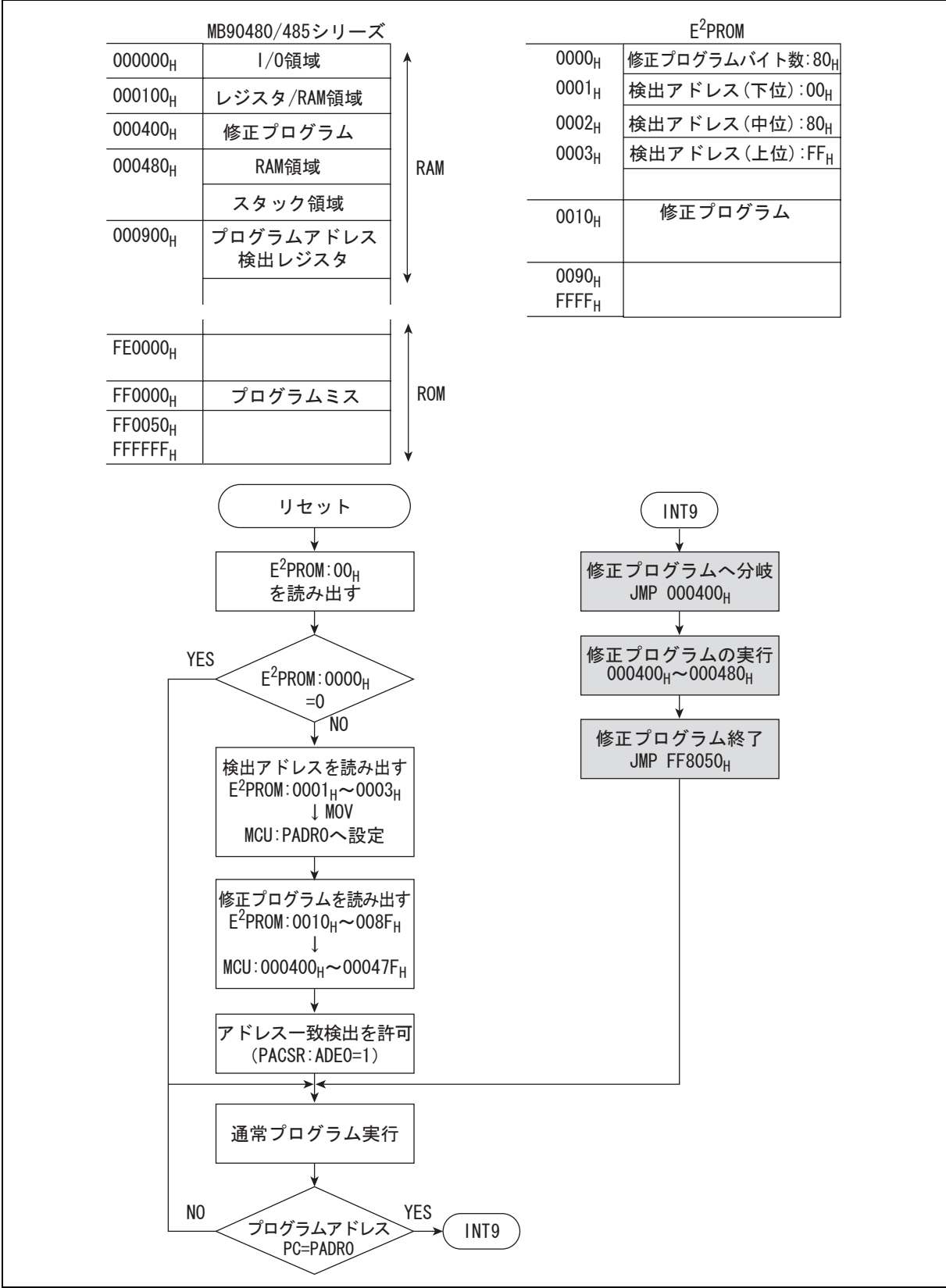
図 21.4-4 修正プログラムを E²PROM に格納する場合のアドレス一致検出機能の動作



■ プログラム修正のパッチ処理フロー

図 21.4-5 に、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

プログラムで実行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 (PADR0) 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概要およびレジスタ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)

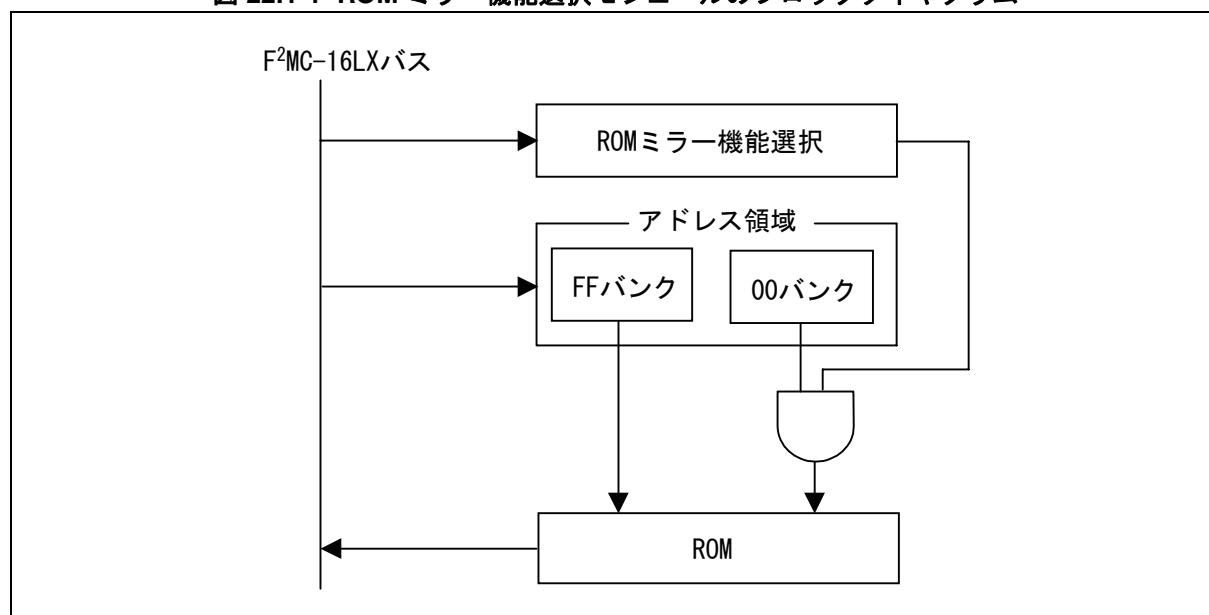
22.1 ROM ミラー機能選択モジュールの概要

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で読み出せるように設定します。

■ 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 に、ROM ミラー機能選択モジュールのブロックダイアグラムを示します。

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 ミラー機能選択モジュールのレジスタ

下図に、ROM ミラー機能選択モジュールの構成を示します。

図 22.1-2 ROM ミラー機能選択レジスタ (ROMM)

ビット	15	14	13	12	11	10	9	8	初期値
アドレス: 00006F _H	—	—	—	—	—	—	MS	MI	----- (+) 1 _B
							R/W (+)	R/W	

(+) : MB90F489B : リードオンリ, "1" に固定
 その他 : 選択可能, 初期値 "0"

22.2 ROM ミラー機能選択レジスタ (ROMM)

ROM ミラー機能選択レジスタ (ROMM) の構成および機能について説明します。

■ ROM ミラー機能選択レジスタ (ROMM)

下図に、ROM ミラー機能選択レジスタ (ROMM) のビット構成を示します。

ビット	15	14	13	12	11	10	9	8	初期値
ROMM アドレス: 00006F _H	—	—	—	—	—	—	MS R/W (+)	MI R/W	----- (+) 1 _B
(+)	: MB90F489B: リードオンリ, "1" に固定 その他: 選択可能, 初期値 "0"								

[bit9] MS

このビットは ROM ミラーの領域を選択します。

"1": ROM ミラー領域が 32K バイト (008000_H ~ 00FFFF_H) となります。

"0": ROM ミラー領域が 48K バイト (004000_H ~ 00FFFF_H) となります。

<注意事項>

MB90F489B では、このビットは "1" に固定され、読出しのみ使用できます。
他のデバイスは選択可能です。

[bit8] MI

このビットは 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。

"1": ミラー機能を有効にします。

"0": ミラー機能を無効にします。

<注意事項>

- このレジスタは、アドレス 004000_H ~ 00FFFF_H(008000_H ~ 00FFFF_H) 番地の使用中にアクセスしないでください。
- ROM ミラー機能を起動している場合、00 バンクの "004000_H ~ 00FFFF_H(008000_H ~ 00FFFF_H) " 番地へ "FF4000_H ~ FFFFFFF_H(FF8000_H ~ FFFFFFF_H) " 番地のデータがミラーされます。"FF3FFF_H(FF7FFF_H) " 番地以下の ROM アドレスでは ROM のミラー機能を設定しても 00 バンクにミラーされません。

() 内は MS ビットが "1" の場合のアドレスです。

第23章

2M/3M ビットフラッシュメモリ

2M/3M ビットフラッシュメモリの機能や動作について説明します。

- 23.1 2M/3M ビットフラッシュメモリの概要
- 23.2 2M/3M ビットフラッシュメモリのセクタ構成
- 23.3 フラッシュメモリコントロールステータスレジスタ (FMCS)
- 23.4 フラッシュメモリの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 フラッシュメモリの書込み / 消去
- 23.7 フラッシュセキュリティ機能

23.1 2M/3M ビットフラッシュメモリの概要

2M/3M ビットフラッシュメモリは、CPU メモリマップ上の FC ～ FF バンクに配置され、フラッシュメモリインタフェース回路の機能によって、マスク ROM と同様に CPU からの読出しアクセスおよびプログラムアクセスが可能です。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フェース回路を介して、CPU からの命令動作により実行されます。このため、内蔵 CPU の制御による実装状態での書換えが可能となり、プログラムおよびデータの改善が効率よくできます。

なお、イネーブルセクタプロテクトなどのセクタオペレーションは使用できません。

■ 2M/3M ビットフラッシュメモリの特長

2M/3M ビットフラッシュメモリは、以下の特長をもっています。

- 2M: 256K ワード× 8/128K ワード× 16 ビット (16K+8K+8K+32K+64K+64K+64K) セクタ構成
- 3M: 384K ワード× 8/192K ワード× 16 ビット (16K+8K+8K+32K+64K+64K+64K+64K) セクタ構成
- 自動プログラムアルゴリズム (Embedded Algorithm: MBM29F400TA 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 / 消去の完了検出
- CPU 割込みによる書込み / 消去の完了検出
- JEDEC 標準型コマンドと互換
- セクタごとの消去が可能 (セクタの組合せ自由)
- 書込み / 消去回数 (最小) 10000 回

Embedded Algorithm は、Advanced Micro Devices, Inc. の商標です。

■ フラッシュメモリの書込み / 消去の方法

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データ書込み / 消去の動作をさせる際には、フラッシュメモリ上にあるプログラムを RAM に一度コピーし、RAM 上で実行してください。これによって、フラッシュメモリの読出しをせずに書込み動作だけを行うことが可能となります。

■ フラッシュメモリコントロールステータスレジスタ (FMCS)

下図に、フラッシュメモリで使用するフラッシュメモリコントロールステータスレジスタ (FMCS) のビット構成を示します。

ビットNo.	7	6	5	4	3	2	1	0
アドレス: 0000AE _H	INTE	RDYINT	WE	RDY	予約	LPM1	予約	LPM0
リード/ライト	(R/W)	(R/W)	(R/W)	(W)	(W)	(R/W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(X)	(0)	(0)	(0)	(0)

23.2 2M/3M ビットフラッシュメモリのセクタ構成

2M/3M ビットフラッシュメモリのセクタ構成を示します。

■ セクタ構成

図 23.2-1 に、2M/3M ビットフラッシュメモリのセクタ構成を示します。図中のアドレスには、各セクタの上位アドレスと下位アドレスを示します。

CPU からアクセスする場合、FC バンクレジスタに SA0, FD バンクレジスタに SA1, FE バンクレジスタに SA2, FF バンクレジスタに SA3 ～ 6 が配置されています。

図 23.2-1 2M/3M ビットフラッシュメモリのセクタ構成

2Mビットフラッシュメモリ			3Mビットフラッシュメモリ		
	ライタアドレス	CPUアドレス		ライタアドレス	CPUアドレス
SA6 (16Kバイト)	7FFFFH	FFFFFFH	SA8 (16Kバイト)	7FFFFH	FFFFFFH
	7C000H	FFC000H		7C000H	FFC000H
SA5 (8Kバイト)	7BFFFH	FFBFFFH	SA7 (8Kバイト)	7BFFFH	FFBFFFH
	7A000H	FFA000H		7A000H	FFA000H
SA4 (8Kバイト)	79FFFH	FF9FFFH		79FFFH	FF9FFFH
	78000H	FF8000H	SA6 (8Kバイト)	78000H	FF8000H
SA3 (32Kバイト)	77FFFH	FF7FFFH		77FFFH	FF7FFFH
	70000H	FF0000H	SA5 (32Kバイト)	70000H	FF0000H
SA2 (64Kバイト)	6FFFFH	FEFFFFH		6FFFFH	FEFFFFH
	60000H	FE0000H	SA4 (64Kバイト)	60000H	FE0000H
SA1 (64Kバイト)	5FFFFH	FDFFFFH		5FFFFH	FDFFFFH
	50000H	FD0000H	SA3 (64Kバイト)	50000H	FD0000H
SA0 (64Kバイト)	4FFFFH	FCFFFFH		4FFFFH	FCFFFFH
	40000H	FC0000H	未使用	40000H	FC0000H
				3FFFFH	FBFFFFH
			SA2 (64Kバイト)	30000H	FB0000H
				2FFFFH	FAFFFFH
			SA1 (64Kバイト)	20000H	FA0000H
				1FFFFH	F9FFFFH
			SA0 (64Kバイト)	10000H	F90000H
				0FFFFH	F8FFFFH
			未使用	00000H	F80000H

● ライタアドレス

図 23.2-1 の中のライタアドレスとは、フラッシュメモリにパラレルライタでデータを書き込む際、CPU アドレスに相当するアドレスです。汎用ライタを使用して書込み/消去を行う際には、このアドレスで書込み/消去を実行します。

23.3 フラッシュメモリコントロールステータスレジスタ (FMCS)

フラッシュメモリコントロールステータスレジスタ (FMCS) は、フラッシュメモリインタフェース回路にあるレジスタで、フラッシュメモリの書込み / 消去の際に使用します。

■ フラッシュメモリコントロールステータスレジスタ (FMCS)

下図に、フラッシュメモリコントロールステータスレジスタ (FMCS) のビット構成を示します。

ビットNo.	7	6	5	4	3	2	1	0
アドレス: 0000AE _H	INTE	RDYINT	WE	RDY	予約	LPM1	予約	LPM0
リード/ライト	(R/W)	(R/W)	(R/W)	(W)	(W)	(R/W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(X)	(0)	(0)	(0)	(0)

以下に、フラッシュメモリコントロールステータスレジスタ (FMCS) の各ビットの機能を説明します。

[bit7] INTE:INTerrupt Enable

このビットは、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。

INTE ビットが "1" で、かつ RDYINT ビットが "1" のときに CPU へ割込みが発生します。INTE ビットが "0" であれば、割込みは発生しません。

0	書込み / 消去終了での割込み禁止
1	書込み / 消去終了での割込み許可

[bit6] RDYINT:ReaDY INTerrupt

このビットは、フラッシュメモリの動作状態を表します。

フラッシュメモリの書込み / 消去が終了後、"1" となります。フラッシュメモリ書込み / 消去後、このビットが "0" の間は、フラッシュメモリへの書込み / 消去はできません。書込み / 消去が終了して "1" になった後は、フラッシュメモリへの書込み / 消去が可能になります。

"0" の書込みによって "0" にクリアされます。"1" の書込みは無視されます。フラッシュメモリ自動アルゴリズム (「23.4 フラッシュメモリの自動アルゴリズム起動方法」参照) 終了タイミングで、"1" にセットされます。リードモディファイライト (RMW) 命令使用時は、必ず "1" が読み出されます。

0	書込み / 消去動作実行中
1	書込み / 消去動作終了 (割込み要求発生)

[bit5] WE:Write Enable

このビットは、フラッシュメモリ領域への書込みイネーブルビットです。

このビットが "1" のとき、FC ～ FF バンクへのコマンドシーケンス（「23.4 フラッシュメモリの自動アルゴリズム起動方法」参照）発行後の書込みは、フラッシュメモリ領域への書込みになります。このビットが "0" の場合、書込み / 消去の信号は発生されません。このビットは、フラッシュメモリの書込み / 消去のコマンドを起動する際に使用します。

書込み / 消去を行わないときは、誤ってフラッシュメモリにデータを書き込まないように、常に "0" に設定することを推奨します。

0	フラッシュメモリ書込み / 消去禁止
1	フラッシュメモリ書込み / 消去許可

[bit4] RDY:ReadDY

このビットは、フラッシュメモリの書込み / 消去許可ビットです。

このビットが "0" の間は、フラッシュメモリへの書込み / 消去はできません。なお、この状態でも読出し / リセットコマンド、セクタ消去一時停止コマンドは受け付けられます。

0	書込み / 消去動作実行中
1	書込み / 消去動作終了（次データ書込み / 消去許可）

[bit3] 予約ビット

このビットは、予約ビットです。通常使用時は、必ず "0" に設定してください。

[bit1] 予約ビット

このビットは、予約ビットです。通常使用時は、必ず "0" に設定してください。

[bit2, bit0] LPM1, LPM0:Low Power Mode

これらのビットは、フラッシュメモリの消費電力を制御します。このビットが "00" のとき、フラッシュメモリは通常動作となります。また、"01", "10", "11" のときは、フラッシュメモリアクセス時のフラッシュメモリへのセレクト信号により、フラッシュメモリ本体の消費電力が抑えられます。しかし、アクセスタイムは LPM=00 のときに比べて大幅に大きくなりますので、CPU の高速動作時はメモリアクセスが不可能になります。このモードを使用するときは、それぞれ CPU を 4MHz, 8MHz, 10MHz の周波数で動作させてください。

LPM1	LPM0	消費電力モード
0	0	通常消費電力モード
0	1	低電力消費モード（内部動作周波数 4MHz 以下で動作）
1	0	低電力消費モード（内部動作周波数 8MHz 以下で動作）
1	1	低電力消費モード（内部動作周波数 10MHz 以下で動作）

<注意事項>

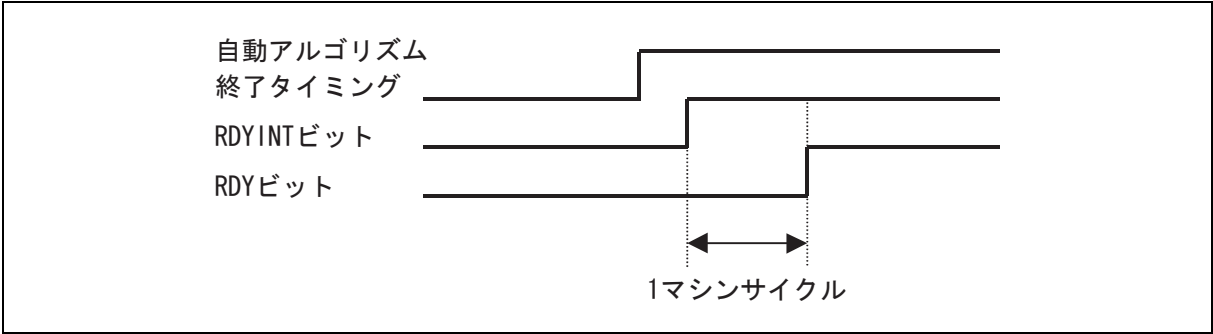
CPU の周波数を 10MHz を超えて使用する場合は、必ず通常モードで使用してください。

■ 自動アルゴリズム終了タイミング

図 23.3-1 に、自動アルゴリズム終了タイミングと RDYINT ビットおよび RDY ビットの関係を示します。

RDYINT ビットと RDY ビットは同時には変化しません。どちらか一方のビットで自動アルゴリズム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。

図 23.3-1 自動アルゴリズム終了タイミングと RDYINT ビットおよび RDY ビットの関係



■ サブクロックモード使用時の注意点

サブクロックモードを使用する場合には、表 23.3-1 に示すとおりフラッシュメモリコントロールステータスレジスタ (AE_H 番地 :FMCS レジスタ) の低消費電力モード選択ビット (LPM1, LPM0) を、通常消費電力モード (LPM1, LPM0=0, 0) 以外に設定する必要があります。したがって、内部クロックが 10MHz を超えて動作している状態からサブクロックモードへ遷移することはできません。

表 23.3-1 低消費電力モード選択ビット

LPM1	LPM0	低消費電力モード選択ビット	
		低消費電力モード	サブクロックモード時の設定
0	0	通常消費電力モード	× : 設定禁止
0	1	低消費電力モード 内部クロック 4MHz 以下	○ : 設定可
1	0	低消費電力モード 内部クロック 8MHz 以下	○ : 設定可
1	1	低消費電力モード 内部クロック 10MHz 以下	○ : 設定可

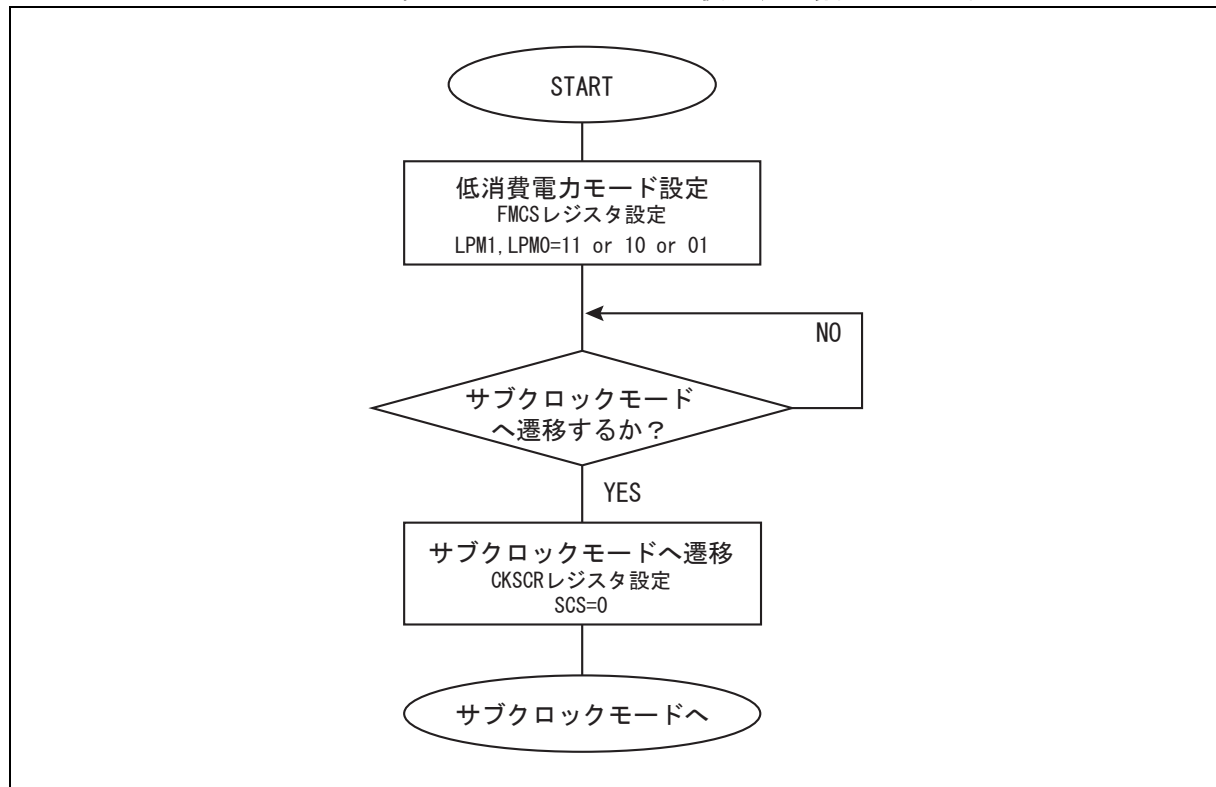
■ サブクロックモード設定方法

サブクロックモードを使用する場合には、以下の設定が必要となります。

● 内部クロック 10MHz 以下で使用する場合

サブクロックモードを使用する場合には、初期設定にてフラッシュメモリコントロールステータスレジスタの低消費電力モード選択ビットの変更 (LPM1, LPM0=0, 0 以外) を行ってください。図 23.3-2 に、フロー例を示します。

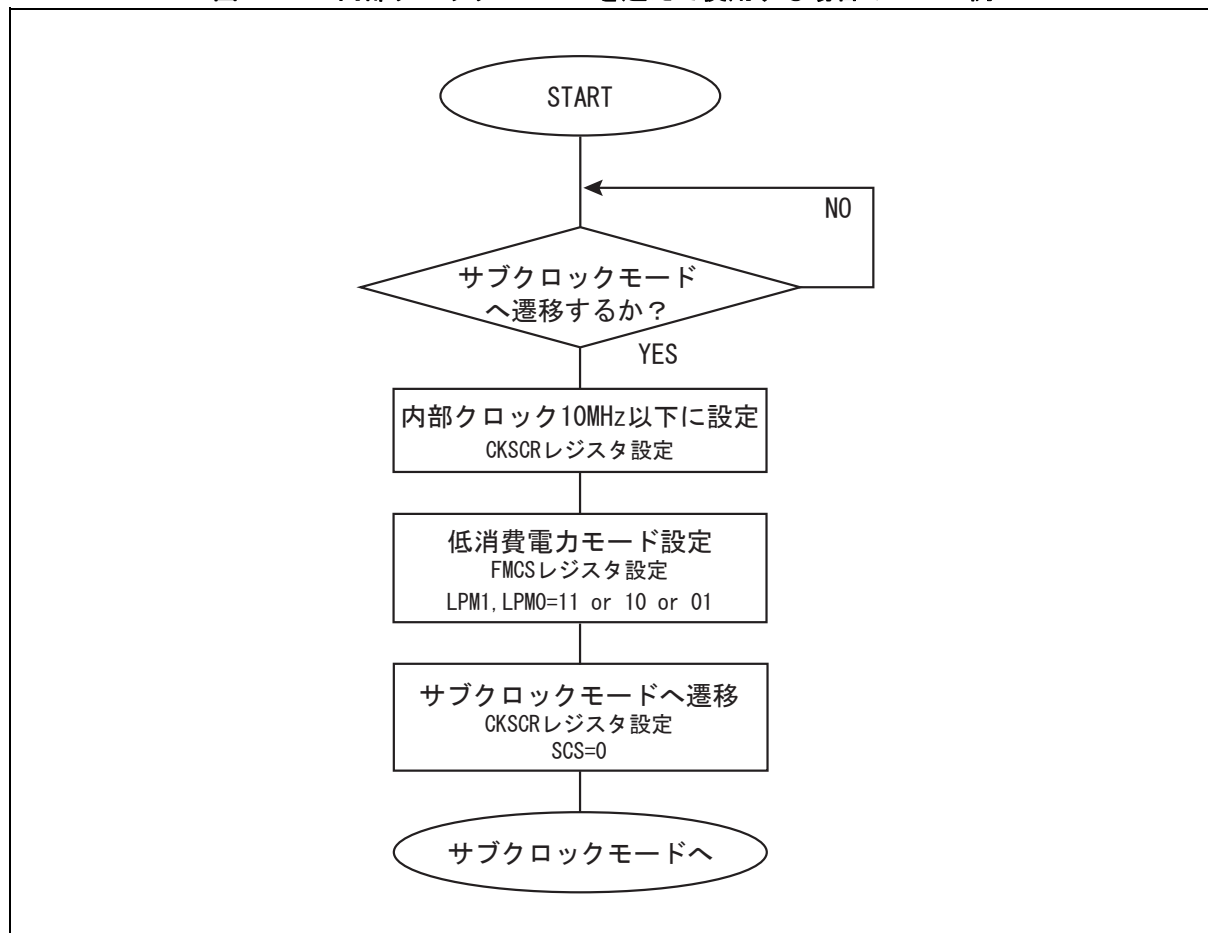
図 23.3-2 内部クロック 10MHz 以下で使用する場合のフロー例



● 内部クロック 10MHz を超えて使用する場合

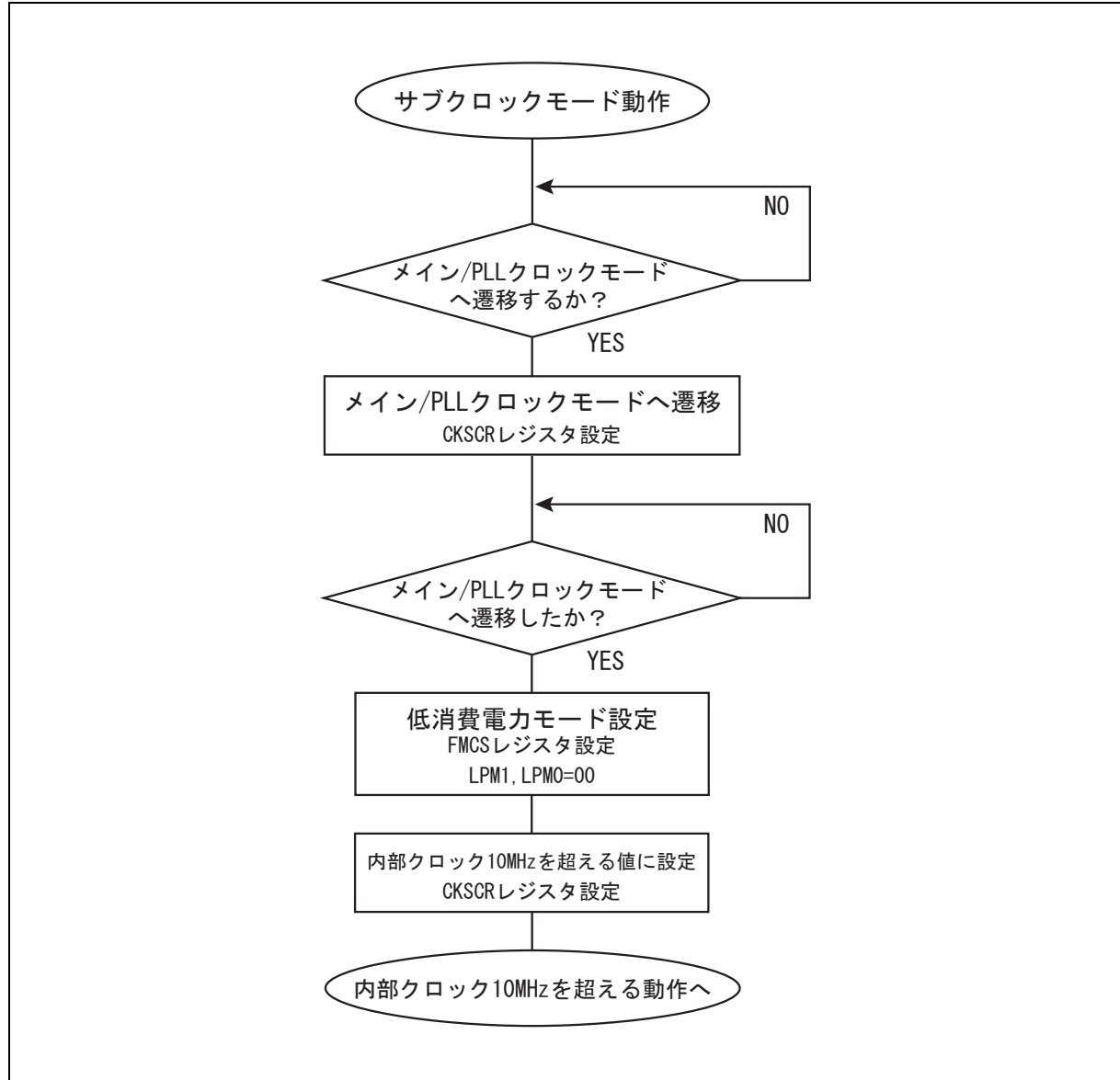
サブクロックモードを使用する場合には、サブクロックモードへ遷移する前に、まず内部クロックを 10MHz 以下に設定し、次に低消費電力モード選択ビットの変更 (LPM1, LPM0=0, 0 以外) を行ってください。図 23.3-3 に、フロー例を示します。

図 23.3-3 内部クロック 10MHz を超えて使用する場合のフロー例 1



サブクロックモード解除後，内部クロック 10MHz を超えて動作させる場合には，メインクロックモードまたは PLL クロックモード（内部クロックを 10MHz 以下）へ遷移した後に，低消費電力モード選択ビットの変更（LPM1, LPM0=0, 0）を行い，その後，内部クロックを 10MHz を超える値に設定してください。図 23.3-4 に，フロー例を示します。

図 23.3-4 内部クロック 10MHz を超えて使用する場合のフロー例 2



<注意事項>

サブクロックモードを使用する場合には，メインクロックに 20MHz 以下のクロック入力を行ってください。20MHz を超えるクロック入力を行った場合，内部クロック 10MHz 以下の設定が行えませんが，サブクロックモードは使用できません。

23.4 フラッシュメモリの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、チップ消去、およびセクタ消去の 4 種類があります。セクタ消去については、一時停止と再開の制御が可能です。

■ 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書き込むデータはすべてバイトですが、ワードアクセスで書き込むようにしてください。このときの上位バイト分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 書込み サイクル	1st バス書込み サイクル		2nd バス書込み サイクル		3rd バス書込み サイクル		4th バス書込み サイクル		5th バス書込み サイクル		6th バス書込み サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し / リセット *	1	FxXXXX _H	XXF0 _H	-	-	-	-	-	-	-	-	-	-
読出し / リセット *	4	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	FxAAAA _H	XXF0 _H	RA	RD	-	-	-	-
書込み プログラム	4	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	FxAAAA _H	XXA0 _H	PA (even)	PD (word)	-	-	-	-
チップ消去	6	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	FxAAAA _H	XX80 _H	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	FxAAAA _H	XX10 _H
セクタ消去	6	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	FxAAAA _H	XX80 _H	FxAAAA _H	XXAA _H	Fx5554 _H	XX55 _H	SA (even)	XX30 _H
セクタ消去一時停止		Address "FxXXXX _H " Data(xxB0 _H) 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 "FxXXXX _H " Data(xx30 _H) の入力で、セクタ消去一時停止後、消去再開											

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 (「23.2 2M/3M ビットフラッシュメモリのセクタ構成」を参照してください。)

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

*: 2 種類の読出し / リセットコマンドは、どちらもフラッシュメモリを読出しモードにリセットすることができます。

(注意事項)・表中のアドレス Fx は、FF, FE, FD, FC を意味します。それぞれの操作時には、アクセス対象バンクの値を指定してください。

- ・表中のアドレスは、CPU メモリマップ上の値です。アドレスおよびデータはすべて 16 進数表記しています。ただし、「X」は任意の値です。

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行うため、フラッシュメモリ内部の動作状態や動作が完了したことを知らせるハードウェアシーケンスフラグを持っています。

■ 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は、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)、セクタ消去タイムアウトフラグ (DQ3) の 4 ビットの出力で構成されます。各ビットの状態により、書込み / チップセクタ消去終了、消去コード書込みの有効性を確認します。

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	-	-	-

コマンドシーケンス (表 23.4-1 を参照) 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をリードアクセスすることにより、ハードウェアシーケンスフラグを参照できます。

自動アルゴリズム実行状態は、以下の方法で確認できます。

- ハードウェアシーケンスフラグの参照による確認
- フラッシュメモリコントロールレジスタ (FMCS) の RDY ビット参照による確認

実際にプログラムを作成する際には、これらの確認方法で自動アルゴリズム実行の完了を確認後に、次の命令を実行するようにしてください。次項より各ハードウェアシーケンスフラグについて説明します。

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の機能一覧

状態		DQ7	DQ6	DQ5	DQ3
正常動作時の 状態変化	書込み動作→書込み完了 (書込みアドレス指定時)	DQ7 → DATA:7	Toggle → DATA:6	0 → DATA:5	0 → DATA:3
	チップセクタ消去動作→消去完了	0 → 1	Toggle → Stop	0 → 1	1
	セクタ消去ウェイト→消去開始	0	Toggle	0	0 → 1
	消去動作→セクタ消去一時停止 (消去中のセクタ)	0 → 1	Toggle → 1	0	1 → 0
	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
	セクタ消去一時停止中 (消去中でないセクタ)	DATA:7	DATA:6	DATA:5	DATA:3
異常動作	書込み動作	DQ7	Toggle	1	0
	チップセクタ消去動作	0	Toggle	1	1

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、自動アルゴリズム実行が進行中また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ためのフラグです。

■ データポーリングフラグ (DQ7) の状態遷移

● 正常動作時の状態変化

動作状態	書込み動作 →完了	チップセクタ消去 →完了	セクタ消去ウェイト →開始	セクタ消去 →消去一時停止 消去中のセクタ	セクタ消去一時停止 →再開 消去中のセクタ	セクタ消去一時停止中 消去中でないセクタ
DQ7	$\overline{DQ7} \rightarrow$ DATA:7	0 → 1	0	0 → 1	1 → 0	DATA:7

● 異常動作時の状態変化

動作状態	書込み動作	チップセクタ 消去動作
DQ7	$\overline{DQ7}$	0

■ 書込み動作時

自動書込みアルゴリズム実行中に読出しアクセスすると、フラッシュメモリはアドレスの示す番地によらず、最後に書き込まれたデータのビット 7 の反転データを出力します。自動書込みアルゴリズム終了時に読出しアクセスすると、フラッシュメモリはアドレスの示す番地の読出し値のビット 7 を出力します。

■ チップ/セクタ消去動作時

チップ消去/セクタ消去アルゴリズム実行中に、セクタ消去時は現在消去しているセクタから、チップ消去時はアドレスの示す番地に関係なく、読出しアクセスするとフラッシュメモリは "0" を出力します。同様に、終了時には "1" を出力します。

■ セクタ消去一時停止時

セクタ消去一時停止時に読出しアクセスすると、フラッシュメモリはアドレスの示す番地が消去中のセクタであれば "1" を出力し、消去中のセクタでなければアドレスの示す番地の読出し値のビット 7 (DATA:7) を出力します。トグルビットフラグ (DQ6) とともに参照することで、現在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>

自動アルゴリズム起動時は、指定したアドレスへのリードアクセスは無視されます。データポーリングフラグ (DQ7) の終了確認後、データの読出しができます。このため、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直後に行うようにしてください。

23.5.2 トグルビットフラグ (DQ6)

トグルビットフラグ (DQ6) は、データポーリングフラグ (DQ7) と同様に、主に自動アルゴリズム実行が進行中、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ためのフラグです。

■ トグルビットフラグ (DQ6) の状態遷移

● 正常動作時の状態変化

動作状態	書込み動作 →完了	チップセクタ消去 →完了	セクタ消去ウェイト →開始	セクタ消去 →消去一時停止 消去中のセクタ	セクタ消去一時停止 →再開 消去中のセクタ	セクタ消去一時停止中 消去中でないセクタ
DQ6	Toggle → DATA:6	Toggle → Stop	Toggle	Toggle → 1	1 → Toggle	DATA:6

● 異常動作時の状態変化

動作状態	書込み動作	チップセクタ 消去動作
DQ6	Toggle	Toggle

■ 書込み / チップセクタ消去時

自動書込みアルゴリズムおよびチップセクタ消去アルゴリズム実行中に、連続した読出しアクセスを行うと、フラッシュメモリはアドレスの示す番地によらず、読出しごとに "1" と "0" を交互に出力するトグル状態を出力します。自動書込みアルゴリズムおよびチップ / セクタ消去アルゴリズム終了時に連続したリードアクセスを行うと、フラッシュメモリは bit6 のトグル動作を止め、アドレスの示す番地の読出し値の bit6(DATA:6) を出力します。

■ セクタ消去一時停止時

セクタ消去一時停止時にリードアクセスすると、フラッシュメモリはアドレスの示す番地が消去中のセクタに属する場合は、"1" を出力します。消去中のセクタに属しない場合は、アドレスの示す番地の読出し値の bit6(DATA:6) を出力します。

<参考>

書込みの際、書き込もうとしているセクタが書換え保護されている場合は、約 2 μ s のトグル動作をした後、データを書き換えることなくトグル動作を終わります。

消去の際、選択されたすべてのセクタが書換え保護されている場合は、トグルビットは約 100 μ s のトグル動作を行い、その後データを書き換えずに読出し / 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フラッシュメモリ内部に規定された時間 (内部パルス回数) を超えてしまったことを知らせるフラグです。

■ タイミングリミット超過フラグ (DQ5) の状態遷移

● 正常動作時の状態変化

動作状態	書込み動作 →完了	チップセクタ消去 →完了	セクタ消去ウェイト →開始	セクタ消去 →消去一時停止 消去中のセクタ	セクタ消去一時停止 →再開 消去中のセクタ	セクタ消去一時停止中 消去中でないセクタ
DQ5	0 → DATA:5	0 → 1	0	0	0	DATA:5

● 異常動作時の状態変化

動作状態	書込み動作	チップセクタ 消去動作
DQ5	1	1

■ 書込み / チップセクタ消去時

書込みまたはチップセクタ消去自動アルゴリズム起動後にリードアクセスすると、規定時間 (書込み / 消去に要する時間) 内であれば "0" を、規定時間を超えた場合は "1" を出力します。これは、自動アルゴリズムが実行中か終了状態にあるかに無関係なため、書込み / 消去が成功したか失敗したかの判定に使用できます。すなわち、このフラグが "1" を出力したとき、データポーリング機能またはトグルビット機能によって自動アルゴリズムがまだ実行中であれば、書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" を書き込もうとすると、フェイルが発生します。この場合、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しません。まれに "1" が書き込めたように正常終了する場合があります。したがって、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) はトグル動作を止めず、タイムリミットを超えると、タイミングリミット超過フラグ (DQ5) は "1" を出力します。この状態は、フラッシュメモリが不良ということではなく、正しく使用されなかったことを表しています。この状態が発生したときは、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フラグです。

■ セクタ消去タイマフラグ (DQ3) の状態遷移

● 正常動作時の状態変化

動作状態	書込み動作 →完了	チップセクタ消去 →完了	セクタ消去ウェイト →開始	セクタ消去 →消去一時停止 消去中のセクタ	セクタ消去一時停止 →再開 消去中のセクタ	セクタ消去一時停止中 消去中でないセクタ
DQ3	0 → DATA:3	1	0 → 1	1 → 0	0 → 1	DATA:3

● 異常動作時の状態変化

動作状態	書込み動作	チップセクタ 消去動作
DQ3	0	1

■ セクタ消去動作時

セクタ消去コマンド起動後に読出しアクセスすると、フラッシュメモリはコマンドを発行したセクタのアドレス信号の示す番地によらずに、セクタ消去ウェイト期間中であれば "0" を、セクタ消去ウェイト期間を超えてしまっている場合は "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って消去アルゴリズムが実行中を示している場合、このフラグが "1" であれば、内部で制御される消去が始まっています。続けてのセクタ消去コードの書込みまたは消去一時停止以外のコマンド発行は、消去が終了されるまで無視されます。

このフラグが "0" であれば、フラッシュメモリは、追加のセクタ消去コードの書込みを受け付けます。このことを確認するために、次のセクタ消去コードの書込みに先立って、このフラグの状態をチェックしておくことを推奨します。2 回目の状態チェックでフラグが "1" であった場合は、追加セクタの消去コードは受け付けられていない可能性があります。

■ セクタ消去一時停止時

セクタ消去一時停止中に読出しアクセスすると、フラッシュメモリはアドレスの示す番地が、消去中のセクタに属する場合は "1" を出力します。消去中のセクタに属さない場合は、アドレスの示す番地の読出し値の bit3(DATA:3) を出力します。

23.6 フラッシュメモリの書込み / 消去

自動アルゴリズムの起動によるフラッシュメモリの書込み / 消去の手順について説明します。

■ フラッシュメモリの書込み / 消去

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、消去再開のいずれかのコマンドシーケンス（表 23.4-1 を参照）を CPU からフラッシュメモリに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。

以下の項で、フラッシュメモリの書込み / 消去に関する下記項目について説明します。

- 読出し / リセット状態にする方法
- データを書き込む方法
- 全データを消去する方法（チップ全消去）
- 任意のデータを消去する方法（セクタ消去）
- セクタ消去を一時停止する方法
- セクタ消去を再開する方法

23.6.1 フラッシュメモリを讀出し / リセット状態にする方法

讀出し / リセットコマンドを発行し、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方法

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 (表 23.4-1 を参照) の讀出し / リセットコマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行できます。

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 とおりのコマンドシーケンスがありますが、これらの本質的な違いはありません。

讀出し / リセット状態は、フラッシュメモリの初期状態であり、電源投入時、およびコマンドの正常終了時は常に讀出し / リセット状態になります。讀出し / リセット状態は、ほかのコマンドの入力待ち状態です。

讀出し / リセット状態では、通常の讀出しアクセスで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可能です。通常讀出しでのデータ讀出しに、このコマンドは必要ありません、何らかの理由でコマンドが正常に終了しなかったときなど、自動アルゴリズムを初期化する場合に、主にこのコマンドを使用します。

23.6.2 フラッシュメモリにデータを書き込む方法

書込みコマンドを発行し、フラッシュメモリにデータを書き込む手順について説明します。

■ フラッシュメモリへのデータ書込み

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には、コマンドシーケンス表(表 23.4-1 を参照)の書込みコマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行できます。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時点で、自動アルゴリズムが起動され自動書込みが開始されます。

● アドレス指定方法

書込みデータサイクルの中で指定する書込みアドレスは、偶数アドレスのみ指定が可能です。奇数アドレスを指定すると正しく書き込むことができません。つまり、偶数アドレスへのワードデータ単位での書込みが必要となります。

書込みはどのようなアドレスの順番でも、また、セクタの境界を超えても可能ですが、1 回の書込みコマンドによって書き込まれるデータは 1 ワードのみです。

● データ書込み上の注意

フラッシュメモリ上のデータを "0" から "1" に書き込むと、データポーリングフラグ(DQ7)またはトグルビットフラグ(DQ6)が終了状態にならないため、フラッシュメモリ素子が不良と判定され以下の状態に陥るため、書込みによってフラッシュメモリ上のデータを "0" から "1" へ戻すことは行わないでください。

- 書込み規定時間を超えてタイムリミット超過フラグ(DQ5)がエラー状態となる。
- 見かけ上フラッシュメモリ上のデータに "1" が書き込まれたように見える状態となる(読出し/リセット状態でデータを読み出すと、データは "0" のままです。データを "0" から "1" に戻す場合は、消去を行ってください)。

自動書込み実行中は、すべ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ると、書き込んでいるアドレスのデータは保証されません。

■ フラッシュメモリの書込み手順

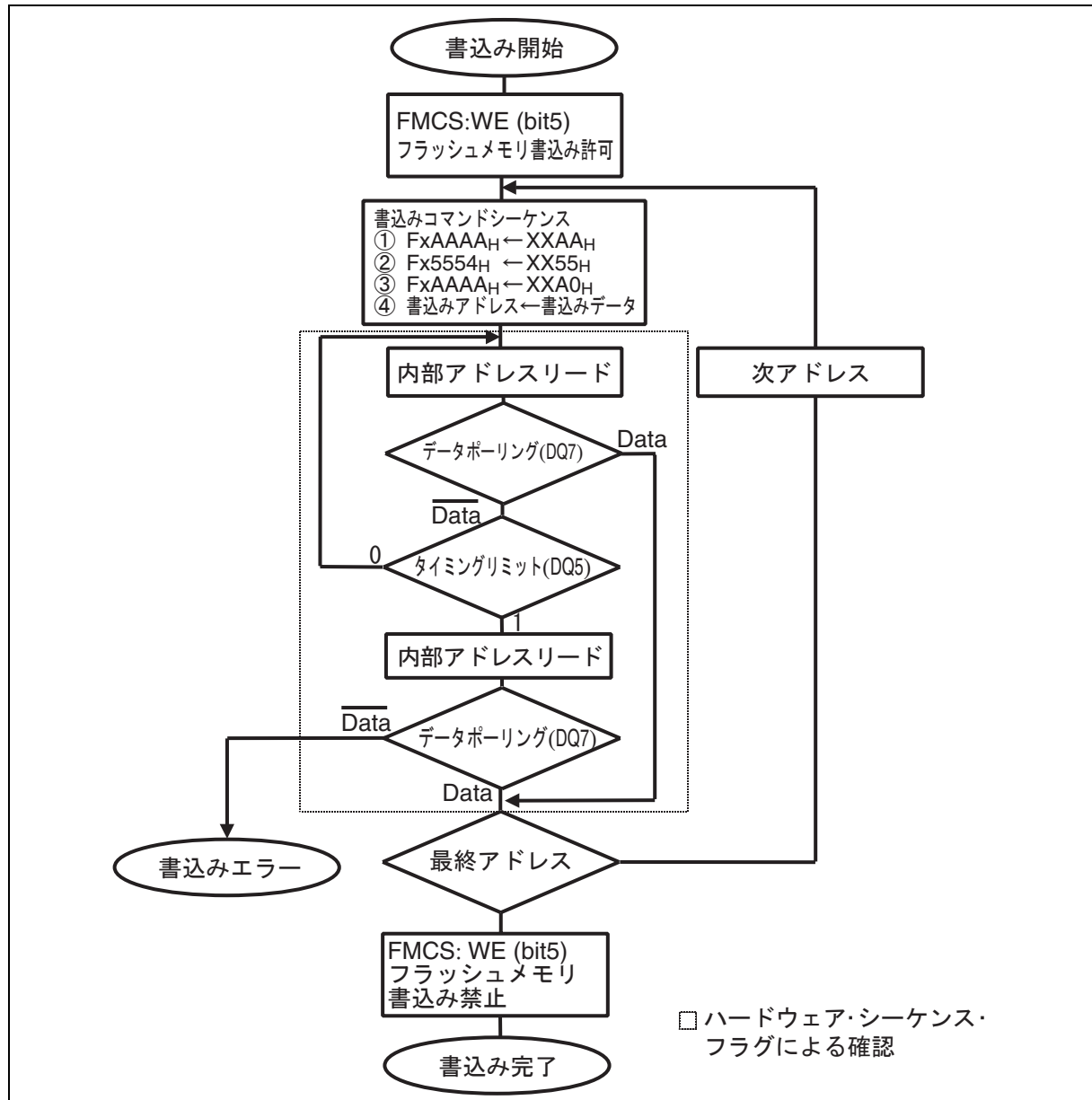
図 23.6-1 に、フラッシュメモリ書込み手順の例を示します。ハードウェアシーケンスフラグ (「23.5 自動アルゴリズム実行状態の確認」を参照) を用いることでフラッシュメモリ内部の自動アルゴリズムの状態判定ができます。ここでは、書込み終了の確認に、データポーリングフラグ (DQ7) を用いています。

フラグチェックのために読み込むデータは、最後に書込みを行ったアドレスからの読み込みとなります。

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) と同時に変わるので、たとえタイミングリミット超過フラグ (DQ5) が "1" であっても、データポーリングフラグビット (DQ7) を再チェックする必要があります。

トグルビットフラグ (DQ6) も同様に、タイミングリミット超過フラグビット (DQ5) が "1" に変わるのと同時にトグル動作を止めるので、トグルビットフラグ (DQ6) を再チェックする必要があります。

図 23.6-1 フラッシュメモリ書込み手順の例



23.6.3 フラッシュメモリの全データを消去する方法 (チップ消去)

チップ消去コマンドを発行し、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全データを消去する方法 (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には、コマンドシーケンス表(表 23.4-1 を参照)のチップ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることで行えます。チップ消去コマンドは、6 回のバス動作で行われます。6 サイクル目の書込みが完了した時点でチップ消去動作が開始します。

チップ消去では、消去前にユーザがフラッシュメモリに書込みを行う必要はありません。自動消去アルゴリズム実行中には、フラッシュメモリは自動的にすべてのセルを消去する前に "0" を書き込んで検証します。

23.6.4 フラッシュメモリの任意のデータを消去する方法 (セクタ消去)

セクタ消去コマンドを発行し、フラッシュメモリの任意のセクタを消去する手順について説明します。セクタごとの消去が可能であり、複数のセクタを同時に指定することもできます。

■ フラッシュメモリの任意のデータを消去する方法 (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(表 23.4-1 を参照)のセクタ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることによって実行できます。

● 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 サイクル目に目的のセクタ内のアクセス可能な任意の偶数の 1 アドレスへ、セクタ消去コード(30_H)を書き込みます。これによって 50 μ 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、上記の処理に引き続き、消去する目的のセクタ内のアドレスに消去コード(30_H)を書き込んでください。

● 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き込みから 50 μ 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。つまり、複数のセクタを同時に消去する場合は、次の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)をそれぞれ 50 μ s 以内に入力する必要があります。引き続くセクタ消去コードの書き込みが有効かどうかは、セクタ消去タイマ(ハードウェアシーケンスフラグ:DQ3)によって調べることができます。なおこのとき、セクタ消去タイマを読み出すアドレスは、消去しようとしているセクタを指すようにしてください。

■ セクタ消去の手順

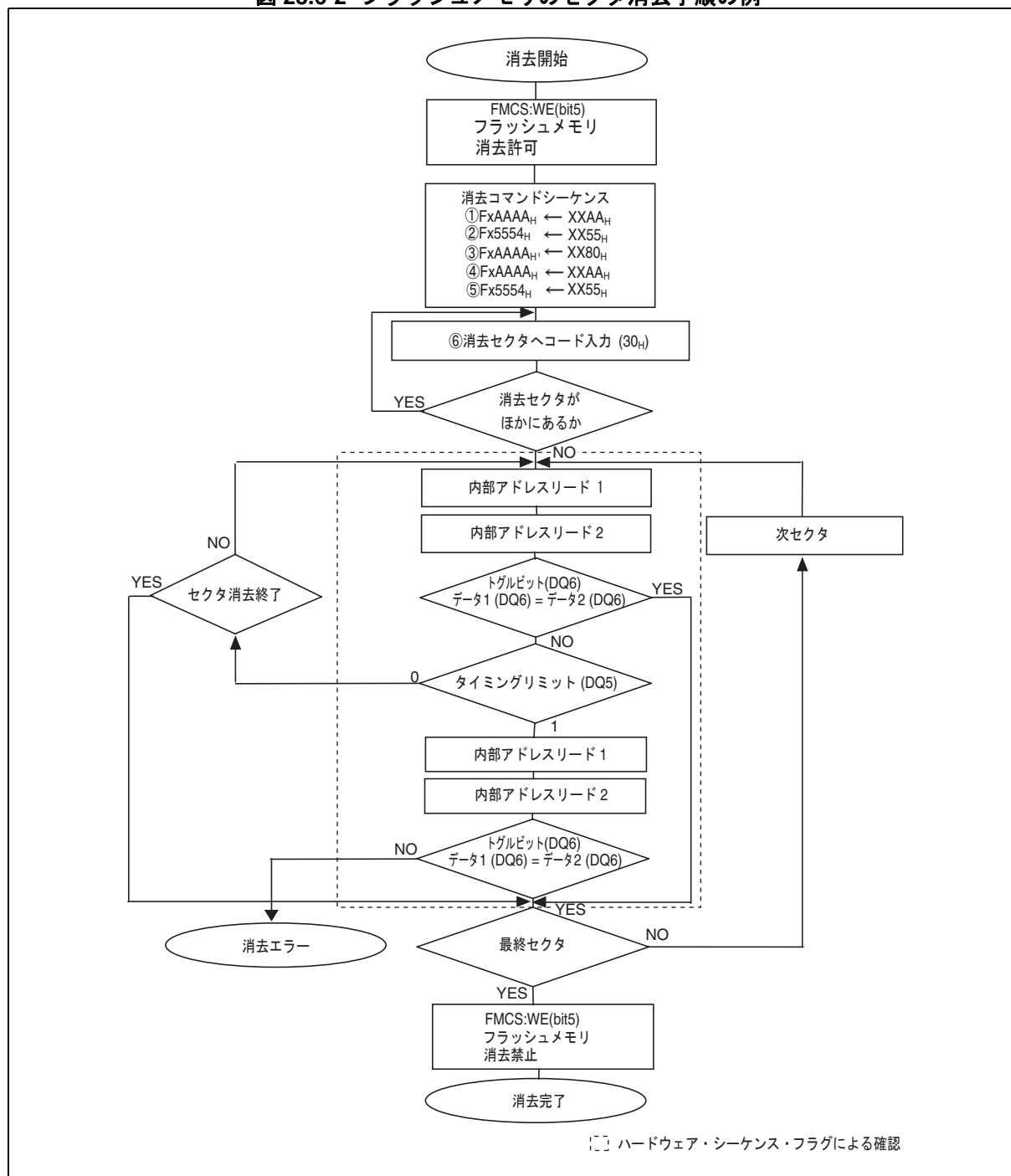
ハードウェアシーケンスフラグ(「23.5 自動アルゴリズム実行状態の確認」を参照)を用いることによってフラッシュメモリ内部の自動アルゴリズムの状態判定ができます。図 23.6-2 に、フラッシュメモリのセクタ消去手順の例を示します。ここでは、消去終了の確認にトグルビットフラグ(DQ6)を用いています。

フラグチェックのために読み込むデータは、消去しようとしているセクタからの読み込みとなりますので、注意が必要です。

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"に変わるのと同時にトグル動作を止めるので、たとえタイミングリミット超過フラグ(DQ5)が"1"であっても、トグルビットフラグ(DQ6)を再チェックする必要があります(図 23.6-2 の[...]内の処理)。

データポーリングフラグ(DQ7)も同様に、タイミングリミット超過フラグ(DQ5)と同時に変わるので、データポーリングフラグ(DQ7)を再チェックする必要があります。

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方法

セクタ消去一時停止コマンドを発行し、フラッシュメモリのセクタ消去を一時停止する手順について説明します。一時停止中に、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。

■ フラッシュメモリのセクタ消去を一時停止する方法

フラッシュメモリのセクタ消去を一時停止するには、コマンドシーケンス表(表 23.4-1 を参照)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内に送ってください。

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のデータ読出しを可能にします。セクタ消去一時停止状態では、読出しのみが可能で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、チップ消去中や書込み動作中は無視されます。

セクタ消去の一時停止は、消去一時停止コード(B0_H)の書込みを行うことで実施されます。このとき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。消去一時停止中の再度の消去一時停止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が入力されると、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動作中に消去一時停止コマンドが入力されると、最大 20 μ s の時間の後、消去一時停止状態に入ります。セクタ消去一時停止コマンドでは、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 μ 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方法

セクタ消去再開コマンドを発行し、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方法

一時停止したセクタ消去を再開させるには、コマンドシーケンス表 (表 23.4-1 を参照) のセクタ消去再開コマンドを、フラッシュメモリ内に送ることで実行できます。

セクタ消去再開コマンドは、セクタ消去一時停止コマンドによ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。このコマンドは消去再開コード (30_H) の書込みを行うことで実施されますが、このときのアドレスはフラッシュメモリ領域内の任意のアドレスを示すようにします。

なお、セクタ消去中のセクタ消去再開コマンドの発行は無視されます。

23.7 フラッシュセキュリティ機能

フラッシュセキュリティ機能によりフラッシュメモリの内容を保護することができます。

■ 概要

フラッシュメモリのセキュリティビットに保護コード 01_H を書き込むと、フラッシュメモリへのアクセスを制限できます。フラッシュメモリを一度保護すると、チップ消去を行うまで保護された状態を解除することはできません。保護された状態を解除しない限り、外部端子からフラッシュメモリのデータを読み出し/書き込みすることはできません。

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。

セキュリティビットのアドレスは、搭載されたフラッシュメモリのサイズによって異なります。表 23.7-1 に、セキュリティビットのアドレスを示します。

表 23.7-1 フラッシュセキュリティビットのアドレス

	フラッシュメモリサイズ	セキュリティビットのアドレス
MB90F488B	2M ビットフラッシュメモリ搭載	FC0001 _H
MB90F489B	3M ビットフラッシュメモリ搭載	F90001 _H

■ セキュリティのかけ方

セキュリティビットへ保護コード 01_H を書き込みます。外部リセットまたは電源再投入後セキュリティがかかります。

■ セキュリティの解除

チップ消去の実行により、セキュリティは解除されます。

■ セキュリティ許可中の動作

リード：外部端子から無効データが読み出されます。

ライト：書き込めません。

■ その他

- 汎用パラレルライタの設定は、使用するパラレルライタの仕様に従ってください。
- 保護コードは、フラッシュプログラミングの最後に関き込むことを推奨します。これは、プログラミング中、不意に保護されてしまうのを防ぐためです。

<注意事項>

セキュリティビットはフラッシュメモリの領域内に配置されます。セキュリティビットに保護コード 01_H を書き込むとセキュリティがかかりますので、セキュリティ機能を使用しない場合は、このアドレスには 01_H を書き込まないでください。

セキュリティビットのアドレスは、上記の表 23.7-1 を参照してください。

第24章

MB90F481B/F482B/F488B/F489B

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 AF220/
AF210/AF120/AF110 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 MB90F481B, MB90F482B, MB90F488B, MB90F489B
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 (ユーザ電源使用時)

24.3 フラッシュマイコンプログラマとの接続例 (ユーザ電源
使用時)

24.1 MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成

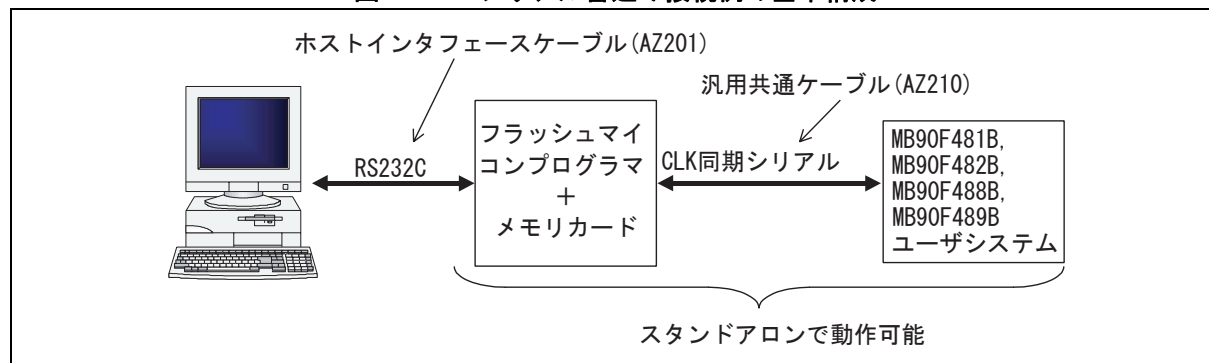
MB90F481B, MB90F482B, MB90F488B, MB90F489B ではフラッシュ ROM のシリアルオンボード書込み (富士通標準) をサポートしています。その仕様について以下に解説します。

■ MB90F481B, MB90F482B, MB90F488B, MB90F489B 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。シングルチップモードで動作するプログラム、あるいは内 ROM 外バスモードで動作するプログラムのどちらかを選択して書き込むことができます。

図 24.1-1 に、シリアル書込み接続例の基本構成を示します。

図 24.1-1 シリアル書込み接続例の基本構成



フラッシュマイコンプログラマ (AF220/AF210/AF120/AF110) の機能、操作方法および接続用汎用共通ケーブル (AZ210)、コネクタについては、横河デジタルコンピュータ株式会社殿にお問い合わせください。

■ 富士通標準シリアルオンボード書込みに使用する端子

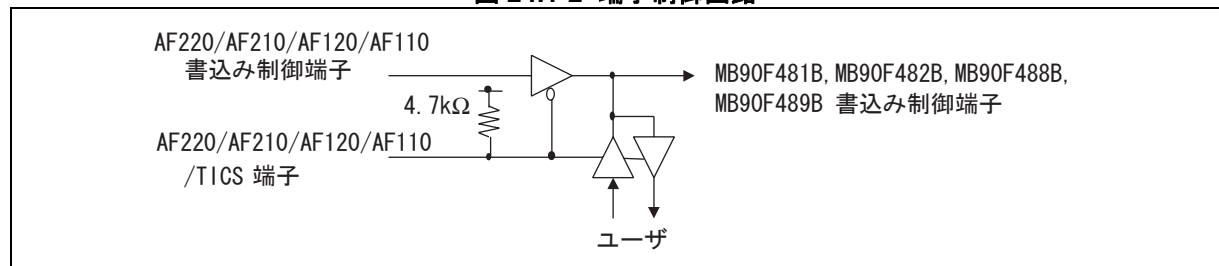
表 24.1-1 に、富士通標準シリアルオンボード書込みに使用する端子の機能を示します。

表 24.1-1 使用する端子の機能

端子	機能	補足説明
MD2, MD1, MD0	モード端子	MD2=1, MD1=1, MD0=0 に設定することで、シリアル書込みモードになります。
X0, X1	発振用端子	シリアル書込みモード時に、CPU 内部動作クロックは PLL クロック 1 通倍となりますので、内部動作クロック周波数は、発振クロック周波数と同じになります。したがって、シリアル書込みを行う場合、高速発振入力端子に入力可能な周波数は 4.5 MHz ～ 25 MHz となります。
P80, P81	書込みプログラム起動端子	P80 に "L" レベル、P81 に "H" レベルを入力してください。
$\overline{\text{RST}}$	リセット端子	-
SIN0	シリアルデータ入力端子	UART0 を CLK 同期モードとして使用します。
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	
V _{CC}	電源電圧供給端子	書込み電圧 (V _{CC} =3.3V ± 5%)
V _{SS}	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。

P80, P81, SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には、図 24.1-2 に示す制御回路が必要です。フラッシュマイコンプログラムの /TICS 信号の "L" 出力により、シリアル書込み中はユーザ回路を切り離すことができます。

図 24.1-2 端子制御回路



■ 発振クロック周波数とシリアルクロック入力周波数について

MB90F481B, MB90F482B, MB90F488B, MB90F489B の入力可能なシリアルクロック周波数は、以下の計算式により求められます。

したがって、フラッシュマイコンプログラマの設定により、使用する発振クロック周波数に対応したシリアルクロック入力周波数に、変更してください。

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 に、入力可能なシリアルクロック周波数を示します。

表 24.1-2 入力可能なシリアルクロック周波数

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
8MHz 時	1MHz	850kHz	500kHz
16MHz 時	2MHz	1.25MHz	500kHz

■ フラッシュマイコンプログラムのシステム構成(横河ディジタルコンピュータ株式会社製)

表 24.1-3 に、フラッシュマイコンプログラムのシステム構成を示します。

表 24.1-3 フラッシュマイコンプログラムのシステム構成

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 F ² MC-16LX フラッシュマイコン用コントロールモジュール
AZ290		リモートコントローラ
/P2		2M バイト PC Card(Optional) フラッシュメモリ容量 ～ 128K バイト対応
/P4		4M バイト PC Card(Optional) フラッシュメモリ容量 ～ 512K バイト対応

問い合わせ先：横河ディジタルコンピュータ株式会社

電話：042-333-6224

■ シリアル書込み接続例

以下に、シリアル書込み接続例を示しますので、参照してください。

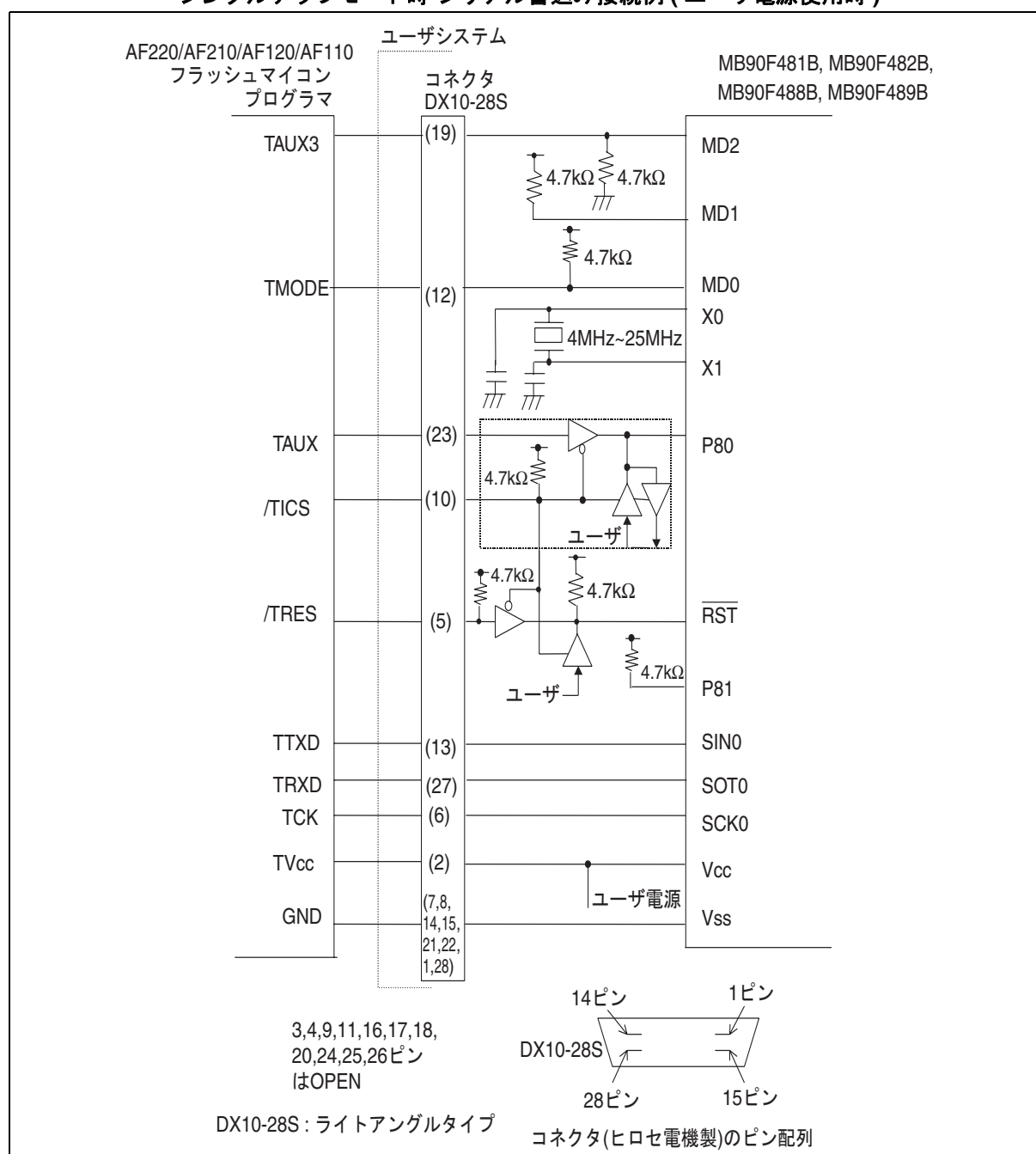
- シングルチップモード時の接続例 (ユーザ電源使用時)
- フラッシュマイコンプログラムの接続例 (ユーザ電源使用時)

24.2 シングルチップモード時の接続例（ユーザ電源使用時）

ユーザシステムにて、シングルチップモードに設定されているモード端子 MD2, MD0 には、AF220/AF210/AF120/AF110 の TAUX3, TMODE より MD2=1, MD0=0 が入力され、シリアル書込みモードになります (シリアル書込みモード :MD2, MD1, MD0=110)。

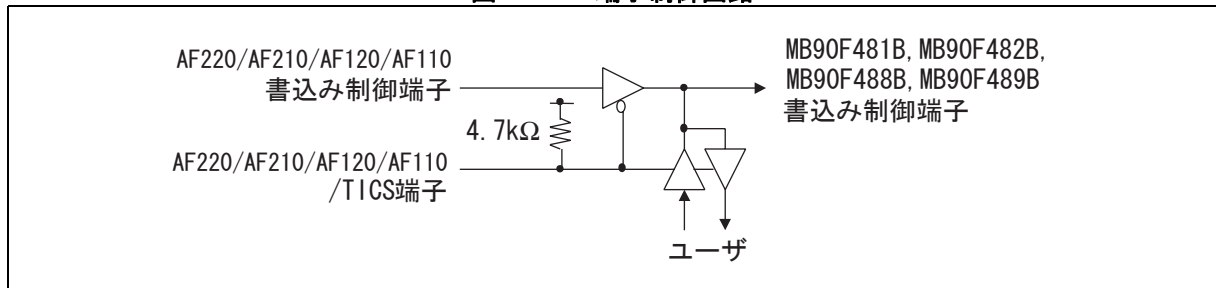
■ シングルチップモード時の接続例（ユーザ電源使用時）

**図 24.2-1 MB90F481B, MB90F482B, MB90F488B, MB90F489B
シングルチップモード時 シリアル書き込み接続例 (ユーザ電源使用時)**



SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, P80 と同様, 図 24.2-2 に示す制御回路が必要です。フラッシュマイコンプログラムの /TICS 信号の "L" 出力により, シリアル書き込み中はユーザ回路を切り離すことができます。

図 24.2-2 端子制御回路



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。

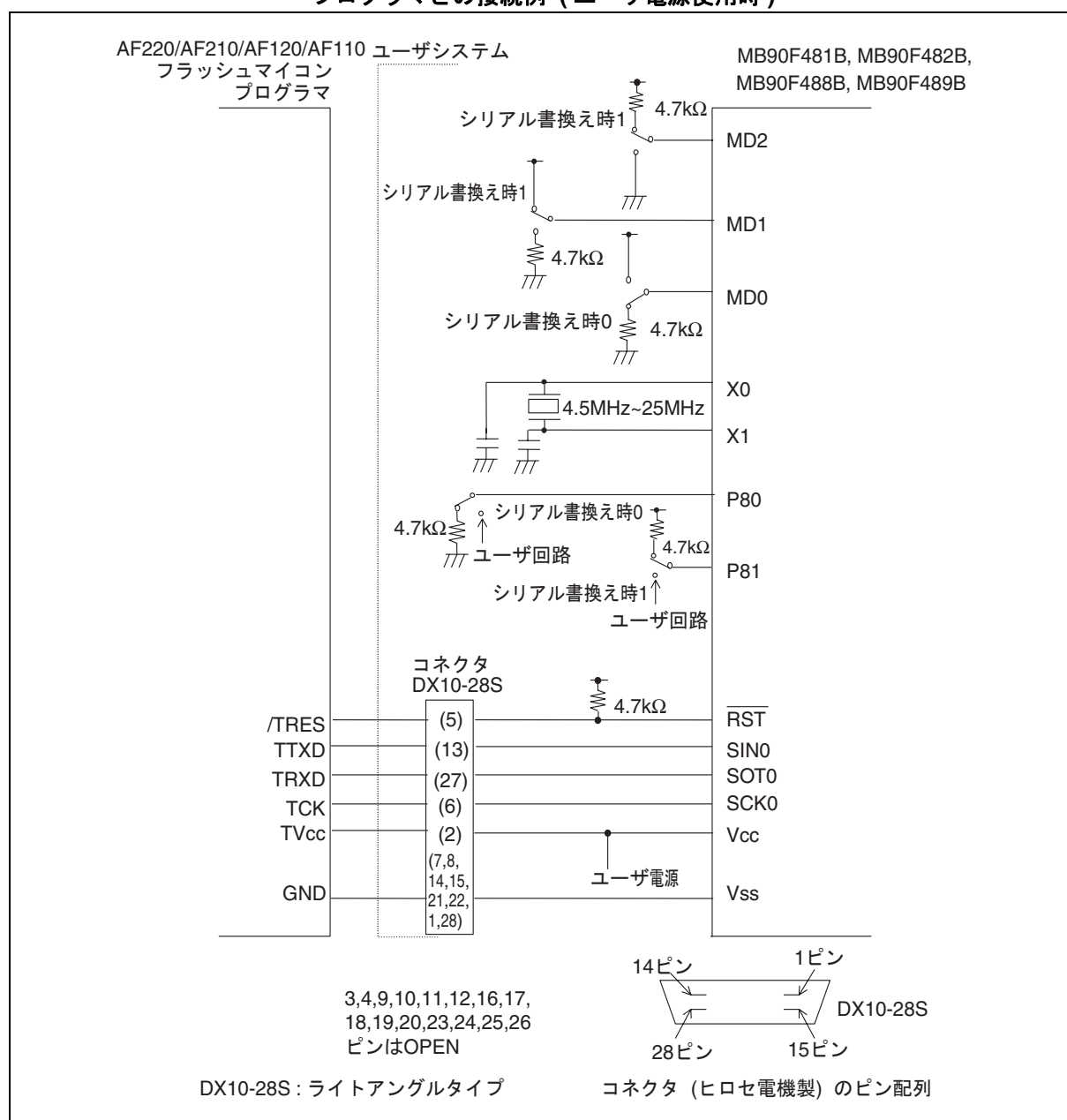
24.3 フラッシュマイコンプログラマとの接続例 (ユーザ電源使用時)

シリアル書き込み時に , 各端子 (MD2, MD0, P80) を図 24.3-1 に示すように設定した場合は , MD2, MD0, P80 とフラッシュマイコンプログラマとの接続は必要ありません。

■ フラッシュマイコンプログラマとの接続例 (ユーザ電源使用時)

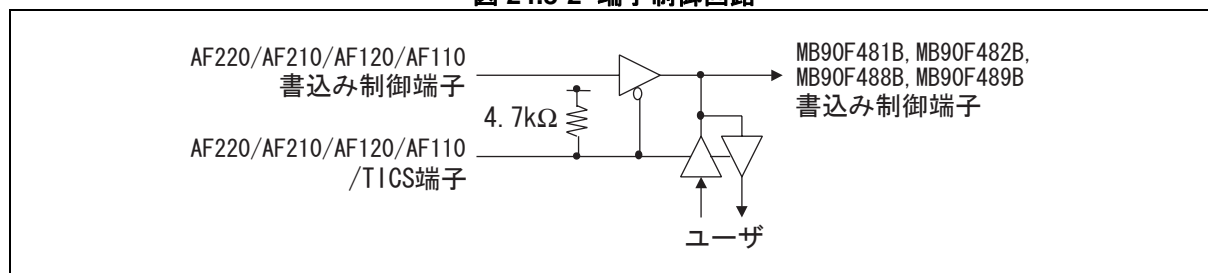
図 24.3-1 に , フラッシュマイコンプログラマとの接続例を示します。

図 24.3-1 MB90F481B, MB90F482B, MB90F488B, MB90F489B フラッシュマイコン
プログラマとの接続例 (ユーザ電源使用時)



SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には, 図 24.3-2 に示す制御回路が必要です。フラッシュマイコンプログラムの /TICS 信号の "L" 出力により, シリアル書き込み中はユーザ回路を切り離すことができます。

図 24.3-2 端子制御回路



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。

第 25 章

PWC タイマ (MB90485 シリーズのみ)

PWC タイマの概要，構成，レジスタの構成と機能，割込み動作および使用上の注意について説明します。

25.1 PWC タイマの概要

25.2 PWC タイマの構成

25.3 PWC タイマのレジスタの構成と機能

25.4 PWC タイマの割込み

25.5 PWC タイマの動作

25.6 PWC タイマの使用上の注意

25.1 PWC タイマの概要

PWC タイマは、入力信号のパルス幅測定機能を持つ、16 ビットの高機能アップカウントタイマです。

PWC:Pulse Width Count(パルス幅測定)

■ PWC タイマの機能

ハードウェアとして、16 ビットアップカウントタイマ 1 個、入力パルス分周器 & 分周比制御レジスタ 1 個、測定入力端子 1 本、16 ビット制御レジスタ 1 個を 1 チャンネルとして、合計 3 チャンネルを有し、これらによって以下の機能を実現します。

● タイマ機能

- 設定された時間ごとに割込み要求を発生することができます。
- 基準となる内部クロックは、3 種類 (マシクロックの 4 分周 / 16 分周 / 32 分周) の中から選択できます。

● パルス幅測定機能

- 外部からのパルス入力の任意イベント間の時間を測定します。
- 基準となる内部クロックは、3 種類 (マシクロックの 4 分周 / 16 分周 / 32 分周) の中から選択可能です。
- 各種測定モードに対応できます。
 - "H" パルス幅 (↑ ~ ↓) / "L" パルス幅 (↑ ~ ↓)
 - 立上り周期 (↑ ~ ↑) / 立下り周期 (↓ ~ ↓)
 - エッジ間測定 (↑ or ↓ ~ ↓ or ↑)
- 8 ビット入力分周器で、入力パルスを 2ⁿ 分周 (n=1, 2, 3, 4) して分周測定を行うことができます。
- 測定終了時に割込みを発生させることができます。
- 1 回のみの測定か、連続測定かを選択できます。

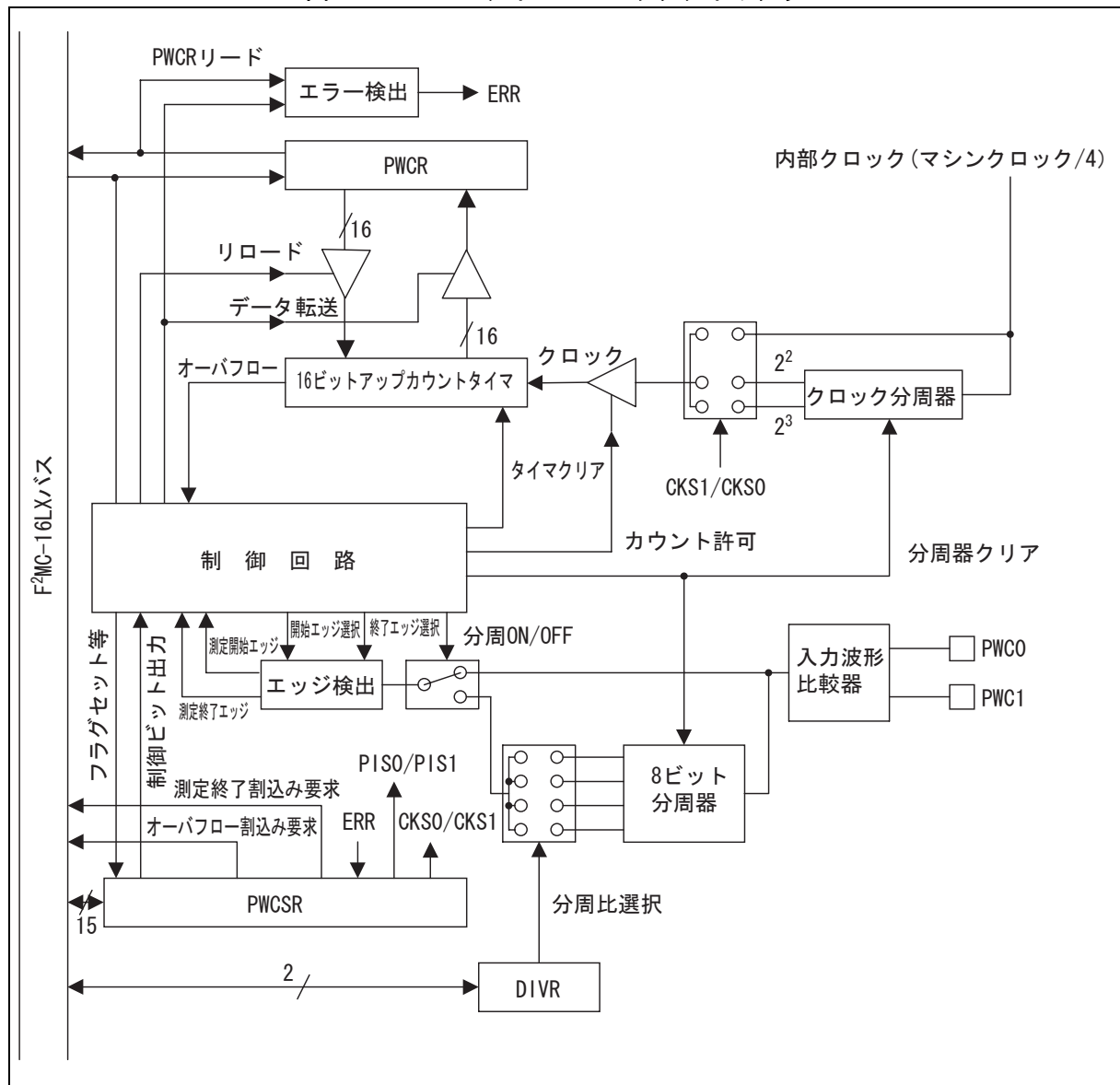
25.2 PWC タイマの構成

PWC タイマは、PWC コントロール/ステータスレジスタ、PWC データバッファ、分周比制御レジスタから構成されています。

■ PWC タイマのブロックダイアグラム

図 25.2-1 に、PWC タイマのブロックダイアグラムを示します。

図 25.2-1 PWC タイマのブロックダイアグラム



■ PWC タイマに関する端子

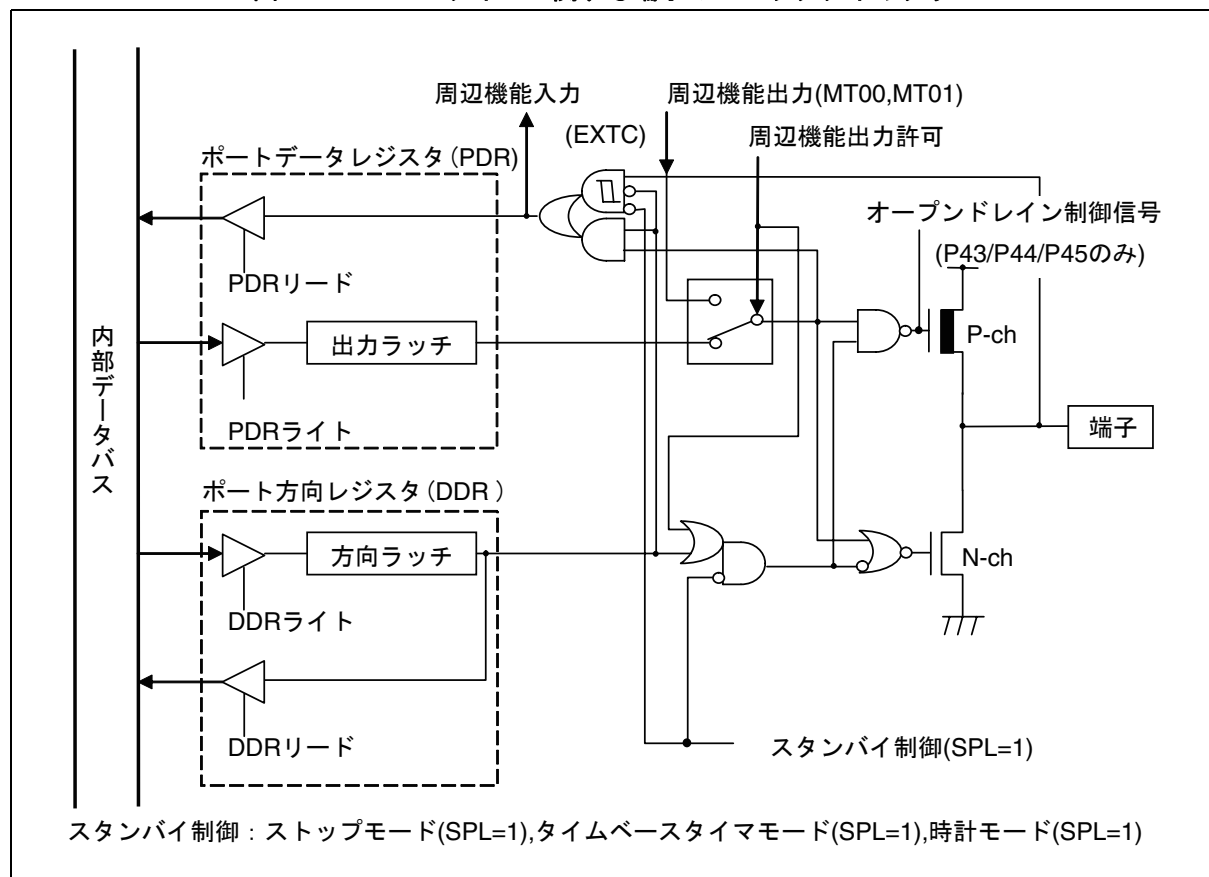
PWC タイマに関する端子は、PWC0/PWC1/PWC2 の 3 チャンネルがあり、PWC 使用時には、入力ポートとして機能します。PWC0/PWC1/PWC2 端子は、汎用入出力ポート (P36/PWC0, P37/PWC1, P75/PWC2) と入力端子として機能を兼用しています。

● PWC0/PWC1/PWC2 端子として使用する場合の設定

PWC タイマで PWC0/PWC1/PWC2 を入力として使用する場合、P36/PWC0, P37/PWC1, P75/PWC2 端子は、ポート方向レジスタにより入力ポート (DDR3 bit14, 15 → "0" DDR7 bit13 → "0") に設定してください。

■ PWC タイマに関する端子のブロックダイアグラム

図 25.2-2 PWC タイマに関する端子のブロックダイアグラム



25.3 PWC タイマのレジスタの構成と機能

PWC タイマに関連するレジスタの構成と機能を説明します。

■ PWC タイマのレジスタ一覧

図 25.3-1 に、PWC タイマのレジスタ一覧を示します。

図 25.3-1 PWC タイマのレジスタ一覧

		15	8 7		0		PWC SR0~2		(R/W)	
		PWC R0~2							(R/W)	
		DIV R0~2							(R/W)	
ch. 0	000077 _H	15	14	13	12	11	10	9	8	PWC SR0~PWC SR2
ch. 1	00007B _H	STRT	STOP	EDIR	EDIE	OVIR	OVIE	ERR	予約	PWCコントロール/ステータスレジスタ
ch. 2	00007F _H	(R/W)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	(R)	(—)	初期値00000000 _B
ch. 0	000076 _H	7	6	5	4	3	2	1	0	PWC SR0~PWC SR2
ch. 1	00007A _H	CKS1	CKS0	PIS1	PIS0	S/C	MOD2	MOD1	MOD0	PWCコントロール/ステータスレジスタ
ch. 2	00007E _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 0	000079 _H	15	14	13	12	11	10	9	8	PWC R0~PWC R2
ch. 1	00007D _H	D15	D14	D13	D12	D11	D10	D9	D8	PWCデータバッファレジスタ
ch. 2	000081 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 0	000078 _H	7	6	5	4	3	2	1	0	PWC R0~PWC R2
ch. 1	00007C _H	D7	D6	D5	D4	D3	D2	D1	D0	PWCデータバッファレジスタ
ch. 2	000080 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	初期値00000000 _B
ch. 0	000082 _H	7	6	5	4	3	2	1	0	DIV R0~DIV R2
ch. 1	000084 _H	—	—	—	—	—	—	DIV1	DIV0	分周比制御レジスタ
ch. 2	000086 _H	(—)	(—)	(—)	(—)	(—)	(—)	(R/W)	(R/W)	初期値-----00 _B

25.3.1 PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2)

PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) の構成と機能について説明します。

■ PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2)

図 25.3-2 に、PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) のビット構成を示します。

図 25.3-2 PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) のビット構成

ch. 0	000077 _H	15	14	13	12	11	10	9	8	ビットNo.
ch. 1	00007B _H	STRT	STOP	EDIR	EDIE	OVIR	OVIE	ERR	予約	
ch. 2	00007F _H	(R/W)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	(R)	(-)	リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(X)	初期値

ch. 0	000076 _H	7	6	5	4	3	2	1	0	PWCSR
ch. 1	00007A _H	CKS1	CKS0	PIS1	PIS0	S/C	MOD2	MOD1	MOD0	
ch. 2	00007E _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	初期値

以下に、PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) の各ビットの機能を説明します。

[bit15, bit14] STRT, STOP(タイマスタートビット, タイマストップビット)

これらのビットは、16 ビットアップカウントタイマの起動 / 再起動 / 停止を制御します。読出し時には、タイマの動作状態を表示します。

下表に、STRT および STOP ビットの機能を示します。

表 25.3-1 書き込み時機能 (16 ビットアップカウントタイマの動作制御)

STRT	STOP	動作制御機能
0	0	機能なし / 動作に影響なし
0	1	タイマ起動 / 再起動 (カウント許可) *
1	0	タイマ動作強制停止 (カウント禁止) *
1	1	機能なし / 動作に影響なし

*: クリアビット命令使用可能

表 25.3-2 読出し時機能 (16 ビットアップカウントタイマの動作状態表示)

STRT	STOP	動作制御機能
0	0	タイマ停止中 (起動されていないかまたは測定終了) (初期値)
1	1	タイマカウント動作中 (測定中)

- リセット時, "00_B" に初期化されます。
- 読出し / 書込みが可能です。ただし, 書込み時と読出し時では, 表 25.3-1, 表 25.3-2 に示すように機能が異なります。
- リードモディファイライト系命令における読出し値は, ビット値にかかわらず "11_B" です。
- タイマの起動 / 停止のための STRT/STOP ビットへの書込みは, それぞれのビットに対するビット命令 (ビットクリア) を用いることが可能ですが, 動作状態の読出しには, ビット命令は使用できませんので注意してください (読み出すと必ず "1" が読み出せます)。

[bit13] EDIR(測定終了割込み要求フラグ)

このビットは, パルス幅測定時, 測定が終了したことを示すフラグです。測定終了割込み要因が許可されているとき (bit12:EDIE=1) に, このビットがセットされると, 測定終了割込み要求が発生します。

EDIR	要因
セット要因	パルス幅測定が終了するとセット (PWCR に測定結果が収納されている)
クリア要因	PWCR(測定結果) を読み出すことによってクリア

- リセット時 "0" に初期化されます。
- 読出しのみ可能です。
- 書き込んでもビット値は変化しません。

[bit12] EDIE(測定終了割込み許可)

このビットは, パルス幅測定時の, 測定終了割込み要求を下表のように制御します。

EDIE	動作
0	測定終了割込み要求出力禁止 (EDIR がセットされていても割込み発生せず) [初期値]
1	測定終了割込み要求出力許可 (EDIR がセットされると割込みが発生する)

- リセット時 "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit11] OVIR(タイマオーバフロー割込み要求フラグ)

このビットは, 16 ビットアップカウントタイマが FFFF_H から 0000_H に変化しオーバフローが発生したことを示すフラグです。タイマオーバフロー割込み要求が許可されているとき (bit10: OVIE=1 のとき) にこのビットがセットされると, タイマオーバフロー割込み要求が発生します。

OVIR	要因
セット要因	タイマオーバフローが発生するとセット (FFFF _H から 0000 _H)
クリア要因	"0" 書込み, または μ DMAC によりクリア

- リセット時 "0" に初期化されます。
- 読出し / 書込みが可能です。書込みは "0" のみ可能です。"1" を書き込んでもビット値は変化しません。
- リードモディファイライト系命令における読出しは, ビット値にかかわらず "1" です。

[bit10] OVIE(タイマオーバフロー割込み要求許可)

このビットは、パルス幅測定時の測定終了割込み要求を、下表のように制御します。

OVIE	動作
0	オーバフロー割込み要求出力禁止 (OVIR がセットされていても割込み発生しません。) [初期値]
1	オーバフロー割込み要求出力許可 (OVIR がセットされると割込みが発生します。)

- リセット時 "0" に初期化されます。
- 読出し / 書込みが可能です。

[bit9] ERR(エラーフラグ)

パルス幅測定時の連続測定モード時において、PWCR 内の測定結果を読み出さないうちに、次の測定が終了してしまったことを示すフラグです。この際、PWCR の値は新しい測定結果に更新され、1 つ前の測定結果は消失します。測定は、本ビット値に関係なく続行されます。

ERR	要因
セット要因	読み出していない測定結果が次の結果により消失するとセット
クリア要因	PWCR(測定結果) を読み出すことによりクリア

- リセット時 "0" に初期化されます。
- 読出しのみ可能です。書き込んでもビット値は変化しません。

[bit8] 予約ビット

このビットは、予約ビットです。

[bit7, bit6] CKS1, CKS0(クロック選択)

これらのビットは、内部カウントクロックを選択します。表 25.3-3 に示す 3 種類の、いずれかを選択できます。

表 25.3-3 16 ビットアップカウントタイマのカウントクロック

CKS1	CKS0	カウントクロック選択
0	0	マシニングロックの 4 分周クロック (マシニングロック =16MHz 時 0.25μs) [初期値]
0	1	マシニングロックの 16 分周クロック (マシニングロック =16MHz 時 1.00μs)
1	0	マシニングロックの 32 分周クロック (マシニングロック =16MHz 時 2.00μs)
1	1	設定禁止 (未定義)

- リセット時, "00_B" に初期化されます。
- 読出し / 書込みが可能です。ただし, "11_B" の設定は禁止です。

<注意事項>

PWC タイマ起動後の書換えは 禁止です。必ず起動前か停止後に書き込んでください。

[bit5, bit4] PIS1, PIS0(パルス幅測定入力端子選択)

これらのビットは、パルス幅測定入力端子を選択します。

表 25.3-4 パルス幅測定入力端子の選択

PIS1	PIS0	入力クロック選択
0	0	(PWC0 端子を選択) [初期値]
0	1	2 入力比較選択 (立上りエッジ比較)
1	0	2 入力比較選択 (立下りエッジ比較)
1	1	設定禁止 (未定義)

- リセット時 "00_B" に初期化されます。
- 読出し / 書込みが可能です。ただし、"11_B" の設定は禁止です。
- このビットは、PWC0 のみに有効です (PWC0/PWC1 を入力として使用します)。詳細は「25.5.2 パルス幅測定機能の動作」を参照してください。

<注意事項>

PWC タイマ起動後の書換えは禁止です。必ず起動前か停止後に書き込んでください。

[bit3] S/C(測定モード (単発 / 連続) の選択)

このビットは、測定モードを選択します。

表 25.3-5 16 ビットアップカウンタタイマの測定モードの選択

S/C	測定モード選択	タイマモード時	パルス幅
0	単発測定モード [初期値]	リロードなし (ワンショット)	1 回測定後停止
1	連続測定モード	リロードあり (リロードタイマ)	連続測定 : バッファレジスタ有効

- リセット時 "0" に初期化されます。
- 読出し / 書込みが可能です。

<注意事項>

PWC タイマ起動後の書換えは禁止です。必ず起動前か停止後に書き込んでください。

[bit2, bit1, bit0] MOD2, MOD1, MOD0(動作モード / 測定エッジ選択)

これらのビットは、動作モードおよび幅測定を行うエッジを選択します。

表 25.3-6 16 ビットアップカウントタイマの動作モード / 測定エッジの選択

MOD2	MOD1	MOD0	動作モード / 測定エッジ選択
0	0	0	タイマモード [初期値]
0	0	1	タイマモード (リロードモードのみ)
0	1	0	全エッジ間パルス幅測定モード (↑ or ↓ ~ ↓ or ↑)
0	1	1	分周周期測定モード (入力分周有効)
1	0	0	立上りエッジ間周期測定モード (↑ ~ ↑)
1	0	1	"H" パルス幅測定モード (↑ ~ ↓)
1	1	0	"L" パルス幅測定モード (↓ ~ ↑)
1	1	1	立下りエッジ間周期測定モード (↓ ~ ↓)

- リセット時 "000_B" に初期化されます。
- 読出し / 書込みが可能です。

<注意事項>

PWC タイマ起動後の書換えは禁止です。必ず起動前か停止後に書き込んでください。

25.3.2 PWC データバッファレジスタ (PWCR0 ~ PWCR2)

PWC データバッファレジスタ (PWCR0 ~ PWCR2) の構成および機能について説明します。

■ PWC データバッファレジスタ (PWCR0 ~ PWCR2)

図 25.3-3 に、PWC データバッファレジスタ (PWCR0 ~ 2) のビット構成を示します。

図 25.3-3 PWC データバッファレジスタ (PWCR0 ~ 2) のビット構成

ch. 0	000079 _H	15	14	13	12	11	10	9	8	PWCR
ch. 1	00007D _H	D15	D14	D13	D12	D11	D10	D9	D8	PWCデータバッファレジスタ
ch. 2	000081 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	↳ リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	↳ 初期値
ch. 0	000078 _H	7	6	5	4	3	2	1	0	PWCR
ch. 1	00007C _H	D7	D6	D5	D4	D3	D2	D1	D0	PWCデータバッファレジスタ
ch. 2	000080 _H	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	↳ リード/ライト
		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	↳ 初期値

PWC データバッファレジスタ (PWCR0 ~ PWCR2) は、PWCSR レジスタの bit2 ~ bit0(MOD2 ~ MOD0) の設定によるタイマモード時と、パルス幅測定モード時で機能が異なります。

● タイマモード時 (読出し / 書込み可能)

リロードタイマ動作時 (PWCSR の bit3:S/C=1) は、リロードデータを保持するバッファレジスタとなります。この場合、読出し / 書込みともに可能です。

ワンショットタイマ動作モード時 (PWCSR の bit3:S/C=0) は、アップカウンタイマを直接アクセスするバッファレジスタになります。この場合も、読出し / 書込みともに可能ですが、書込みはタイマ停止中に行ってください。読出しは随時可能で、カウント中のタイマ値を読み出すことができます。

● パルス幅測定モード時 (読出しのみ可能)

連続測定モード時 (PWCSR の bit3:S/C=1) は、前回の測定結果を保持するバッファレジスタとなります。

この場合は読出しのみ可能で、書き込んでもレジスタ値は変化しません。

単発モード時 (PWCSR の bit3:S/C=0) は、アップカウンタイマを直接アクセスするバッファレジスタとなります。

この場合も読出しのみ可能で、書き込んでもレジスタ値は変化しません。読出しは随時可能で、カウント中のタイマ値が得られます。測定終了後は、測定結果を保持します。

<注意事項>

このレジスタのアクセスは、必ずワード転送命令で行ってください。また、リセット時は "0000_H" に初期化されます。

25.3.3 分周比制御レジスタ (DIVR0 ~ DIVR2)

分周比制御レジスタ (DIVR0 ~ DIVR2) の構成および機能について説明します。

■ 分周比制御レジスタ (DIVR0 ~ DIVR2)

図 25.3-4 に、分周比制御レジスタ (DIVR0 ~ DIVR2) のビット構成を示します。

図 25.3-4 分周比制御レジスタ (DIVR0 ~ DIVR2) のビット構成

ch. 0	000082 _H	7	6	5	4	3	2	1	0	DIVR
ch. 1	000084 _H	—	—	—	—	—	—	DIV1	DIV0	分周比制御レジスタ
ch. 2	000086 _H	(—)	(—)	(—)	(—)	(—)	(—)	(R/W)	(R/W)	↔ リード/ライト
		(—)	(—)	(—)	(—)	(—)	(—)	(0)	(0)	↔ 初期値

分周周期測定モード (PWCSR の bit2, bit1, bit0:MOD2, MOD1, MOD0=001) において使用するレジスタであり、ほかのモードでは分周比制御レジスタ (DIVR0 ~ DIVR2) の設定は意味をもちません。

分周周期測定モードには、このレジスタによって設定された分周比だけ測定端子に入力されたパルスを分周し、分周後の 1 周期幅を測定します。

表 25.3-7 分周比の選択

DIV1	DIV0	カウントクロック選択
0	0	4 分周 [初期値]
0	1	16 分周
1	0	64 分周
1	1	256 分周

- リセット時 "00_B" に初期化されます。
- 読出し / 書込みが可能です。

<注意事項>

PWC タイマ起動後の書換えは禁止です。必ず起動前か停止後に書き込んでください。

25.4 PWC タイマの割込み

PWC タイマの割込みは、タイマ機能時のアップカウンタのオーバフローとパルス幅測定終了時に発生します。DMA 転送および拡張インテリジェント I/O サービス (EI²OS) を起動することができる PWC 割込みは PWC0 のみです。

■ 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下記の表に示します。

	パルス幅測定終了割込み	タイマモード動作時のタイマの オーバフロー割込み
割込み要求フラグ	PWCSR0:EDIR (bit13) ch.0 PWCSR1:EDIR (bit13) ch.1 PWCSR2:EDIR (bit13) ch.2	PWCSR0:OVIR (bit11) ch.0 PWCSR1:OVIR (bit11) ch.1 PWCSR2:OVIR (bit11) ch.2
割込み要求出力許可ビット	PWCSR0:EDIE (bit12) ch.0 PWCSR1:EDIE (bit12) ch.1 PWCSR2:EDIE (bit12) ch.2	PWCSR0:OVIE (bit10) ch.0 PWCSR1:OVIE (bit10) ch.1 PWCSR2:OVIE (bit10) ch.2
割込み発生要因	パルス幅測定終了した時	16 ビットアップカウンタタイマが FFFF _H → 0000 _H にオーバフローした時

■ PWC タイマに関する割込み要因

割込み要因は下記の 2 つがあります。

- タイマ機能時に PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) の OVIR(bit11) フラグがセットされ、OVIE(bit10) が "1" の場合にオーバフロー割込みが発生します。
- パルス幅測定時に PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2) の EDIR(bit13) フラグがセットされ、EDIE(bit12) が "1" の場合 (送信終了時) に割込みが発生します。

■ PWC タイマの割込みと DMA 転送および EI²OS

表 25.4-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 25.4-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
PWC1 (MB90485 シリーズのみ)	○	×	#19	FFFFB0 _H	ICR04	0000B4 _H
PWC2 (MB90485 シリーズのみ)	○	×	#20	FFFFAC _H		
PWC0 (MB90485 シリーズのみ)*	○	1	#21	FFFA8 _H	ICR05	0000B5 _H

×：割込み要求フラグはクリアされません。

○：割込み要求フラグはクリアされます。

*：この割込み要因は、ほかの周辺機能の割込み要因と割込み番号を共有しています。詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

PWC タイマは EI²OS 機能のみに対応し、DMA 転送機能には対応していません。EI²OS 機能を使用する場合には、割込み制御レジスタ (ICR) を共有する他の割込みを、禁止に設定する必要があります。

25.5 PWC タイマの動作

PWC タイマの動作について説明します。

■ PWC タイマ動作概略

PWC タイマは、16 ビットアップカウントタイマを基本にした多機能タイマであり、測定入力端子と 8 ビット入力分周回路などが組み込まれています。PWC タイマには、次の 2 つの主要機能があります。

- タイマ機能
- パルス幅カウント機能

どちらの機能でも 3 種類 (マシナクロックの 4 分周 / 16 分周 / 32 分周) のカウントクロックを選択できます。以下に、各機能における基本性能 / 動作について説明します。

25.5.1 タイマ機能の動作

アップカウントタイマはリロード動作とワンショット動作が可能です。

■ タイマ機能の動作

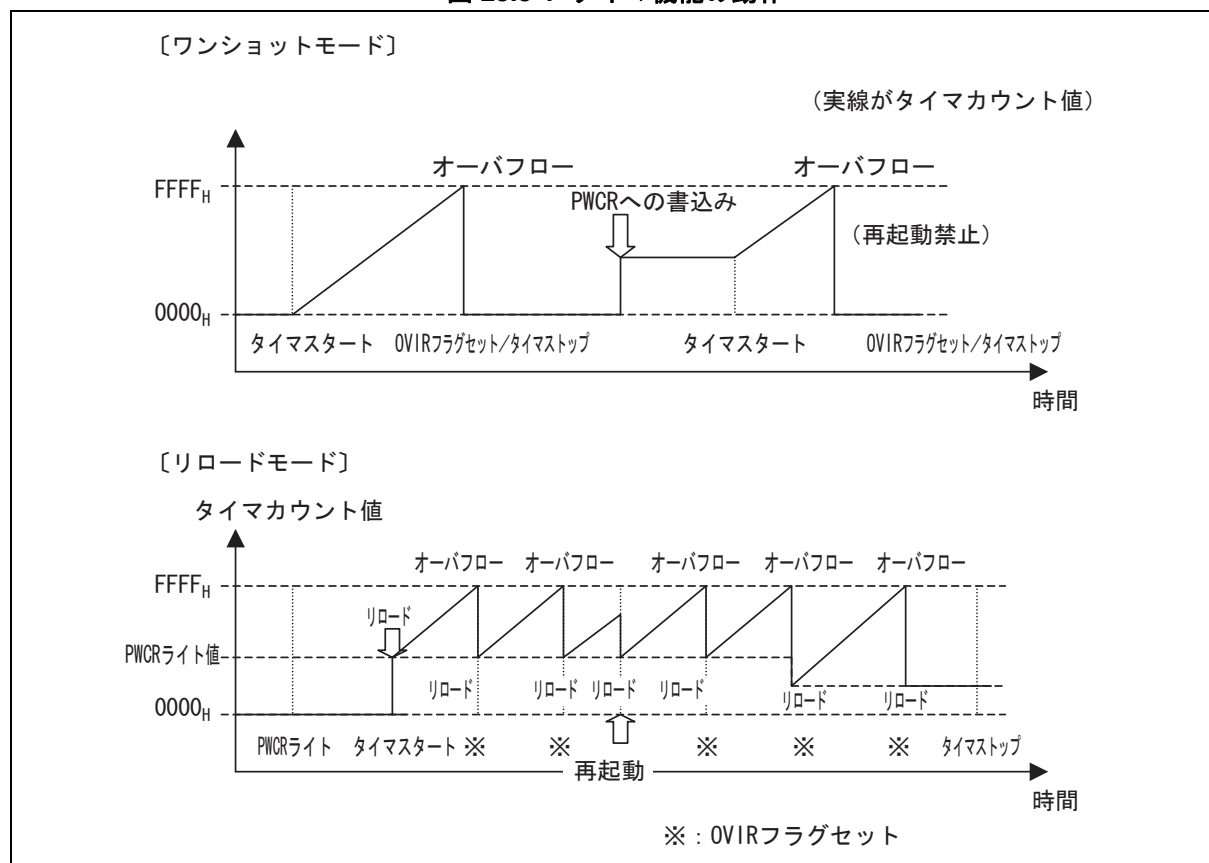
タイマ起動後, カウントクロックごとにカウントアップを行います。オーバーフロー (FFFF_H → 0000_H) が発生すると, 割込み要求が発生する場合があります。

オーバーフロー発生した場合, 本タイマはモードによって以下のように動作します。

- ワンショットモード: カウントを停止します。
- リロードモード: タイマにリロードレジスタの内容をリロードし, 再びカウントを開始します。

図 25.5-1 に, タイマ機能のワンショットモードおよびリロードモードの動作を示します。

図 25.5-1 タイマ機能の動作



25.5.2 パルス幅測定機能の動作

入力パルスの任意イベント間の時間周期をタイマで測定できます。

■ パルス幅測定機能の動作

パルス幅測定機能は、設定した測定開始エッジが入力されるまでカウントを開始しません。開始エッジを検出するとタイマを 0000_H にクリア後カウントアップを開始し、停止エッジを検出するとカウントを停止します。この間のカウント値がパルス幅としてレジスタに保存されます。測定終了時に割込みを検出します。

測定終了後は、本タイマは測定モードに応じて以下のように動作します。

- 単発測定モード：動作を中断します。
- 連続測定モード：タイマ値はバッファレジスタに転送され、次の開始エッジが入力されるまで測定が中断されます。

図 25.5-2 に単発測定モードの動作を、図 25.5-3 に連続測定モードの動作を示します。

図 25.5-2 パルス幅測定動作 (単発測定モード/"H" 幅測定)

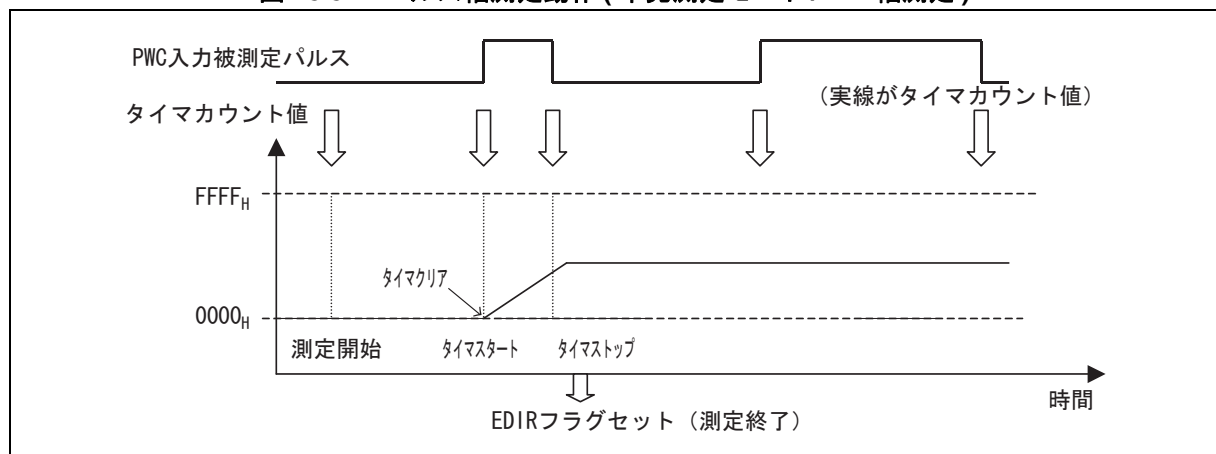
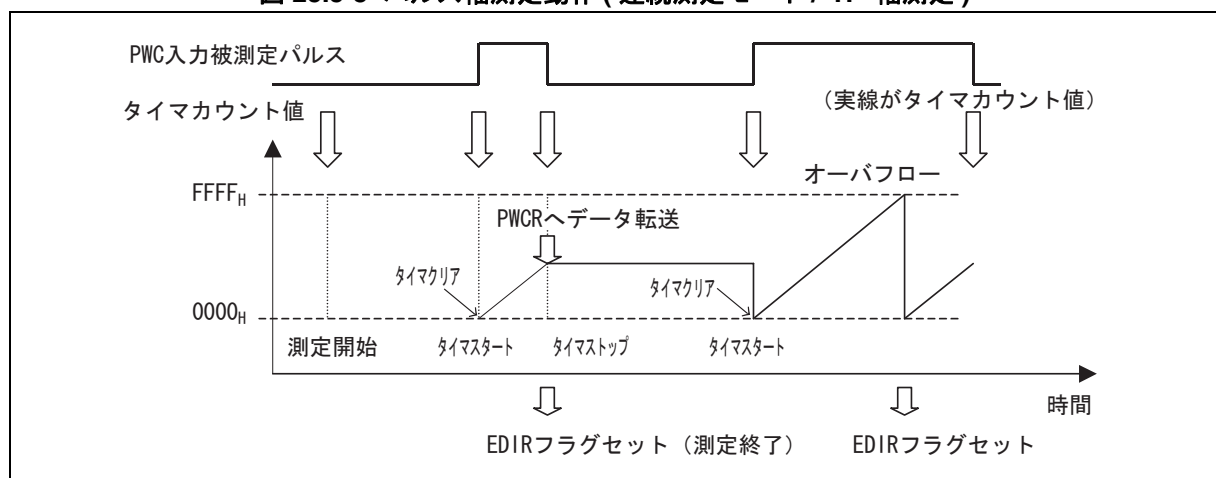


図 25.5-3 パルス幅測定動作 (連続測定モード/"H" 幅測定)



25.5.3 カウントクロックおよび動作モードの選択

カウントクロックの選択および動作モードの選択について説明します。

■ カウントクロックの選択

タイマのカウントクロックは、PWCSR の bit7(CKS1) および bit6(CKS0) の設定によって内部クロックソース 3 種類から選択ができます。

表 25.5-1 に、カウントクロックの選択内容を示します。

表 25.5-1 カウントクロックの選択内容

PWCSR/bit7, bit6:CKS1, 0	選択される内部カウントクロック
00 _B	マシニングロックの 4 分周 (16MHz のマシニングロックの場合は 0.25μs) [初期値]
01 _B	マシニングロックの 16 分周 (16MHz のマシニングロックの場合は 1.0μs)
10 _B	マシニングロックの 32 分周 (16MHz のマシニングロックの場合は 2.0μs)

リセット後の初期値では、マシニングロックの 4 分周が選択されます。

<注意事項>

カウントクロックの選択は、必ずタイマ起動前に行ってください。

■ 動作モードの選択

各動作モード / 測定モードの選択は、PWCSR ビットの設定によって行います。

- 動作モードの選択 :PWCSR の bit2, 1, 0(MOD2, MOD1, MOD0 ビット)
タイマモード / パルス幅測定モードの選択, 測定エッジの決定など
- 測定モードの設定 :PWCSR の bit3(S/C ビット)
単発測定 / 連続測定またはリロード / ワンショットの選択

表 25.5-2 に、動作モード / 測定モードの設定内容を示します。

表 25.5-2 動作モード / 測定モードの設定内容

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
パルス幅測定	↑ or ↓ ~ ↑ or ↓ 全エッジ間測定	単発測定：バッファ無効	0	0	1	0
		連続測定：バッファ有効	1	0	1	0
	分周周期測定 (1 ~ 256 分周)	単発測定：バッファ無効	0	0	1	1
		連続測定：バッファ有効	1	0	1	1
	↑ ~ ↑ 立上り間周期測定	単発測定：バッファ無効	0	1	0	0
		連続測定：バッファ有効	1	1	0	0
	↑ ~ ↓ "H" パルス幅測定	単発測定：バッファ無効	0	1	0	1
		連続測定：バッファ有効	1	1	0	1
	↓ ~ ↑ "L" パルス幅測定	単発測定：バッファ無効	0	1	1	0
		連続測定：バッファ有効	1	1	1	0
	↓ ~ ↓ 立下り間周期測定	単発測定：バッファ無効	0	1	1	1
		連続測定：バッファ有効	1	1	1	1

リセット後の初期設定では、ワンショットタイマが選択されています。

<注意事項>

動作モードの選択は、必ずタイマ起動前に行ってください。

25.5.4 タイマ / パルス幅測定 of 起動と停止

各動作の起動 / 再起動 / 停止 / 強制停止は , PWCSR の bit15, 14 (STRT, STOP ビット) によって制御されます。

■ タイマ / パルス幅測定 of 起動と停止

タイマ / パルス幅測定 of 起動 / 再起動は STRT ビットで制御され , 強制停止は STOP ビットとで制御されます。それぞれ "0" を書き込むことによって機能します。しかし , このときに両ビットに書き込む値が排他的でないとは機能しません。ビット操作命令以外の命令 (バイト以上の命令) で書き込む場合は , 必ず表 25.5-3 に示すビットの組合せに限定されます。

表 25.5-3 STRT ビットと STOP ビットの機能

STRT	STOP	機能
0	1	タイマ / パルス幅測定 of 起動 / 再起動
1	0	タイマ / パルス幅測定 of 強制停止

ビット操作命令 (クリアビット命令) を用いる場合は , ハードウェアによって自動的に表 25.5-3 に示す組合せで書き込まれるため , 特に設定する必要はありません。

■ 起動後の動作

タイマモードおよびパルス幅測定モード of 起動後の動作は , 以下のとおりです。

● タイマモード

直ちにカウント動作を開始します。

● パルス幅測定モード

測定開始エッジが入力されるまでカウントは行いません。測定開始エッジ検出後 , 16 ビットアップカウントタイマを 0000_B にクリアし , カウントを開始します。

■ 再起動

タイマ / パルス幅測定モード of 起動後 , 動作中に起動をかける (STRT ビットに "0" を書き込む) ことを再起動とよびます。

再起動をかけると , モードによって , 以下の動作が行われます。

● ワンショットタイマモード

動作に影響はありません。

● リロードタイマモード

リロード動作を行い , 動作を続行します。オーバフロー発生タイミングと同時に再起動をかけた時 , オーバフローフラグ (OVIR) がセットされます。

● パルス幅測定モード

測定開始エッジ待ち状態の場合 , 動作に影響はありません。測定中 of 場合は , カウントを停止し再度測定開始エッジ待ち状態となります。この際 , 測定終了エッジ検出と再起動 of タイミングが同時になると , 測定終了割込み要求フラグ (EDIR) がセットされ , 連続測定モード時は結果が PWCR に転送されます。

■ 停止

ワンショットタイマモード/単発測定モードでは、タイマのオーバフローまたは測定終了により自動的にカウントを停止しますので、特に意識して停止させる必要はありません。ただし、ほかのモードでは、タイマを強制的に停止させる必要があります。タイマが自動的に停止する前にタイマを停止させたい場合にも同様です。

● 2 入力比較選択時

選択されている PWC1 からエッジが強制停止の前に検出されていないと、再起動後 1 回目の測定結果が誤ったものとなります。強制停止は、PWC1 からエッジが検出されてからにしてください。

■ 動作状態の確認

前述の STRT, STOP ビットは、読出し時には、動作状態表示ビットとして機能します。

表 25.5-4 に、動作状態表示ビットの機能を示します。

表 25.5-4 動作状態表示ビットの機能

START	STOP	動作状態
0	0	タイマ停止 (測定開始エッジ待ち状態を除く): 起動されていないかまたは測定が終了したことを示します。
1	1	タイマカウント動作中または測定開始エッジ状態待ち

<注意事項>

STRT, STOP どちらのビットを読み出しても同じ値となります。ただし、リードモディファイライト系命令 (ビット処理命令など) でこのビットを読み出すと常に "11_B" になるため、これらの命令を使用して読み出さないでください。

25.5.5 タイマモードの動作

タイマモードの動作について説明します。

■ タイマのクリア

次の場合には、16 ビットアップカウントタイマが 0000_H にクリアされます。

- リセット時
- パルス幅測定モードで、測定開始エッジを検出しカウントを開始した時

■ ワンショット動作モード

ワンショット動作モードでは、タイマ起動後カウントクロックごとにカウントアップを行い、FFFF_H から 0000_H へカウントする際のオーバフロー発生によって自動的に停止します。タイマ起動前に PWCR に値を設定しておく、その値からカウントが開始されます。この場合、設定した値は保持されず、PWCR の値は現行のカウント値となります。

■ リロード動作モード

リロード動作モードでは、タイマ起動後 PWCR 中のリロード値をタイマに設定してからカウントクロックごとにカウントアップを行います。FFFF_H から 0000_H へカウントする際のオーバフロー発生によって、再度 PWCR 中のリロード値をタイマに設定し (リロード動作)、カウント動作を繰り返します。PWCSR の STOP ビットへの書き込みによる強制停止されるかまたはリセットされるまで、タイマは停止しません。タイマ起動前に PWCR に設定した値は、リロード値としてカウント中も保持されます。起動 / 再起動時およびオーバフロー発生時には必ずタイマにリロード値が設定されます。カウント中に設定値を変更する場合は、次のオーバフロー発生時またはタイマの再起動の時に、変更した値が新規リロード値として使用されます。

■ タイマ値とリロード値

ワンショット動作モードでの PWCR は、直接アップカウントタイマにアクセスします。PWCR に値を書き込むとそのままアップカウントタイマに書き込まれ、タイマ動作中に PWCR の値を読み出すとカウント中のタイマ値が得られます。タイマ起動前に PWCR に任意の値を書き込んでから起動すると、この指定された値からカウントが開始されます。リロード動作モードでは、直接アップカウントタイマにアクセスすることは不可能です。PWCR は、リロード値を保持するリロードレジスタとして機能します。タイマの起動 / 再起動 / オーバフローが生じる際には必ず、PWCR に書き込まれた値がタイマに設定されます。PWCR を読み出すと、格納されたリロード値が読み出されます。

ただし、リロード動作モードを強制的に中断させた後にタイマをワンショット動作モードに設定した場合は PWCR の値とタイマの値が定まりません。したがって、タイマの使用前に、必ず値を書き込んでから使用してください。

■ 割込み発生要求

タイマモード動作においては、タイマのオーバフローによる割込み要求を発生させることが可能です。カウントアップによってオーバフローが発生するとオーバフローフラグがセットされ、オーバフロー割込み要求が許可されている場合は、割込み要求が発生します。

■ タイマ周期

ワンショット動作モードで PWCR に 0000_H を設定してタイマを起動すると、65536 回カウントアップ後にオーバフローが発生し、カウントを停止します。起動から停止までの時間は、次の式によって計算できます。

$$T_1 = (65536 - n_1) \times t$$

T_1 : 起動から停止までの時間 [μ s]

n_1 : 起動時に PWCR に書き込まれていたタイマ値

t : カウントクロックの周期 [μ s]

リロード動作モードで PWCR に 0000_H を設定してタイマを起動すると、65536 回カウントアップごとにオーバフローが発生します。リロード周期の時間は、次の式によって計算できます。

$$T_R = (65536 - n_R) \times t$$

T_R : リロード周期 (オーバフロー周期) [μ s]

n_R : PWCR に保持されているリロード値

t : カウントクロックの周期 [μ s]

■ カウントクロックと最大周期

タイマモードの場合、PWCR に設定する値が 0000_H のときに最大周期となります。

マシンクロック (以下 ϕ とする) = 16MHz 時のカウントクロックの周期とタイマの最大周期を表 25.5-5 に示します。

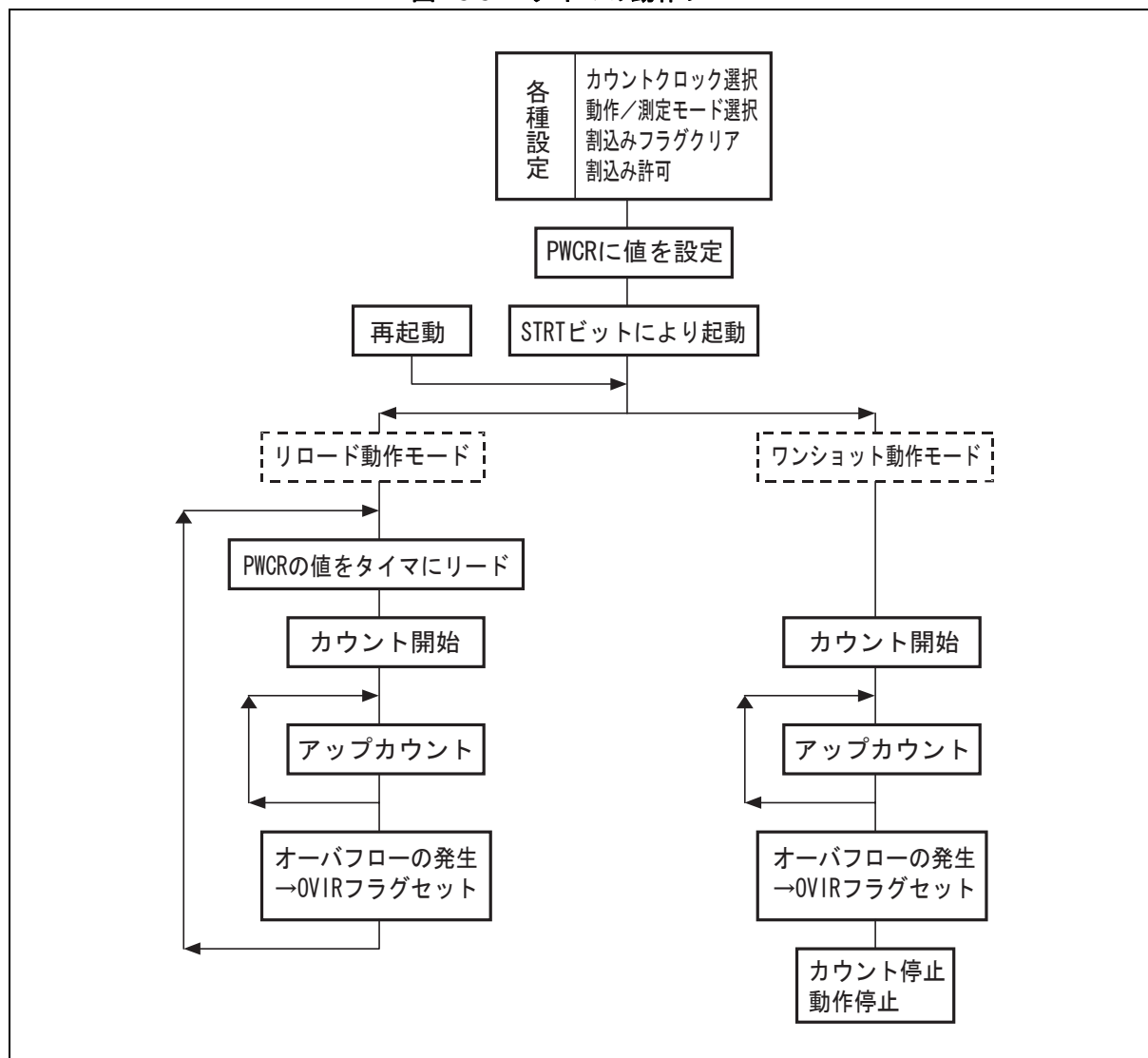
表 25.5-5 カウントクロックと周期

カウントクロック選択	CSK1, 0=00:($\phi/4$) 時	CSK1, 0=00:($\phi/16$) 時	CSK1, 0=00:($\phi/32$) 時
カウントクロック周期	0.25 μ s	1.0 μ s	2.0 μ s
タイマ最大周期	16.38ms	65.5ms	131.1ms

■ タイマの動作フロー

図 25.5-4 に、タイマの動作フローを示します。

図 25.5-4 タイマの動作フロー



25.5.6 パルス幅測定モードの動作

パルス幅測定モードの動作について説明します。

■ 単発測定と連続測定

パルス幅測定モードには、1 回のみ測定を行う単発測定モードと、連続して測定を行う連続測定モードがあります。各モードは、PWCSR の S/C ビットによって選択されます（「25.5.3 カウントクロックおよび動作モードの選択」を参照してください）。

● 単発測定モード

1 回目の測定終了エッジが検出されると、タイマのカウンタが停止して PWCSR 中の測定終了割込み要求フラグ (EDIR) がセットされ、以降は測定が行われません（ただし、同時に再起動がかかった場合は、測定開始待ちとなります）。

● 連続測定モード

測定終了エッジが検出されると、タイマのカウンタが停止して PWCSR 中の測定終了割込み要求フラグ (EDIR) がセットされ、再度測定開始エッジが検出されるまでカウントを停止します。再度、測定開始エッジが検出されると、タイマを 0000_H にクリアした後測定を開始します。測定終了時タイマ内の測定結果は PWCR に転送されます。

<注意事項>

測定モードの選択変更は、必ずタイマの停止中に行ってください。

■ 測定結果のデータ

単発測定モードと連続測定モードでは、測定結果とタイマ値の扱い、および PWCR の機能に違いがあります。両モードにおける測定結果は以下のとおりです。

● 単発測定モード

- 動作中に PWCR を読み出すと、測定中のタイマ値が得られます。
- 測定終了後に PWCR を読み出すと、測定結果データが得られます。

● 連続測定モード

- 測定終了時、タイマ内の測定結果は PWCR に転送されます。
- PWCR を読み出すと直前の測定結果が得られます。測定動作中も前回の測定結果を保持しています。測定中のタイマ値は読み出せません。

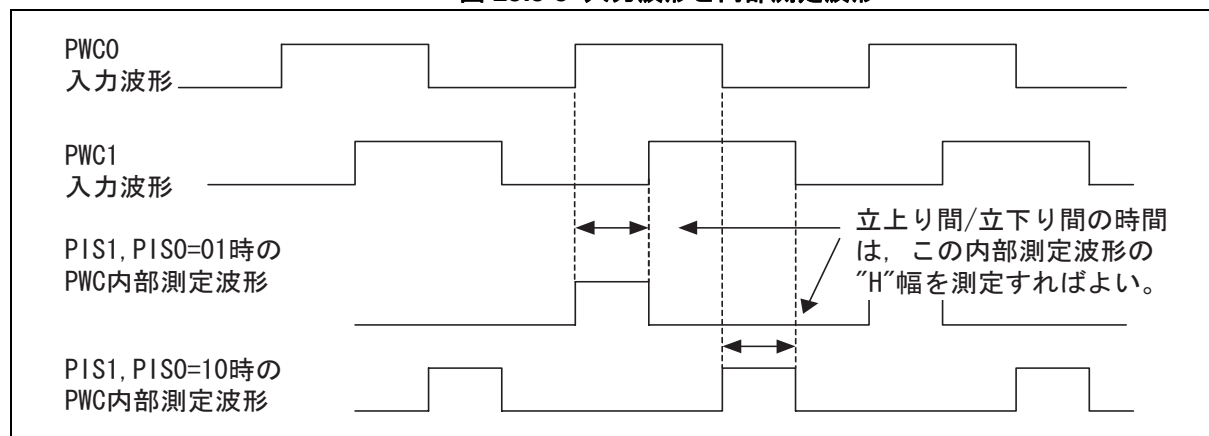
連続測定モードで、測定結果を読み出さないうちに次の測定が終了してしまった場合は、前回の測定結果は新しい測定結果に上書きされ消失してしまいます。この場合、PWCSR 中のエラーフラグ (ERR) がセットされます。エラーフラグ (ERR) は PWCR を読み出すと自動的にクリアされます。

■ 入力端子の選択

PWC タイマは、3 チャンネル用意されています。パルス幅カウント用の信号入力に使用する端子は、PWC0, PWC1, PWC2 の 3 チャンネルです。チャンネルごとに単独での使用も可能です。また、入力端子 PWC0, PWC1 については、PWCSR0 中の PIS1, PIS0 の組合せで、各入力波形の立上り / 立下り間の時間を測定することができます。この際に使用する PWC レジスタは、PWC0 を使用しますので注意してください。

図 25.5-5 に、入力波形と内部測定波形の関係を示します。

図 25.5-5 入力波形と内部測定波形



<注意事項>

- 2 入力比較の場合、立上り / 立下り検出のいずれでも PWC0 からカウントを開始し、PWC1 でカウントを停止してください。
- 立上りと立下りの検出モードを変える場合は、必ず測定終了後に行ってください。

■ 測定モードとカウンタ動作

入力されたパルスのどこを測定するかによって、測定モードは6種類のうちから選択できます。高い周波数のパルスの幅を精度よく測定できるように、入力されたパルスを任意分周して周期を測定するモードも用意されています。表 25.5-6 に、測定モード一覧を示します。

表 25.5-6 測定モード一覧 (1 / 2)

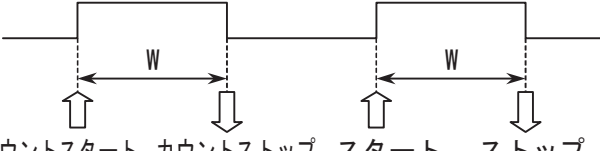
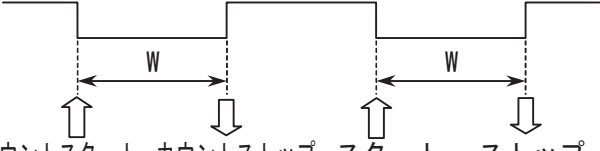
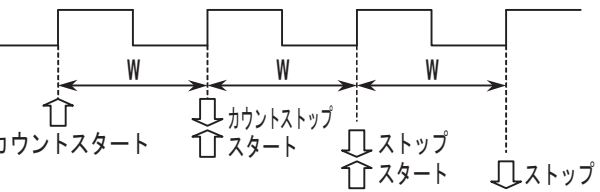
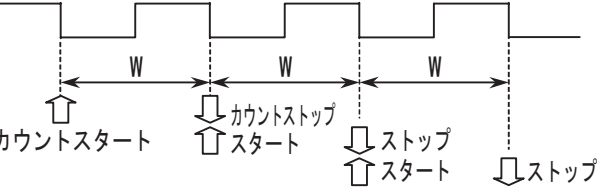
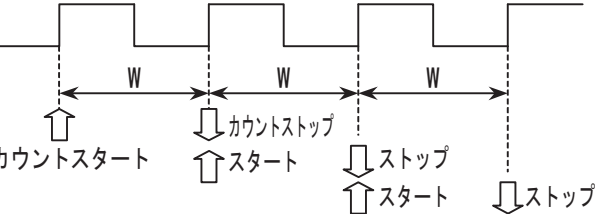
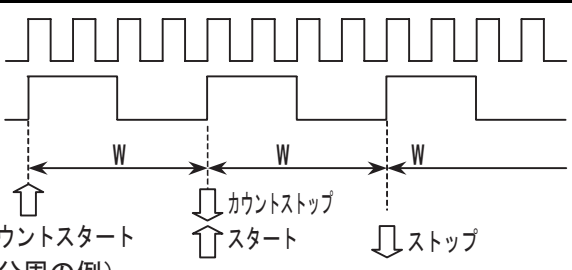
測定モード	MOD2	MOD1	MOD0	測定モード測定内容 (W: 測定するパルス幅)
"H" パルス幅測定	1	0	1	 <p>カウントスタート カウントストップ スタート ストップ</p> <p>"H" 期間の幅を測定します。 カウント (測定) 開始: 立上りエッジ検出時 カウント (測定) 終了: 立下りエッジ検出時</p>
"L" パルス幅測定	1	1	0	 <p>カウントスタート カウントストップ スタート ストップ</p> <p>"L" 期間の幅を測定します。 カウント (測定) 開始: 立下りエッジ検出時 カウント (測定) 終了: 立上りエッジ検出時</p>
立上りエッジ間周期測定	1	0	0	 <p>カウントスタート カウントストップ スタート ストップ</p> <p>立上りエッジ間の周期を測定します。 カウント (測定) 開始: 立上りエッジ検出時 カウント (測定) 終了: 立上りエッジ検出時</p>
立下りエッジ間周期測定	1	1	1	 <p>カウントスタート カウントストップ スタート ストップ</p> <p>立下りエッジ間の周期を測定します。 カウント (測定) 開始: 立下りエッジ検出時 カウント (測定) 終了: 立下りエッジ検出時</p>
全エッジ間パルス幅測定	0	1	0	 <p>カウントスタート カウントストップ スタート ストップ</p> <p>連続して入力されるエッジ間の幅を測定します。 カウント (測定) 開始: エッジ検出時 カウント (測定) 終了: エッジ検出時</p>

表 25.5-6 測定モード一覧 (2 / 2)

測定モード	MOD2	MOD1	MOD0	測定モード測定内容 (W: 測定するパルス幅)
分周周期測定	0	1	1	<div><p>カウントスタート (4分周の例) カウントストップ スタート ストップ</p><p>分周比制御レジスタ DIVR で選択した分周比だけ入力パルスを分周してその周期を測定します。 カウント (測定) 開始: 起動直後の立上りエッジ検出時 カウント (測定) 終了: 分周後の 1 周期終了時</p></div>

どのモードでも , カウントの開始からカウント開始エッジの入力までの間はタイマによるカウントは行われません。カウント開始エッジ入力後に , タイマは 0000_H にクリアされ , カウント終了エッジが入力されるまでは , カウントクロックのたびにタイマのカウントが行われます。カウント終了エッジの入力時には , 次の動作が実行されます。

- PWCSR 中の測定終了割込み要求フラグ (EDIR) がセットされます。
- タイマのカウント動作が停止します (再起動と同時であった場合を除く)。
- 連続測定モードではタイマの値 (測定結果) は PWCR に転送され , 次の測定開始エッジが入力されるまでカウントを中断したままとなります。
- 単発測定モードでは測定を終了します (再起動と同時であった場合を除く)。

なお , 連続測定モードの場合で , 全エッジ間パルス幅測定や分周測定などを行った場合 , 終了エッジが次の測定エッジとなります。

■ 最小入力パルス幅

パルス幅測定入力端子(PWC2～PWC0)に入力できるパルスには、以下の制限があります。

- パルス幅は、4 マシンサイクル (16MHz のマシンクロックの場合は、0.25μs) 以上とすること

■ パルス幅 / 周期算出方法

測定するパルスの幅 / 周期は、次の式で算出できます。

$$T_W = n \times t \div D_{IV} [\mu s]$$

T_W : 被測定パルス幅 / 周期 [μs]

n : PWCR 内の測定結果データ

t : カウントクロックの周期 [μs]

D_{IV} : 分周比制御レジスタ DIVR で選択した分周比

(分周周波数測定モード以外は 1 を代入)

■ パルス幅 / 周期カウントの範囲

カウントクロックおよび入力分周器の分周比の組合せにより、測定可能なパルス幅 / 周期のレンジが変化します。

表 25.5-7 に、マシンクロック (以下 ϕ とする) = 16MHz 時の測定レンジ一覧を示します。

表 25.5-7 パルス幅測定レンジ一覧

分周比	DIV1	DIV0	CKS1, 0=00($\phi/4$) 時	CKS1, 0=01($\phi/16$) 時	CKS1, 0=10($\phi/32$) 時
分周なし	∅	∅	0.25μs ~ 16.38ms [0.25μs]	0.25μs ~ 65.5ms [1.6μs]	0.25μs ~ 131ms [3.2μs]
4 分周	0	0	0.25μs ~ 4.1ms [6.25μs]	0.25μs ~ 16.38ms [0.4μs]	0.25μs ~ 52.4ms [0.8μs]
16 分周	0	1	0.25μs ~ 1.024ms [15.6ns]	0.25μs ~ 4.1ms [1.6μs]	0.25μs ~ 13.1ms [0.2μs]
64 分周	1	0	0.25μs ~ 256μs [3.91μs]	0.25μs ~ 1.024ms [25μs]	0.25μs ~ 3.27ms [50ns]
256 分周	1	1	0.25μs ~ 64μs [0.98μs]	0.25μs ~ 256μs [6.25μs]	0.25μs ~ 817μs [12.5ns]

(注意事項) [] 内は、1 ビットあたりの分解能を示します。

■ 割込み要求発生

パルス幅測定モードにおいては、以下の 2 つの割込み要求を発生させることができます。

● タイマのオーバフローによる割込み要求

測定中、カウントアップによってオーバフローが発生すると、オーバフローフラグがセットされます。オーバフロー割込み要求が許可されている場合は、割込み要求が発生します。

● 測定終了による割込み要求

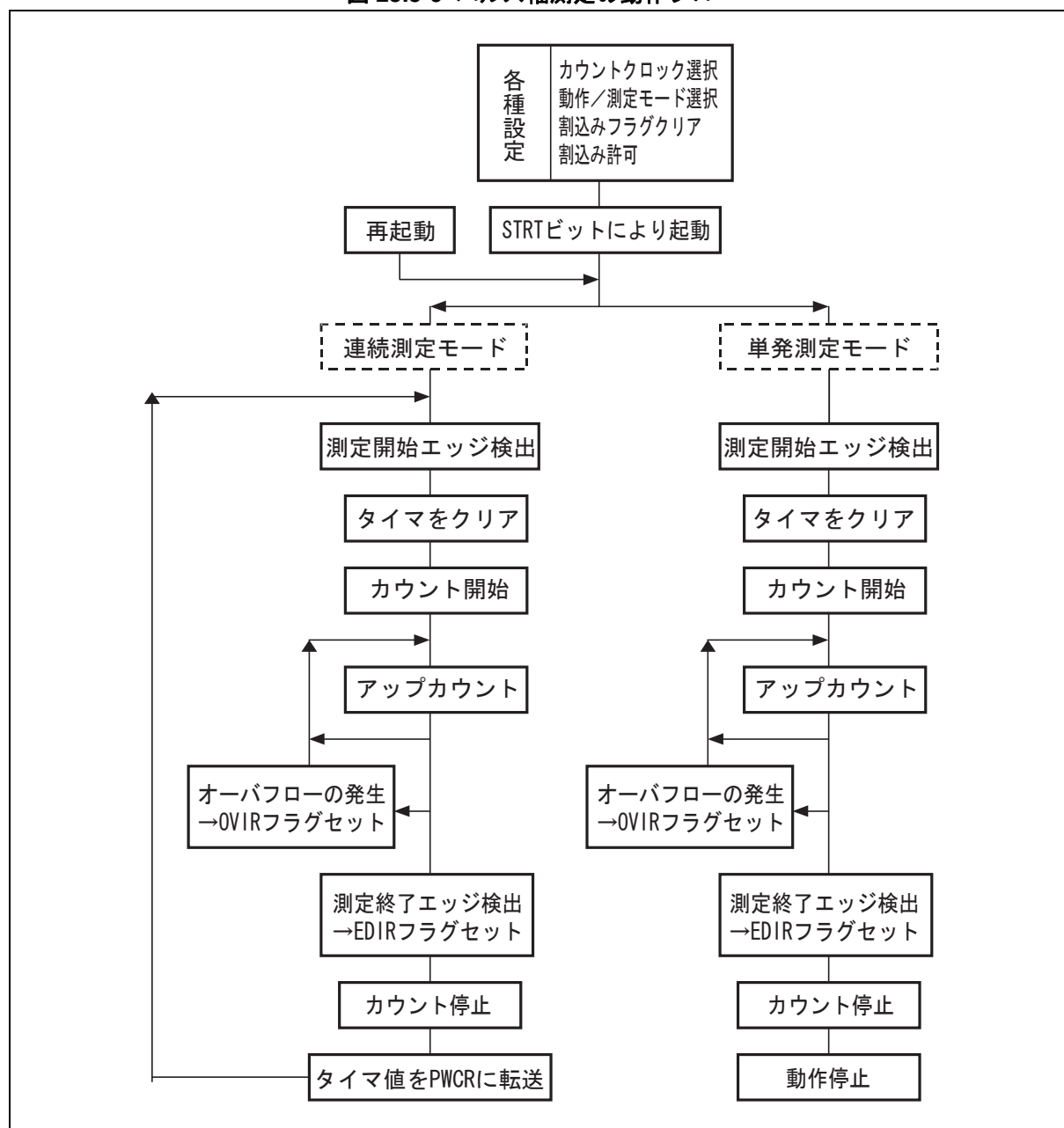
測定終了エッジを検出すると PWCSR 中の測定終了フラグ (EDIR) がセットされ、測定終了割込み要求が許可されていると割込み要求が発生します。

測定終了フラグ EDIR は、測定結果 PWCR を読み出すと同時に自動的にクリアされます。

■ パルス幅測定 of 動作フロー

図 25.5-6 に、パルス幅測定 of 動作フローを示します。

図 25.5-6 パルス幅測定 of 動作フロー



25.6 PWC タイマの使用上の注意

PWC タイマの使用上の注意について説明します。

■ PWC タイマの使用上の注意

● レジスタの書換えに関する注意事項

PWCSR 中の 以下に示すビットは、動作中に書き換えることを禁止します。書換えは、必ず起動前か停止後に行ってください。

- [bit7, bit6] CKS1, CKS0(クロック選択)
- [bit5, bit4] PIS1, PIS0(入力信号選択)
- [bit3] CKS1, CKS0(クロック選択)
- [bit2, bit1, bit0] MOD2, MOD1, MOD0(動作モード / 測定エッジ選択)

DIVR は、動作中に書き換えることを禁止します。書換えは必ず起動前か停止後に行ってください。

● タイマモードの測定終了フラグの扱い

PWCSR 中の測定終了割込み要求フラグ (EDIR) の値は、タイマモードでは、意味を持ちません。このため、PWCSR 中の測定終了割込み許可ビット (EDIE) は、タイマモードで使用するときは必ず "0" を設定してください。

● PWCSR 中の STRT/ STOP ビットの扱い

両ビットとも書込み時と読出し時では、意味が異なるので注意してください (「25.3.1 PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR02)」を参照)。また、リードモディファイライト系命令における読出し値は、ビット値にかかわらず "11_B" です。このため、動作状態の読出しには、ビット処理命令は使用できません(読み出すと必ず動作中となります)ので注意してください。タイマの起動/停止のための STRT/ STOP ビットへの書込みは、それぞれのビットに対するビット処理命令 (ビットクリア命令など) を用いることができます。

● タイマのクリア

パルス幅測定モードの場合、測定開始エッジでタイマがクリアされますので、起動前にタイマ中のデータは無効となります。

● モード変更時の PWCR とタイマの値

- リロードタイマモードで使用した後、タイマを強制停止してからワンショットタイマモードにした場合は、PWCR に保持された値およびタイマの値は不定となりますので、必ず値を書き込んでから使用してください。
- ワンショットタイマモードで使用した場合、PWCR の値は不定となりますので、必ず値を書き込んでから使用してください。
- パルス幅測定モードから、タイマモードへ切り換える場合は、起動前に必ず PWCR に値を再設定してから起動してください。

● 最小パルス幅

パルス幅測定入力端子に入力できるパルスには、以下の制限があります。

- 最小パルス幅：マシクロックの 2 分周 (16MHz マシクロック時, 0.25 μ s 以上)
 - 最小入力周波数：マシクロックの 4 分周 (16MHz マシクロック時, 4MHz 以下)
- 上記パルスより小さい幅や高い周波数のパルスを入力した場合は、動作は保証できません。入力信号にそのようなノイズがのる可能性がある場合は、チップ外部でフィルタなどを通して除去してから入力してください。

● 分周周波数測定モード

パルス幅測定モードのうちの分周周期測定モードでは、入力パルスを分周するため、測定結果より算出して得られるパルス幅は、平均値となります。

● クロック選択ビットの扱い

PWCR 中の [bit7, bit6] CKS1, CKS0(クロック選択)には, "11_B"を設定しないでください。

● 予約ビットの扱い

PWCR 中の [bit8] は、予約ビットとなっています。このビットに書き込む場合は, "0" にしてください。

● 動作中の再起動

カウント動作を開始した後に再起動をする場合は、そのタイミングによっては以下に示すようなことが起こり得ます。

- リロードタイマモード時において、オーバフロー発生と同時に再起動をかけた場合、オーバフローフラグ (OVIR) がセットされます。
- パルス幅単発測定モードにおいて、測定終了エッジと同時に再起動をかけた場合、測定終了割込み要求フラグ (EDIR) がセットされます。
- パルス幅連続測定モードにおいて、測定終了エッジと同時に再起動をかけた場合、測定終了割込み要求フラグ (EDIR) はセットされ、その時点での測定結果は PWCR に転送されます。

● PWC タイマを「"H" パルス幅または "L" パルス幅測定モードを連続測定モード」で使用している場合

パルス幅測定が完了し次のパルス幅測定開始待ち状態になっている間、本来停止されるはずタイマが動作し続け、次のパルス幅測定開始前にタイマのオーバフローフラグ (OVIR) がセットされてしまうことがあります。このため、次回パルス幅測定終了時にオーバフローしていない場合でもオーバフローフラグがセットされている場合があります。したがって、PWC タイマを "H" パルス幅または "L" パルス幅測定モードを連続測定モードでご使用になる場合には、オーバフローフラグを使用しないでください。

第 26 章

μ PG タイマ (MB90485 シリーズのみ)

μ PG タイマの概要と構成, レジスタの構成と機能およびタイミングチャートについて説明します。

26.1 μ PG タイマの概要と構成

26.2 μ PG タイマのレジスタの構成と機能

26.3 μ PG タイマのタイミングチャート

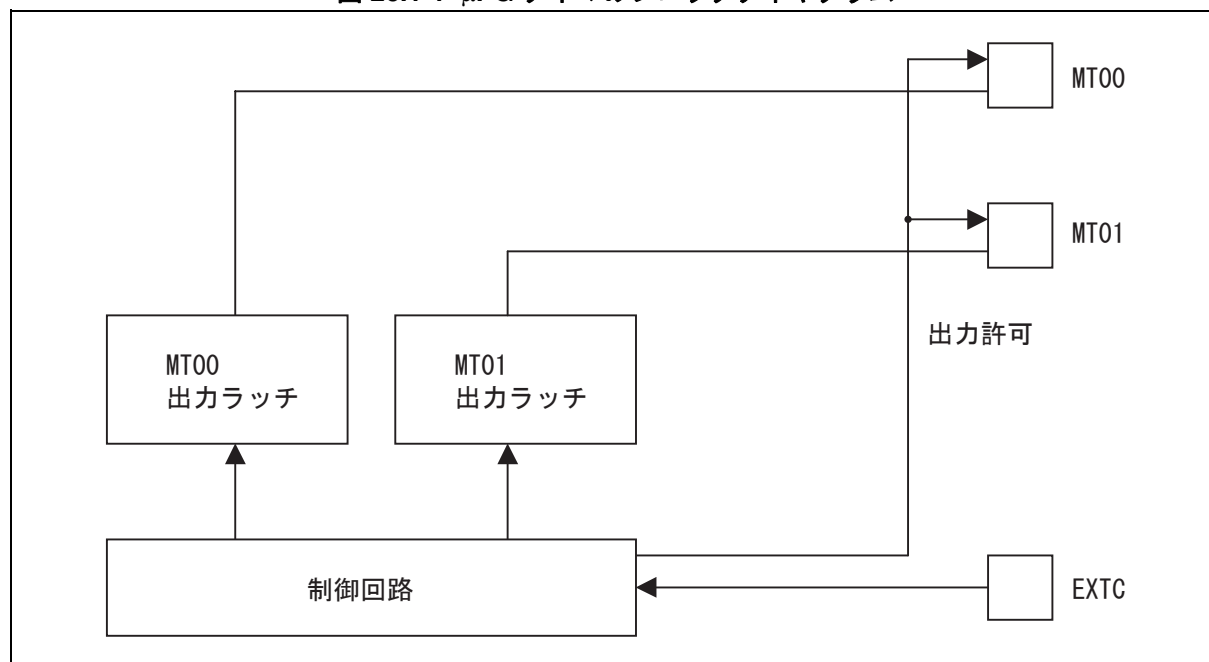
26.1 μ PG タイマの概要と構成

μ PG タイマは、外部入力に応じたパルス出力を行います。

■ μ PG タイマのブロックダイアグラム

図 26.1-1 に、 μ PG タイマのブロックダイアグラムを示します。

図 26.1-1 μ PG タイマのブロックダイアグラム



■ μ PG タイマに関する端子

μ PG タイマは、EXTC 端子が入力ポート、MT00、MT01 端子が出力ポートとして機能します。EXTC 端子は、汎用入出力ポート (P45/EXTC) と、 μ PG タイマの入力端子を兼用しています。MT00、MT01 端子は、汎用入出力ポート (P43/MT00、P44/MT01) と、 μ PG タイマの出力端子とを兼用しています。

● EXTC 端子として使用する場合の設定

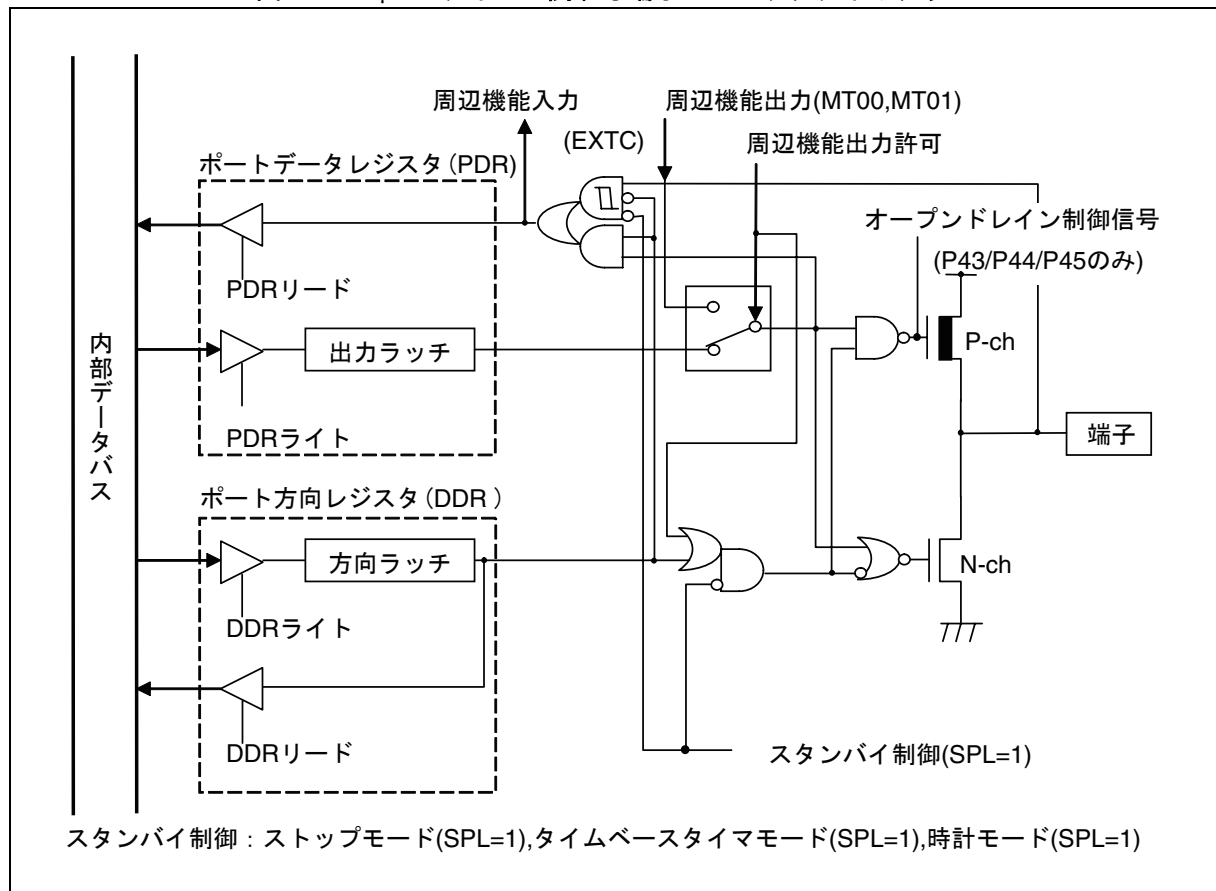
μ PG タイマで EXTC 端子として使用する場合は、P45/EXTC 端子を、ポート方向レジスタにより入力ポート (DDR4 bit5 → "0") に設定してください。

● MT00/MT01 端子として使用する場合の設定

μ PG タイマで MT00/MT01 を出力として使用する場合は、 μ PG コントロール/ステータスレジスタ (PGCSR) によりパルス出力許可 (PE0/PE1 bit5, 6 → "1") に設定してください。

■ μ PG タイマに関する端子のブロックダイアグラム

図 26.1-2 μ PG タイマに関する端子のブロックダイアグラム



26.2 μPG タイマのレジスタの構成と機能

μPG タイマに関するレジスタの構成と機能について説明します。

■ μPG コントロール / ステータスレジスタ (PGCSR)

μPG コントロール / ステータスレジスタ (PGCSR) のビット構成を下図に示します。

00008E _H	7	6	5	4	3	2	1	0	PGCSR μPGコントロール/ ステータスレジスタ リード/ライト 初期値
	PEN0	PE1	PE0	PMT1	PMT0	—	—	—	
	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)	
	(0)	(0)	(0)	(0)	(0)	(—)	(—)	(—)	

以下に , μPG コントロール / ステータスレジスタ (PGCSR) の各ビットの機能を説明します。

[bit7] PEN0(動作許可)

このビットは , μPG タイマの動作許可を制御します。

PEN0	機能
0	停止 ("L" レベル保持) (初期値)
1	PG 動作許可

このビットは , リセットにより初期化されます。

[bit6, bit5] PE1, PE0(出力許可)

このビットは , パルス出力外部端子 (MT00, MT01) を制御します。

PE1	PE0	動作制御機能
0	0	汎用ポート端子 (パルス出力禁止) (初期値)
0	1	MT00 パルス出力端子のみ (出力許可)
1	0	MT01 パルス出力端子のみ (出力許可)
1	1	MT00, MT01 パルス出力端子 (出力許可)

このビットは , リセット時 "00_B" に初期化されます。

[bit4, bit3] PMT1, PMT0(出力反転)

これらのビットは , 各パルス出力の反転制御を行います。

PMT1	PMT0	動作制御機能
0	0	起動時の波形 (初期値)
0	1	MT00 のみ反転
1	0	MT01 のみ反転
1	1	MT00/MT01 反転

これらのビットは , リセット時 "00_B" に初期化されます。

[bit2, bit1, bit0] 未定義ビット

これらのビットは , 使用しませんので , 通常は "000" を指定してください。

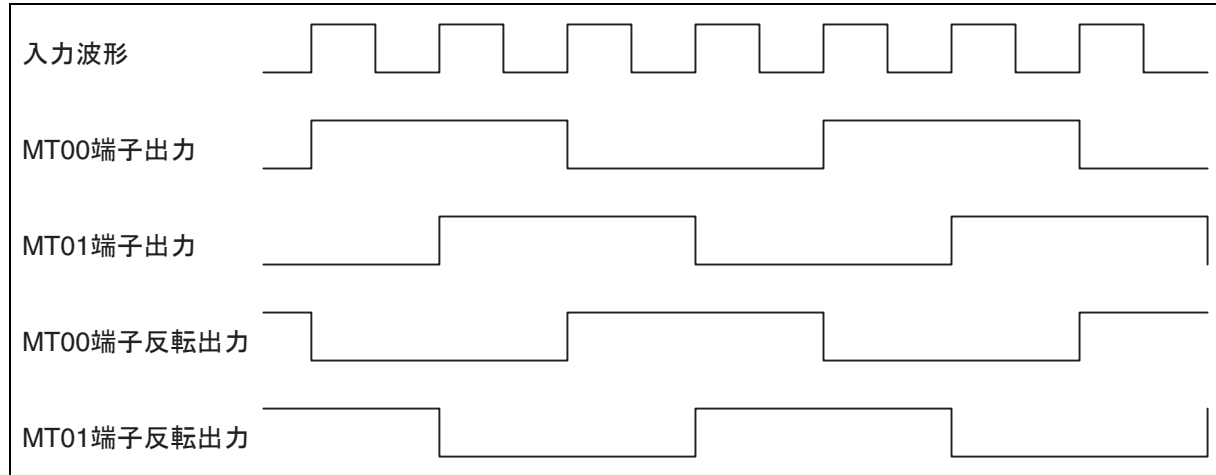
26.3 μ PG タイマのタイミングチャート

μ PG タイマのタイミングチャートおよびタイミングに関する注意を示します。

■ μ PG タイマのタイミングチャート

図 26.3-1 に、 μ PG タイマの入力信号および出力信号のタイミングチャートを示します。

図 26.3-1 μ PG タイマの入力信号および出力信号のタイミングチャート



■ タイミングに関する注意

- 入力波形に対して図 26.3-1 に示す波形を出力します。デューティは 50% 固定です。
- 起動後、1 つ目の立上りパルスで同期をとっているため、出力は 2 つ目の立上りパルスから出力されます。
- 入力波形に対して 2 つの出力は、プログラムでの反転制御しか行いません。
- 入力パルス (EXTC 端子の波形) の同期は、内部クロック (マシナクロック) より 10 倍以上にしてください。
- μ PG コントロール/ステータスレジスタに PE0, PE1=11_B を書き込む (各端子の出力を許可する) ときは、入力端子 (EXTC) へはパルス入力を行わないでください。入力端子 (EXTC) にパルスを入力中に端子出力を許可すると、第 1 出力パルス幅が入力パルス幅より短くなる場合があります。

第27章

I²C インタフェース (MB90485 シリーズのみ)

I²C インタフェースの概要，構成，レジスタの構成と機能，割込みおよび動作について説明します。

27.1 I²C インタフェースの概要

27.2 I²C インタフェースの構成

27.3 I²C インタフェースのレジスタの構成と機能

27.4 I²C インタフェースの割込み

27.5 I²C インタフェースの動作

27.1 I²C インタフェースの概要

I²C インタフェースは、Inter IC BUS をサポートし、I²C バス上のマスタ / スレーブ デバイスとして動作するシリアル I/O ポートです。

■ I²C インタフェースの機能

I²C インタフェースは、以下の機能をもっています。

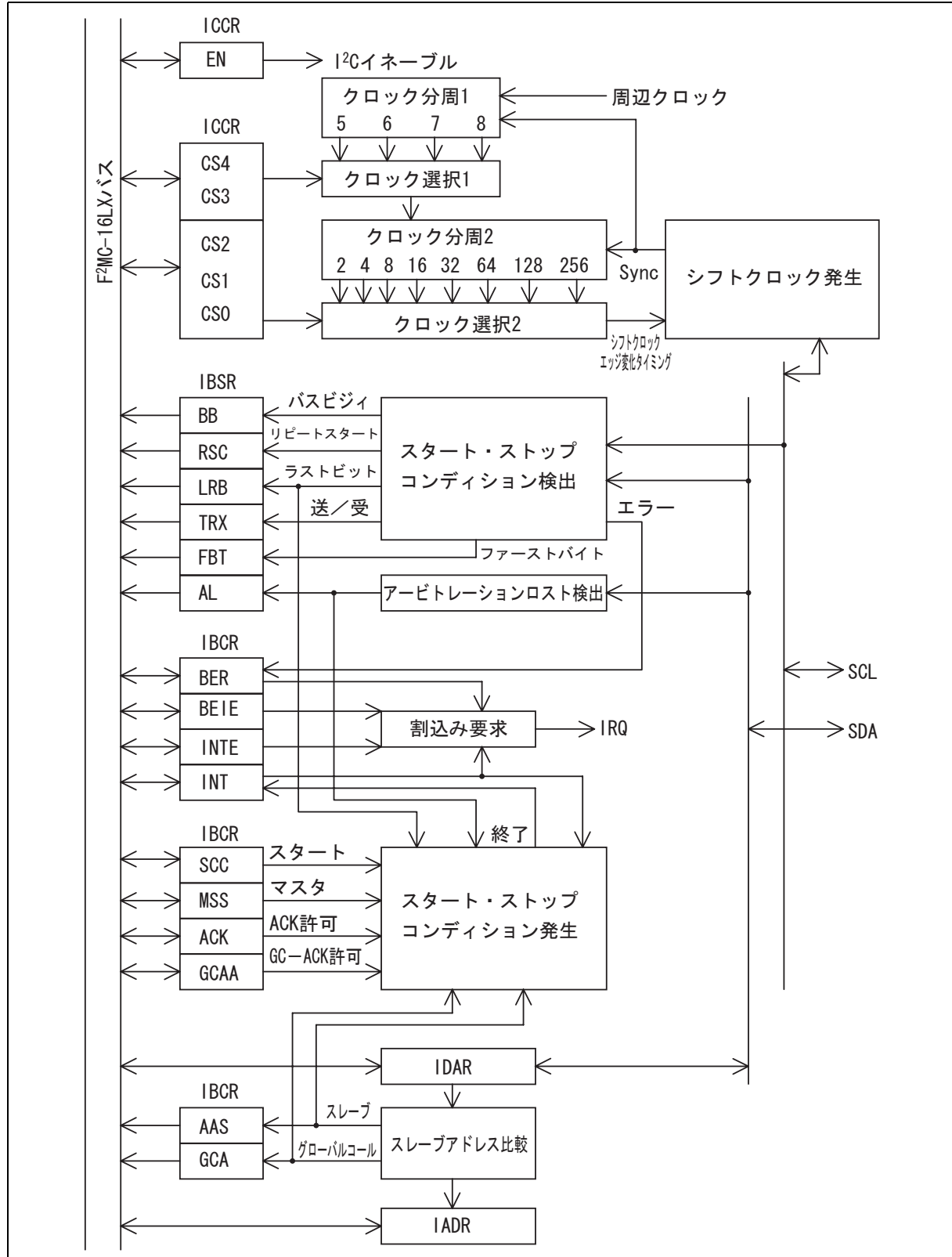
- マスタ / スレーブ 送受信機能
- アービトレーション機能
- クロック同期化機能
- スレーブアドレス / ゼネラルコールアドレス検出機能
- 転送方向検出機能
- スタートコンディションの繰り返し発生および検出機能
- バスエラー検出機能

27.2 I²C インタフェースの構成

■ I²C インタフェースのブロックダイアグラム

図 27.2-1 に、I²C インタフェースのブロックダイアグラムを示します。

図 27.2-1 I²C インタフェースのブロックダイアグラム



■ I²C インタフェースに関する端子

I²C インタフェースに関する端子は、SDA データ入出力端子と SCL クロック入出力端子があります。SCL/SDA 端子は、汎用入出力ポート (P76/SCL, P77/SDA) と I²C インタフェースを兼用しています。

汎用入出力ポートとして使用する場合は、P76, P77 のポート方向レジスタ (DDR) が存在しませんので、常にポートとしてデータを有効にします (MB90485 シリーズのみ)。

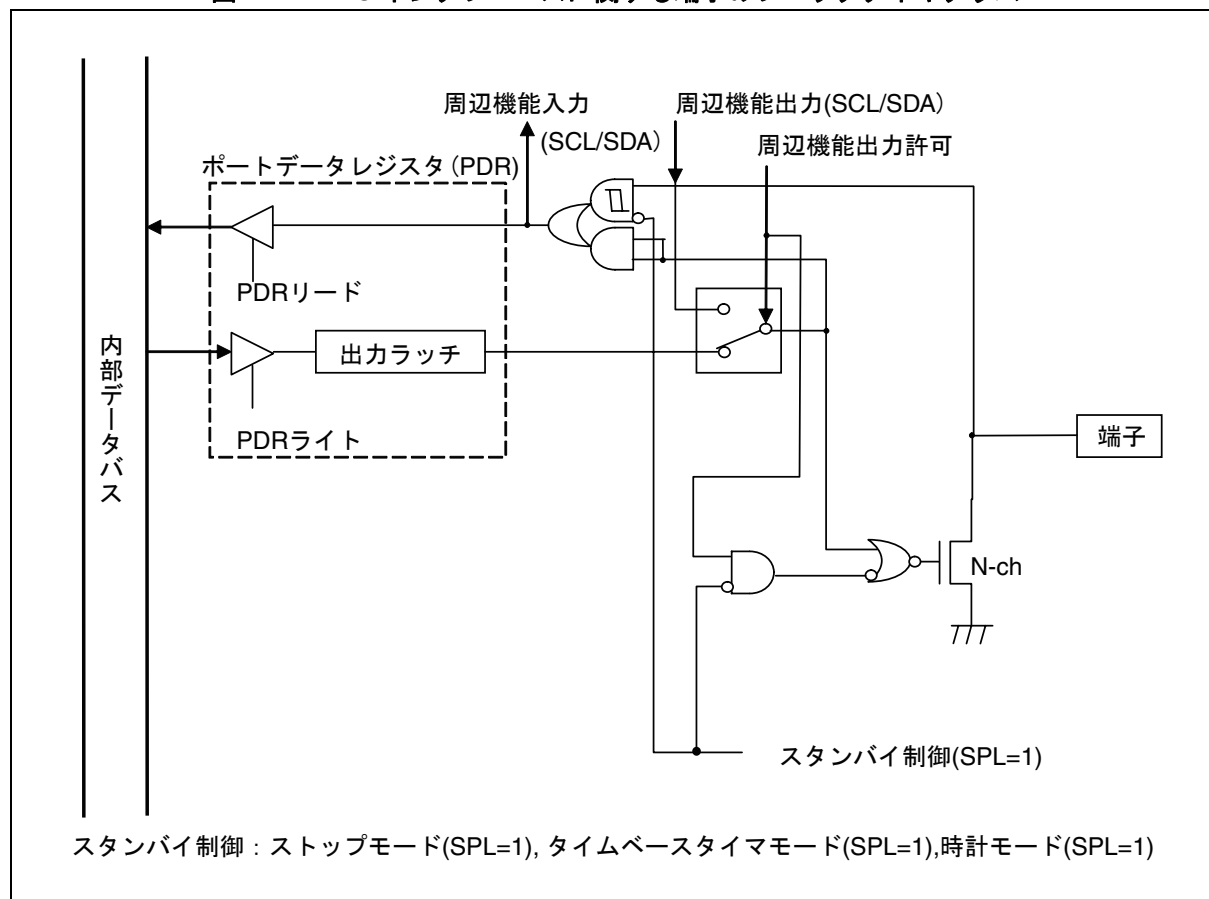
MB90485 シリーズでは、P76/SCL, P77/SDA 端子は、N-ch オープンドレイン端子です。

● SCL/SDA 端子として使用する場合の設定

I²C インタフェースとして使用する場合は、ポートデータレジスタ (PDR7 :bit14, 15 → "1") を設定してください。また、入力ポートとして使用する場合は、外部端子には、プルアップ抵抗を付加する必要があります。

■ I²C インタフェースに関する端子のブロックダイヤグラム

図 27.2-2 I²C インタフェースに関する端子のブロックダイヤグラム



27.3 I²C インタフェースのレジスタの構成と機能

I²C インタフェースのレジスタの構成と機能について説明します。

■ I²C インタフェースのレジスタ一覧

● バステータスレジスタ (IBSR)

バステータスレジスタ アドレス : 000088 _H	7	6	5	4	3	2	1	0	↩ ビットNo.
	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	IBSR
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)	

● バスコントロールレジスタ (IBCR)

バスコントロールレジスタ アドレス : 000089 _H	15	14	13	12	11	10	9	8	↩ ビットNo.
	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR
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)	

● クロックコントロールレジスタ (ICCR)

クロックコントロールレジスタ アドレス : 00008A _H	7	6	5	4	3	2	1	0	↩ ビットNo.
	—	—	EN	CS4	CS3	CS2	CS1	CS0	ICCR
リード/ライト	⇒ (—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値	⇒ (—)	(—)	(0)	(X)	(X)	(X)	(X)	(X)	

● アドレスレジスタ (IADR)

アドレスレジスタ アドレス : 00008B _H	15	14	13	12	11	10	9	8	↩ ビットNo.
	—	A6	A5	A4	A3	A2	A1	A0	IADR
リード/ライト	⇒ (—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値	⇒ (—)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

● データレジスタ (IDAR)

データレジスタ アドレス : 00008C _H	7	6	5	4	3	2	1	0	↩ ビットNo.
	D7	D6	D5	D4	D3	D2	D1	D0	IDAR
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)	

27.3.1 バステータスレジスタ (IBSR)

バステータスレジスタ (IBSR) の構成と機能について説明します。

■ バステータスレジスタ (IBSR)

下図に、バステータスレジスタ (IBSR) のビット構成を示します。

バステータスレジスタ アドレス : 000088 _H	7	6	5	4	3	2	1	0	↔ ビットNo.
	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	IBSR
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)	

以下に、バステータスレジスタ (IBSR) の各ビットの機能を説明します。

[bit7] BB:Bus Busy

このビットは、I²C バスの状態を示します。

0	ストップコンディションを検出した。
1	スタートコンディションを検出した (バスは使用されている)。

[bit6] RSC:Repeated Start Condition

このビットは、繰返しスタートコンディションの検出を示します。

0	繰返しスタートコンディションは検出されていない。
1	バス使用中に、再びスタートコンディションを検出した。

このビットは、INT ビットへの "0" の書き込み時、スレーブ時にアドレスされなかった場合、バス停止中のスタートコンディション検出またはストップコンディション検出時にクリアされます。

[bit5] AL:Arbitration Lost

このビットは、アービトレーションロストの検出を示します。

0	アービトレーションロスト検出されていない。
1	マスタ送信中にアービトレーションロストが発生した。または、ほかのシステムがバスを使用中のときに MSS ビットに "1" を書き込んだ。

このビットは、INT ビットへの "0" の書き込みによってクリアされます。

[bit4] LRB:Last Received ビット

このビットは、アクノリッジ格納用です。受信側からのアクノリッジを格納します。

0	受信を確認した。
1	受信が確認されない。

このビットは、スタートコンディションまたはストップコンディションの検出によってクリアされます。

[bit3] TRX:Transfer/Receive

このビットは、データ転送の送受信状態を示します。

0	受信状態
1	送信状態

[bit2] AAS:Addressed As Slave

このビットは、アドレッシングの検出を示します。

0	スレーブ時に、アドレッシングされていない。
1	スレーブ時に、アドレッシングされた。

このビットは、スタートコンディションまたはストップコンディション検出時にクリアされます。

[bit1] GCA:General Call Address

このビットは、ゼネラルコールアドレス (00_H) の検出を示します。

0	スレーブ時に、ゼネラルコールアドレスは受信されていない。
1	スレーブ時に、ゼネラルコールアドレスを受信した。

このビットは、スタートコンディションまたはストップコンディション検出時にクリアされます。

[bit0] FBT:First Byte Transfer

このビットは、第 1 バイト (アドレスデータ) の検出を示します。

0	受信データが第 1 バイト以外である。
1	受信データが第 1 バイト (アドレスデータ) である。

このビットは、スタートコンディションの検出で "1" にセットされても、INT ビットの "0" の書込みまたはスレーブ時にアドレスされなかった場合にクリアされます。

27.3.2 バスコントロールレジスタ (IBCR)

バスコントロールレジスタ (IBCR) の構成と機能について説明します。

■ バスコントロールレジスタ (IBCR)

下図に、バスコントロールレジスタ (IBCR) のビット構成を示します。

バスコントロールレジスタ アドレス：000089 _H	15	14	13	12	11	10	9	8	← ビットNo.
	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR
	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)	

以下に、バスコントロールレジスタ (IBCR) の各ビットの機能を説明します。

[bit15] BER:Bus ERRor

このビットは、バスエラー割込み要求フラグです。書込み時と読出し時で機能が異なります。

(書込み時)

0	バスエラー割込み要求フラグをクリアする。
1	関係なし。

(読出し時)

0	バスエラーは検出されていない。
1	データ転送中に、不正なスタート、ストップコンディションを検出した。

このビットがセットされた場合、ICCR レジスタの EN ビットはクリアされて I²C インタフェースは停止状態になり、データ転送は中断されます。

[bit14] BEIE:Bus Error Interrupt Enable

このビットは、バスエラー割込み許可を示します。

0	バスエラー割込み禁止。
1	バスエラー割込み許可。

このビットが "1" のとき BER ビットが "1" であれば、割込みを発生します。

[bit13] SCC:Start Condition Continue

このビットは、スタートコンディションの発生を示します。

(書込み時)

0	関係なし。
1	マスタ転送時に再びスタートコンディションを発生させます。

このビットの読出し値は、常に "0" です。

[bit12] MSS:Master Slave Select

このビットは、マスタ / スレーブの選択を示します。

0	ストップコンディションを発生します。転送終了後にスレーブモードとなります。
1	マスタモードとなり、スタートコンディションを発生し転送を開始します。

このビットは、マスタ送信中にアービトレーションロストが発生した場合にクリアされ、スレーブモードになります。

<注意事項>

下記条件で使用する場合、スレーブとして受信できませんのでゼネラルコールアドレスの送信は禁止です。

- 本 LSI のほかにマスタモードとなるほかの LSI がバス上に存在し、本 LSI がマスタとしてゼネラルコールアドレスを送信し第 2 バイト目以降でアービトレーションロストが発生する場合

[bit11] ACK:ACKnowledge

このビットは、データを受信した場合のアクノリッジ発生許可を示します。

0	アクノリッジを発生しない。
1	アクノリッジを発生する。

このビットは、スレーブにおけるアドレスデータ受信時には、無効となります。

[bit10] GCAA:General Call Address Acknowledge

このビットは、ゼネラルコールアドレス (00_H) を受信した場合のアクノリッジ発生許可を示します。

0	アクノリッジを発生しない。
1	アクノリッジを発生する。

[bit9] INTE:INTerrupt Enable

このビットは、割込み許可を示します。

0	割込み禁止
1	割込み許可

このビットが "1" のとき INT ビットが "1" であれば、割込みを発生します。

[bit8] INT:INTerrupt

このビットは、転送終了割込み要求フラグです。

(書込み時)

0	転送終了割込み要求フラグをクリアする。
1	関係なし

(読出し時)

0	転送が終了していない。
1	アクノリッジビットを含めた 1 バイト転送が終了したときに、次の条件に該当する場合にセットされます。 <ul style="list-style-type: none"> ・バスマスタである。 ・アドレスされたスレーブである。 ・ゼネラルコールアドレスを受信した。 ・アービトレーションロストが起こった。 ・ほかのシステムがバスを使用中にスタートコンディションを発生しようとした。

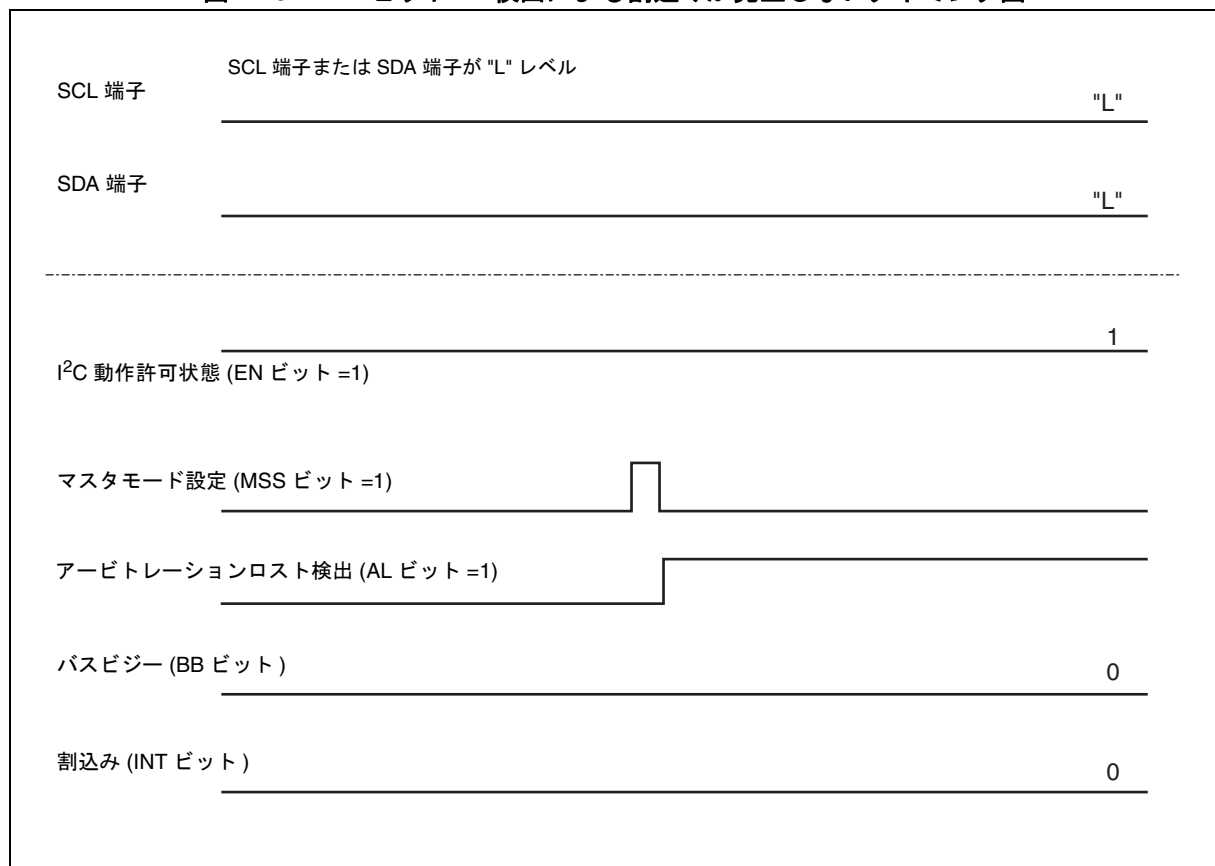
このビットが "1" のとき、SCL ラインは "L" レベルに保たれます。このビットへ "0" を書き込むことによって、SCL ラインが開放され次バイトが転送されます。また、マスタ時にスタートコンディションまたはストップコンディションの発生によって "0" にリセットされます。

<注意事項>

図 27.3-1 と図 27.3-2 に示したタイミングで、スタートコンディションを発生させる命令を実行 (MSS ビットに "1" を設定) すると、アービトレーションロスト検出 (AL ビット =1) による割込み (INT ビット =1) が発生しません。

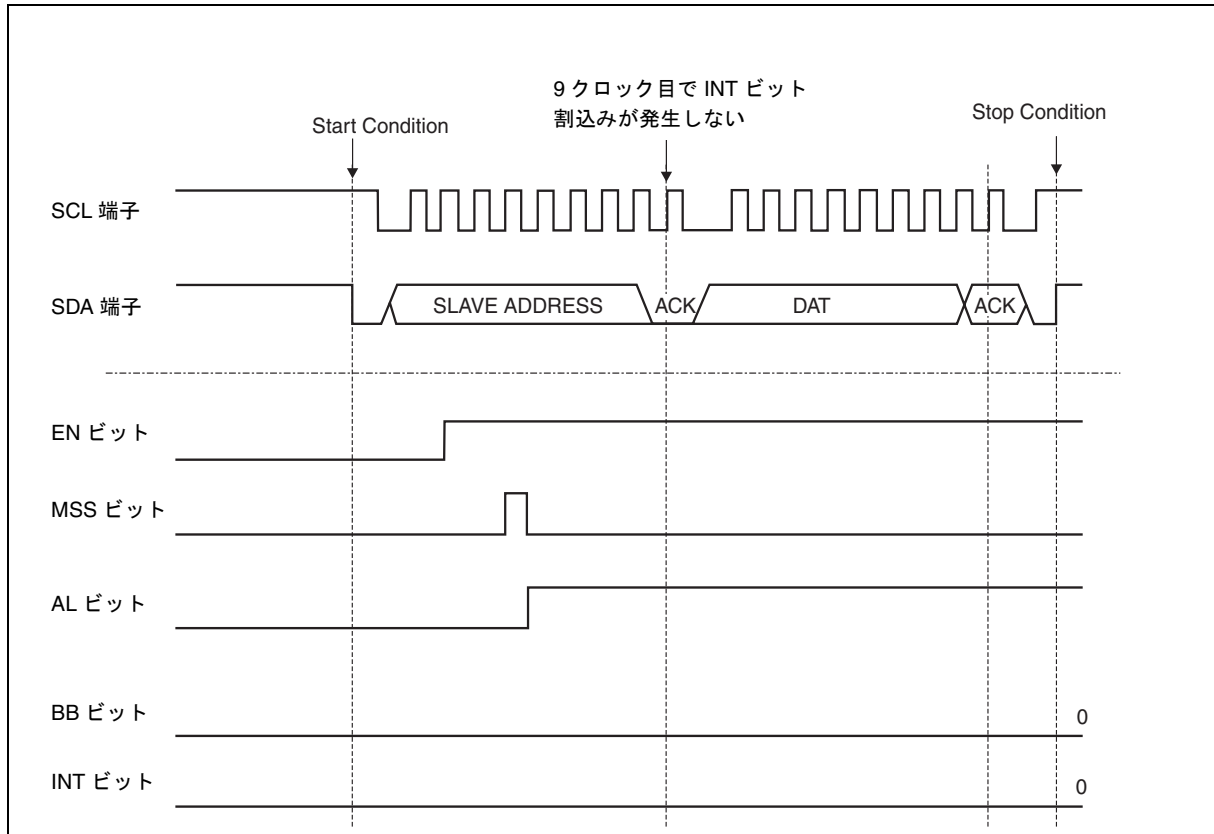
- AL ビット =1 検出による割込み (INT ビット =1) が発生しない条件 1
スタートコンディション未検出 (BB ビット =0) 状態で、SDA 端子または SCL 端子レベルが "L" のときに、スタートコンディションを発生させる命令を実行 (IBCR レジスタの MSS ビットに "1" を設定) した場合。

図 27.3-1 AL ビット =1 検出による割込みが発生しないタイミング図



- AL ビット=1 検出による割込み (INT ビット=1) が発生しない条件 2
他のマスタに I²C バスが占有されている状態で, I²C 動作許可 (EN ビット=1) をしてスタートコンディションを発生させる命令を実行 (IBCR レジスタの MSS ビットに "1" を設定) した場合。
図 27.3-2 に示したとおり, I²C が動作禁止 (EN ビット=0) の状態で, I²C バス上のほかのマスタが通信を開始すると, スタートコンディション未検出 (BB ビット=0) で I²C バスが占有されている状態になるためです。

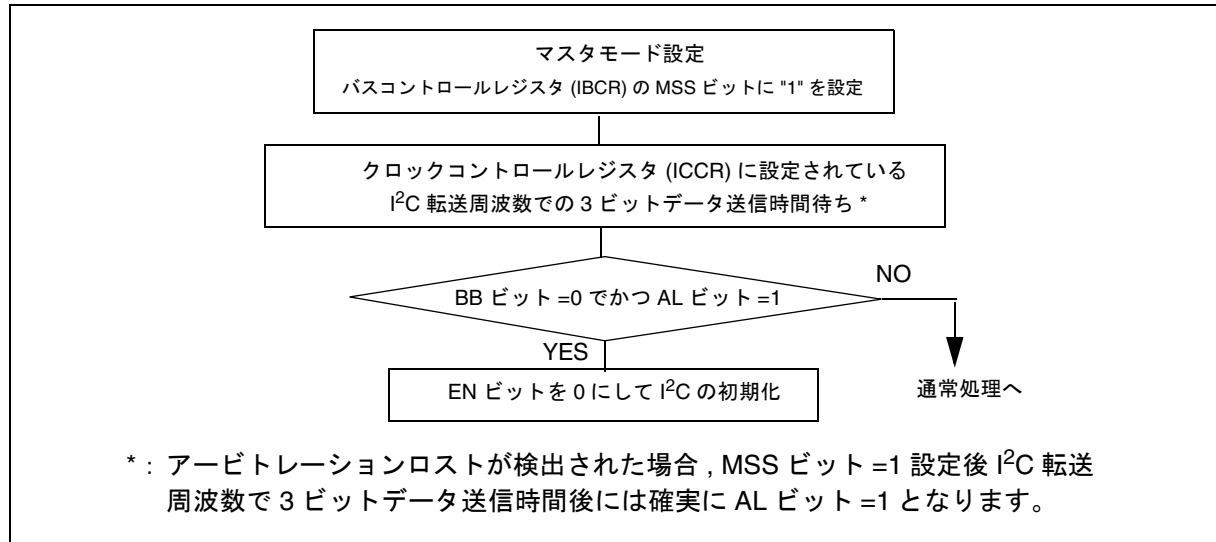
図 27.3-2 AL ビット=1 検出による割込みが発生しないタイミング図



上記のような現象が発生する可能性がある場合には, 以下の手順でソフトウェアによる処理を行ってください。

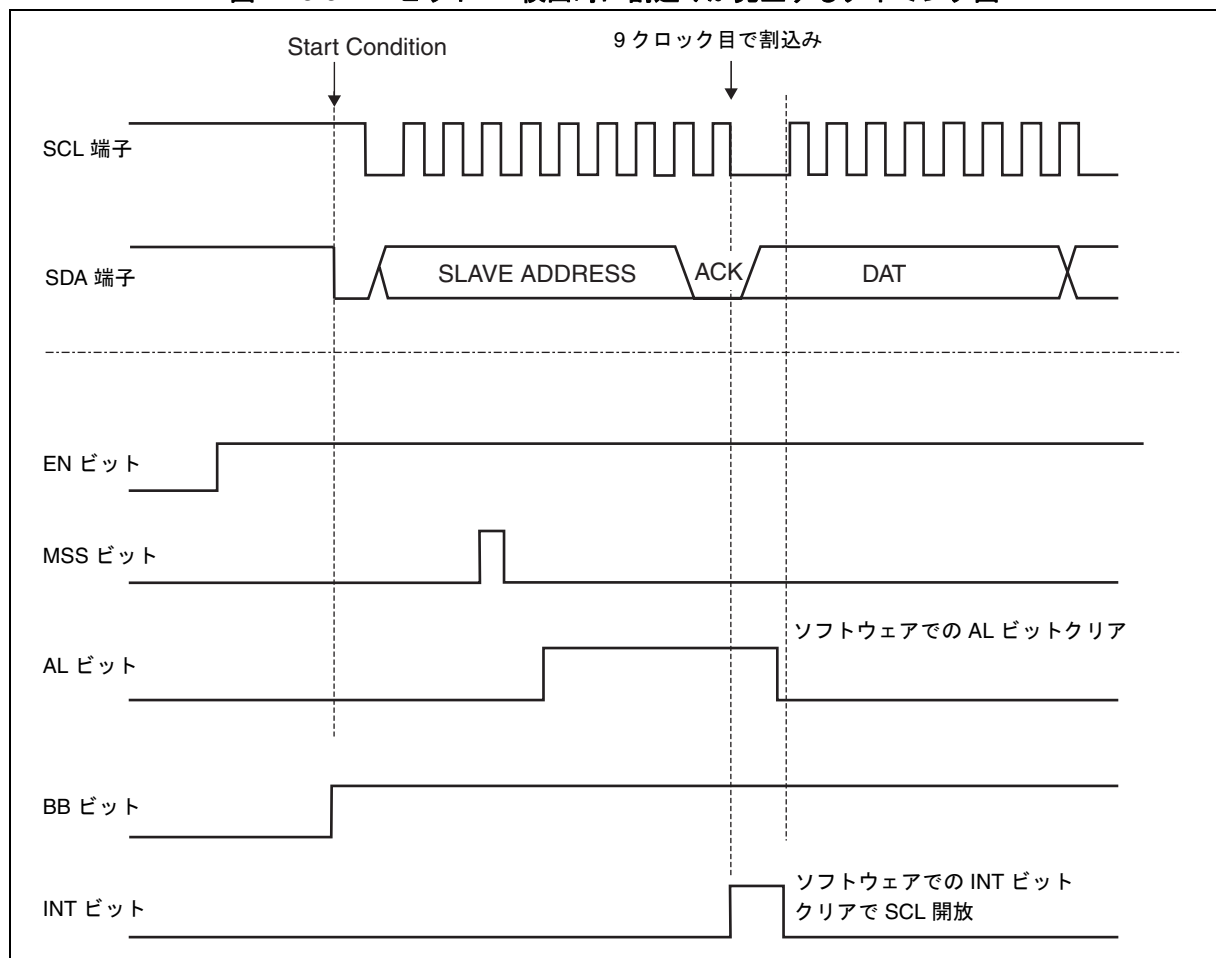
- スタートコンディションを発生させる命令を実行 (MSS ビットに "1" を設定)。
- タイマ機能などを用いて, ICCR レジスタに設定されている I²C 転送周波数で 3 ビットデータ送信時間分待つ。
例: I²C 転送周波数 100kHz の場合
3 ビットデータ送信時間 $\{1/(100 \times 10^3)\} \times 3 = 30 \mu\text{s}$
- IBSR レジスタの AL ビットと BB ビットを確認し, AL ビット=1, BB ビット=0 の場合には, ICCR レジスタの EN ビットを "0" にして I²C を初期化する。AL ビットと BB ビットがその他の状態の場合は通常処理を行う。

以下にフロー例を示します。



- AL ビット = 1 検出による割込み (INT ビット = 1) が発生する例
バスビジー検出 (BB ビット = 1) されている状態で, スタートコンディションを発生させる命令を実行 (MSS ビットに "1" を設定) しアービトレーションロストした場合には, AL ビット = 1 検出時に INT ビット割込みが発生します。

図 27.3-3 AL ビット = 1 検出時に割込みが発生するタイミング図



■ バスコントロールレジスタ (IBCR) 使用上の注意

SCC ビットに "1", MSS ビットに "0" の同時書込みは、禁止です。

また、INT ビットと MSS または SCC ビットの同時書込みにより、次バイト転送、スタートコンディション発生およびストップコンディション発生の競合が起こります。この場合の優先度は、次のようになります。

- 次のバイト転送とストップコンディション発生
INT ビットに "0", MSS ビットに "0" を書き込むと、MSS ビットの "0" の書込みが優先され、ストップコンディションが発生します。
- 次のバイト転送とスタートコンディション発生
INT ビットに "0", SCC ビットに "1" を書き込むと、SCC ビットの "1" の書込みが優先され、スタートコンディションが発生します。

27.3.3 クロックコントロールレジスタ (ICCR)

クロックコントロールレジスタ (ICCR) の構成と機能について説明します。

■ クロックコントロールレジスタ (ICCR)

下図に、クロックコントロールレジスタ (ICCR) のビット構成を示します。

アドレス : 00008A _H	7	6	5	4	3	2	1	0	ビットNo.
クロックコントロールレジスタ	—	—	EN	CS4	CS3	CS2	CS1	CS0	ICCR
リード/ライト	⌵	⌵	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値	⌵	⌵	(0)	(X)	(X)	(X)	(X)	(X)	

以下に、クロックコントロールレジスタ (ICCR) の機能を説明します。

[bit7, bit6] 未使用

これらのビットは、未使用です。

[bit5] EN:ENable

このビットは、I²C インタフェースの動作許可を示します。

0	動作禁止
1	動作許可

- このビットが "0" のとき、IBSR レジスタおよび IBCR レジスタ (BER, BEIE ビットを除く) の各ビットがクリアされます。
- BER ビットがセットされると、このビットはクリアされます。

[bit4 ~ bit0] CS4 ~ CS0:Clock Period Select 4-0

これらのビットは、シリアルクロックの周波数を設定します。シフトクロックの周波数 f_{sck} は、次式で求められます。

m, n は、CS4-0 に対して表 27.3-1 に示すようになります。

$$f_{sck} = \frac{\phi}{m \times n + 4}$$

ϕ : マシンクロック

表 27.3-1 シリアルクロック周波数設定

m	CS4	CS3
5	0	0
6	0	1
7	1	0
8	1	1

n	CS2	CS1	CS0
4	0	0	0
8	0	0	1
16	0	1	0
32	0	1	1
64	1	0	0
128	1	0	1
256	1	1	0
512	1	1	1

<注意事項>

f_{sck} を求める式の分母の+4のサイクルは、SCL端子の出力レベルが変化したことをチェックするための最小のオーバーヘッドです。SCL 端子の立上りのディレイが大きい場合や、スレーブデバイスでクロックを引き延ばしている場合には、+4 のサイクルより大きくなります。なお、シリアルクロックの周波数は 100kHz 以上に設定しないでください。

27.3.4 アドレスレジスタ (IADR)

アドレスレジスタ (IADR) の構成と機能について説明します。

■ アドレスレジスタ (IADR)

下図に , アドレスレジスタ (IADR) のビット構成を示します。

アドレスレジスタ アドレス : 00008B _H	15	14	13	12	11	10	9	8	↩ビットNo.
	—	A6	A5	A4	A3	A2	A1	A0	IADR
リード/ライト ↷	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	---
初期値 ↷	(—)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

[bit15] 未使用

このビットは未使用です。

[bit14 ~ bit8] A6 ~ A0

これらのビットは , スレーブアドレスを示します。本レジスタはスレーブアドレスを指定するためのレジスタです。スレーブ時 , アドレスデータ受信後に IDAR レジスタとの比較が行われ , 一致している場合はマスタに対してアクノリッジを送信します。

27.3.5 データレジスタ (IDAR)

データレジスタ (IDAR) の構成と機能について説明します。

■ データレジスタ (IDAR)

下図に、データレジスタ (IDAR) のビット構成を示します。

データレジスタ アドレス : 00008C _H	7	6	5	4	3	2	1	0	↵ ビットNo.
	D7	D6	D5	D4	D3	D2	D1	D0	IDAR
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)	

[bit7 ~ bit0] D7 ~ D0

これらのビットは、データビットです。

本レジスタは、シリアル転送用のデータレジスタとして使用します。データは **MSB** から転送されます。データ受信時 (TRX=0) は、データレジスタのデータ出力値は "1" になります。

このレジスタの書込み側はダブルバッファになっており、バスが使用中 (BB=1) である場合は、書込みデータはシリアル転送用のレジスタにロードされます。読出し時は、シリアル転送用のレジスタを直接読み出すため、受信データは **IBCR** レジスタの **INT** ビットがセットされている場合のみ有効です。

27.4 I²C インタフェースの割込み

I²C インタフェースに関する割込みは、データの転送終了時に発生します。

■ I²C インタフェースの割込み制御ビットと割込み要因

I²C インタフェースの割込み制御ビットと割込み要因を下記の表に示します。

	I ² C 割込み
割込み要求フラグ	IBCR:INT (bit8)
割込み要求出力許可ビット	IBCR:INTE (bit9)
割込み発生要因	I ² C 転送終了

■ I²C インタフェースに関する割込み要因

I²C バスには 1 つの割込み要因しかありません。1 バイト転送終了した時点で割込み条件に該当していた場合、割込みが発生します。

1 つの割込みで、複数の割込み条件を判断しなければならないため、割込みルーチン内で、各フラグをチェックしなければなりません。1 バイト転送終了時点での割込み条件を下記に示します。

- バスマスタであった場合
- アドレスされたスレーブであった場合
- ゼネラルコールアドレスを受信した場合
- アービトレーションロストが生じた場合

■ I²C インタフェースの割込みと DMA 転送および EI²OS

表 27.4-1 に、ソフトウェア割込みを除く割込み要因と割込みベクタおよび割込み制御レジスタの関係を示します。

表 27.4-1 割込み要因と割込みベクタ / 割込み制御レジスタ

割込み要因	EI ² OS の クリア	μDMAC チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
I ² C インタフェース (MB90485 シリーズのみ) *	×	×	#39	FFFF60 _H	ICR14	0000BE _H

×：割込み要求フラグはクリアされません。

*： この割込み要因は、他の周辺機能の割込み要因と割込み番号を共有しています。詳細につきましては、表 3.2-2 を参照してください。

(注意事項) 同一割込み番号に 2 つの割込み要因があった場合は、リソースは両方の割込み要求フラグがクリアされます。したがって、2 要因のどちらか 1 つが EI²OS/μDMAC 機能を使用すると、もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして、ソフトウェアポーリング処理で対処してください。

■ DMA 転送、および EI²OS 機能への対応

I²C インタフェースは DMA 転送機能、および EI²OS 機能に対応していません。

27.5 I²C インタフェースの動作

I²C バスは、シリアルデータライン (SDA) とシリアルクロックライン (SCL) の 2 本の双方向バスラインを使用して通信を実行します。I²C インタフェースには、これらに対応する 2 本のオープンドレイン入出力端子 (SDA/SCL) を有し、ワイヤード論理を使用することができます。

■ スタートコンディション

バスが開放されている状態 (BB=0, MSS=0) で MSS ビットに "1" を書き込むと、I²C インタフェースはマスタモードとなり、同時にスタートコンディションを発生します。マスタモードでは、バスが使用状態 (BB=1) であっても、SCC ビットに "1" を書き込むことによって、再びスタートコンディションを発生させることができます。スタートコンディションを発生させる条件として、次の 2 通りがあります。

- バスが使用されていない状態 (MSS=0 & BB=0 & INT=0 & AL=0) での MSS ビットへの "1" の書込み。
- バスマスタ時の割込み状態 (MSS=1 & BB=1 & INT=1 & AL=0) での SCC ビットへの "1" の書込み。

ほかのシステム (アイドル状態中) がバス使用中に、MSS ビットへ "1" を書き込むと、AL ビットが "1" にセットされます。

上記条件以外では MSS ビット、SCC ビットへの "1" の書込みは無視されます。

■ ストップコンディション

マスタモード (MSS=1) 時に、MSS ビットに "0" を書き込むと、ストップコンディションを発生し、スレーブモードになります。ストップコンディションを発生させる条件は、次のとおりです。

- バスマスタ時の割込み状態 (MSS=1 & BB=1 & INT=1 & AL=0) での MSS ビットへの "0" の書込み。

上記条件以外では、MSS ビットへの "0" の書込みは無視されます。

<注意事項>

MSS ビットに "0" を書き込んでからストップコンディションを発生するまでに時間を要します。ストップコンディションを発生前に I²C インタフェースを動作禁止 (EN=0:ICCR) にすると、直ちに動作が停止して SCL ラインに不正クロックが発生します。この場合、I²C バスは好ましくない状態に置かれることがあります。

I²C インタフェースを動作禁止する際には、ストップコンディションの発生を確認 (BB=0:IBSR) した後、動作を禁止 (EN=0:ICCR) してください。

■ アドレッシング

マスタモードでは、スタートコンディション発生後、BB=1, TRX=1 にセットされ、IDAR レジスタの内容が MSB から出力されます。アドレスデータ送信後、スレーブからアクノリッジを受信すると、送信データのビット 0(送信後の IDAR レジスタの bit0) を反転して、TRX ビットへ格納します。

スレーブモードでは、スタートコンディション発生後、BB=1, TRX=0 にセットされ、マスタからの送信データが IDAR レジスタで受信されます。アドレスデータ受信後、IDAR レジスタと IADR レジスタとの比較が行われ、一致している場合、AAS=1 にセットしてマスタに対してアクノリッジを送信します。その後、受信データの bit0(受信後の IDAR レジスタの bit0) を TRX ビットへ格納します。

■ アービトレーション

マスタ送信時に、ほかのマスタも同時にデータを送信している場合、アービトレーションが起こります。自分の送信データが "1", SDA ライン上のデータが "L" レベルの場合、自分はアービトレーションを失ったとみなし、AL=1 にセットします。また、前述のようにバスが使用状態のときに、スタートコンディションを発生させようとした場合も AL=1 にセットされます。AL=1 にセットされると、MSS=0, TRX=0 となり、スレーブ受信モードとなります。

■ アクノリッジ

アクノリッジは、受信側が送信側に対して送信します。データ受信時は、ACK ビットによってアクノリッジの有無を選択できます。データ送信時は、受信側からのアクノリッジが LRB ビットに格納されます。

スレーブ送信側に、マスタ受信側からアクノリッジを受信しなかった場合、TRX=0 となりスレーブ受信モードになります。これによって、マスタはスレーブが SCL ラインを開放したときに、ストップコンディションを発生させることができます。

■ バスエラー

以下の条件が成立した場合はバスエラーと判断され、I²C インタフェースは停止状態になります。

- データ転送中 (ACK ビットを含む) の I²C バス上の基本規定違反の検出
- マスタ時のストップコンディション検出
- バスアイドルにおける I²C バス上の基本規定違反の検出

■ その他

● アービトレーションロスト検出後の処理

アービトレーションロストした後、自分がアドレッシングされている場合と、アドレッシングされていなかった場合をソフトウェアで判断しなければなりません。

アービトレーションロストするとハードウェア的にスレーブになりますが、1 バイト転送終了後、CLK ライン、DATA ライン双方を "L" に設定されます。このため、アドレッシングされていない場合は、即座に CLK ライン、DATA ラインを開放し、アドレッシングされていた場合には、スレーブ送信、またはスレーブ受信の準備をしてから、CLK ライン、DATA ラインを開放する必要があります(すべてソフトウェアで処理する必要があります)。

● アービトレーションロスト検出時の割込み要因

アービトレーションロストを検出した場合、即時に割込み要因が立つのではなく、1 バイト転送終了した時点で割込み要因が立ちます。

アービトレーションロストを検出するとハードウェア的にスレーブになりますが、スレーブになっても、割込み要因を立てるために、計 9 クロックを出力します。したがって、即時に割込み要因が立たないため、アービトレーションロスト後に即時に割込み処理を行うことはできません。

● 割込み条件

I²C バスには 1 つの割込みしかありません。1 バイト転送終了した時点で、割込み条件に該当していた場合に、割込みが発生します。

1 つの割込みで、複数の割込み条件を判断しなければならないため、割込みルーチン内で、各フラグをチェックしなければなりません。1 バイト転送終了時点での割込み条件を下記に示します。

- バスマスタであった場合
- アドレスされたスレーブであった場合
- ゼネラルコールアドレスを受信した場合
- アービトレーションロストが生じた場合

● 転送速度

I²C バスは、最大でシリアルクロック周波数 100kHz の転送までしか行えません。

メモリマップおよび F²MC-16LX に使用している命令について説明します。

付録 A メモリマップ

付録 B I/O マップ

付録 C 割込み要因と割込みベクタ，割込み制御レジスタ

付録 D 命令

付録 A メモリマップ

メモリ空間は、3つのモードに分かれます。

■ メモリ空間

メモリ空間は、図 A-1 に示すように3つのモードにそれぞれ依存して分かります。

図 A-1 メモリマップ

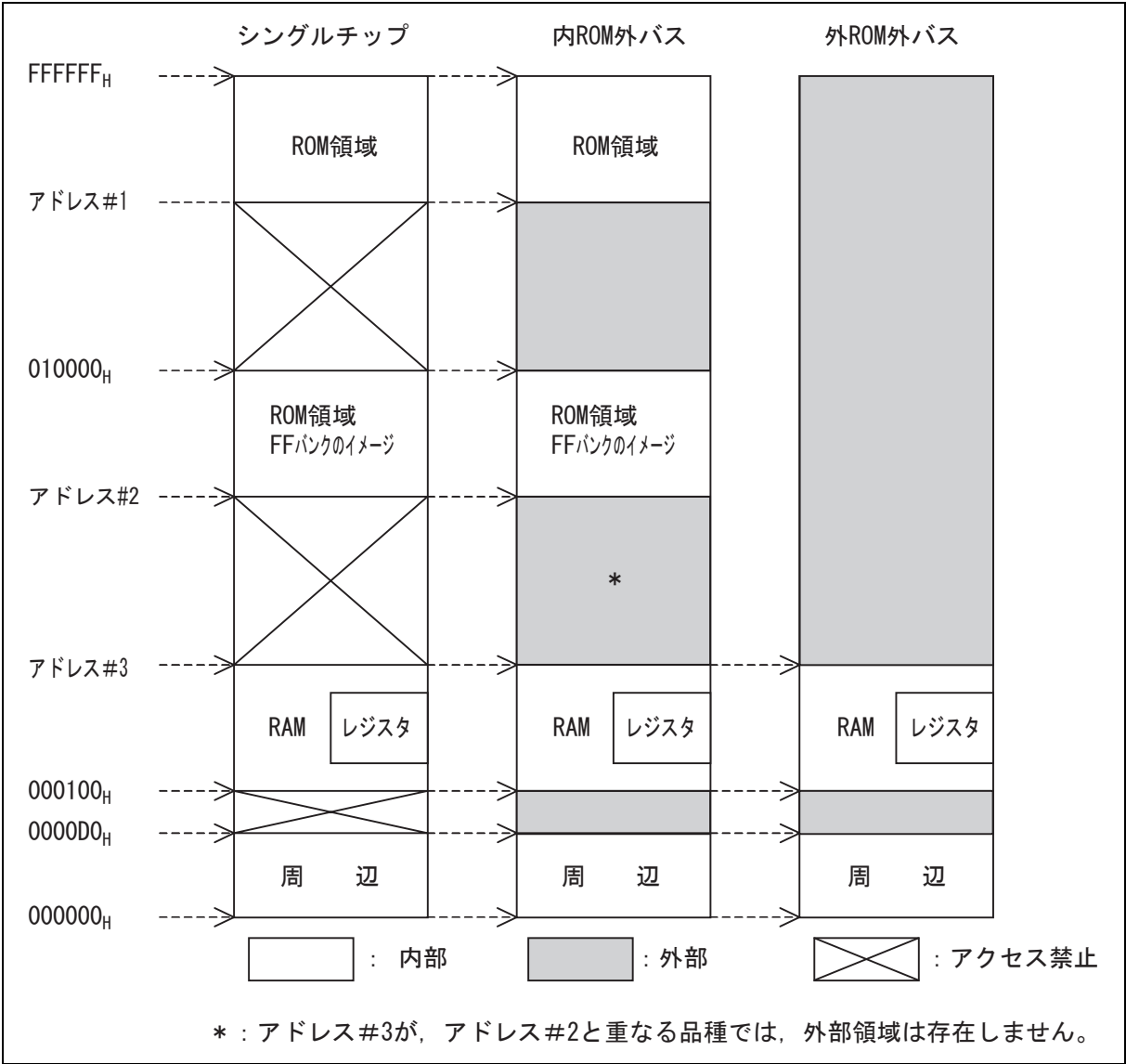


表 A-1 に、各品種のアドレス #1, アドレス #2 およびアドレス #3 の対応を示します。

表 A-1 各品種のアドレス #1, アドレス #2 およびアドレス #3 の対応

品種	アドレス #1	アドレス #2	アドレス #3
MB90F481B	FC0000 _H	ROMM レジスタの MS ビットにより 004000 _H , 008000 _H を選択可能	001100 _H
MB90F482B	FC0000 _H		001900 _H
MB90487B	FD0000 _H		002900 _H
MB90488B	FC0000 _H		002900 _H
MB90F488B	FC0000 _H		002900 _H
MB90V480B	(FC0000 _H)		004000 _H
MB90V485B	(FC0000 _H)		004000 _H
MB90483C	FB0000 _H ^{*1}		004000 _H
MB90F489B	F90000 _H ^{*2}	008000 _H 固定	006100 _H ^{*3}

*1: MB90483Cは、図 A-3 のようにF8バンク～FAバンクおよびFCバンクが、シングルチップモードおよび内 ROM 外バスモードで、アクセスなしとなります。

*2: MB90F489Bは、図 A-2 のようにF8バンクおよびFCバンクが、シングルチップモードおよび内 ROM 外バスモードで、アクセスなしとなります。

*3: MB90F489Bは、搭載RAM領域がMB90V485Bよりも大きいため、004000_Hより大きい領域のエミュレーションは、ツール側でエミュレーションメモリエリアの設定を行う必要があります。

<注意事項>

MB90F481B は、FC0000_H ～ FC7FFF_H, FE0000_H ～ FE7FFF_H が 00 バンクの上に、FF バンクの ROM がイメージで見えるようになっていますが、これは C コンパイラのスモールモデルを有効に生かすためです。下位 16 ビットは同じになるようにしてありますので、ポインタの宣言において far 指定をしなくても ROM 内のテーブルを参照することができます。例えば、00C000_H をアクセスした場合に、実際には、FFC000_H の ROM の内容がアクセスされることになります。ここで、ROMM レジスタの MS ビットが "0" の場合 FF バンクの ROM 領域は、48K バイトを超えるので、00 バンクのイメージにすべての領域をみせることはできません。そこで、FF4000_H ～ FFFFFFF_H までのイメージを 00 バンクに見えるようにし、FF0000_H ～ FF3FFF_H は FF バンクだけに見えるようにしてあります。

図 A-2 に MB90F489B のメモリマップを示します。

図 A-2 MB90F489B メモリマップ

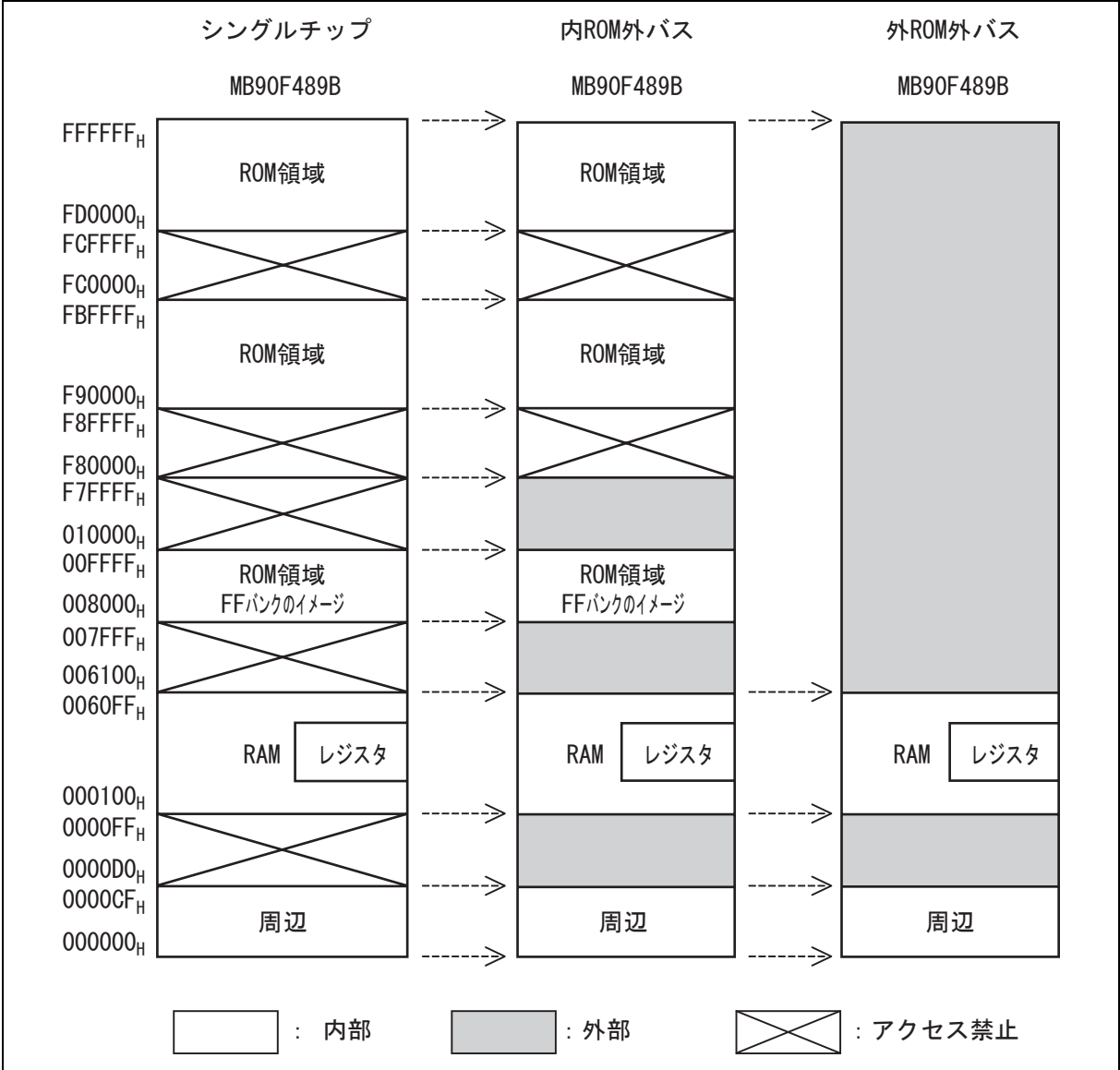
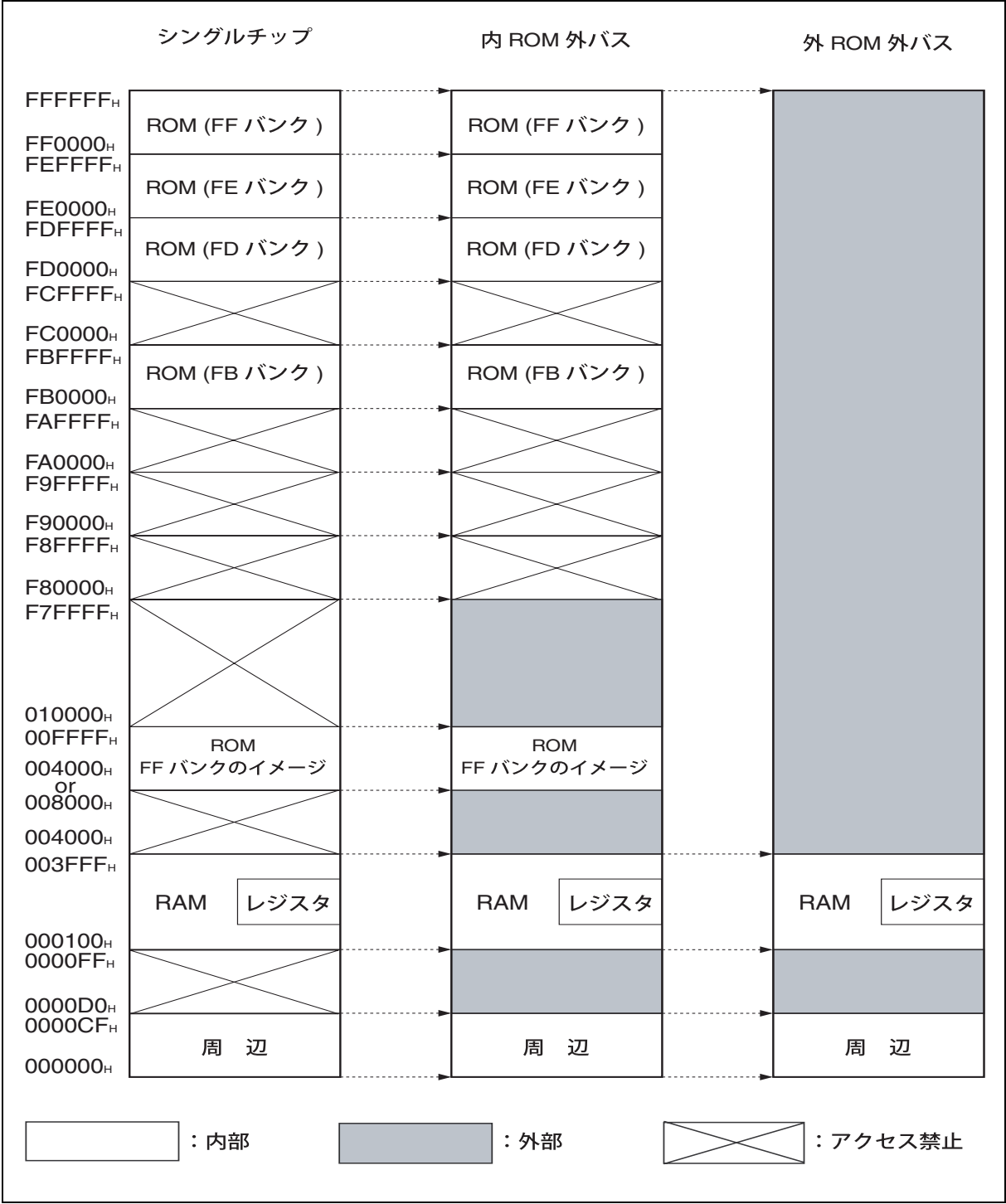


図 A-3 に MB90483C のメモリマップを示します。

図 A-3 MB90483C メモリマップ



付録 B I/O マップ

表 B-1 に、各周辺機能のレジスタに割り当てられるアドレスを示します。

■ I/O マップ

表 B-1 I/O マップ (1 / 6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
00 _H	ポート 0 データレジスタ	PDR0	R/W	ポート 0	XXXXXXXX _B
01 _H	ポート 1 データレジスタ	PDR1	R/W	ポート 1	XXXXXXXX _B
02 _H	ポート 2 データレジスタ	PDR2	R/W	ポート 2	XXXXXXXX _B
03 _H	ポート 3 データレジスタ	PDR3	R/W	ポート 3	XXXXXXXX _B
04 _H	ポート 4 データレジスタ	PDR4	R/W	ポート 4	XXXXXXXX _B
05 _H	ポート 5 データレジスタ	PDR5	R/W	ポート 5	XXXXXXXX _B
06 _H	ポート 6 データレジスタ	PDR6	R/W	ポート 6	XXXXXXXX _B
07 _H	ポート 7 データレジスタ	PDR7	R/W	ポート 7	XXXXXXXX _B (MB90480 シリーズ) 11XXXXXXXX _B (MB90485 シリーズ)
08 _H	ポート 8 データレジスタ	PDR8	R/W	ポート 8	XXXXXXXX _B
09 _H	ポート 9 データレジスタ	PDR9	R/W	ポート 9	XXXXXXXX _B
0A _H	ポート A データレジスタ	PDRA	R/W	ポート A	----XXXX _B
0B _H	アップダウンタイマ入力許可レジスタ	UDER	R/W	アップダウン タイマ入力制御	XX000000 _B
0C _H	割込み /DTP 許可レジスタ	ENIR	R/W	DTP/ 外部割込み	00000000 _B
0D _H	割込み /DTP 要因レジスタ	EIRR	R/W		XXXXXXXX _B
0E _H	要求レベル設定レジスタ	ELVR	R/W		00000000 _B
0F _H	要求レベル設定レジスタ		R/W		00000000 _B
10 _H	ポート 0 方向レジスタ	DDR0	R/W	ポート 0	00000000 _B
11 _H	ポート 1 方向レジスタ	DDR1	R/W	ポート 1	00000000 _B
12 _H	ポート 2 方向レジスタ	DDR2	R/W	ポート 2	00000000 _B
13 _H	ポート 3 方向レジスタ	DDR3	R/W	ポート 3	00000000 _B
14 _H	ポート 4 方向レジスタ	DDR4	R/W	ポート 4	00000000 _B
15 _H	ポート 5 方向レジスタ	DDR5	R/W	ポート 5	00000000 _B
16 _H	ポート 6 方向レジスタ	DDR6	R/W	ポート 6	00000000 _B
17 _H	ポート 7 方向レジスタ	DDR7	R/W	ポート 7	00000000 _B (MB90480 シリーズ) XX000000 _B (MB90485 シリーズ)
18 _H	ポート 8 方向レジスタ	DDR8	R/W	ポート 8	00000000 _B
19 _H	ポート 9 方向レジスタ	DDR9	R/W	ポート 9	00000000 _B
1A _H	ポート A 方向レジスタ	DDRA	R/W	ポート A	----0000 _B
1B _H	ポート 4 出力端子レジスタ	ODR4	R/W	ポート 4 (オープンドレイン制 御)	00000000 _B

表 B-1 I/O マップ (2/6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
1C _H	ポート 0 入力抵抗レジスタ	RDR0	R/W	ポート 0 (プルアップ抵抗制御)	00000000 _B
1D _H	ポート 1 入力抵抗レジスタ	RDR1	R/W	ポート 1 (プルアップ抵抗制御)	00000000 _B
1E _H	ポート 7 出力端子レジスタ	ODR7	R/W	ポート 7 (オープンドレイン 制御)	00000000 _B (MB90480 シリーズ) XX000000 _B (MB90485 シリーズ)
1F _H	アナログ入力許可レジスタ	ADER	R/W	ポート 6, A/D	11111111 _B
20 _H	シリアルモードレジスタ	SMR	R/W	UART	00000X00 _B
21 _H	シリアルコントロールレジスタ	SCR	W, R/W		00000100 _B
22 _H	シリアルインプット / アウトプットレジスタ	SIDR/ SODR	R/W		XXXXXXXX _B
23 _H	シリアルステータスレジスタ	SSR	R, R/W		00001000 _B
24 _H	予約領域				
25 _H	通信プリスケラコントロールレジスタ	CDCR	R/W	通信プリスケラ (UART)	00--0000 _B
26 _H	シリアルモードコントロールステータスレジスタ 0	SMCS0	R, R/W	SIO1(ch.0)	----0000 _B
27 _H	シリアルモードコントロールステータスレジスタ 0	SMCS0	R, R/W		00000010 _B
28 _H	シリアルデータレジスタ 0	SDR0	R/W		XXXXXXXX _B
29 _H	通信プリスケラコントロールレジスタ 0	SDCR0	R/W	通信プリスケラ SIO1(ch.0)	0---0000 _B
2A _H	シリアルモードコントロールステータスレジスタ 1	SMCS1	R, R/W	SIO2(ch.1)	----0000 _B
2B _H	シリアルモードコントロールステータスレジスタ 1	SMCS1	R, R/W		00000010 _B
2C _H	シリアルデータレジスタ 1	SDR1	R/W		XXXXXXXX _B
2D _H	通信プリスケラコントロールレジスタ 1	SDCR1	R/W	通信プリスケラ SIO2(ch.1)	0---0000 _B
2E _H	リロードレジスタ L(ch.0)	PRL0	R/W	8/16 ビット PPG (ch.0 ～ ch.5)	XXXXXXXX _B
2F _H	リロードレジスタ H(ch.0)	PRLH0	R/W		XXXXXXXX _B
30 _H	リロードレジスタ L(ch.1)	PRL1	R/W		XXXXXXXX _B
31 _H	リロードレジスタ H(ch.1)	PRLH1	R/W		XXXXXXXX _B
32 _H	リロードレジスタ L(ch.2)	PRL2	R/W		XXXXXXXX _B
33 _H	リロードレジスタ H(ch.2)	PRLH2	R/W		XXXXXXXX _B
34 _H	リロードレジスタ L(ch.3)	PRL3	R/W		XXXXXXXX _B
35 _H	リロードレジスタ H(ch.3)	PRLH3	R/W		XXXXXXXX _B
36 _H	リロードレジスタ L(ch.4)	PRL4	R/W		XXXXXXXX _B
37 _H	リロードレジスタ H(ch.4)	PRLH4	R/W		XXXXXXXX _B
38 _H	リロードレジスタ L(ch.5)	PRL5	R/W		XXXXXXXX _B
39 _H	リロードレジスタ H(ch.5)	PRLH5	R/W		XXXXXXXX _B
3A _H	PPG0 動作モード制御レジスタ	PPGC0	R/W	8/16 ビット PPG	0X000XX1 _B
3B _H	PPG1 動作モード制御レジスタ	PPGC1	R/W		0X000001 _B
3C _H	PPG2 動作モード制御レジスタ	PPGC2	R/W		0X000XX1 _B
3D _H	PPG3 動作モード制御レジスタ	PPGC3	R/W		0X000001 _B
3E _H	PPG4 動作モード制御レジスタ	PPGC4	R/W		0X000XX1 _B
3F _H	PPG5 動作モード制御レジスタ	PPGC5	R/W		0X000001 _B
40 _H	PPG0, 1 出力制御レジスタ	PPG01	R/W	8/16 ビット PPG	00000000 _B
41 _H	予約領域				
42 _H	PPG2, 3 出力制御レジスタ	PPG23	R/W	8/16 ビット PPG	00000000 _B

表 B-1 I/O マップ (3 / 6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
43 _H	予約領域				
44 _H	PPG4, 5 出力制御レジスタ	PPG45	R/W	8/16 ビット PPG	00000000 _B
45 _H	予約領域				
46 _H	コントロールステータスレジスタ	ADCS1	R/W	A/D コンバータ	00000000 _B
47 _H		ADCS2	W, R/W		00000000 _B
48 _H	データレジスタ	ADCR1	R		XXXXXXXXXX _B
49 _H		ADCR2	W, R		00000XXX _B
4A _H	アウトプットコンペアレジスタ (ch.0) 下位	OCCP0	R/W	16 ビット 入出力タイマ アウトプット コンペア (ch.0 ～ ch.5)	00000000 _B
4B _H	アウトプットコンペアレジスタ (ch.0) 上位				00000000 _B
4C _H	アウトプットコンペアレジスタ (ch.1) 下位	OCCP1	R/W		00000000 _B
4D _H	アウトプットコンペアレジスタ (ch.1) 上位				00000000 _B
4E _H	アウトプットコンペアレジスタ (ch.2) 下位	OCCP2	R/W		00000000 _B
4F _H	アウトプットコンペアレジスタ (ch.2) 上位				00000000 _B
50 _H	アウトプットコンペアレジスタ (ch.3) 下位	OCCP3	R/W		00000000 _B
51 _H	アウトプットコンペアレジスタ (ch.3) 上位				00000000 _B
52 _H	アウトプットコンペアレジスタ (ch.4) 下位	OCCP4	R/W		00000000 _B
53 _H	アウトプットコンペアレジスタ (ch.4) 上位				00000000 _B
54 _H	アウトプットコンペアレジスタ (ch.5) 下位	OCCP5	R/W		00000000 _B
55 _H	アウトプットコンペアレジスタ (ch.5) 上位				00000000 _B
56 _H	アウトプットコンペアコントロールレジスタ (ch.0, ch.1) 下位	OCS01	R/W		0000--00 _B
57 _H	アウトプットコンペアコントロールレジスタ (ch.0, ch.1) 上位		R/W		---00000 _B
58 _H	アウトプットコンペアコントロールレジスタ (ch.2, ch.3) 下位	OCS23	R/W		0000--00 _B
59 _H	アウトプットコンペアコントロールレジスタ (ch.2, ch.3) 上位		R/W		---00000 _B
5A _H	アウトプットコンペアコントロールレジスタ (ch.4, ch.5) 下位	OCS45	R/W		0000--00 _B
5B _H	アウトプットコンペアコントロールレジスタ (ch.4, ch.5) 上位		R/W		---00000 _B
5C _H	インプットキャプチャデータレジスタ (ch.0) 下位	IPCP0	R	16 ビット入出力 タイマ インプット キャプチャ (ch.0, ch.1)	XXXXXXXXXX _B
5D _H	インプットキャプチャデータレジスタ (ch.0) 上位		R		XXXXXXXXXX _B
5E _H	インプットキャプチャデータレジスタ (ch.1) 下位	IPCP1	R		XXXXXXXXXX _B
5F _H	インプットキャプチャデータレジスタ (ch.1) 上位		R		XXXXXXXXXX _B
60 _H	インプットキャプチャコントロールステータスレジスタ	ICS01	R/W		00000000 _B
61 _H	予約領域				
62 _H	タイマカウンタデータレジスタ 下位	TCDT	R/W	16 ビット入出力 タイマ フリーランタイマ	00000000 _B
63 _H	タイマカウンタデータレジスタ 上位	TCDT	R/W		00000000 _B
64 _H	タイマカウンタコントロールステータスレジスタ	TCCS	R/W		00000000 _B
65 _H	タイマカウンタコントロールステータスレジスタ	TCCS	R/W		0--00000 _B
66 _H	コンペアクリアレジスタ下位	CPCLR	R/W		XXXXXXXXXX _B
67 _H	コンペアクリアレジスタ上位				XXXXXXXXXX _B
68 _H	アップダウンカウントレジスタ (ch.0)	UDCR0	R	8/16 ビットアップ ダウンタイマ カウンタ	00000000 _B
69 _H	アップダウンカウントレジスタ (ch.1)	UDCR1	R		00000000 _B
6A _H	リロード / コンペアレジスタ (ch.0)	RCR0	W		00000000 _B
6B _H	リロード / コンペアレジスタ (ch.1)	RCR1	W		00000000 _B
6C _H	カウンタコントロールレジスタ (ch.0) 下位	CCRL0	W, R/W		0X00X000 _B
6D _H	カウンタコントロールレジスタ (ch.0) 上位	CCRH0	R/W		00000000 _B

表 B-1 I/O マップ (4/6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
6E _H	予約領域				
6F _H	ROM ミラー機能選択レジスタ	ROMM	R/W	ROM ミラー機能	-----+1 _B
70 _H	カウンタコントロールレジスタ (ch.1) 下位	CCRL1	W, R/W	8/16 ビットアップ ダウンタイム カウンタ	0X00X000 _B
71 _H	カウンタコントロールレジスタ (ch.1) 上位	CCRH1	R/W		-0000000 _B
72 _H	カウンタステータスレジスタ (ch.0)	CSR0	R, R/W		00000000 _B
73 _H	予約領域				
74 _H	カウンタステータスレジスタ (ch.1)	CSR1	R, R/W	8/16 ビット UDC	00000000 _B
75 _H	予約領域				
76 _H ^{*1}	PWC コントロール / ステータスレジスタ	PWCSR0	R, R/W	PWC タイマ (ch.0)	00000000 _B
77 _H ^{*1}					0000000X _B
78 _H ^{*1}	PWC データバッファレジスタ	PWCR0	R/W		00000000 _B
79 _H ^{*1}					00000000 _B
7A _H ^{*1}	PWC コントロール / ステータスレジスタ	PWCSR1	R, R/W	PWC タイマ (ch.1)	00000000 _B
7B _H ^{*1}					0000000X _B
7C _H ^{*1}	PWC データバッファレジスタ	PWCR1	R/W		00000000 _B
7D _H ^{*1}					00000000 _B
7E _H ^{*1}	PWC コントロール / ステータスレジスタ	PWCSR2	R, R/W	PWC タイマ (ch.2)	00000000 _B
7F _H ^{*1}					0000000X _B
80 _H ^{*1}	PWC データバッファレジスタ	PWCR2	R/W		00000000 _B
81 _H ^{*1}					00000000 _B
82 _H ^{*1}	分周比制御レジスタ	DIVR0	R/W	PWC (ch.0)	-----00 _B
83 _H	予約領域				
84 _H ^{*1}	分周比制御レジスタ	DIVR1	R/W	PWC (ch.1)	-----00 _B
85 _H	予約領域				
86 _H ^{*1}	分周比制御レジスタ	DIVR2	R/W	PWC (ch.2)	-----00 _B
87 _H	予約領域				
88 _H ^{*1}	バスステータスレジスタ	IBSR	R	I ² C	00000000 _B
89 _H ^{*1}	バスコントロールレジスタ	IBCR	R/W		00000000 _B
8A _H ^{*1}	クロックコントロールレジスタ	ICCR	R/W		--0XXXXX _B
8B _H ^{*1}	アドレスレジスタ	IADR	R/W		-XXXXXXXX _B
8C _H ^{*1}	データレジスタ	IDAR	R/W		XXXXXXXXX _B
8D _H	予約領域				
8E _H ^{*1}	μPG コントロール / ステータスレジスタ	PGCSR	R/W	μPG	00000--- _B
8F _H ~ 9B _H	禁止領域				
9C _H	μDMAC ステータスレジスタ	DSRL	R/W	μDMAC	00000000 _B
9D _H	μDMAC ステータスレジスタ	DSRH	R/W	μDMAC	00000000 _B
9E _H	プログラムアドレス検出制御 ステータスレジスタ	PACSR	R/W	アドレス一致 検出機能	00000000 _B
9F _H	遅延割込み要因発生 / 解除レジスタ	DIRR	R/W	遅延割込発生 モジュール	-----0 _B
A0 _H	低消費電力モード制御レジスタ	LPMCR	W, R/W	低消費電力	00011000 _B
A1 _H	クロック選択レジスタ	CKSCR	R, R/W	低消費電力	11111100 _B
A2 _H , A3 _H	予約領域				
A4 _H	μDMAC ストップステータスレジスタ	DSSR	R/W	μDMAC	00000000 _B

表 B-1 I/O マップ (5/6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
A5 _H	自動レディ機能選択レジスタ	ARSR	W	外部端子	0011--00
A6 _H	外部アドレス出力制御レジスタ	HACR	W	外部端子	***** _B
A7 _H	バス制御信号選択レジスタ	EPCR	W	外部端子	1000*10- _B
A8 _H	ウォッチドッグタイマ制御レジスタ	WDTC	R, W	ウォッチドッグ タイマ	XXXXXX111 _B
A9 _H	タイムベースタイマ制御レジスタ	TBTC	W, R/W	タイムベース タイマ	1XX00100 _B
AA _H	時計タイマ制御レジスタ	WTC	R, R/W	時計タイマ	10001000 _B
AB _H	予約領域				
AC _H	μDMAC イネーブルレジスタ	DERL	R/W	μDMAC	00000000 _B
AD _H	μDMAC イネーブルレジスタ	DERH	R/W	μDMAC	00000000 _B
AE _H	フラッシュメモリコントロールステータスレジスタ	FMCS	W, R/W	フラッシュメモリ I/F	000X0000 _B
AF _H	禁止領域				
B0 _H	割込み制御レジスタ 00	ICR00	W,R/W	割込みコントローラ	00000111 _B
B1 _H	割込み制御レジスタ 01	ICR01	W,R/W		00000111 _B
B2 _H	割込み制御レジスタ 02	ICR02	W,R/W		00000111 _B
B3 _H	割込み制御レジスタ 03	ICR03	W,R/W		00000111 _B
B4 _H	割込み制御レジスタ 04	ICR04	W,R/W		00000111 _B
B5 _H	割込み制御レジスタ 05	ICR05	W,R/W		00000111 _B
B6 _H	割込み制御レジスタ 06	ICR06	W,R/W		00000111 _B
B7 _H	割込み制御レジスタ 07	ICR07	W,R/W		00000111 _B
B8 _H	割込み制御レジスタ 08	ICR08	W,R/W		00000111 _B
B9 _H	割込み制御レジスタ 09	ICR09	W,R/W		00000111 _B
BA _H	割込み制御レジスタ 10	ICR10	W,R/W		00000111 _B
BB _H	割込み制御レジスタ 11	ICR11	W,R/W		00000111 _B
BC _H	割込み制御レジスタ 12	ICR12	W,R/W		00000111 _B
BD _H	割込み制御レジスタ 13	ICR13	W,R/W		00000111 _B
BE _H	割込み制御レジスタ 14	ICR14	W,R/W		00000111 _B
BF _H	割込み制御レジスタ 15	ICR15	W,R/W		00000111 _B
C0 _H	チップセレクト領域 MASK レジスタ 0	CMR0	R/W	チップセレクト 機能	00001111 _B
C1 _H	チップセレクト領域レジスタ 0	CAR0	R/W		11111111 _B
C2 _H	チップセレクト領域 MASK レジスタ 1	CMR1	R/W		00001111 _B
C3 _H	チップセレクト領域レジスタ 1	CAR1	R/W		11111111 _B
C4 _H	チップセレクト領域 MASK レジスタ 2	CMR2	R/W		00001111 _B
C5 _H	チップセレクト領域レジスタ 2	CAR2	R/W		11111111 _B
C6 _H	チップセレクト領域 MASK レジスタ 3	CMR3	R/W		00001111 _B
C7 _H	チップセレクト領域レジスタ 3	CAR3	R/W		11111111 _B
C8 _H	チップセレクトコントロールレジスタ	CSCR	R/W		---000* _B
C9 _H	チップセレクトアクティブレベルレジスタ	CALR	R/W		---0000 _B
CA _H	タイマ制御ステータスレジスタ	TMCSR	R/W	16 ビットリロード タイマ	00000000 _B
CB _H					---0000 _B
CC _H	16 ビットタイマレジスタ I	TMR/ TMRLR	R/W		XXXXXXXXX _B
CD _H	16 ビットリロードレジスタ				
CE _H	予約領域				
CF _H	PLL 出力選択レジスタ	PLLOS	W	低消費電力	-----00 _B
D0 _H ~ FF _H	外部領域				
100 _H ~ # _H	RAM 領域				

表 B-1 I/O マップ (6 / 6)

アドレス	レジスタ	略称	アクセス	リソース	初期値
1FF0 _H	プログラムアドレス検出レジスタ 0 (下位)	PADR0	R/W	アドレス一致検出 機能	XXXXXXXX _B
1FF1 _H	プログラムアドレス検出レジスタ 0 (中位)				
1FF2 _H	プログラムアドレス検出レジスタ 0 (上位)				
1FF3 _H	プログラムアドレス検出レジスタ 1 (下位)	PADR1	R/W	アドレス一致検出 機能	XXXXXXXX _B
1FF4 _H	プログラムアドレス検出レジスタ 1 (中位)				
1FF5 _H	プログラムアドレス検出レジスタ 1 (上位)				

*: MB90485 シリーズでのレジスタです。MB90480 シリーズでは「予約領域」になります。

(注意事項)

・ 書込みについて

R/W: リード・ライト可能

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 : このビットの初期値は, "0" です。

1 : このビットの初期値は, "1" です。

X : このビットの初期値は, 不定です。

- : このビットは, 利用されていません。

* : このビットの初期値は, "1" または "0" です。モード端子 (MD2, MD1, MD0 端子) によって異なります。

+ : このビットの初期値は, "1" または "0" です。デバイスの RAM 領域によって異なります。

付録 C 割込み要因と割込みベクタ，割込み制御レジスタ

表 C-1 に，割込み要因と割込みベクタおよび割込み制御レジスタとの対応を示します。

■ 割込み要因と割込みベクタ，割込み制御レジスタ

表 C-1 割込み要因と割込みベクタおよび割込み制御レジスタとの対応

割込み要因	EI ² OS のクリア	μDMA チャンネル 番号	割込みベクタ		割込み制御レジスタ	
			番号	アドレス	番号	アドレス
リセット	×		#08	FFFFDC _H	-	-
INT9 命令	×		#09	FFFFD8 _H	-	-
例外	×		#10	FFFFD4 _H	-	-
INT0 (IRQ0)	○	0	#11	FFFFD0 _H	ICR00	0000B0 _H
INT1 (IRQ1)	○	×	#12	FFFFCC _H		
INT2 (IRQ2)	○	×	#13	FFFFC8 _H	ICR01	0000B1 _H
INT3 (IRQ3)	○	×	#14	FFFFC4 _H		
INT4 (IRQ4)	○	×	#15	FFFFC0 _H	ICR02	0000B2 _H
INT5 (IRQ5)	○	×	#16	FFFFBC _H		
INT6 (IRQ6)	○	×	#17	FFFFB8 _H	ICR03	0000B3 _H
INT7 (IRQ7)	○	×	#18	FFFFB4 _H		
PWC1 (MB90485 シリーズのみ)	○	×	#19	FFFFB0 _H	ICR04	0000B4 _H
PWC2 (MB90485 シリーズのみ)	○	×	#20	FFFFAC _H		
PWC0 (MB90485 シリーズのみ)	○	1	#21	FFFFA8 _H	ICR05	0000B5 _H
PPG0/PPG1 カウンタポロー	×	2	#22	FFFFA4 _H		
PPG2/PPG3 カウンタポロー	×	3	#23	FFFFA0 _H	ICR06	0000B6 _H
PPG4/PPG5 カウンタポロー	×	4	#24	FFFF9C _H		
8/16 ビットアップダウンカウンタ / タイマ (ch.0, ch.1) コンペア / アンダフロー / オーバフロー / アップダウン反転	○	×	#25	FFFF98 _H	ICR07	0000B7 _H
インプットキャプチャ (ch.0) 取込み	○	5	#26	FFFF94 _H		
インプットキャプチャ (ch.1) 取込み	○	6	#27	FFFF90 _H	ICR08	0000B8 _H
アウトプットコンペア (ch.0) 一致	○	8	#28	FFFF8C _H		
アウトプットコンペア (ch.1) 一致	○	9	#29	FFFF88 _H	ICR09	0000B9 _H
アウトプットコンペア (ch.2) 一致	○	10	#30	FFFF84 _H		
アウトプットコンペア (ch.3) 一致	○	×	#31	FFFF80 _H	ICR10	0000BA _H
アウトプットコンペア (ch.4) 一致	○	×	#32	FFFF7C _H		
アウトプットコンペア (ch.5) 一致	○	×	#33	FFFF78 _H	ICR11	0000BB _H
UART 送信完了	○	11	#34	FFFF74 _H		
16 ビットフリーランタイム オーバフロー / 16 ビットリロードタイム アンダフロー *2	○	12	#35	FFFF70 _H	ICR12	0000BC _H
UART 受信完了	◎	7	#36	FFFF6C _H		
SIO1 (ch.0)	○	13	#37	FFFF68 _H	ICR13	0000BD _H
SIO2 (ch.1)	○	14	#38	FFFF64 _H		
I ² C インタフェース (MB90485 シリーズのみ)	×	×	#39	FFFF60 _H	ICR14	0000BE _H
A/D コンバータ	○	15	#40	FFFF5C _H		
フラッシュの書込み / 消去， タイムベースタイマ / 時計タイマ *1	×	×	#41	FFFF58 _H	ICR15	0000BF _H
遅延割込み発生モジュール	×	×	#42	FFFF54 _H		

×：割込み要求フラグはクリアされません。

○：割込み要求フラグはクリアされます。

◎：割込み要求フラグはクリアされます。ストップ要求あり。

*1：フラッシュ書込み / 消去とタイムベースタイマと時計タイマを同時には使用できませんので，注意してください。

*2：リロードタイムアンダフロー割込みを許可(TMCSR レジスタのINTEビット = 1)から禁止(TMCSR レジスタのINTEビット = 0)にする場合は，割込み制御レジスタ (ICR12) の IL2 ～ IL0 ビット = "111" として割込みを禁止にしてから INTE ビットに "0" を書き込んでください。

<注意事項>

同一割込み番号に 2 つの割込み要因があった場合は，リソースは両方の割込み要求フラグがクリアされます。したがって，2 つの要因のどちらか 1 つの EI²OS/μDMAC 機能を使用すると，もう 1 つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを "0" にして，ソフトウェアポーリング処理で対処してください。

付録 D 命令

F²MC-16LX に使用している命令について説明します。

- D.1 命令の種類
- D.2 アドレッシング
- D.3 直接アドレッシング
- D.4 間接アドレッシング
- D.5 実行サイクル数
- D.6 実効アドレスフィールド
- D.7 命令一覧表の読み方
- D.8 F²MC-16LX 命令一覧表
- D.9 命令マップ

D.1 命令の種類

F²MC-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

D.2 アドレッシング

F²MC-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

F²MC-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 D.2-1 に示します。

表 D.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

D.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 D.3-1 に例を示します。

図 D.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 D.3-1 に示します。

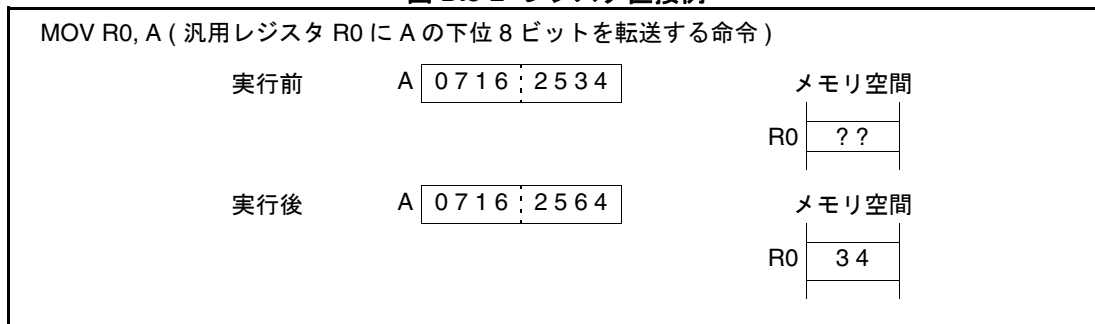
表 D.3-1 レジスタ直接

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 D.3-2 に例を示します。

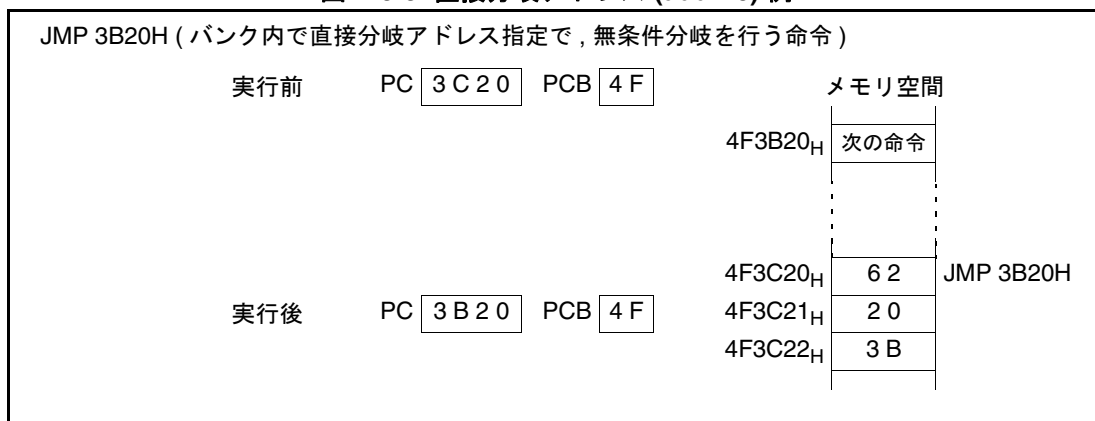
図 D.3-2 レジスタ直接例



● 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 D.3-3 に例を示します。

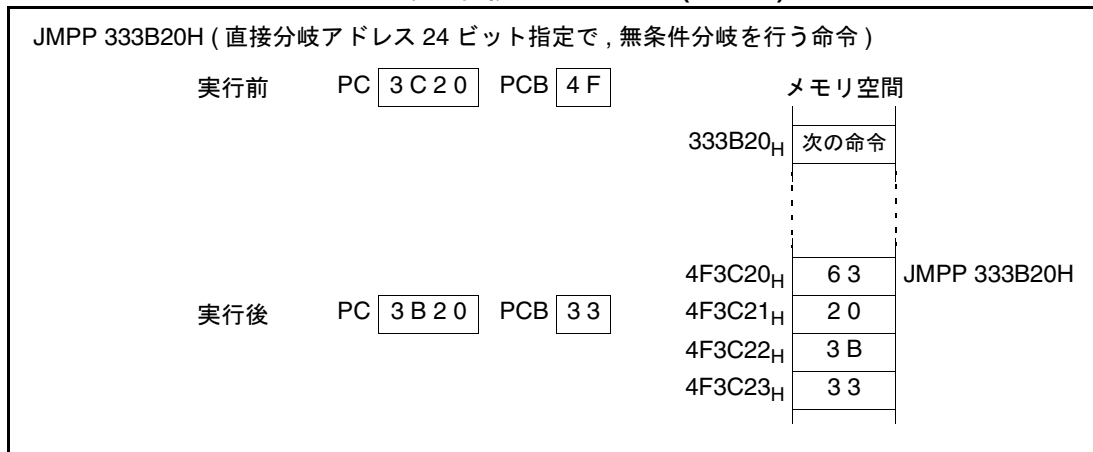
図 D.3-3 直接分岐アドレス (addr16) 例



● 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 D.3-4 に例を示します。

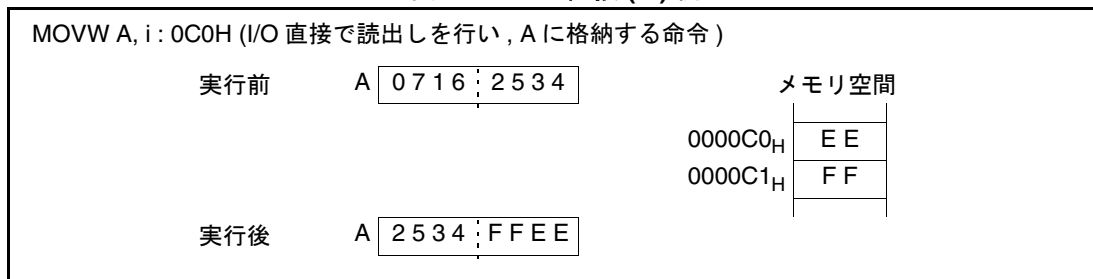
図 D.3-4 物理直接分岐アドレス (addr24) 例



● 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 "000000_H" ~ "0000FF_H" 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 D.3-5 に例を示します。

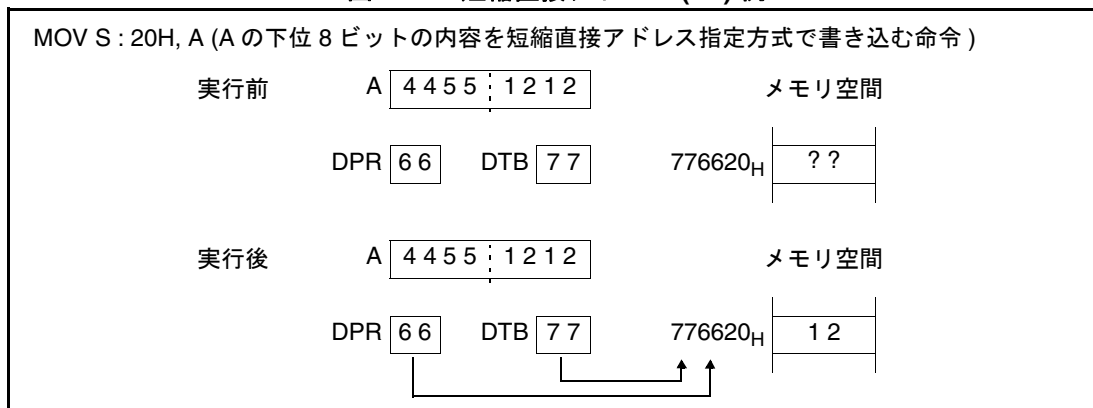
図 D.3-5 I/O 直接 (io) 例



● 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 D.3-6 に例を示します。

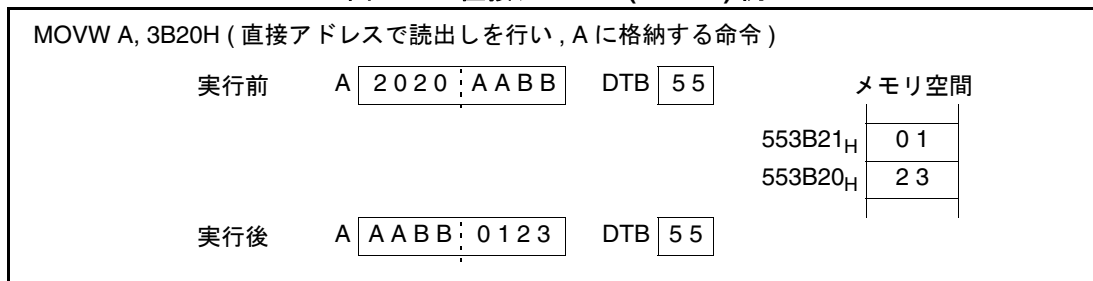
図 D.3-6 短縮直接アドレス (dir) 例



● 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 D.3-7 に例を示します。

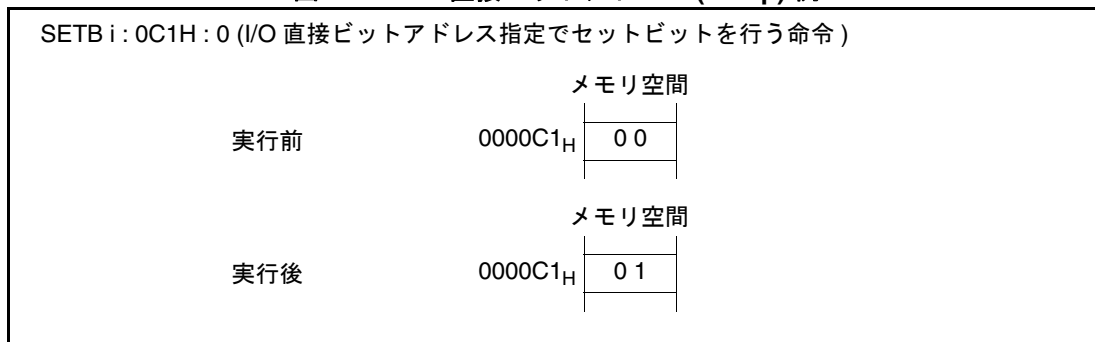
図 D.3-7 直接アドレス (addr16) 例



● I/O 直接ビットアドレス (io: bp)

物理アドレス "000000_H" ~ "0000FF_H" 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 D.3-8 に例を示します。

図 D.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 D.3-9 に例を示します。

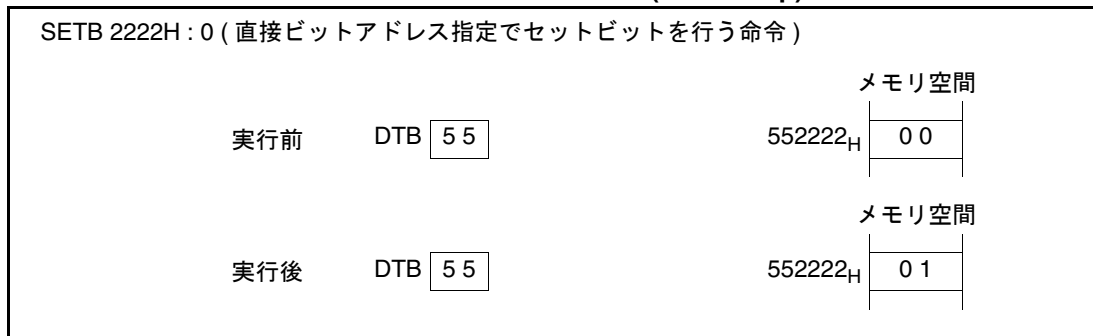
図 D.3-9 短縮直接ビットアドレス (dir: bp) 例



● 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 D.3-10 に例を示します。

図 D.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 D.3-11 に例を示します。

図 D.3-11 ベクタアドレス (#vct) 例

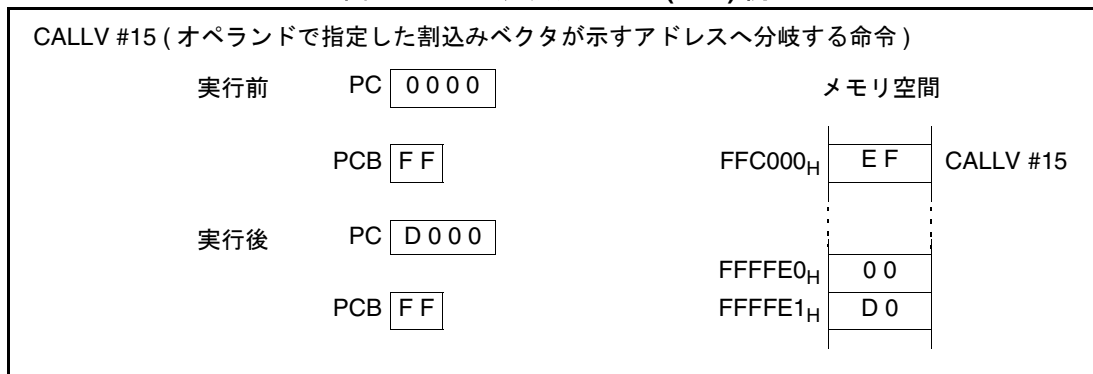


表 D.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE _H	XXFFFF _H
CALLV #1	XXFFFC _H	XXFFFD _H
CALLV #2	XXFFFA _H	XXFFFB _H
CALLV #3	XXFFF8 _H	XXFFF9 _H
CALLV #4	XXFFF6 _H	XXFFF7 _H
CALLV #5	XXFFF4 _H	XXFFF5 _H
CALLV #6	XXFFF2 _H	XXFFF3 _H
CALLV #7	XXFFF0 _H	XXXXF1 _H
CALLV #8	XXFFEE _H	XXFFEF _H
CALLV #9	XXFFEC _H	XXFFED _H
CALLV #10	XXFFEA _H	XXFFEB _H
CALLV #11	XXFFE8 _H	XXFFE9 _H
CALLV #12	XXFFE6 _H	XXFFE7 _H
CALLV #13	XXFFE4 _H	XXFFE5 _H
CALLV #14	XXFFE2 _H	XXFFE3 _H
CALLV #15	XXFFE0 _H	XXFFE1 _H

(注意事項) 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_H" 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 D.3-2 を参照してください)

D.4 間接アドレッシング

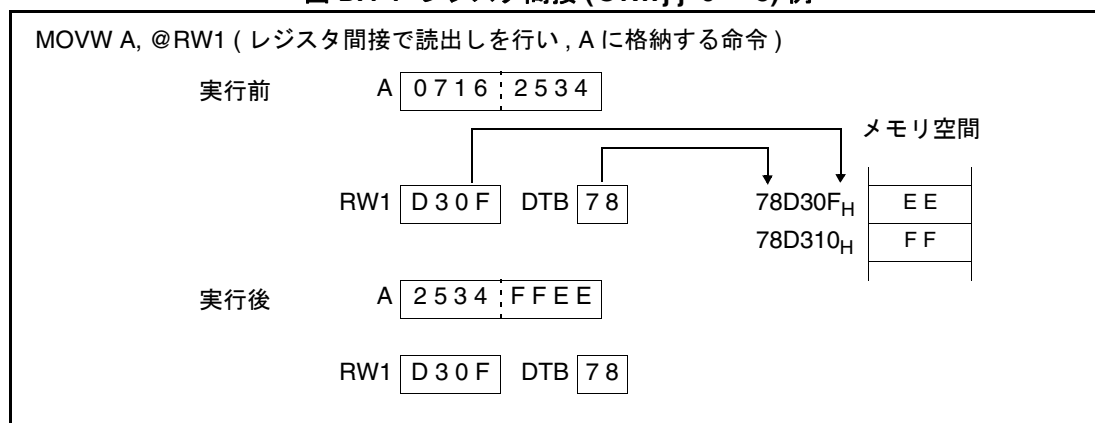
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

■ 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 D.4-1 に例を示します。

図 D.4-1 レジスタ間接 (@RWj j=0 ~ 3) 例



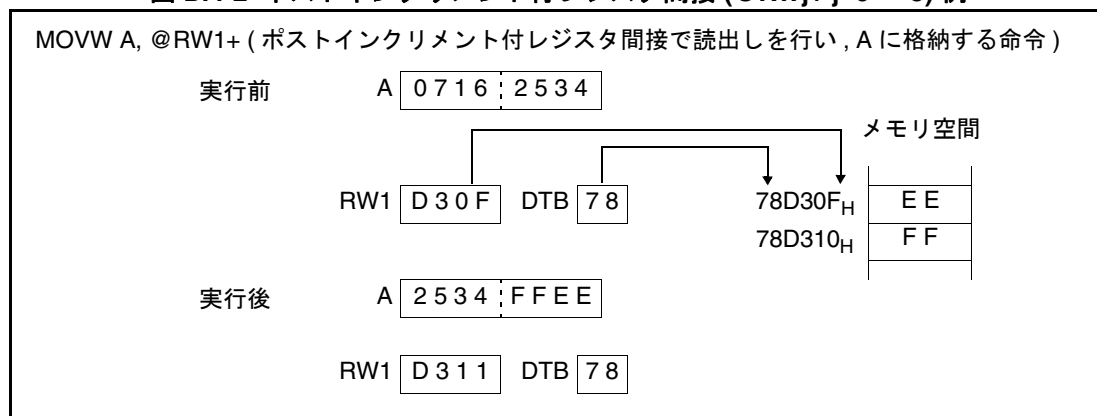
● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

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、命令が書き込みだった場合は命令による書き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き込みデータとなります。

図 D.4-2 に例を示します。

図 D.4-2 ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3) 例

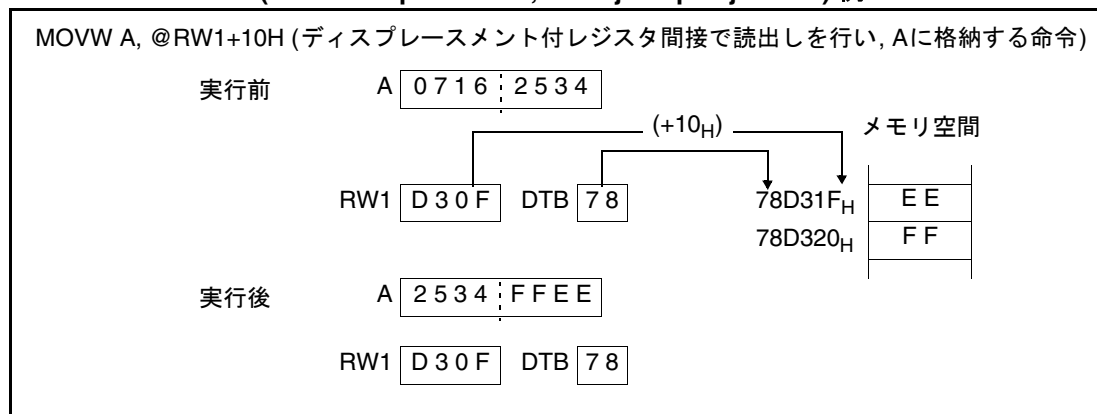


● ディスプレースメント付レジスタ間接

(@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 D.4-3 に例を示します。

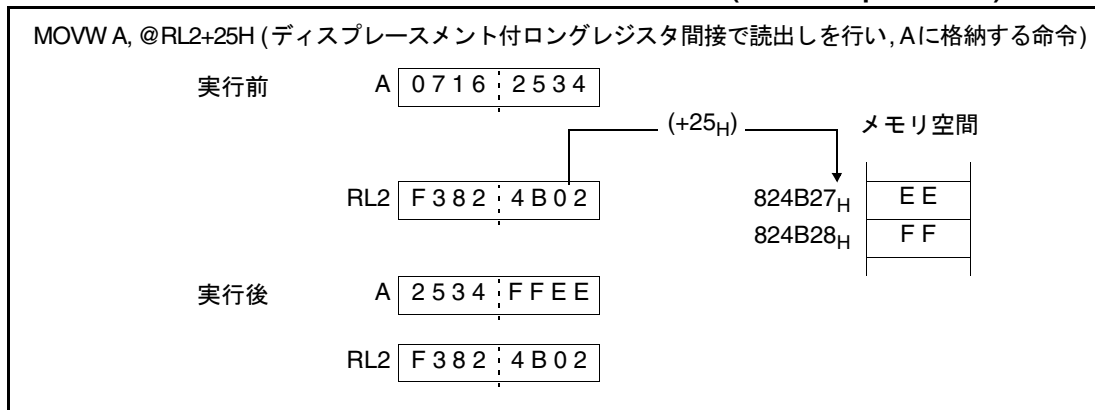
図 D.4-3 ディスプレースメント付レジスタ間接
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● 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 D.4-4 に例を示します。

図 D.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



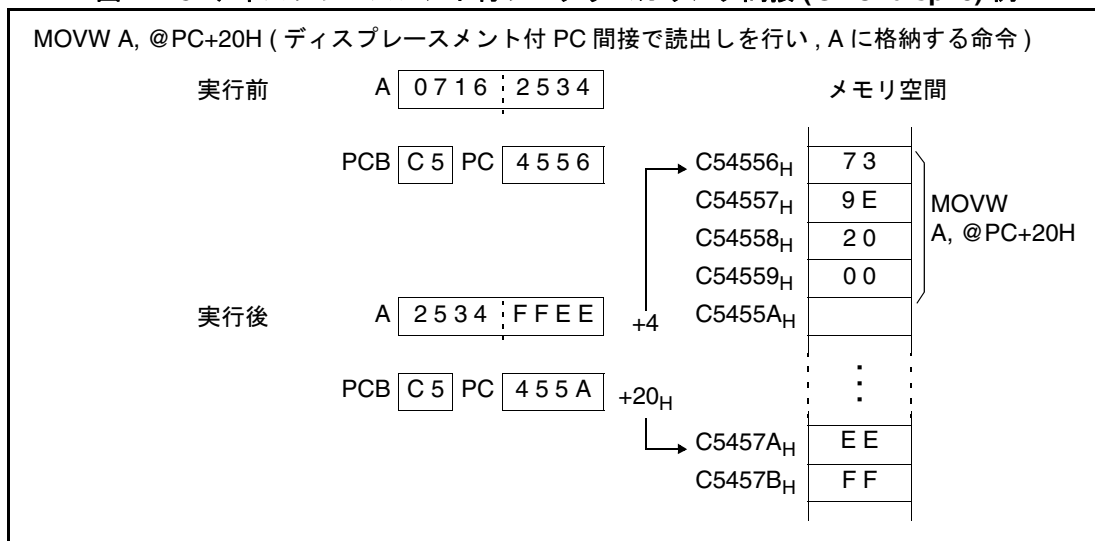
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(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 D.4-5 に例を示します。

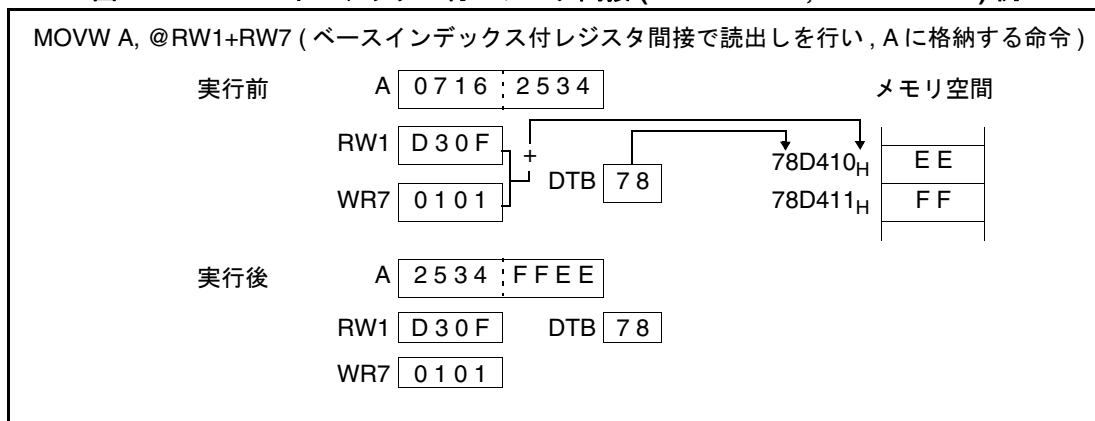
図 D.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例



● 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 D.4-6 に例を示します。

図 D.4-6 ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7) 例

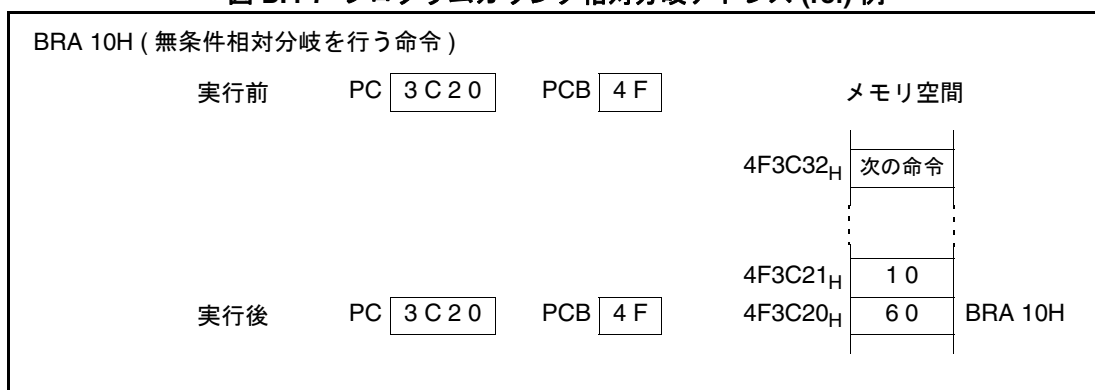


● プログラムカウンタ相対分岐アドレス (rel)

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 D.4-7 に例を示します。

図 D.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 D.4-8 にレジスタリストの構成, 図 D.4-9 に例を示します。

図 D.4-8 レジスタリストの構成

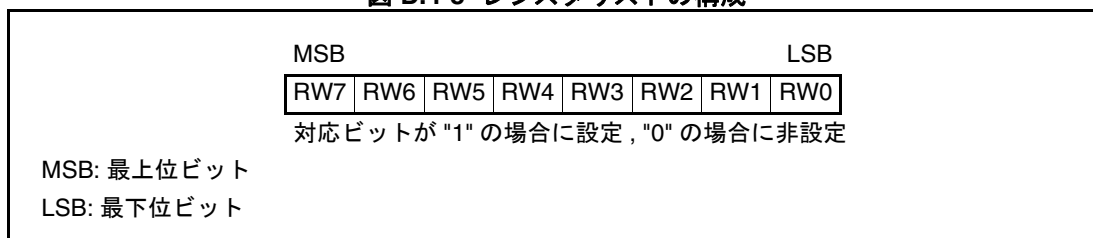
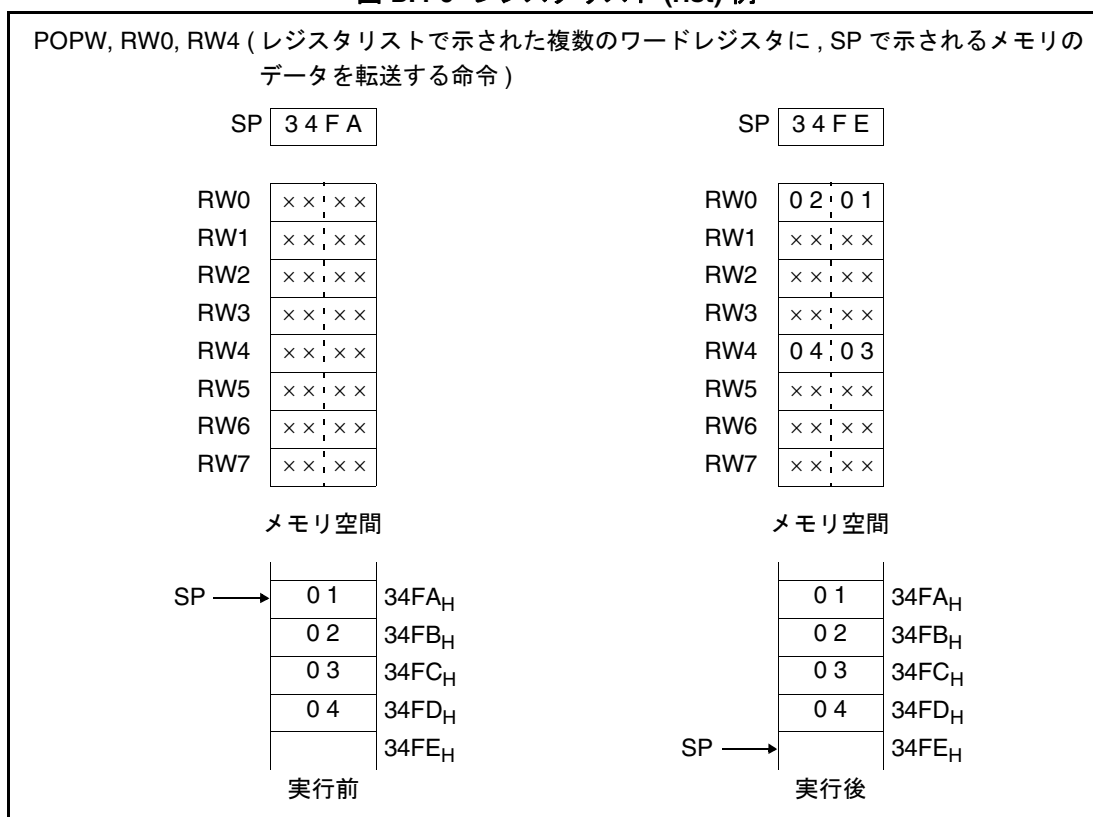


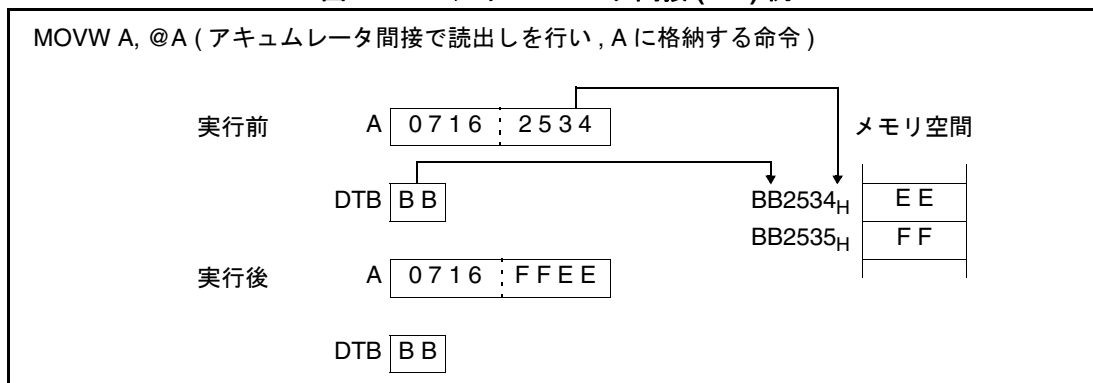
図 D.4-9 レジスタリスト (rlst) 例



● 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 D.4-10 に例を示します。

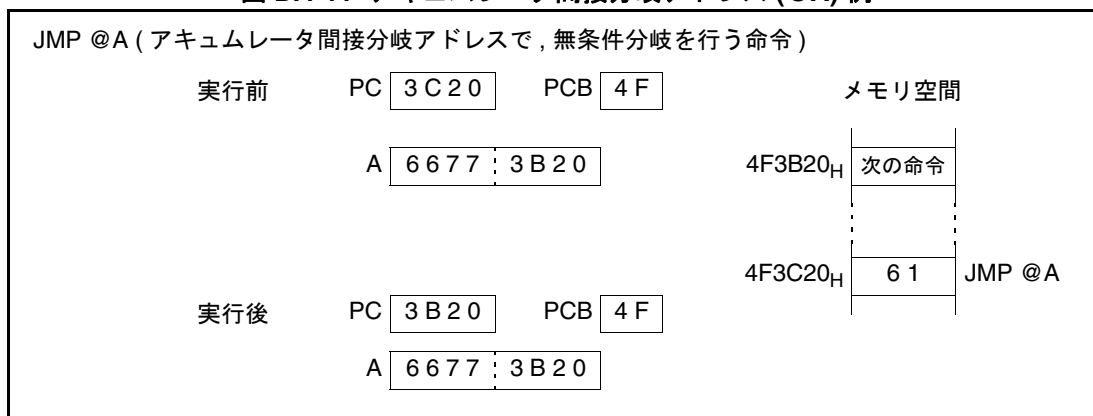
図 D.4-10 アキュムレータ間接 (@A) 例



● 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 D.4-11 に例を示します。

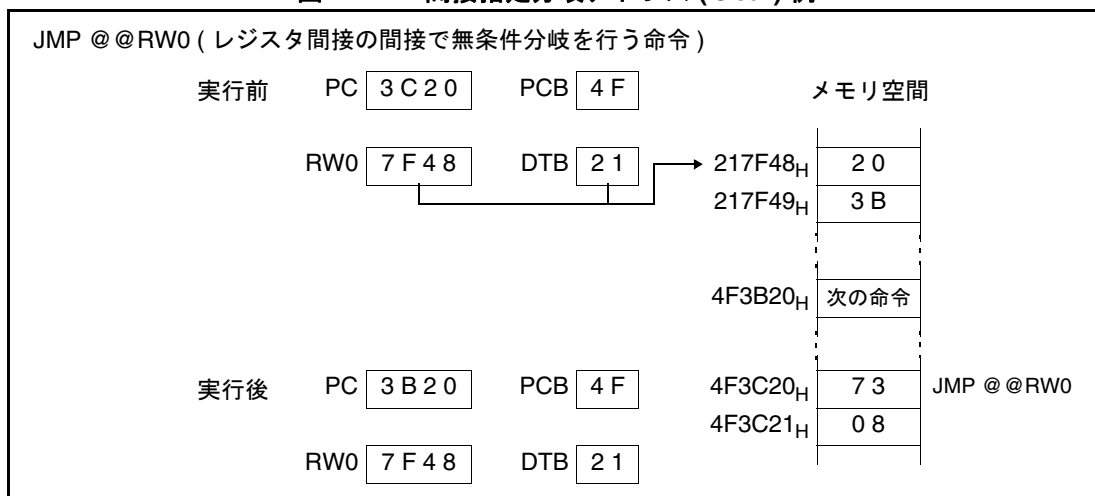
図 D.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 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 D.4-12 に例を示します。

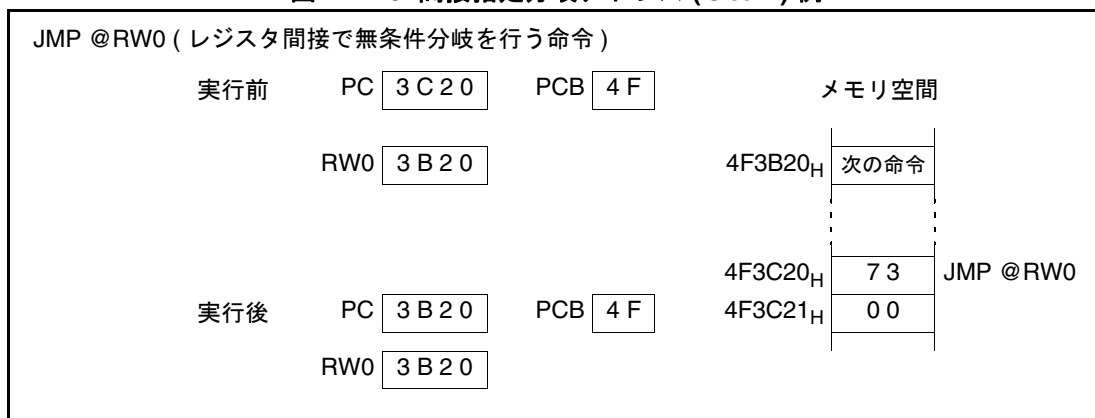
図 D.4-12 間接指定分岐アドレス (@ear) 例



● 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 D.4-13 に例を示します。

図 D.4-13 間接指定分岐アドレス (@eam) 例



D.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 D.5-1、表 D.5-2、表 D.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 D.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「D.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「D.7 命令一覧表の読み方」を参照してください。

表 D.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 * ¹		(c) ワード * ¹		(d) ロング * ¹	
	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 * ² 16 ビット偶数アドレス	+1	1	+1	1	+2	2
外部データバス * ² 16 ビット奇数アドレス	+1	1	+4	2	+8	4
外部データバス * ² 8 ビット	+1	1	+4	2	+8	4

*1: (b), (c), (d) は「D.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 D.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。

D.6 実効アドレスフィールド

表 D.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 D.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 D.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、「D.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「D.7 命令一覧表の読み方」を参照してください。

D.7 命令一覧表の読み方

「D.8 F²MC-16LX 命令一覧表」で使用している項目の説明を表 D.7-1 に、記号の説明を表 D.7-2 に示します。

■ 命令の表示記号の説明

表 D.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 _H を転送する。 X: AL の符号拡張で AH へ 00 _H または FF _H を転送する。
I	I (割込み許可), S (スタック), T (ステッキイビット), N (ネガティブ), Z (ゼロ), V (オーバーフロー), 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 D.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 _H ~ 0000FF _H)
#imm4	4 ビット即値データ
#imm8	8 ビット即値データ
#imm16	16 ビット即値データ
#imm32	32 ビット即値データ
ext (imm8)	8 ビット即値データを符号拡張した 16 ビットデータ

表 D.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

D.8 F²MC-16LX 命令一覧表

F²MC-16LX で使用している命令の一覧を示します。

■ F²MC-16LX 命令一覧表

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD A,dir	2	5	0	(b)	byte (A) \leftarrow (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD A,ear	2	3	1	0	byte (A) \leftarrow (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD ear,A	2	3	2	0	byte (ear) \leftarrow (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) \leftarrow (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC A	1	2	0	0	byte (A) \leftarrow (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,ear	2	3	1	0	byte (A) \leftarrow (A) + (ear) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) + (eam) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A	1	3	0	0	byte (A) \leftarrow (AH) + (AL) + (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
SUB A,#imm8	2	2	0	0	byte (A) \leftarrow (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB A,dir	2	5	0	(b)	byte (A) \leftarrow (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB A,ear	2	3	1	0	byte (A) \leftarrow (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB ear,A	2	3	2	0	byte (ear) \leftarrow (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) \leftarrow (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC A	1	2	0	0	byte (A) \leftarrow (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,ear	2	3	1	0	byte (A) \leftarrow (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC A	1	3	0	0	byte (A) \leftarrow (AH) - (AL) - (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
ADDW A	1	2	0	0	word (A) \leftarrow (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW A,ear	2	3	1	0	word (A) \leftarrow (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW A,#imm16	3	2	0	0	word (A) \leftarrow (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW ear,A	2	3	2	0	word (ear) \leftarrow (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW A,ear	2	3	1	0	word (A) \leftarrow (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW A	1	2	0	0	word (A) \leftarrow (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW A,ear	2	3	1	0	word (A) \leftarrow (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW A,#imm16	3	2	0	0	word (A) \leftarrow (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW ear,A	2	3	2	0	word (ear) \leftarrow (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW eam,A	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW A,ear	2	3	1	0	word (A) \leftarrow (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL A,ear	2	6	2	0	long (A) \leftarrow (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL A,#imm32	5	4	0	0	long (A) \leftarrow (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL A,ear	2	6	2	0	long (A) \leftarrow (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL A,#imm32	5	4	0	0	long (A) \leftarrow (A) - imm32	-	-	-	-	-	*	*	*	*	-

(注意事項) 表中の (a) ～ (d) は , 表 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	5+(a)	0	2 × (b)	byte (eam) \leftarrow (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	3	2	0	byte (ear) \leftarrow (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	5+(a)	0	2 × (b)	byte (eam) \leftarrow (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	3	2	0	word (ear) \leftarrow (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	3	2	0	word (ear) \leftarrow (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	7	4	0	long (ear) \leftarrow (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	9+(a)	0	2 × (d)	long (eam) \leftarrow (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	7	4	0	long (ear) \leftarrow (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	9+(a)	0	2 × (d)	long (eam) \leftarrow (eam) - 1	-	-	-	-	-	*	*	*	-	*

(注意事項) 表中の (a) ～ (d) は , 表 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL A,ear	2	6	2	0	long (A) \leftarrow (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL A,ear	2	6	2	0	long (A) \leftarrow (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

(注意事項) 表中の (a), (d) は, 表 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG ear	2	3	2	0	byte (ear) \leftarrow 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG eam	2+	5+(a)	0	2 \times (b)	byte (eam) \leftarrow 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW A	1	2	0	0	word (A) \leftarrow 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW ear	2	3	2	0	word (ear) \leftarrow 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW eam	2+	5+(a)	0	2 \times (c)	word (eam) \leftarrow 0 - (eam)	-	-	-	-	-	*	*	*	*	*

(注意事項) 表中の (a), (b), (c) は, 表 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow 最初に "1" が立っていた所まで左シフトする byte (R0) \leftarrow 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 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 : $3 \times (c) + (b)$

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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 D.5-1 および表 D.5-2 を参照してください。

表 D.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) \leftarrow (dir:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,addr16:bp	4	5	0	(b)	byte (A) \leftarrow (addr16:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,io:bp	3	4	0	(b)	byte (A) \leftarrow (io:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB dir:bp,A	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
MOVB addr16:bp,A	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
MOVB io:bp,A	3	6	0	2 × (b)	bit (io:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
SETB dir:bp	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
SETB addr16:bp	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
SETB io:bp	3	7	0	2 × (b)	bit (io:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
CLRB dir:bp	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow 0	-	-	-	-	-	-	-	-	-	*
CLRB addr16:bp	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow 0	-	-	-	-	-	-	-	-	-	*
CLRB io:bp	3	7	0	2 × (b)	bit (io:bp)b \leftarrow 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 \leftarrow 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 D.5-1 および表 D.5-2 を参照してください。

表 D.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 \longleftrightarrow (A)8-15	-	-	-	-	-	-	-	-	-	-
SWAPW	1	2	0	0	word (AH) \longleftrightarrow (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 D.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 D.5-1 および表 D.5-2 を参照してください。

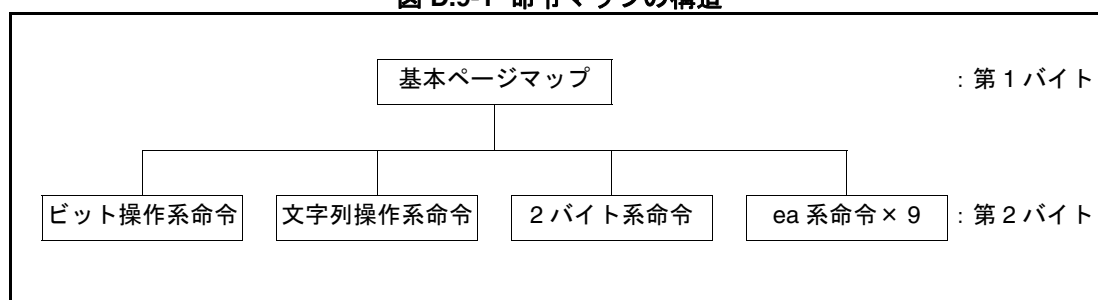
D.9 命令マップ

F²MC-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。

F²MC-16LX の命令マップを示します。

■ 命令マップの構造

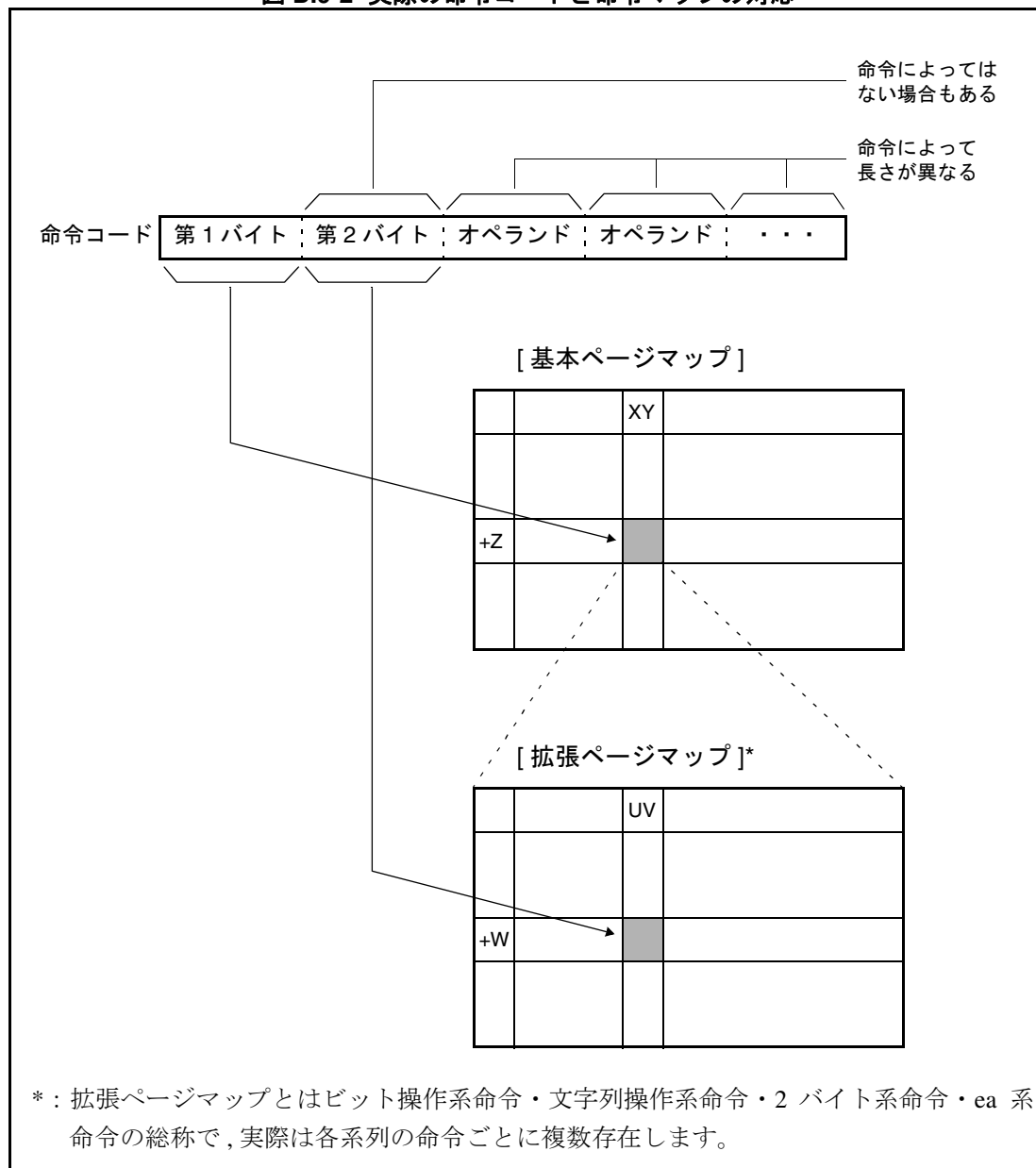
図 D.9-1 命令マップの構造



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 D.9-2 に示します。

図 D.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 D.9-1 に示します。

表 D.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 D.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	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	BNZ/BNIE rel
+2	ADDDC 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	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	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	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	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	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	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, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BT rel
+9	UNLINK	SUBL A, #32	SUBW A	SUBW A, #16	MOVW dir, A	MOVW io, A	INT	MOVEA Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BNT rel
+A	MOV RP, #8	MOV ILM, #8	CBNE A, #8, rel	CWBNE A, #16, rel	MOVW A, #16	MOVW A, addr16	INTP	MOV Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BLT rel
+B	NEGW A	CMPL A, #32	CMPW A	CMPW A, #16	MOVL A, #32	MOVW addr16, A	RETI	MOVW Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	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	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, Ri	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BGT rel
+E	ASRW A	SWAPW	XORW A	XORW A, #16	PUSHW PS	POPW PS	文字列操 作系命令	XCH Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BLS rel
+F	LSRW A	ADDSP #16	MULW A	NOTW A	PUSHW r1st	POPW r1st	2 バイト 系命令	XCHW Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWI+d8 @RWI+d8,A	MOVW A, #4	CALL	BHI rel

表 D.9-3 ビット操作系命令マップ (第1バイト=6C_H)

	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 D.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 PCB, PCB	MOVSD	MOVSWI	MOVSWD					SCEQI PCB	SCEQD PCB	SCWEQI PCB	SCWEQD PCB	FILSI PCB			
+1	PCB, DTB								DTB	DTB	DTB	DTB	DTB		DTB	
+2	PCB, ADB								ADB	ADB	ADB	ADB	ADB		ADB	
+3	PCB, SPB								SPB	SPB	SPB	SPB	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 D.9-5 2 バイト系命令マップ (第 1 バイト = 6F_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV A, DTB	MOV DTB, A	MOVX A, @RL0+d8	MOV @RL0+d8, A	MOV @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 D.9-6 ea 系命令 その1 (第1バイト=70_H)

	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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	CWNE↓ #16, rel	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+, @RW0+RW7	SUBL A, A, @RW0+, @RW0+RW7	SUBL A, A, @RW0+, @RW0+RW7	SUBL A, A, @RW0+, @RW0+RW7	@RW0+RW7 #16, rel	CWNE↓ #16, rel	CMPL A, A, @RW0+, @RW0+RW7	CMPL A, A, @RW0+, @RW0+RW7	ANDL A, A, @RW0+, @RW0+RW7	ANDL A, A, @RW0+, @RW0+RW7	ORL A, A, @RW0+, @RW0+RW7	ORL A, A, @RW0+, @RW0+RW7	XORL A, A, @RW0+, @RW0+RW7	XORL A, A, @RW0+, @RW0+RW7	使用禁止	@RW0+RW7 #8, rel
+D	ADDL A, A, @RW1+, @RW1+RW7	SUBL A, A, @RW1+, @RW1+RW7	SUBL A, A, @RW1+, @RW1+RW7	SUBL A, A, @RW1+, @RW1+RW7	@RW1+RW7 #16, rel	CWNE↓ #16, rel	CMPL A, A, @RW1+, @RW1+RW7	CMPL A, A, @RW1+, @RW1+RW7	ANDL A, A, @RW1+, @RW1+RW7	ANDL A, A, @RW1+, @RW1+RW7	ORL A, A, @RW1+, @RW1+RW7	ORL A, A, @RW1+, @RW1+RW7	XORL A, A, @RW1+, @RW1+RW7	XORL A, A, @RW1+, @RW1+RW7	使用禁止	@RW1+RW7 #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	CWNE↓ #16, rel	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	使用禁止	@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	CWNE↓ #16, rel	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	使用禁止	addr16 #8, rel

表 D.9-7 ea 系命令 その2 (第1バイト=71_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMPP @RL0, @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, A	MOVL RL0, A, @RW0+d8, A	MOV R0, #8, @RW0+d8, #8	MOV R0, #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, A	MOVL RL0, A, @RW1+d8, A	MOV R1, #8, @RW1+d8, #8	MOV R1, #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, A	MOVL RL1, A, @RW2+d8, A	MOV R2, #8, @RW2+d8, #8	MOV R2, #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, A	MOVL RL1, A, @RW3+d8, A	MOV R3, #8, @RW3+d8, #8	MOV R3, #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, A	MOVL RL2, A, @RW4+d8, A	MOV R4, #8, @RW4+d8, #8	MOV R4, #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, A	MOVL RL2, A, @RW5+d8, A	MOV R5, #8, @RW5+d8, #8	MOV R5, #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, A	MOVL RL3, A, @RW6+d8, A	MOV R6, #8, @RW6+d8, #8	MOV R6, #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, A	MOVL RL3, A, @RW7+d8, A	MOV R7, #8, @RW7+d8, #8	MOV R7, #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, A	MOVL @RW0, A, @RW0+d16, A	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, A	MOVL @RW1, A, @RW1+d16, A	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, A	MOVL @RW2, A, @RW2+d16, A	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, A	MOVL @RW3, A, @RW3+d16, A	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, A	MOVL @RW0+, A, @RW0+RW7, A	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, A	MOVL @RW1+, A, @RW1+RW7, A	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, @PC+d16, A	MOVL @PC+d16, A, @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, @addr16, A	MOVL @addr16, A, @addr16, A	MOV addr16, #8, addr16, #8	MOV addr16, #8, addr16, #8	MOVEA A, @addr16, A	MOVEA A, @addr16, A

表 D.9-8 ea 系命令 その 3 (第 1 バイト = 72_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ROL	R0' @RW0+d8	RORC	R0' @RW0+d8	INC	R0' @RW0+d8	DEC	R0' @RW0+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+1	ROL	R1' @RW1+d8	RORC	R1' @RW1+d8	INC	R1' @RW1+d8	DEC	R1' @RW1+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+2	ROL	R2' @RW2+d8	RORC	R2' @RW2+d8	INC	R2' @RW2+d8	DEC	R2' @RW2+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+3	ROL	R3' @RW3+d8	RORC	R3' @RW3+d8	INC	R3' @RW3+d8	DEC	R3' @RW3+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+4	ROL	R4' @RW4+d8	RORC	R4' @RW4+d8	INC	R4' @RW4+d8	DEC	R4' @RW4+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+5	ROL	R5' @RW5+d8	RORC	R5' @RW5+d8	INC	R5' @RW5+d8	DEC	R5' @RW5+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+6	ROL	R6' @RW6+d8	RORC	R6' @RW6+d8	INC	R6' @RW6+d8	DEC	R6' @RW6+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+7	ROL	R7' @RW7+d8	RORC	R7' @RW7+d8	INC	R7' @RW7+d8	DEC	R7' @RW7+d8	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+8	ROL	R0' @RW0+d16	RORC	R0' @RW0+d16	INC	R0' @RW0+d16	DEC	R0' @RW0+d16	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+9	ROL	R1' @RW1+d16	RORC	R1' @RW1+d16	INC	R1' @RW1+d16	DEC	R1' @RW1+d16	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+A	ROL	R2' @RW2+d16	RORC	R2' @RW2+d16	INC	R2' @RW2+d16	DEC	R2' @RW2+d16	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+B	ROL	R3' @RW3+d16	RORC	R3' @RW3+d16	INC	R3' @RW3+d16	DEC	R3' @RW3+d16	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+C	ROL	R0' @RW0+R7	RORC	R0' @RW0+R7	INC	R0' @RW0+R7	DEC	R0' @RW0+R7	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+D	ROL	R1' @RW1+R7	RORC	R1' @RW1+R7	INC	R1' @RW1+R7	DEC	R1' @RW1+R7	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+E	ROL	R2' @RW2+R7	RORC	R2' @RW2+R7	INC	R2' @RW2+R7	DEC	R2' @RW2+R7	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH
+F	ROL	R3' @RW3+R7	RORC	R3' @RW3+R7	INC	R3' @RW3+R7	DEC	R3' @RW3+R7	MOV	MOV	MOV	MOV	MOVX	MOVX	XCH	XCH

表 D.9-9 ea 系命令 その4 (第1バイト=73_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMP @RW0', @@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, A', @RW0+d8, A	MOVW RW0, A', @RW0+d8, A	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, A', @RW1+d8, A	MOVW RW1, A', @RW1+d8, A	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, A', @RW2+d8, A	MOVW RW2, A', @RW2+d8, A	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, A', @RW3+d8, A	MOVW RW3, A', @RW3+d8, A	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, A', @RW4+d8, A	MOVW RW4, A', @RW4+d8, A	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, A', @RW5+d8, A	MOVW RW5, A', @RW5+d8, A	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, A', @RW6+d8, A	MOVW RW6, A', @RW6+d8, A	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, A', @RW7+d8, A	MOVW RW7, A', @RW7+d8, A	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, A', @RW0+d16, A	MOVW @RW0, A', @RW0+d16, A	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, A', @RW1+d16, A	MOVW @RW1, A', @RW1+d16, A	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, A', @RW2+d16, A	MOVW @RW2, A', @RW2+d16, A	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, A', @RW3+d16, A	MOVW @RW3, A', @RW3+d16, A	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+, A', @RW0+RW7, A	MOVW @RW0+, A', @RW0+RW7, A	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+, A', @RW1+RW7, A	MOVW @RW1+, A', @RW1+RW7, A	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+, A', @PC+d16, A	MOVW @RW2+, A', @PC+d16, A	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+, A', @addr16, A	MOVW @RW3+, A', @addr16, A	MOVW @RW3+, #16', @addr16, #16	MOVW @RW3+, #16', @addr16, #16	XCHW A, @RW3+', @addr16	XCHW A, @RW3+', @addr16

表 D.9-10 ea 系命令 その5 (第1バイト=74_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD A, R0, @RW0+d8	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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	SUB 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 D.9-11 ea 系命令 その6 (第1バイト=75_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD R0, A', @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 D.9-12 ea 系命令 その 7 (第 1 バイト = 76_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW A, RW0, @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	DWBNZ RW0, r' @RW0+d8, r	DWBNZ 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	DWBNZ RW1, r' @RW1+d8, r	DWBNZ 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	DWBNZ RW2, r' @RW2+d8, r	DWBNZ 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	DWBNZ RW3, r' @RW3+d8, r	DWBNZ 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	DWBNZ RW4, r' @RW4+d8, r	DWBNZ 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	DWBNZ RW5, r' @RW5+d8, r	DWBNZ 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	DWBNZ RW6, r' @RW6+d8, r	DWBNZ 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	DWBNZ RW7, r' @RW7+d8, r	DWBNZ 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	DWBNZ @RW0, r' @RW0+d16, r	DWBNZ @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	DWBNZ @RW1, r' @RW1+d16, r	DWBNZ @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	DWBNZ @RW2, r' @RW2+d16, r	DWBNZ @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	DWBNZ @RW3, r' @RW3+d16, r	DWBNZ @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	DWBNZ @RW0+, r' @RW0+RW7, r	DWBNZ @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	DWBNZ @RW1+, r' @RW1+RW7, r	DWBNZ @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	DWBNZ @RW2+, r' @PC+d16, r	DWBNZ @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	DWBNZ @RW3+, r' addr16, r	DWBNZ @RW3+, r' addr16, r

表 D.9-13 ea 系命令 その 8 (第 1 バイト = 77H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW ,ADDW RW0, A' ,@RW0+d8, A	SUBW ,SUBW RW0, A' ,@RW0+d8, A	SUBCW ,SUBCW A, A, RW0' ,@RW0+d8	NEGW RW0' ,@RW0+d8	ANDW ,ANDW RW0, A' ,@RW0+d8, A	ORW RW0, A' ,@RW0+d8, A	XORW ,XORW RW0, A' ,@RW0+d8, A	NOTW ,NOTW RW0' ,@RW0+d8								
+1	ADDW ,ADDW RW1, A' ,@RW1+d8, A	SUBW ,SUBW RW1, A' ,@RW1+d8, A	SUBCW ,SUBCW A, A, RW1' ,@RW1+d8	NEGW RW1' ,@RW1+d8	ANDW ,ANDW RW1, A' ,@RW1+d8, A	ORW RW1, A' ,@RW1+d8, A	XORW ,XORW RW1, A' ,@RW1+d8, A	NOTW ,NOTW RW1' ,@RW1+d8								
+2	ADDW ,ADDW RW2, A' ,@RW2+d8, A	SUBW ,SUBW RW2, A' ,@RW2+d8, A	SUBCW ,SUBCW A, A, RW2' ,@RW2+d8	NEGW RW2' ,@RW2+d8	ANDW ,ANDW RW2, A' ,@RW2+d8, A	ORW RW2, A' ,@RW2+d8, A	XORW ,XORW RW2, A' ,@RW2+d8, A	NOTW ,NOTW RW2' ,@RW2+d8								
+3	ADDW ,ADDW RW3, A' ,@RW3+d8, A	SUBW ,SUBW RW3, A' ,@RW3+d8, A	SUBCW ,SUBCW A, A, RW3' ,@RW3+d8	NEGW RW3' ,@RW3+d8	ANDW ,ANDW RW3, A' ,@RW3+d8, A	ORW RW3, A' ,@RW3+d8, A	XORW ,XORW RW3, A' ,@RW3+d8, A	NOTW ,NOTW RW3' ,@RW3+d8								
+4	ADDW ,ADDW RW4, A' ,@RW4+d8, A	SUBW ,SUBW RW4, A' ,@RW4+d8, A	SUBCW ,SUBCW A, A, RW4' ,@RW4+d8	NEGW RW4' ,@RW4+d8	ANDW ,ANDW RW4, A' ,@RW4+d8, A	ORW RW4, A' ,@RW4+d8, A	XORW ,XORW RW4, A' ,@RW4+d8, A	NOTW ,NOTW RW4' ,@RW4+d8								
+5	ADDW ,ADDW RW5, A' ,@RW5+d8, A	SUBW ,SUBW RW5, A' ,@RW5+d8, A	SUBCW ,SUBCW A, A, RW5' ,@RW5+d8	NEGW RW5' ,@RW5+d8	ANDW ,ANDW RW5, A' ,@RW5+d8, A	ORW RW5, A' ,@RW5+d8, A	XORW ,XORW RW5, A' ,@RW5+d8, A	NOTW ,NOTW RW5' ,@RW5+d8								
+6	ADDW ,ADDW RW6, A' ,@RW6+d8, A	SUBW ,SUBW RW6, A' ,@RW6+d8, A	SUBCW ,SUBCW A, A, RW6' ,@RW6+d8	NEGW RW6' ,@RW6+d8	ANDW ,ANDW RW6, A' ,@RW6+d8, A	ORW RW6, A' ,@RW6+d8, A	XORW ,XORW RW6, A' ,@RW6+d8, A	NOTW ,NOTW RW6' ,@RW6+d8								
+7	ADDW ,ADDW RW7, A' ,@RW7+d8, A	SUBW ,SUBW RW7, A' ,@RW7+d8, A	SUBCW ,SUBCW A, A, RW7' ,@RW7+d8	NEGW RW7' ,@RW7+d8	ANDW ,ANDW RW7, A' ,@RW7+d8, A	ORW RW7, A' ,@RW7+d8, A	XORW ,XORW RW7, A' ,@RW7+d8, A	NOTW ,NOTW RW7' ,@RW7+d8								
+8	ADDW ,ADDW @RW0, A' ,@RW0+d16, A	SUBW ,SUBW @RW0, A' ,@RW0+d16, A	SUBCW ,SUBCW A, A' ,@RW0' ,@RW0+d16	NEGW @RW0' ,@RW0+d16	ANDW ,ANDW @RW0, A' ,@RW0+d16, A	ORW @RW0, A' ,@RW0+d16, A	XORW ,XORW @RW0, A' ,@RW0+d16, A	NOTW ,NOTW @RW0' ,@RW0+d16								
+9	ADDW ,ADDW @RW1, A' ,@RW1+d16, A	SUBW ,SUBW @RW1, A' ,@RW1+d16, A	SUBCW ,SUBCW A, A' ,@RW1' ,@RW1+d16	NEGW @RW1' ,@RW1+d16	ANDW ,ANDW @RW1, A' ,@RW1+d16, A	ORW @RW1, A' ,@RW1+d16, A	XORW ,XORW @RW1, A' ,@RW1+d16, A	NOTW ,NOTW @RW1' ,@RW1+d16								
+A	ADDW ,ADDW @RW2, A' ,@RW2+d16, A	SUBW ,SUBW @RW2, A' ,@RW2+d16, A	SUBCW ,SUBCW A, A' ,@RW2' ,@RW2+d16	NEGW @RW2' ,@RW2+d16	ANDW ,ANDW @RW2, A' ,@RW2+d16, A	ORW @RW2, A' ,@RW2+d16, A	XORW ,XORW @RW2, A' ,@RW2+d16, A	NOTW ,NOTW @RW2' ,@RW2+d16								
+B	ADDW ,ADDW @RW3, A' ,@RW3+d16, A	SUBW ,SUBW @RW3, A' ,@RW3+d16, A	SUBCW ,SUBCW A, A' ,@RW3' ,@RW3+d16	NEGW @RW3' ,@RW3+d16	ANDW ,ANDW @RW3, A' ,@RW3+d16, A	ORW @RW3, A' ,@RW3+d16, A	XORW ,XORW @RW3, A' ,@RW3+d16, A	NOTW ,NOTW @RW3' ,@RW3+d16								
+C	ADDW ,ADDW @RW0+, A' ,@RW0+RW7, A	SUBW ,SUBW @RW0+, A' ,@RW0+RW7, A	SUBCW ,SUBCW A, A' ,@RW0+' ,@RW0+RW7	NEGW @RW0+' ,@RW0+RW7	ANDW ,ANDW @RW0+, A' ,@RW0+RW7, A	ORW @RW0+, A' ,@RW0+RW7, A	XORW ,XORW @RW0+, A' ,@RW0+RW7, A	NOTW ,NOTW @RW0+' ,@RW0+RW7								
+D	ADDW ,ADDW @RW1+, A' ,@RW1+RW7, A	SUBW ,SUBW @RW1+, A' ,@RW1+RW7, A	SUBCW ,SUBCW A, A' ,@RW1+' ,@RW1+RW7	NEGW @RW1+' ,@RW1+RW7	ANDW ,ANDW @RW1+, A' ,@RW1+RW7, A	ORW @RW1+, A' ,@RW1+RW7, A	XORW ,XORW @RW1+, A' ,@RW1+RW7, A	NOTW ,NOTW @RW1+' ,@RW1+RW7								
+E	ADDW ,ADDW @RW2+, A' ,@PC+d16, A	SUBW ,SUBW @RW2+, A' ,@PC+d16, A	SUBCW ,SUBCW A, A' ,@RW2+' ,@PC+d16	NEGW @RW2+' ,@PC+d16	ANDW ,ANDW @RW2+, A' ,@PC+d16, A	ORW @RW2+, A' ,@PC+d16, A	XORW ,XORW @RW2+, A' ,@PC+d16, A	NOTW ,NOTW @RW2+' ,@PC+d16								
+F	ADDW ,ADDW @RW3+, A' ,addr16, A	SUBW ,SUBW @RW3+, A' ,addr16, A	SUBCW ,SUBCW A, A' ,@RW3+' ,addr16	NEGW @RW3+' ,addr16	ANDW ,ANDW @RW3+, A' ,addr16, A	ORW @RW3+, A' ,addr16, A	XORW ,XORW @RW3+, A' ,addr16, A	NOTW ,NOTW @RW3+' ,addr16								

表 D.9-14 ea 系命令 その9 (第1バイト=78_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MULU A, R0', @RW0+d8	MULU A, R0', @RW0+d8	MULW A, RW0', @RW0+d8	MULW A, RW0', @RW0+d8	MUL A, R0', @RW0+d8	MUL A, R0', @RW0+d8	MULW A, RW0', @RW0+d8	MULW A, RW0', @RW0+d8	DIVU A, R0', @RW0+d8	DIVU A, R0', @RW0+d8	DIVUW A, RW0', @RW0+d8	DIVUW A, RW0', @RW0+d8	DIV A, R0', @RW0+d8	DIV A, R0', @RW0+d8	DIVW A, RW0', @RW0+d8	DIVW A, RW0', @RW0+d8
+1	MULU A, R1', @RW1+d8	MULU A, R1', @RW1+d8	MULW A, RW1', @RW1+d8	MULW A, RW1', @RW1+d8	MUL A, R1', @RW1+d8	MUL A, R1', @RW1+d8	MULW A, RW1', @RW1+d8	MULW A, RW1', @RW1+d8	DIVU A, R1', @RW1+d8	DIVU A, R1', @RW1+d8	DIVUW A, RW1', @RW1+d8	DIVUW A, RW1', @RW1+d8	DIV A, R1', @RW1+d8	DIV A, R1', @RW1+d8	DIVW A, RW1', @RW1+d8	DIVW A, RW1', @RW1+d8
+2	MULU A, R2', @RW2+d8	MULU A, R2', @RW2+d8	MULW A, RW2', @RW2+d8	MULW A, RW2', @RW2+d8	MUL A, R2', @RW2+d8	MUL A, R2', @RW2+d8	MULW A, RW2', @RW2+d8	MULW A, RW2', @RW2+d8	DIVU A, R2', @RW2+d8	DIVU A, R2', @RW2+d8	DIVUW A, RW2', @RW2+d8	DIVUW A, RW2', @RW2+d8	DIV A, R2', @RW2+d8	DIV A, R2', @RW2+d8	DIVW A, RW2', @RW2+d8	DIVW A, RW2', @RW2+d8
+3	MULU A, R3', @RW3+d8	MULU A, R3', @RW3+d8	MULW A, RW3', @RW3+d8	MULW A, RW3', @RW3+d8	MUL A, R3', @RW3+d8	MUL A, R3', @RW3+d8	MULW A, RW3', @RW3+d8	MULW A, RW3', @RW3+d8	DIVU A, R3', @RW3+d8	DIVU A, R3', @RW3+d8	DIVUW A, RW3', @RW3+d8	DIVUW A, RW3', @RW3+d8	DIV A, R3', @RW3+d8	DIV A, R3', @RW3+d8	DIVW A, RW3', @RW3+d8	DIVW A, RW3', @RW3+d8
+4	MULU A, R4', @RW4+d8	MULU A, R4', @RW4+d8	MULW A, RW4', @RW4+d8	MULW A, RW4', @RW4+d8	MUL A, R4', @RW4+d8	MUL A, R4', @RW4+d8	MULW A, RW4', @RW4+d8	MULW A, RW4', @RW4+d8	DIVU A, R4', @RW4+d8	DIVU A, R4', @RW4+d8	DIVUW A, RW4', @RW4+d8	DIVUW A, RW4', @RW4+d8	DIV A, R4', @RW4+d8	DIV A, R4', @RW4+d8	DIVW A, RW4', @RW4+d8	DIVW A, RW4', @RW4+d8
+5	MULU A, R5', @RW5+d8	MULU A, R5', @RW5+d8	MULW A, RW5', @RW5+d8	MULW A, RW5', @RW5+d8	MUL A, R5', @RW5+d8	MUL A, R5', @RW5+d8	MULW A, RW5', @RW5+d8	MULW A, RW5', @RW5+d8	DIVU A, R5', @RW5+d8	DIVU A, R5', @RW5+d8	DIVUW A, RW5', @RW5+d8	DIVUW A, RW5', @RW5+d8	DIV A, R5', @RW5+d8	DIV A, R5', @RW5+d8	DIVW A, RW5', @RW5+d8	DIVW A, RW5', @RW5+d8
+6	MULU A, R6', @RW6+d8	MULU A, R6', @RW6+d8	MULW A, RW6', @RW6+d8	MULW A, RW6', @RW6+d8	MUL A, R6', @RW6+d8	MUL A, R6', @RW6+d8	MULW A, RW6', @RW6+d8	MULW A, RW6', @RW6+d8	DIVU A, R6', @RW6+d8	DIVU A, R6', @RW6+d8	DIVUW A, RW6', @RW6+d8	DIVUW A, RW6', @RW6+d8	DIV A, R6', @RW6+d8	DIV A, R6', @RW6+d8	DIVW A, RW6', @RW6+d8	DIVW A, RW6', @RW6+d8
+7	MULU A, R7', @RW7+d8	MULU A, R7', @RW7+d8	MULW A, RW7', @RW7+d8	MULW A, RW7', @RW7+d8	MUL A, R7', @RW7+d8	MUL A, R7', @RW7+d8	MULW A, RW7', @RW7+d8	MULW A, RW7', @RW7+d8	DIVU A, R7', @RW7+d8	DIVU A, R7', @RW7+d8	DIVUW A, RW7', @RW7+d8	DIVUW A, RW7', @RW7+d8	DIV A, R7', @RW7+d8	DIV A, R7', @RW7+d8	DIVW A, RW7', @RW7+d8	DIVW A, RW7', @RW7+d8
+8	MULU A, @RW0, @RW0+d16	MULU A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	MUL A, @RW0, @RW0+d16	MUL A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	DIVU A, @RW0, @RW0+d16	DIVU A, @RW0, @RW0+d16	DIVUW A, @RW0, @RW0+d16	DIVUW A, @RW0, @RW0+d16	DIV A, @RW0, @RW0+d16	DIV A, @RW0, @RW0+d16	DIVW A, @RW0, @RW0+d16	DIVW A, @RW0, @RW0+d16
+9	MULU A, @RW1, @RW1+d16	MULU A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	MUL A, @RW1, @RW1+d16	MUL A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	DIVU A, @RW1, @RW1+d16	DIVU A, @RW1, @RW1+d16	DIVUW A, @RW1, @RW1+d16	DIVUW A, @RW1, @RW1+d16	DIV A, @RW1, @RW1+d16	DIV A, @RW1, @RW1+d16	DIVW A, @RW1, @RW1+d16	DIVW A, @RW1, @RW1+d16
+A	MULU A, @RW2, @RW2+d16	MULU A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	MUL A, @RW2, @RW2+d16	MUL A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	DIVU A, @RW2, @RW2+d16	DIVU A, @RW2, @RW2+d16	DIVUW A, @RW2, @RW2+d16	DIVUW A, @RW2, @RW2+d16	DIV A, @RW2, @RW2+d16	DIV A, @RW2, @RW2+d16	DIVW A, @RW2, @RW2+d16	DIVW A, @RW2, @RW2+d16
+B	MULU A, @RW3, @RW3+d16	MULU A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	MUL A, @RW3, @RW3+d16	MUL A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	DIVU A, @RW3, @RW3+d16	DIVU A, @RW3, @RW3+d16	DIVUW A, @RW3, @RW3+d16	DIVUW A, @RW3, @RW3+d16	DIV A, @RW3, @RW3+d16	DIV A, @RW3, @RW3+d16	DIVW A, @RW3, @RW3+d16	DIVW A, @RW3, @RW3+d16
+C	MULU A, @RW0+, @RW0+RW7	MULU A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	MUL A, @RW0+, @RW0+RW7	MUL A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	DIVU A, @RW0+, @RW0+RW7	DIVU A, @RW0+, @RW0+RW7	DIVUW A, @RW0+, @RW0+RW7	DIVUW A, @RW0+, @RW0+RW7	DIV A, @RW0+, @RW0+RW7	DIV A, @RW0+, @RW0+RW7	DIVW A, @RW0+, @RW0+RW7	DIVW A, @RW0+, @RW0+RW7
+D	MULU A, @RW1+, @RW1+RW7	MULU A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	MUL A, @RW1+, @RW1+RW7	MUL A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	DIVU A, @RW1+, @RW1+RW7	DIVU A, @RW1+, @RW1+RW7	DIVUW A, @RW1+, @RW1+RW7	DIVUW A, @RW1+, @RW1+RW7	DIV A, @RW1+, @RW1+RW7	DIV A, @RW1+, @RW1+RW7	DIVW A, @RW1+, @RW1+RW7	DIVW A, @RW1+, @RW1+RW7
+E	MULU A, @RW2+, @PC+d16	MULU A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	MUL A, @RW2+, @PC+d16	MUL A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	DIVU A, @RW2+, @PC+d16	DIVU A, @RW2+, @PC+d16	DIVUW A, @RW2+, @PC+d16	DIVUW A, @RW2+, @PC+d16	DIV A, @RW2+, @PC+d16	DIV A, @RW2+, @PC+d16	DIVW A, @RW2+, @PC+d16	DIVW A, @RW2+, @PC+d16
+F	MULU A, @RW3+, addr16	MULU A, @RW3+, addr16	MULW A, @RW3+, addr16	MULW A, @RW3+, addr16	MUL A, @RW3+, addr16	MUL A, @RW3+, addr16	MULW A, @RW3+, addr16	MULW A, @RW3+, addr16	DIVU A, @RW3+, addr16	DIVU A, @RW3+, addr16	DIVUW A, @RW3+, addr16	DIVUW A, @RW3+, addr16	DIV A, @RW3+, addr16	DIV A, @RW3+, addr16	DIVW A, @RW3+, addr16	DIVW A, @RW3+, addr16

表 D.9-15 MOVEA RWi, ea 命令 (第1バイト=79_H)[illegible]

表 D.9-16 MOV Ri, ea 命令 (第1バイト = 7A_H)[illegible]

表 D.9-17 MOVW RWi, ea 命令 (第1バイト=7B_H)[illegible]

表 D.9-18 MOV Ri, ea 命令 (第1バイト = 7C_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0, @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 D.9-19 MOVW RWi, ea 命令 (第1バイト = 7D_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW RW0, RW0, @RW0-d8, RW0	MOVW RW0, RW1, @RW0-d8, RW1	MOVW RW0, RW1, @RW1-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, @RW4-d8, RW4	MOVW RW0, RW5, @RW4-d8, RW5	MOVW RW0, RW5, @RW5-d8, RW5	MOVW RW0, RW6, @RW5-d8, RW6	MOVW RW0, RW6, @RW6-d8, RW6	MOVW RW0, RW7, @RW6-d8, RW7	MOVW RW0, RW7, @RW7-d8, RW7
+1	MOVW RW1, RW0, @RW1-d8, RW0	MOVW RW1, RW1, @RW1-d8, RW1	MOVW RW1, RW2, @RW1-d8, RW2	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, @RW4-d8, RW4	MOVW RW1, RW5, @RW4-d8, RW5	MOVW RW1, RW5, @RW5-d8, RW5	MOVW RW1, RW6, @RW5-d8, RW6	MOVW RW1, RW6, @RW6-d8, RW6	MOVW RW1, RW7, @RW6-d8, RW7	MOVW RW1, RW7, @RW7-d8, RW7
+2	MOVW RW2, RW0, @RW2-d8, RW0	MOVW RW2, RW1, @RW2-d8, RW1	MOVW RW2, RW2, @RW2-d8, RW2	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, @RW4-d8, RW4	MOVW RW2, RW5, @RW4-d8, RW5	MOVW RW2, RW5, @RW5-d8, RW5	MOVW RW2, RW6, @RW5-d8, RW6	MOVW RW2, RW6, @RW6-d8, RW6	MOVW RW2, RW7, @RW6-d8, RW7	MOVW RW2, RW7, @RW7-d8, RW7
+3	MOVW RW3, RW0, @RW3-d8, RW0	MOVW RW3, RW1, @RW3-d8, RW1	MOVW RW3, RW2, @RW3-d8, RW2	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, @RW4-d8, RW4	MOVW RW3, RW5, @RW4-d8, RW5	MOVW RW3, RW5, @RW5-d8, RW5	MOVW RW3, RW6, @RW5-d8, RW6	MOVW RW3, RW6, @RW6-d8, RW6	MOVW RW3, RW7, @RW6-d8, RW7	MOVW RW3, RW7, @RW7-d8, RW7
+4	MOVW RW4, RW0, @RW4-d8, RW0	MOVW RW4, RW1, @RW4-d8, RW1	MOVW RW4, RW2, @RW4-d8, RW2	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, RW5, @RW5-d8, RW5	MOVW RW4, RW6, @RW5-d8, RW6	MOVW RW4, RW6, @RW6-d8, RW6	MOVW RW4, RW7, @RW6-d8, RW7	MOVW RW4, RW7, @RW7-d8, RW7
+5	MOVW RW5, RW0, @RW5-d8, RW0	MOVW RW5, RW1, @RW5-d8, RW1	MOVW RW5, RW2, @RW5-d8, RW2	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, @RW4-d8, RW4	MOVW RW5, RW5, @RW4-d8, RW5	MOVW RW5, RW5, @RW5-d8, RW5	MOVW RW5, RW6, @RW5-d8, RW6	MOVW RW5, RW6, @RW6-d8, RW6	MOVW RW5, RW7, @RW6-d8, RW7	MOVW RW5, RW7, @RW7-d8, RW7
+6	MOVW RW6, RW0, @RW6-d8, RW0	MOVW RW6, RW1, @RW6-d8, RW1	MOVW RW6, RW2, @RW6-d8, RW2	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, @RW4-d8, RW4	MOVW RW6, RW5, @RW4-d8, RW5	MOVW RW6, RW5, @RW5-d8, RW5	MOVW RW6, RW6, @RW5-d8, RW6	MOVW RW6, RW6, @RW6-d8, RW6	MOVW RW6, RW7, @RW6-d8, RW7	MOVW RW6, RW7, @RW7-d8, RW7
+7	MOVW RW7, RW0, @RW7-d8, RW0	MOVW RW7, RW1, @RW7-d8, RW1	MOVW RW7, RW2, @RW7-d8, RW2	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, @RW4-d8, RW4	MOVW RW7, RW5, @RW4-d8, RW5	MOVW RW7, RW5, @RW5-d8, RW5	MOVW RW7, RW6, @RW5-d8, RW6	MOVW RW7, RW6, @RW6-d8, RW6	MOVW RW7, RW7, @RW6-d8, RW7	MOVW RW7, RW7, @RW7-d8, RW7
+8	MOVW @RW0, RW0, @RW0-d16, RW0	MOVW @RW0, RW1, @RW0-d16, RW1	MOVW @RW0, RW2, @RW0-d16, RW2	MOVW @RW0, RW3, @RW0-d16, RW3	MOVW @RW0, RW4, @RW0-d16, RW4	MOVW @RW0, RW5, @RW0-d16, RW5	MOVW @RW0, RW6, @RW0-d16, RW6	MOVW @RW0, RW7, @RW0-d16, RW7	MOVW @RW0, RW8, @RW0-d16, RW8	MOVW @RW0, RW4, @RW4-d16, RW4	MOVW @RW0, RW5, @RW4-d16, RW5	MOVW @RW0, RW5, @RW5-d16, RW5	MOVW @RW0, RW6, @RW5-d16, RW6	MOVW @RW0, RW6, @RW6-d16, RW6	MOVW @RW0, RW7, @RW6-d16, RW7	MOVW @RW0, RW7, @RW7-d16, RW7
+9	MOVW @RW1, RW0, @RW1-d16, RW0	MOVW @RW1, RW1, @RW1-d16, RW1	MOVW @RW1, RW2, @RW1-d16, RW2	MOVW @RW1, RW3, @RW1-d16, RW3	MOVW @RW1, RW4, @RW1-d16, RW4	MOVW @RW1, RW5, @RW1-d16, RW5	MOVW @RW1, RW6, @RW1-d16, RW6	MOVW @RW1, RW7, @RW1-d16, RW7	MOVW @RW1, RW8, @RW1-d16, RW8	MOVW @RW1, RW4, @RW4-d16, RW4	MOVW @RW1, RW5, @RW4-d16, RW5	MOVW @RW1, RW5, @RW5-d16, RW5	MOVW @RW1, RW6, @RW5-d16, RW6	MOVW @RW1, RW6, @RW6-d16, RW6	MOVW @RW1, RW7, @RW6-d16, RW7	MOVW @RW1, RW7, @RW7-d16, RW7
+A	MOVW @RW2, RW0, @RW2-d16, RW0	MOVW @RW2, RW1, @RW2-d16, RW1	MOVW @RW2, RW2, @RW2-d16, RW2	MOVW @RW2, RW3, @RW2-d16, RW3	MOVW @RW2, RW4, @RW2-d16, RW4	MOVW @RW2, RW5, @RW2-d16, RW5	MOVW @RW2, RW6, @RW2-d16, RW6	MOVW @RW2, RW7, @RW2-d16, RW7	MOVW @RW2, RW8, @RW2-d16, RW8	MOVW @RW2, RW4, @RW4-d16, RW4	MOVW @RW2, RW5, @RW4-d16, RW5	MOVW @RW2, RW5, @RW5-d16, RW5	MOVW @RW2, RW6, @RW5-d16, RW6	MOVW @RW2, RW6, @RW6-d16, RW6	MOVW @RW2, RW7, @RW6-d16, RW7	MOVW @RW2, RW7, @RW7-d16, RW7
+B	MOVW @RW3, RW0, @RW3-d16, RW0	MOVW @RW3, RW1, @RW3-d16, RW1	MOVW @RW3, RW2, @RW3-d16, RW2	MOVW @RW3, RW3, @RW3-d16, RW3	MOVW @RW3, RW4, @RW3-d16, RW4	MOVW @RW3, RW5, @RW3-d16, RW5	MOVW @RW3, RW6, @RW3-d16, RW6	MOVW @RW3, RW7, @RW3-d16, RW7	MOVW @RW3, RW8, @RW3-d16, RW8	MOVW @RW3, RW4, @RW4-d16, RW4	MOVW @RW3, RW5, @RW4-d16, RW5	MOVW @RW3, RW5, @RW5-d16, RW5	MOVW @RW3, RW6, @RW5-d16, RW6	MOVW @RW3, RW6, @RW6-d16, RW6	MOVW @RW3, RW7, @RW6-d16, RW7	MOVW @RW3, RW7, @RW7-d16, RW7
+C	MOVW @RW0+, RW0, @RW0+, RW0	MOVW @RW0+, RW1, @RW0+, RW1	MOVW @RW0+, RW2, @RW0+, RW2	MOVW @RW0+, RW3, @RW0+, RW3	MOVW @RW0+, RW4, @RW0+, RW4	MOVW @RW0+, RW5, @RW0+, RW5	MOVW @RW0+, RW6, @RW0+, RW6	MOVW @RW0+, RW7, @RW0+, RW7	MOVW @RW0+, RW8, @RW0+, RW8	MOVW @RW0+, RW4, @RW4+, RW4	MOVW @RW0+, RW5, @RW4+, RW5	MOVW @RW0+, RW5, @RW5+, RW5	MOVW @RW0+, RW6, @RW5+, RW6	MOVW @RW0+, RW6, @RW6+, RW6	MOVW @RW0+, RW7, @RW6+, RW7	MOVW @RW0+, RW7, @RW7+, RW7
+D	MOVW @RW1+, RW0, @RW1+, RW0	MOVW @RW1+, RW1, @RW1+, RW1	MOVW @RW1+, RW2, @RW1+, RW2	MOVW @RW1+, RW3, @RW1+, RW3	MOVW @RW1+, RW4, @RW1+, RW4	MOVW @RW1+, RW5, @RW1+, RW5	MOVW @RW1+, RW6, @RW1+, RW6	MOVW @RW1+, RW7, @RW1+, RW7	MOVW @RW1+, RW8, @RW1+, RW8	MOVW @RW1+, RW4, @RW4+, RW4	MOVW @RW1+, RW5, @RW4+, RW5	MOVW @RW1+, RW5, @RW5+, RW5	MOVW @RW1+, RW6, @RW5+, RW6	MOVW @RW1+, RW6, @RW6+, RW6	MOVW @RW1+, RW7, @RW6+, RW7	MOVW @RW1+, RW7, @RW7+, RW7
+E	MOVW @RW2+, RW0, @RW2+, RW0	MOVW @RW2+, RW1, @RW2+, RW1	MOVW @RW2+, RW2, @RW2+, RW2	MOVW @RW2+, RW3, @RW2+, RW3	MOVW @RW2+, RW4, @RW2+, RW4	MOVW @RW2+, RW5, @RW2+, RW5	MOVW @RW2+, RW6, @RW2+, RW6	MOVW @RW2+, RW7, @RW2+, RW7	MOVW @RW2+, RW8, @RW2+, RW8	MOVW @RW2+, RW4, @RW4+, RW4	MOVW @RW2+, RW5, @RW4+, RW5	MOVW @RW2+, RW5, @RW5+, RW5	MOVW @RW2+, RW6, @RW5+, RW6	MOVW @RW2+, RW6, @RW6+, RW6	MOVW @RW2+, RW7, @RW6+, RW7	MOVW @RW2+, RW7, @RW7+, RW7
+F	MOVW @RW3+, RW0, @RW3+, RW0	MOVW @RW3+, RW1, @RW3+, RW1	MOVW @RW3+, RW2, @RW3+, RW2	MOVW @RW3+, RW3, @RW3+, RW3	MOVW @RW3+, RW4, @RW3+, RW4	MOVW @RW3+, RW5, @RW3+, RW5	MOVW @RW3+, RW6, @RW3+, RW6	MOVW @RW3+, RW7, @RW3+, RW7	MOVW @RW3+, RW8, @RW3+, RW8	MOVW @RW3+, RW4, @RW4+, RW4	MOVW @RW3+, RW5, @RW4+, RW5	MOVW @RW3+, RW5, @RW5+, RW5	MOVW @RW3+, RW6, @RW5+, RW6	MOVW @RW3+, RW6, @RW6+, RW6	MOVW @RW3+, RW7, @RW6+, RW7	MOVW @RW3+, RW7, @RW7+, RW7

表 D.9-20 XCH Ri, ea 命令 (第1バイト = 7EH)

	00	10	20	30	40	50	60	70	80	90	A	B0	C0	D0	E0	F0
+0	XCH R0, R0, R0' @RW0+d8	XCH R1, R1, R1' @RW0+d8	XCH R2, R2, R2' @RW0+d8	XCH R3, R3, R3' @RW0+d8	XCH R4, R4, R4' @RW0+d8	XCH R5, R5, R5' @RW0+d8	XCH R6, R6, R6' @RW0+d8	XCH R7, R7, R7' @RW0+d8	XCH R0, R0, R0' @RW0+d8	XCH R1, R1, R1' @RW0+d8	XCH R2, R2, R2' @RW0+d8	XCH R3, R3, R3' @RW0+d8	XCH R4, R4, R4' @RW0+d8	XCH R5, R5, R5' @RW0+d8	XCH R6, R6, R6' @RW0+d8	XCH R7, R7, R7' @RW0+d8
+1	XCH R0, R1' @RW1+d8	XCH R1, R0' @RW0+d8	XCH R2, R1' @RW1+d8	XCH R3, R1' @RW1+d8	XCH R4, R1' @RW1+d8	XCH R5, R1' @RW1+d8	XCH R6, R1' @RW1+d8	XCH R7, R1' @RW1+d8	XCH R1, R1' @RW1+d8	XCH R2, R2' @RW2+d8	XCH R3, R2' @RW2+d8	XCH R4, R2' @RW2+d8	XCH R5, R2' @RW2+d8	XCH R6, R2' @RW2+d8	XCH R7, R2' @RW2+d8	XCH R0, R0' @RW0+d8
+2	XCH R0, R2' @RW2+d8	XCH R2, R0' @RW0+d8	XCH R3, R2' @RW2+d8	XCH R4, R2' @RW2+d8	XCH R5, R2' @RW2+d8	XCH R6, R2' @RW2+d8	XCH R7, R2' @RW2+d8	XCH R0, R0' @RW0+d8	XCH R2, R2' @RW2+d8	XCH R3, R3' @RW3+d8	XCH R4, R3' @RW3+d8	XCH R5, R3' @RW3+d8	XCH R6, R3' @RW3+d8	XCH R7, R3' @RW3+d8	XCH R0, R0' @RW0+d8	XCH R1, R1' @RW1+d8
+3	XCH R0, R3' @RW3+d8	XCH R3, R0' @RW0+d8	XCH R4, R3' @RW3+d8	XCH R5, R3' @RW3+d8	XCH R6, R3' @RW3+d8	XCH R7, R3' @RW3+d8	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R4' @RW4+d8	XCH R6, R4' @RW4+d8	XCH R7, R4' @RW4+d8	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8
+4	XCH R0, R4' @RW4+d8	XCH R4, R0' @RW0+d8	XCH R5, R4' @RW4+d8	XCH R6, R4' @RW4+d8	XCH R7, R4' @RW4+d8	XCH R0, R0' @RW0+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8	XCH R6, R6' @RW6+d8	XCH R7, R6' @RW6+d8	XCH R0, R0' @RW0+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8
+5	XCH R0, R5' @RW5+d8	XCH R5, R0' @RW0+d8	XCH R6, R5' @RW5+d8	XCH R7, R5' @RW5+d8	XCH R0, R0' @RW0+d8	XCH R5, R5' @RW5+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R5, R5' @RW5+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R1, R1' @RW1+d8
+6	XCH R0, R6' @RW6+d8	XCH R6, R0' @RW0+d8	XCH R7, R6' @RW6+d8	XCH R0, R0' @RW0+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R6, R6' @RW6+d8	XCH R7, R7' @RW7+d8	XCH R1, R1' @RW1+d8
+7	XCH R0, R7' @RW7+d8	XCH R7, R0' @RW0+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R7, R7' @RW7+d8	XCH R0, R0' @RW0+d8	XCH R2, R2' @RW2+d8
+8	XCH R0, @RW0' @RW0+d16	XCH R1, @RW1' @RW1+d16	XCH R2, @RW2' @RW2+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW4' @RW4+d16	XCH R5, @RW5' @RW5+d16	XCH R6, @RW6' @RW6+d16	XCH R7, @RW7' @RW7+d16	XCH R0, @RW0' @RW0+d16	XCH R1, @RW1' @RW1+d16	XCH R2, @RW2' @RW2+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW4' @RW4+d16	XCH R5, @RW5' @RW5+d16	XCH R6, @RW6' @RW6+d16	XCH R7, @RW7' @RW7+d16
+9	XCH R0, @RW1' @RW1+d16	XCH R1, @RW1' @RW1+d16	XCH R2, @RW2' @RW2+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW4' @RW4+d16	XCH R5, @RW5' @RW5+d16	XCH R6, @RW6' @RW6+d16	XCH R7, @RW7' @RW7+d16	XCH R1, @RW1' @RW1+d16	XCH R2, @RW2' @RW2+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW4' @RW4+d16	XCH R5, @RW5' @RW5+d16	XCH R6, @RW6' @RW6+d16	XCH R7, @RW7' @RW7+d16	XCH R0, R0' @RW0+d8
+A	XCH R0, @RW2' W2+d16, A	XCH R1, @RW2' W2+d16, A	XCH R2, @RW2' W2+d16, A	XCH R3, @RW2' W2+d16, A	XCH R4, @RW2' W2+d16, A	XCH R5, @RW2' W2+d16, A	XCH R6, @RW2' W2+d16, A	XCH R7, @RW2' W2+d16, A	XCH R2, @RW2' W2+d16, A	XCH R3, @RW2' W2+d16, A	XCH R4, @RW2' W2+d16, A	XCH R5, @RW2' W2+d16, A	XCH R6, @RW2' W2+d16, A	XCH R7, @RW2' W2+d16, A	XCH R0, R0' @RW0+d8	XCH R1, R1' @RW1+d8
+B	XCH R0, @RW3' @RW3+d16	XCH R1, @RW3' @RW3+d16	XCH R2, @RW3' @RW3+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW3' @RW3+d16	XCH R5, @RW3' @RW3+d16	XCH R6, @RW3' @RW3+d16	XCH R7, @RW3' @RW3+d16	XCH R3, @RW3' @RW3+d16	XCH R4, @RW3' @RW3+d16	XCH R5, @RW3' @RW3+d16	XCH R6, @RW3' @RW3+d16	XCH R7, @RW3' @RW3+d16	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8
+C	XCH R0, @RW0+ @RW0+RW7	XCH R1, @RW0+ @RW0+RW7	XCH R2, @RW0+ @RW0+RW7	XCH R3, @RW0+ @RW0+RW7	XCH R4, @RW0+ @RW0+RW7	XCH R5, @RW0+ @RW0+RW7	XCH R6, @RW0+ @RW0+RW7	XCH R7, @RW0+ @RW0+RW7	XCH R4, @RW0+ @RW0+RW7	XCH R5, @RW0+ @RW0+RW7	XCH R6, @RW0+ @RW0+RW7	XCH R7, @RW0+ @RW0+RW7	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8
+D	XCH R0, @RW1+ @RW1+RW7	XCH R1, @RW1+ @RW1+RW7	XCH R2, @RW1+ @RW1+RW7	XCH R3, @RW1+ @RW1+RW7	XCH R4, @RW1+ @RW1+RW7	XCH R5, @RW1+ @RW1+RW7	XCH R6, @RW1+ @RW1+RW7	XCH R7, @RW1+ @RW1+RW7	XCH R5, @RW1+ @RW1+RW7	XCH R6, @RW1+ @RW1+RW7	XCH R7, @RW1+ @RW1+RW7	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8	XCH R6, R6' @RW6+d8
+E	XCH R0, @RW2+ @PC+d16	XCH R1, @RW2+ @PC+d16	XCH R2, @RW2+ @PC+d16	XCH R3, @RW2+ @PC+d16	XCH R4, @RW2+ @PC+d16	XCH R5, @RW2+ @PC+d16	XCH R6, @RW2+ @PC+d16	XCH R7, @RW2+ @PC+d16	XCH R4, @RW2+ @PC+d16	XCH R5, @RW2+ @PC+d16	XCH R6, @RW2+ @PC+d16	XCH R7, @RW2+ @PC+d16	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8
+F	XCH R0, @RW3+ R0, addr16	XCH R1, @RW3+ R1, addr16	XCH R2, @RW3+ R2, addr16	XCH R3, @RW3+ R3, addr16	XCH R4, @RW3+ R4, addr16	XCH R5, @RW3+ R5, addr16	XCH R6, @RW3+ R6, addr16	XCH R7, @RW3+ R7, addr16	XCH R4, @RW3+ R4, addr16	XCH R5, @RW3+ R5, addr16	XCH R6, @RW3+ R6, addr16	XCH R7, @RW3+ R7, addr16	XCH R0, R0' @RW0+d8	XCH R3, R3' @RW3+d8	XCH R4, R4' @RW4+d8	XCH R5, R5' @RW5+d8

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	XCHW RW0, RW0	XCHW RW0, @RW0-d8	XCHW RW1, RW0	XCHW RW1, @RW0-d8	XCHW RW2, RW2	XCHW RW2, @RW0-d8	XCHW RW3, RW3	XCHW RW3, RW0	XCHW RW4, RW0	XCHW RW4, @RW0-d8	XCHW RW5, RW0	XCHW RW5, @RW0-d8	XCHW RW6, RW0	XCHW RW6, @RW0-d8	XCHW RW7, RW0	XCHW RW7, @RW0-d8
+1	XCHW RW0, RW1	XCHW RW1, @RW1-d8	XCHW RW1, RW1	XCHW RW1, @RW1-d8	XCHW RW2, RW2	XCHW RW2, @RW1-d8	XCHW RW3, RW3	XCHW RW3, RW1	XCHW RW4, RW1	XCHW RW4, @RW1-d8	XCHW RW5, RW1	XCHW RW5, @RW1-d8	XCHW RW6, RW1	XCHW RW6, @RW1-d8	XCHW RW7, RW1	XCHW RW7, @RW1-d8
+2	XCHW RW0, RW2	XCHW RW2, @RW2-d8	XCHW RW1, RW2	XCHW RW1, @RW2-d8	XCHW RW2, RW2	XCHW RW2, @RW2-d8	XCHW RW3, RW3	XCHW RW3, RW2	XCHW RW4, RW2	XCHW RW4, @RW2-d8	XCHW RW5, RW2	XCHW RW5, @RW2-d8	XCHW RW6, RW2	XCHW RW6, @RW2-d8	XCHW RW7, RW2	XCHW RW7, @RW2-d8
+3	XCHW RW0, RW3	XCHW RW3, @RW3-d8	XCHW RW1, RW3	XCHW RW1, @RW3-d8	XCHW RW2, RW2	XCHW RW2, @RW3-d8	XCHW RW3, RW3	XCHW RW3, RW3	XCHW RW4, RW3	XCHW RW4, @RW3-d8	XCHW RW5, RW3	XCHW RW5, @RW3-d8	XCHW RW6, RW3	XCHW RW6, @RW3-d8	XCHW RW7, RW3	XCHW RW7, @RW3-d8
+4	XCHW RW0, RW4	XCHW RW4, @RW4-d8	XCHW RW1, RW4	XCHW RW1, @RW4-d8	XCHW RW2, RW2	XCHW RW2, @RW4-d8	XCHW RW3, RW3	XCHW RW3, RW4	XCHW RW4, RW4	XCHW RW4, @RW4-d8	XCHW RW5, RW4	XCHW RW5, @RW4-d8	XCHW RW6, RW4	XCHW RW6, @RW4-d8	XCHW RW7, RW4	XCHW RW7, @RW4-d8
+5	XCHW RW0, RW5	XCHW RW5, @RW5-d8	XCHW RW1, RW5	XCHW RW1, @RW5-d8	XCHW RW2, RW2	XCHW RW2, @RW5-d8	XCHW RW3, RW3	XCHW RW3, RW5	XCHW RW4, RW5	XCHW RW4, @RW5-d8	XCHW RW5, RW5	XCHW RW5, @RW5-d8	XCHW RW6, RW5	XCHW RW6, @RW5-d8	XCHW RW7, RW5	XCHW RW7, @RW5-d8
+6	XCHW RW0, RW6	XCHW RW6, @RW6-d8	XCHW RW1, RW6	XCHW RW1, @RW6-d8	XCHW RW2, RW2	XCHW RW2, @RW6-d8	XCHW RW3, RW3	XCHW RW3, RW6	XCHW RW4, RW6	XCHW RW4, @RW6-d8	XCHW RW5, RW6	XCHW RW5, @RW6-d8	XCHW RW6, RW6	XCHW RW6, @RW6-d8	XCHW RW7, RW6	XCHW RW7, @RW6-d8
+7	XCHW RW0, RW7	XCHW RW7, @RW7-d8	XCHW RW1, RW7	XCHW RW1, @RW7-d8	XCHW RW2, RW2	XCHW RW2, @RW7-d8	XCHW RW3, RW3	XCHW RW3, RW7	XCHW RW4, RW7	XCHW RW4, @RW7-d8	XCHW RW5, RW7	XCHW RW5, @RW7-d8	XCHW RW6, RW7	XCHW RW6, @RW7-d8	XCHW RW7, RW7	XCHW RW7, @RW7-d8
+8	XCHW RW0, RW0	XCHW RW0, @RW0-d16	XCHW RW1, @RW0	XCHW RW1, @RW0-d16	XCHW RW2, RW2	XCHW RW2, @RW0-d16	XCHW RW3, RW3	XCHW RW3, @RW0	XCHW RW4, RW0	XCHW RW4, @RW0-d16	XCHW RW5, RW0	XCHW RW5, @RW0-d16	XCHW RW6, RW0	XCHW RW6, @RW0-d16	XCHW RW7, RW0	XCHW RW7, @RW0-d16
+9	XCHW RW0, RW0	XCHW RW1, @RW1-d16	XCHW RW1, @RW1	XCHW RW1, @RW1-d16	XCHW RW2, RW2	XCHW RW2, @RW1-d16	XCHW RW3, RW3	XCHW RW3, @RW1	XCHW RW4, RW1	XCHW RW4, @RW1-d16	XCHW RW5, RW1	XCHW RW5, @RW1-d16	XCHW RW6, RW1	XCHW RW6, @RW1-d16	XCHW RW7, RW1	XCHW RW7, @RW1-d16
+A	XCHW RW0, RW0	XCHW RW2, @RW2-d16	XCHW RW1, @RW2	XCHW RW1, @RW2-d16	XCHW RW2, RW2	XCHW RW2, @RW2-d16	XCHW RW3, RW3	XCHW RW3, @RW2	XCHW RW4, RW2	XCHW RW4, @RW2-d16	XCHW RW5, RW2	XCHW RW5, @RW2-d16	XCHW RW6, RW2	XCHW RW6, @RW2-d16	XCHW RW7, RW2	XCHW RW7, @RW2-d16
+B	XCHW RW0, RW0	XCHW RW3, @RW3-d16	XCHW RW1, @RW3	XCHW RW1, @RW3-d16	XCHW RW2, RW2	XCHW RW2, @RW3-d16	XCHW RW3, RW3	XCHW RW3, @RW3	XCHW RW4, RW3	XCHW RW4, @RW3-d16	XCHW RW5, RW3	XCHW RW5, @RW3-d16	XCHW RW6, RW3	XCHW RW6, @RW3-d16	XCHW RW7, RW3	XCHW RW7, @RW3-d16
+C	XCHW RW0, RW0	XCHW RW0, @RW0-RW7	XCHW RW1, @RW0	XCHW RW1, @RW0-RW7	XCHW RW2, RW2	XCHW RW2, @RW0-RW7	XCHW RW3, RW3	XCHW RW3, @RW0+R7	XCHW RW4, RW3	XCHW RW4, @RW0+R7	XCHW RW5, RW3	XCHW RW5, @RW0+R7	XCHW RW6, RW3	XCHW RW6, @RW0+R7	XCHW RW7, RW3	XCHW RW7, @RW0+R7
+D	XCHW RW0, RW0	XCHW RW1, @RW1+R7	XCHW RW1, @RW1	XCHW RW1, @RW1+R7	XCHW RW2, RW2	XCHW RW2, @RW1+R7	XCHW RW3, RW3	XCHW RW3, @RW1+R7	XCHW RW4, RW1	XCHW RW4, @RW1+R7	XCHW RW5, RW1	XCHW RW5, @RW1+R7	XCHW RW6, RW1	XCHW RW6, @RW1+R7	XCHW RW7, RW1	XCHW RW7, @RW1+R7
+E	XCHW RW0, RW0	XCHW RW2, @PC-d16	XCHW RW1, @RW2	XCHW RW1, @PC-d16	XCHW RW2, RW2	XCHW RW2, @PC-d16	XCHW RW3, RW3	XCHW RW3, @RW2	XCHW RW4, RW2	XCHW RW4, @PC-d16	XCHW RW5, RW2	XCHW RW5, @PC-d16	XCHW RW6, RW2	XCHW RW6, @PC-d16	XCHW RW7, RW2	XCHW RW7, @PC-d16
+F	XCHW RW0, RW3	XCHW RW0, addr16	XCHW RW1, @RW3	XCHW RW1, addr16	XCHW RW2, RW2	XCHW RW2, @RW3										

用語索引

Numerics

16 ビットタイマレジスタ

16 ビットタイマレジスタ (TMR)291

16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR)291

16 ビットデータバスモード

外部バス 16 ビットデータバスモードおよびノンマルチプレクス 16 ビット外部バスモードの端子状態142

外部バス 16 ビットデータバスモードおよびマルチプレクス 16 ビット外部バスモードの端子状態140

16 ビット入出力タイマ

16 ビット入出力タイマに関する端子211

16 ビット入出力タイマに関する端子のブロックダイアグラム212

16 ビット入出力タイマの機能208

16 ビット入出力タイマの動作, タイミング227

16 ビット入出力タイマの割込み224

16 ビット入出力タイマの割込みと DMA 転送および EI²OS225

16 ビットリロードタイマ

16 ビットリロードタイマに関する端子284

16 ビットリロードタイマに関する端子のブロックダイアグラム285

16 ビットリロードタイマの設定294

16 ビットリロードタイマの動作モード282

16 ビットリロードタイマのプログラム例302

16 ビットリロードタイマのブロックダイアグラム284

16 ビットリロードタイマのレジスタ構成286

16 ビットリロードタイマの割込み293

16 ビットリロードタイマの割込みと DMA 転送および EI²OS293

16 ビットリロードレジスタ

16 ビットタイマレジスタ (TMR)/16 ビットリロードレジスタ (TMRLR)291

16 ビットリロードレジスタ (TMRLR)292

2M

2M/3M ビットフラッシュメモリの特長458

3M ビットフラッシュメモリ

2M/3M ビットフラッシュメモリの特長458

8/10 ビット A/D コンバータ

8/10 ビット A/D コンバータに関する端子347

8/10 ビット A/D コンバータに関する端子のブロックダイアグラム348

8/10 ビット A/D コンバータの特長346

8/10 ビット A/D コンバータのプログラム例365

8/10 ビット A/D コンバータのブロックダイアグラム347

8/10 ビット A/D コンバータのレジスタ一覧349

8/10 ビット A/D コンバータの割込み356

8/10 ビット A/D コンバータの割込みと DMA 転送および EI²OS356

8/16 ビット PPG タイマ

8/16 ビット PPG タイマに関する端子310

8/16 ビット PPG タイマに関する端子のブロックダイアグラム311

8/16 ビット PPG タイマの機能308

8/16 ビット PPG タイマの動作概略322

8/16 ビット PPG タイマのプログラム例328

8/16 ビット PPG タイマのブロックダイアグラム309

8/16 ビット PPG タイマのレジスタ一覧312

8/16 ビット PPG タイマの割込み320, 326

8/16 ビット PPG タイマの割込みと DMA 転送および EI²OS321

8/16 ビットアップダウンカウンタ / タイマ

8/16 ビットアップダウンカウンタ / タイマに関する端子251

8/16 ビットアップダウンカウンタ / タイマに関する端子のブロックダイアグラム252

8/16 ビットアップダウンカウンタ / タイマの主な機能248

8/16 ビットアップダウンカウンタ / タイマのプログラム例274

8/16 ビットアップダウンカウンタ / タイマのブロックダイアグラム250

8/16 ビットアップダウンカウンタ / タイマのレジスタ一覧253

8/16 ビットアップダウンカウンタ / タイマの割込み264

8/16 ビットアップダウンカウンタ / タイマの割込みと DMA 転送および EI²OS265

8 ビット外部バスモード

外部バス 8 ビットデータバスモードおよびマルチプレクス 8 ビット外部バスモードの端子状態141

8 ビットデータバスモード

外部バス 8 ビットデータバスモードおよびノンマルチプレクス 8 ビット外部バスモードの端子状態143

外部バス 8 ビットデータバスモードおよびマルチプレクス 8 ビット外部バスモードの端子状態141

A**A**

アキュムレータ (A)	29
A/D コンバータ	
8/10 ビット A/D コンバータに関する端子	347
8/10 ビット A/D コンバータに関する端子の ブロックダイアグラム	348
8/10 ビット A/D コンバータの特長	346
8/10 ビット A/D コンバータのプログラム例	365
8/10 ビット A/D コンバータのブロックダイア グラム	347
8/10 ビット A/D コンバータのレジスター一覧	349
8/10 ビット A/D コンバータの割込み	356
8/10 ビット A/D コンバータの割込みと DMA 転送および EI ² OS	356
A/D コンバータの割込み	356
ADB	
バンクセレクトプリフィックス (PCB,DTB,ADB,SPB)	38
ADCR	
データレジスタ (ADCR2,ADCR1)	355
ADCS	
コントロールステータスレジスタ 1(ADCS1)	350
コントロールステータスレジスタ 2(ADCS2)	352
ADER	
アナログ入力許可レジスタ (ADER)	176
ARSR	
自動レディ機能選択レジスタ (ARSR)	155

B**BAP**

バッファアドレスポインタ (BAP)	70, 79
--------------------	--------

C**CALR**

チップセレクトアクティブレベルレジスタ (CALR)	437
-------------------------------	-----

CARx

チップセレクト領域レジスタ (CARx)	435
----------------------	-----

CCR

コンディションコードレジスタ (CCR)	31
----------------------	----

CCRH

カウンタコントロールレジスタ (ch.0) 上位 (CCR0)	254
カウンタコントロールレジスタ (ch.1) 上位 (CCR1)	256

CCRL

カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)	258
--	-----

CDCR

通信プリスケラコントロールレジスタ (CDCR)	405
--------------------------	-----

CKSCR

クロック選択レジスタ (CKSCR) の構成	108
------------------------	-----

CLK 同期モード

CLK 同期モード (動作モード 2) 時の動作	415
--------------------------	-----

CMR

コモンレジスタバンクプリフィックス (CMR)	39
-------------------------	----

CMRx

チップセレクト領域 MASK レジスタ (CMRx)	434
----------------------------	-----

CPCLR

コンペアクリアレジスタ (CPCLR)	214
---------------------	-----

CPU

CPU 間欠動作モード	121, 127
CPU の概略仕様	22
双方向通信時の CPU 間接続	417
マスタ / スレーブ通信時の CPU 間接続	419

CPU 間欠動作モード

CPU 間欠動作モード	121, 127
-------------	----------

CPU 間接続

双方向通信時の CPU 間接続	417
マスタ / スレーブ通信時の CPU 間接続	419

CPU 動作モード

CPU 動作モードと消費電流	120
----------------	-----

CSCR

チップセレクトコントロールレジスタ (CSCR)	436
--------------------------	-----

CSR

カウンタステータスレジスタ 0/1(CSR0/1)	260
---------------------------	-----

D**DCT**

データカウンタ (DCT)	68, 77
---------------	--------

DDR

ポート方向レジスタ (DDR0 ~ DDRA)	173
-------------------------	-----

DIVR

分周比制御レジスタ (DIVR0 ~ DIVR2)	504
---------------------------	-----

DMACS

DMA コントロールステータスレジスタ (DMACS)	69
-----------------------------	----

DMA コントロールステータスレジスタ

DMA コントロールステータスレジスタ (DMACS)	69
-----------------------------	----

DMA ディスクリプタ

DMA ディスクリプタの構成	67
----------------	----

DMA 転送

16 ビット入出力タイマの割込みと DMA 転送 および EI ² OS	225
16 ビットリロードタイマの割込みと DMA 転送 および EI ² OS	293
8/10 ビット A/D コンバータの割込みと DMA 転送および EI ² OS	356

- 8/16 ビット PPG タイマの割込みと DMA 転送
および EI²OS321
- 8/16 ビットアップダウンカウンタ / タイマの
割込みと DMA 転送および EI²OS265
- DMA 転送, および EI²OS 機能への対応
.....226, 265, 293, 321, 337,
356, 381, 408, 506, 549
- DTP/ 外部割込みと DMA 転送および EI²OS
.....337
- I²C インタフェースの割込みと DMA 転送および
EI²OS549
- PWC タイマの割込みと DMA 転送および EI²OS
.....506
- UART の割込みと DMA 転送および EI²OS408
- 拡張 I/O シリアルインタフェースの割込みと
DMA 転送および EI²OS381
- DPR**
ダイレクトページレジスタ (DPR)< 初期値 :01H>
.....36
- DQ3**
セクタ消去タイマフラグ (DQ3) の状態遷移
.....471
- DQ5**
タイミングリミット超過フラグ (DQ5) の
状態遷移470
- DQ6**
トグルビットフラグ (DQ6) の状態遷移469
- DQ7**
データポーリングフラグ (DQ7) の状態遷移
.....468
- DTB**
バンクセレクトプリフィックス
(PCB,DTB,ADB,SPB)38
- DTP**
DTP の動作339
- DTP/ 外部割込み**
DTP/ 外部割込み336
- DTP/ 外部割込みと DMA 転送および EI²OS
.....337
- DTP/ 外部割込みに関する端子332
- DTP/ 外部割込みに関する端子のブロックダイア
グラム333
- DTP/ 外部割込みの概要332
- DTP/ 外部割込みの動作手順340
- DTP/ 外部割込みのプログラム例341
- DTP/ 外部割込みのブロックダイアグラム332
- DTP/ 外部割込みのレジスタ334
- E**
- E²PROM**
E²PROM のメモリマップ448
- システム構成と E²PROM のメモリ構成448
- 修正プログラムを E²PROM に格納する場合の
アドレス一致検出機能の動作450
- EI²OS**
16 ビット入出力タイマの割込みと DMA 転送
および EI²OS225
- 16 ビットリロードタイマの割込みと DMA 転送
および EI²OS293
- 8/10 ビット A/D コンバータの割込みと DMA
転送および EI²OS356
- 8/16 ビット PPG タイマの割込みと DMA 転送
および EI²OS321
- 8/16 ビットアップダウンカウンタ / タイマの
割込みと DMA 転送および EI²OS265
- DMA 転送, および EI²OS 機能への対応
.....226, 265, 293, 321, 337,
356, 381, 408, 506, 549
- DTP/ 外部割込みと DMA 転送および EI²OS
.....337
- I²C インタフェースの割込みと DMA 転送および
EI²OS549
- PWC タイマの割込みと DMA 転送および EI²OS
.....506
- UART の割込みと DMA 転送および EI²OS408
- 拡張 I/O シリアルインタフェースの割込みと
DMA 転送および EI²OS381
- 拡張インテリジェント I/O サービス (EI²OS)
.....74
- 拡張インテリジェント I/O サービス (EI²OS) 処理
時間 (1 回の転送時間)82
- 拡張インテリジェント I/O サービス (EI²OS)
ステータスレジスタ (ISCS)78
- 拡張インテリジェント I/O サービス (EI²OS)
ディスクリプタ (ISD) の構成76
- 拡張インテリジェント I/O サービス (EI²OS) の
処理手順80
- 拡張インテリジェント I/O サービス (EI²OS) の
設定手順81
- 拡張インテリジェント I/O サービス (EI²OS) の
動作75
- EIRR**
割込み /DTP 要因レジスタ (EIRR:External Interrupt
Request Register)335
- ELVR**
要求レベル設定レジスタ (ELVR: External Level
Register)335
- ENIR**
割込み /DTP 許可レジスタ (ENIR:ENable Interrupt
Request Register)334
- EPCR**
バス制御信号選択レジスタ (EPCR)157
- F**
- F²MC-16LX**
F²MC-16LX 命令一覧表591
- FMCS**
フラッシュメモリコントロールステータス
レジスタ (FMCS)458, 460
- H**
- HACR**
外部アドレス出力制御レジスタ (HACR)156

I

I/O ポート	
I/O ポートの機能	170
I/O ポートのレジスタ	171
I/O マップ	
I/O マップ	558
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)	68, 77
I ² C インタフェース	
I ² C インタフェースに関する端子	534
I ² C インタフェースに関する端子のブロック ダイアグラム	534
I ² C インタフェースに関する割込み要因	548
I ² C インタフェースの機能	532
I ² C インタフェースのブロックダイアグラム	533
I ² C インタフェースのレジスタ一覧	535
I ² C インタフェースの割込み制御ビットと 割込み要因	548
I ² C インタフェースの割込みと DMA 転送および EI2OS	549
IADR	
アドレスレジスタ (IADR)	546
IBCR	
バスコントロールレジスタ (IBCR)	538
バスコントロールレジスタ (IBCR) 使用上の注意	543
IBSR	
バスステータスレジスタ (IBSR)	536
ICCR	
クロックコントロールレジスタ (ICCR)	544
ICR	
割込み制御レジスタ (ICR00 ~ ICR15) の機能	48
ICS	
インプットキャプチャコントロールステータス レジスタ (ICS01)	223
IDAR	
データレジスタ (IDAR)	547
ILM	
インタラプトレベルマスクレジスタ (ILM)	32
IOA	
I/O レジスタアドレスポインタ (IOA)	68, 77
IPCP	
インプットキャプチャデータレジスタ (IPCP0,1)	222
ISCS	
拡張インテリジェント I/O サービス (EI2OS) ステータスレジスタ (ISCS)	78
ISD	
拡張インテリジェント I/O サービス (EI2OS) ディスクリプタ (ISD) の構成	76

L

LPMCR	
スタンバイモード遷移のための低消費電力モード 制御レジスタ (LPMCR) へアクセスする際の 注意事項	146
低消費電力モード制御レジスタ (LPMCR)	124
LQFP-100	
外形寸法図 (LQFP-100)	7
端子配列図 (LQFP-100)	10

M

MB90480/485 シリーズ	
MB90480/485 シリーズの特長	2
MB90480/485 シリーズのブロックダイアグラム	5
MB90F481B	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成	484
MB90F482B	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成	484
MB90F488B	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成	484
MB90F489B	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書込み接続の基本構成	484
MD	
モード端子 (MD2 ~ MD0) の設定	149
μDMAC	
μDMAC の機能	64
μDMAC の処理時間 (1 回の転送時間)	72
μDMAC の処理手順	71
μDMAC の動作説明	66
μDMAC のレジスタ一覧	64
μDMAC を使った変換動作	358
タイムベースタイマの割込みと μDMAC	183
単発モードにおける μDMAC の起動例	359
停止モードにおける μDMAC の起動例	361
変換データ保護機能の動作フロー (μDMAC を 使用時)	363
連続モードにおける mDMAC の起動例	360
μPG コントロール / ステータスレジスタ	
μPG コントロール / ステータスレジスタ (PGCSR)	528
μPG タイマ	
μPG タイマに関する端子	526
μPG タイマに関する端子のブロックダイア グラム	527
μPG タイマのタイミングチャート	529
μPG タイマのブロックダイアグラム	526

N

NCC

フラグ変化抑止プリフィックス (NCC)39

O

OCCP

アウトプットコンペアレジスタ (OCCP0 ~ 5)
.....219

OCS

アウトプットコンペアコントロールレジスタ
(OCS01/23/45)220

ODR

ポート出力端子レジスタ (ODR7,ODR4)175

P

PACSR

プログラムアドレス検出制御ステータスレジスタ
(PACSR)443

PADR

プログラムアドレス検出レジスタ
(PADR0,PADR1)445
プログラムアドレス検出レジスタ
(PADR0H,M,L,PADR1H,M,L)446

PC

プログラムカウンタ (PC)34

PCB

バンクセレクトプリフィックス
(PCB,DTB,ADB,SPB)38
プログラムカウンタバンクレジスタ (PCB)
＜初期値: リセットベクタ中の値＞35

PDR

ポートデータレジスタ (PDR0 ~ PDRA)172

PGCSR

μPG コントロール / ステータスレジスタ (PGCSR)
.....528

PLL

PLL クロック通倍率の選択114
メインクロックモード,PLL クロックモード,
サブクロックモード113

PLLOS

PLL 出力選択レジスタ (PLLOS) の構成111

PLL クロック

PLL クロック通倍率の選択114

PLL クロックモード

メインクロックモード,PLL クロックモード,
サブクロックモード113

PLL 出力選択レジスタ

PLL 出力選択レジスタ (PLLOS) の構成111

PPG

PPG0 ~ 5 出力制御レジスタ (PPG01/PPG23/
PPG45)317
PPG カウンタアンダフロー割込み320
PPG 出力動作323

PPG0/2/4 動作モード制御レジスタ

PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)
.....313

PPG0 ~ 5 出力制御レジスタ

PPG0 ~ 5 出力制御レジスタ (PPG01/PPG23/
PPG45)317

PPG1/3/5 動作モード制御レジスタ

PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)
.....315

PPGC

PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)
.....313

PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)
.....315

PPG カウンタアンダフロー割込み

PPG カウンタアンダフロー割込み320

PPG タイマ

8/16 ビット PPG タイマに関する端子310
8/16 ビット PPG タイマに関する端子のブロック
ダイアグラム311
8/16 ビット PPG タイマの機能308
8/16 ビット PPG タイマの動作概略322
8/16 ビット PPG タイマのプログラム例328
8/16 ビット PPG タイマのブロックダイアグラム
.....309
8/16 ビット PPG タイマのレジスタ一覧312
8/16 ビット PPG タイマの割込み320, 326
8/16 ビット PPG タイマの割込みと DMA 転送
および EI²OS321

PRL

リロードレジスタ (PRL0 ~ 5,PRLH0 ~ 5)
.....319

PS

プロセッサステータス (PS)31

PWCR

PWC データバッファレジスタ (PWCR0 ~
PWCR2)503

PWCSR

PWC コントロール / ステータスレジスタ
(PWCSR0 ~ PWCSR2)498

PWC コントロール / ステータスレジスタ

PWC コントロール / ステータスレジスタ
(PWCSR0 ~ PWCSR2)498

PWC タイマ

PWC タイマ動作概略507
PWC タイマに関する端子496
PWC タイマに関する端子のブロックダイア
グラム496
PWC タイマに関する割込み要因505
PWC タイマの機能494
PWC タイマの使用上の注意523
PWC タイマのブロックダイアグラム495
PWC タイマのレジスタ一覧497
PWC タイマの割込み505
PWC タイマの割込みと DMA 転送および EI²OS
.....506

PWC データバッファレジスタ	
PWC データバッファレジスタ (PWCR0 ~ PWCR2)	503

Q

QFP-100	
外形寸法図 (QFP-100)	8
端子配列図 (QFP-100)	9

R

RCR	
リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1)	263
RDR	
ポート入力抵抗レジスタ (RDR0,RDR1)	175
ROMM	
ROM ミラー機能選択レジスタ (ROMM)	455
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	454
ROM ミラー機能選択モジュールのレジスタ	
.....	454
ROM ミラー機能選択レジスタ	
ROM ミラー機能選択レジスタ (ROMM)	455
RP	
レジスタバンクポインタ (RP)	32

S

SCR	
シリアルコントロールレジスタ (SCR)	400
SDCR	
通信プリスケラコントロールレジスタ 0/1 (SDCR0/SDCR1)	380
SDR	
シリアルデータレジスタ 0/1(SDR0/SDR1)	379
SIDR	
シリアルインプット / アウトプットレジスタ (SIDR/SODR)	402
SLP	
STP,SLP,TMD ビットの優先順位	126
SMCS	
シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1)	376
SMR	
シリアルモードレジスタ (SMR)	398
SODR	
シリアルインプット / アウトプットレジスタ (SIDR/SODR)	402
SPB	
バンクセレクトプリフィックス (PCB,DTB,ADB,SPB)	38
SSP	
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)	30

SSR	
シリアルステータスレジスタ (SSR)	403
STP	
STP,SLP,TMD ビットの優先順位	126

T

TBTC	
タイムベースタイマ制御レジスタ (TBTC)	181
TCCS	
タイマカウンタコントロールステータスレジスタ (TCCS)	215
TCDT	
タイマカウンタデータレジスタ (TCDT)	215
TMCSR	
タイマ制御ステータスレジスタ (TMCSR)	287
TMD ビット	
STP,SLP,TMD ビットの優先順位	126
TMR	
16 ビットタイマレジスタ (TMR)	291
16 ビットタイマレジスタ (TMR)/16 ビット	
リロードレジスタ (TMRLR)	291
TMRLR	
16 ビットタイマレジスタ (TMR)/16 ビット	
リロードレジスタ (TMRLR)	291
16 ビットリロードレジスタ (TMRLR)	292

U

UART	
UART に関する端子	396
UART に関する端子のブロックダイアグラム	396
UART に関する割込み要因	407
UART のクロック選択	410
UART の使用上の注意	421
UART の特長	394
UART のプログラム例	422
UART のブロックダイアグラム	395
UART のレジスタ一覧	397
UART の割込み	407
UART の割込みと DMA 転送および EI ² OS	408
UDCR	
UDCR へのデータの書込み	272
アップダウンカウンタレジスタ (ch.0/ch.1)(UDCR0/1)	262
UDER	
アップダウンタイマ入力許可レジスタ (UDER)	176
USP	
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)	30

W

WDTC

ウォッチドッグタイマ制御レジスタ (WDTC)
.....191

WTC

時計タイマ制御レジスタ (WTC) の構成202

あ

アービトレーション

アービトレーション551

アウトプットコンペア

アウトプットコンペアのプログラム例238

アウトプットコンペアのレジスタ一覧219

アウトプットコンペアコントロールレジスタ

アウトプットコンペアコントロールレジスタ
(OCS01/23/45)220

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 (OCCP0 ~ 5)
.....219

アキュムレータ

アキュムレータ (A)29

アクセスモード

アクセスモード148

アクノリッジ

アクノリッジ551

アップ/ダウンカウンタ

リロード/コンペア機能任意幅でのアップ/
ダウンカウンタ270

アップダウンカウンタ/タイマ

8/16 ビットアップダウンカウンタ/タイマに
関する端子251

8/16 ビットアップダウンカウンタ/タイマの
主な機能248

8/16 ビットアップダウンカウンタ/タイマの
プログラム例274

8/16 ビットアップダウンカウンタ/タイマの
ブロックダイアグラム250

8/16 ビットアップダウンカウンタ/タイマの
レジスタ一覧253

8/16 ビットアップダウンカウンタ/タイマの
割込み264

8/16 ビットアップダウンカウンタ/タイマの
割込みと DMA 転送および EI²OS265

アップダウンカウンタレジスタ

アップダウンカウンタレジスタ (ch.0/ch.1)
(UDCR0/1)262

アップダウンタイマ入力許可レジスタ

アップダウンタイマ入力許可レジスタ (UDER)
.....176

アドレス一致検出機能

アドレス一致検出機能の概要440

アドレス一致検出機能の動作447

アドレス一致検出機能のプログラム例452

アドレス一致検出機能のブロックダイアグラム
.....441

アドレス一致検出機能のレジスタと初期値の一覧
.....442

修正プログラムを E²PROM に格納する場合の
アドレス一致検出機能の動作450

アドレス生成

アドレス生成の種類23

アドレスレジスタ

アドレスレジスタ (IADR)546

アドレッシング

アドレッシング	551, 568
間接アドレッシング	577
直接アドレッシング	570

アナログ入力許可レジスタ

アナログ入力許可レジスタ (ADER)	176
---------------------------	-----

アナログ入力端子

アナログ入力端子の扱い	364
-------------------	-----

アルゴリズム

自動アルゴリズム終了タイミング	462
-----------------------	-----

い

一時停止

フラッシュメモリのセクタ消去を一時停止する 方法	479
-----------------------------------	-----

イベントカウントモード

イベントカウントモード (外部クロックモード)	282
イベントカウントモードの動作	300

インターバルタイマ

インターバルタイマ機能	178
インターバルタイマ機能 (タイムベースタイマ) の動作	184

インターバル割込み

時計タイマのインターバル割込み機能	204
-------------------------	-----

インタラプトレベルマスキレジスタ

インタラプトレベルマスキレジスタ (ILM)	32
------------------------------	----

インプットキャプチャ

インプットキャプチャの取込みタイミング例	231
インプットキャプチャのプログラム例	244
インプットキャプチャのレジスタ一覧	222

インプットキャプチャコントロールステータス レジスタ

インプットキャプチャコントロールステータス レジスタ (ICS01)	223
---	-----

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 (IPCP0,1)	222
--------------------------------------	-----

う

ウォッチドッグタイマ

ウォッチドッグタイマ機能	190
ウォッチドッグタイマ使用上の注意	197
ウォッチドッグタイマのクロックソースの 指定機能	205
ウォッチドッグタイマの動作	195
ウォッチドッグタイマのプログラム例	198
ウォッチドッグタイマのブロックダイアグラム	193

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)	191
----------------------------------	-----

え

エラー

バスエラー	551
-------------	-----

か

外形寸法図

外形寸法図 (LQFP-100)	7
外形寸法図 (QFP-100)	8

外部アドレス出力制御レジスタ

外部アドレス出力制御レジスタ (HACR)	156
-----------------------------	-----

外部クロック

振動子と外部クロックの接続	118
---------------------	-----

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)	282
----------------------------------	-----

外部シフトクロックモード

外部シフトクロックモード	383
--------------------	-----

外部端子

各モードにおける外部端子の動作	152
-----------------------	-----

外部トリガ

外部トリガ / 内部タイマで起動する場合の注意	364
----------------------------------	-----

外部バス

外部バス 16 ビットデータバスモードおよびノン マルチプレクス 16 ビット外部バスモードの 端子状態	142
外部バス 16 ビットデータバスモードおよび マルチプレクス 16 ビット外部バスモードの 端子状態	140
外部バス 8 ビットデータバスモードおよびノンマ ルチプレクス 8 ビット外部バスモードの端子 状態	143
外部バス 8 ビットデータバスモードおよびマルチ プレクス 8 ビット外部バスモードの端子状態	141

外部メモリ

外部メモリアクセスの入出力信号端子	153
-------------------------	-----

外部メモリアクセス制御信号

外部メモリアクセス制御信号	160
---------------------	-----

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	96
---------------------------	----

外部割込み

外部割込みの動作	338
外部割込み要求レベル	340

カウンタ

カウンタ動作のモード	283
------------------	-----

カウンタアンダフロー

PPG カウンタアンダフロー割込み	320
-------------------------	-----

カウンタコントロールレジスタ

カウンタコントロールレジスタ (ch.0) 上位 (CCRHO)	254
カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)	258
カウンタコントロールレジスタ (ch.1) 上位 (CCRHI)	256

カウンタステータスレジスタ	
カウンタステータスレジスタ 0/1(CSR0/1)	260
カウント	
カウント動作の状態遷移	295
カウントクリア	
カウントクリア / ゲート機能	272
カウントクリア / ゲート機能	
カウントクリア / ゲート機能	272
カウントクロック	
カウントクロックと最大周期	515
カウントクロックの選択	324, 510
カウントタイミグ	
フリーランタイマのカウントタイミグ	232
カウント方向転換フラグ	
カウント方向フラグ, カウント方向転換フラグ	273
カウント方向フラグ	
カウント方向フラグ, カウント方向転換フラグ	273
カウントモード	
カウントモードの選択	266
書き込み	
書き込み / チップセクタ消去時	469, 470
書き込み動作時	468
富士通標準シリアルオンボード書き込みに使用する端子	485
フラッシュメモリの書き込み / 消去	472
フラッシュメモリの書き込み / 消去の方法	458
フラッシュメモリの書き込み手順	475
フラッシュメモリへのデータ書き込み	474
拡張 I/O シリアルインタフェース	
拡張 I/O シリアルインタフェースに関する端子	373
拡張 I/O シリアルインタフェースに関する端子のブロックダイアグラム	374
拡張 I/O シリアルインタフェースに関する割込み要因	381
拡張 I/O シリアルインタフェースの概要	372
拡張 I/O シリアルインタフェースの動作概要	382
拡張 I/O シリアルインタフェースのプログラム例	389
拡張 I/O シリアルインタフェースのブロックダイアグラム	373
拡張 I/O シリアルインタフェースのレジスタ一覧	375
拡張 I/O シリアルインタフェースの割込み	381
拡張 I/O シリアルインタフェースの割込み機能	388
拡張 I/O シリアルインタフェースの割込みと DMA 転送および EI ² OS	381
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 ² OS)	74
拡張インテリジェント I/O サービス (EI ² OS) 処理時間 (1 回の転送時間)	82
拡張インテリジェント I/O サービス (EI ² OS) ステータスレジスタ (ISCS)	78
拡張インテリジェント I/O サービス (EI ² OS) ディスクリプタ (ISD) の構成	76
拡張インテリジェント I/O サービス (EI ² OS) の処理手順	80
拡張インテリジェント I/O サービス (EI ² OS) の設定手順	81
拡張インテリジェント I/O サービス (EI ² OS) の動作	75
間接アドレッシング	
間接アドレッシング	577
き	
起動例	
単発モードにおける μ DMAC の起動例	359
停止モードにおける μ DMAC の起動例	361
連続モードにおける μ DMAC の起動例	360
機能	
カウントクリア / ゲート機能	272
リロード / コンペア機能任意幅でのアップ / ダウンカウント	270
リロード / コンペア機能の選択	269
基本構成	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書き込み接続の基本構成	484
キャプチャタイミグ	
入力信号に対するキャプチャタイミグ	234
く	
クリアタイミグ	
フリーランタイマのクリアタイミグ (コンペアレジスタ 0 との一致)	232
クロック	
PLL クロック通倍率の選択	114
UART のクロック選択	410
イベントカウントモード (外部クロックモード)	282
ウォッチドッグタイマのクロックソースの指定機能	205
外部シフトクロックモード	383
カウントクロックと最大周期	515
カウントクロックの選択	324, 510
クロック供給機能	178, 185
クロック供給マップ	105
クロックの概要	104
クロックモード	121
クロックモードの移行	113
クロックモードの切換え	145
サブクロックモード使用時の注意点	462
サブクロックモード設定方法	463
振動子と外部クロックの接続	118
内部クロックモード	282
内部クロックモード (リロードモード) の動作	296
内部クロックモード (ワンショットモード) の動作	298
内部シフトクロックモード	383

発振クロック周波数とシリアルクロック入力周波数について	486
マシクロック	114
メインクロックモード,PLL クロックモード, サブクロックモード	113
クロック供給マップ	
クロック供給マップ	105
クロックコントロールレジスタ	
クロックコントロールレジスタ (ICCR)	544
クロック選択レジスタ	
クロック選択レジスタ (CKSCR) の構成	108
クロックソース	
ウォッチドッグタイマのクロックソースの 指定機能	205
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	106
クロックモード	
外部シフトクロックモード	383
クロックモード	121
クロックモードの移行	113
クロックモードの切り換え	145
内部シフトクロックモード	383
け	
ゲート機能	
カウンタクリア / ゲート機能	272
検出アドレス	
検出アドレスの設定方法	447
こ	
コマンドシーケンス表	
コマンドシーケンス表	466
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)	39
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)	31
コントロールステータスレジスタ	
コントロールステータスレジスタ 1(ADCS1)	350
コントロールステータスレジスタ 2(ADCS2)	352
コンペア機能	
コンペア機能	270
リロード / コンペア機能任意幅でのアップ / ダウンカウント	270
リロード / コンペア機能の選択	269
コンペアクリアレジスタ	
コンペアクリアレジスタ (CPCLR)	214
コンペアレジスタ	
フリーランタイマのクリアタイミング (コンペア レジスタ 0 との一致)	232

さ

再開	
フラッシュメモリのセクタ消去を再開する方法	480
再起動	
再起動	512
最小入力パルス幅	
最小入力パルス幅	521
最大周期	
カウントクロックと最大周期	515
サブクロック	
サブクロックの発振安定時間待ち機能	205
サブクロックモード	
サブクロックモード使用時の注意点	462
サブクロックモード設定方法	463
メインクロックモード,PLL クロックモード, サブクロックモード	113

し

システム構成	
システム構成と E ² PROM のメモリ構成	448
フラッシュマイコンプログラムのシステム構成 (横河ディジタルコンピュータ株式会社製)	487
システムスタックポインタ	
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)	30
実効アドレス	
実効アドレスフィールド	586
実効アドレスフィールド	
実効アドレスフィールド	569
実行サイクル	
実行サイクル数	584
実行サイクル数計算方法	584
自動アルゴリズム	
自動アルゴリズム終了タイミング	462
自動レディ機能選択レジスタ	
自動レディ機能選択レジスタ (ARSR)	155
シフトクロックモード	
外部シフトクロックモード	383
内部シフトクロックモード	383
シフト動作	
シフト動作のスタート / ストップタイミングと 入出力のタイミング	386
修正プログラム	
修正プログラムを E ² PROM に格納する場合の アドレス一致検出機能の動作	450
周辺装置	
外部に接続する周辺装置の条件	340
出力波形例	
出力波形例	229
消去	
セクタ消去の手順	477
フラッシュメモリの書き込み / 消去	472
フラッシュメモリの書き込み / 消去の方法	458

フラッシュメモリのセクタ消去を一時停止する 方法	479
フラッシュメモリのセクタ消去を再開する方法	480
フラッシュメモリの全データを消去する方法 (チップ消去)	476
フラッシュメモリの任意のデータを消去する方法 (セクタ消去)	477
消去一時停止時	
セクタ消去一時停止時	468, 469, 471
消去時	
書き込み/チップセクタ消去時	469, 470
消去動作時	
セクタ消去動作時	471
状態遷移	
カウント動作の状態遷移	295
セクタ消去タイムフラグ (DQ3) の状態遷移	471
タイミングリミット超過フラグ (DQ5) の 状態遷移	470
データポーリングフラグ (DQ7) の状態遷移	468
トグルビットフラグ (DQ6) の状態遷移	469
状態遷移図	
状態遷移図	137
消費電流	
CPU 動作モードと消費電流	120
シリアル I/O	
シリアル I/O の動作状態	384
シリアルインプット/アウトプットレジスタ	
シリアルインプット/アウトプットレジスタ (SIDR/SODR)	402
シリアルオンボード	
富士通標準シリアルオンボード書き込みに使用する 端子	485
シリアル書き込み接続	
MB90F481B, MB90F482B, MB90F488B, MB90F489B シリアル書き込み接続の基本構成	484
シリアル書き込み接続例	
シリアル書き込み接続例	487
シリアルクロック	
発振クロック周波数とシリアルクロック入力周波 数について	486
シリアルコントロールレジスタ	
シリアルコントロールレジスタ (SCR)	400
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 (SSR)	403
シリアルデータ	
シリアルデータ転送中の動作	387
シリアルデータレジスタ	
シリアルデータレジスタ 0/1 (SDR0/SDR1)	379
シリアルモードコントロールステータスレジスタ	
シリアルモードコントロールステータスレジスタ 0/1 (SMCS0/SMCS1)	376
シリアルモードレジスタ	
シリアルモードレジスタ (SMR)	398

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 (ユーザ電源使用時)	488
シングルチップモード時の端子状態	139
振動子	
振動子と外部クロックの接続	118
す	
推奨例	
モード端子とモードデータの関係 (推奨例)	151
スタートコンディション	
スタートコンディション	550
スタック	
スタック領域	85
割込み処理開始時のスタック動作	85
割込み処理復帰のスタック動作	85
スタンバイモード	
スタンバイモード	121
スタンバイモード時の動作状態	128
スタンバイモード遷移のための低消費電力モード 制御レジスタ (LPMCR) へアクセスする際の 注意事項	146
スタンバイモードの割込みによる解除	144
スタンバイモードへの移行と割込み	144
ステータスレジスタ	
拡張インテリジェント I/O サービス (EI ² OS) ステータスレジスタ (ISCS)	78
ストップコンディション	
ストップコンディション	550
ストップモード	
ストップモードの解除	136
ストップモードへの遷移	135
スリープモード	
スリープモードの解除	130
スリープモードへの遷移	129
スレーブ	
マスタ/スレーブ通信機能の通信手順	420
マスタ/スレーブ通信時の CPU 間接続	419
マスタ/スレーブ通信時の各レジスタの設定	419
せ	
セキュリティ	
セキュリティ許可中の動作	481
セキュリティの解除	481
セキュリティのかけ方	481
セクタ	
セクタ構成	459
セクタ消去一時停止時	468, 469, 471
セクタ消去動作時	471
チップ/セクタ消去動作時	468
セクタ消去	
セクタ消去の手順	477

用語索引

フラッシュメモリのセクタ消去を 一時停止する方法	479
フラッシュメモリのセクタ消去を 再開する方法	480
フラッシュメモリの任意のデータを消去する方法 (セクタ消去)	477
セクタ消去タイマフラグ セクタ消去タイマフラグ (DQ3) の状態遷移	471
接続例 シリアル書き込み接続例	487
シングルチップモード時の接続例 (ユーザ電源 使用時)	488
フラッシュマイコンプログラマとの接続例 (ユーザ電源使用時)	490
専用レジスタ 専用レジスタ	27
そ	
双方向通信機能 双方向通信機能の通信手順	418
双方向通信時 双方向通信時における各レジスタの設定	417
双方向通信時の CPU 間接続	417
測定モード 測定モードとカウンタ動作	519
ソフトウェア割込み ソフトウェア割込みからの復帰	62
ソフトウェア割込みに関する注意事項	63
ソフトウェア割込みの起動	62
ソフトウェア割込みの動作	63
た	
タイマカウンタコントロールステータスレジスタ タイマカウンタコントロールステータス レジスタ (TCCS)	215
タイマカウンタデータレジスタ タイマカウンタデータレジスタ (TCDT)	215
タイマ周期 タイマ周期	515
タイマ制御ステータスレジスタ タイマ制御ステータスレジスタ (TMCSR)	287
タイマ値 タイマ値とリロード値	514
タイミング インプットキャプチャの取込みタイミング例	231
自動アルゴリズム終了タイミング	462
シフト動作のスタート/ストップタイミングと 入出力のタイミング	386
出力端子の変化タイミング	233
タイミングに関する注意	529
入力信号に対するキャプチャタイミング	234
フリーランタイマのカウントタイミング	232
フリーランタイマのクリアタイミング (コンペアレジスタ 0 との一致)	232

リロードレジスタへの書き込みタイミング	326
割込みタイミング	233
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) の 状態遷移	470
タイムベースタイマ インターバルタイマ機能 (タイムベースタイマ) の動作	184
タイムベースタイマ使用上の注意	187
タイムベースタイマの動作例	186
タイムベースタイマのプログラム例	188
タイムベースタイマのブロックダイアグラム	179
タイムベースタイマの割込み	183
タイムベースタイマの割込みと μ DMAC	183
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)	181
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	132
タイムベースタイマモードへの遷移	131
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)< 初期値 :01H>	36
多重割込み 多重割込み動作	58
多重割込みの例	59
多バイト長データ メモリ空間における多バイト長データの配置	25
端子機能 端子機能	11
端子配列図 端子配列図 (LQFP-100)	10
端子配列図 (QFP-100)	9
単発測定 単発測定と連続測定	517
単発モード 単発モードにおける μ DMAC の起動例	359
ち	
遅延割込み 遅延割込み発生モジュールの使用上の注意 (遅延割込み要求ラッチ)	90
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使用上の注意 (遅延割込み要求ラッチ)	90
遅延割込み発生モジュールの動作	90
遅延割込み発生モジュールのブロックダイア グラム	89
遅延割込み発生モジュールのレジスタ一覧	89
チップ チップ/セクタ消去動作時	468
チップ消去 フラッシュメモリの全データを消去する方法 (チップ消去)	476

チップセレクトアクティブレベルレジスタ	
チップセレクトアクティブレベルレジスタ (CALR)	437
チップセレクト機能	
チップセレクト機能に関する端子	431
チップセレクト機能に関する端子のブロック	
ダイアグラム	432
チップセレクト機能の概要	430
チップセレクト機能の使用上の注意点	438
チップセレクト機能の使用例	438
チップセレクト機能のブロックダイアグラム	
.....	431
チップセレクト機能のレジスタ一覧	433
チップセレクトコントロールレジスタ	
チップセレクトコントロールレジスタ (CSCR)	436
チップセレクト領域 MASK レジスタ	
チップセレクト領域 MASK レジスタ (CMRx)	434
チップセレクト領域レジスタ	
チップセレクト領域レジスタ (CARx)	435
直接アドレッシング	
直接アドレッシング	570
つ	
通信プリスケラコントロールレジスタ	
通信プリスケラコントロールレジスタ (CDCR)	405
通信プリスケラコントロールレジスタ 0/1 (SDCR0/SDCR1)	380
て	
停止モード	
停止モードにおける μ DMA の起動例	361
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	122
低消費電力モード	
低消費電力モードの動作状態	138
低消費電力モード制御レジスタ	
スタンバイモード遷移のための低消費電力モード	
制御レジスタ (LPMCR) へアクセスする際の	
注意事項	146
低消費電力モード制御レジスタ (LPMCR)	124
低消費電力モード制御レジスタへのアクセス	
.....	126
ディスクリプタ	
DMA ディスクリプタの構成	67
拡張インテリジェント I/O サービス (EI ² OS)	
ディスクリプタ (ISD) の構成	76
逓倍率	
PLL クロック逓倍率の選択	114
データカウンタ	
データカウンタ (DCT)	68, 77

データポーリングフラグ	
データポーリングフラグ (DQ7) の状態遷移	468
データレジスタ	
データレジスタ (ADCR2, ADCR1)	355
データレジスタ (IDAR)	547
デバイス	
デバイス取扱い上の注意	19
電源	
電源に関する取扱い上の注意	20
転送	
16 ビット入出力タイマの割込みと DMA 転送	
および EI ² OS	225
16 ビットリロードタイマの割込みと DMA 転送	
および EI ² OS	293
8/10 ビット A/D コンバータの割込みと DMA	
転送および EI ² OS	356
8/16 ビット PPG タイマの割込みと DMA 転送	
および EI ² OS	321
8/16 ビットアップダウンカウンタ / タイマの	
割込みと DMA 転送および EI ² OS	265
DMA 転送, および EI ² OS 機能への対応	
.....	226, 265, 293, 321, 337,
356, 381, 408, 506, 549	
DTP/ 外部割込みと DMA 転送および EI ² OS	337
I ² C インタフェースの割込みと DMA 転送および	
EI ² OS	549
μ DMA の処理時間 (1 回の転送時間)	72
PWC タイマの割込みと DMA 転送および EI ² OS	
.....	506
UART の割込みと DMA 転送および EI ² OS	408
拡張 I/O シリアルインタフェースの割込みと	
DMA 転送および EI ² OS	381
拡張インテリジェント I/O サービス (EI ² OS) 処理	
時間 (1 回の転送時間)	82
シリアルデータ転送中の動作	387
と	
動作モード	
16 ビットリロードタイマの動作モード	282
CLK 同期モード (動作モード 2) 時の動作	415
動作モード	148, 322, 357, 409
動作モードの選択	511
非同期モード (動作モード 0,1) 時の動作	412
トグルビットフラグ	
トグルビットフラグ (DQ6) の状態遷移	469
時計カウンタ	
時計カウンタ	204
時計タイマ	
時計タイマのインターバル割込み機能	204
時計タイマの機能	200
時計タイマのブロックダイアグラム	201
時計タイマ制御レジスタ	
時計タイマ制御レジスタ (WTC) の構成	202
時計モード	
時計モードの解除	134
時計モードへの遷移	133

取込みタイミング

インプットキャプチャの取込みタイミング例	231
----------------------	-----

な

内部クロックモード

内部クロックモード	282
内部クロックモード (リロードモード) の動作	296
内部クロックモード (ワンショットモード) の動作	298

内部シフトクロックモード

内部シフトクロックモード	383
--------------	-----

内部タイマ

外部トリガ / 内部タイマで起動する場合の注意	364
-------------------------	-----

に

入出力回路形式

入出力回路形式	16
---------	----

入出力信号端子

外部メモリアクセスの入出力信号端子	153
-------------------	-----

入出力タイマ

16 ビット入出力タイマに関する端子	211
16 ビット入出力タイマに関する端子のブロックダイアグラム	212
16 ビット入出力タイマの機能	208
16 ビット入出力タイマの動作, タイミング	227
16 ビット入出力タイマの割込み	224
16 ビット入出力タイマの割込みと DMA 転送および EI ² OS	225

の

ノンマルチプレクス

外部バス 16 ビットデータバスモードおよびノンマルチプレクス 16 ビット外部バスモードの端子状態	142
外部バス 8 ビットデータバスモードおよびノンマルチプレクス 8 ビット外部バスモードの端子状態	143
ノンマルチプレクスモード	167

ノンマルチプレクスモード

ノンマルチプレクスモード	167
--------------	-----

は

ハードウェアコンポーネント

ハードウェアコンポーネントの初期値	326
-------------------	-----

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	467
----------------	-----

ハードウェア割込み

ハードウェア割込みからの復帰	54
ハードウェア割込み処理時間	60
ハードウェア割込みの起動	54

ハードウェア割込みの機能	51
ハードウェア割込みの構造	52
ハードウェア割込みの使用手順	57
ハードウェア割込みの動作	55
ハードウェア割込みの動作フロー	56
ハードウェア割込み抑止	52

バスエラー

バスエラー	551
-------	-----

バスコントロールレジスタ

バスコントロールレジスタ (IBCR)	538
バスコントロールレジスタ (IBCR) 使用上の注意	543

バスステータスレジスタ

バスステータスレジスタ (IBSR)	536
--------------------	-----

バス制御信号選択レジスタ

バス制御信号選択レジスタ (EPCR)	157
---------------------	-----

バスモード

バスモード	148
-------	-----

バスモード設定ビット

バスモード設定ビット (M1,M0)	150
--------------------	-----

パッケージ

対応品種のパッケージ	4
------------	---

発振安定時間待ち

サブクロックの発振安定時間待ち機能	205
-------------------	-----

発振安定待ち

発振安定待ちリセット状態	95
--------------	----

発振安定待ち時間

発振安定待ち時間	117, 145
リセット要因と発振安定待ち時間	94

発振安定待ち時間用タイマ

発振安定待ち時間用タイマ機能	184
----------------	-----

発振クロック

発振クロック周波数とシリアルクロック入力周波数について	486
-----------------------------	-----

パッチ処理フロー

プログラム修正のパッチ処理フロー	451
------------------	-----

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)	70, 79
--------------------	--------

パルス

パルスの端子出力制御	325
------------	-----

パルス幅

最小入力パルス幅	521
リロード値とパルス幅の関係	323

パルス幅 / 周期カウント

パルス幅 / 周期カウントの範囲	521
------------------	-----

パルス幅 / 周期算出方法

パルス幅 / 周期算出方法	521
---------------	-----

パルス幅測定

タイマ / パルス幅測定の起動と停止	512
パルス幅測定機能の動作	509
パルス幅測定の動作フロー	522

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)	38
----------------------------------	----

汎用レジスタ

汎用レジスタ	28
--------	----

汎用レジスタ (レジスタバンク)	37
ひ	
ビット	
STP,SLP,TMD ビットの優先順位	126
ビットフラッシュメモリ	
2M/3M ビットフラッシュメモリの特長	458
非同期モード	
非同期モード (動作モード 0,1) 時の動作	412
品種	
品種構成	3
ふ	
フェッチ	
モードフェッチ	97
富士通標準	
富士通標準シリアルオンボード書込みに使用する端子	485
フラグ	
タイミングリミット超過フラグ (DQ5) の状態遷移	470
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)	39
フラッシュマイコンプログラマ	
フラッシュマイコンプログラマとの接続例 (ユーザ電源使用時)	490
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)	487
フラッシュメモリ	
2M/3M ビットフラッシュメモリの特長	458
フラッシュメモリの書込み / 消去	472
フラッシュメモリの書込み / 消去の方法	458
フラッシュメモリの書込み手順	475
フラッシュメモリのセクタ消去を一時停止する方法	479
フラッシュメモリのセクタ消去を再開する方法	480
フラッシュメモリの全データを消去する方法 (チップ消去)	476
フラッシュメモリの任意のデータを消去する方法 (セクタ消去)	477
フラッシュメモリへのデータ書込み	474
フラッシュメモリを読み出し / リセット状態にする方法	473
フラッシュメモリコントロールステータスレジスタ	
フラッシュメモリコントロールステータスレジスタ (FMCS)	458, 460
フリーランタイム	
フリーランタイムのカウントタイミング	232
フリーランタイムのクリアタイミング (コンペアレジスタ 0 との一致)	232
フリーランタイムの動作	228
フリーランタイムのプログラム例	235
フリーランタイムのレジスタ一覧	214
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)	39
バンクセレクトプリフィックス (PCB,DTB,ADB,SPB)	38
フラグ変化抑止プリフィックス (NCC)	39
連続したプリフィックスコード	40
割込み抑止命令とプリフィックス命令に関する制約	40
プログラム	
修正プログラムを E ² PROM に格納する場合のアドレス一致検出機能の動作	450
プログラム修正のパッチ処理フロー	451
プログラムの実行	447
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)	443
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 (PADR0,PADR1)	445
プログラムアドレス検出レジスタ (PADR0H,M,L,PADR1H,M,L)	446
プログラムカウンタ	
プログラムカウンタ (PC)	34
プログラムカウンタバンクレジスタ	
プログラムカウンタバンクレジスタ (PCB) < 初期値 : リセットベクタ中の値 >	35
プログラム例	
16 ビットリロードタイマのプログラム例	302
8/10 ビット A/D コンバータのプログラム例	365
8/16 ビット PPG タイマのプログラム例	328
8/16 ビットアップダウンカウンタ / タイマのプログラム例	274
DTP / 外部割込みのプログラム例	341
UART のプログラム例	422
アウトプットコンペアのプログラム例	238
アドレス一致検出機能のプログラム例	452
インプットキャプチャのプログラム例	244
ウォッチドッグタイマのプログラム例	198
拡張 I/O シリアルインタフェースのプログラム例	389
タイムベースタイマのプログラム例	188
フリーランタイムのプログラム例	235
割込み処理のプログラム例	87
プロセッサステータス	
プロセッサステータス (PS)	31
ブロックダイアグラム	
16 ビット入出力タイマに関する端子のブロックダイアグラム	212
16 ビットリロードタイマに関する端子のブロックダイアグラム	285
16 ビットリロードタイマのブロックダイアグラム	284
8/10 ビット A/D コンバータに関する端子のブロックダイアグラム	348
8/10 ビット A/D コンバータのブロックダイアグラム	347

8/16 ビット PPG タイマに関する端子のブロック ダイアグラム	311
8/16 ビット PPG タイマのブロックダイアグラム	309
8/16 ビットアップダウンカウンタ / タイマに 関する端子のブロックダイアグラム	252
8/16 ビットアップダウンカウンタ / タイマの ブロックダイアグラム	250
DTP/ 外部割込みに関する端子のブロック ダイアグラム	333
DTP/ 外部割込みのブロックダイアグラム	332
I2C インタフェースに関する端子のブロック ダイアグラム	534
I ² C インタフェースのブロックダイアグラム	533
MB90480/485 シリーズのブロック ダイアグラム	5
μPG タイマに関する端子のブロック ダイアグラム	527
μPG タイマのブロックダイアグラム	526
PWC タイマに関する端子のブロック ダイアグラム	496
PWC タイマのブロックダイアグラム	495
ROM ミラー機能選択モジュールのブロック ダイアグラム	454
UART に関する端子のブロックダイアグラム	396
UART のブロックダイアグラム	395
アドレス一致検出機能のブロックダイアグラム	441
ウォッチドッグタイマのブロックダイアグラム	193
外部リセット端子のブロックダイアグラム	96
拡張 I/O シリアルインタフェースに関する端子の ブロックダイアグラム	374
拡張 I/O シリアルインタフェースのブロック ダイアグラム	373
クロック発生部のブロックダイアグラム	106
タイムベースタイマのブロックダイア グラム	179
遅延割込み発生モジュールのブロックダイ グラム	89
チップセレクト機能に関する端子のブロック ダイアグラム	432
チップセレクト機能のブロックダイアグラム	431
低消費電力制御回路のブロックダイアグラム	122
時計タイマのブロックダイアグラム	201
ブロックダイアグラム	153, 209
分周比制御レジスタ 分周比制御レジスタ (DIVR0 ~ DIVR2)	504
へ	
変化タイミング 出力端子の変化タイミング	233
変換 μDMAC を使った変換動作	358

変換データ保護機能 変換データ保護機能	362
変換データ保護機能の使用上の注意	362
変換データ保護機能の動作フロー (μDMAC を 使用時)	363

ほ

ポインタ I/O レジスタアドレスポインタ (IOA)	68, 77
バッファアドレスポインタ (BAP)	70, 79
ポート出力端子レジスタ ポート出力端子レジスタ (ODR7, ODR4)	175
ポートデータレジスタ ポートデータレジスタ (PDR0 ~ PDRA)	172
ポート入力抵抗レジスタ ポート入力抵抗レジスタ (RDR0, RDR1)	175
ポート方向レジスタ ポート方向レジスタ (DDR0 ~ DDRA)	173
ホールド機能 ホールド機能の動作	166
保護機能 変換データ保護機能	362
変換データ保護機能の使用上の注意	362
変換データ保護機能の動作フロー (μDMAC を 使用時)	363

ま

マシクロック マシクロック	114
マスタ マスタ / スレーブ通信機能の通信手順	420
マスタ / スレーブ通信時の CPU 間接続	419
マスタ / スレーブ通信時の各レジスタの設定	419
マスタ / スレーブ通信機能 マスタ / スレーブ通信機能の通信手順	420
マスタ / スレーブ通信時 マスタ / スレーブ通信時の CPU 間接続	419
マスタ / スレーブ通信時の各レジスタの設定	419
マルチプレクス 外部バス 16 ビットデータバスモードおよびマル チプレクス 16 ビット外部バスモードの端子 状態	140
外部バス 8 ビットデータバスモードおよびマルチ プレクス 8 ビット外部バスモードの端子状態	141
マルチプレクスモード	167
マルチプレクスモード マルチプレクスモード	167

み

未定義命令 未定義命令の実行による例外処理	84
--------------------------------	----

め

命令

未定義命令の実行による例外処理	84
命令の種類	567
命令マップの構造	605
割込み抑止命令	40
割込み抑止命令とプリフィックス命令に関する 制約	40

命令一覧表

F ² MC-16LX 命令一覧表	591
------------------------------------	-----

メインクロックモード

メインクロックモード, PLL クロックモード, サブクロックモード	113
---	-----

メモリ空間

メモリ空間	554
メモリ空間における多バイト長データの配置	25

メモリ構成

システム構成と E ² PROM のメモリ構成	448
--	-----

メモリマップ

E ² PROM のメモリマップ	448
メモリマップ	23

も

モード

16 ビットリロードタイマの動作モード	282
CLK 同期モード (動作モード 2) 時の動作	415
CPU 間欠動作モード	121, 127
CPU 動作モードと消費電流	120
アクセスモード	148
イベントカウントモード (外部クロックモード)	282
イベントカウントモードの動作	300
外部シフトクロックモード	383
外部バス 16 ビットデータバスモードおよびノ ンマルチプレクス 16 ビット外部バスモードの 端子状態	142
外部バス 16 ビットデータバスモードおよびマル チプレクス 16 ビット外部バスモードの端子 状態	140
外部バス 8 ビットデータバスモードおよびノ ンマルチプレクス 8 ビット外部バスモードの端子 状態	143
外部バス 8 ビットデータバスモードおよびマルチ プレクス 8 ビット外部バスモードの端子状態	141
カウンタ動作のモード	283
カウントモードの選択	266
各モードにおける外部端子の動作	152
クロックモード	121
クロックモードの移行	113
クロックモードの切り換え	145
サブクロックモード使用時の注意点	462
サブクロックモード設定方法	463
シングルチップモード時の接続例 (ユーザ電源 使用時)	488

スタンバイモード	121
スタンバイモード時の動作状態	128
スタンバイモード遷移のための低消費電力モード 制御レジスタ (LPMCR) ヘアクセスする際の 注意事項	146
スタンバイモードの割込みによる解除	144
スタンバイモードへの移行と割込み	144
ストップモードの解除	136
ストップモードへの遷移	135
スリープモードの解除	130
スリープモードへの遷移	129
測定モードとカウンタ動作	519
タイムベースタイマモードの解除	132
タイムベースタイマモードへの遷移	131
単発モードにおける μ DMAC の起動例	359
停止モードにおける μ DMAC の起動例	361
低消費電力モードの動作状態	138
動作モード	148, 322, 357, 409
動作モードの選択	511
時計モードの解除	134
時計モードへの遷移	133
内部クロックモード	282
内部クロックモード (リロードモード) の動作	296
内部クロックモード (ワンショットモード) の 動作	298
内部シフトクロックモード	383
ノンマルチプレクスモード	167
バスモード	148
非同期モード (動作モード 0,1) 時の動作	412
マルチプレクスモード	167
メインクロックモード, PLL クロックモード, サブクロックモード	113
モード端子	97
モード端子 (MD2 ~ MD0) の設定	149
モード端子とモードデータの関係 (推奨例)	151
モードデータ	150
モードデータ読出し後の端子の状態	101
モードの種類	159
モードフェッチ	97
モード分類	148
リロード動作モード	514
連続モードにおける μ DMAC の起動例	360
ワンショット動作モード	514

モード設定ビット

各種モード設定ビット (S1, S0)	150
---------------------------	-----

モード端子

モード端子	97
モード端子 (MD2 ~ MD0) の設定	149
モード端子とモードデータの関係 (推奨例)	151

モードデータ

モード端子とモードデータの関係 (推奨例)	151
モードデータ	150
モードデータ読出し後の端子の状態	101

モードフェッチ	
モードフェッチ	97

ゆ

ユーザスタックポインタ	
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)	30
ユーザ電源使用時	
シングルチップモード時の接続例 (ユーザ電源	
使用時)	488
フラッシュマイコンプログラマとの接続例	
(ユーザ電源使用時)	490
優先順位	
STP,SLP,TMD ビットの優先順位	126

よ

要求レベル設定レジスタ	
要求レベル設定レジスタ (ELVR: External Level	
Register)	335
読出し	
フラッシュメモリを読出し/リセット状態にする	
方法	473

ら

ラッチ	
遅延割込み発生モジュールの使用上の注意	
(遅延割込み要求ラッチ)	90

り

リセット	
外部リセット端子のブロックダイアグラム	96
発振安定待ちリセット状態	95
フラッシュメモリを読出し/リセット状態にする	
方法	473
リセット中の端子の状態	101
リセット動作の概要	97
リセット要因	92
リセット要因ビット	99
リセット要因ビットとリセット要因の対応	99
リセット要因ビットの注意事項	100
リセットベクタ	
プログラムカウンタバンクレジスタ (PCB)	
<初期値: リセットベクタ中の値>	35
リセット要因	
リセット要因	92
リセット要因と発振安定待ち時間	94
リセット要因ビット	99
リセット要因ビットとリセット要因の対応	99
リセット要因ビットの注意事項	100
リセット要因ビット	
リセット要因ビット	99
リセット要因ビットとリセット要因の対応	99
リセット要因ビットの注意事項	100

リロード	
リロード/コンペア機能任意幅でのアップ/	
ダウンカウント	270
リロード/コンペア機能の選択	269
リロード/コンペア機能	
リロード/コンペア機能任意幅でのアップ/	
ダウンカウント	270
リロード/コンペア機能の選択	269
リロード/コンペアレジスタ	
リロード/コンペアレジスタ (ch.0/ch.1)(RCR0/1)	
.....	263
リロード機能	
リロード機能	269
リロードタイマ	
16 ビットリロードタイマに関する端子	284
16 ビットリロードタイマに関する端子の	
ブロックダイアグラム	285
16 ビットリロードタイマの設定	294
16 ビットリロードタイマの動作モード	282
16 ビットリロードタイマのプログラム例	302
16 ビットリロードタイマのブロックダイヤ	
グラム	284
16 ビットリロードタイマのレジスタ構成	286
16 ビットリロードタイマの割込み	293
16 ビットリロードタイマの割込みと DMA 転送	
および EI ² OS	293
リロード値	
タイマ値とリロード値	514
リロード値とパルス幅の関係	323
リロード動作モード	
リロード動作モード	514
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	
.....	296
リロードレジスタ	
リロードレジスタ (PRLLO ~ 5,PRLHO ~ 5)	319
リロードレジスタへの書込みタイミング	326

れ

例外処理	
未定義命令の実行による例外処理	84
レジスタ	
16 ビットタイマレジスタ (TMR)	291
16 ビットタイマレジスタ (TMR)/16 ビット	
リロードレジスタ (TMRLR)	291
16 ビットリロードレジスタ (TMRLR)	292
DMA コントロールステータスレジスタ	
(DMACS)	69
μPG コントロール/ステータスレジスタ (PGCSR)	
.....	528
PLL 出力選択レジスタ (PLLOS) の構成	111
PPG0/2/4 動作モード制御レジスタ (PPGC0/2/4)	
.....	313
PPG0 ~ 5 出力制御レジスタ (PPG01/PPG23/	
PPG45)	317
PPG1/3/5 動作モード制御レジスタ (PPGC1/3/5)	
.....	315

PWC コントロール / ステータスレジスタ (PWCSR0 ~ PWCSR2)	498	通信プリスケラコントロールレジスタ (CDCR)	405
PWC データバッファレジスタ (PWCR0 ~ PWCR2)	503	通信プリスケラコントロールレジスタ 0/1 (SDCR0/SDCR1)	380
ROM ミラー機能選択レジスタ (ROMM)	455	低消費電力モード制御レジスタ (LPMCR)	124
アウトプットコンペアコントロールレジスタ (OCS01/23/45)	220	低消費電力モード制御レジスタへのアクセス	126
アウトプットコンペアレジスタ (OCCP0 ~ 5)	219	データレジスタ (ADCR2, ADCR1)	355
アップダウンカウンタレジスタ (ch.0/ ch.1)(UDCR0/1)	262	データレジスタ (IDAR)	547
アップダウンタイマ入力許可レジスタ (UDER)	176	時計タイマ制御レジスタ (WTC) の構成	202
アドレスレジスタ (IADR)	546	バスコントロールレジスタ (IBCR)	538
アナログ入力許可レジスタ (ADER)	176	バスコントロールレジスタ (IBCR) 使用上の注意	543
インタラプトレベルマスクレジスタ (ILM)	32	バスステータスレジスタ (IBSR)	536
インプットキャプチャコントロールステータス レジスタ (ICS01)	223	バス制御信号選択レジスタ (EPCR)	157
インプットキャプチャデータレジスタ (IPCP0,1)	222	フラッシュメモリコントロールステータス レジスタ (FMCS)	458, 460
ウォッチドッグタイマ制御レジスタ (WDTC)	191	フリーランタイマのクリアタイミング (コンペア レジスタ 0 との一致)	232
外部アドレス出力制御レジスタ (HACR)	156	プログラムアドレス検出制御ステータスレジスタ (PACSR)	443
カウンタコントロールレジスタ (ch.0) 上位 (CCR0)	254	プログラムアドレス検出レジスタ (PADR0, PADR1)	445
カウンタコントロールレジスタ (ch.0/ch.1) 下位 (CCRL0/1)	258	プログラムアドレス検出レジスタ (PADR0H, M, L, PADR1H, M, L)	446
カウンタコントロールレジスタ (ch.1) 上位 (CCR1)	256	プログラムカウンタバンクレジスタ (PCB) < 初期値: リセットベクタ中の値 >	35
カウンタステータスレジスタ 0/1(CSR0/1)	260	分周比制御レジスタ (DIVR0 ~ DIVR2)	504
拡張インテリジェント I/O サービス (EI ² OS) ステータスレジスタ (ISCS)	78	ポート出力端子レジスタ (ODR7, ODR4)	175
クロックコントロールレジスタ (ICCR)	544	ポートデータレジスタ (PDR0 ~ PDRA)	172
クロック選択レジスタ (CKSCR) の構成	108	ポート入力抵抗レジスタ (RDR0, RDR1)	175
コントロールステータスレジスタ 1(ADCS1)	350	ポート方向レジスタ (DDR0 ~ DDRA)	173
コントロールステータスレジスタ 2(ADCS2)	352	要求レベル設定レジスタ (ELVR: External Level Register)	335
自動レディ機能選択レジスタ (ARSR)	155	リロード / コンペアレジスタ (ch.0/ch.1)(RCR0/1)	263
シリアルインプット / アウトプットレジスタ (SIDR/SODR)	402	リロードレジスタ (PRL0 ~ 5, PRLH0 ~ 5)	319
シリアルコントロールレジスタ (SCR)	400	リロードレジスタへの書き込みタイミング	326
シリアルステータスレジスタ (SSR)	403	割込み / DTP 要因レジスタ (EIRR: External Interrupt Request Register)	335
シリアルデータレジスタ 0/1(SDR0/SDR1)	379	割込み制御レジスタ (ICR00 ~ ICR15) の各ビット の機能	49
シリアルモードコントロールステータスレジスタ 0/1(SMCS0/SMCS1)	376	割込み制御レジスタ (ICR00 ~ ICR15) の機能	48
シリアルモードレジスタ (SMR)	398	割込み制御レジスタ (ICR00 ~ ICR15) の構成	48
スタンバイモード遷移のための低消費電力モード 制御レジスタ (LPMCR) へアクセスする際の 注意事項	146	レジスタバンク 汎用レジスタ (レジスタバンク)	37
タイマカウンタコントロールステータスレジスタ (TCCS)	215	レジスタバンクポインタ レジスタバンクポインタ (RP)	32
タイマカウンタデータレジスタ (TCDT)	215	レディ機能 レディ機能	163
タイマ制御ステータスレジスタ (TMCSR)	287	連続測定 単発測定と連続測定	517
タイムベースタイマ制御レジスタ (TBTC)	181	連続モード 連続モードにおける μ DMAC の起動例	360
ダイレクトページレジスタ (DPR) < 初期値 : 01H >	36		
チップセレクト領域 MASK レジスタ (CMRx)	434		
チップセレクト領域レジスタ (CARx)	435		

わ

割込み

16 ビット入出力タイマの割込み	224
16 ビット入出力タイマの割込みと DMA 転送 および EI ² OS	225
16 ビットリロードタイマの割込み	293
16 ビットリロードタイマの割込みと DMA 転送 および EI ² OS	293
8/10 ビット A/D コンバータの割込み	356
8/10 ビット A/D コンバータの割込みと DMA 転送および EI ² OS	356
8/16 ビット PPG タイマの割込み	320, 326
8/16 ビット PPG タイマの割込みと DMA 転送 および EI ² OS	321
8/16 ビットアップダウンカウンタ / タイマの 割込み	264
8/16 ビットアップダウンカウンタ / タイマの 割込みと DMA 転送および EI ² OS	265
A/D コンバータの割込み	356
I ² C インタフェースに関する割込み要因	548
I ² C インタフェースの割込み制御ビットと 割込み要因	548
I ² C インタフェースの割込みと DMA 転送および EI ² OS	549
PPG カウンタアンダフロー割込み	320
PWC タイマに関する割込み要因	505
PWC タイマの割込み	505
PWC タイマの割込みと DMA 転送および EI ² OS	506
UART に関する割込み要因	407
UART の割込み	407
UART の割込みと DMA 転送および EI ² OS	408
外部割込みの動作	338
外部割込み要求レベル	340
拡張 I/O シリアルインタフェースに関する割込み 要因	381
拡張 I/O シリアルインタフェースの割込み	381
拡張 I/O シリアルインタフェースの割込み機能	388
拡張 I/O シリアルインタフェースの割込みと DMA 転送および EI ² OS	381
スタンバイモードの割込みによる解除	144
スタンバイモードへの移行と割込み	144
ソフトウェア割込みからの復帰	62
ソフトウェア割込みに関する注意事項	63
ソフトウェア割込みの起動	62
ソフトウェア割込みの動作	63
タイムベースタイマの割込み	183
タイムベースタイマの割込みと μ DMAC	183
多重割込み動作	58
多重割込みの例	59
遅延割込み発生モジュールの使用上の注意 (遅延割込み要求ラッチ)	90
時計タイマのインターバル割込み機能	204

ハードウェア割込みからの復帰	54
ハードウェア割込み処理時間	60
ハードウェア割込みの起動	54
ハードウェア割込みの機能	51
ハードウェア割込みの構造	52
ハードウェア割込みの使用手順	57
ハードウェア割込みの動作	55
ハードウェア割込みの動作フロー	56
ハードウェア割込み抑止	52
割込み処理開始時のスタック動作	85
割込み処理のプログラム例	87
割込み処理復帰のスタック動作	85
割込みタイミング	233
割込み動作の概略フロー	43
割込みの種類と機能	42
割込み発生要求	515
割込み要因と割込みベクタ, 割込み制御レジスタ	45, 564
割込み要求発生	521
割込み抑止命令	40
割込み抑止命令とプリフィックス命令に関する 制約	40
割込み /DTP 許可レジスタ	
割込み /DTP 許可レジスタ (ENIR:ENable Interrupt Request Register)	334
割込み /DTP 要因レジスタ	
割込み /DTP 要因レジスタ (EIRR:External Interrupt Request Register)	335
割込み制御レジスタ	
割込み制御レジスタ (ICR00 ~ ICR15) の各ビット の機能	49
割込み制御レジスタ (ICR00 ~ ICR15) の機能	48
割込み制御レジスタ (ICR00 ~ ICR15) の構成	48
割込み制御レジスタ一覧	47
割込み要因と割込みベクタ, 割込み制御レジスタ	45, 564
割込みタイミング	
割込みタイミング	233
割込み発生要求	
割込み発生要求	515
割込みベクタ	
割込みベクタ	44
割込み要因と割込みベクタ, 割込み制御レジスタ	45, 564
割込み要因	
I ² C インタフェースに関する割込み要因	548
I ² C インタフェースの割込み制御ビットと割込み 要因	548
PWC タイマに関する割込み要因	505
UART に関する割込み要因	407
拡張 I/O シリアルインタフェースに関する割込み 要因	381
割込み要因と割込みベクタ, 割込み制御レジスタ	45, 564

割込み抑止命令	
割込み抑止命令	40
割込み抑止命令とプリフィックス命令に関する 制約	40
ワンショット動作モード	
ワンショット動作モード	514
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) の 動作	298

レジスタ索引

A

ADCR1	データレジスタ	355
ADCR2	データレジスタ	355
ADCS1	コントロールステータスレジスタ 1	350
ADCS2	コントロールステータスレジスタ 2	352
ADER	アナログ入力許可レジスタ	176
ARSR	自動レディ機能選択レジスタ	155

B

BAP	バッファアドレスポインタ	79
-----	--------------------	----

C

CALR	チップセレクトアクティブレベルレジスタ	437
CAR0	チップセレクト領域レジスタ 0	435
CAR1	チップセレクト領域レジスタ 1	435
CAR2	チップセレクト領域レジスタ 2	435
CAR3	チップセレクト領域レジスタ 3	435
CCRHO	カウンタコントロールレジスタ (ch.0) 上位	254
CCRHI	カウンタコントロールレジスタ (ch.1) 上位	256
CCRL0	カウンタコントロールレジスタ (ch.0) 下位	258
CCRL1	カウンタコントロールレジスタ (ch.1) 下位	258
CDCR	通信プリスケラコントロールレジスタ	405
CKSCR	クロック選択レジスタ	108
CMR0	チップセレクト領域 MASK レジスタ 0	434
CMR1	チップセレクト領域 MASK レジスタ 1	434
CMR2	チップセレクト領域 MASK レジスタ 2	434
CMR3	チップセレクト領域 MASK レジスタ 3	434
CPCLR	コンペアクリアレジスタ	214
CSCR	チップセレクトコントロールレジスタ	436
CSR0	カウンタステータスレジスタ ch.0	260
CSR1	カウンタステータスレジスタ ch.1	260

D

DCT	データカウンタ	77
DDR0	ポート方向レジスタ 0	173
DDR1	ポート方向レジスタ 1	173
DDR2	ポート方向レジスタ 2	173
DDR3	ポート方向レジスタ 3	173

DDR4	ポート方向レジスタ 4	173
DDR5	ポート方向レジスタ 5	173
DDR6	ポート方向レジスタ 6	173
DDR7	ポート方向レジスタ 7	173
DDR8	ポート方向レジスタ 8	173
DDR9	ポート方向レジスタ 9	173
DDRA	ポート方向レジスタ A	173
DER	μDMAC イネーブルレジスタ	64
DIRR	遅延割込み要因発生 / 解除レジスタ	89
DIVR0	分周比制御レジスタ 0	504
DIVR1	分周比制御レジスタ 1	504
DIVR2	分周比制御レジスタ 2	504
DSR	μDMAC ステータスレジスタ	65
DSSR	μDMAC ストップステータスレジスタ	65

E

EIRR	割込み / DTP 要因レジスタ	335
ELVR	要求レベル設定レジスタ	335
ENIR	割込み / DTP 許可レジスタ	334
EPCR	バス制御信号選択レジスタ	157

F

FMCS	フラッシュメモリコントロールステータスレジスタ	460
------	-------------------------------	-----

H

HACR	外部アドレス出力制御レジスタ	156
------	----------------------	-----

I

IADR	アドレスレジスタ	546
IBCR	バスコントロールレジスタ	538
IBSR	バスステータスレジスタ	536
ICCR	クロックコントロールレジスタ	544
ICR00	割込み制御レジスタ 00	48
ICR01	割込み制御レジスタ 01	48
ICR02	割込み制御レジスタ 02	48
ICR03	割込み制御レジスタ 03	48
ICR04	割込み制御レジスタ 04	48
ICR05	割込み制御レジスタ 05	48
ICR06	割込み制御レジスタ 06	48
ICR07	割込み制御レジスタ 07	48
ICR08	割込み制御レジスタ 08	48
ICR09	割込み制御レジスタ 09	48
ICR10	割込み制御レジスタ 10	48
ICR11	割込み制御レジスタ 11	48
ICR12	割込み制御レジスタ 12	48
ICR13	割込み制御レジスタ 13	48
ICR14	割込み制御レジスタ 14	48
ICR15	割込み制御レジスタ 15	48

レジスタ索引

ICS01	インプットキャプチャコントロールステータスレジスタ	223
IDAR	データレジスタ	547
IOA	I/O レジスタアドレスポインタ	77
IPCP0	インプットキャプチャデータレジスタ (ch.0)	222
IPCP1	インプットキャプチャデータレジスタ (ch.1)	222
ISCS	拡張インテリジェント I/O サービス (EI ² OS) ステータスレジスタ	78

L

LPMCR	低消費電力モード制御レジスタ	124
-------	----------------------	-----

O

OCCP0	アウトプットコンペアレジスタ (ch.0)	219
OCCP1	アウトプットコンペアレジスタ (ch.1)	219
OCCP2	アウトプットコンペアレジスタ (ch.2)	219
OCCP3	アウトプットコンペアレジスタ (ch.3)	219
OCCP4	アウトプットコンペアレジスタ (ch.4)	219
OCCP5	アウトプットコンペアレジスタ (ch.5)	219
OCS01	アウトプットコンペアコントロールレジスタ (ch.0, ch.1)	220
OCS23	アウトプットコンペアコントロールレジスタ (ch.2, ch.3)	220
OCS45	アウトプットコンペアコントロールレジスタ (ch.4, ch.5)	220
ODR4	ポート出力端子レジスタ	175
ODR7	ポート出力端子レジスタ	175

P

PACSR	プログラムアドレス検出制御ステータスレジスタ	443
PADR0	プログラムアドレス検出レジスタ 0	445
PADR1	プログラムアドレス検出レジスタ 1	445
PDR0	ポートデータレジスタ 0	172
PDR1	ポートデータレジスタ 1	172
PDR2	ポートデータレジスタ 2	172
PDR3	ポートデータレジスタ 3	172
PDR4	ポートデータレジスタ 4	172
PDR5	ポートデータレジスタ 5	172
PDR6	ポートデータレジスタ 6	172
PDR7	ポートデータレジスタ 7	172
PDR8	ポートデータレジスタ 8	172
PDR9	ポートデータレジスタ 9	172
PDRA	ポートデータレジスタ A	172
PGCSR	μPG コントロール / ステータスレジスタ	528
PLLOS	PLL 出力選択レジスタ	111
PPG01	PPG0, 1 出力制御レジスタ	317

PPG23	PPG2, 3 出力制御レジスタ	317
PPG45	PPG4, 5 出力制御レジスタ	317
PPGC0	PPG0 動作モード制御レジスタ	313
PPGC1	PPG1 動作モード制御レジスタ	315
PPGC2	PPG2 動作モード制御レジスタ	313
PPGC3	PPG3 動作モード制御レジスタ	315
PPGC4	PPG4 動作モード制御レジスタ	313
PPGC5	PPG5 動作モード制御レジスタ	315
PRLH0	リロードレジスタ H(ch.0)	319
PRLH1	リロードレジスタ H(ch.1)	319
PRLH2	リロードレジスタ H(ch.2)	319
PRLH3	リロードレジスタ H(ch.3)	319
PRLH4	リロードレジスタ H(ch.4)	319
PRLH5	リロードレジスタ H(ch.5)	319
PRL0	リロードレジスタ L(ch.0)	319
PRL1	リロードレジスタ L(ch.1)	319
PRL2	リロードレジスタ L(ch.2)	319
PRL3	リロードレジスタ L(ch.3)	319
PRL4	リロードレジスタ L(ch.4)	319
PRL5	リロードレジスタ L(ch.5)	319
PWCR0	PWC データバッファレジスタ 0	503
PWCR1	PWC データバッファレジスタ 1	503
PWCR2	PWC データバッファレジスタ 2	503
PWCSR0	PWC コントロール / ステータスレジスタ 0	498
PWCSR1	PWC コントロール / ステータスレジスタ 1	498
PWCSR2	PWC コントロール / ステータスレジスタ 2	498

R

RCR0	リロード / コンペアレジスタ (ch.0)	263
RCR1	リロード / コンペアレジスタ (ch.1)	263
RDR0	ポート入力抵抗レジスタ	175
RDR1	ポート入力抵抗レジスタ	175
ROMM	ROM ミラー機能選択レジスタ	455

S

SCR	シリアルコントロールレジスタ	400
SDCR0	通信プリスケラコントロールレジスタ 0	380
SDCR1	通信プリスケラコントロールレジスタ 1	380
SDR0	シリアルデータレジスタ 0	379
SDR1	シリアルデータレジスタ 1	379
SIDR	シリアルインプットレジスタ	402
SMCS0	シリアルモードコントロールステータスレジスタ 0	376
SMCS1	シリアルモードコントロールステータスレジスタ 1	376
SMR	シリアルモードレジスタ	398
SODR	シリアルアウトプットレジスタ	402
SSR	シリアルステータスレジスタ	403

T

TBTC	タイムベースタイマ制御レジスタ	181
------	-----------------------	-----

TCCS	タイマカウンタコントロールステータス レジスタ	215
TCDT	タイマカウンタデータレジスタ	215
TMCSR	タイマ制御ステータスレジスタ	287
TMR	16 ビットタイマレジスタ	291
TMRLR	16 ビットリロードレジスタ	291

U

UDCR0	アップダウンカウントレジスタ (ch.0)	262
UDCR1	アップダウンカウントレジスタ (ch.1)	262
UDER	アップダウンタイマ入力許可レジスタ	176

W

WDTC	ウォッチドッグタイマ制御レジスタ	191
WTC	時計タイマ制御レジスタ	202

端子機能索引

A

A00	外部アドレス端子	152
A01	外部アドレス端子	152
A02	外部アドレス端子	152
A03	外部アドレス端子	152
A04	外部アドレス端子	152
A05	外部アドレス端子	152
A06	外部アドレス端子	152
A07	外部アドレス端子	152
A08	外部アドレス端子	152
A09	外部アドレス端子	152
A10	外部アドレス端子	152
A11	外部アドレス端子	152
A12	外部アドレス端子	152
A13	外部アドレス端子	152
A14	外部アドレス端子	152
A15	外部アドレス端子	152
A16	アドレス上位出力端子	152
A17	アドレス上位出力端子	152
A18	アドレス上位出力端子	152
A19	アドレス上位出力端子	152
A20	アドレス上位出力端子	152
A21	アドレス上位出力端子	152
A22	アドレス上位出力端子	152
A23	アドレス上位出力端子	152
AD00	外部アドレス / データバス下位 入出力端子	152
AD01	外部アドレス / データバス下位 入出力端子	152
AD02	外部アドレス / データバス下位 入出力端子	152
AD03	外部アドレス / データバス下位 入出力端子	152
AD04	外部アドレス / データバス下位 入出力端子	152
AD05	外部アドレス / データバス下位 入出力端子	152
AD06	外部アドレス / データバス下位 入出力端子	152
AD07	外部アドレス / データバス下位 入出力端子	152
AD12	外部アドレス / データバス上位 入出力端子	152
AD13	外部アドレス / データバス上位 入出力端子	152
AD14	外部アドレス / データバス上位 入出力端子	152
AD15	外部アドレス / データバス上位 入出力端子	152
AIN0	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.0)	251
AIN1	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.1)	251
ALE	アドレス取込み許可信号 (ALE) 端子	152
AV _{CC}	A/D コンバータ電源端子	347

AVRH	A/D コンバータ外部基準電源端子	347
AV _{SS}	A/D コンバータ電源端子	347

B

BIN0	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.0)	251
BIN1	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.1)	251

C

CLK	マシンサイクルクロック出力 (CLK) 端子	153
-----	---------------------------------	-----

D

D00	外部データバス下位出力端子	152
D01	外部データバス下位出力端子	152
D02	外部データバス下位出力端子	152
D03	外部データバス下位出力端子	152
D04	外部データバス下位出力端子	152
D05	外部データバス下位出力端子	152
D06	外部データバス下位出力端子	152
D07	外部データバス下位出力端子	152
D08	外部データバス上位出力端子	152
D09	外部データバス上位出力端子	152
D10	外部データバス上位出力端子	152
D11	外部データバス上位出力端子	152
D12	外部データバス上位出力端子	152
D13	外部データバス上位出力端子	152
D14	外部データバス上位出力端子	152
D15	外部データバス上位出力端子	152

F

FRCK	フリーランタイム使用時の外部クロック 入力端子	211
------	----------------------------------	-----

H

HAK	ホールドアクノリッジ出力 (HAK) 端子	153
HRQ	ホールド要求入力 (HRQ) 端子	153

I

IN0	インプットキャプチャチャネル0のトリガ 入力端子	211
IN1	インプットキャプチャチャネル1のトリガ 入力端子	211
IRQ0	外部割込み入力端子	332
IRQ1	外部割込み入力端子	332
IRQ2	外部割込み入力端子	332

R

<u>RD</u>	読出しストローブ出力 (RD) 端子	153
<u>RDY</u>	外部レディ入力 (RDY) 端子	153
<u>RST</u>	外部リセット入力端子 (RST 端子)	93

T

TIN0	16 ビットリロードタイマ入力端子	284
TOT0	16 ビットリロードタイマ出力端子	284

V

V _{CC3}	3.3V ± 0.3V 用電源端子 (V _{CC3})	15
V _{CC5}	3V/5V 用電源端子	15
V _{SS}	電源入力 (GND)	15

W

<u>WRH</u>	上位側データ書込みストローブ出力 (WRH) 端子	153
<u>WRL</u>	下位側データ書込みストローブ出力 (WRL) 端子	153

X

X0	発振端子	118
X0A	32kHz 発振端子	118
X1	発振端子	118
X1A	32kHz 発振端子	118

Z

ZIN0	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.0)	251
ZIN1	8/16 ビットアップダウンカウンタ / タイマ 入力端子 (ch.1)	251

割込みベクタ索引

#08	リセット	92
#09	INT9 命令	440
#10	例外処理	84
#11	INT0(IRQ0)	337
#12	INT1(IRQ1)	337
#13	INT2(IRQ2)	337
#14	INT3(IRQ3)	337
#15	INT4(IRQ4)	337
#16	INT5(IRQ5)	337
#17	INT6(IRQ6)	337
#18	INT7(IRQ7)	337
#19	PWC1(MB90485 シリーズのみ)	506
#20	PWC2(MB90485 シリーズのみ)	506
#21	PWC0(MB90485 シリーズのみ)	506
#22	PPG0/PPG1 カウンタボロー	321
#23	PPG2/PPG3 カウンタボロー	321
#24	PPG4/PPG5 カウンタボロー	321
#25	8/16 ビットアップダウンカウンタ / タイマ (ch.0, ch.1) コンペア / アンダフロー / オーバフロー / アップダウン反転	265
#26	インプットキャプチャ (ch.0) 取込み	225
#27	インプットキャプチャ (ch.1) 取込み	225
#28	アウトプットコンペア (ch.0) 一致	225
#29	アウトプットコンペア (ch.1) 一致	225
#30	アウトプットコンペア (ch.2) 一致	225
#31	アウトプットコンペア (ch.3) 一致	225
#32	アウトプットコンペア (ch.4) 一致	225
#33	アウトプットコンペア (ch.5) 一致	225
#34	UART 送信完了	408
#35	16 ビットフリーランタイムオーバフロー	225
#35	16 ビットリロードタイムアンダフロー	293
#36	UART 受信完了	408
#37	SIO1	381
#38	SIO2	381
#39	I ² C インタフェース (MB90485 シリーズ のみ)	549
#40	A/D コンバータ	356
#41	タイムベースタイマの割込み	183
#41	フラッシュメモリの書込み / 消去	460
#41	時計タイマ	204
#42	遅延割込み発生モジュール	89

CM44-10121-7

富士通マイクロエレクトロニクス・CONTROLLER MANUAL

F²MC-16LX

16 ビット・マイクロコントローラ

MB90480/485 Series

ハードウェアマニュアル

2008 年 7 月 第 7 版発行

発行	富士通マイクロエレクトロニクス株式会社
編集	マーケティング統括部 ビジネス推進部
