



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

MB90550A/B 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

MB90550A/B 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://edevic.fujitsu.com/micom/jp-support/>

富士通マイクロエレクトロニクス株式会社

はじめに

■ 本書の目的と対象読者

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

MB90550A/Bシリーズは、ASIC (Application Specific IC) 対応が可能なオリジナル16ビット・ワンチップマイクロコントローラである F²MC[®]-16LX シリーズの汎用品の1つとして開発された製品です。

本書は、実際にこの半導体を使って製品を設計する技術者の方を対象に、MB90550A/Bシリーズの機能や動作について記載しています。本書をご一読ください。

■ 商標

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 AlgorithmTM 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 24 の章および付録から構成されています。

第 1 章 概要

MB90550A/B シリーズの特長や基本的な仕様について示します。

第 2 章 CPU

F²MC[®]-16LX シリーズの CPU の内部構成および MB90550A/B シリーズに内蔵されているハードウェアの仕様について説明します。

第 3 章 割込み

割込みの機能と動作について説明します。

第 4 章 クロックとリセット

クロックとリセットの機能と動作について説明します。

第 5 章 低消費電力制御回路

低消費電力制御回路の機能と動作について説明します。

第 6 章 メモリアクセスモード

F²MC®-16LX シリーズの内部メモリアクセスモードと外部メモリアクセスについて説明します。

第 7 章 I/O ポート

I/O ポートの機能と動作について説明します。

第 8 章 タイムベースタイマ

タイムベースタイマの機能と動作について説明します。

第 9 章 ウォッチドッグタイマ

ウォッチドッグタイマの機能と動作について説明します。

第 10 章 16 ビット入出力タイマ

16 ビット入出力タイマの機能と動作について説明します。

第 11 章 16 ビットリロードタイマ (イベントカウント機能付)

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

第 12 章 8/16 ビット PPG

8/16 ビット PPG の機能と動作について説明します。

第 13 章 DTP/ 外部割込み

DTP/ 外部割込みの機能と動作および使用上の注意について説明します。

第 14 章 遅延割込み発生モジュール

遅延割込み発生モジュールの機能と動作および使用上の注意について説明します。

第 15 章 A/D コンバータ

A/D コンバータの機能と動作および使用上の注意について説明します。

第 16 章 通信プリスケアラレジスタ

通信プリスケアラの機能と動作について説明します。

第 17 章 UART

UART の機能と動作 , 使用上の注意および応用例について説明します。

第 18 章 I/O 拡張シリアルインタフェース

I/O 拡張シリアルインタフェースの機能と動作について説明します。

第 19 章 I²C インタフェース

I²C インタフェースの機能と動作について説明します。

第 20 章 クロックモニタ機能

クロックモニタ機能について説明します。

第 21 章 アドレス一致検出機能

アドレス一致検出機能とその動作について説明します。

第 22 章 ROM ミラー機能選択モジュール

ROM ミラー機能選択モジュールの機能と動作について説明します。

第 23 章 1M ビットフラッシュメモリ

1M ビットフラッシュメモリの機能と動作について説明します。

第 24 章 MB90F553A のシリアル書込み接続例

MB90F553A のシリアル書込み接続例について説明します。

付録

付録では、I/O マップ、命令および OTPROM 書込みなどを掲載しています。

- 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、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	概要	1
1.1	特長	2
1.2	品種構成	4
1.3	ブロックダイヤグラム	5
1.4	パッケージ外形寸法図	6
1.5	端子配列図	8
1.6	端子機能説明	10
1.7	入出力回路形式	16
1.8	デバイスの取扱いに関する注意事項	19
第 2 章	CPU	23
2.1	メモリ空間	24
2.2	アドレス指定	25
2.3	メモリ空間における多バイト長データの配置	28
2.4	専用レジスタ	29
2.4.1	アキュムレータ (A)	31
2.4.2	ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP)	33
2.4.3	プロセッサステータス (PS)	35
2.4.4	プログラムカウンタ (PC)	38
2.4.5	ダイレクトページレジスタ (DPR)	39
2.4.6	バンクレジスタ (PCB, DTB, USB, SSB, ADB)	40
2.5	汎用レジスタ	41
2.6	プリフィックスコード	43
2.7	割込み抑止命令とプリフィックスコード	46
2.8	「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意	47
第 3 章	割込み	49
3.1	割込みの概要	50
3.2	割込み要因	51
3.3	割込みベクタ	54
3.4	ハードウェア割込み	56
3.4.1	ハードウェア割込みの動作	59
3.4.2	ハードウェア割込みの動作フロー	61
3.4.3	ハードウェア割込み使用手順例	62
3.5	ソフトウェア割込み	63
3.6	拡張インテリジェント I/O サービス (EI ² OS)	65
3.6.1	割込み制御レジスタ (ICR)	67
3.6.2	拡張インテリジェント I/O サービスディスクリプタ (ISD)	70
3.6.3	拡張インテリジェント I/O サービス (EI ² OS) の動作	74
3.6.4	拡張インテリジェント I/O サービス (EI ² OS) の実行時間	76
3.7	未定義命令の実行による例外発生	77

第4章	クロックとリセット	79
4.1	クロック発生部	80
4.2	クロック供給マップ	81
4.3	リセット要因	82
4.4	リセット解除後の動作	84
4.5	リセット入力によって初期化されないレジスタ	85
第5章	低消費電力制御回路	87
5.1	低消費電力制御回路の概要	88
5.2	低消費電力モード制御レジスタ (LPMCR)	91
5.3	クロック選択レジスタ (CKSCR)	93
5.4	低消費電力制御回路の動作	96
5.4.1	スリープモード	98
5.4.2	時計モード	99
5.4.3	ストップモード	101
5.4.4	ハードウェアスタンバイモード	102
5.4.5	スリープ、ストップ、ホールド、リセット、ハードウェアスタンバイ時の端子状態	103
5.5	CPU 間欠動作機能	106
5.6	発振安定待ち時間設定	107
5.7	マシクロック	108
第6章	メモリアクセスモード	111
6.1	メモリアクセスモードの概要	112
6.1.1	モード端子	113
6.1.2	モードデータ	114
6.1.3	バスモード別メモリ空間	115
6.2	外部メモリアクセス (外部バス端子制御回路)	118
6.2.1	外部メモリアクセス (外部バス端子制御回路) のレジスタ	119
6.2.2	自動レディ機能選択レジスタ (ARSR)	120
6.2.3	外部アドレス出力制御レジスタ (HACR)	122
6.2.4	バス制御信号選択レジスタ (ECSR)	123
6.3	外部メモリアクセス制御信号の動作	126
6.3.1	レディ機能	128
6.3.2	ホールド機能	130
第7章	I/O ポート	131
7.1	I/O ポートの概要	132
7.2	I/O ポートブロックダイヤグラム	133
7.3	I/O ポートのレジスタ	135
7.3.1	ポートデータレジスタ (PDRx)	137
7.3.2	ポート方向レジスタ (DDRx)	139
7.3.3	出力端子レジスタ (ODR4)	140
7.3.4	入力抵抗レジスタ (RDR0, RDR1)	141
7.3.5	アナログ入力許可レジスタ (ADER)	142

第 8 章	タイムベースタイマ	143
8.1	タイムベースタイマの概要.....	144
8.2	タイムベースタイマ制御レジスタ (TBTC).....	145
8.3	タイムベースタイマの動作.....	147
第 9 章	ウォッチドッグタイマ	149
9.1	ウォッチドッグタイマの概要.....	150
9.2	ウォッチドッグタイマ制御レジスタ (WDTC).....	151
9.3	ウォッチドッグタイマの動作.....	153
第 10 章	16 ビット入出力タイマ	155
10.1	16 ビット入出力タイマの概要.....	156
10.2	16 ビット入出力タイマのブロックダイアグラム.....	158
10.3	16 ビット入出力タイマのレジスタ	159
10.3.1	16 ビットフリーランタイマ	161
10.3.2	アウトプットコンペア	165
10.3.3	インプットキャプチャ	169
10.4	16 ビットフリーランタイマの動作	172
10.5	16 ビットアウトプットコンペアの動作	174
10.6	16 ビットインプットキャプチャの動作	176
第 11 章	16 ビットリロードタイマ (イベントカウント機能付)	179
11.1	16 ビットリロードタイマ (イベントカウント機能付) の概要	180
11.2	16 ビットリロードタイマ (イベントカウント機能付) のレジスタ	181
11.2.1	タイマコントロールステータスレジスタ (TMCSR).....	182
11.2.2	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).....	186
11.3	クロック動作	187
11.4	アンダフロー動作	188
11.5	入出力端子機能	189
11.6	カウンタの動作状態	191
第 12 章	8/16 ビット PPG	193
12.1	8/16 ビット PPG の概要.....	194
12.2	8 ビット PPG のブロックダイアグラム	195
12.3	8/16 ビット PPG のレジスタ.....	197
12.3.1	PPG0 動作モード制御レジスタ (PPGC0).....	198
12.3.2	PPG1 動作モード制御レジスタ (PPGC1).....	200
12.3.3	PPG0/1 出力端子制御レジスタ (PPGE).....	203
12.3.4	リロードレジスタ (PRL/PRLH).....	205
12.4	8/16 ビット PPG の動作	206
12.4.1	8/16 ビット PPG の動作モード	208
12.4.2	PPG 出力動作.....	209
12.4.3	カウントクロックの選択	211
12.4.4	パルスの端子出力の制御	212
12.4.5	リロードレジスタへのライトタイミング	213

第 13 章	DTP/ 外部割込み	215
13.1	DTP/ 外部割込みの概要	216
13.2	DTP/ 外部割込みのレジスタ	217
13.3	DTP/ 外部割込みの動作	219
13.4	DTP/ 外部割込みの使用上の注意	222
第 14 章	遅延割込み発生モジュール	225
14.1	遅延割込み発生モジュールの概要	226
14.2	遅延割込み発生モジュールの動作	227
第 15 章	A/D コンバータ	229
15.1	A/D コンバータの概要	230
15.2	A/D コンバータのレジスタ	232
15.2.1	コントロールステータスレジスタ (ADCS0, ADCS1)	233
15.2.2	データレジスタ (ADCR1, ADCR0)	238
15.3	A/D コンバータの動作	240
15.3.1	単発モード時の EI ² OS の起動例	242
15.3.2	連続モード時の EI ² OS の起動例	244
15.3.3	停止モード時の EI ² OS の起動例	246
15.4	変換データ保護機能	248
第 16 章	通信プリスケアラレジスタ	251
16.1	通信プリスケアラレジスタの概要	252
16.2	通信プリスケアラレジスタの動作	253
第 17 章	UART	255
17.1	UART の概要	256
17.2	UART のブロックダイアグラム	257
17.3	UART のレジスタ	258
17.3.1	シリアルモードレジスタ (SMR)	259
17.3.2	シリアルコントロールレジスタ (SCR)	262
17.3.3	シリアルインプットデータレジスタ (SIDR) / シリアルアウトプットデータレジスタ (SODR)	265
17.3.4	シリアルステータスレジスタ (SSR)	266
17.4	UART の動作	269
17.4.1	UART のクロック選択	270
17.4.2	非同期 (調歩同期) モード	272
17.4.3	CLK 同期モード	274
17.4.4	割込み発生およびフラグのセットタイミング	276
17.5	UART (モード 1 使用時) の応用例	279
第 18 章	I/O 拡張シリアルインタフェース	281
18.1	I/O 拡張シリアルインタフェースの概要	282
18.2	I/O 拡張シリアルインタフェースのレジスタ	284
18.2.1	シリアルモードコントロールステータスレジスタ (SMCS)	285
18.2.2	シリアルシフトデータレジスタ (SDR)	289
18.3	I/O 拡張シリアルインタフェースの動作	290
18.3.1	シフトクロック	291
18.3.2	シリアル I/O の動作状態	293

18.3.3	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	295
18.3.4	I/O 拡張シリアルインタフェースの割込み機能	298
第 19 章	I²C インタフェース	299
19.1	I ² C インタフェースの概要	300
19.2	I ² C インタフェースのブロックダイアグラムと構成図	301
19.3	I ² C インタフェースのレジスタ	303
19.3.1	バスステータスレジスタ (IBSR)	304
19.3.2	バスコントロールレジスタ (IBCR)	307
19.3.3	クロックコントロールレジスタ (ICCR)	310
19.3.4	アドレスレジスタ (IADR)	312
19.3.5	データレジスタ (IDAR)	313
19.3.6	ポート選択レジスタ (ISEL)	314
19.4	I ² C インタフェースの動作	315
19.4.1	I ² C インタフェースの転送フロー	317
19.4.2	I ² C インタフェースのモードフロー	319
第 20 章	クロックモニタ機能	321
20.1	クロックモニタ機能の概要	322
20.2	クロック出力許可レジスタ (CLKR)	323
第 21 章	アドレス一致検出機能	325
21.1	アドレス一致検出機能の概要	326
21.2	アドレス一致検出機能のレジスタ	327
21.3	アドレス一致検出機能の動作	329
21.4	アドレス一致検出機能の使用例	330
21.5	アドレス一致検出機能のプログラム例	332
第 22 章	ROM ミラー機能選択モジュール	335
22.1	ROM ミラー機能選択モジュールの概要	336
22.2	ROM ミラー機能選択レジスタ (ROMM)	337
第 23 章	1M ビットフラッシュメモリ	339
23.1	1M ビットフラッシュメモリの概要	340
23.2	フラッシュメモリ全体のブロックダイアグラムとセクタ構成	341
23.3	書込み / 消去モード	343
23.4	フラッシュメモリコントロールステータスレジスタ (FMCS)	345
23.5	フラッシュメモリ自動アルゴリズム起動方法	347
23.6	自動アルゴリズム実行状態の確認	348
23.6.1	データポーリングフラグ (DQ7)	350
23.6.2	トグルビットフラグ (DQ6)	351
23.6.3	タイミングリミット超過フラグ (DQ5)	352
23.6.4	セクタ消去タイマフラグ (DQ3)	353
23.7	フラッシュメモリ書込み / 消去の詳細説明	354
23.7.1	フラッシュメモリを讀出し / リセット状態にする	355
23.7.2	フラッシュメモリへデータを書き込む	356
23.7.3	フラッシュメモリの全データを消去する (チップ消去)	358
23.7.4	フラッシュメモリの任意のデータを消去する (セクタ消去)	359
23.7.5	フラッシュメモリのセクタ消去を一時停止する	361

23.7.6	フラッシュメモリのセクタ消去を再開する	362
23.8	1M ビットフラッシュメモリのプログラム例	363
第 24 章	MB90F553A シリアル書込み接続例	367
24.1	MB90F553A シリアル書込み接続の基本構成	368
24.2	シリアル書込み接続例 (ユーザ電源使用時)	372
24.3	シリアル書込み接続例 (ライターから電源供給時)	374
24.4	フラッシュマイコンプログラマとの最小限の接続例 (ユーザ電源使用時)	376
24.5	フラッシュマイコンプログラマとの最小限の接続例 (ライターから電源供給時)	378
付録	381
付録 A	I/O マップ	382
付録 B	命令	388
B.1	命令の種類	389
B.2	アドレッシング	390
B.3	直接アドレッシング	392
B.4	間接アドレッシング	399
B.5	実行サイクル数	406
B.6	実効アドレスフィールド	408
B.7	命令一覧表の読み方	410
B.8	F ² MC-16LX 命令一覧表	413
B.9	命令マップ	427
付録 C	OTPROM 書込み	449
索引	451

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。）	
388 ～ 448	付録 B 命令	「付録 B 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 > 2 版 3 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。）	
-	-	端子名を変更 (TOUT TOT)
103	第 5 章 低消費電力制御回路 5.4.5 スリープ、ストップ、 ホールド、リセット、ハード ウェアスタンバイ時の 端子状態	表 5.4-2 シングルチップモード時の各端子状態 の *4 を変更 (「入力遮断」とは端子からすぐの入力ゲートの動作を禁止する状態を 示します。 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。)
158	第 10 章 16 ビット入出力 タイマ 10.2 16 ビット入出力タイ マのブロックダイアグラ ム	図 10.2-1 16 ビット入出力タイマのブロックダイアグラム を変更
163	第 10 章 16 ビット入出力 タイマ 10.3.1 16 ビットフリーラ ンタイマ	[bit2] CLR を変更 ("1" を書き込んだあと、次のカウントクロックまでにこのビットへの "0" 書き込みを行うと、カウンタ値の初期化は行われません。) を追加
217	第 13 章 DTP/ 外部割込み 13.2 DTP/ 外部割込みのレ ジスタ	割込み /DTP 許可レジスタ (ENIR) に < 注意事項 > を追加
218		割込み /DTP 要因レジスタ (EIRR) の < 注意事項 > を変更
233	第 15 章 A/D コンバータ 15.2.1 コントロールス テータスレジスタ (ADCS0, ADCS1)	< 注意事項 > を変更
234		[bit14] INT(Interrupt) を変更
241	第 15 章 A/D コンバータ 15.3 A/D コンバータの動 作	図 15.3-1 A/D 変換起動から変換データ転送までのフロー例 (連続モード) を変更
263	第 17 章 UART 17.3.2 シリアルコント ロールレジスタ (SCR)	[bit10] REC(Receiver Error Clear) を変更 (SSR レジスタのエラーフラグ (PE, ORE, FRE) をクリアします。 "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。)
270	第 17 章 UART 17.4.1 UART のクロック選 択	表 17.4-3 ボーレート (CLK 同期時) を変更

< 参考 > 2 版 3 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。）	
285	第 18 章 I/O 拡張シリアル インタフェース 18.2.1 シリアルモードコ ントロールステータスレ ジスタ (SMCS)	図 18.2-2 シリアルモードコントロールステータスレジスタ (SMCS) を 変更
305	第 19 章 I ² C インタフェー ス 19.3.1 バスステータスレ ジスタ (IBSR)	[bit4] LRB (Last Recieved Bit) を変更
340	第 23 章 1M ビットフラッ シュメモリ 23.1 1M ビットフラッシュ メモリの概要	1M ビットフラッシュメモリの特長 を変更 (Advanced Micro Device Advanced Micro Devices)
425	付録 B 命令 B.8 F ² MC-16LX 命令一覧 表	表 B.8-17 アキュムレータ操作命令 (バイト・ワード) 6 命令 を変更 (SWAPW / XCHW A,T SWAPW)

(変更箇所は、本文中のページ左側の によって示しています。)

第1章

概要

MB90550A/B シリーズの概要について説明します。

- 1.1 特長
- 1.2 品種構成
- 1.3 ブロックダイヤグラム
- 1.4 パッケージ外形寸法図
- 1.5 端子配列図
- 1.6 端子機能説明
- 1.7 入出力回路形式
- 1.8 デバイスの取扱いに関する注意事項

1.1 特長

MB90550A/B シリーズは、高速リアルタイム処理が要求される各種産業用、OA 機器用、プロセス制御用などの用途向けに設計された汎用の高性能 16 ビットマイクロコントローラです。命令体系は、F²MC-8 シリーズの AT アーキテクチャを継承するとともに、高級言語対応命令の追加やアドレッシングモードの拡張、乗除算命令の強化、ビット処理命令の充実化を図っています。さらに、32 ビットアキュムレータの搭載により、ロングワードデータの処理も可能となっています。

■ 特長

MB90550A/B シリーズの特長を、以下に示します。

- 最小命令実行時間
62.5ns/4MHz 原発振 4 通倍 (PLL クロック通倍方式)
- 最大メモリ空間
16Mbyte
- コントローラ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 / リード / ロングワード
 - ・ 標準アドレッシングモード.....23 種類
 - ・ 32bit アキュムレータの採用による高精度演算の強化
 - ・ 符号付き乗除算命令, reti 命令機能強化
- 高級言語 (C 言語) / マルチタスクに対応した命令体系
 - ・ システムスタックポインタの採用
 - ・ 命令セットの対称性とバレルシフト命令
- アドレス一致検出機能内蔵 (2 アドレスポインタ分あり)
- 実行速度の向上
4byte のキュー
- 強力な割込み機能
 - ・ プライオリティレベルがプログラマブルに 8 レベル設定可能
 - ・ 外部割込み入力.....8 本
- データ転送機能
 - ・ インテリジェント I/O サービス.....最大 16ch
 - ・ DTP 要求入力.....8 本
- 内蔵 ROM
 - ・ EPROM 版, Flash 版.....128KB
 - ・ MASK ROM 版.....64KB/128KB

- 内蔵 RAM
 - EPROM 版 , Flash 版.....4KB
 - MASK ROM 版.....2KB/4KB
- 汎用ポート

最大 83 本 (うち入力プルアップ抵抗設定可能 :16 本 / オープンドレイン設定可能 :8 本 / 入出力オープンドレイン :6 本)
- A/D コンバータ
 - RC 逐次比較方式.....8ch
 - 分解能.....8/10bit 選択可能 / 変換時間 26.3 μ s [最小]
- UART:1ch
- I/O 拡張シリアルインタフェース :2ch
- I²C インタフェース :2ch

2ch の内 1ch はターミナル入出力切換え可能 [2 系統]
- 16bit リロードタイマ :2ch
- 8/16bit PPG:3ch

8bit \times 2ch, 16bit \times 1ch モード切換え機能付き
- 16bit 入出力タイマ構成
 - インプットキャプチャ \times 4ch
 - アウトプットコンペア \times 4ch
 - フリーランタイマ \times 1ch
- クロックモニタ機能内蔵

発振クロックの $2^1 \sim 2^8$ 分周のクロックを出力)
- タイムベースタイマ / ウォッチドッグタイマ :18bit
- 低消費電力モード
 - スリープ
 - ストップ
 - ハードウェアスタンバイモード
 - CPU 間欠動作モード機能
- パッケージ
 - QFP-100
 - LQFP-100
- CMOS テクノロジー
- 輻射ノズルの低減 (MB90550B シリーズ)

1.2 品種構成

MB90550A/B シリーズの品種構成を、表 1.2-1 に示します。ROM/RAM 容量以外の機能は共通になっています。

■ 品種構成

表 1.2-1 MB90550A/B シリーズ品種構成一覧

品種名	ROM 容量	RAM 容量	備考
MB90V550A	-	6Kbytes	評価用デバイス
MB90P553A	128Kbytes	4Kbytes	OTPROM
MB90F553A	128Kbytes	4Kbytes	Flash ROM
MB90T553A	-	4Kbytes	ROM 外付け品
MB90553A/B	128Kbytes	4Kbytes	マスク ROM
MB90T552A	-	2Kbytes	ROM 外付け品
MB90552A/B	64Kbytes	2Kbytes	マスク ROM

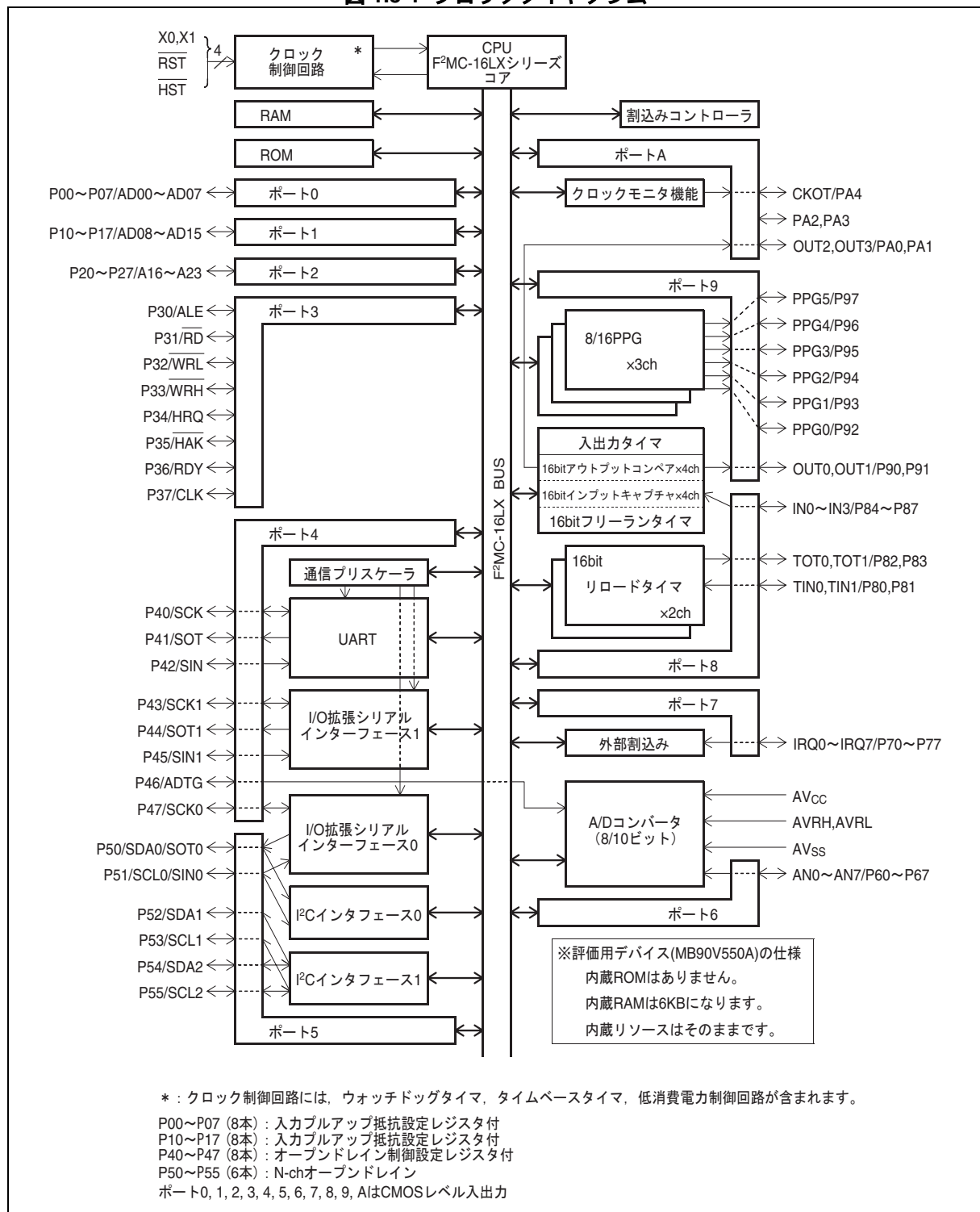
- 本製品は、お客様が Philips 社の定めた I²C 標準仕様書に従う I²C システムの中で使用されることを条件に、Philips 社 I²C 特許がライセンスされています。
- Purchase of Fujitsu I²C components conveys a license under the Philips I²C Patent Rights to use, these components in an I²C system provided that the system conforms to the I²C Standard Specification as defined by Philips.

1.3 ブロックダイアグラム

図 1.3-1 に、ブロックダイアグラムを示します。

■ ブロックダイアグラム

図 1.3-1 ブロックダイアグラム



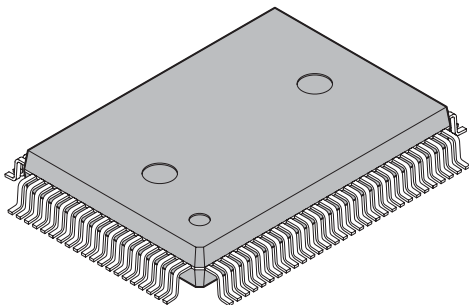
1.4 パッケージ外形寸法図

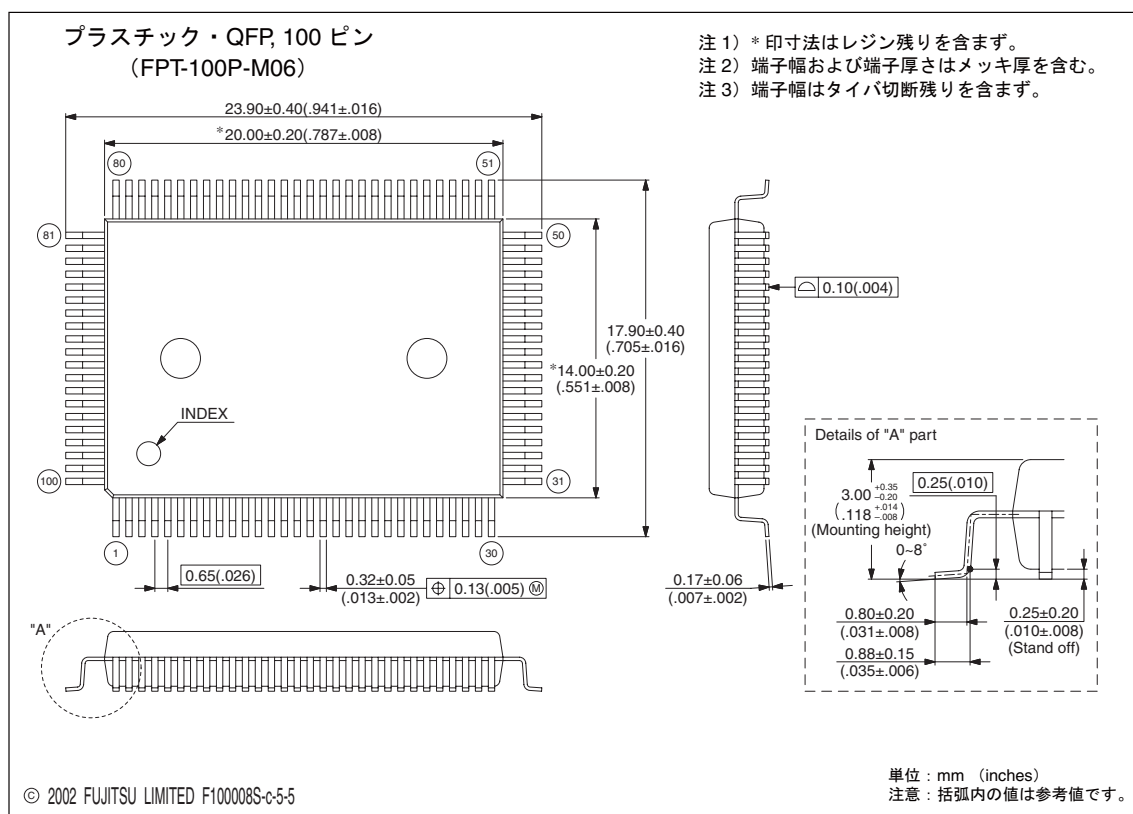
図 1.4-1 に QFP-100 のパッケージ外形寸法図を，図 1.4-2 に LQFP-100 のパッケージ外形寸法図を示します。

なお，本外形寸法図は参考用です。正式版につきましては，別途ご相談ください。

■ FPT-100P-M06 パッケージ外形寸法図

図 1.4-1 QFP-100 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

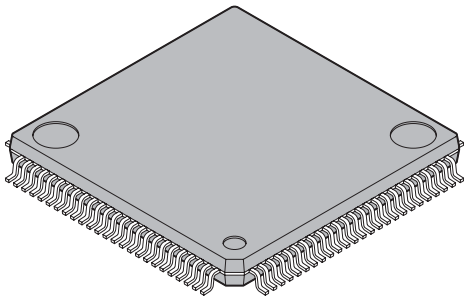


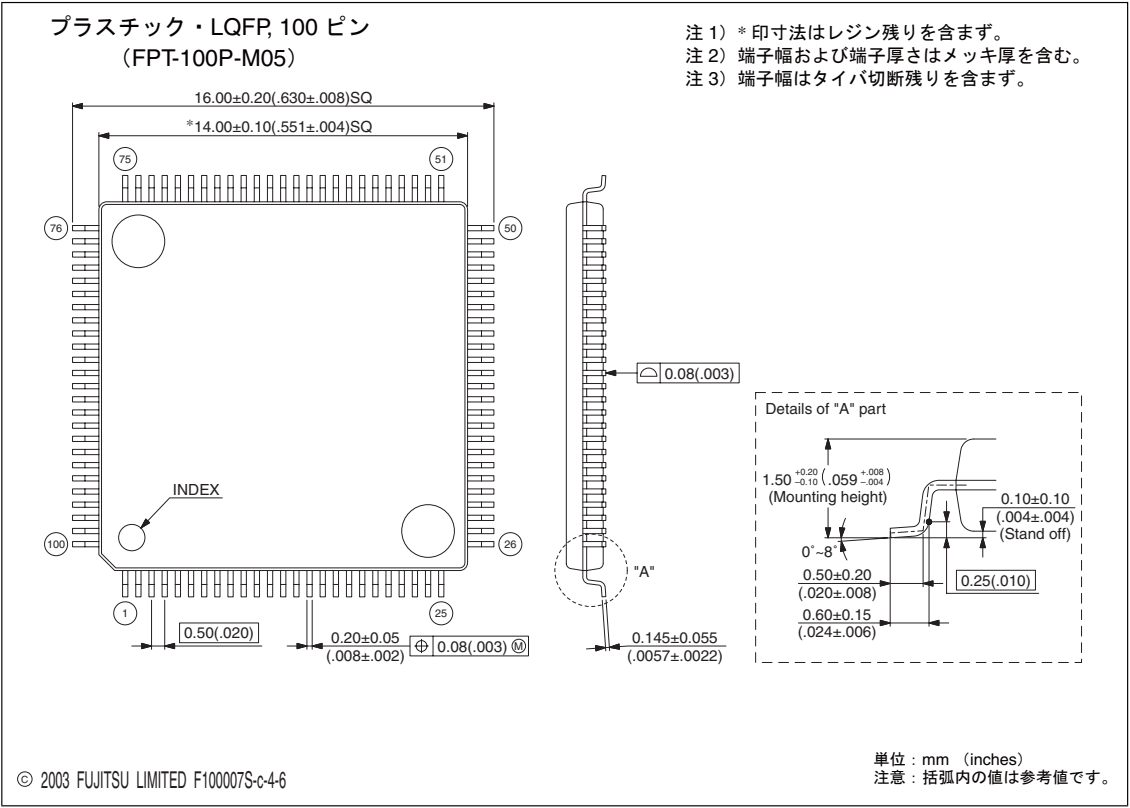
最新の外形寸法図については，下記の URL にてご確認ください。

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

■ FPT-100P-M05 パッケージ外形寸法図

図 1.4-2 LQFP-100 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



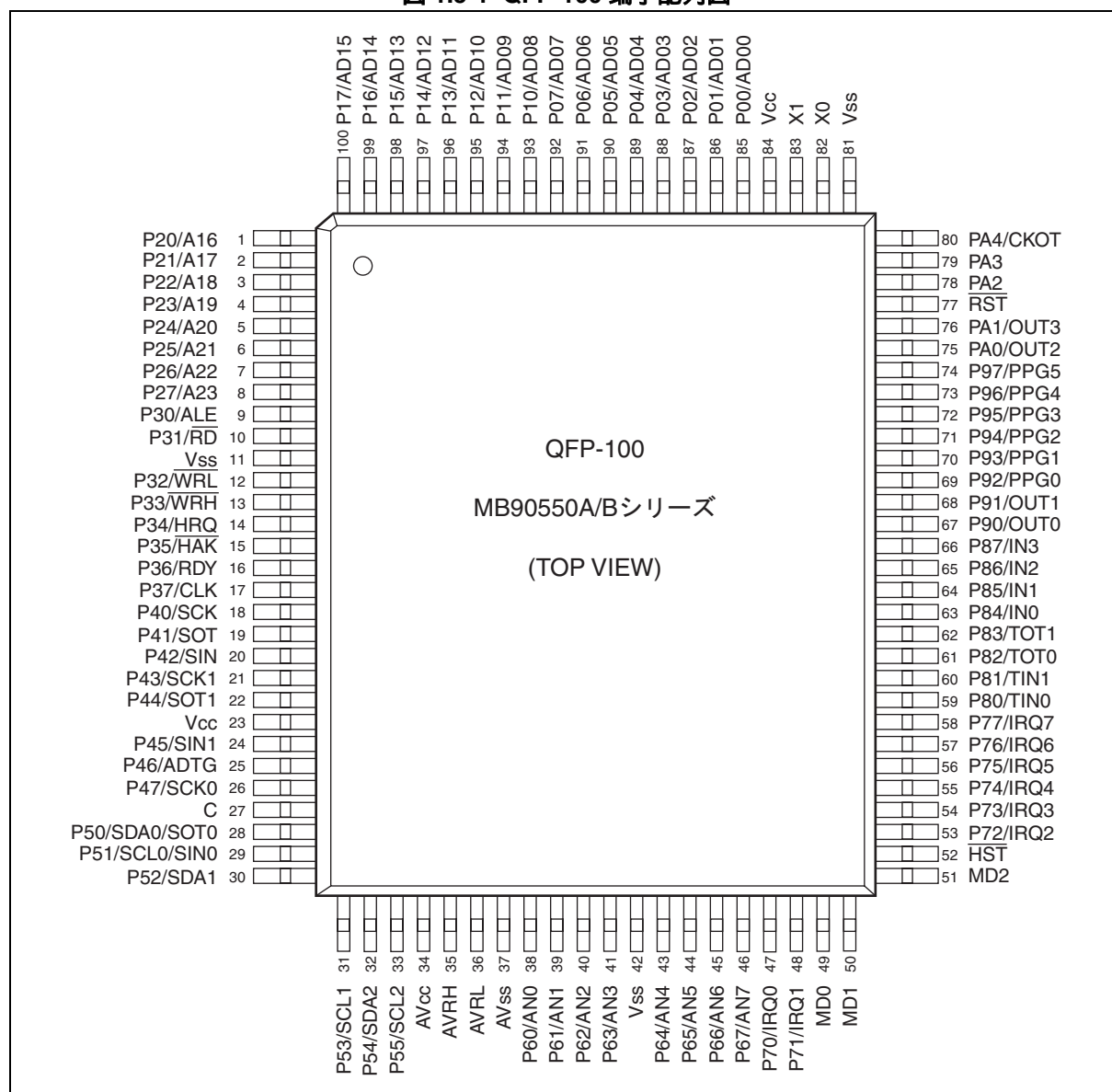
最新の外形寸法図については、下記の URL にてご確認ください。
<http://edevice.fujitsu.com/jp/datasheet/jf-ovpklv.html>

1.5 端子配列図

図 1.5-1 に QFP-100 の端子配列図を，図 1.5-2 に LQFP-100 の端子配列図を示します。

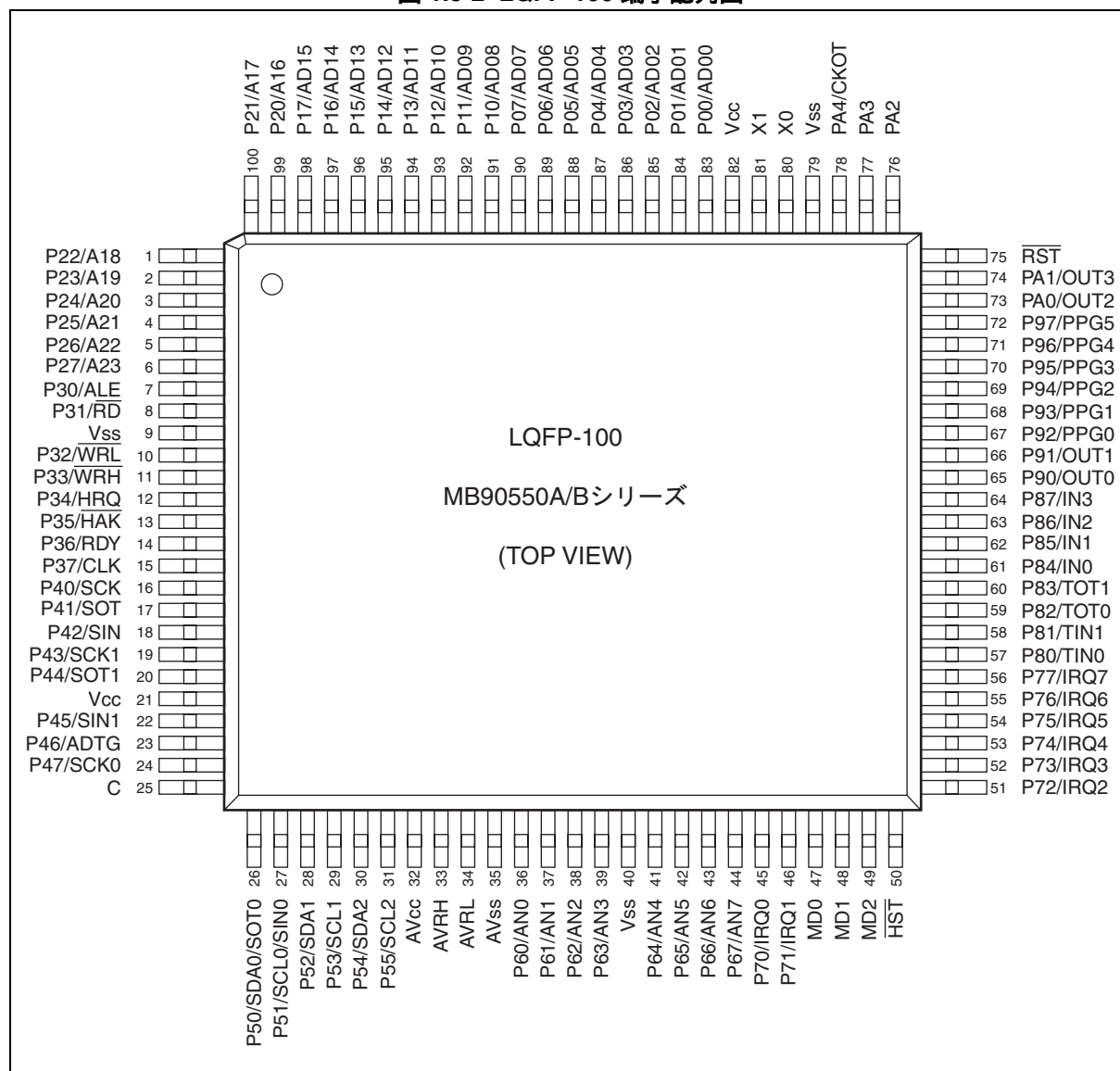
■ FTP-100P-M06 端子配列図

図 1.5-1 QFP-100 端子配列図



■ FTP-100P-M05 端子配列図

図 1.5-2 LQFP-100 端子配列図



1.6 端子機能説明

表 1.6-1 に、端子機能について示します。

■ 端子機能説明

表 1.6-1 端子機能説明 (1 / 6)

QFP	LQFP	端子名	回路	機能説明
82	80	X0	A	発振用端子です。
83	81	X1	A	発振用端子です。
77	75	$\overline{\text{RST}}$	B	リセット入力端子です。
52	50	$\overline{\text{HST}}$	C	ハードウェアスタンバイ入力端子です。
85 ~ 92	83 ~ 90	P00 ~ P07	D (CMOS)	汎用の入出力ポートです。 プルアップ抵抗設定レジスタ (RDR0) により、プルアップ抵抗を付加 (RD07 ~ RD00=1) できます。 D07 ~ D00=1: 出力設定時無効
		AD00 ~ AD07		外バスモード時にはデータ下位入出力 / アドレス下位出力 (AD00 ~ AD07) 端子として、機能します。
93 ~ 100	91 ~ 98	P10 ~ P17	D (CMOS)	汎用の入出力ポートです。 プルアップ抵抗設定レジスタ (RDR1) により、プルアップ抵抗を付加 (RD17 ~ RD10=1) できます。 D17 ~ D10=1: 出力設定時無効
		AD08 ~ AD15		バス幅 16bit 外バスモード時は、データ上位入出力 / アドレス中位出力 (AD08 ~ AD15) として機能します。
1 ~ 8	99, 100, 1 ~ 6	P20 ~ P27	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモード時、あるいは外バスモード時に、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" の場合に有効になります。
		A16 ~ A23		外アドレスバス (A16 ~ A23) の出力端子です。 この機能は、外バスモード時に、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" の場合に有効になります。
9	7	P30	E (CMOS)	汎用の入出力ポートです。 この機能はシングルチップモードのときに有効となります。
		ALE		アドレスラッチイネーブル出力端子です。 この機能は、外バスが有効なモードで有効になります。
10	8	P31	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモードのときに有効となります。
		$\overline{\text{RD}}$		データバスに対する、リードストローブ出力端子です。 この機能は、外バスが有効なモードで有効になります。

表 1.6-1 端子機能説明 (2 / 6)

QFP	LQFP	端子名	回路	機能説明
12	10	P32	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモードのときに有効となります。
		$\overline{\text{WRL}}$		データバス下位 8 ビットに対する、ライトストロープ出力端子です。 この機能は、外バスが有効なモードで有効になります。
13	11	P33	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモードのときに有効となります。
		$\overline{\text{WRH}}$		データバス上位 8 ビットに対する、ライトストロープ出力端子です。 この機能は、外バスが有効なモードで有効になります。
14	12	P34	E (CMOS)	汎用の入出力ポートです。 この機能はシングルチップモードのときに有効となります。
		HRQ		ホールドリクエスト入力端子です。 この機能は、外バスが有効なモードで有効になります。
15	13	P35	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモードのときに有効となります。
		$\overline{\text{HAK}}$		ホールドアクノリッジ出力端子です。 この機能は、外バスが有効なモードで有効になります。
16	14	P36	E (CMOS)	汎用の入出力ポートです。 この機能は、シングルチップモードのときに有効となります。
		RDY		レディ入力端子です。 この機能は、外バスが有効なモードで有効になります。
17	15	P37	E (CMOS)	汎用の入出力ポートです。 この機能はシングルチップモードのときに有効となります。
		CLK		CLK 出力端子です。 この機能は、外バスが有効なモードで有効になります。
18	16	P40	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、オープンドレイン出力ポート (OD40=1) になります (D40=0: 入力設定時無効)。
		SCK		UART シリアルクロック入出力端子です。 この機能は、UART のクロック出力指定が許可のときに有効となります。

表 1.6-1 端子機能説明 (3 / 6)

QFP	LQFP	端子名	回路	機能説明
19	17	P41	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD41=1) になります (D41=0: 入力設定時無効)。
		SOT		UART シリアルデータ出力端子です。 この機能は、UART のシリアルデータ出力指定が許可の とき有効となります。
20	18	P42	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD42=1) になります (D42=0: 入力設定時無効)。
		SIN		UART シリアルデータ入力端子です。UART が入力動作 をしている間はこの入力を随時使用していますので、意 図的に行う以外は、ほかの機能による出力を停止させ ておく必要があります。
21	19	P43	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD43=1) になります (D43=0: 入力設定時無効)。
		SCK1		拡張 I/O シリアルクロック入出力端子です。この機能は 拡張 I/O シリアルクロック出力が許可のときに有効にな ります。
22	20	P44	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD44=1) になります (D44=0: 入力設定時無効)。
		SOT1		拡張 I/O シリアルデータ出力端子です。 この機能は、拡張 I/O シリアルデータ出力が許可のとき 有効になります。
24	22	P45	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD45=1) になります (D45=0: 入力設定時無効)。
		SIN1		拡張 I/O シリアルデータ入力端子です。 拡張 I/O シリアルが入力動作をしている間この入力を随 時使用していますので、意図的に行う以外は、ほかの機 能による出力を停止させておく必要があります。
25	23	P46	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD46=1) になります。 (D46=0: 入力設定時無効)
		ADTG		A/D コンバータ外部トリガ入力端子です。 A/D コンバータが入力動作をしている間この入力を随時 使用していますので、意図的に行う以外は、ほかの機能 による出力を停止させておく必要があります。

表 1.6-1 端子機能説明 (4 / 6)

QFP	LQFP	端子名	回路	機能説明
26	24	P47	F (CMOS/H)	汎用の入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) により、 オープンドレイン出力ポート (OD47=1) になります。 D47=0: 入力設定時無効
		SCK0		拡張 I/O シリアルクロック入出力端子です。この機能は 拡張 I/O シリアルクロック出力が許可のときに有効にな ります。
27	25	C		電源安定化の容量端子です。 外部に 0.1 μ F 程度のセラミックコンデンサを接続してく ださい。
28	26	P50	G (NchOD/H)	N-ch オープンドレイン形式の入出力ポートです。
		SDA0		I ² C インタフェースのデータ入出力端子です。 この機能は I ² C インタフェースが動作許可のときに有効 となります。 I ² C インタフェースが動作している間は、ポート出力を Hi-Z (PDR=1) としてください。
		SOT0		拡張 I/O シリアルデータ出力端子です。 この機能は、拡張 I/O シリアルデータ出力が許可のとき 有効になります。
29	27	P51	G (NchOD/H)	N-ch オープンドレイン形式の入出力ポートです。
		SCL0		I ² C インタフェースのクロック入出力端子です。 この機能は、I ² C インタフェースが動作許可のときに有 効となります。 I ² C インタフェースが動作している間は、ポート出力を Hi-Z (PDR=1) としてください。
		SIN0		拡張 I/O シリアルデータ入力端子です。 拡張 I/O シリアルが入力動作をしている間この入力を随 時使用していますので、意図的に行う以外は、ほかの機 能による出力を停止させておく必要があります。
30, 32	28, 30	P52, P54	G (NchOD/H)	N-ch オープンドレイン形式の入出力ポートです。
		SDA1, SDA2		I ² C インタフェースのデータ入出力端子です。 この機能は、I ² C インタフェースが動作許可のときに有 効となります。 I ² C インタフェースが動作している間は、ポート出力を Hi-Z (PDR=1) としてください。
31, 33	29, 31	P53, P55	G (NchOD/H)	N-ch オープンドレイン形式の入出力ポートです。
		SCL1, SCL2		I ² C インタフェースのクロック入出力端子です。 この機能は I ² C インタフェースが動作許可のときに有効 となります。 I ² C インタフェースが動作している間は、ポート出力を Hi-Z (PDR=1) としてください。

表 1.6-1 端子機能説明 (5 / 6)

QFP	LQFP	端子名	回路	機能説明
38 ~ 41, 43 ~ 46	36 ~ 39, 41 ~ 44	P60 ~ P67	H (CMOS/H)	汎用の入出力ポートです。
		AN0 ~ AN7		A/D コンバータのアナログ入力端子です。この機能はアナログ入力指定が許可のときに有効になります。
47, 48, 53 ~ 58	45, 46, 51 ~ 56	P70 ~ P77	I (CMOS/H)	汎用の入出力ポートです。
		IRQ0 ~ IRQ7		外部割込み要求入力端子です。 外部割込みを許可している間はこの入力を随時使用しますので、意図的に行う以外は、ほかの機能による出力を停止させておく必要があります。
59, 60	57, 58	P80, P81	J (CMOS/H)	汎用の入出力ポートです。
		TIN0, TIN1		リロードタイマのイベント入力端子です。 リロードタイマが入力動作をしている間はこの入力を随時使用していますので、意図的に行う以外はほかの機能による出力を停止させておく必要があります。
61, 62	59, 60	P82, P83	J (CMOS/H)	汎用の入出力ポートです。
		TOT0, TOT1		リロードタイマの出力端子です。 この機能はリロードタイマの出力指定が許可のときに有効となります。
63 ~ 66	61 ~ 64	P84 ~ P87	J (CMOS/H)	汎用の入出力ポートです。
		IN0 ~ IN3		インプットキャプチャトリガ入力端子です。 この機能は、インプットキャプチャが入力動作をしている間はこの入力を随時使用していますので、意図的に行う以外はほかの機能による出力を停止させておく必要があります。
67, 68	65, 66	P90, P91	J (CMOS/H)	汎用の入出力ポートです。
		OUT0, OUT1		アウトプットコンペアイイベント出力端子です。
69 ~ 74	67 ~ 72	P92 ~ P97	J (CMOS/H)	汎用の入出力ポートです。
		PPG0 ~ PPG5		PPG 出力端子です。この機能は PPG の出力指定が許可のときに有効になります。
75, 76	73, 74	PA0, PA1	J (CMOS/H)	汎用の入出力ポートです。
		OUT2, OUT3		アウトプットコンペアイイベント出力端子です。
78, 79	76, 77	PA2, PA3	J (CMOS/H)	汎用の入出力ポートです。
80	78	PA4	J (CMOS/H)	汎用の入出力ポートです。
		CKOT		CKOT 動作時は、CKOT 出力として動作します。
34	32	AV _{CC}	-	A/D コンバータの電源端子です。

表 1.6-1 端子機能説明 (6 / 6)

QFP	LQFP	端子名	回路	機能説明
35	33	AVRH	-	A/D コンバータの外部基準電源端子です。
36	34	AVRL	-	A/D コンバータの外部基準電源端子です。
37	35	AV _{SS}	-	A/D コンバータの電源端子です。
49, 50	47, 48	MD0, MD1	C	動作モード指定用入力端子です。 V _{CC} または V _{SS} に直結してください。
51	49	MD2	K	動作モード指定用入力端子です。 V _{CC} または V _{SS} に直結してください (MB90552A/552B/553A/553B/V550A)。
			C	動作モード指定用入力端子です。 V _{CC} または V _{SS} に直結してください (MB90P553A/ F553A)。
23, 84	21, 82	V _{CC}	-	電源 (5V) 入力端子です。
11, 42, 81	9, 40, 79	V _{SS}	-	電源 (0V) 入力端子です。

1.7 入出力回路形式

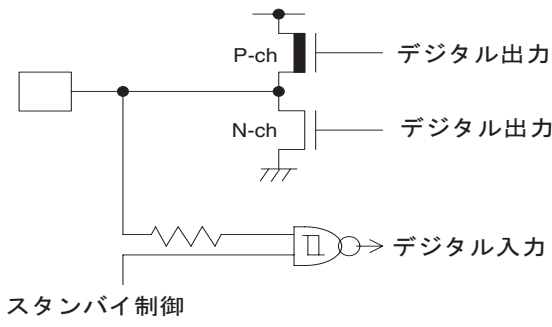
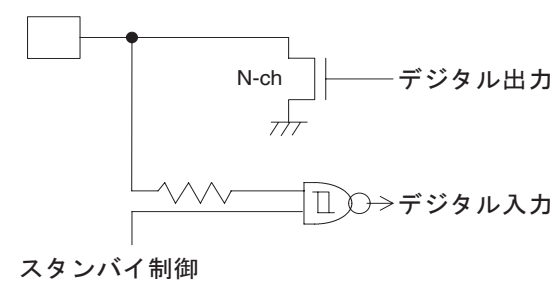
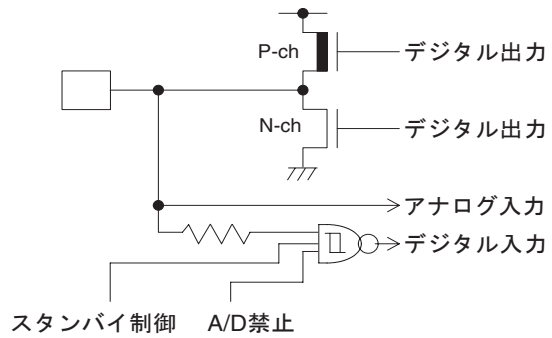
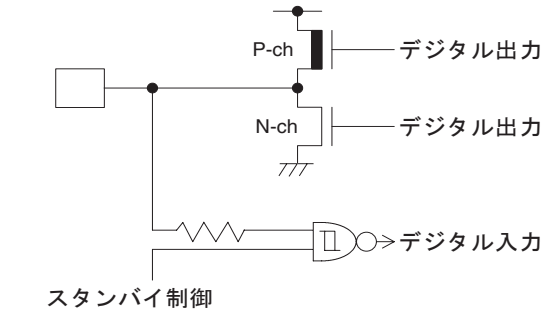
表 1.7-1 に，入出力回路形式を示します。

■ 入出力回路形式

表 1.7-1 入出力回路形式 (1 / 3)

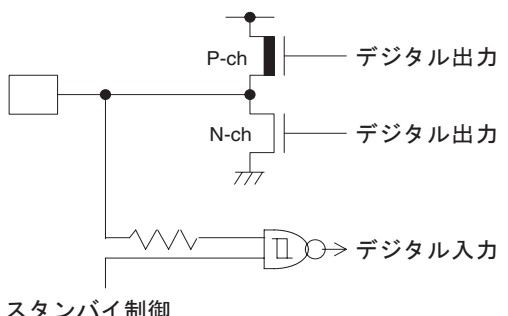
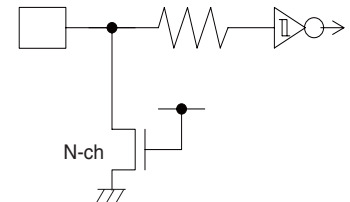
分類	回路形式	備考
A	<p>スタンバイ制御信号</p>	<ul style="list-style-type: none"> • 3MHz ~ 16MHz • 発振帰還抵抗 約 1M
B		<ul style="list-style-type: none"> • CMOS レベルヒステリシス入力 • プルアップあり 抵抗約 50k
C		<ul style="list-style-type: none"> • CMOS レベルヒステリシス入力
D	<p>プルアップ抵抗制御</p> <p>デジタル出力</p> <p>デジタル出力</p> <p>デジタル入力</p> <p>スタンバイ制御</p>	<ul style="list-style-type: none"> • CMOS レベル出力 • CMOS レベル入力 • スタンバイ制御あり • 入力プルアップ抵抗制御あり 抵抗 約 50k
E	<p>デジタル出力</p> <p>デジタル出力</p> <p>デジタル入力</p> <p>スタンバイ制御</p>	<ul style="list-style-type: none"> • CMOS レベル出力 • CMOS レベル入力 • スタンバイ制御あり

表 1.7-1 入出力回路形式 (2 / 3)

分類	回路形式	備考
F	 <p>スタンバイ制御</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 オープンドレイン制御あり
G	 <p>スタンバイ制御</p>	<ul style="list-style-type: none"> N-ch オープンドレイン出力 CMOS レベルヒステリシス入力 スタンバイ制御あり <p>(注意事項) 通常の CMOS 入出力端子と異なり P-ch トランジスタが存在しませんので、IC の電源が OFF の状態で、外部からこの端子に電圧が印加されても、V_{CC} 側への電流の流れ込みはありません。</p>
H	 <p>スタンバイ制御 A/D禁止</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 スタンバイ制御あり アナログ入力
I	 <p>スタンバイ制御</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 スタンバイ制御あり

第 1 章 概要

表 1.7-1 入出力回路形式 (3 / 3)

分類	回路形式	備考
J		<ul style="list-style-type: none"> • CMOS レベル出力 • CMOS レベルヒステリシス入力 • スタンバイ制御あり
K		<ul style="list-style-type: none"> • CMOS レベルヒステリシス入力 • プルダウンあり 抵抗 約 50 k

1.8 デバイスの取扱いに関する注意事項

MB90550A/B のデバイスを扱う際には、次の点について特に注意が必要です。

- ラッチアップの防止
- 供給電圧の安定化
- 未使用端子の処理
- 外部クロック使用時の注意
- 電源端子 (V_{CC}/V_{SS}) の取扱い
- 水晶発振回路
- A/D コンバータの電源・アナログ入力 of 投入順序
- スタンバイからの復帰
- 電源投入時の注意
- A/D コンバータの電源端子処理
- ポート 0, 1 からの出力が不定になる場合
- 「DIV A, Ri」, 「DIVW A, RWi」命令を使用する場合
- REALOS を使用する場合
- PLL クロックモード動作中の注意について

■ デバイスの取扱いに関する注意事項

● ラッチアップの防止

CMOS IC では、次に示すような場合にラッチアップ現象を生じることがあります。

- 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。
- $V_{CC} \sim V_{SS}$ 間に定格を超える電圧が印加された場合。
- V_{CC} の電圧より、 AV_{CC} の電源が先に供給された場合。

ラッチアップ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、使用に際しては十分に注意してください。

また、同一の理由により、アナログ電源電圧はデジタル電源電圧を超えないように十分に注意してください。

● 供給電圧の安定化

V_{CC} 電源電圧の動作保証範囲内においても、電源電圧が急激に変化した場合は、誤動作を起こしますので、 V_{CC} 電源電圧を安定させてください。安定化の基準は、商用周波数 (50Hz ~ 60Hz) での V_{CC} リプル変動 (peak to peak 値) を標準 V_{CC} 電源電圧値の 10% 以下に、また電源の切換えを行う場合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源電圧を安定させてください。

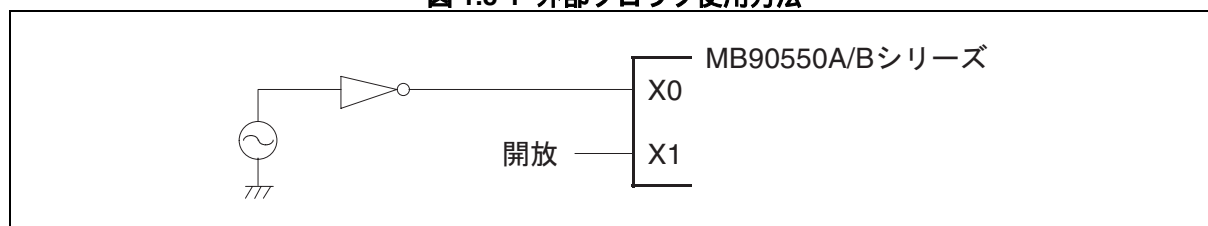
● 未使用端子の処理

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

● 外部クロック使用時の注意

外部クロックを使用する際には、X0 端子のみ駆動し、X1 端子は開放としてください。

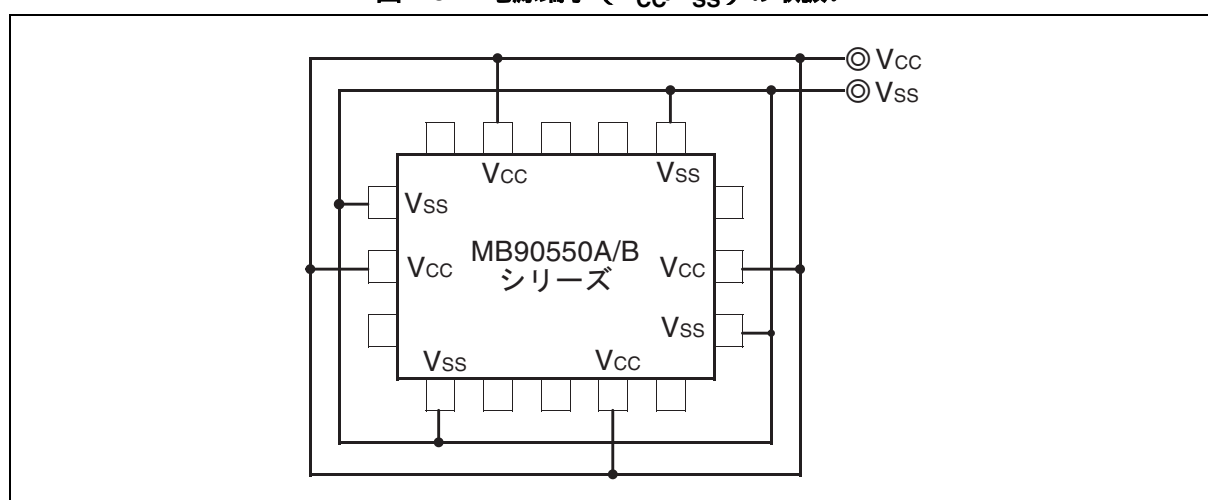
図 1.8-1 外部クロック使用方法



● 電源端子 (V_{CC}/V_{SS}) の取扱い

V_{CC} , V_{SS} が複数ある場合、デバイス設計上はラッチアップなどの誤動作を防止するために、デバイス内部で同電位にすべきもの同士を接続してありますが、不要輻射の低減・グラウンドレベルの上昇によるストローブ信号の誤動作の防止・総出力電流規格を遵守などのために、必ずそれらすべてを外部で電源およびグラウンドに接続してください。

図 1.8-2 電源端子 (V_{CC}/V_{SS}) の取扱い



また、電流供給源からできるかぎり低インピーダンスで本デバイスの V_{CC} , V_{SS} に接続するような配慮をお願い致します。

さらに、本デバイスの近くで、 V_{CC} と V_{SS} 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勧めいたします。

● 水晶発振回路

X0, X1 端子の近辺のノイズは本デバイスの誤動作のもととなります。X0 と X1 および水晶発振器（あるいはセラミック発振器）さらにグラウンドへのバイパスコンデンサはできるかぎり近くなるようにまた、その配線においてはほかの配線とできる限り交差しないようにプリント板を設計してください。

また、X0, X1 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、強くお勧めいたします。

● A/D コンバータの電源・アナログ入力投入順序

A/D コンバータの電源 (AV_{CC} , $AVRH$, $AVRL$) およびアナログ入力 ($AN0 \sim AN7$) の印加は、必ずデジタル電源 (V_{CC}) の投入後に行ってください。

また、電源切断時は、A/D コンバータの電源およびアナログ入力の切断の後で、デジタル電源の切断を行ってください。その際、 $AVRH$ は AV_{CC} を超えないように投入・切断を行ってください (アナログ電源とデジタル電源を同時に投入、切断することは問題ありません)。

● スタンバイからの復帰

スタンバイ状態時に電源電圧がスタンバイ RAM 保持電圧以下に低下した場合、スタンバイからの復帰ができなくなる場合があります。この場合、外部リセット端子よりリセットを印加することにより復帰させてください。

● 電源投入時の注意

内蔵している降圧回路の誤動作を防ぐために、電源投入時における電圧の立上り時間は $50\mu s$ ($0.2V \sim 2.7V$ の間) 以上を確保してください。

● A/D コンバータの電源端子処理

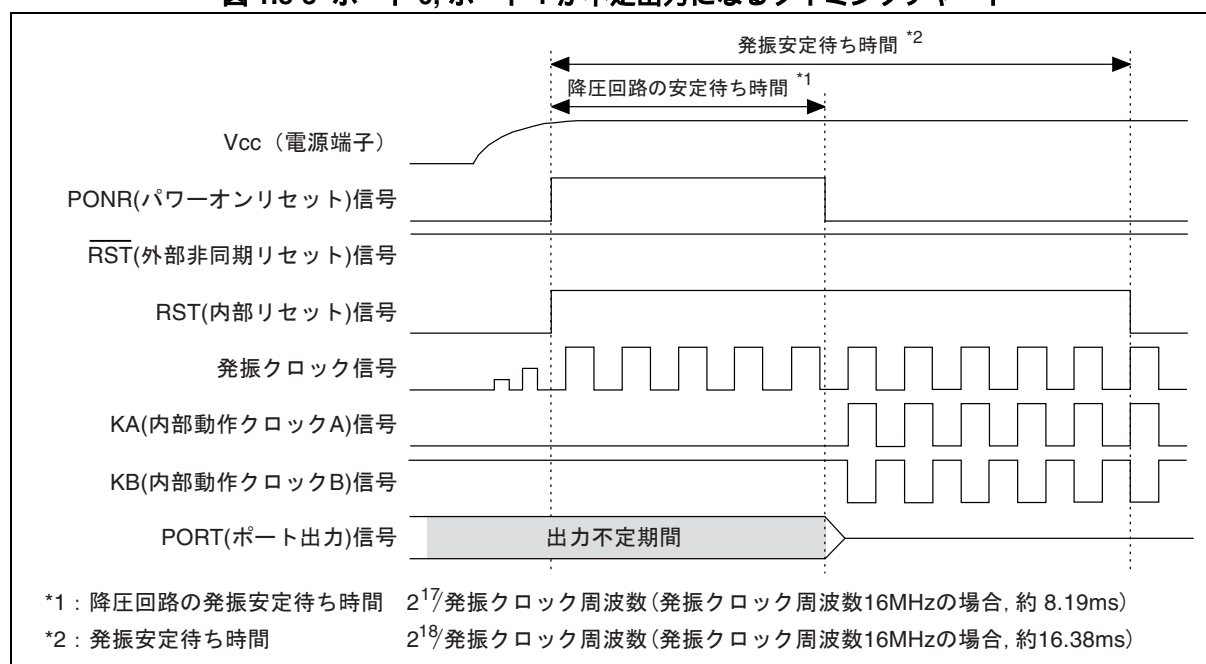
A/D コンバータを使用しない場合においても、 $AV_{CC}=V_{CC}$ 、 $AV_{SS}=AVRH=AVRL=V_{SS}$ とするように接続してください。

● ポート 0, 1 からの出力が不定になる場合

電源を投入後、降圧回路の発振安定待ち時間 (パワーオンリセット中) にポート 0, 1 から不定を出力します。タイミングは、図 1.8-3 のようになりますので注意してください (対象品種: MB90V550A, MB90F553A, MB90T553A, MB90553A, MB90553B, MB90T552A, MB90552A, MB90552B)。

なお、降圧回路を内蔵していない品種では、降圧回路の発振安定待ち時間がありませんので、不定を出力しません (対象品種: MB90P553A)。

図 1.8-3 ポート 0, ポート 1 が不定出力になるタイミングチャート



- 「DIV A, Ri」, 「DIVW A, RWi」命令を使用する場合

符号付乗除算命令「DIV A, Ri」, 「DIVW A, RWi」命令は, 対応するバンクレジスタ (DTB, ADB, USB, SSB) の値を "00_H" に設定し, 使用してください。

対応するバンクレジスタ (DTB, ADB, USB, SSB) の値を "00_H" 以外に設定した場合, 命令実行結果により得られる余りは, 命令オペランドのレジスタに格納されません。

詳細は, 「2.8 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意」を参照してください。

- REALOS を使用する場合

REALOS を使用する場合は, 拡張インテリジェント I/O サービス (EI²OS) が使用できません。

- PLL クロックモード動作中の注意について

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

第2章

CPU

CPU の機能と動作について説明します。

- 2.1 メモリ空間
- 2.2 アドレス指定
- 2.3 メモリ空間における多バイト長データの配置
- 2.4 専用レジスタ
- 2.5 汎用レジスタ
- 2.6 プリフィックスコード
- 2.7 割込み抑止命令とプリフィックスコード
- 2.8 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意

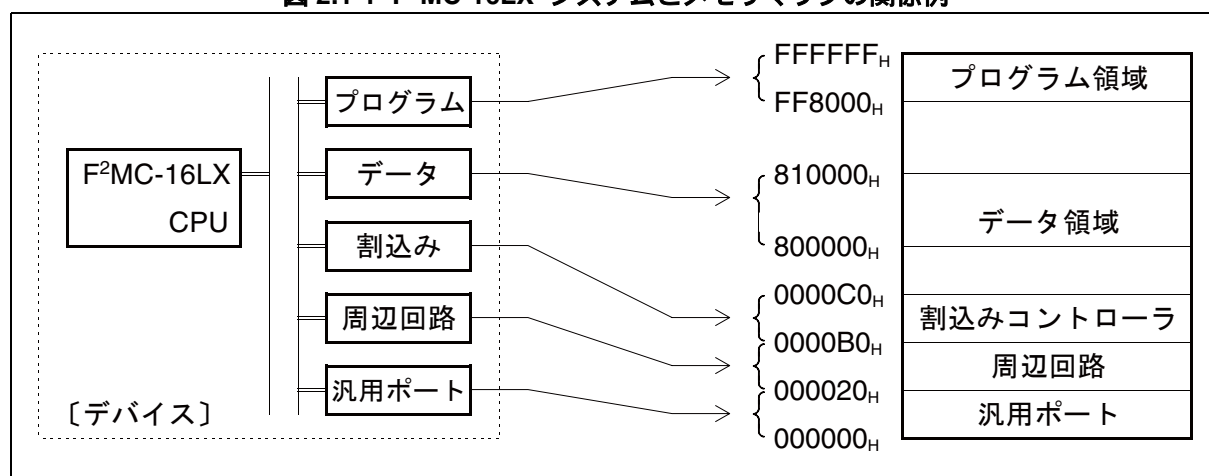
2.1 メモリ空間

F²MC-16LX CPU コアは、民生用・車載用機器等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 16bit 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ており、各種制御の高速・高効率処理が可能です。F²MC-16LX は 16 ビットデータ処理は勿論、内部に 32 ビットアキュムレータを搭載しているため 32 ビットデータ処理も可能です。メモリ空間は最大 16Mbyte（拡張可能）、リニア方式およびバンク方式のいずれかにてアクセス可能です。また、命令体系は F²MC-8L の A-T 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・ビット処理の充実化により命令が強化されています。

■ メモリ空間

F²MC-16LX CPU が管理するデータ・プログラム I/O は、すべて F²MC-16LX CPU が持つ 16Mbyte のメモリ空間のいずれかに配置されます。CPU は 24bit のアドレスバスでこれらのアドレスを示すことにより、各リソースをアクセスすることができます。

図 2.1-1 F²MC-16LX システムとメモリマップの関係例



2.2 アドレス指定

F²MC-16LX のアドレス指定には、以下に示す 2 つの方式があります。

- リニア方式：24bit アドレスすべてを命令により指定する方式
- バンク方式：アドレス上位 8bit を用途に応じたバンクレジスタで、アドレス下位 16bit を命令により指定する方式

■ リニア方式によるアドレス指定

リニア方式には以下に示す 2 つの方式があります。

- 24 ビットオペランド指定：オペランドにて直接 24bit のアドレスを指定する方式
- 32 ビットレジスタ間接指定：32bit の汎用レジスタの内容の下位 24bit をアドレスとして引用する方式

図 2.2-1 リニア方式の 24 ビットオペランド指定の例

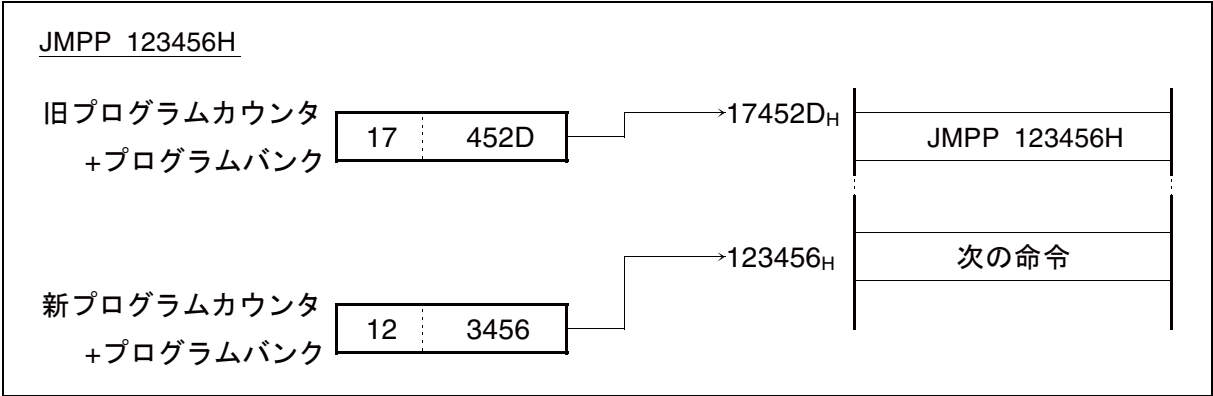
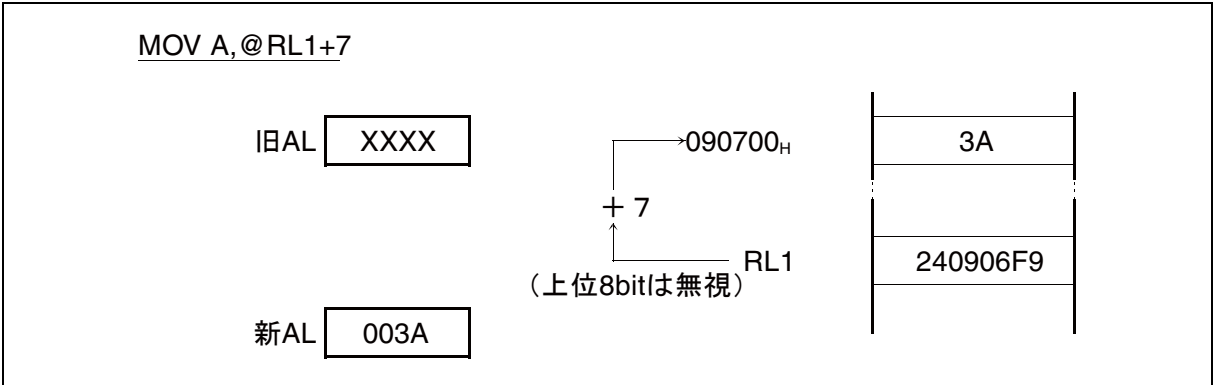


図 2.2-2 リニア方式の 32 ビットレジスタ間接指定の例



■ バンク方式によるアドレス指定

バンク方式は 16Mbyte の空間を 64Kbyte ごとの 256 個のバンクに分割し、以下に示す 5 つのバンクレジスタで各空間に対応するバンクを指定します。

- プログラムバンクレジスタ (PCB) リセット時初期値 FF_H

PCB によって指定される 64Kbyte のバンクをプログラム (PC) 空間とよび、主として命令コードやベクタテーブル、即値データなどが存在します。

- データバンクレジスタ (DTB) リセット時初期値 00_H

DTB によって指定される 64Kbyte のバンクをデータ (DT) 空間とよび、主として読み書き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。

- ユーザスタックバンクレジスタ (USB) リセット時初期値 00_H / システムスタックバンクレジスタ (SSB) リセット時初期値 00_H

USB あるいは SSB によって指定される 64Kbyte のバンクをスタック (SP) 空間とよび、プッシュ / ポップ命令や割込みのレジスタ退避などの際にスタックアクセスが生じた場合にアクセスされる領域です。どちらの空間が使用されるかはコンディションコードレジスタ中の S フラグの値に依存します。

- アディショナルデータバンクレジスタ (ADB) リセット時初期値 00_H

ADB によって指定される 64Kbyte のバンクをアディショナル (AD) 空間とよび、主として DT 空間に入りきらなかったデータなどが存在します。

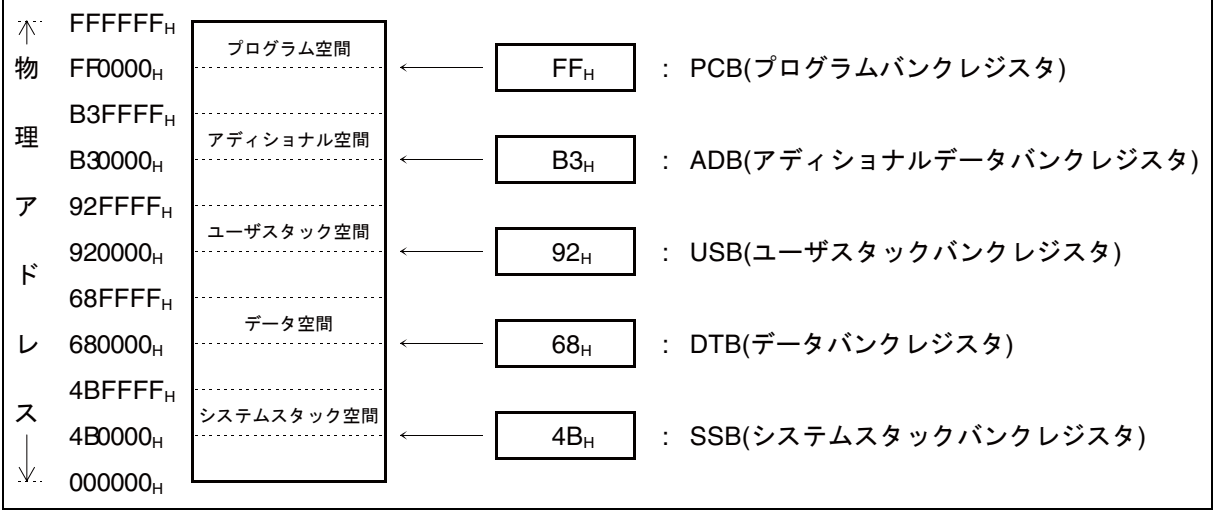
また、命令のコード効率を向上するために各命令はアドレッシングごとに以下に示すようなデフォルト空間が決められています。また、あるアドレッシングを用いたときにデフォルト以外の空間を使用したいときは、各バンクに対応しているプリフィックスコードを命令に先行して指定することによりそのプリフィックスコードに対応した任意のバンク空間をアクセスすることができます。

DTB, USB, SSB, ADB はリセットにより 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
プログラム空間	PC 間接、プログラムアクセス、分岐系
データ空間	@RW0, @RW1, @RW4, @RW5 を用いたアドレッシング, @A, addr16, dir
スタック空間	PUSHW, POPW, @RW3, @RW7 を用いたアドレッシング
アディショナル空間	@RW2, @RW6 を用いたアドレッシング

図 2.2-3 各空間の物理アドレスの例



2.3 メモリ空間における多バイト長データの配置

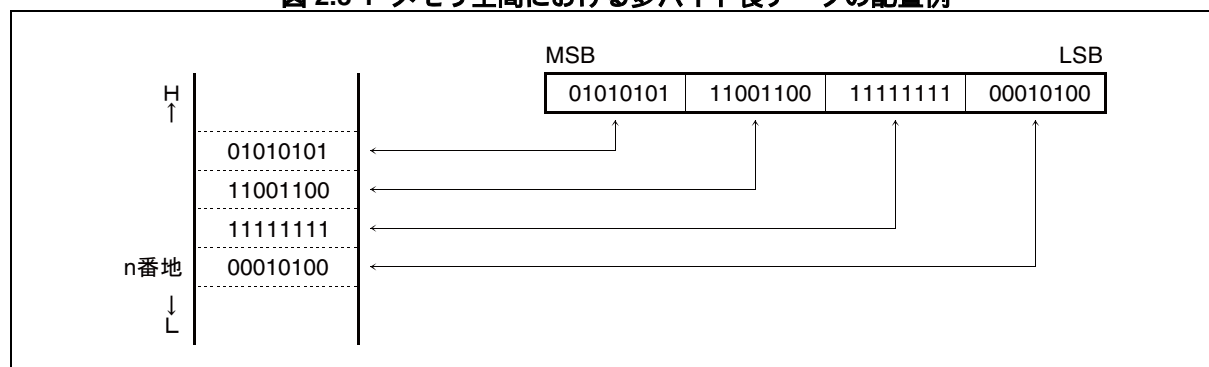
メモリ空間における多バイト長データは、下位 8bit が n 番地に、以下 $n+1$ 番地、 $n+2$ 番地、 $n+3$ 番地・・・の順に配置されます。

■ メモリ空間における多バイト長データの配置

図 2.3-1 に示すように、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 ビット長データであれば下位 16bit が先に転送され、続いて上位 16bit が転送されます。

また、下位データの書込み直後にリセット信号を入力すると上位データが書き込まれないことがあります。したがって、データを正しく保持するためには、上位データの書込み終了後にリセット信号を入力する必要があります。

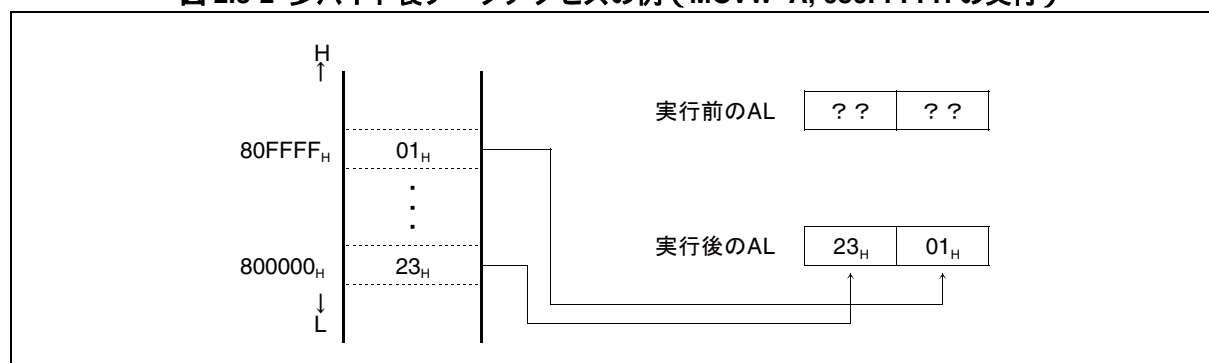
図 2.3-1 メモリ空間における多バイト長データの配置例



■ 多バイト長データのアクセス

図 2.3-2 に示すように、アクセスはすべてバンク内を基本に行われますので、多バイト長のデータをアクセスする命令では、 FFFF_H 番地の次のアドレスは同じバンクの 0000_H 番地になります。

図 2.3-2 多バイト長データアクセスの例 (MOVW A, 080FFFFH の実行)



2.4 専用レジスタ

専用レジスタとは、CPU の内部に専用ハードウェアとして存在し、使用する用途が CPU のアーキテクチャ上で限定されているものです。

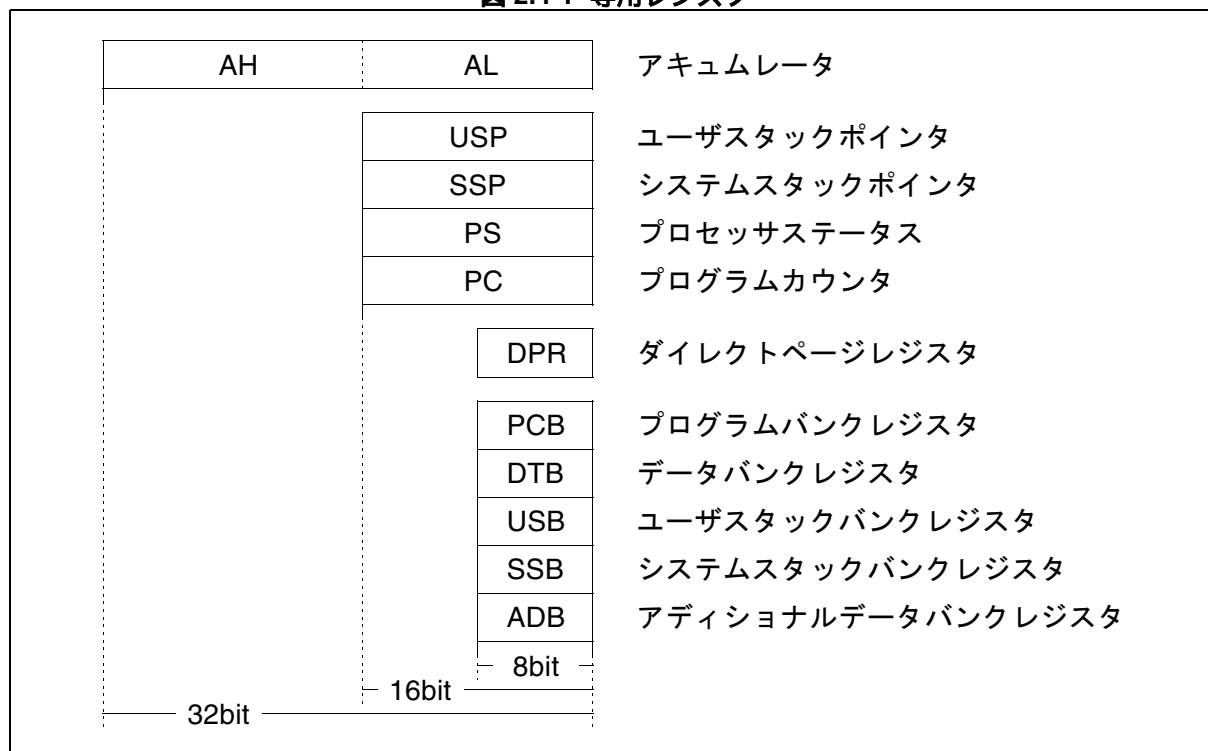
■ 専用レジスタ

専用レジスタとは、CPU の内部に専用ハードウェアとして存在し、使用する用途が CPU のアーキテクチャ上で限定されているものです。

F²MC-16LX の専用レジスタは、以下に示す 13 本があります。

- アキュムレータ (A=AH:AL)
16bit × 2 本のアキュムレータです (合計 32bit のアキュムレータとしても使用可能)。
- ユーザスタックポインタ (USP)
ユーザスタック領域を示す 16bit のポインタです。
- 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 16bit のポインタです。
- プロセッサステータス (PS)
システムの状態を示す 16bit のレジスタです。
- プログラムカウンタ (PC)
プログラムが格納されているアドレスを持つ 16bit のレジスタです。
- プログラムバンクレジスタ (PCB)
PC 空間を示す 8bit のレジスタです。
- データバンクレジスタ (DTB)
DT 空間を示す 8bit のレジスタです。
- 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 8bit のレジスタです。
- 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 8bit のレジスタです。
- アディショナルデータバンクレジスタ (ADB)
AD 空間を示す 8bit のレジスタです。
- ダイレクトページレジスタ (DPR)
ダイレクトページを示す 8bit のレジスタです。

図 2.4-1 専用レジスタ



2.4.1 アキュムレータ (A)

アキュムレータ (A) は、2 つの 16bit 長の演算用レジスタ AH および AL で構成され、演算などを行った際の結果やデータ転送の一時記憶などに使用されます。32bit データ処理時は AH と AL を連結して使用し、16bit データ処理のワード処理や 8bit データ処理のバイト処理の時は AL のみが使用されます。

■ アキュムレータ (A)

アキュムレータ (A) 中のデータは、メモリ / レジスタ (Ri, RWi, RLi) 中のデータと各種演算ができます。F²MC-8 と同様に、F²MC-16LX でも基本的にワード長以下のデータを AL へ転送すると、転送前の AL 中のデータが自動的に AH に転送されます (データ保持機能) ので、データ保持機能と AL-AH 間演算にて各種処理効率を上げることが可能です。

図 2.4-2 32bit データ転送例

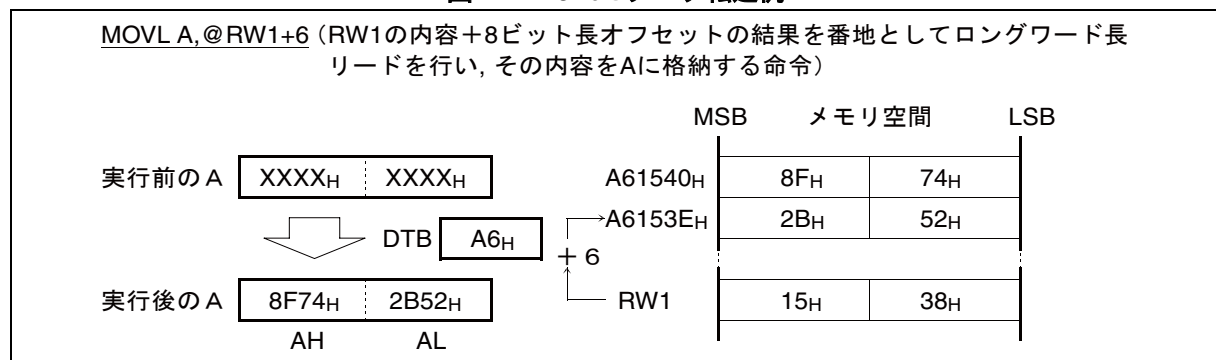


図 2.4-3 AL-AH 転送例

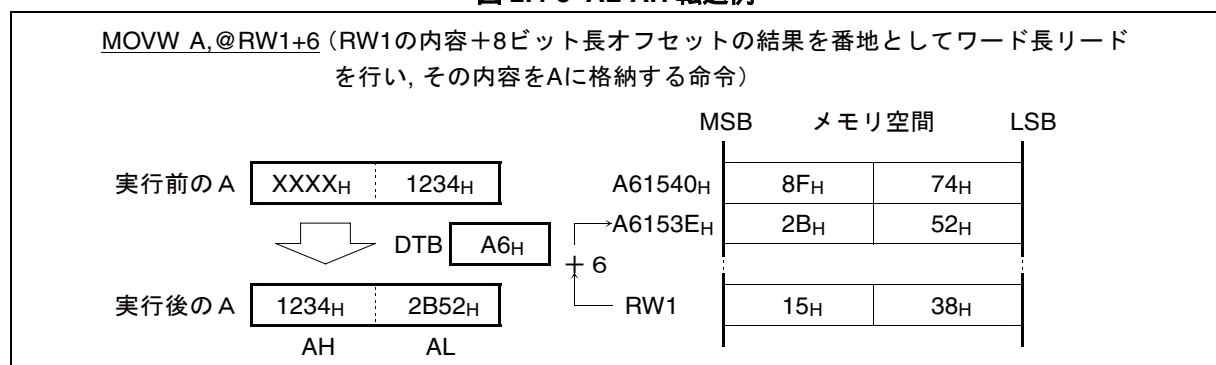


図 2.4-4 に示すように、ALへバイト長以下のデータを転送するとき、データは符号拡張またはゼロ拡張され 16bit 長となって AL へ格納されます。AL 中のデータは、ワード長としてもバイト長としても扱えます。バイト処理の算術演算命令を実行すると、演算前の AL の上位 8bit は無視され演算結果の上位 8bit はすべて "0" になります。

なお、アキュムレータ (A) は、リセットでは初期化されず、リセット直後は不定値になります (図 2.4-5 を参照)。

図 2.4-4 ゼロ拡張実行例

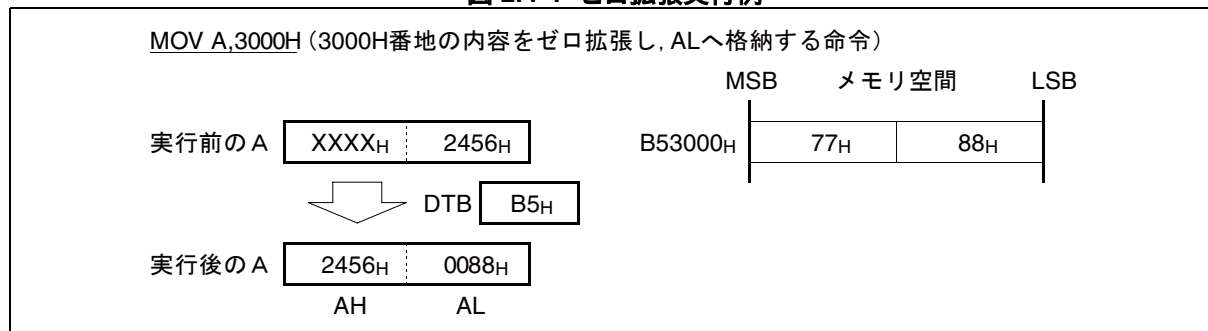
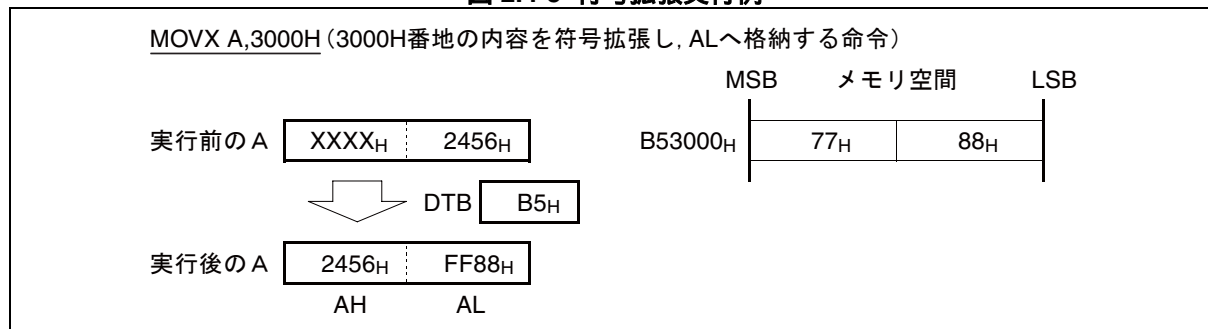


図 2.4-5 符号拡張実行例



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

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) は、16bit のレジスタであり、プッシュ / ポップ命令およびサブルーチン実行時のデータ退避 / 復帰のメモリアドレスを示します。

■ 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) とシステムスタックポインタ (SSP) 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中の S フラグが "0" のときは、USP レジスタが有効になり、S フラグが "1" のときは、SSP レジスタが有効になります (図 2.4-6 と図 2.4-7 を参照)。また、割込みが受け付けられると S フラグがセットされるため、割込み時のレジスタ退避は必ず SSP の示すメモリ中に行われます。割込みルーチンでのスタック処理は SSP が用いられ、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る必要のない場合は SSP だけをお使いください。スタック時のアドレスの上位 8bit 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 2.4-6 スタック操作命令とスタックポインタ (S フラグが "0" のときの PUSHW A 例)

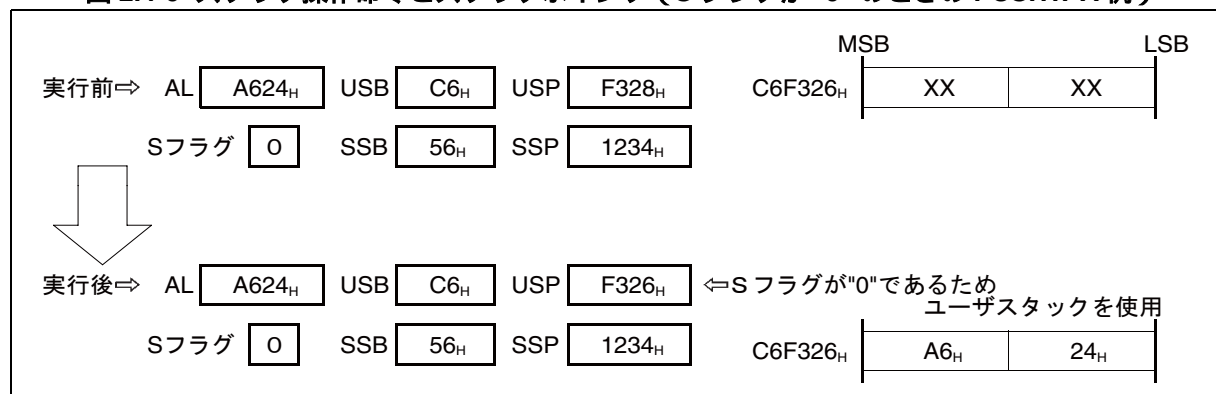
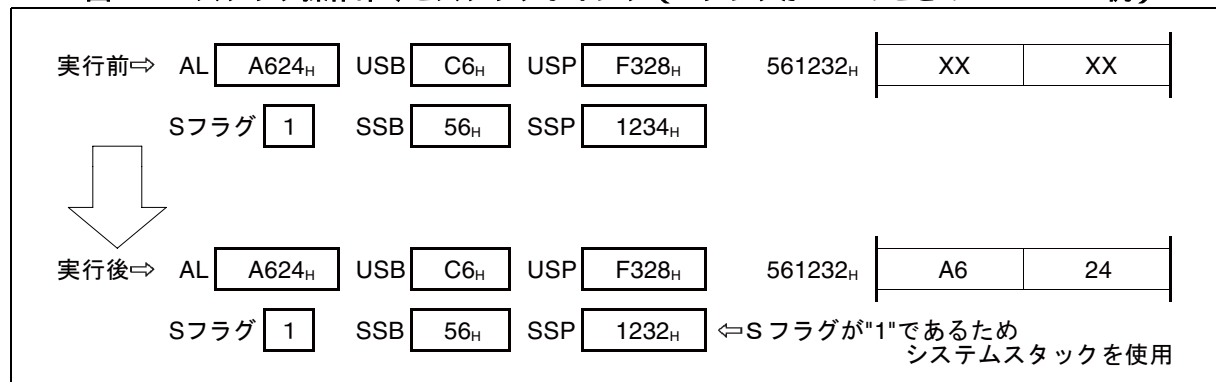


図 2.4-7 スタック操作命令とスタックポインタ (S フラグが "1" のときの PUSHW A 例)



< 注意事項 >

スタックポインタに設定すべき値は、原則として偶数アドレスを使用してください。

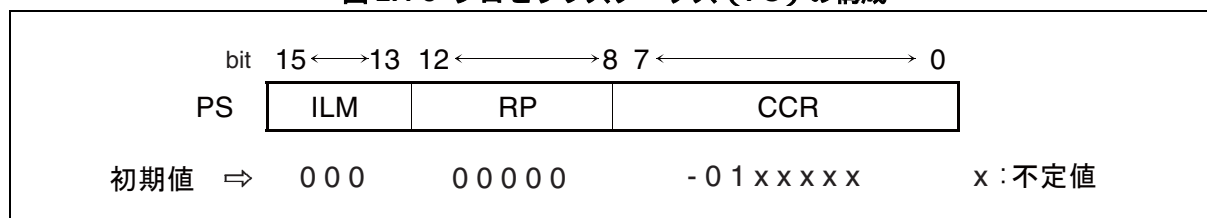
2.4.3 プロセッサステータス (PS)

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

■ プロセッサステータス (PS)

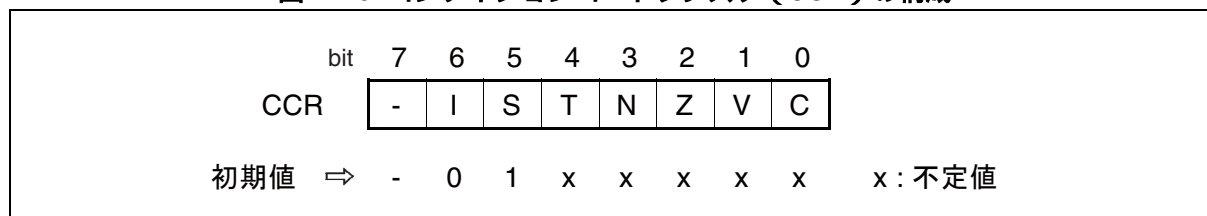
プロセッサステータス (PS) の上位バイトは, レジスタバンクの先頭アドレスを示すレジスタバンクポインタ (RP) およびインタラプトレベルマスクレジスタ (ILM) より構成され, PS の下位バイトは命令実行結果および割込み発生などによりセット / リセットされる各種フラグより構成されているコンディションコードレジスタ (CCR) より構成されています。

図 2.4-8 プロセッサステータス (PS) の構成



■ コンディションコードレジスタ (CCR)

図 2.4-9 コンディションコードレジスタ (CCR) の構成



● 割込み許可フラグ (I)

ソフトウェアインタラプト以外のすべての割込み要求に対し, I が "1" のときは割込みが許可され, "0" のときは割込みがマスクされ, リセット時にはクリアされます。

● スタックフラグ (S)

S が "0" のときは, スタック操作ポインタとして USP が有効になり, "1" のときは SSP が有効になります。また, 割込み受付け時およびリセット時にはセットされます。

● スティックイビットフラグ (T)

論理右 / 算術右シフト命令を実行後に, キャリよりシフトアウトされたデータに 1 つ以上 "1" があったら "1", それ以外は "0" になります。また, シフト量がゼロの場合も "0" になります。

● ネガティブフラグ (N)

演算結果の MSB が "1" ならセットされ, "0" ならクリアされます。

● ゼロフラグ (Z)

演算結果がすべて "0" ならセットされ、それ以外はクリアされます。

● オーバフローフラグ (V)

演算の実行により符号付き数値としてオーバーフローが生じるとセットされ、生じないとクリアされます。

● キャリーフラグ (C)

演算の実行により、MSB より桁上り/桁下りが生じるとセットされ、生じないとクリアされます。

■ 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 の持つ汎用レジスタとそれが存在する内部 RAM のアドレスとの関係を示すレジスタで、現在使用中のレジスタバンクの先頭のメモリアドレスを $[000180_H + (RP) \times 10_H]$ という変換式で示します。RP は 5bit で構成され、 $00_H \sim 1F_H$ までの値をとることができ、また、 $000180_H \sim 00037F_H$ 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。RP はリセットによりすべて "0" に初期化されます。命令上では RP に 8bit の即値が転送できますが、実際に使用されるのはそのデータの低位 5bit のみです。

図 2.4-10 レジスタバンクポインタ (RP) の構成

	bit	12	11	10	9	8
RP		B4	B3	B2	B1	B0
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 3bit 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 3bit により示されるレベルより強いレベルの割込み要求のみが受け付けられます。表 2.4-1 に示すように、レベルの強弱は "0" が最強で、"7" が最弱と定義されています。したがって、割込みが受け付けられるには、現状の ILM の保持値より小さい値の要求でなければなりません。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、これ以降の同じかそれより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 "0" に初期化されます。命令上では ILM に 8bit の即値が転送できますが、実際に使用されるのはそのデータの低位 3bit のみです。

図 2.4-11 インタラプトレベルマスクレジスタ (ILM) の構成

	bit	15	14	13
ILM		ILM2	ILM1	ILM0
初期値		0	0	0

表 2.4-1 インタラプトレベルマスクレジスタ (ILM) で示されるレベルの強弱

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 より小さい値のレベル
0	1	1	3	2 より小さい値のレベル
1	0	0	4	3 より小さい値のレベル
1	0	1	5	4 より小さい値のレベル
1	1	0	6	5 より小さい値のレベル
1	1	1	7	6 より小さい値のレベル

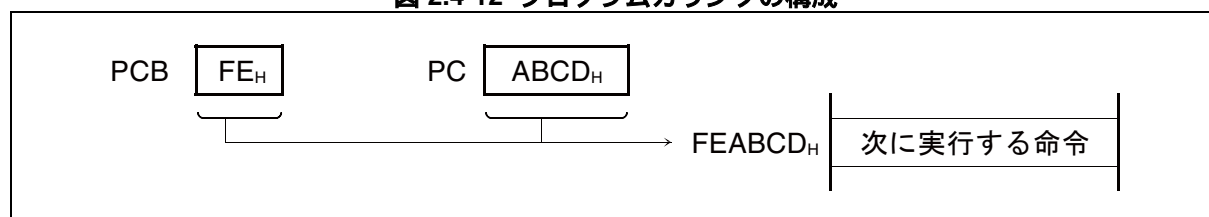
2.4.4 プログラムカウンタ (PC)

プログラムカウンタ (PC) は、16bit 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 16bit を示します。上位 8bit アドレスは PCB により示されます。

■ プログラムカウンタ (PC)

プログラムカウンタ (PC) は、条件分岐命令、サブルーチンコール命令、割込み、および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トとして使用することもできます。

図 2.4-12 プログラムカウンタの構成



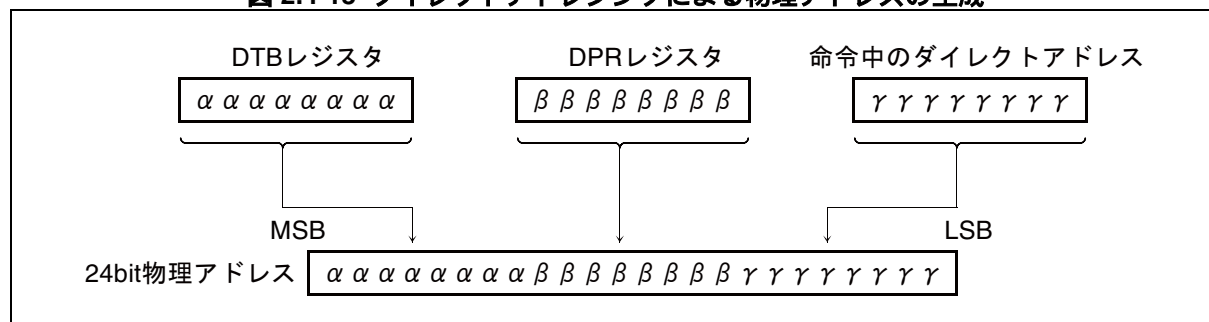
2.4.5 ダイレクトページレジスタ (DPR)

ダイレクトページレジスタ (DPR) は、ダイレクトアドレッシングの命令時に、オペランドの `addr8 ~ addr15` を指定します。DPR は 8bit 長であり、リセットにより `01H` に初期化されます。また、命令により読出し / 書込みが可能です。

■ ダイレクトページレジスタ (DPR)

図 2.4-13 に、ダイレクトアドレッシングによる物理アドレスの生成を示します。

図 2.4-13 ダイレクトアドレッシングによる物理アドレスの生成



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

バンクレジスタは、バンクアドレッシングで 24 ビットアドレスの最上位 8 ビットを設定します。次の 5 つのレジスタで構成されます。

- プログラムバンクレジスタ (PCB)
- データバンクレジスタ (DTB)
- ユーザスタックバンクレジスタ (USB)
- システムスタックバンクレジスタ (SSB)
- アディショナルデータバンクレジスタ (ADB)

各バンクレジスタは、それぞれプログラム、データ、ユーザスタック、システムスタックおよびアディショナルの、各空間が配置されるメモリバンクを示します。

■ プログラムバンクレジスタ (PCB)

プログラムバンクレジスタ (PCB) は、プログラム (PC) 空間を設定するバンクレジスタです。

16M バイト全空間に分岐する、JMPP, CALLP, RETP および RETI 命令実行時、ソフトウェア割込み命令実行時、ハードウェア割込み時および例外処理割込み発生時に書き換わります。

■ データバンクレジスタ (DTB)

データバンクレジスタ (DTB) は、データ (DT) 空間を設定するバンクレジスタです。

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

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) は、スタック (SP) 空間を設定するバンクレジスタです。スタックフラグ (CCR:S) の値によって使用されるバンクレジスタが決定されます。

■ アディショナルデータバンクレジスタ (ADB)

アディショナルデータバンクレジスタ (ADB) は、アディショナル (AD) 空間を設定するバンクレジスタです。

■ 各バンクの設定とデータアクセス

バンクレジスタは 8 ビット長であり、リセット時にプログラムバンクレジスタ (PCB) は "FF_H" に初期化され、その他のバンクレジスタは "00_H" に初期化されます。

プログラムバンクレジスタ (PCB) は、読出し専用です。PCB 以外のバンクレジスタは、読出しと書込みが行えます。

各バンクレジスタの動作については、「2.1 メモリ空間」を参照してください。

2.5 汎用レジスタ

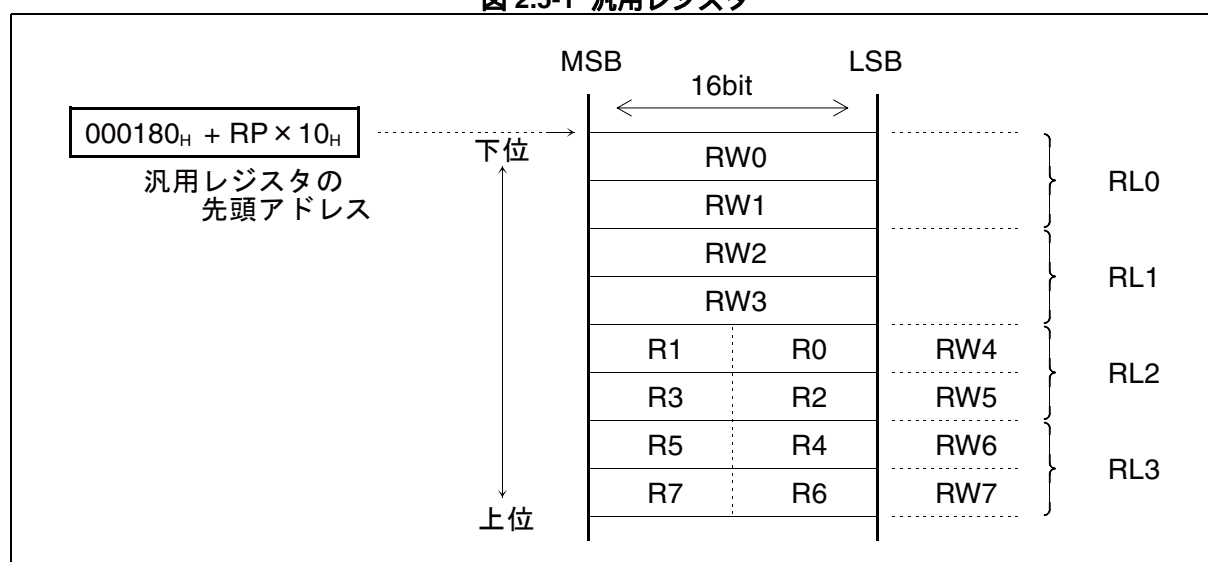
汎用レジスタとは、通常のメモリと同じく、使用する用途をユーザが指定することができるものです。CPU のアドレス空間上に RAM と共存し、アドレスを指定しないでアクセスできるという点では専用レジスタと同じです。

■ 汎用レジスタ

F²MC-16LX の汎用レジスタは、主記憶の 000180_H ~ 00037F_H(最大の場合) に存在します。レジスタバンクポインタ (RP) によって、先に述べたアドレスのどの部分が現在使用中のレジスタバンクであるかを指定します。各バンクには以下に示す 3 種のレジスタが存在します。これらは独立ではなく、以下に示すような関係があります。

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

図 2.5-1 汎用レジスタ



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

$$RW_{(i+4)} = R_{(i \times 2 + 1)} \times 256 + R_{(i \times 2)} \quad [i=0 \sim 3]$$

また、RL_i の上位・下位と RW の関係は、次の式で表すことができます。

$$RL_{(i)} = RW_{(i \times 2 + 1)} \times 65536 + RW_{(i \times 2)} \quad [i=0 \sim 3]$$

■ レジスタバンク

レジスタバンクは、8word で構成されています。レジスタバンクの内容は、通常の RAM 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、不定値になります。

表 2.5-1 に示すように、レジスタバンクは、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) の汎用レジスタとして使用でき、また、各種演算、ポインタとして各種命令に使用できます。また、RL0 ~ RL3 は、全空間を直接アクセスするリニアポインタとしても使用できます。

表 2.5-1 レジスタバンクの機能

レジスタ	機能
R0 ~ R7	各種命令のオペランドとして使用
RW0 ~ RW7	ポインタ、各種命令のオペランドとして使用
RL0 ~ RL3	ロングポインタ、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およびノーマライズ命令のカウンタとしても使用します。
 - RW0 はストリング命令のカウンタとしても使用します。
-

2.6 プリフィックスコード

プリフィックスコードには、バンクセレクトプリフィックス、コモンレジスタバンクプリフィックス、フラグ変化抑止プリフィックスの3種類があります。命令の前に、これらのプリフィックスコードを置くことで、動作の一部を変更することが可能です。

■ バンクセレクトプリフィックス

データアクセスの際に用いられるメモリ空間は、アドレッシングごとに定められています。バンクセレクトプリフィックスを命令の前に置くことで、命令によるデータアクセスのメモリ空間をアドレッシングモードと無関係に、任意に選択することが可能です。

表 2.6-1 に、バンクセレクトプリフィックスとそれにより選択されるメモリ空間を示します。

表 2.6-1 バンクセレクトプリフィックス

バンクセレクト プリフィックス	選択される空間
PCB	プログラム空間
DTB	データ空間
ADB	アデショナル空間
SPB	そのときのスタックフラグの内容により、システムスタック空間、ユーザスタック空間のどちらかが用いられます。

バンクセレクトプリフィックスを使用する際には、以下に示す命令に対して注意してください。

- スtring命令〔MOV_S, MOV_{SW}, SCEQ, SCWEQ, FILS, FILSW〕

プリフィックスの有無にかかわらず、オペランド指定されたバンクレジスタを使用してください。

- スタック操作命令〔PUSHW, POPW〕

プリフィックスの有無にかかわらず、S フラグに応じて SSB または USB を使用してください。

- I/O アクセス命令〔MOV A, io/MOV io, A/MOVX A, io/MOVW A, io/MOVW io, A/MOV io, #imm8 / MOVW io, #imm16 / MOV_B A, io:bp / MOV_B io:bp, A / SET_B io:bp / CLR_B io:bp / BBC io:bp, rel / BBS io:bp, rel / WBTC, WBTS〕

プリフィックスの有無にかかわらず、バンクの I/O 空間が使用されます。

- 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次の命令まで及びます。

● POPW ps

プリフィックスの有無にかかわらず、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次の命令まで及びます。

● RETI

プリフィックスの有無にかかわらず、SSB が使用できます。

■ コモンレジスタバンクプリフィックス (CMR)

複数のタスク間でのデータ交換を容易にするためには、そのときの RP がどのような値でも、比較的簡単に定められた同一のレジスタバンクをアクセスする手段が必要です。コモンレジスタバンクプリフィックス (CMR) を、レジスタバンクをアクセスする命令の前に置くことで、現在の RP の値に関係なくその命令のレジスタアクセスをすべて 000180_H ~ 00018F_H 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) に変更することができます。

コモンレジスタバンクプリフィックス (CMR) を使用する際には、以下に示す命令に対して注意してください。

● スtring命令 [MOV S, MOV SW, 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、割り込み復帰後の String 命令に対しては、プリフィックスが無効であるため誤動作となります。String 命令 [MOV S, MOV SW, SCEQ, SCWEQ, FILS, FILSW] 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/POPW PS]

命令動作は正常ですが、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ためには、フラグ変化抑止プリフィックスコード (NCC) を用います。不要フラグ変化を抑止する命令の前に置くことで、命令実行に伴うフラグ変化を抑止可能です。

フラグ変化抑止プリフィックス (NCC) を使用する際には、以下に示す命令に対して注意してください。

● String命令 [MOV S, MOV SW, SCEQ, SCWEQ, FILS, FILSW]

プリフィックスコードを付加した String 命令の実行中に割り込み要求が発生すると、割り込み復帰後の String 命令に対してはプリフィックスが無効であるために、誤動作となります。String 命令 [MOV S, MOV SW, SCEQ, SCWEQ, FILS, FILSW] には、NCC プリフィックスを付加しないでください。

● フラグ変更命令 [AND CCR, #imm8/OR CCR, #imm8/POPW PS]

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

- 割込み命令〔INT #vct8/INT9/INT addr16/INTP addr24/RETI〕

プリフィックスの有無にかかわらず,CCR は命令の仕様どおり変化します。

- JCTX @A

プリフィックスの有無にかかわらず,CCR は命令の仕様どおり変化します。

- MOV ILM, imm8

命令動作は正常ですが,プリフィックスの効果が次の命令まで及びます。

2.7 割込み抑止命令とプリフィックスコード

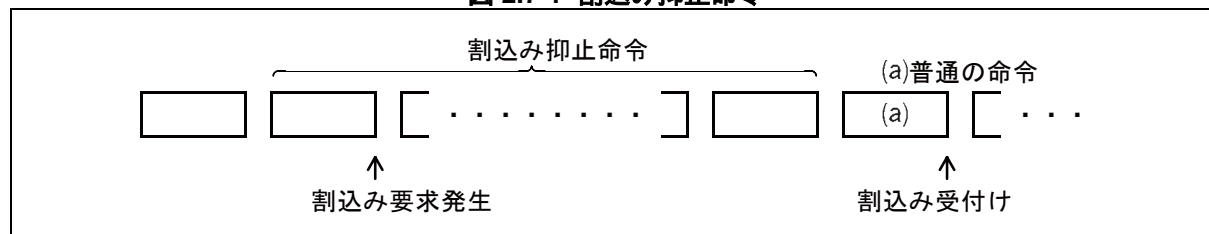
割込み抑止命令は、ハードウェア割込み要求の有無を検出せず、割込み要求を無視します。割込み抑止命令には、次の10種類があります。

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

■ 割込み抑止命令

図 2.7-1 に示すように、割込み抑止命令を実行中に、有効なハードウェア割込み要求が発生しても、割込み処理が行われるのは、割込み抑止命令以降に、割込み抑止命令以外の命令が実行された後になります。

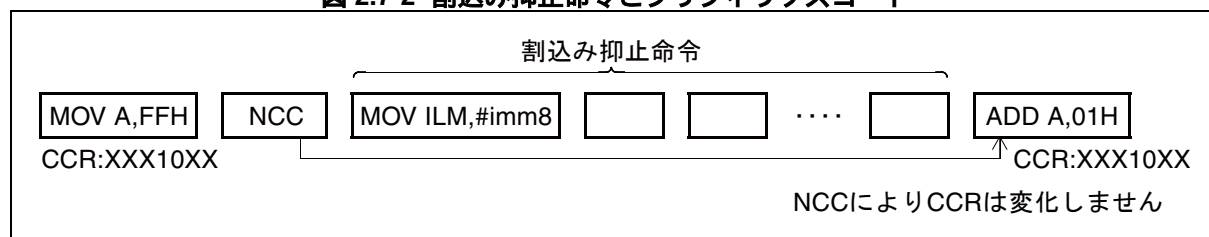
図 2.7-1 割込み抑止命令



■ 割込み抑止命令とプリフィックス命令に関する制約

図 2.7-2 に示すように、割込み抑止命令の前にプリフィックスコードを付加した場合、プリフィックスコードの効果は、プリフィックスコード後の最初の（割込み抑止命令以外の命令）まで及びます。

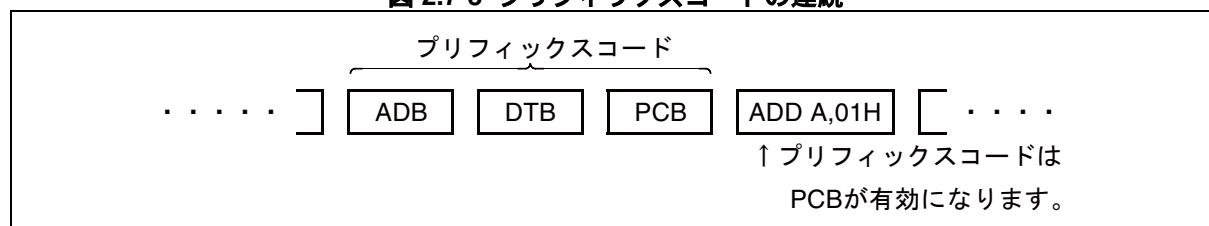
図 2.7-2 割込み抑止命令とプリフィックスコード



■ プリフィックスコードが連続している場合

図 2.7-3 に示すように、競合するプリフィックスコードが連続していた場合、後方の方が有効になります。競合するプリフィックスコードとはここでいう PCB, ADB, DTB, SPB のことを意味します。

図 2.7-3 プリフィックスコードの連続



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

「DIV A, Ri」, 「DIVW A, RWi」命令を使用する場合は, バンクレジスタを "00_H" に設定してください。

■「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意

表 2.8-1 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意 (i=0 ~ 7)

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

*1:CCR レジスタの S ビットによる

*2:CCR レジスタの S ビットが "0" の場合

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

【例】

DTB=053_H, RP=03_H の場合に「DIV A, R0」を実行した場合は, R0 のアドレスが 0180_H+RP (03_H) × 10_H+08_H (R0 相当アドレス) =0001B8_H となります。

ここで, 「DIV A, R0」で指定されるバンクレジスタはデータバンクレジスタ (DTB) ですので, バンクアドレス 053_H を付加したアドレス 05301B8_H に余りが格納されます。

< 参考 >

- バンクレジスタについては, 「2.4.6 バンクレジスタ (PCB, DTB, USB, SSB, ADB) 」を参照してください。
 - Ri および RWi のレジスタについては, 「2.5 汎用レジスタ」を参照してください。
-

■ 注意事項の回避について

「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意事項を回避してプログラムを開発していただくために, コンパイラでは表 2.8-1 の命令を生成しないように変更し, アセンブラでは表 2.8-1 の命令を同等の命令列に置き換える機能を追加したものが用意されています。MB90550A/B シリーズのコンパイラおよびアセンブラは, 以下のものを使用してください。

- コンパイラ
 - cc907 の V02L06 以降のバージョンおよび fcc907s の V30L02 以降のバージョン
- アセンブラ
 - asm907a の V03L04 以降のバージョンおよび fasm907s の V30L04 (Rev.3000004) 以降のバージョン

第3章

割込み

割込みの機能と動作について説明します。

- 3.1 割込みの概要
- 3.2 割込み要因
- 3.3 割込みベクタ
- 3.4 ハードウェア割込み
- 3.5 ソフトウェア割込み
- 3.6 拡張インテリジェント I/O サービス (EI²OS)
- 3.7 未定義命令の実行による例外発生

3.1 割込みの概要

F²MC-16LX はイベントなどの発生により、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割込み機能があります。

■ 割込みの概要

割込み機能は以下の 4 つに分けることができます。

- ハードウェア割込み：内蔵リソースのイベント発生による割込み処理
- ソフトウェア割込み：ソフトウェアのイベント発生命令による割込み処理
- 拡張インテリジェント I/O サービス (EI²OS)：内蔵リソースのイベント発生による転送処理
- 例外：動作例外事項の発生による中断処理

3.2 割込み要因

表 3.2-1 に、割込み要因と割込みベクタ・割込み制御レジスタを示します。

■ 割込み要因

表 3.2-1 割込み要因と割込みベクタ・割込み制御レジスタ (1 / 2)

割込み要因	EI ² OS クリア	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	-	-
A/D コンバータ		# 11	FFFFD0 _H	ICR00	0000B0 _H
タイムベースタイマ	×	# 12	FFFFCC _H		
DTP0 (外割り 0)		# 13	FFFFC8 _H	ICR01	0000B1 _H
DTP4/5 (外割り 4/5)		# 14	FFFFC4 _H		
DTP1 (外割り 1)		# 15	FFFFC0 _H	ICR02	0000B2 _H
8/16bit PPG0 カウンタ・ポロー	×	# 16	FFFFBC _H		
DTP2 (外割り 2)		# 17	FFFFB8 _H	ICR03	0000B3 _H
8/16bit PPG1 カウンタ・ポロー	×	# 18	FFFFB4 _H		
DTP3 (外割り 3)		# 19	FFFFB0 _H	ICR04	0000B4 _H
8/16bit PPG2 カウンタ・ポロー	×	# 20	FFFFAC _H		
拡張 I/O シリアル 0		# 21	FFFFA8 _H	ICR05	0000B5 _H
8/16bit PPG3 カウンタ・ポロー	×	# 22	FFFFA4 _H		
拡張 I/O シリアル 1		# 23	FFFFA0 _H	ICR06	0000B6 _H
16 bit フリーランタイム (入出力タイム) オーバーフロー		# 24	FFFF9C _H		
16bit リロードタイマ 0		# 25	FFFF98 _H	ICR07	0000B7 _H
DTP6/7 (外割り 6/7)		# 26	FFFF94 _H		
16bit リロードタイマ 1		# 27	FFFF90 _H	ICR08	0000B8 _H
8/16bit PPG4/5 カウンタ・ポロー	×	# 28	FFFF8C _H		

第3章 割り込み

表 3.2-1 割り込み要因と割り込みベクタ・割り込み制御レジスタ (2 / 2)

割り込み要因	EI ² OS クリア	割り込みベクタ		割り込み制御レジスタ	
		番号	アドレス	番号	アドレス
インプットキャプチャ (CH.0) 取込み (入力タイマ)		# 29	FFFF88 _H	ICR09	0000B9 _H
インプットキャプチャ (CH.1) 取込み (入力タイマ)		# 30	FFFF84 _H		
インプットキャプチャ (CH.2) 取込み (入力タイマ)		# 31	FFFF80 _H	ICR10	0000BA _H
インプットキャプチャ (CH.3) 取込み (入力タイマ)		# 32	FFFF7C _H		
アウトプットコンペア (CH.0) 一致 (出力タイマ)		# 33	FFFF78 _H	ICR11	0000BB _H
アウトプットコンペア (CH.1) 一致 (出力タイマ)		# 34	FFFF74 _H		
アウトプットコンペア (CH.2) 一致 (出力タイマ)		# 35	FFFF70 _H	ICR12	0000BC _H
アウトプットコンペア (CH.3) 一致 (出力タイマ)		# 36	FFFF6C _H		
UART 送信完了		# 37	FFFF68 _H	ICR13	0000BD _H
I ² C インタフェース 0	×	# 38	FFFF64 _H		
UART 受信完了		# 39	FFFF60 _H	ICR14	0000BE _H
I ² C インタフェース 1	×	# 40	FFFF5C _H		
フラッシュメモリステータス	×	# 41	FFFF58 _H	ICR15	0000BF _H
遅延割り込み	×	# 42	FFFF54 _H		

・EI²OS 割り込みクリア信号にて割り込み要求フラグはクリアされます。ストップ要求あり。

・EI²OS 割り込みクリア信号にて割り込み要求フラグはクリアされます。

× :EI²OS 割り込みクリア信号にて割り込み要求フラグはクリアされません。

■ 拡張 I/O シリアル 2 で EI²OS 機能を使用した場合の注意事項

同一割込み番号に 2 つの割込み要因があるリソースは、両方の割込み要求フラグが EI²OS 割込みクリア信号でクリアされます。したがって、2 つの要因の中で、どちらか 1 つに EI²OS 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にて対処してください（表 3.2-2 を参照）。

表 3.2-2 拡張 I/O シリアル 2 で EI²OS 機能を使用した場合

割込み要因	割込み番号	割込み制御レジスタ	リソース割込み要求
拡張 I/O シリアル 1	# 23	ICR06	許可
16 ビット・フリーランタイム（入出力タイム）オーバーフロー	# 24		禁止

3.3 割込みベクタ

表 3.3-1 に、割込みベクター一覧表を示します。

■ 割込みベクタ

表 3.3-1 割込みベクター一覧表 (1 / 2)

ソフトウェア 割込み命令	ベクタ アドレス L	ベクタ アドレス M	ベクタ アドレス H	モード レジスタ	割込み No	ハードウェア割込み
INT 0	FFFFFC _H	FFFFFD _H	FFFFFE _H	未使用	#0	なし
:	:	:	:	:	:	:
INT 7	FFFFE0 _H	FFFFE1 _H	FFFFE2 _H	未使用	#7	なし
INT 8	FFFFDC _H	FFFFDD _H	FFFFDE _H	FFFFDF _H	#8	(RESET ベクタ)
INT 9	FFFFD8 _H	FFFFD9 _H	FFFFDA _H	未使用	#9	なし
INT 10	FFFFD4 _H	FFFFD5 _H	FFFFD6 _H	未使用	#10	< 例外 >
INT 11	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	未使用	#11	A/D
INT 12	FFFFCC _H	FFFFCD _H	FFFFCE _H	未使用	#12	タイムベースタイマ
INT 13	FFFFC8 _H	FFFFC9 _H	FFFFCA _H	未使用	#13	DTP0
INT 14	FFFFC4 _H	FFFFC5 _H	FFFFC6 _H	未使用	#14	DTP4/DTP5
INT 15	FFFFC0 _H	FFFFC1 _H	FFFFC2 _H	未使用	#15	DTP1
INT 16	FFFFBC _H	FFFFBD _H	FFFFBE _H	未使用	#16	PPG0 ボロー
INT 17	FFFFB8 _H	FFFFB9 _H	FFFFBA _H	未使用	#17	DTP
INT 18	FFFFB4 _H	FFFFB5 _H	FFFFB6 _H	未使用	#18	PPG1 ボロー
INT 19	FFFFB0 _H	FFFFB1 _H	FFFFB2 _H	未使用	#19	DTP3
INT 20	FFFFAC _H	FFFFAD _H	FFFFAE _H	未使用	#20	PPG2 ボロー
INT 21	FFFFA8 _H	FFFFA9 _H	FFFFAA _H	未使用	#21	拡張 I/O シリアル 0
INT 22	FFFFA4 _H	FFFFA5 _H	FFFFA6 _H	未使用	#22	PPG3 ボロー
INT 23	FFFFA0 _H	FFFFA1 _H	FFFFA2 _H	未使用	#23	拡張 I/O シリアル 1
INT 24	FFFF9C _H	FFFF9D _H	FFFF9E _H	未使用	#24	16bit フリーランタイマ
INT 25	FFFF98 _H	FFFF99 _H	FFFF9A _H	未使用	#25	16bit リロードタイマ 0
INT 26	FFFF94 _H	FFFF95 _H	FFFF96 _H	未使用	#26	DTP6/DTP7
INT 27	FFFF90 _H	FFFF91 _H	FFFF92 _H	未使用	#27	16bit リロードタイマ 1
INT 28	FFFF8C _H	FFFF8D _H	FFFF8E _H	未使用	#28	PPG4/PPG5 ボロー

表 3.3-1 割り込みベクター一覧表 (2 / 2)

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 レジスタ	割り込み No	ハードウェア割り込み
INT 29	FFFF88 _H	FFFF89 _H	FFFF8A _H	未使用	#29	インプットキャプチャ #0
INT 30	FFFF84 _H	FFFF85 _H	FFFF86 _H	未使用	#30	インプットキャプチャ #1
INT 31	FFFF80 _H	FFFF81 _H	FFFF82 _H	未使用	#31	インプットキャプチャ #2
INT 32	FFFF7C _H	FFFF7D _H	FFFF7E _H	未使用	#32	インプットキャプチャ #3
INT 33	FFFF78 _H	FFFF79 _H	FFFF7A _H	未使用	#33	アウトプットコンペア #0
INT 34	FFFF74 _H	FFFF75 _H	FFFF76 _H	未使用	#34	アウトプットコンペア #1
INT 35	FFFF70 _H	FFFF71 _H	FFFF72 _H	未使用	#35	アウトプットコンペア #2
INT 36	FFFF6C _H	FFFF6D _H	FFFF6E _H	未使用	#36	アウトプットコンペア #3
INT 37	FFFF68 _H	FFFF69 _H	FFFF6A _H	未使用	#37	UART 送信完了
INT 38	FFFF64 _H	FFFF65 _H	FFFF66 _H	未使用	#38	I ² C0
INT 39	FFFF60 _H	FFFF61 _H	FFFF62 _H	未使用	#39	UART 受信完了
INT 40	FFFF5C _H	FFFF5D _H	FFFF5E _H	未使用	#40	I ² C1
INT 41	FFFF58 _H	FFFF59 _H	FFFF5A _H	未使用	#41	フラッシュメモリ ステータス
INT 42	FFFF54 _H	FFFF55 _H	FFFF56 _H	未使用	#42	遅延割り込み
INT 43	FFFF50 _H	FFFF51 _H	FFFF52 _H	未使用	#43	なし
:	:	:	:	:	:	:
INT 254	FFFC04 _H	FFFC05 _H	FFFC06 _H	未使用	#254	なし
INT 255	FFFC00 _H	FFFC01 _H	FFFC02 _H	未使用	#255	なし

3.4 ハードウェア割込み

ハードウェア割込みは、内蔵リソース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へ制御を移行する機能です。

■ ハードウェア割込みの概要

ハードウェア割込みの起動は、割込み要求がもつ割込みレベルと、CPU の PS がもつインタラプトレベルマスクレジスタ (ILM) の比較および PS 内の I フラグの内容をハードウェアで参照したのち、発生条件に合致すれば発生します。

ハードウェア割込みが発生するときに CPU が行う処理は以下のものがあります。

- CPU 内部の PC, PS, A, PCB, DTB, ADB, DPR レジスタのシステムスタックへの退避
- PS レジスタ内の ILM の設定。自動的に現在要求している割込みレベルと同じになる
- 対応する割込みベクタの内容の取り込みとそこへの分岐

■ ハードウェア割込みの構造

ハードウェア割込みに関連する機構は、以下の 3 つの部分に分かれて存在します。

● 内蔵リソース

割込み許可ビット、割込み要求ビット：リソースからの割込み要求の制御

● 割込みコントローラ

ICR: 割込みのレベル付け、同時要求割込みの優先度判定

● CPU

I, ILM: 要求割込みレベルと現レベルの比較、割込み許可状態の識別

マイクロコード：割込み処理用ステップ

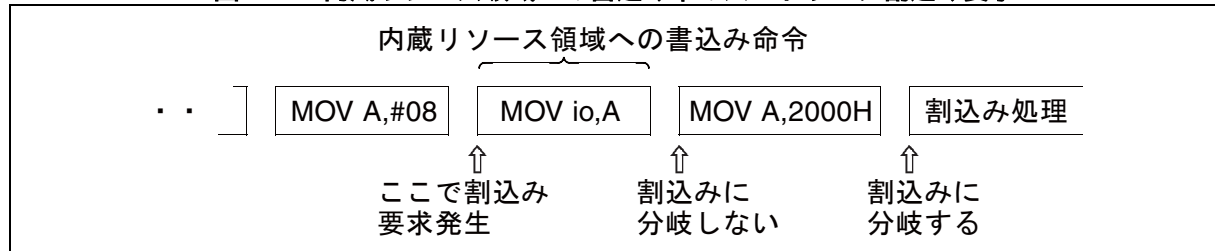
各機構は、内蔵リソースではリソースの制御レジスタに、割込みコントローラでは ICR に、CPU では CCR の内容などとして現れます。ハードウェア割込みを使用する場合には、あらかじめソフトウェアでこの 3カ所の設定を行う必要があります。ICR については、「3.6.1 割込み制御レジスタ (ICR)」を参照してください。

割込み処理の際に参照する割込みベクタのテーブルはメモリ領域の $\text{FFFC00}_H \sim \text{FFFFFF}_H$ に割当ててあり、ソフトウェア割込みと共通で使用しています。

■ 内蔵リソース領域への書き込み中のハードウェア割り込み要求

内蔵リソース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。内蔵リソース領域とは、000000_H ~ 0000FF_H の I/O アドレッシング領域のことではなく、内蔵リソースの制御レジスタ / データレジスタに割当てられている領域のことを示します。

図 3.4-1 内蔵リソース領域への書き込み中のハードウェア割り込み要求



■ 割り込み抑止命令

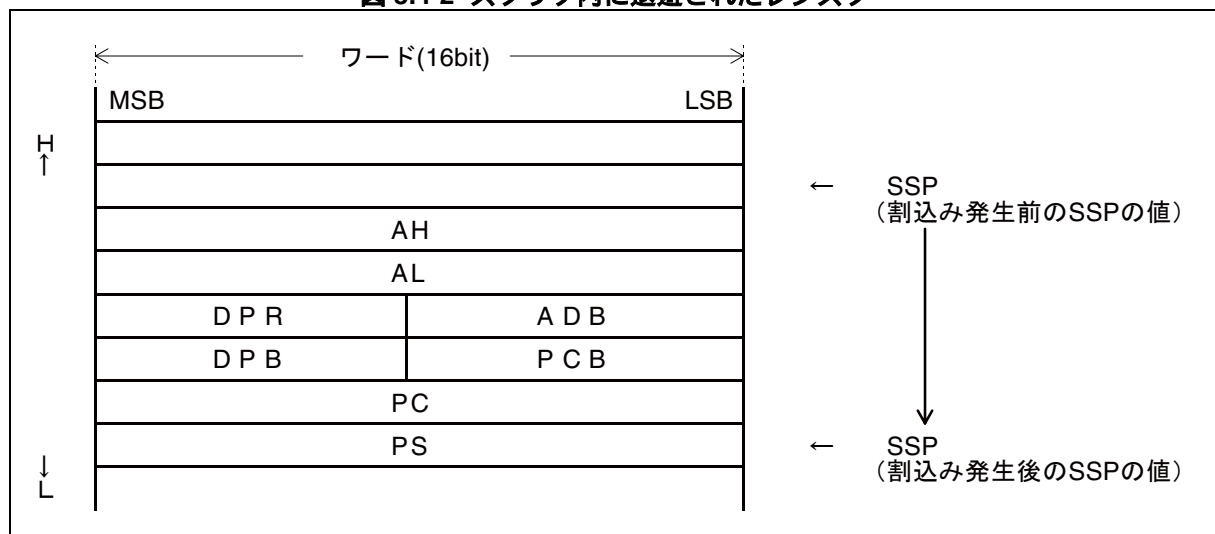
「2.7 割り込み抑止命令とプリフィックスコード」を参照してください。

■ 多重割り込み

F²MC-16LX CPU は多重割り込みをサポートしています。そのため、割り込み処理の実行中にその割り込みのもつ割り込みレベルより強いレベルの割り込みが発生すると、現在実行中の命令を終了したあとでそちらに制御が移ります。強いレベルでの割り込みが終了すると、もとの割り込み処理に戻ります。割り込み処理実行中に、その割り込みと同等かより弱いレベルの割り込み処理が発生した場合は、ILM の内容や I フラグの命令による変更を施さない限り現在の割り込み処理の終了まで、新しい割り込み要求は保留されます。なお、拡張インテリジェント I/O サービスは多重に起動されることはなく、1 つの拡張インテリジェント I/O サービスの処理中は、ほかの割り込み要求および拡張インテリジェント I/O サービス要求はすべて保留されます。

■ 割り込み時におけるスタック内へのレジスタの回避

図 3.4-2 スタック内に退避されたレジスタ



■ ハードウェア割込みの使用上の注意

ハードウェア割込みが存在する場合、誤動作を避けるには、対応する割込みルーチンから復帰する前に割込み要求フラグをクリアする必要があります。

特定のレジスタを読み出す場合、リソースの割込み要求フラグの一部は自動的にクリアされます。この場合は、割込みルーチンから復帰する前に、それらのレジスタを読み出して割込み要求フラグをクリアします。

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の機能を備えた内部リソースには、割込み要求フラグと割込み許可フラグが存在します。割込み要求フラグは、割込み要求の有無を指示します。割込み許可フラグは、該当する内部リソースによる CPU への割込み要求の有無を指示します。割込み要求フラグがセットされるのは、内部リソースに特有のイベントが発生する場合です。割込み許可フラグが「許可」を指示すると、リソースは割込みコントローラへ割込み要求を発生します。

■ ハードウェア割込みの動作

割込みコントローラは、同時に受け取った各割込み要求に対する ICR 内の割込みレベル (IL) 同士を比較し、もっと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、割込み番号が少ないものを優先します。各割込み要求と各 ICR の関係はハードウェアで決まっています。

CPU は受け取った割込みレベル (IL) と PS レジスタ内の ILM とを比較し、割込みレベル (IL) < (ILM) であり、PS レジスタ内の I ビットが "1" に設定されている場合に、現在実行中の命令が終了したあとで、割込み処理用マイクロコードを起動します。割込み処理マイクロコードの先頭で割込みコントローラの ICR 中の ISE ビットを参照し、ISE ビットが "0" (すなわち割込み) であることを確認して割込み処理の本体を起動します。

割込み処理の本体では SSB と SSP の示すメモリ中へ PS と PC, PCB, DTB, ADB, DPR, A の 12 バイトを退避した後、割込みベクタの中から、3byte を取り出して PC と PCB へロードします。PS 中の ILM を、受け付けた割込み要求のレベル値に更新、S フラグを "1" に設定して分岐処理を行います。その結果、次に実行する命令はユーザの定義した割込み処理プログラムになります。

図 3.4-3 ハードウェア割込みの発生と解除まで

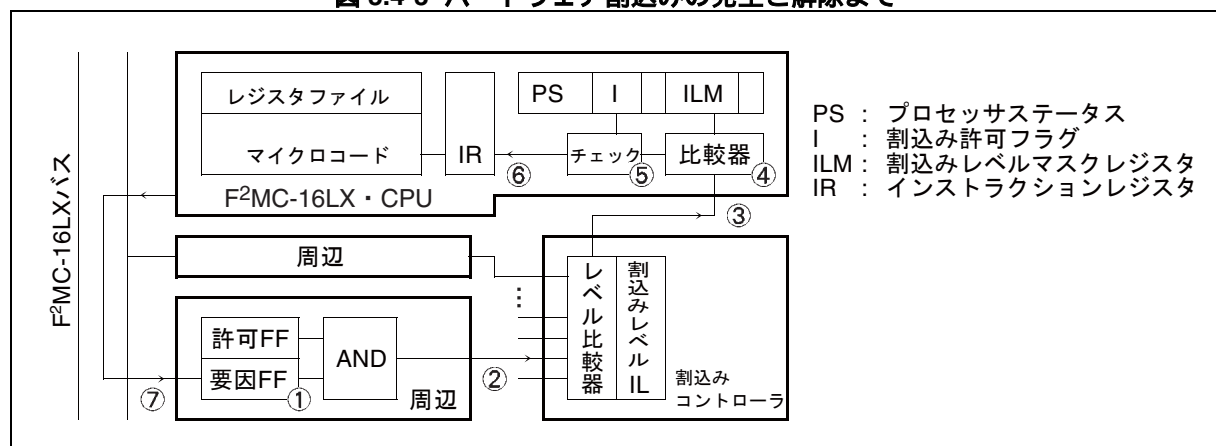


図 3.4-3 の図中における、～ の内容を以下に説明します。

周辺の内部で割り込み要因が発生します。

周辺内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が発生します。

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上で該当する割り込みに対応する割り込みレベルを CPU へ転送します。

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をプロセッサステータスレジスタ内の ILM ビットと比較します。

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のみ、同じプロセッサステータスレジスタ内の I フラグの内容をチェックします。

のチェックの結果 I フラグが割り込み許可状態である場合のみ、ILM ビットの内容を要求されたレベルに設定し、現在実行中の命令の実行が終了し次第割り込み処理を行い、制御を割り込み処理ルーチンへ移します。

ユーザの割り込み処理ルーチン内のソフトウェアで で発生した割り込み要因をクリアすることで割り込み要求が終了します。

および で、CPU が行う割り込み処理の実行時間を以下に示します。割り込みシーケンスへ移行するまでの時間は、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。

■ ハードウェア割り込み処理時間

割り込み要求が発生してから割り込みが受け付けられて、割り込み処理ルーチンが実行されるまでには、割り込み要求サンプル待ち時間と割り込みハンドリング時間が必要です。

● 割り込み要求サンプル待ち時間

割り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を指します。割り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、割り込み要求をサンプリングして判断します。そのため、待ち時間が発生します。

割り込み要求サンプル待ち時間は、最も実行サイクルの長い POPW, PW0 ~ PW7 命令(45 マシンサイクル) 開始直後に割り込み要求が発生した場合最大となります。

● 割り込みハンドリング時間 (割り込み処理準備に要する時間)

割り込み起動 : $24+6 \times \text{表 3.4-1}$ マシンサイクル

割り込み復帰 : $15+6 \times \text{表 3.4-1}$ マシンサイクル (RETI 命令)

表 3.4-1 割り込み処理時のサイクル数の補正值

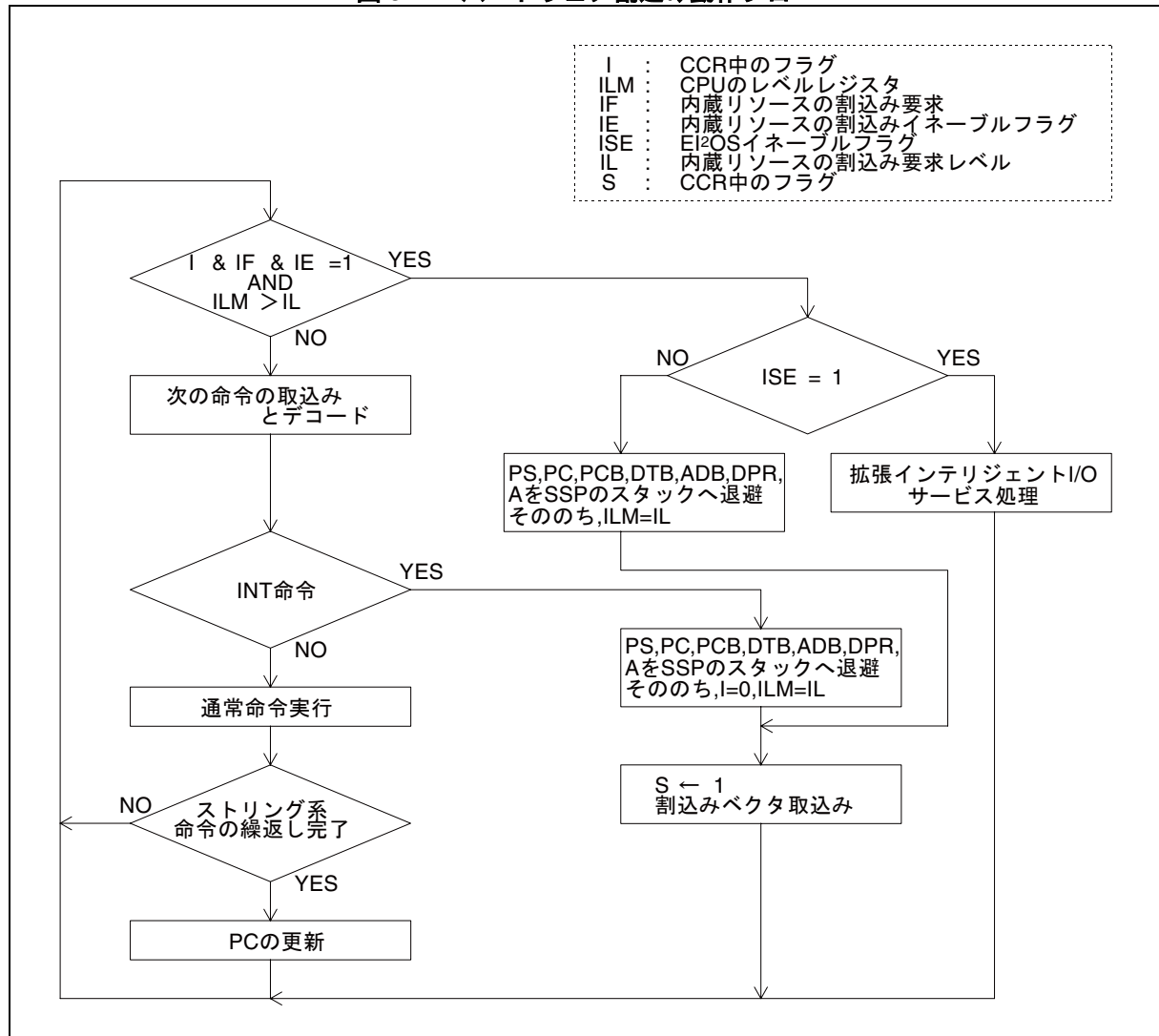
スタックポインタが示しているアドレス	サイクル数の補正值
外部領域 8 ビットデータバスの場合	+4
外部領域 偶数アドレスの場合	+1
外部領域 奇数アドレスの場合	+4
内部領域 偶数アドレスの場合	0
内部領域 奇数アドレスの場合	+2

3.4.2 ハードウェア割り込みの動作フロー

図 3.4-4 に、ハードウェア割り込みの動作フローを示します。

■ ハードウェア割り込みの動作フロー

図 3.4-4 ハードウェア割り込み動作フロー



3.4.3 ハードウェア割り込み使用手順例

図 3.4-5 に、ハードウェア割り込み使用手順例を示します。

■ ハードウェア割り込み使用手順例

図 3.4-5 ハードウェア割り込み使用手順例

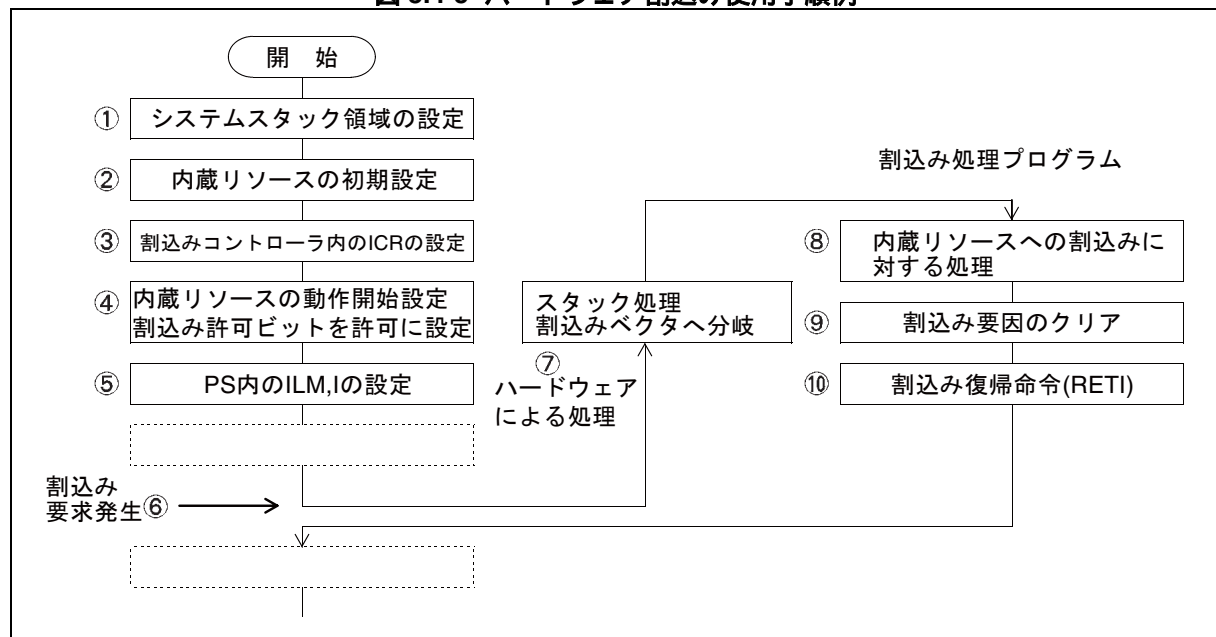


図 3.4-5 の図中における、～ の使用手順を以下に説明します。

システムスタック領域を設定する。

割り込み要求を発生可能な内蔵リソースの初期設定を行う。

割り込みコントローラ内の ICR を設定する。

内蔵リソースを動作開始状態にし、割り込み許可ビットを許可に設定する。

CPU 内の ILM, I フラグを割り込み受け付け可能に設定する。

内蔵リソースの割り込み発生で、ハードウェア割り込みが要求が発生する。

割り込み処理ハードウェアにより、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する。

割り込み処理プログラムにて、割り込み発生に対する内蔵リソースへの処理を行う。

内蔵リソース回路の割り込み要求を解除する。

割り込み復帰命令を実行し、分岐前のプログラムに復帰する。

3.5 ソフトウェア割込み

ソフトウェア割込みは、専用の命令の実行に対応して、CPU がそれまで実行していたプログラムの実行から、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の概要

ソフトウェア割込みの起動は、ソフトウェア割込み命令の実行で常に発生します。ソフトウェア割込みが発生するときに CPU が行う処理は以下のものがあります。

- CPU 内部の PC, PS, AH, AL, PCB, DTB, ADB, DPR レジスタのシステムスタックへ退避させます。
- PS レジスタ内に I を設定します。割込みは、自動的に禁止されます。
- 対応する割込みベクタの値を取り出し、次に値が指示する処理へ分岐します。

INT 命令から出されたソフトウェア割込み要求には、割込み要求フラグや許可フラグは存在しません。ソフトウェア割込み要求が出されるのは、必ず INT 命令の実行によります。

INT 命令には割込みレベルが存在しません。したがって、INT 命令は、ILM を更新しません。INT 命令は、I フラグをクリアして後続の割込み要求を保留にします。

■ ソフトウェア割込みの構造

ソフトウェア割込みは、すべて CPU の内部で取り扱われます。

● CPU

マイクロコード：割込み処理手順

ソフトウェア割込みを使用する場合には、対応する命令を実行する必要があります。

表 3.3-1 に示すように、ソフトウェア割込みは、ハードウェア割込みと同じ割込みベクタ領域を共用します。たとえば、割込み要求番号の INT 13 は、ハードウェア割込みの外部割込み #0 に使用されると同時に、ソフトウェア割込みの INT #13 に使用されます。したがって、外部割込み #0 と INT #13 は、同じ割込み処理ルーチンを呼び出します。

■ ソフトウェア割込みの動作

CPU がソフトウェア割込み命令取り込みを実行すると、ソフトウェア割込み処理用マイクロコードを起動します。ソフトウェア割込み処理用マイクロコードは SSB と SSP が指示するメモリ領域に 12 バイト (PS と PC, PCB, DTB, ADB, DPR, A) を退避させます。次に、マイクロコードは割込みベクタの中から、3byte を取り出して PC と PCB へ格納します。I フラグをリセット ("0") して、S フラグをセット ("1") します。その結果、ユーザアプリケーションプログラムが定義した割込み処理プログラムを次に実行します。

図 3.5-1 ソフトウェア割込みの発生と解除まで

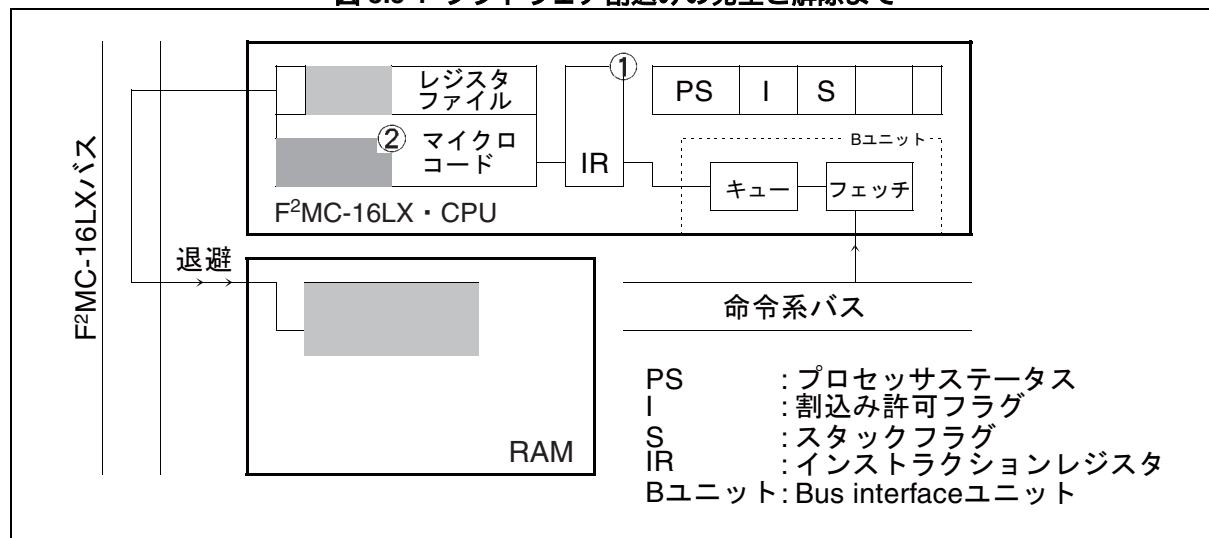


図 3.5-1 の図中における、～ のソフトウェア割込みの発生と解除までの流れを以下に示します。

ソフトウェア割込み命令を実行します

ソフトウェア割込み命令に対応したマイクロコードにしたがってレジスタファイル中の CPU 内専用レジスタが退避されます。

ユーザの割込み処理ルーチン内の RETI 命令で割込み処理が終了します。

■ ソフトウェア割込みに関する注意事項

プログラムバンクレジスタ (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複します。ソフトウェアを設計する際は、必ず CALLV 命令が INT #vct8 命令と同じアドレスを使用しないように注意してください。

3.6 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 (EI²OS) とは、ハードウェア割り込み動作の一種で、I/O とメモリとの間でデータを自動的に転送します。従来は、割り込み処理プログラムにより、I/O とメモリの間でデータを転送していましたが、EI²OS では、DMA のようなデータの転送が可能になります。

■ 拡張インテリジェント I/O サービス (EI²OS) の概要

拡張インテリジェント I/O サービス (EI²OS) は、従来の割り込み処理で行っていた方式と比べると以下に示すような利点があります。

- 転送プログラムの作成が不要となるため、プログラム全体のサイズを小さくすることができます。
- 転送の際に内部レジスタを使用しないので、レジスタの退避が不要になり転送速度が速くなります。
- I/O により転送を停止することができ、不要なデータを転送しません。
- バッファアドレスのインクリメントまたは未更新が選択できます。
- I/O レジスタアドレスのインクリメントまたは未更新が選択できます (バッファアドレスを更新する場合)。

EI²OS が終了時には、終了条件を設定した後に、CPU が自動的に割り込み処理ルーチンへ分岐します。したがって、ユーザは終了条件の種類を判別することができます。

EI²OS を実現するために2箇所の部分にハードウェアが分散して存在し、それぞれのブロックの中に以下に示すようなレジスタやディスクリプタがあります。

● 割り込み制御レジスタ

割り込みコントローラ内にあり ISD のアドレスを示します。

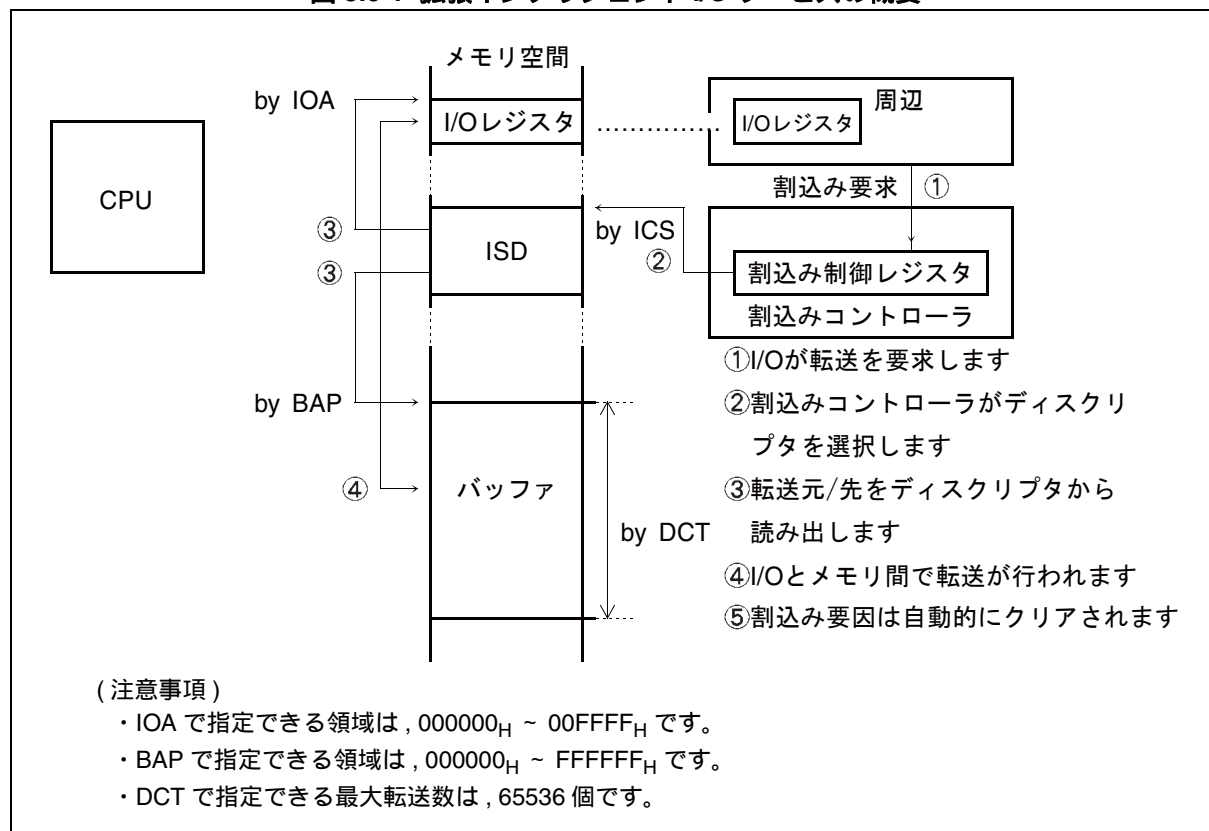
● 拡張インテリジェント I/O サービスディスクリプタ

RAM 上にあり転送モード、I/O のアドレスや転送数、バッファアドレスを保持します。

< 注意事項 >

REALOS を使用する場合は、拡張インテリジェント I/O サービス (EI²OS) が使用できません。

図 3.6-1 拡張インテリジェント I/O サービスの概要



■ 拡張インテリジェント I/O サービス (EI²OS) の構造

EI²OS に関連する機構は、以下の 4 つの部分に分かれて存在します。

● 内蔵リソース

割り込み許可ビット, 割り込み要求ビット: リソースからの割り込み要求の制御

● 割り込みコントローラ

ICR: 割り込みのレベル付け, 同時要求割り込みの優先度判定, EI²OS 動作の選択

● CPU

I, ILM: 要求割り込みレベルと現レベルの比較, 割り込み許可状態の識別

マイクロコード: EI²OS 処理用ステップ

● RAM

ディスクリプタ: EI²OS の転送情報を記述する

3.6.1 割り込み制御レジスタ (ICR)

割り込み制御レジスタ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 I/O に対応して存在します。このレジスタには以下に示す 3 つの機能があります。

- 対応する周辺の割り込みレベルの設定
- 対応する周辺の割り込みを通常割り込みにするか、拡張インテリジェント I/O サービスにするかの選択
- 拡張インテリジェント I/O サービスのチャンネルの選択

このレジスタに対するリードモディファイライト系の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。

■ 割り込み制御レジスタ (ICR)

図 3.6-2 割り込み制御レジスタ (ICR)

割り込み制御レジスタ (ICR)	bit 15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス: B0 _H ~ BF _H	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	書込み時
リード/ライト⇒ 初期値⇒	(W) (0)	(W) (0)	(W) (0)	(W) (0)	(W) (0)	(W) (1)	(W) (1)	(W) (1)	
アドレス: B0 _H ~ BF _H	—	—	S1	S0	ISE	IL2	IL1	IL0	読出し時
リード/ライト⇒ 初期値⇒	(-) (X)	(-) (X)	(R) (0)	(R) (0)	(R) (0)	(R) (1)	(R) (1)	(R) (1)	

< 注意事項 >

ICS3 ~ ICS0 は EI²OS を起動する場合のみ有効となります。EI²OS を起動する場合は ISE を "1" に設定し、起動しない場合は "0" に設定してください。EI²OS を起動しない場合、ICS3 ~ ICS0 は何を設定してもかまいません。

ICS1 と ICS0 は書込みのみ、S1 と S0 は読出しのみ、それぞれ有効です。

[bit15 ~ bit12, bit7 ~ bit4] ICS3 ~ ICS0 (EI²OS チャンネルセレクトビット)

ICS3 ~ ICS0 ビットは、EI²OS チャンネル選択ビットです。

書込み専用のビットで、このビットにより EI²OS のチャンネルを指定します。ここで設定された値によりメモリ上の拡張インテリジェント I/O サービスディスクリプタのアドレスが決定します。ICS はリセットにより "0000_B" に初期化されます。

表 3.6-1 ICS3 ~ ICS0 (EI²OS チャネルセレクトビット)

ICS3	ICS2	ICS1	ICS0	セレクトされるチャネル	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

[bit13, bit12, bit5, bit4] S1, S0

S1, S0 は、EI²OS 終了ステータスビットです。

読出し専用のビットで、EI²OS 終了時にこのビットの値を調べることにより、終了条件が何であったかを判別することができます。リセットにより "00_B" に初期化されます。

表 3.6-2 拡張インテリジェント I/O サービスステータスの終了条件

S1	S0	終了条件
0	0	EI ² OS 動作中あるいは非起動時
0	1	カウント終了による停止状態
1	0	リザーブ
1	1	内蔵リソースからの要求による停止状態

[bit11, bit3] ISE

ISE ビットは、EI²OS を許可します。

割込み要求の発生時にこのビットが"1"であるとEI²OSが起動され、"0"のときは割込みシーケンスが起動されません。また、EI²OSが終了（カウント終了による場合と内蔵リソースからの要求による場合）したとき、ISE ビットは"0"になります。対応する内蔵リソースにEI²OSの機能がないときには、ソフトウェアでISEを"0"にしておく必要があります。読み書き可能なビットです。リセットにより"0"に初期化されます。

[bit10 ~ bit8, bit2 ~ bit0] IL2, IL1, IL0

IL2, IL1, IL0 ビットは、割込みレベルを設定します。

対応する内蔵リソースの割込みレベルを指定します。読出しと書込みができます。リセットによりレベル7（割込みなし）に初期化されます。

表 3.6-3 割込みレベル設定ビットのレベル値

IL2	IL1	IL0	割込みレベル値
0	0	0	0（最強割込み）
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6（最弱割込み）
1	1	1	7（割込みなし）

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

拡張インテリジェント I/O サービスディスクリプタは、内部 RAM 中の 000100_H ~ 00017F_H に存在し、以下に示すものから構成されます。

- データ転送の各種コントロールデータ
- ステータスデータ
- バッファアドレスポインタ

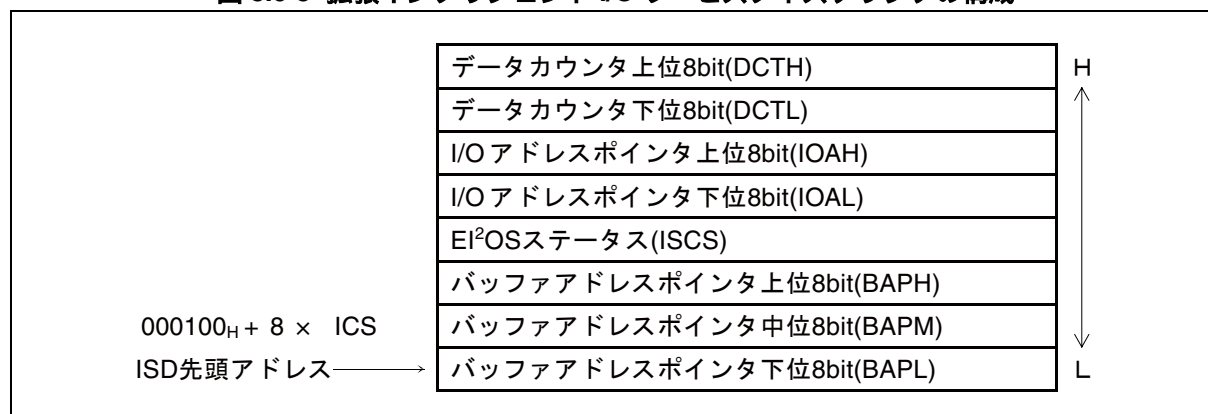
■ 拡張インテリジェント I/O サービスディスクリプタ (ISD)

拡張インテリジェント I/O サービスディスクリプタ (ISD) は、内部 RAM の 000100_H ~ 00017F_H の範囲に存在し、次の要素から構成されます。

- データ転送用コントロールデータ
- ステータスデータ
- バッファアドレスポインタ

図 3.6-3 に、拡張インテリジェント I/O サービスディスクリプタの構成を示します。

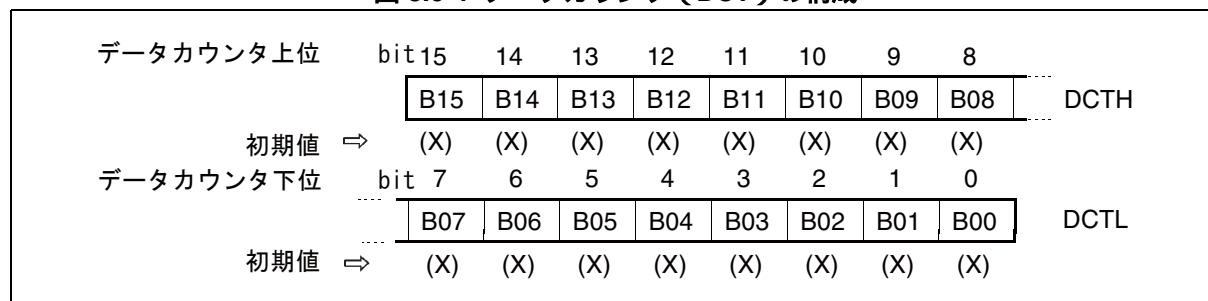
図 3.6-3 拡張インテリジェント I/O サービスディスクリプタの構成



■ 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、転送されたデータ要素の数に対応したカウンタとして機能します。データ転送の前に、このカウンタのデクリメントが1つだけ行われます。このカウンタがゼロになるとEI²OSは終了します。

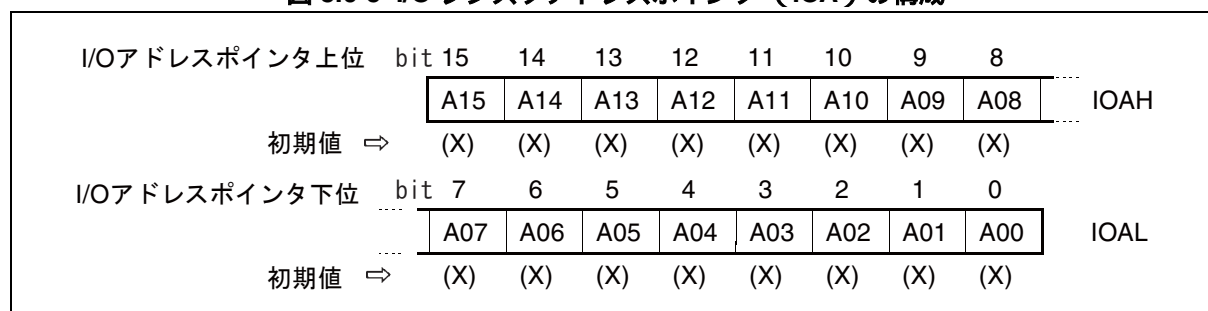
図 3.6-4 データカウンタ (DCT) の構成



■ 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、I/O レジスタアドレスポインタの16bitによりバッファとデータ転送するI/Oレジスタの下位アドレス(A15 ~ A00)を示します。上位アドレス(A23 ~ A16)はすべて"0"であり、000000_Hから00FFFF_H番地までの任意のI/Oを指定できます。

図 3.6-5 I/O レジスタアドレスポインタ (IOA) の構成



■ EI²OS ステータスレジスタ (ISCS)

EI²OS ステータスレジスタ (ISCS) は、8ビット長のレジスタで、更新の向き (インクリメント/デクリメント)、転送データ形式 (バイト/ワード)、バッファアドレスポインタおよびI/Oレジスタアドレスポインタの転送方向を指示します。また、このレジスタは、バッファアドレスポインタまたはI/Oレジスタアドレスポインタが更新または固定されているかどうかを指示します。

図 3.6-6 EI²OS ステータスレジスタ (ISCS) の構成



[bit7 ~ bit5]

予約ビットです。ISCS を設定するときは、必ず "0" に設定してください。

[bit4] IF

IF ビットは、I/O レジスタアドレスポインタの更新 / 固定を指定します。

表 3.6-4 I/O レジスタアドレスポインタの更新 / 固定ビット (IF)

IF	機能
0	データ転送後 I/O レジスタアドレスポインタは更新 (インクリメント) される。
1	データ転送後 I/O レジスタアドレスポインタは固定される。

[bit3] BW

BW ビットは、転送データ長を指定します。

表 3.6-5 転送データ長を指定ビット (BW)

BW	機能
0	バイト
1	ワード

[bit2] BF

BF ビットは、バッファアドレスポインタの更新 / 固定を指定します。

表 3.6-6 バッファアドレスポインタの更新 / 固定指定ビット (BF)

BF	機能
0	データ転送後バッファアドレスポインタは更新 (インクリメント) される。
1	データ転送後バッファアドレスポインタは固定される。

< 注意事項 >

更新する場合、バッファアドレスポインタは下位 16bit のみ変化します。

[bit1] DIR

DIR ビットは、データの転送方向を指定します。

表 3.6-7 データの転送方向指定ビット (DIR) の設定

DIR	設定
0	I/O アドレスポインタ バッファアドレスポインタ
1	バッファアドレスポインタ I/O アドレスポインタ

[bit0] SE

SE ビットは、内蔵リソースからの要求での拡張インテリジェント I/O サービスの終了を制御します。

表 3.6-8 EI²OS 終了制御ビット

SE	設定
0	内蔵リソースからの要求により終了しない。
1	内蔵リソースからの要求により終了する。

■ バッファアドレスポインタ (BAP)

24bit のレジスタで次に EI²OS による転送で使用するアドレスを保持します。BAP は EI²OS の各チャネルに対してそれぞれ独立に存在しますので、EI²OS の各チャネルは 16Mbyte の任意の空間と転送することができます。

< 注意事項 >

ISCS の BF ビットにより更新ありにした場合、BAP は下位 16 ビットのみ変化し、BAPH は変化しません。

3.6.3 拡張インテリジェント I/O サービス (EI²OS) の動作

図 3.6-7 に拡張インテリジェント I/O サービス (EI²OS) の動作フローを、図 3.6-8 に、拡張インテリジェント I/O サービス (EI²OS) の使用手順フローを示します。

■ 拡張インテリジェント I/O サービス (EI²OS) の動作フロー

図 3.6-7 拡張インテリジェント I/O サービス (EI²OS) 動作フロー

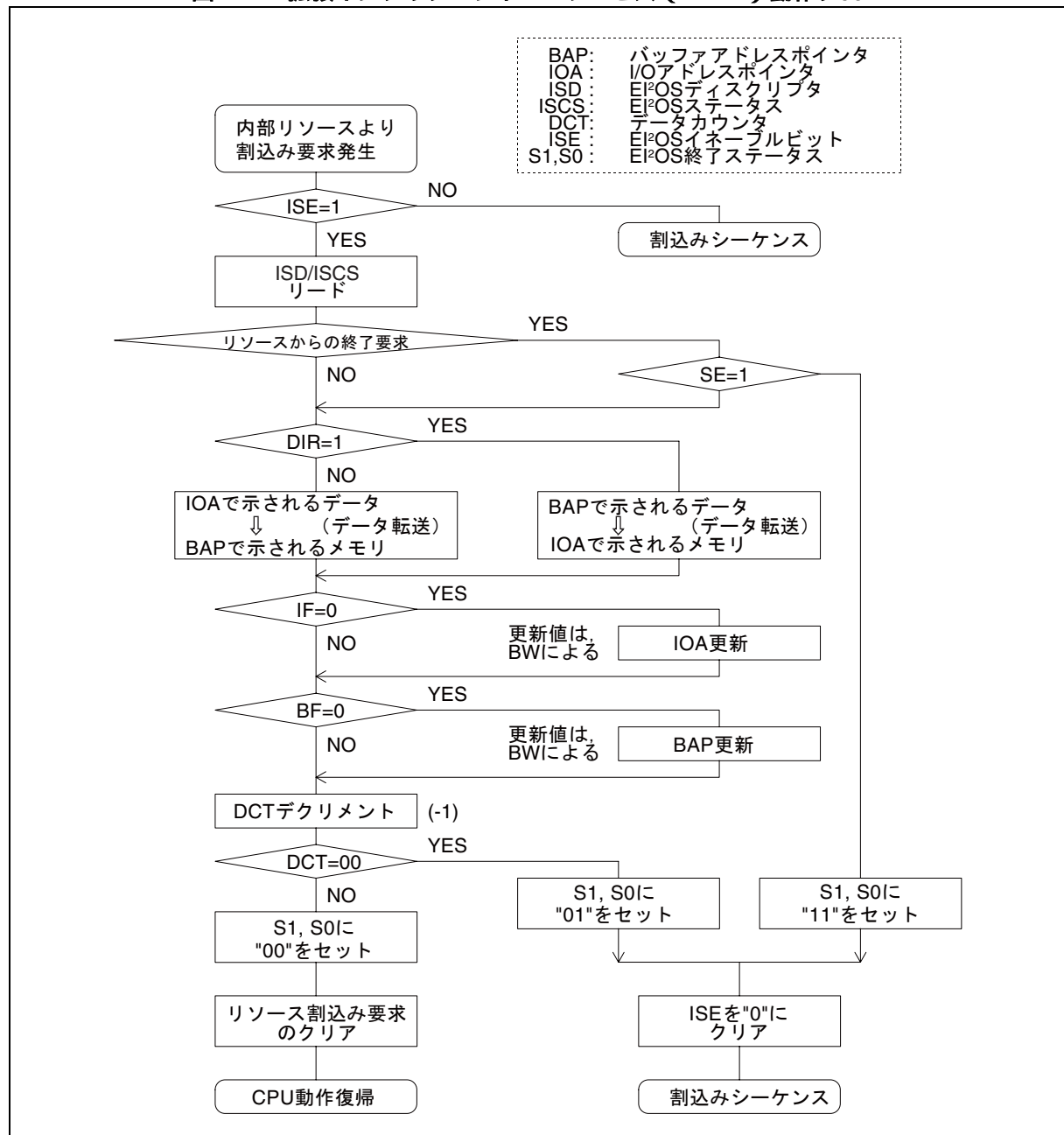
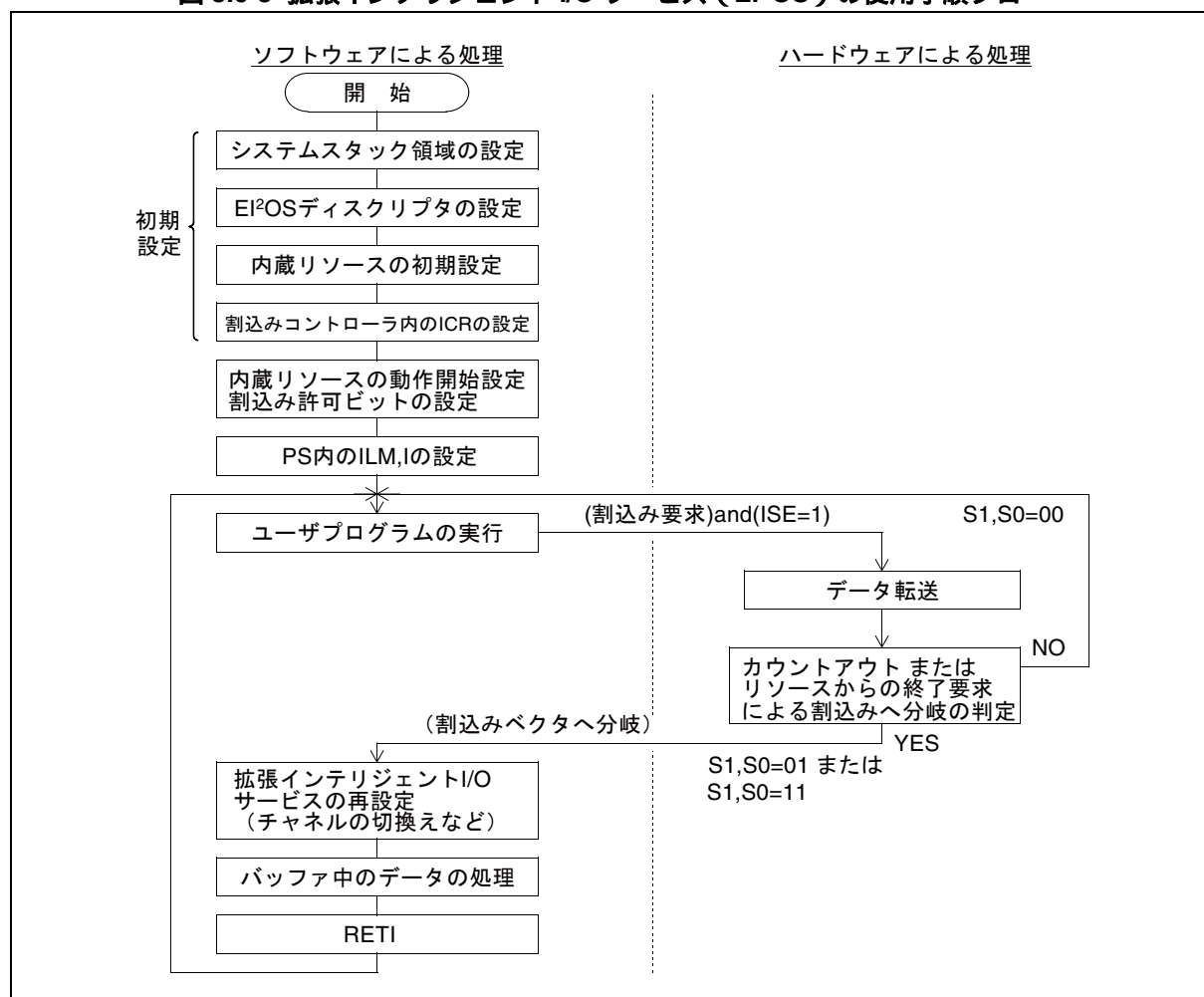


図 3.6-8 拡張インテリジェント I/O サービス (EI²OS) の使用手順フロー

3.6.4 拡張インテリジェント I/O サービス (EI²OS) の実行時間

拡張インテリジェント I/O サービス (EI²OS) の実行時間は次の 3 つの場合があります。

- データ転送継続時 (停止条件が成立しない場合)
- リソースからの停止要求時
- カウント終了時

■ 拡張インテリジェント I/O サービス (EI²OS) の実行時間

- データ転送継続時 (停止条件が成立しない場合)

(表 3.6-9 + 表 3.6-10) マシンサイクル

表 3.6-9 EI²OS の継続時の実行時間

ISCS・SE ビット		"0" に設定		"1" に設定	
I/O アドレスポインタ		固定	更新	固定	更新
バッファアドレスポインタ	固定	32	34	33	35
	更新	34	36	35	37

- リソースからの停止要求時

(36+6 × 表 3.4-1) マシンサイクル

- カウント終了時

(表 3.6-9 + 表 3.6-10 + (21+6 × 表 3.4-1)) マシンサイクル

表 3.6-10 EI²OS の実行時間のデータ転送の補正值

I/O 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

偶 : 偶数アドレス・ワード転送

奇 : 奇数アドレス・ワード転送

8 : 外部バス幅 8 ビット・ワード転送

3.7 未定義命令の実行による例外発生

F²MC-16LX では、未定義命令の実行により例外が発生し、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、命令の境目で例外事項が発生したことを検出した段階で、通常処理から離れて例外処理を行います。一般的に、例外処理は予想外の動作を行った結果で発生するものですので、デバッグ時だけの使用や緊急時の復旧ソフトウェアの起動などのみに使用することをお勧めします。

■ 未定義命令の実行による例外発生

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。すなわち、AL, AH, DPR, DTB, ADB, PCB, PC, PS の内容をシステムスタックに退避した後に、I フラグを "0" に設定し、S フラグを "1" に設定し、割込み番号 10 のベクタで示されるルーチンへ分岐します。スタックへ退避した PC の値は未定義命令を格納しているアドレスそのものです。2 バイト以上の命令コードでは、未定義であることが識別できたコードを格納しているアドレスになります。このため、RETI 命令で復帰させることは可能ですが、再び例外が発生するため、意味がありません。

第4章

クロックとリセット

クロックとリセットの機能と動作について説明します。

- 4.1 クロック発生部
- 4.2 クロック供給マップ
- 4.3 リセット要因
- 4.4 リセット解除後の動作
- 4.5 リセット入力によって初期化されないレジスタ

4.1 クロック発生部

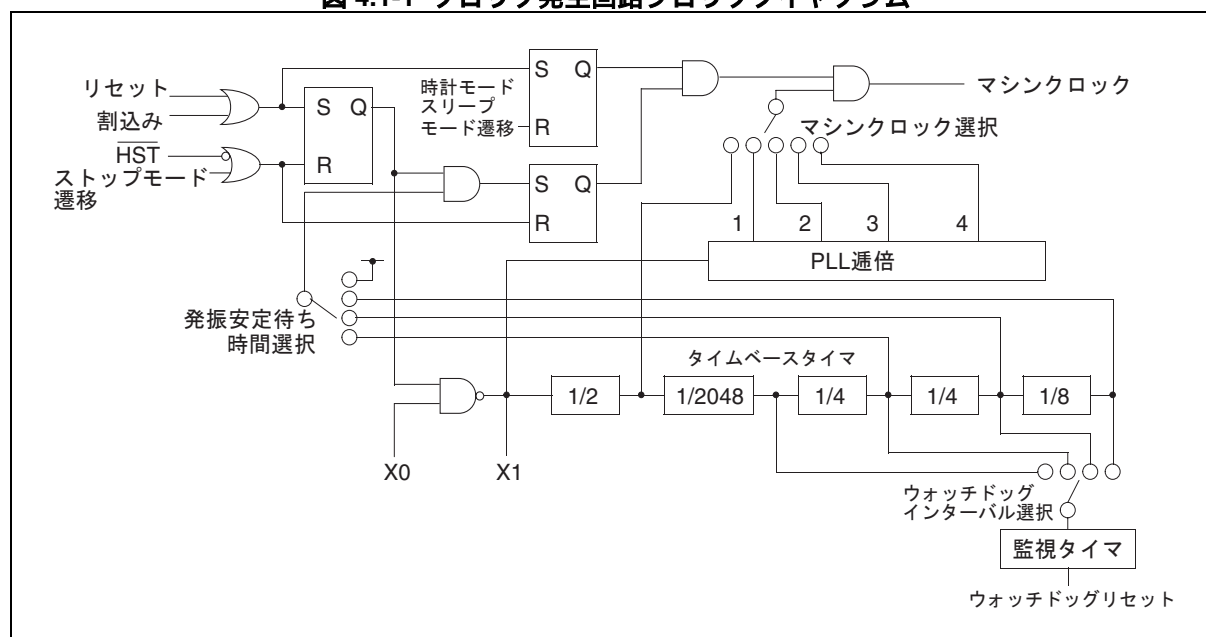
クロック発生部により、スリープ、時計、ストップおよびPLL クロック通倍機能など内部クロックの動作制御を行います。この内部クロックのことをマシナクロックとよび、その1周期をマシンサイクルとしています。また、原発振によるクロックをメインクロックとよび、内部のVCO 発振によるクロックをPLL クロックとしています。

■ クロック発生部に関する注意事項

動作電圧が5V のとき、OSC 原発振は3MHz ~ 16MHz の発振が可能です。CPU および周辺リソース回路は、最大動作周波数が16MHz です。この最大動作周波数を超える通倍率を指定した場合には、正常に動作しません。例えば、原発振が16MHz で発振している場合には、1 通倍のみ指定が可能です。

また、VCO 発振の最低動作周波数は8MHz であり、この周波数以下の発振も指定することもできません。

図 4.1-1 クロック発生回路ブロックダイヤグラム

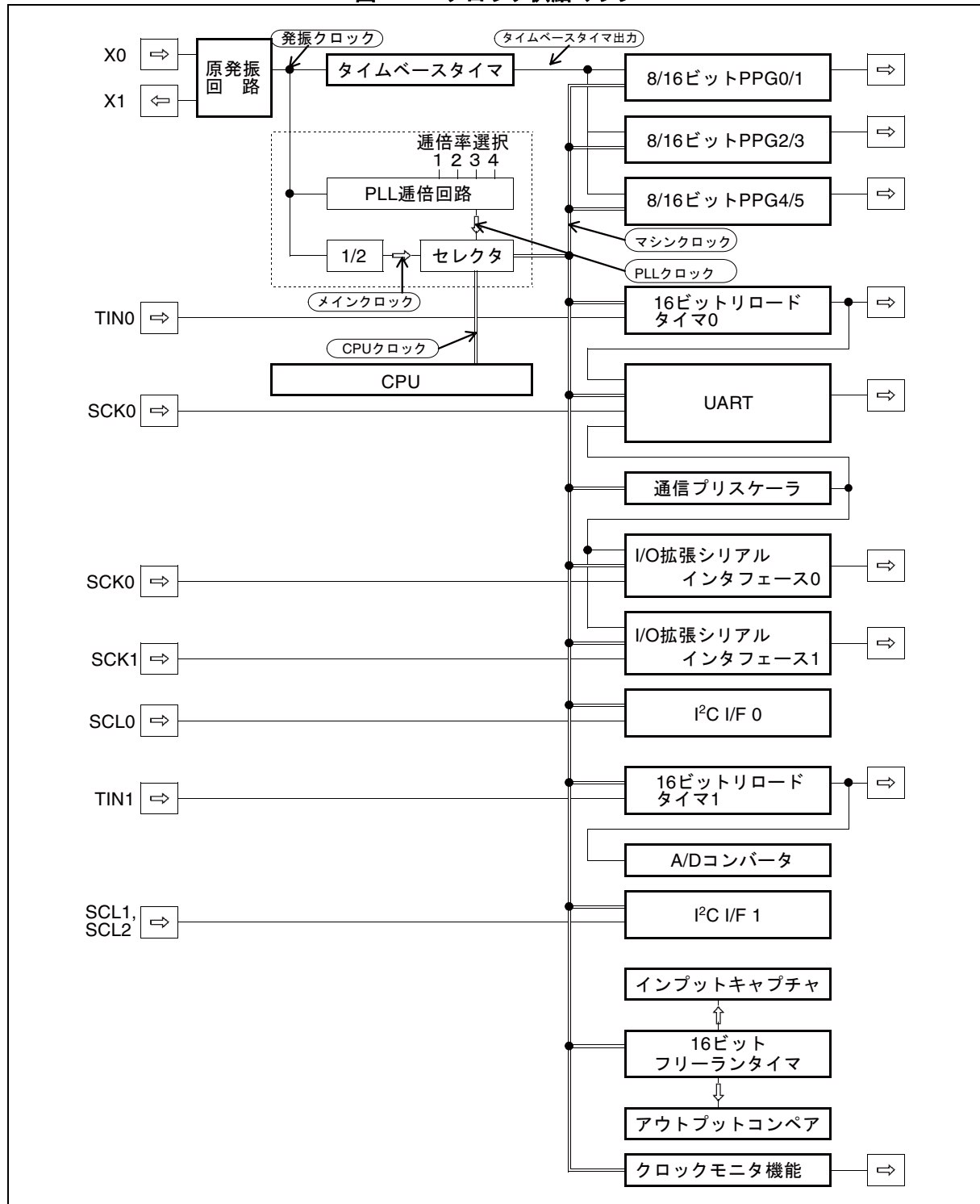


4.2 クロック供給マップ

図 4.2-1 に、クロック供給マップを示します。

■ クロック供給マップ

図 4.2-1 クロック供給マップ



4.3 リセット要因

リセットの要因には、以下に示す5種類があります。

- パワーオンリセットの発生
- ハードウェアスタンバイ状態の解除
- ウォッチドッグタイマのオーバフロー
- $\overline{\text{RST}}$ 端子による外部リセット要求の発生
- ソフトウェアによるリセット要求の発生

■ リセット要因

ストップモードからの解除時およびパワーオンリセット時には、発振安定時間をとってから動作を開始します。リセット要因が発生すると、F²MC-16LX は直ちに現在実行中の処理を中断し、リセット解除待ち状態になります。リセット要因によりマシニングロックとウォッチドッグ機能の初期化状態が異なります。

ウォッチドッグタイマ制御レジスタのリセット要因ビットにより、リセットの要因を判別することが可能です。

< 注意事項 >

- ストップモード以外は外部リセット入力を内部クロックでサンプルしていますので、外部から供給するクロックを停止した状態ではリセット入力は受け付けられません。
- 外部バスを使用している場合、リセット要因が発生するとリセット中にデバイスが発生するアドレスは不定となります。 $\overline{\text{RD}}$, $\overline{\text{WR}}$ などの外部バスアクセス用の信号はすべてインアクティブになります。

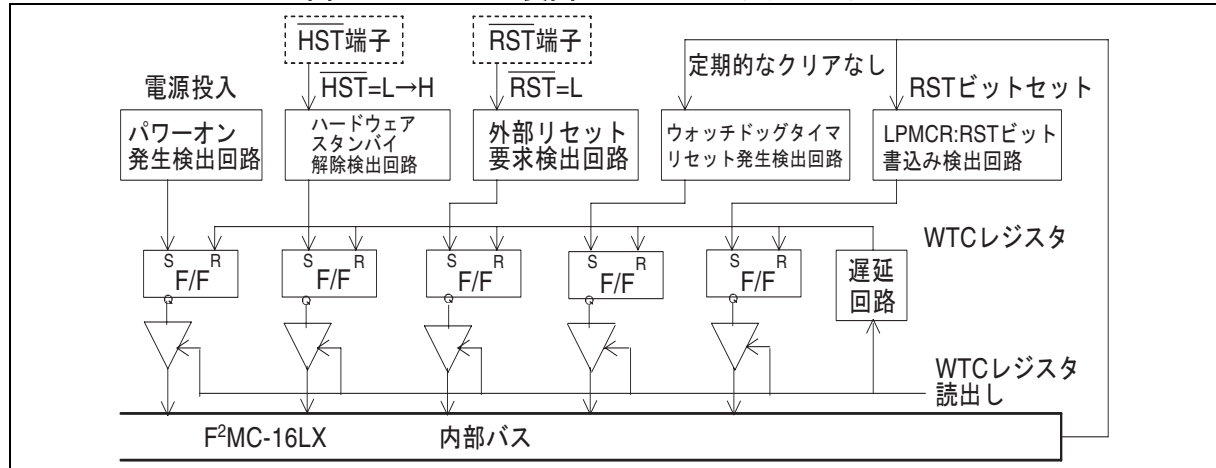
表 4.3-1 リセット要因

リセット	発生要因	マシニングロック	ウォッチドッグタイマ	発振安定待ち
パワーオン	電源立上げ時	メインクロック	停止	あり
ハードウェアスタンバイ	$\overline{\text{HST}}$ 端子への "L" レベル入力	メインクロック	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	停止	あり
外部端子	$\overline{\text{RST}}$ 端子への "L" レベル入力	前の状態を保持	前の状態を保持	なし
ソフトウェア	LPMCR 中の RST ビットに "0" ライト	前の状態を保持	前の状態を保持	なし

- ストップ、ハードウェアスタンバイモード時のリセット入力は、リセット要因に関係なく発振安定待ち時間を取ります。
- パワーオンリセットの発振安定待ち時間は、原発振 2¹⁸ サイクル固定です。それ以外の発振安定待ち時間は、クロック選択レジスタの WS1/WS0 により決定されます。

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があります。これらの内容はウォッチドッグタイマ制御レジスタを読み出すことで得ることができますので、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、ウォッチドッグタイマ制御レジスタを読み出した値をソフトウェアで処理した上で、適切なプログラムへ分岐するようにしてください。

図 4.3-1 リセット要因ビットブロックダイヤグラム



リセット要因が複数発生する場合でも、ウォッチドッグタイマ制御レジスタの対応するそれぞれのリセット要因ビットを立てるようになっています。したがって、外部リセット要求とウォッチドッグリセットが同時に発生した場合でも、ERST ビットとWRST ビットの両方が"1"になります。

ただし、パワーオンリセットの場合だけは別で、PONR ビットが"1"であるときはそれ以外のビットの内容は正常なリセット要因を示しておりません。このため、PONR ビットが"1"の場合はそれ以外のリセット要因ビットの内容は無視するようにソフトウェアを作成してください。

表 4.3-2 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	STBR	WRST	ERST	SRST
パワーオン	1	-	-	-	-
ハードウェアスタンバイ	*	1	*	*	*
ウォッチドッグタイマ	*	*	1	*	*
外部端子	*	*	*	1	*
RST ビット	*	*	*	*	1

※: 前の値を保持します。

< 注意事項 >

リセット要因ビットのクリアはウォッチドッグタイマ制御レジスタの読出しだけです。一度発生したリセット要因に対応するリセット要因ビットは、それ以外のリセット要因が発生しても"1"になったままとなっています。

ウォッチドッグタイマ制御レジスタの構成およびリセット要因ビットについては、「第9章 ウォッチドッグタイマ」を参照してください。

4.4 リセット解除後の動作

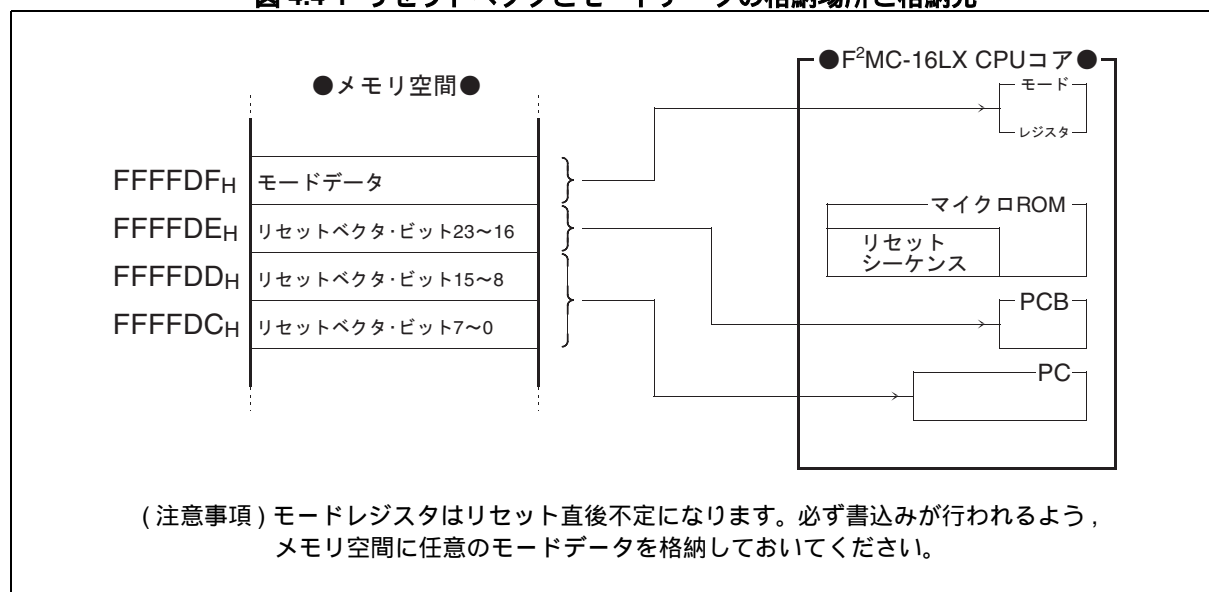
リセット要因が取り除かれると、F²MC-16LX は直ちにリセットベクタを格納してあるアドレスを出力し、リセットベクタとモードデータを取り込みます。リセットベクタとモードデータは、FFFFDC_H ~ FFFFDF_H の4バイトに割り当てられており、それぞれリセット解除後にレジスタにハードウェアで転送されます。

■ リセット解除後の動作

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を指定するのは、モード端子により行います。モード端子で外部ベクタモードを指定すると、内部 ROM ではなく、外部メモリにリセットベクタとモードデータを読み出しにいきますので、シングルチップモードおよび内ROM外バスモードで 사용되는場合には、モード端子で内部ベクタモードを指定することを推奨します。

リセットベクタとモードデータを読み出した後のバスモードは、モードデータにより指定します。

図 4.4-1 リセットベクタとモードデータの格納場所と格納先



4.5 リセット入力によって初期化されないレジスタ

本製品には、パワーオンリセットでしか初期化されないレジスタがあります。
表 4.5-1 に、各リセット要因によって初期化されないレジスタを示します。

■ リセット入力によって初期化されないレジスタ

表 4.5-1 リセット入力によって初期化されないレジスタ

リセットの種類	CKSCR					LPMCR		WDTC
	WS1	WS0	MCS	CS1	CS0	CG1	CG0	WTE
パワーオンリセット								
ハードウェアスタンバイ ($\overline{\text{HST}}$)	×	×		×	×			
ウォッチドッグリセット	×	×		×	×			
外部リセット ($\overline{\text{RST}}$)	×	×	×	×	×	×	×	×
ソフトウェアリセット	×	×	×	×	×	×	×	×

【記号説明】

：初期化する

×：初期化しない（前の状態を保持）

【ビット説明】

- ・ WS1, WS0：メインクロック発振安定時間設定
- ・ MCS：マシクロック設定（"0": PLL クロック, "1": メインクロック）
- ・ CS1, CS0：PLL クロック通倍設定
- ・ CG1, CG0：CPU 間欠動作の設定
- ・ WTE：ウォッチドッグタイマ起動設定

MCS ビットについては、マシクロック設定ビットですので、特に注意が必要です。例えば、電源投入がパワーオンリセット規格を満足しないときは、パワーオンリセットが発生しません。そのため、MCS が初期化されないことにより、内部動作周波数が動作保証範囲外になり、正常動作しなくなることが考えられます。

また、何らかの原因により CPU が暴走し、MCS, CS1, CS0 が書き換わった際にも内部動作周波数が動作保証範囲外になり、その状態から $\overline{\text{RST}}$ 入力のみでは正常復帰しなくなることが考えられます（ただし、内部ウォッチドックが発生した場合には、MCS が初期化され正常動作します）。

上記のようなケースが考えられる場合には、 $\overline{\text{HST}} + \overline{\text{RST}}$ ($\overline{\text{HST}}$ と $\overline{\text{RST}}$ をショート) として使用することを推奨します。

表 4.5-2 に、 $\overline{\text{HST}} + \overline{\text{RST}}$ を行った場合のリセット入力による初期化されないレジスタについて示します。また、表 4.5-2 に示すように、リセット解除時の動作状態が、 $\overline{\text{HST}} + \overline{\text{RST}}$ のリセット入力と、 $\overline{\text{RST}}$ のみの入力とで異なるので、注意が必要です。

表 4.5-2 リセット入力によって初期化されないレジスタ

リセットの種類	CKSCR					LPMCR		WDTC
	WS1	WS0	MCS	CS1	CS0	CG1	CG0	WTE
$\overline{\text{RST}} + \overline{\text{HST}}$ 接続	×	×		×	×			

：初期化する

×：初期化しない（前の状態を保持）

図 4.5-1 リセット入力による動作遷移

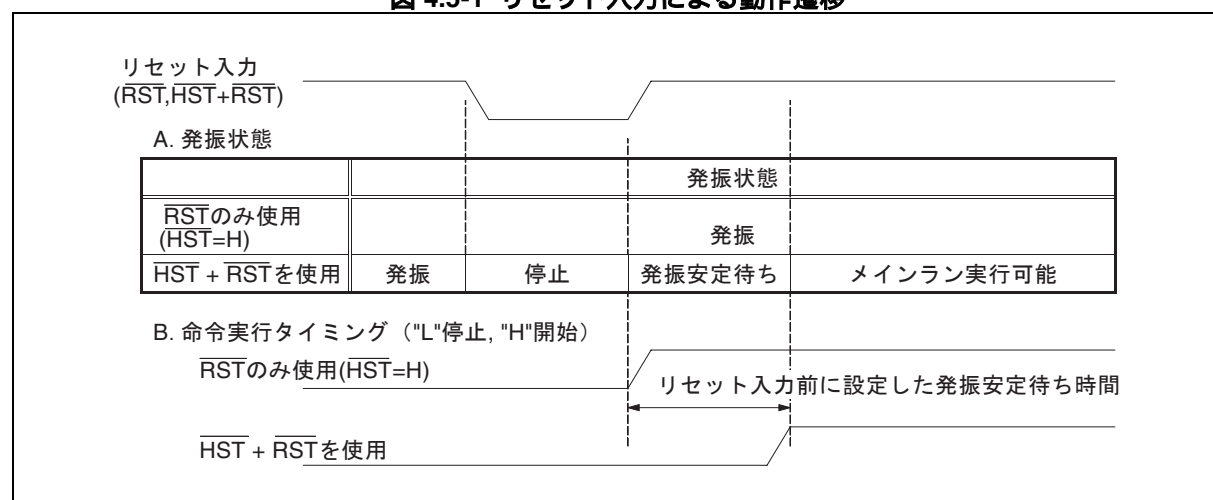
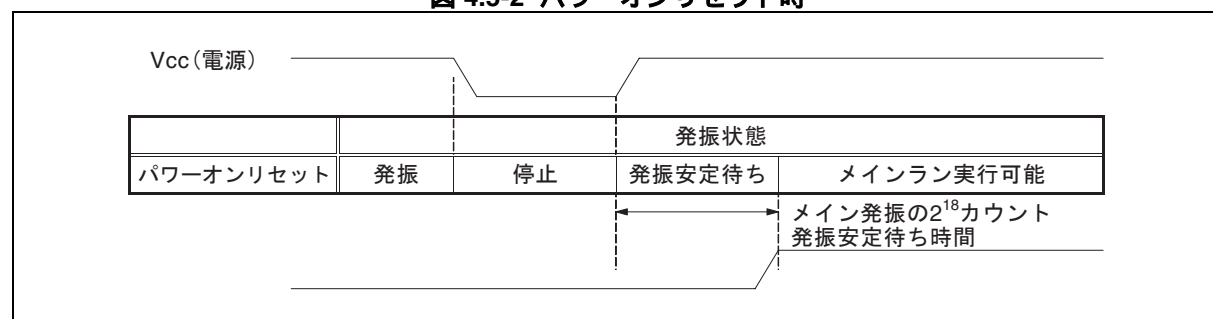


図 4.5-2 パワーオンリセット時



第5章

低消費電力制御回路

低消費電力制御回路（CPU 間欠動作機能，発振安定待ち時間，クロック逡倍機能）の機能と動作について説明します。

- 5.1 低消費電力制御回路の概要
- 5.2 低消費電力モード制御レジスタ（LPMCR）
- 5.3 クロック選択レジスタ（CKSCR）
- 5.4 低消費電力制御回路の動作
- 5.5 CPU 間欠動作機能
- 5.6 発振安定待ち時間設定
- 5.7 マシンクロック

5.1 低消費電力制御回路の概要

動作モードとして、PLL クロックモード、PLL スリープモード、時計モード、メインクロックモード、メインスリープモード、ストップモードおよびハードウェアスタンバイモードがあり、PLL クロックモード以外の動作モードが低消費電力モードに分類しています。

■ 低消費電力回路の概要

- メインクロックモード / メインスリープモード

発振クロック（OSC 発振）のみで動作させるモードで、動作クロックとしてメインクロックを使用し、PLL クロック（VCO 発振）を停止させるモードです。

- PLL スリープモード / メインスリープモード

CPU の動作クロックのみ停止させるモードで、CPU クロック以外は動作しています。

- 時計モード

タイムベースタイマのみを動作させるモードです。

- ストップモード / ハードウェアスタンバイモード

発振を停止させるモードであり、もっとも低消費電力でデータを保持できます。

CPU 間欠動作機能は、レジスタ、内蔵メモリ、内蔵リソースおよび外部バスアクセスを行うときに、CPU に供給するクロックを間欠動作させる機能であり、内蔵リソースに高速クロックを供給したまま CPU 実行速度を下げることで、低消費電力で処理が行えます。

PLL クロックの逡倍率は、クロック選択レジスタの CS1, CS0 ビットにより、1, 2, 3, 4 逡倍の中から選択することができます。

WS1, WS0 ビットにより、ストップモードおよびハードウェアスタンバイモードの解除時のメインクロックの発振安定待ち時間を設定することが出来ます。

< 注意事項 >

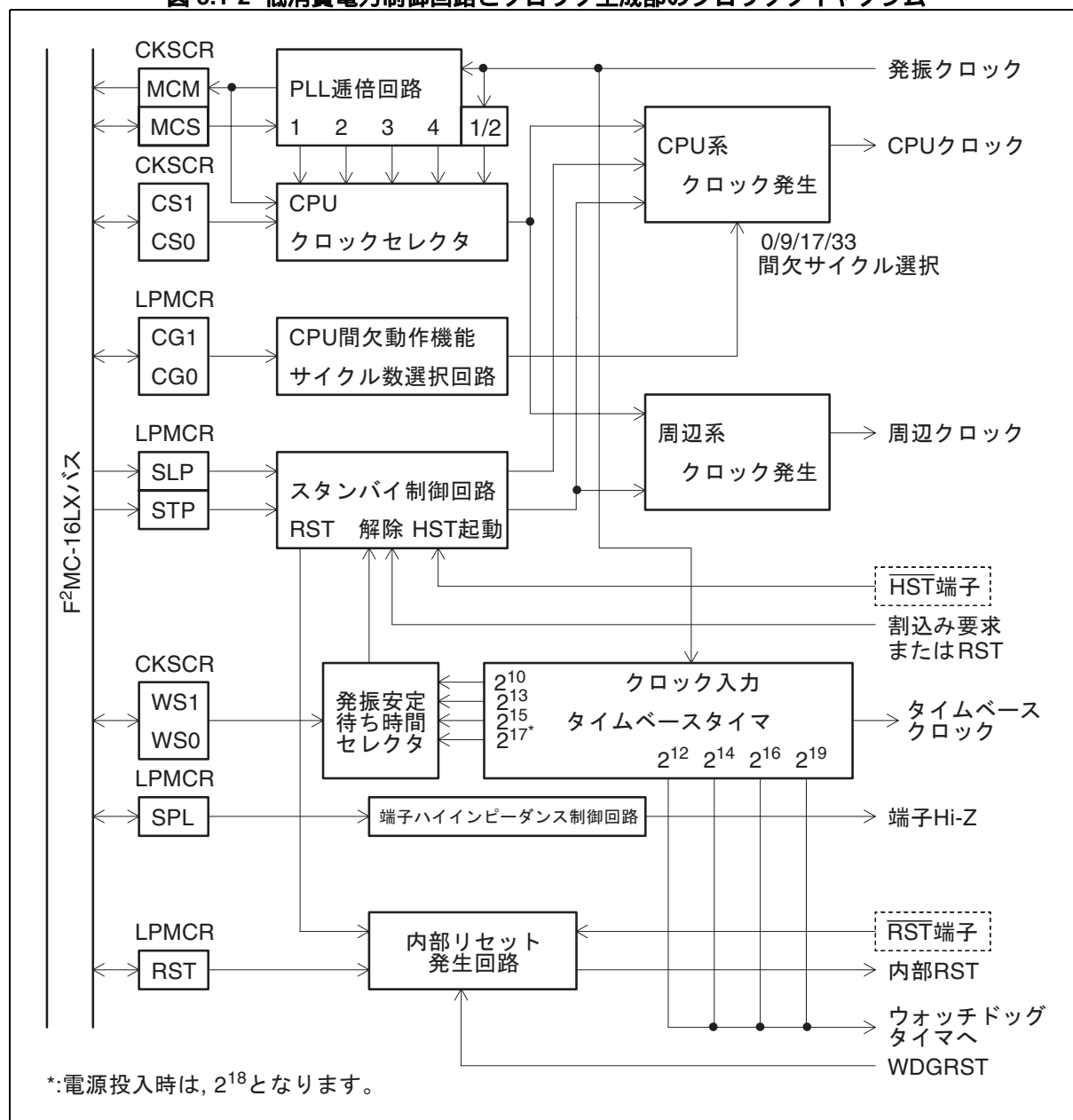
クロックモードを切り換えた場合、切換えが完了するまでは、ほかのクロックモードおよび低消費電力モードへの切換えを行わないでください。切換えの完了はクロック選択レジスタ（CKSCR）の MCM ビットを参照して確認してください。

図 5.1-1 低消費電力回路のレジスタ

低消費電力モード制御レジスタ									
	bit 7	6	5	4	3	2	1	0	
アドレス: 0000A0H	STP	SLP	SPL	RST	予約	CG1	CG0	予約	LPMCR
リード/ライト⇒	(W)	(W)	(R/W)	(W)	(-)	(R/W)	(R/W)	(-)	
初期値⇒	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)	
クロック選択レジスタ									
	bit 15	14	13	12	11	10	9	8	
アドレス: 0000A1H	予約	MCM	WS1	WS0	予約	MCS	CS1	CS0	CKSCR
リード/ライト⇒	(-)	(R)	(R/W)	(R/W)	(-)	(R/W)	(R/W)	(R/W)	
初期値⇒	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	

■ 低消費電力回路のブロックダイアグラム

図 5.1-2 低消費電力制御回路とクロック生成部のブロックダイアグラム



5.2 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、クロック選択レジスタと共に、消費電力の削減に関する各種動作モードを設定します。

■ 低消費電力モード制御レジスタ (LPMCR)

図 5.2-1 低消費電力制御回路のレジスタ

低消費電力モード制御レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス: 0000A0H	STP	SLP	SPL	RST	予約	CG1	CG0	予約
リード/ライト ⇒	(W)	(W)	(R/W)	(W)	(-)	(R/W)	(R/W)	(-)
初期値 ⇒	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)

● 低消費電力モード制御レジスタアクセスに関する注意事項

低消費電力モード制御レジスタへの書込みにより、低消費電力モード(ストップモード、スリープモード)への遷移が行われますが、このときに使用する命令は表 5.2-1 の命令を使用するようにしてください。これらの命令以外の命令で低消費電力モードへ遷移された場合には、誤動作の原因となる場合があります。低消費電力モード制御レジスタの低消費電力モードへの遷移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。

ワード長で低消費電力モード制御レジスタへの書込みを行う場合には、偶数アドレスで書込みが行われるようにしてください。奇数アドレスの書込みで低消費電力モードへ遷移された場合には、誤動作の原因となる場合があります。

表 5.2-1 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	

[bit7] STP

STP ビットは、"1" を書き込むことにより時計モード (CKSCR:MCS=0) またはストップモード (CKSCR:MCS=1) に遷移します。"0" の書込みでは何の操作もしません。リセット、時計解除またはストップ解除で "0" にクリアされます。書込みのみ可能なビットです。読出し値は、常に "0" です。

[bit6] SLP

SLP は, "1" を書き込むことによりスリープモードに遷移します。"0" の書込みでは何の操作もしません。リセット, スリープ解除, またはストップ解除で "0" にクリアされます。STP ビットと SLP ビットに同時に "1" を書き込んだ場合, 時計モードまたはストップモードに遷移します。書込みのみ可能なビットです。読出し値は, 常に "0" です。

[bit5] SPL

SPL は, "0" のとき, 時計モードおよびストップモード時の外部端子のレベルを保持します。"1" のとき, 時計モードおよびストップモード時の外部端子をハインピーダンスにします。リセットで "0" にクリアされます。読み書き可能なビットです。

[bit4] RST

RST ビットは, "0" を書き込むことにより 3 マシンサイクルの内部リセット信号を発生します。"1" の書込みでは何の操作もしません。読出し時は "1" が, 読み出されます。

[bit3] 予約

必ず "1" を設定してください。

[bit2, bit1] CG1, CG0

CG1, CG0 ビットは, CPU 間欠動作機能のクロック一時停止サイクル数を設定します。パワーオン, ハードウェアスタンバイ, ウォッチドッグによるリセットにより "00" に初期化されます。その他のリセット要因によるリセットでは初期化されません。読み書き可能なビットです。

CPU 間欠機能は, レジスタ, 内蔵メモリ, 内蔵リソースおよび外部バスアクセスを行うときに CPU に供給するクロックを一定期間停止させ, 内部バスサイクルの起動を遅らせる機能です。内蔵リソースに高速クロックを供給したまま CPU の実行速度を下げることにより, 低消費電力で処理が行えます。

表 5.2-2 クロック一時停止サイクル数を設定ビット (CG1, CG0) の設定

CG1	CG0	CPU クロック一時停止サイクル数
0	0	0 サイクル (CPU クロック = リソースクロック)
0	1	9 サイクル (CPU クロック : リソースクロック = 1: 約 3 ~ 4)
1	0	17 サイクル (CPU クロック : リソースクロック = 1: 約 5 ~ 6)
1	1	33 サイクル (CPU クロック : リソースクロック = 1: 約 9 ~ 10)

[bit0] 予約

必ず "0" を設定してください。

5.3 クロック選択レジスタ (CKSCR)

クロック選択レジスタ (CKSCR) は, CPU のマシナクロックを設定・制御し, 電源立上げ時や発振復帰時の発振安定待ち時間を設定します。

■ クロック選択レジスタ (CKSCR)

図 5.3-1 クロック選択レジスタ (CKSCR)

クロック選択レジスタ								
bit 15 14 13 12 11 10 9 8								
アドレス: 0000A1H								
	予約	MCM	WS1	WS0	予約	MCS	CS1	CS0
リード/ライト⇒	(-)	(R)	(R/W)	(R/W)	(-)	(R/W)	(R/W)	(R/W)
初期値	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)

[bit15] 予約

必ず "1" を設定してください。

[bit14] MCM

マシナクロックとして, メインクロックまたは PLL クロックのどちらが選択されているかを表示するビットです。"0" のとき, PLL クロックが選択されていることが示され, "1" のとき, メインクロックが選択されていることが示されます。MCS=0 で, かつ MCM=1 ならば, PLL クロック発振安定待ち期間中であることになります。なお, PLL クロックの発振安定待ち期間は, 2^{13} メインクロックサイクル固定です。

[bit13, bit12] WS1, WS0

ストップモード, ハードウェアスタンバイモード解除時のメインクロックの発振安定待ち時間を設定します。パワーオンリセットにより "11" に初期化され, その他のリセット要因によるリセットでは初期化されません。読み書き可能なビットです。

表 5.3-1 クロック選択レジスタ (ビット 13, 12)

WS1	WS0	発振安定待ち時間 (原発振 4MHz 時)
0	0	約 256 μ s (原振 2^{10} カウント)
0	1	約 2.05ms (原振 2^{13} カウント)
1	0	約 8.19ms (原振 2^{15} カウント)
1	1	約 32.77ms (原振 2^{17} カウント)*

*: 電源投入時は, 約 65.54ms (原振 2^{18} カウント) となります。

[bit11] 予約

必ず "1" を設定してください。

[bit10] MCS

マシンのクロックとして、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示するビットです。"0" を書き込むことにより、PLL クロックを選択します。"1" を書き込むことにより、メインクロックを選択します。"1" のときに、"0" を書き込むと、PLL クロックの発振安定待ち期間を発生するために、自動的にタイムベースタイマがクリアされ、さらにタイムベースタイマ制御レジスタの TBOF ビットもクリアされます。なお、PLL クロックの発振安定待ち期間は、 2^{13} メインクロックサイクル固定です。（原発振 4MHz 時、発振待ち時間は約 2ms になります。）

また、メインクロック選択時の動作クロックは、メインクロックを2分周したクロックとなります。（原発振 4MHz 時、動作クロックは 2MHz になります。）

パワーオン、ハードウェアスタンバイ、ウォッチドッグによるリセットにより"1"に初期化されます。

< 注意事項 >

MCS ビットが"1"のときに、"0"を書き込む場合には、TBIE ビットまたは CPU の ILM ビットによりタイムベースタイマ割込みがマスクされている状態で、行うようにしてください。また、MCS ビットに"1"を書き込んでから 8 マシンサイクルの間は、このビットに"0"が書き込めない場合があります。8 マシンサイクル以上待ってから書き込んでください。

[bit9, bit8] CS1, CS0

PLL クロックの逡倍率を選択するビットです。外部端子、RST ビットおよびウォッチドッグによるリセットとハードウェアスタンバイ解除では初期化されません。パワーオンによるリセットにより"00"に初期化されます。

MCS ビットが"0"のときには書込みが抑止されます。一度 MCS ビットを"1"にした（メインクロックモード）後に、CS ビットを書き換えてください。

読み書き可能なビットです。

表 5.3-2 クロック選択レジスタ（ビット 13, 12）

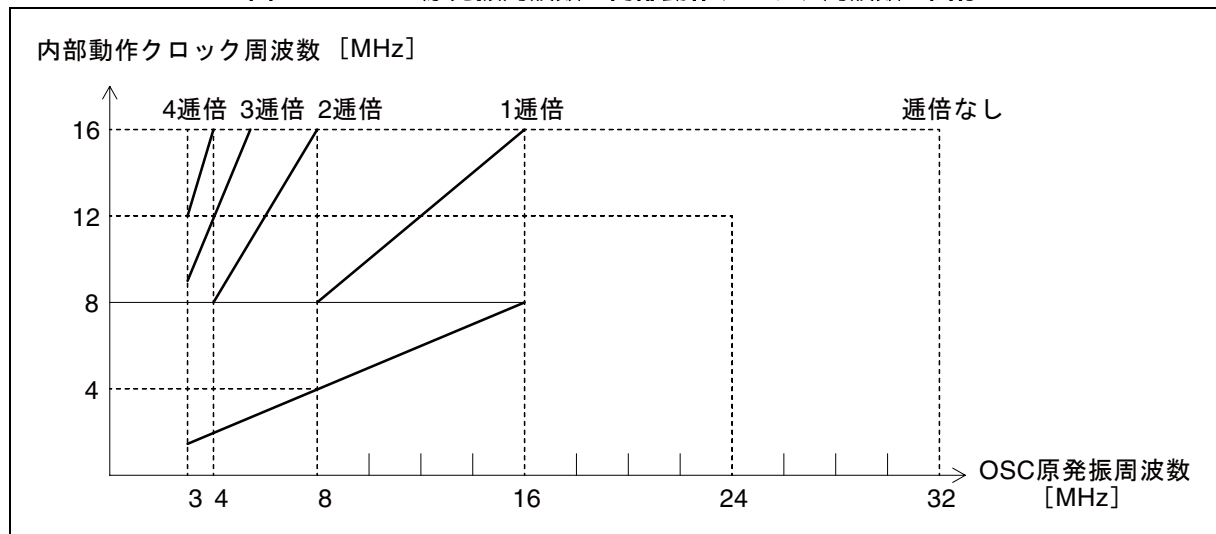
CS1	CS0	逡倍率	内部動作クロック（OSC 原発振×逡倍率）		
			原発振 4MHz 時	原発振 8MHz 時	原発振 16MHz 時
0	0	1 倍	設定禁止	8MHz	16MHz
0	1	2 倍	8MHz	16MHz	設定禁止
1	0	3 倍	12MHz	設定禁止	設定禁止
1	1	4 倍	16MHz	設定禁止	設定禁止

< 注意事項 >

動作電圧が5Vのとき，OSC 原発振は3MHz～16MHzの発振が可能です，CPU および周辺リソース回路は，最大動作周波数が16MHzですので，この最大動作周波数を超える逡倍率を指定した場合には，正常に動作しません。例えば，OSC 原発振が16MHzで発振している場合には，1逡倍のみ指定が可能です。

また，VCO 発振の最低動作周波数は4MHzであり，この周波数以下の発振も指定することもできません。

図 5.3-2 OSC 原発振周波数と内部動作クロック周波数の関係



5.4 低消費電力制御回路の動作

表 5.4-1 に、低消費電力モード動作状態を、図 5.4-1 に、低消費電力モードの状態遷移図を示します。

■ 低消費電力制御回路の動作

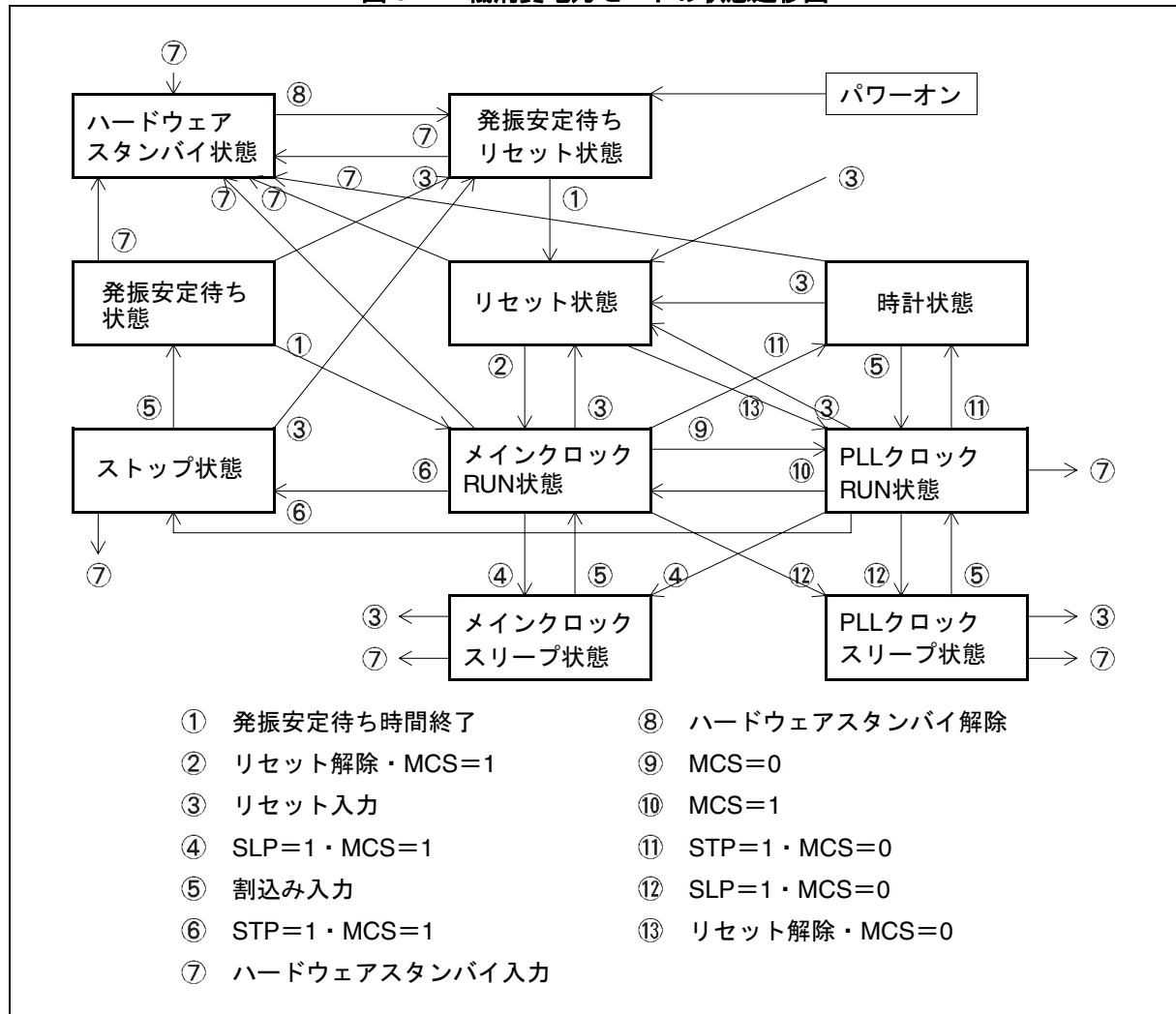
表 5.4-1 低消費電力モード動作状態

	遷移条件	発振 / タイム ベースタイマ	CPU	周辺 (マシン クロック)	端子	解除方法
メインスリープ	MCS=1 SLP=1	動作	停止	動作	動作	リセット 割込み
PLL スリープ	MCS=0 SLP=1	動作	停止	動作	動作	リセット 割込み
時計 (SPL=0)	MCS=0 STP=1	動作	停止	停止	保持	リセット 割込み
時計 (SPL=1)	MCS=0 STP=1	動作	停止	停止	HI-Z	リセット 割込み
ストップ (SPL=0)	MCS=1 STP=1	停止	停止	停止	保持	リセット 割込み
ストップ (SPL=1)	MCS=1 STP=1	停止	停止	停止	HI-Z	リセット 割込み
ハードウェア スタンバイ	$\overline{\text{HST}}=\text{L}$	停止	停止	停止	HI-Z	$\overline{\text{HST}}=\text{H}$

< 注意事項 >

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

図 5.4-1 低消費電力モードの状態遷移図



5.4.1 スリープモード

低消費電力モード制御レジスタ中の SLP ビットに "1", STP ビットに "0" を書き込むことによりスタンバイ制御回路をスリープモードに設定します。スリープモードとは, CPU に供給するクロックのみを停止するモードで, CPU は停止し, 周辺回路は動作を続けます。

■ スリープモードへの遷移

SLP ビットに "1" を書き込んだときに割込み要求が発生していると, スタンバイ制御回路はスリープモードへ遷移しません。そのため, CPU は割込みを受け付けられない状態では次の命令の実行, 受け付ける状態であれば即座に割込み処理ルーチンへの分岐を行います。

スリープモードでは, 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。また, スリープモード中でも, 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。

■ スリープモードの解除

スタンバイ制御回路は, リセット入力または割込みの発生によりスリープモードを解除します。リセット要因によりスリープモードを解除した場合は, スリープモードを解除した上でリセット状態になります。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 より強い割込み要求が発生すると, スタンバイ制御回路はスリープモードを解除します。スリープモードの解除後は, 通常の割込み処理と同じ扱いとなり, I フラグ, ILM と割込み制御レジスタ (ICR) の設定により割込みが受け付けられるとき, CPU は外部割込み等の要因により割込み処理を実行し, 割込みが受け付けられないときはスリープモードに入れた命令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, 通常はスリープモードに入れた命令の次の命令を実行した後に, 割込み処理に遷移します。ただし, スリープモードに遷移するのと, 外部バスホールド要求の受け付けが同時であった場合には, 次の命令を実行する前に, 割込み処理に遷移することがあります。

5.4.2 時計モード

時計モードとは、原発振とタイムベースタイマ以外の動作を止めるモードであり、チップのほぼ全機能が停止します。また、時計モード中の I/O 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を、低消費電力モード制御レジスタ中の SPL ビットにより制御できます。

■ 時計モードへの遷移

クロック選択レジスタの MCS ビットが "0" で、低消費電力モード制御レジスタ中の STP ビットに "1" を書き込むことによりスタンバイ制御回路を時計モードに設定します。STP ビットに "1" を書き込んだときに割込み要求が発生している場合は、スタンバイ制御回路は時計モードに遷移しません。

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。

なお、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、その要求を受け付けません。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 Hi-Z した状態で、 $\overline{\text{HAK}}$ 信号が "L" にならないことがあります。

■ 時計モードの解除

スタンバイ制御回路は、リセット入力または割込みの発生により時計モードを解除します。リセット要因により時計モードを解除した場合は、時計モードを解除した上で、リセット状態になります。時計モードからの復帰の場合、スタンバイ制御回路はまず時計モードを解除し、そののち PLL クロック発振安定待ち状態に遷移します。外部リセットでは、MCS ビットはクリアされませんので、リセット期間が PLL クロック発振安定待ち期間より短い場合には、リセットシーケンスはメインクロックを使用して行われます。なお、この場合の PLL クロック発振安定待ち期間は、タイムベースタイマのクリアを行わないため、タイムベースタイマの状態により $2^{13} \sim 3 \times 2^{13}$ メインクロックサイクルになります。

時計モード中に周辺回路などから割込みレベルが 7 より強い割込み要求が発生すると、スタンバイ制御回路は時計モードを解除します。時計モードの解除後は、通常の割込み処理と同じ扱いとなり、I フラグ、ILM と割込み制御レジスタ (ICR) の設定により割込みが受け付けられるとき、CPU は外部割込み等の要因により割込み処理を実行し、割込みが受け付けられないとき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，通常は時計モードに入れた命令の次の命令を実行した後に，割込み処理に遷移します。ただし，時計モードに遷移すると，外部バスホールド要求の受付けが同時であった場合には，次の命令を実行する前に，割込み処理に遷移することがあります。

時計モード解除時には，PLL クロック発振安定待ち状態に遷移しますので，PLL クロックを使用しない場合には，リセット直後または割込み先の直後の命令で MCS ビットを "1" に変更してください。

外部割込みで時計モードを解除する場合は，外部割込みの割込み要求を "H" レベルに設定してください。"L" レベルの割込み要求設定では，誤動作を起こす可能性があります。

また，エッジの割込み要求では，時計モードの解除はできません。

5.4.3 ストップモード

ストップモードとは、原発振を止めるモードであり、チップの全機能が停止します。したがって、もっとも低消費電力でデータを保持することができます。また、ストップモード中の I/O 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を、LPMCR 中の SPL ビットにより制御できます。

■ ストップモードへの遷移

クロック選択レジスタの MCS ビットが "1" で、低消費電力モード制御レジスタ中の STP ビットに "1" を書き込むことによりスタンバイ制御回路をストップモードに設定します。STP ビットに "1" を書き込んだときに割込み要求が発生している場合は、スタンバイ制御回路はストップモードに遷移しません。

ストップモードでは、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。なお、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 Hi-Z にした状態で、 \overline{HAK} 信号が "L" にならないことがあります。

■ ストップモードの解除

スタンバイ制御回路は、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。リセット要因によりストップモードを解除した場合は、ストップモードを解除した上でリセット状態になります。

ストップモードからの復帰の場合、スタンバイ制御回路はまず発振安定待ち状態に遷移し、その後ストップモードを解除しますので、ストップモードからの解除がリセット要因であったときもリセットシーケンスは発振安定待ち時間経過後実行されます。

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 より強い割込み要求が発生すると、スタンバイ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、CKSCR 中の WS1, WS0 ビットで指定されたメインクロックの発振安定待ち時間を経過したあとで、通常の割込み処理と同じ扱いとなり、I フラグ、ILM と割込み制御レジスタ (ICR) の設定により割込みが受け付けられるとき、CPU は外部割込み等の要因により割込み処理を実行し、割込みが受け付けられないとき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に入れた命令の次の命令を実行した後に、割込み処理に遷移します。ただし、ストップモードに遷移するのと、外部バスホールド要求の受付けが同時であった場合には、次の命令を実行する前に、割込み処理に遷移することがあります。

外部割込みでストップモードを解除する場合は、外部割込みの割込み要求を "H" レベルに設定してください。"L" レベルの割込み要求設定では、誤動作を起こす可能性があります。また、エッジの割込み要求では、ストップモードの解除はできません。

5.4.4 ハードウェアスタンバイモード

ハードウェアスタンバイモードとは、リセットを含むほかの状態に関係なく $\overline{\text{HST}}$ 端子が "L" レベルの間、発振を停止させ I/O 端子を全てハイインピーダンス状態にするモードです。

■ ハードウェアスタンバイモードへの遷移

$\overline{\text{HST}}$ 端子を "L" レベルにすることにより、どんな状態からでもスタンバイ制御回路をハードウェアスタンバイモードに設定することができます。

ハードウェアスタンバイモードでは内部 RAM の内容は保持されますが、アキュムレータなどの専用レジスタは初期状態になります。

■ ハードウェアスタンバイモードの解除

ハードウェアスタンバイモードの解除は、 $\overline{\text{HST}}$ 端子によってのみ解除することができます。

$\overline{\text{HST}}$ 端子が "H" レベルとなるとスタンバイ制御回路はハードウェアスタンバイモードを解除し、内部リセット信号を有効にしたのち発振安定待ち状態に遷移します。さらに発振安定待ち時間の経過後、スタンバイ制御回路は内部リセットを解除し、その結果 CPU はリセットシーケンスから実行を開始します。

5.4.5 スリープ, ストップ, ホールド, リセット, ハードウェアスタンバイ時の端子状態

表 5.4-2 に, シングルチップモード時の各端子状態を, 表 5.4-3 に, 外部データバス 16 ビットモード時の各端子状態を, 表 5.4-4 に, 外部データバス 8 ビットモード時の各端子状態を示します。

■ シングルチップモード時の各端子状態

表 5.4-2 シングルチップモード時の各端子状態

端子名	スリープ時	ストップ時 ^{*6}		ホールド時	リセット時	ハードウェア スタンバイ時 ^{*6}
		SPL=0	SPL=1			
P07 ~ P00, P17 ~ P10, P27 ~ P20, P37 ~ P30, P47 ~ P40, P55 ~ P50, P67 ~ P64, P77 ~ P70, P87 ~ P80, P97 ~ P90, PA4 ~ PA0	直前の状態 を保持 ^{*2}	入力遮断 ^{*4} / 直前の状態 を保持	入力遮断 ^{*4} / 出力 Hi-Z ^{*5}	本状態はない	入力不可 ^{*3} / 出力 Hi-Z ^{*5}	入力遮断 ^{*4} / 出力 Hi-Z ^{*5}
P77 ~ P70		入力可能 ^{*1}			入力可能 ^{*1}	

*1: 「入力可能」とは, 入力機能は使用可能な状態であることを意味するので, Pull Up, Pull Down オプションあるいは外部からの入力が必要です。出力ポート機能として使用している場合にはほかのポートと同じです。

*2: 「直前の状態を保持」とは, 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, あるいは入力であれば入力不可を意味します。出力していた状態をそのまま出力とは, 出力のある内蔵周辺が動作中であれの内蔵周辺に従い出力を行い, ポート等として出力している場合にはその出力を保持するということの意味します。

*3: 「入力不可」とは, 端子からすぐ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。

*4: 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。

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

*6: ストップ時およびハードウェアスタンバイ時は, ポート端子のプルアップオプションは切り離されます。

< 注意事項 >

ストップ時, ハードウェアスタンバイ時は, 外部から入力する信号は, 8 本以内とし, それぞれの周波数は, 1 kHz 以下にしなければなりません。

■ 外部データバス 16 ビットモード時の各端子状態

表 5.4-3 外部データバス 16 ビットモード時の各端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット時	ハードウェアスタンバイ時
		SPL=0	SPL=1			
P07 ~ P00 (AD07 ~ AD00), P17 ~ P10 (AD15 ~ AD08)	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z
P27 ~ P20 (A23 ~ A16)	出力状態 ^{*1}	出力状態 ^{*1}			出力状態 ^{*1}	
P37(CLK)	入力不可 / 出力可能 ^{*2}	入力不可 / 出力状態 ^{*1}		入力不可 / 出力可能 ^{*2}	入力不可 / 出力 Hi-Z	
P36(RDY)	直前の状態 を保持	入力遮断 / 直前の状態 を保持		入力不可 / 出力 Hi-Z		
P35($\overline{\text{HAK}}$)				"L" 出力		
P34(HRQ)				"1" 入力		
P33($\overline{\text{WRH}}$)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z		
P32($\overline{\text{WRL}}$)						
P31($\overline{\text{RD}}$)						
P30(ALE)	"L" 出力	"L" 出力			"L" 出力	
P47 ~ P40, P55 ~ P50, P67 ~ P60, P87 ~ P80, P97 ~ P90, PA4 ~ PA0	直前の状態 を保持	入力遮断 / 直前の状態 を保持		直前の状態を 保持	入力不可 / 出力 Hi-Z	
P77 ~ P70		入力可能			入力可能	

*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にしてありますが、内部回路の動作が停止状態であるため、「H」ないし「L」の固定値を出すことを意味します。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、出力変化があります。

*2: 「出力可能」とは端子駆動用トランジスタを駆動許可状態にしてあり、内部回路の動作が許可状態のため、動作内容が端子に現れることを意味します。

■ 外部データバス 8 ビットモード時の各端子状態

表 5.4-4 外部データバス 8 ビットモード時の各端子状態

端子名	スリープ時	ストップ時		ホールド時	リセット	ハードウェアスタンバイ時		
		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	入力不可 / 出力 Hi-Z	入力不可 / 出力 Hi-Z	入力遮断 / 出力 Hi-Z		
P17 ~ P10 (AD15 ~ AD08), P23 ~ P20 (A23 ~ A16)	出力状態	出力状態			出力状態			
P37(CLK)	入力不可 / 出力可能	入力不可 / 出力状態		入力不可 / 出力可能	入力不可 / 出力 Hi-Z			
P36(RDY)	直前の状態 を保持	入力遮断 / 直前の状態 を保持		入力不可 / 出力 Hi-Z				
P35($\overline{\text{HAK}}$)				"L" 出力				
P34(HRQ)				"I" 入力				
P33				直前の状態 を保持				
P32($\overline{\text{WR}}$)	"H" 出力	"H" 出力		入力不可 / 出力 Hi-Z	"H" 出力			
P31($\overline{\text{RD}}$)					"L" 出力			
P30(ALE)	"L" 出力	"L" 出力		直前の状態 を保持	入力不可 / 出力 Hi-Z			
P27 ~ P24, P47 ~ P40, P55 ~ P50, P67 ~ P60, P87 ~ P80, P97 ~ P90, PA4 ~ PA0	直前の状態 を保持	入力遮断 / 直前の状態 を保持						
P77 ~ P70				入力可能				入力可能

5.5 CPU 間欠動作機能

CPU 間欠動作機能は、レジスタ、内蔵メモリ（ROM, RAM, I/O, リソース）および外部バスアクセスを行うときに、CPU に供給するクロックを一定期間停止させ、内部バスサイクルの起動を遅らせることができます。

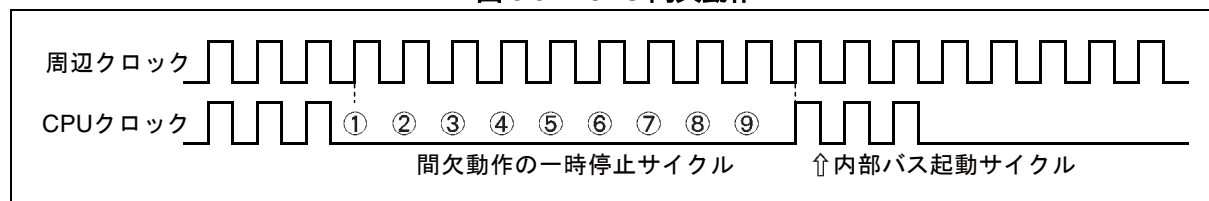
■ CPU 間欠動作機能

CPU 間欠動作機能では、内蔵リソースに高速クロックを供給したまま CPU の実行速度を下げることにより、低消費電力で処理が行えます。CG1, CG0 ビットにより、CPU に供給するクロックの一時停止サイクル数の選択を行います。

なお、外部バス動作そのものは、リソースと同じクロックを使用して行います。

また、CPU 間欠動作機能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、内蔵リソースおよび外部バスアクセスを行う回数に、一時停止サイクル数を掛けた補正値を通常の実行時間に加えることで、求めることができます。

図 5.5-1 CPU 間欠動作



5.6 発振安定待ち時間設定

CKSCR レジスタ中の WS1, WS0 ビットにより, ストップモードおよびハードウェアスタンバイモードの解除時の発振安定待ち時間を選択します。X0, X1 端子に接続する発振回路, 発振素子の種類や特性に応じて発振安定待ち時間を選択してください。

■ 発振安定待ち時間設定

パワーオンリセット以外のリセットは本ビットを初期化しません。パワーオンリセット発生時は本ビットを "11" に初期化します。そのため, パワーオン時の発振安定待ち時間は原発振の約 2^{18} カウントとなります。

PLL クロックの発振安定待ち期間は, 2^{13} メインクロックサイクル固定です (原発振 4MHz 時, 発振待ち時間は約 2ms になります)。

5.7 マシニングロック

マシニングロックの切換えは、クロック選択レジスタ (CKSCR) の MCS ビットの設定によって行ないます。

■ マシニングロックの初期化

MCS ビットは、 $\overline{\text{RST}}$ 端子、RST ビットによるリセットでは、初期化されません。パワーオン、ハードウェアスタンバイ、ウォッチドッグによるリセットにより、“1” に初期化されます。

< 注意事項 >

動作電圧が 5V のとき、OSC 原発振は 3MHz ~ 16MHz の発振が可能です。CPU および周辺リソース回路は、最大動作周波数が 16MHz です。この最大動作周波数を超える逡倍率を指定した場合には、正常に動作しません。例えば、原発振が 16MHz で発振している場合には、1 逡倍のみ指定が可能です。

また、VCO 発振の最低動作周波数は 8MHz であり、この周波数以下の発振も指定することはできません。

■ マシニングロック切換え

CKSCR レジスタ中の MCS ビットへの書込み動作により、メインクロックと PLL クロックを切り換えます。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期間 (2^{13} マシニングロック) 後に、メインクロックから PLL クロックに切り換わります。

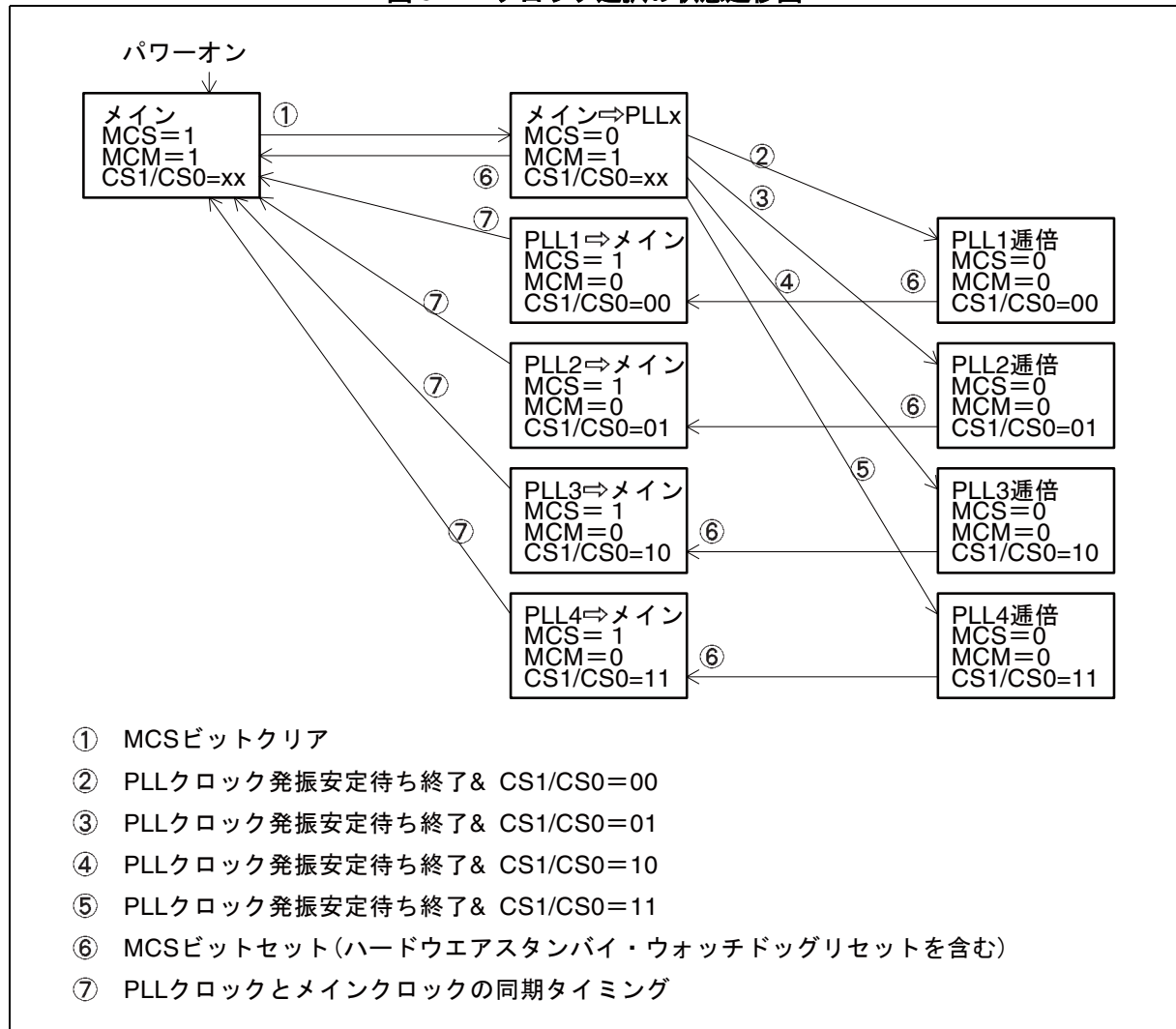
MCS ビットを “0” から “1” に書き換えた場合には、PLL クロックとメインクロックのエッジが一致するタイミング (1 ~ 8PLL クロック後) で PLL クロックからメインクロックに切り換わります。

MCS ビットを書き換えても即座にマシニングロックの切り換わらないため、マシニングロックに依存するリソースを操作する場合には、MCM ビットを参照してマシニングロックの切換えが行われたことを確認したのちに、リソースを操作してください。

< 注意事項 >

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

図 5.7-1 クロック選択の状態遷移図



第6章

メモリアクセスモード

メモリアクセスモードの機能と動作について説明します。

- 6.1 メモリアクセスモードの概要
- 6.2 外部メモリアクセス（外部バス端子制御回路）
- 6.3 外部メモリアクセス制御信号の動作

6.1 メモリアクセスモードの概要

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ます。

■ メモリアクセスモードの概要

表 6.1-1 メモリアクセスモード

動作モード	バスモード	アクセスモード（外部データバス幅）
RUN	シングルチップ	-
	内 ROM 外バス	8 ビット
		16 ビット
	外 ROM 外バス	8 ビット
		16 ビット
フラッシュ書込み	-	-

● 動作モード

動作モードとは、デバイスの動作状態を制御するモードを示すもので、モード設定用端子（MD_x）の内容で指定します。動作モードを選択することで、通常動作の起動や、フラッシュメモリを書き込むことができます。

● バスモード

バスモードとは、内部 ROM の動作と外部アクセス機能の動作を制御するモードを示すもので、モード設定用端子（MD_x）とモードデータ内の M_x ビットの内容で指定します。モード設定用端子（MD_x）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するもので、モードデータ内の M_x ビットは、通常動作時のバスモードを指定するものです。

● アクセスモード

アクセスモードとは、外部データバス幅を制御するモードを示すもので、モード設定用端子（MD_x）とモードデータ内の S0 ビットで指定します。アクセスモードを選択することで、外部データバスを 8 ビット長あるいは 16 ビット長のいずれかを指定します。

6.1.1 モード端子

MD2 ~ MD0 の 3 本の外部端子を組み合わせて設定することで、モードの指定ができます。

■ モード端子

表 6.1-2 に、モード端子と設定モードの関係を示します。

表 6.1-2 モード端子と設定モードの関係

モード端子設定			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
MD2	MD1	MD0				
0	0	0	外部ベクタモード 0	外部	8 ビット	
0	0	1	外部ベクタモード 1	外部	16 ビット	リセットベクタ 16 ビットバス幅アクセス
0	1	0	指定禁止			
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	指定禁止			
1	0	1				
1	1	0	フラッシュシリアル 書込み	-	-	
1	1	1	フラッシュ メモリモード	-	-	パラレルライタ使用時 のモード

(注意事項)

- 外部ベクタモード 0 を選択した場合でも、バス制御選択レジスタの IOBS, LMBS の初期値は "1" にセットされていますので、0000C0_H ~ 0000FF_H, 002000_H ~ 7FFFFFF_H の領域では 16 ビット幅となります。この領域で 8 ビット幅としたい場合はバス制御選択レジスタの IOBS, LMBS に "0" を書き込んでください。外部ベクタモード 1 のときには、HMBS ビットが "0" にセットされ、16 ビットバス幅でアクセスします。
- フラッシュメモリのシリアル書込みは、モード端子の設定だけでは書込みができません。ほかの端子設定も必要です。詳細は、FLASH シリアル書込み接続例を参照してください。

6.1.2 モードデータ

モードデータは、主記憶 FFFFDF_H に置く、CPU の動作制御用のデータです。
リセットシーケンス実行中にこのデータを取り込み、デバイス内部のモードレジスタに格納します。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。
本レジスタによる設定はリセットシーケンス以降に有効となります。
予約ビットは、必ず "0" を設定してください。

■ モードデータ

図 6.1-1 モードデータ構成

モードデータ	bit 7	6	5	4	3	2	1	0
アドレス:FFFFDF _H	M1	M0	予約	予約	S0	予約	予約	予約

[bit7, bit6] M1, M0 (バスモード設定ビット)
M1, M0 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 6.1-3 M1, M0 (バスモード設定ビット) の機能

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

[bit3] S0 (アクセスモード設定ビット)
S0 は、リセットシーケンス終了後のバスモード、アクセスモードを指定するビットです。

表 6.1-4 S0 (アクセスモード設定ビット) の機能

S0	機能
0	外部データバス 8 ビットモード
1	外部データバス 16 ビットモード

6.1.3 バスモード別メモリ空間

バスモードの指定による，アクセス領域と物理アドレスの対応を以下に示します。

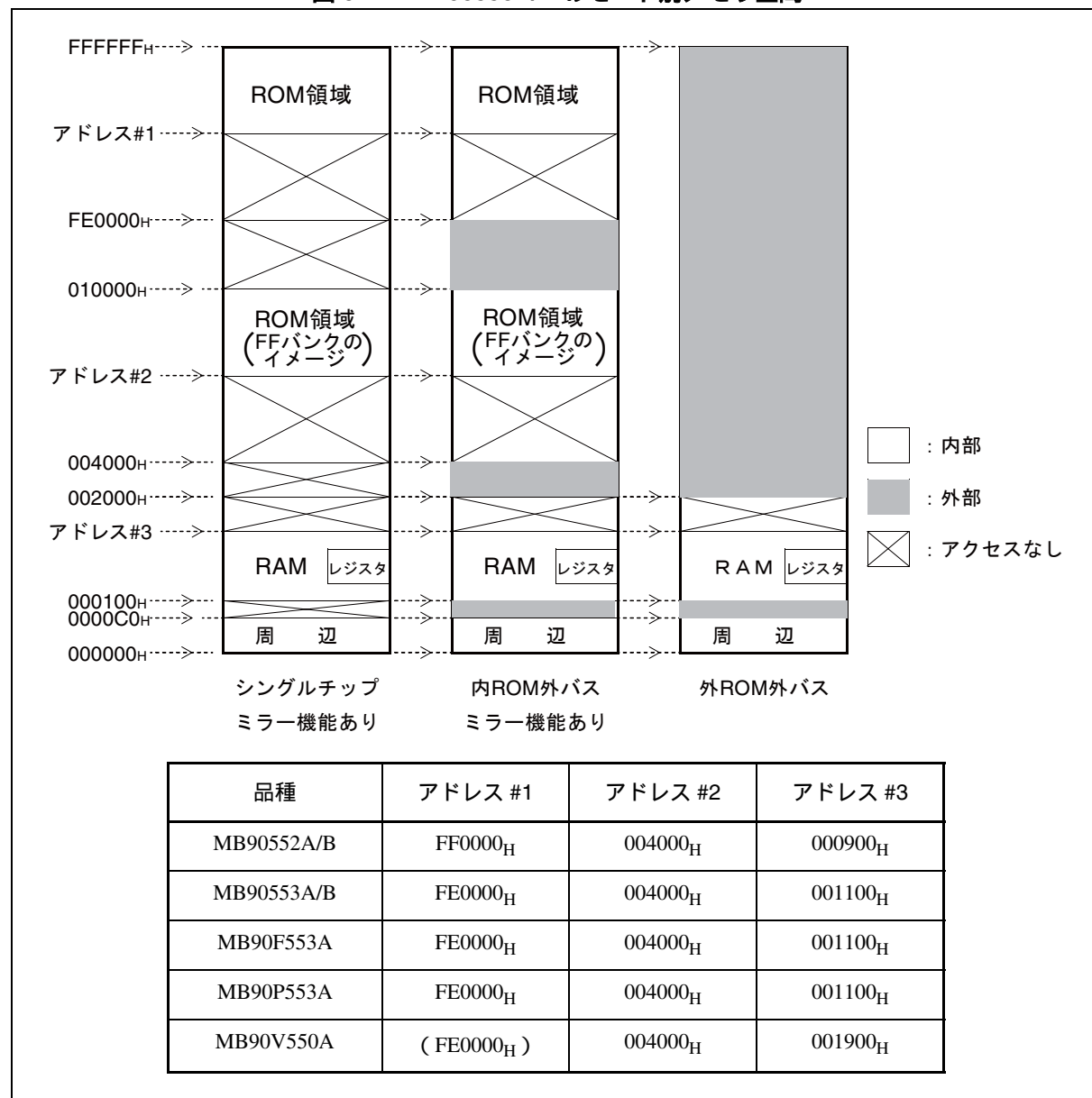
■ バスモード別メモリ空間

図 6.1-2 に示すように，00 バンクの上位に 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 指定をしなくとも 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となります。ここで，FF バンクの ROM 領域は，48K バイトを超しますので，00 バンクのイメージにすべての領域を見せることができません。したがって，FF4000_H ~ FFFFFFF_H の ROM データは 004000_H ~ 00FFFF_H のイメージに見えますので，ROM データテーブルは FF4000_H ~ FFFFFFF_H の領域に格納することを推奨いたします。

ROM のミラー機能なしを選択した場合は，「第 22 章 ROM ミラー機能選択モジュール」を参照してください。

図 6.1-2 MB90550A/B のモード別メモリ空間



■ バスモード別メモリ空間推奨設定例

表 6.1-5 に、モード端子とモードデータの推奨設定例を示します。

表 6.1-5 モード端子とモードデータの推奨設定例

設定例	MD2	MD1	MD0	M1	M0	S0
シングルチップ	0	1	1	0	0	x
内 ROM 外バス・8 ビットバス	0	1	1	0	1	0
内 ROM 外バス・16 ビットバス	0	1	1	0	1	1
外 ROM 外バス・16 ビットバス	0	0	1	1	0	1
外 ROM 外バス・8 ビットバス	0	0	0	1	0	0

外部端子は、各種モードにより入出力する信号が変化します。

表 6.1-6 各種モードと関係する外部端子の動作

端子名	機能			
	シングルチップ	外部バス拡張		EPROM 書込み
		8 ビット	16 ビット	
P07 ~ P00	ポート	AD07 ~ AD00		D07 ~ D00
P17 ~ P10		AD15 ~ AD08	AD15 ~ AD08	A15 ~ A08
P27 ~ P20		A23 ~ A16*		A07 ~ A00
P30		ALE		A16
P31		$\overline{\text{RD}}$		$\overline{\text{CE}}$
P32		$\overline{\text{WRL}}^*$		$\overline{\text{OE}}$
P33		ポート	$\overline{\text{WRH}}^*$	$\overline{\text{PGM}}$
P34		HRQ*		未使用
P35		$\overline{\text{HAK}}^*$		
P36		RDY*		
P37		CLK*		

*: アドレス上位と $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{HAK}}$, HRQ, RDY, CLK は、機能選択によりポートとして使用することが可能です。詳細は、「6.2 外部メモリアクセス（外部バス端子制御回路）」を参照ください。

6.2 外部メモリアクセス（外部バス端子制御回路）

外部バス端子制御回路は、CPU のアドレス / データバスを外部に拡張するための外部バス端子を制御します。

■ 外部メモリアクセス（外部バス端子制御回路）

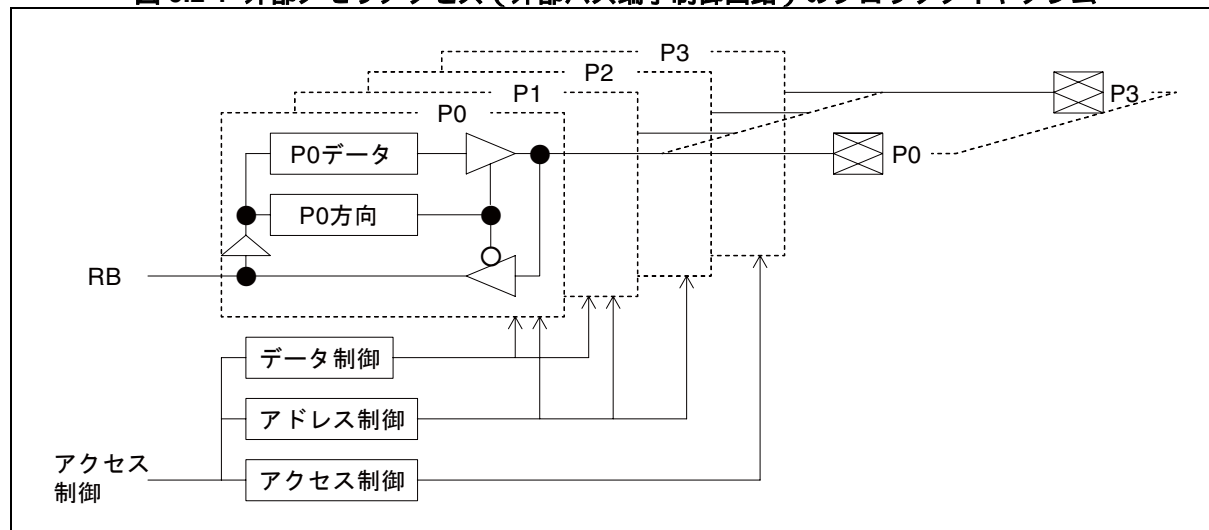
デバイス外部のメモリ / 周辺をアクセスするために、F²MC-16LX は以下に示すようなアドレス / データ / 制御信号を供給します。

- CLK (P37) : マシンサイクルクロック (KBP) を出力します
- RDY (P36) : 外部レディ入力端子です
- $\overline{\text{WRH}}$ (P33) : データバス上位 8 ビットのライト信号です
- $\overline{\text{WRL}}$ (P32) : データバス下位 8 ビットのライト信号です
- $\overline{\text{RD}}$ (P31) : リード信号です
- ALE (P30) : アドレスラッチイネーブル信号です

■ 外部メモリアクセス（外部バス端子制御回路）のブロックダイヤグラム

図 6.2-1 に、外部メモリアクセス（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 6.2-1 外部メモリアクセス（外部バス端子制御回路）のブロックダイヤグラム



6.2.1 外部メモリアクセス（外部バス端子制御回路）のレジスタ

外部メモリアクセス（外部バス端子制御回路）には、次の 3 種類のレジスタがあります。

- 自動レディ機能選択レジスタ
- 外部アドレス出力制御レジスタ
- バス制御信号選択レジスタ

■ 外部メモリアクセス（外部バス端子制御回路）のレジスタ

図 6.2-2 外部メモリアクセス（外部バス端子制御回路）のレジスタ一覧

自動レディ機能選択レジスタ								
	bit 15	14	13	12	11	10	9	8
アドレス:0000A5 _H	IOR1	IOR0	HMR1	HMR0	—	—	LMR1	LMR0
リード/ライト⇒	(W)	(W)	(W)	(W)	(-)	(-)	(W)	(W)
初期値⇒	(0)	(0)	(1)	(1)	(-)	(-)	(0)	(0)
外部アドレス出力制御レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:0000A6 _H	E23	E22	E21	E20	E19	E18	E17	E16
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
バス制御信号選択レジスタ								
	bit 15	14	13	12	11	10	9	8
アドレス:0000A7 _H	CKE	RYE	HDE	IOBS	HMBS	WRE	LMBS	—
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(-)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(-)

6.2.2 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) は、外部アクセス時の領域ごとのメモリアクセスの自動ウェイト時間を設定します。

■ 自動レディ機能選択レジスタ (ARSR)

図 6.2-3 自動レディ機能選択レジスタの構成

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

[bit15, bit14] IOR1, IOR0

IOR1, IOR0 ビットは、0000C0_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指定します。2 ビットの組合せで、表 6.2-1 に示すような設定になります。

表 6.2-1 IOR1, IOR0 (自動ウェイト機能指定ビット) の機能

IOR1	IOR0	機能
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が入る
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイトが入る
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイトが入る

[bit13, bit12] HMR1, HMR0

HMR1, HMR0 ビットでは、800000_H ~ FFFFFFF_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指定します。2 ビットの組合せで、表 6.2-2 に示すような設定になります。

表 6.2-2 HMR1, HMR0 (自動ウェイト機能指定ビット) の機能

HMR1	HMR0	機能
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が入る
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイトが入る
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイトが入る [初期値]

[bit9, bit8] LMR1, LMR0

LMR1, LMR0 ビットでは, 002000_H ~ 7FFFFFF_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指定します。2 ビットの組合せで, 表 6.2-3 に示すような設定になります。

表 6.2-3 LMR1, LMR0 (自動ウェイト機能指定ビット) の機能

LMR1	LMR0	機能
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が入る
1	0	外部アクセス時, 2 マシンサイクルの自動ウェイトが入る
1	1	外部アクセス時, 3 マシンサイクルの自動ウェイトが入る

6.2.3 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 (HACR) 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表 6.2-4 に示すように制御します。

■ 外部アドレス出力制御レジスタ (HACR)

図 6.2-4 外部アドレス出力制御レジスタの構成

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

HACR 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、全端子が I/O ポートとして機能します。

外部アドレス出力制御レジスタの全ビットは、すべて書込み専用で、読出しでは "1" となります。

アドレス出力を使用する場合は、ポート 2 方向レジスタ (DDR2) を "0" に設定してください。

表 6.2-4 外部アドレス出力制御レジスタ (E23 ~ E16 ビット) の機能

E23 ~ E16	機能
0	対応する端子はアドレス出力 (AXX) になります。[初期値]
1	対応する端子は I/O ポート (PXX) になります。

6.2.4 バス制御信号選択レジスタ（ECSR）

バス制御信号選択レジスタ（ECSR）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。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、全端子が I/O ポートとして機能します。バス制御信号選択レジスタのビットは、すべて書込み専用で、読出しでは "1" となります。

■ バス制御信号選択レジスタ（ECSR）

図 6.2-5 バス制御信号選択レジスタの構成

制御信号選択レジスタ								
	bit 15	14	13	12	11	10	9	8
アドレス:0000A7 _H	CKE	RYE	HDE	IOBS	HMBS	WRE	LMBS	—
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(-)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(-)

[bit15] CKE
CKE ビットでは、外部クロック（CLK）の出力を表 6.2-5 に示すように制御します。

表 6.2-5 CKE（外部クロック（CLK）出力制御ビット）の機能

CKE	機能
0	I/O ポート（P37）動作（クロック出力禁止）[初期値]
1	クロック信号（CLK）出力許可

[bit14] RYE
RYE ビットでは、外部レディ（RDY）の入力を、表 6.2-6 に示すように制御します。

表 6.2-6 RYE（外部レディ（RDY）入力制御ビット）の機能

RYE	機能
0	I/O ポート（P36）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

HDE ビットは、ホールド関係の端子の入出力許可を指定します。HDE ビットの設定により、ホールド要求入力 (HRQ) とホールドアクノリッジ出力 ($\overline{\text{HAK}}$) の 2 本を表 6.2-7 に示すように制御します。

表 6.2-7 HDE (ホールド関連端子の入出力許可指定ビット) の機能

HDE	機能
0	I/O ポート (P35, P34) 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

IOBS ビットでは、外部データバス 16 ビットモード時に、0000C0_H ~ 0000FF_H の領域に対する外部アクセスを行ったときのバス幅を指定します。本ビットの設定により、表 6.2-8 に示すように制御します。

表 6.2-8 IOBS (バス幅指定ビット)

IOBS	機能
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

[bit11] HMBS

HMBS 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アクセスを行ったときのバス幅を指定します。本ビットの設定により、表 6.2-9 に示すように制御します。

表 6.2-9 HMBS (バス幅指定ビット) の機能

HMBS	機能
0	16 ビットバス幅アクセス [外部ベクタモード 1 のときの初期値]
1	8 ビットバス幅アクセス [外部ベクタモード 0 のときの初期値]

[bit10] WRE

WRE ビットは、外部ライト信号 (外部データバス 16 ビットモードの場合は $\overline{\text{WRH}}$ / $\overline{\text{WRL}}$ 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、表 6.2-10 に示すように制御します。

外部データバス 8 ビットモードの場合は、本ビットの設定値に関係なく、P33 は I/O ポートとして機能します。

表 6.2-10 WRE (外部ライト信号出力制御ビット) の機能

WRE	機能
0	I/O ポート (P33, P32) 動作 (ライト信号出力禁止) [初期値]
1	ライトストロープ信号 ($\overline{\text{WRH}}$ / $\overline{\text{WRL}}$ または $\overline{\text{WRL}}$ のみ) の出力許可

[bit9] LMBS

LMBS 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アクセスを行ったときのバス幅を指定します。本ビットの設定により、表 6.2-11 のように制御します。

表 6.2-11 LMBS（バス幅指定ビット）の機能

LMBS	機能
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 >

- 外部データバス 16 ビットモード時、WRE ビットにて \overline{WRH} , \overline{WRL} 機能を許可する場合は、P33, P32 を入力モードにしてください（DDR3 の bit3, bit2 は "0" にしてください）。
- 外部データバス 8 ビットモード時、WRE ビットにて \overline{WR} 機能を許可する場合は、P32 を入力モードにしてください（DDR3 の bit2 は "0" にしてください）。
- また、RYE, HDE ビットにて RDY, HRQ 入力を許可した場合でも、そのポートの I/O ポート機能は有効となります。そのため、DDR3 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。

6.3 外部メモリアクセス制御信号の動作

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リード/ライトするための機能です。

■ 外部メモリアクセス制御信号

8 ビットバス幅アクセスは、データバス下位 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、データバスの下位 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 ビットで指定することになります。

なお、アドレス出力と ALE のアサート出力のみを行い、 $\overline{\text{RD}}/\text{WRL}/\text{WRH}$ 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。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。

図 6.3-1 に、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を、図 6.3-2 に、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（16 ビットバス幅アクセス、8 ビットバス幅アクセス時）を示します。

図 6.3-1 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

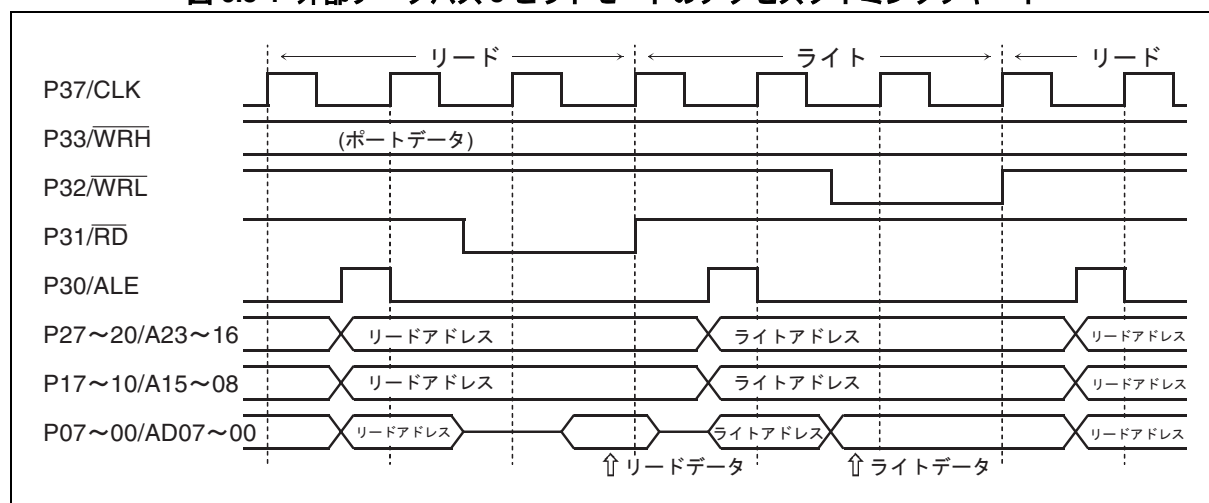
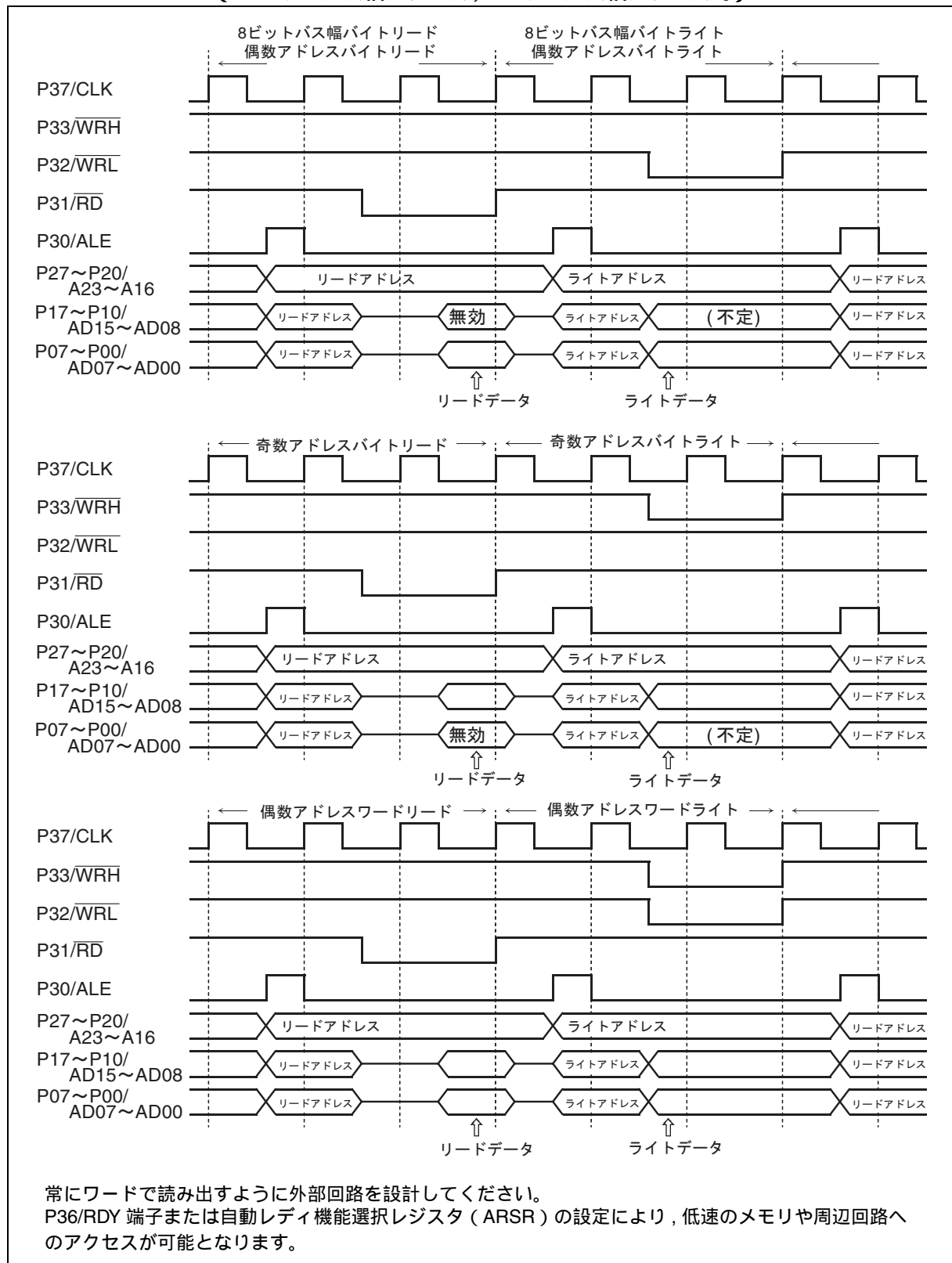


図 6.3-2 外部データバス 16 ビットモードのアクセスタイミングチャート
(16 ビットバス幅アクセス, 8 ビットバス幅アクセス時)



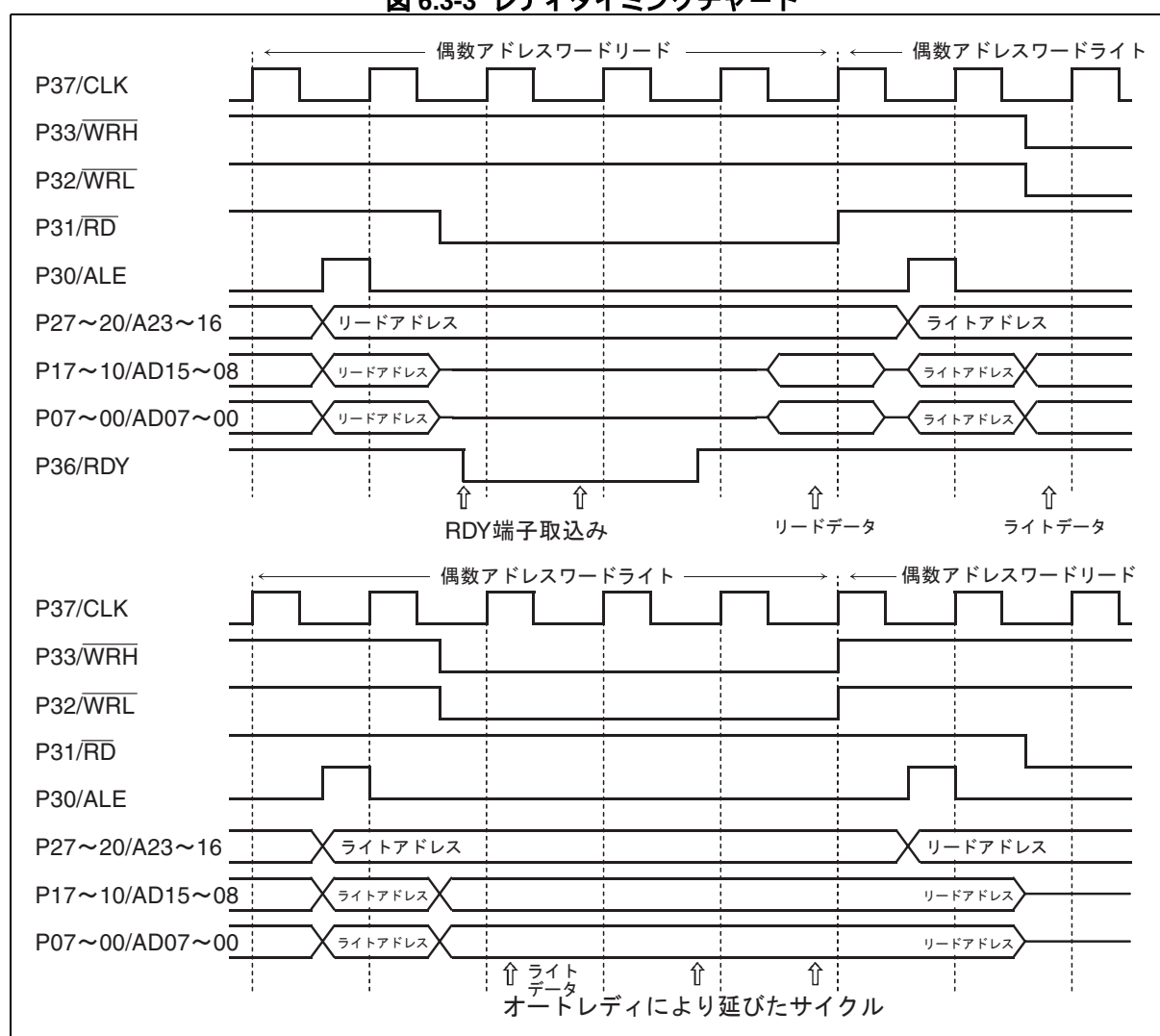
6.3.1 レディ機能

P36/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 P36/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可能です。

■ レディ機能

図 6.3-3 レディタイミングチャート



また、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と、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、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 用の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 0000C0_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、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でも EPCR 中の RYE ビットが "1" に設定されている場合、上記オートレディ機能によるウェイトサイクル終了後に P36/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。

6.3.2 ホールド機能

EPCR 中の HDE ビットが "1" に設定されている場合, P34/HRQ, P35/ $\overline{\text{HAK}}$ 両端子による外部バスホールド機能が有効となります。

■ ホールド機能

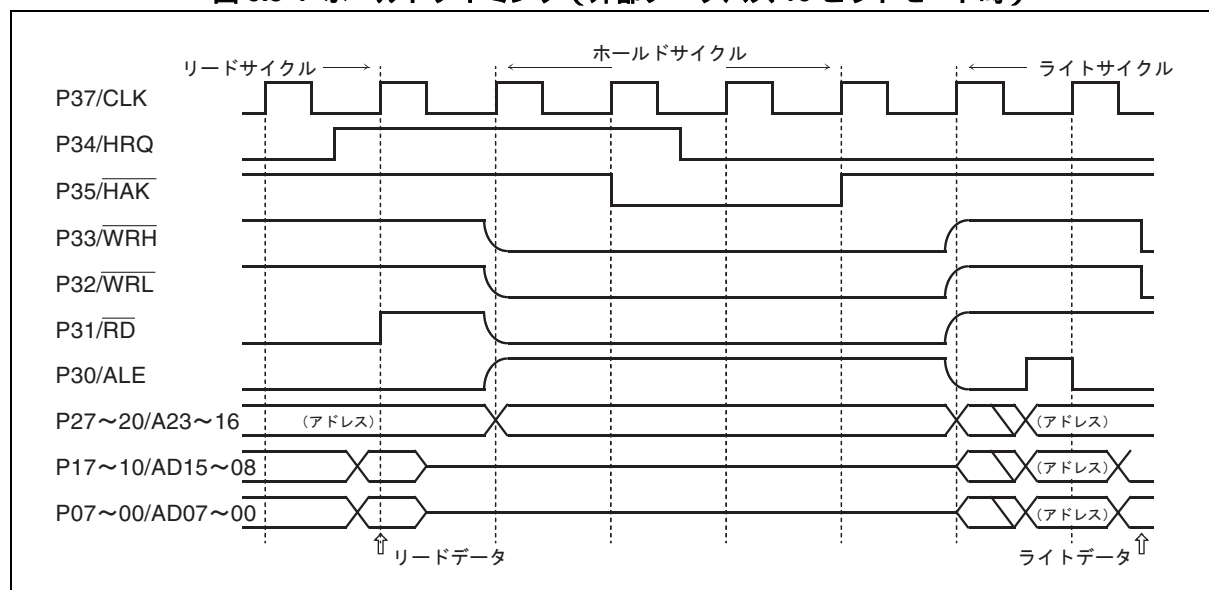
P34/HRQ 端子に H レベルを入力すると, CPU の命令の終了時 (ストリング命令の場合は "1" エLEMENTデータの処理の終了時) にてホールド状態となり, P35/ $\overline{\text{HAK}}$ 端子より L レベルを出力して以下の端子をハイインピーダンス状態にします。

- アドレス出力: P23/A19 ~ P20/A16
- アドレス / データ入出力: P17/D15 ~ P00/D00
- バス制御信号: P30/ALE, P31/ $\overline{\text{RD}}$, P32/ $\overline{\text{WRL}}$, P33/ $\overline{\text{WRH}}$

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。P34/HRQ 端子に L レベルを入力すると, P35/ $\overline{\text{HAK}}$ 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, ホールド要求入力を受け付けません。

図 6.3-4 に, ホールドタイミング (外部データバス 16 ビットモード時) を示します。

図 6.3-4 ホールドタイミング (外部データバス 16 ビットモード時)



第7章

I/O ポート

I/O ポートの機能と動作について説明します。

- 7.1 I/O ポートの概要
- 7.2 I/O ポートブロックダイアグラム
- 7.3 I/O ポートのレジスタ

7.1 I/O ポートの概要

それぞれのポートの各端子は、対応する周辺が端子を用いない設定となっているとき、1本ごとに方向レジスタの設定により入力、出力の指定を行うことができます。また、入力時にデータレジスタを読出した場合は、常に、端子のレベルによる値が読出され、出力時にデータレジスタを読出した場合は、データレジスタのラッチの値が読み出されます。これは、リードモディファイライトのリードでも同じです。

■ I/O ポートの概要

制御出力として使用している場合にデータレジスタを読み出したときは、方向レジスタの値に関係なく、制御出力として出力しているものが、読出されます。入力設定から出力設定に変更する場合、あらかじめデータレジスタに出力データを設定するときに、リードモディファイライト系の命令(ビットセット等の命令)を使用しますと、リードしてくるデータは端子よりの入力データであり、データレジスタのラッチの値ではありませんので、注意が必要です。

ポート0～4, 6～Aは、入出力ポートで方向レジスタの値が"0"のとき入力、"1"のとき出力になります。ポート5は、オープンドレインポートです。

MB90550A/B シリーズでは、ポート0～3は外部バス端子と共用になっています。したがって、外部バスモードで使用する際はこれらのポートは使用が制限されます。

ポート2, 3については機能選択により、外部バスモード時においてもポートとして利用できるビットがあります。詳細については「6.2 外部メモリアクセス(外部バス端子制御回路)」を参照ください。

7.2 I/O ポートブロックダイヤグラム

図 7.2-1 ~ 図 7.2-5 に、次の I/O ポートブロックダイヤグラムを示します。

- パラレルポートブロックダイヤグラム (ポート 0, 1)
- パラレルポートブロックダイヤグラム (ポート 2, 3, 7, 8, 9, A)
- パラレルポートブロックダイヤグラム (ポート 4)
- パラレルポートブロックダイヤグラム (ポート 5)
- パラレルポートブロックダイヤグラム (ポート 6)

■ I/O ポートのブロックダイヤグラム

図 7.2-1 パラレルポートブロックダイヤグラム (ポート 0, 1)

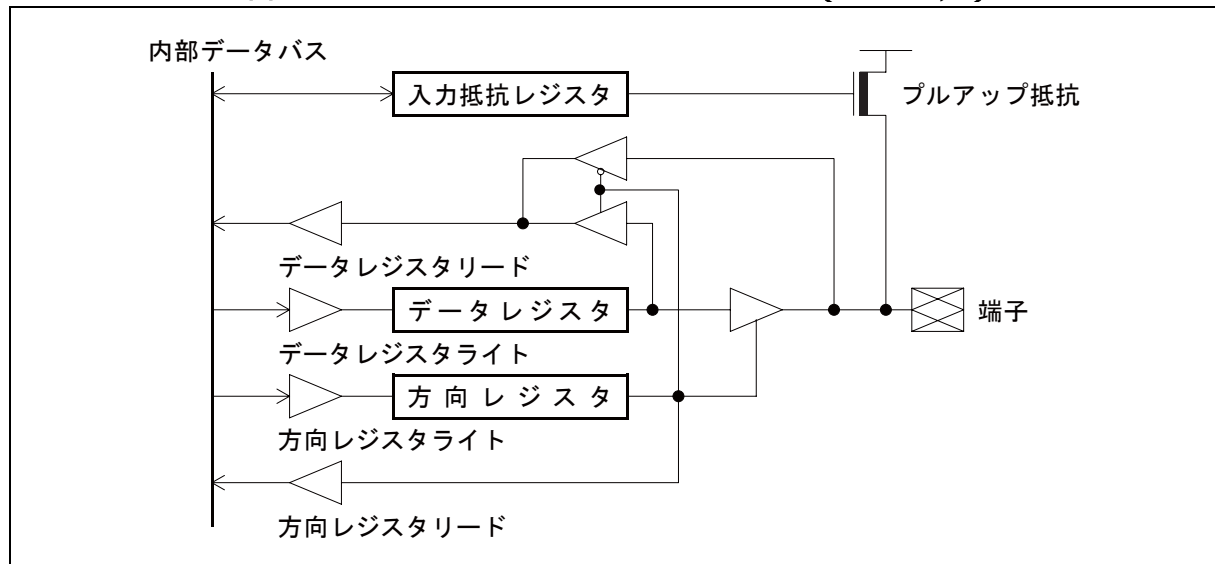


図 7.2-2 パラレルポートブロックダイヤグラム (ポート 2, 3, 7, 8, 9, A)

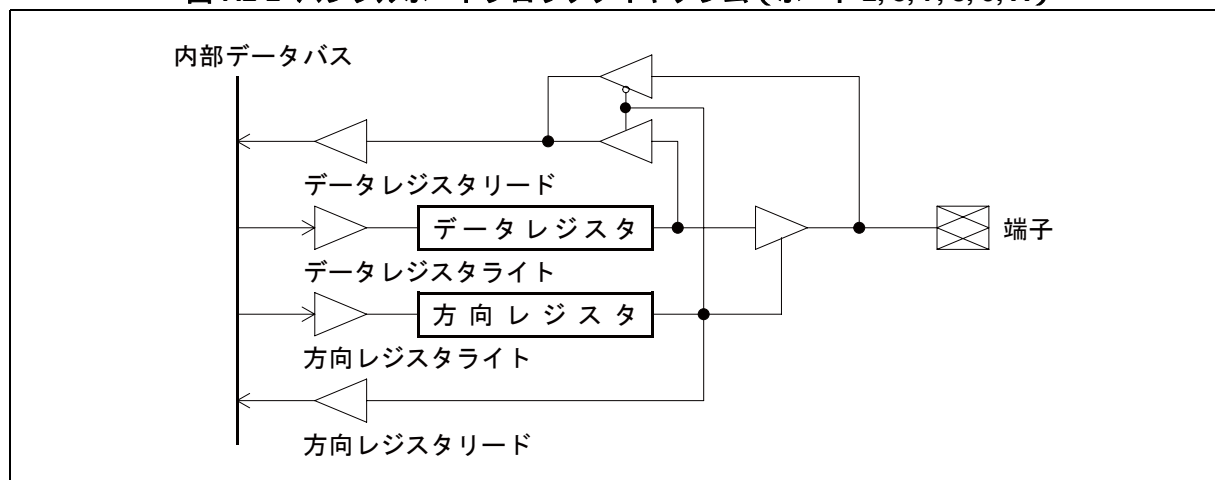


図 7.2-3 パラレルポートブロックダイアグラム (ポート4)

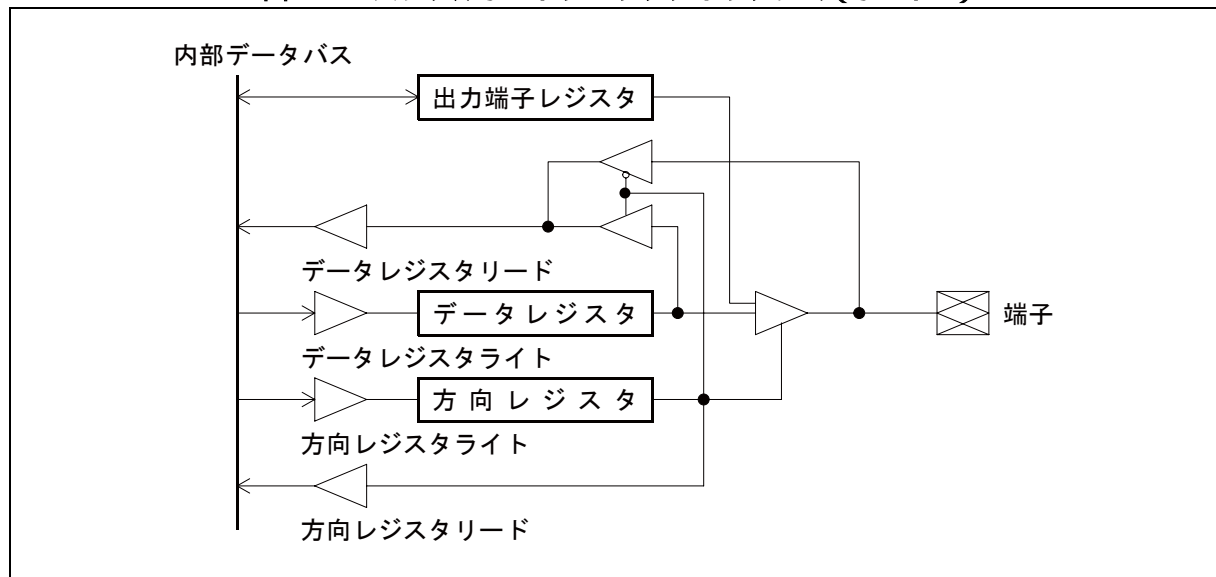


図 7.2-4 パラレルポートブロックダイアグラム (ポート5)

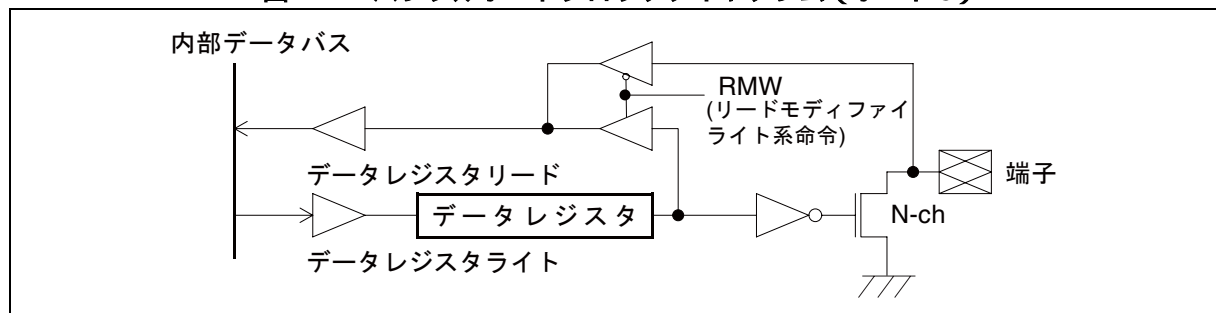
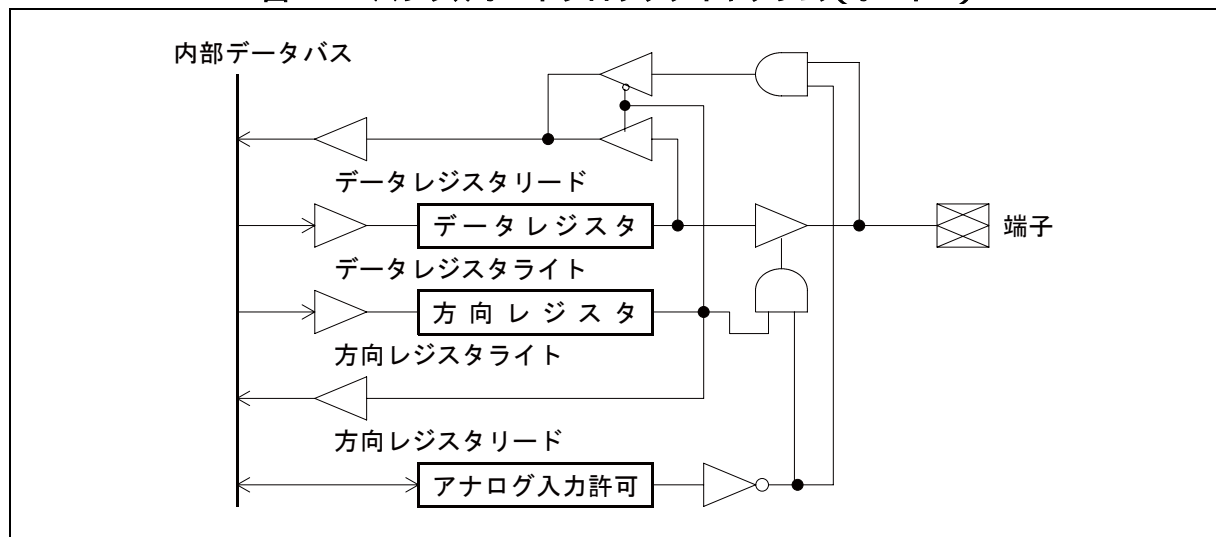


図 7.2-5 パラレルポートブロックダイアグラム (ポート6)



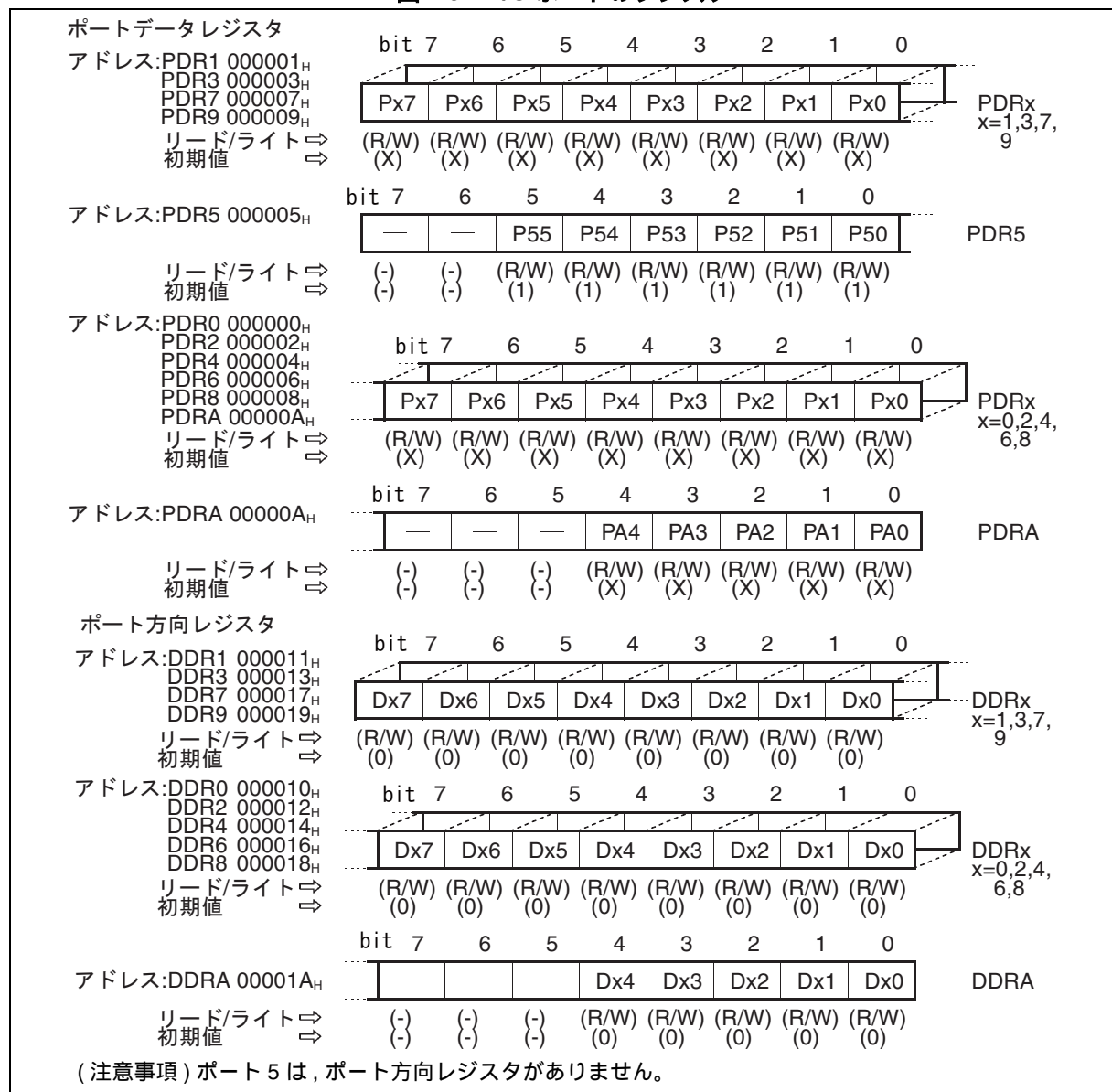
7.3 I/O ポートのレジスタ

I/O ポートのレジスタには、次の5種類があります。

- ポートデータレジスタ (PDRx)
- ポート方向レジスタ (DDRx)
- 出力端子レジスタ (ODRx)
- 入力抵抗レジスタ (RDR0, RDR1)
- アナログ入力許可レジスタ (ADER)

■ I/O ポートのレジスタ

図 7.3-1 I/O ポートのレジスタ



(続く)

(続き)

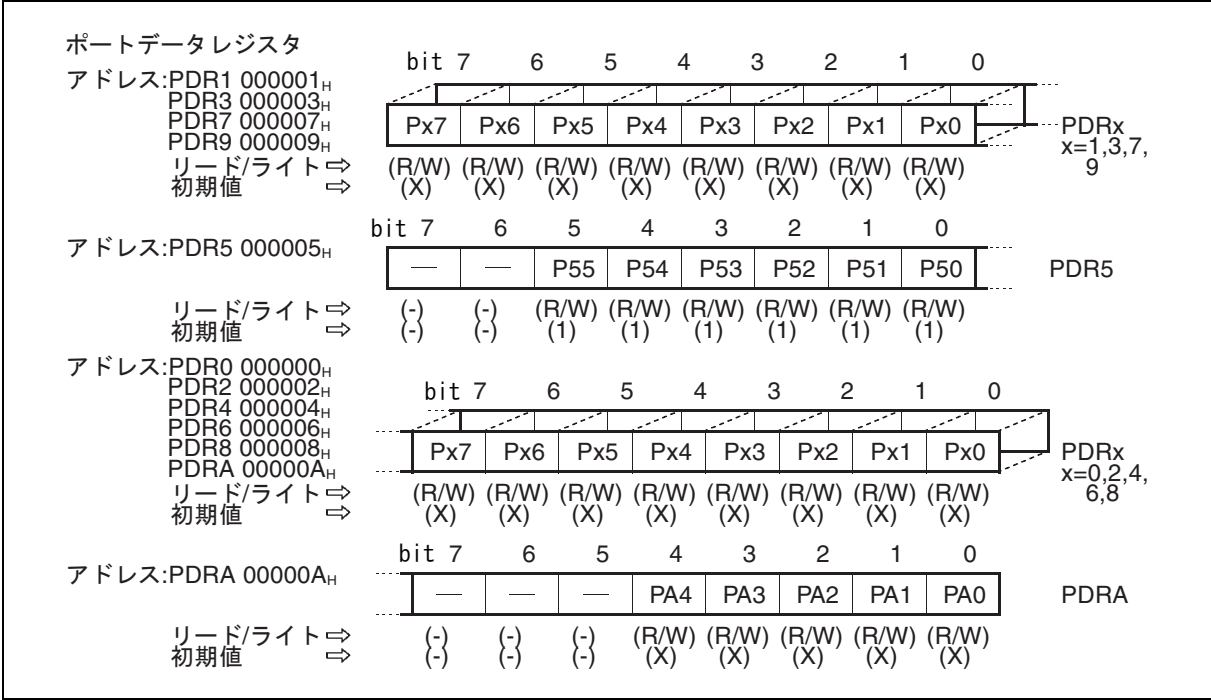
ポート4出力端子レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:00001B _H	OD47	OD46	OD45	OD44	OD43	OD42	OD41	OD40
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
ポート0入力抵抗レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:00001C _H	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00
リード/ライト⇒	(R/W)	(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)
ポート1入力抵抗レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:00001D _H	RD17	RD16	RD15	RD14	RD13	RD12	RD11	RD10
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)

7.3.1 ポートデータレジスタ (PDRx)

ポートデータレジスタ (PDRx) では、端子の状態を読み出します。

■ ポートデータレジスタ (PDRx)

図 7.3-2 ポートデータレジスタ



< 注意事項 >

入出力ポートの R/W は、メモリへの R/W とは動作が異なりますので注意してください。

入力モード

- ・ 読み出し時: 対応する端子のレベルが読み出されます。
- ・ 書き込み時: 出力用のラッチに書き込まれます。

出力モード

- ・ 読み出し時: データレジスタラッチの値が読み出されます。
- ・ 書き込み時: 対応する端子に出力されます。

ポート 5 の R/W は、ほかのポートとは動作が異なりますので注意してください。

ポート 5 (P55 ~ P50) はオープンドレイン出力形式の汎用入出力ポートです。

ポート 5 を入力ポートとして使用する場合は、オープンドレイン出力トランジスタを OFF にするために出力データレジスタの内容を "1" にしておき、外部にプルアップ抵抗を付ける必要があります。

さらに、読み出し時には使用する命令に依存して次の 2 つの異なった動作を行います。

- リードモディファイライト系の命令での読出し時

出力データレジスタの内容を読み出します。外部より強制的に端子を "0" に駆動しても命令で指定していないビットの内容まで変化することはありません。

- 上記以外の命令での読出し時

端子のレベルを読み出すことができます。

なお、出力ポートとして使用する場合は、該当する出力データレジスタへ希望の値を書き込むことで、端子の値を変化させることができます。

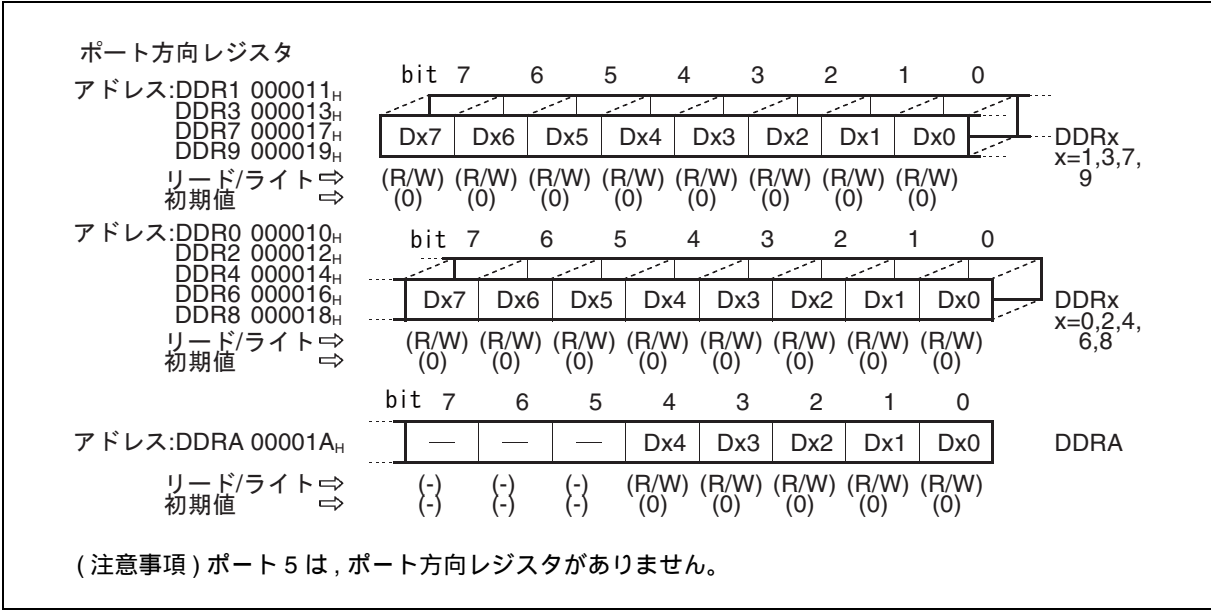
ADER レジスタが "1" のビットに対応する端子は読出し時には "0" が読めます。

7.3.2 ポート方向レジスタ (DDR_x)

ポート方向レジスタ (DDR_x) では、ビットごとの端子の入出力方向を設定します。ポート (端子) に対応するビットを "1" にすると出力ポートになり、"0" にすると入力ポートになります。

■ ポート方向レジスタ (DDR_x)

図 7.3-3 ポート方向レジスタ (DDR_x)



ポート方向レジスタ (DDR_x) では、各端子がポートとして機能しているとき、対応する各端子を表 7.3-1 のように制御します。

表 7.3-1 ポート方向レジスタ (DDR_x) の機能

DDR _x	機能
0	入力モード [初期値]
1	出力モード

7.3.3 出力端子レジスタ（ODR4）

出力端子レジスタ（ODR4）では，出力モード時のオープンドレインを制御します。

■ 出力端子レジスタ（ODR4）

図 7.3-4 出力端子レジスタ（ODR4）

ポート4出力端子レジスタ							
bit 7 6 5 4 3 2 1 0							
アドレス:00001B _H	OD47	OD46	OD45	OD44	OD43	OD42	OD41 OD40
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)

表 7.3-2 出力端子レジスタ（ODR4）の機能

ODR4	機能
0	出力モード時標準出力ポートになります。[初期値]
1	出力モード時オープンドレイン出力ポートになります。

< 注意事項 >

- 入力モード時は意味を持ちません。（出力 Hi-Z）
- 入出力モードは方向レジスタ（DDR）で決まります。

7.3.4 入力抵抗レジスタ (RDR0, RDR1)

入力抵抗レジスタ (RDR0, RDR1) は、入力モード時のプルアップ抵抗を制御します。

■ 入力抵抗レジスタ (RDR0, RDR1)

図 7.3-5 入力抵抗レジスタ (RDR0, RDR1)

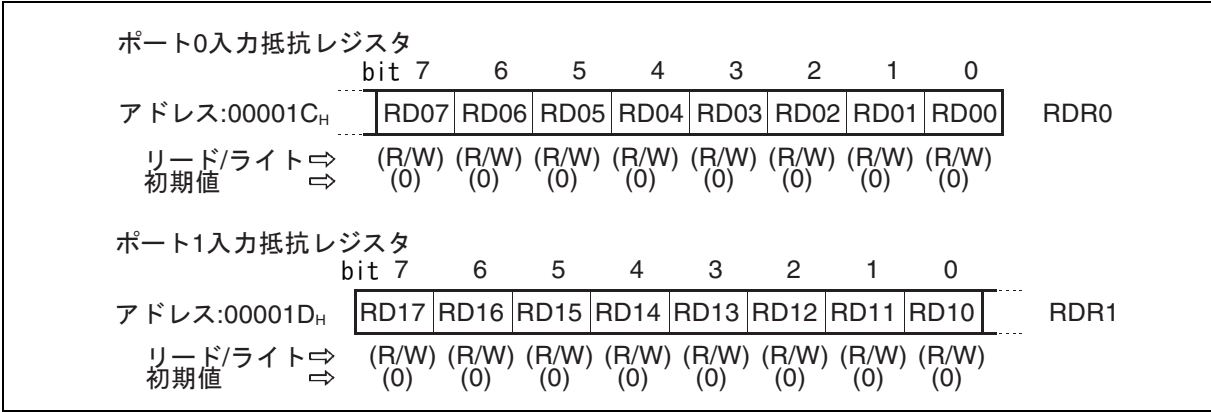


表 7.3-3 入力抵抗レジスタ (RDR0, RDR1) の機能

RDR0, RDR1	機能
0	入力モード時プルアップ抵抗なし。[初期値]
1	入力モード時プルアップ抵抗あり。

< 注意事項 >

- 出力モード時は意味を持ちません（プルアップ抵抗なし）。
- 入出力モードは方向レジスタ（DDR）で決まります。
- ハードウェアスタンバイ時とストップ時（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）。
- 外バスで使用するときは本機能は禁止です。本レジスタは書き込まないようにしてください。

7.3.5 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) は , ポート 6 の各端子を表 7.3-4 に示すように制御します。

■ アナログ入力許可レジスタ (ADER)

図 7.3-6 ポート 6 アナログ入力許可レジスタ (ADER)

ポート6アナログ入力許可レジスタ									
	bit 7	6	5	4	3	2	1	0	
アドレス:00001F _H	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	ADER
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

表 7.3-4 ポート 6 アナログ入力許可レジスタ (ADER)

ADER	設定
0	ポート入力モード
1	アナログ入力モード [初期値]

< 注意事項 >

ポート入力モード時に , 中間レベルの信号が入力されると , 入力リーク電流が流れますので , アナログ入力を行うときは , アナログ入力モードにしてください。

第8章

タイムベースタイマ

タイムベースタイマの機能と動作について説明します。

- 8.1 タイムベースタイマの概要
- 8.2 タイムベースタイマ制御レジスタ (TBTC)
- 8.3 タイムベースタイマの動作

8.1 タイムベースタイマの概要

タイムベースタイマは、18bit のタイマのほかにインターバルな割込みを制御する回路から構成されています。なお、タイムベースタイマは、CKSCR 中の MCS ビットと関係なく、発振クロックを使用します。

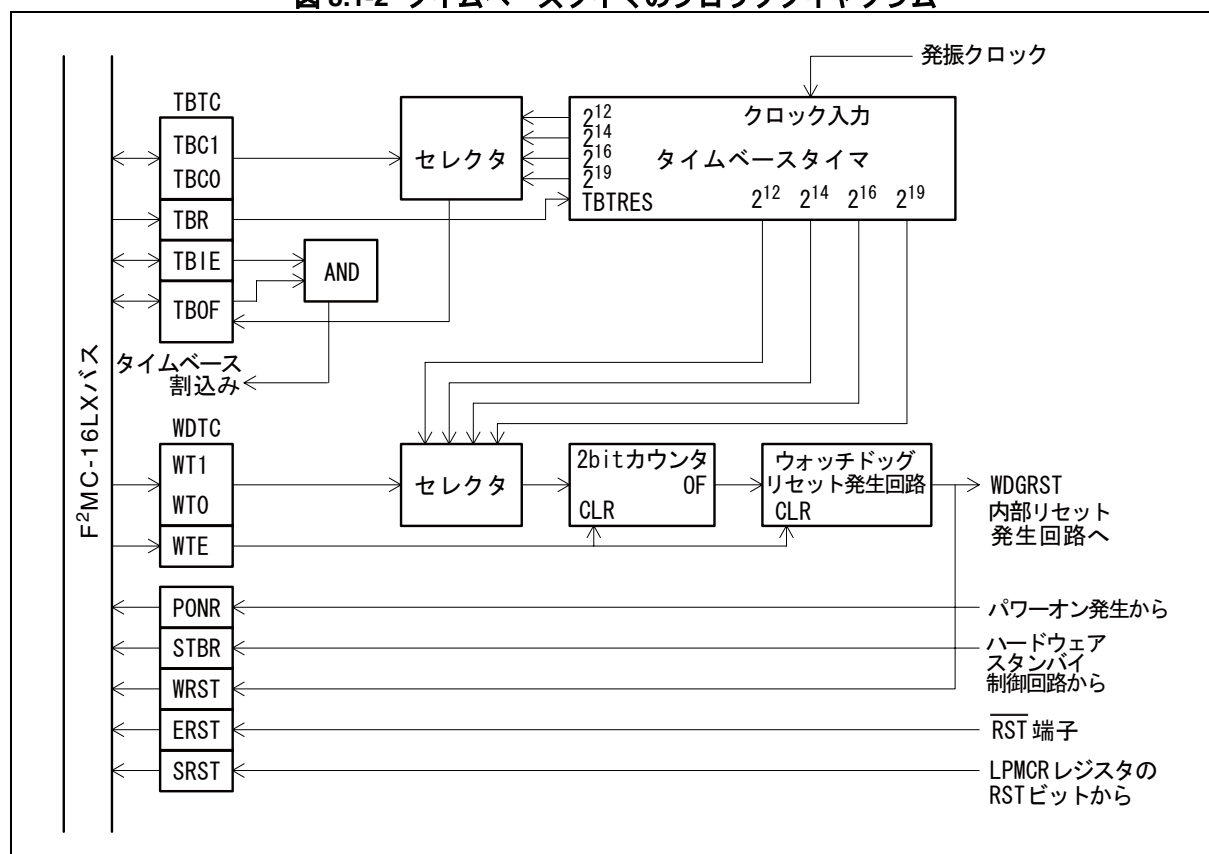
■ タイムベースタイマのレジスタ一覧

図 8.1-1 タイムベースタイマのレジスタ一覧

タイムベースタイマ制御レジスタ								
bit 15 14 13 12 11 10 9 8								
アドレス:0000A9 _H	予約	—	—	TBIE	TBOF	TBR	TBC1	TBC0
リード/ライト ⇒	(-)	(-)	(-)	(R/W)	(R/W)	(W)	(R/W)	(R/W)
初期値 ⇒	(1)	(-)	(-)	(0)	(0)	(1)	(0)	(0)

■ タイムベースタイマのブロックダイアグラム

図 8.1-2 タイムベースタイマのブロックダイアグラム



8.2 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) では、タイムベースタイマの割込みを制御し、タイムベースカウンタをクリアすることができます。

■ タイムベースタイマ制御レジスタ (TBTC)

図 8.2-1 タイムベースタイマ制御レジスタ (TBTC) 構成

タイムベースタイマ制御レジスタ									
bit 15 14 13 12 11 10 9 8									
アドレス:0000A9 _H	予約	—	—	TBIE	TBOF	TBR	TBC1	TBC0	TBTC
リード/ライト ⇒ 初期値 ⇒	(-) (1)	(-) (-)	(-) (-)	(R/W) (0)	(R/W) (0)	(W) (1)	(R/W) (0)	(R/W) (0)	

< 注意事項 >

リードモディファイ系の命令でのアクセスは誤動作を引き起こしますので、これらの命令でアクセスしないでください。

[bit15] 予約 (Reserved)

bit15 は、予約ビットです。TBTC を設定する時は、必ず "1" に設定してください。

[bit12] TBIE

TBIE は、タイムベース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。"1" のとき割込み許可、"0" のとき割込みを禁止します。リセットにて "0" に初期化されます。読み書き可能なビットです。

[bit11] TBOF

TBOF は、タイムベースタイマの割込み要求フラグです。TBIE ビットが "1" のとき TBOF が、"1" になると割込み要求が発生します。TBC1, TBC0 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、ハードウェアスタンバイモードへの遷移およびリセットによりクリアされます。"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における読出し時には "1" が読み出されます。

[bit10] TBR

TBR は、タイムベース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。"0" を書き込むことによりタイムベースカウンタをクリアします。"1" の書込みは意味がありません。読出し時は "1" が読み出されます。

< 注意事項 >

TBOF ビットをクリアする場合は , TBIE ビットまたは CPU の ILM ビットによりタイムベースタイマ割込みがマスクされている状態で行うようにしてください。

[bit9, bit8] TBC1, TBC0

TBC1, TBC0 は , タイムベースタイマのインターバルを設定するビットです。

リセットにより "00" に初期化されます。読み書き可能なビットです。

表 8.2-1 TBC1, TBC0 のインターバル時間とサイクル数

TBC1	TBC0	原振 4MHz 時のインターバル時間	発振クロック（原発振）サイクル数
0	0	1.024 ms	2^{12} サイクル
0	1	4.096 ms	2^{14} サイクル
1	0	16.384 ms	2^{16} サイクル
1	1	131.072 ms	2^{19} サイクル

8.3 タイムベースタイマの動作

タイムベースタイマは、ウォッチドッグタイマのクロックソース、メインクロックおよび PLL クロックの発振安定時間待ちのためのタイマ、一定周期で割込みを発生するインターバルタイマの機能があります。

■ タイムベースタイマの動作

タイムベースタイマは、マシンのクロックを作成するもとになる原振入力をカウントする 18 ビットのカウンタにより構成され、原振が入力されている間常にカウント動作を続けます。

タイムベース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、ハードウェアスタンバイモードへの遷移、CKSCR レジスタ中の MCS ビットによるメインクロックから PLL クロックへの遷移、TBTC レジスタ中の TBR ビットへの "0" 書込みにより行われます。

タイムベース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タイムベースタイマのクリアの影響があります。

■ インターバル割込み機能

タイムベース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。TBTC レジスタ中の TBC1、TBC0 ビットで設定されるインターバル時間ごとに TBOF フラグをセットします。このフラグのセットは、最後にタイムベースタイマがクリアされた時間を基準にして行われます。

メインクロックモードから PLL クロックモードに遷移すると、タイムベースタイマは PLL クロックの発振安定待ち時間のタイマとして使用するために、タイムベースタイマのクリアが行われます。

ストップモードおよびハードウェアスタンバイモードに遷移するとタイムベースタイマは、復帰時の発振安定時間待ち時間のタイマとして使用されるので TBOF フラグは、モード遷移と同時にクリアされます。

第9章

ウォッチドッグタイマ

ウォッチドッグタイマの機能と動作について説明します。

- 9.1 ウォッチドッグタイマの概要
- 9.2 ウォッチドッグタイマ制御レジスタ (WDTC)
- 9.3 ウォッチドッグタイマの動作

9.1 ウォッチドッグタイマの概要

ウォッチドッグタイマは、18bit のタイムベースタイマの桁上り信号をクロックソースとする 2bit のウォッチドッグカウンタ、コントロールレジスタおよびウォッチドッグリセット制御部により構成されています。

■ ウォッチドッグタイマのレジスタ一覧

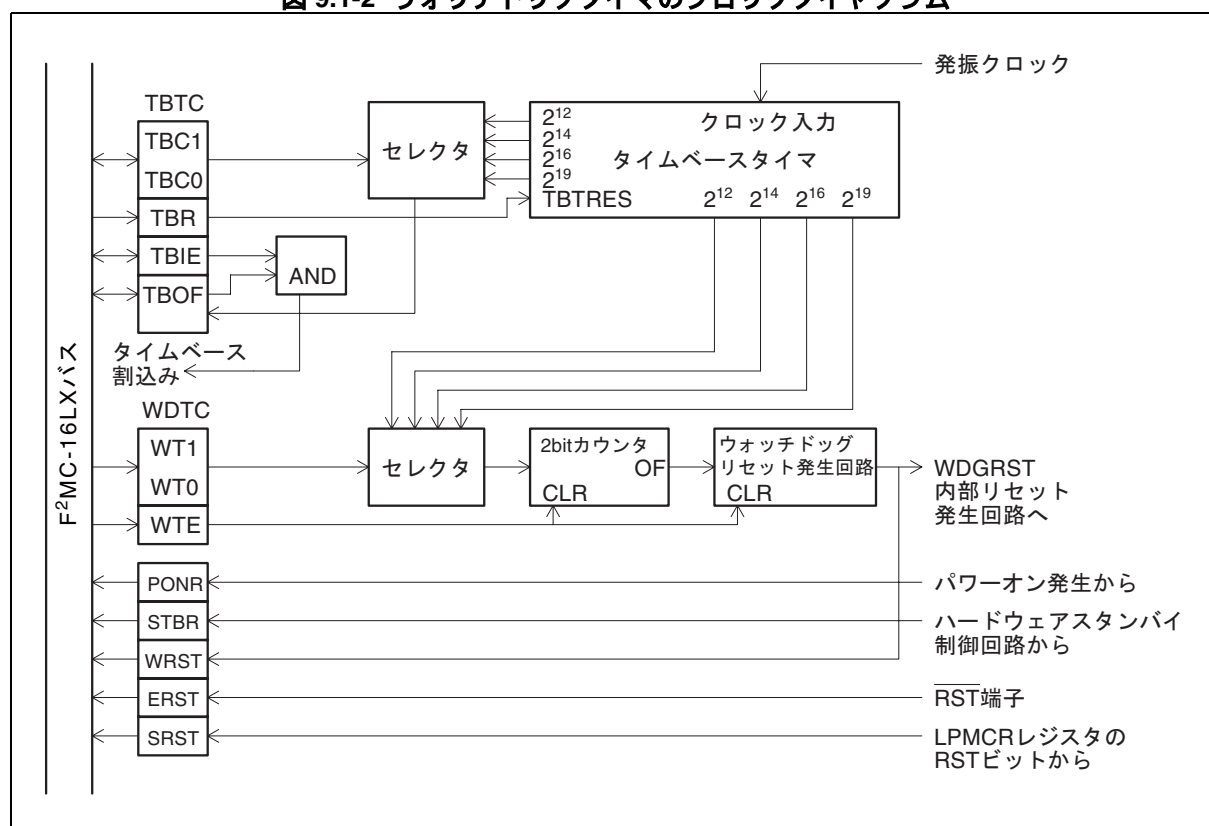
図 9.1-1 ウォッチドッグタイマのレジスタ一覧

ウォッチドッグ制御レジスタ								
bit 7 6 5 4 3 2 1 0								
アドレス:0000A8 _H	PONR	STBR	WRST	ERST	SRST	WTE	WT1	WT0
リード/ライト⇒ 初期値	(R) (X)	(R) (X)	(R) (X)	(R) (X)	(R) (X)	(W) (1)	(W) (1)	(W) (1)

WDTC

■ ウォッチドッグタイマのブロックダイアグラム

図 9.1-2 ウォッチドッグタイマのブロックダイアグラム



9.2 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) では, 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います。

■ ウォッチドッグタイマ制御レジスタ (WDTC)

図 9.2-1 ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグ制御レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:0000A8 _H	PONR	STBR	WRST	ERST	SRST	WTE	WT1	WT0
リード/ライト⇒	(R)	(R)	(R)	(R)	(R)	(W)	(W)	(W)
初期値	(X)	(X)	(X)	(X)	(X)	(1)	(1)	(1)

< 注意事項 >

リードモディファイ系の命令でのアクセスは誤動作を引き起こしますので, これらの命令でアクセスしないでください。

[bit7 ~ bit3] PONR, STBR, WRST, ERST, SRST

PONR, STBR, WRST, ERST, SRST は, リセットの要因を示すフラグです。各リセットにより表 9.2-1 に示すように設定されます。WDTC レジスタのリード動作後に, 全ビットが, "0" にクリアされます。

読出し専用のレジスタです。

表 9.2-1 PONR, STBR, WRST, ERST, SRST (リセットの要因ビット)

リセット要因	PONR	STBR	WRST	ERST	SRST
パワーオン	1				
ハードウェアスタンバイ	*	1	*	*	*
ウォッチドッグタイマ	*	*	1	*	*
外部端子	*	*	*	1	*
RST ビット	*	*	*	*	1

※: 前の値を保持します。

[bit2] WTE

WTE は, ウォッチドッグタイマが停止状態の時 "0" を書き込むことによりウォッチドッグタイマを動作状態にします。2 回目以降の "0" の書込みではウォッチドッグタイマのカウンタをクリアします。"1" の書込みは何の操作もしません。

ウォッチドッグタイマは, パワーオン, ハードウェアスタンバイ, ウォッチドッグタイマによるリセットにより停止状態となります。読出し時は "1" が読み出されます。

[bit1, bit0] WT1, WT0

WT1, WT0 は、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。ウォッチドッグタイマ起動時に書き込まれたデータのみが有効です。ウォッチドッグ起動時以外の書き込みデータは無視されます。書き込みのみが可能なビットです。

表 9.2-2 WT1,WT0 (インターバル時間選択ビット)

WT1	WT0	インターバル時間 (発振クロック周波数 4MHz 時)		発振クロック (原発振) サイクル数
		最小	最大	
0	0	約 3.58 ms	約 4.61 ms	$2^{14} \pm 2^{11}$ サイクル
0	1	約 14.33 ms	約 18.43 ms	$2^{16} \pm 2^{13}$ サイクル
1	0	約 57.23 ms	約 73.73 ms	$2^{18} \pm 2^{15}$ サイクル
1	1	約 458.75 ms	約 589.82 ms	$2^{21} \pm 2^{18}$ サイクル

< 注意事項 >

インターバル時間は、タイムベースタイマまたは時計タイマの桁上り信号をカウントクロックにしているため、タイムベースタイマがクリアされると、ウォッチドッグタイマのインターバル時間が長くなる場合があります。

タイムベースタイマは、タイムベースタイマ制御レジスタ(TBTC)のTBR ビットへの"0"書き込みのほか、メインクロックモードからPLLクロックモードへの遷移時にもクリアされますのでご注意ください。

9.3 ウォッチドッグタイマの動作

ウォッチドッグタイマ機能により、プログラムの暴走を検出することができます。プログラムの暴走などにより、WDTC レジスタの WTE ビットへの "0" 書き込みが定められた時間内に行われなかった場合、ウォッチドッグタイマよりリセット要求が発生します。

■ ウォッチドッグタイマの起動方法

ウォッチドッグタイマの起動は、ウォッチドッグタイマの停止中に WDTC レジスタの WTE ビットに "0" を書き込むことによって行われます。このとき、同時にウォッチドッグタイマのリセット発生インターバルを WT1, WT0 ビットにより設定します。インターバルの設定は、この起動のときのデータのみが有効となります。

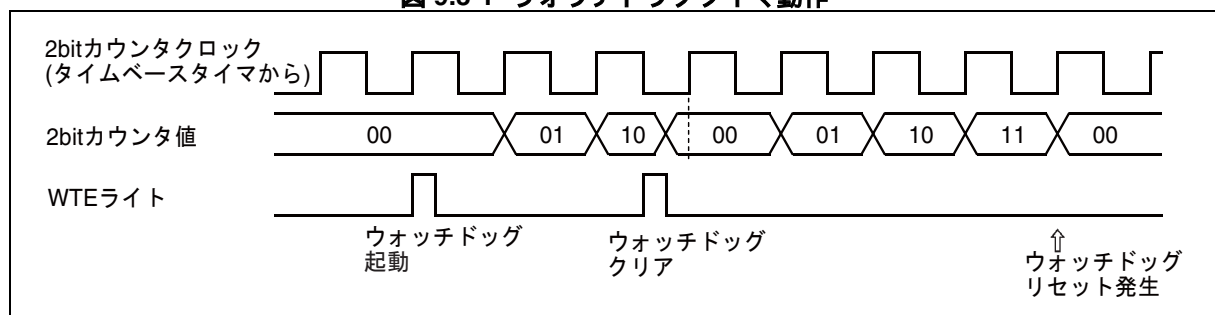
■ ウォッチドッグタイマリセットの阻止

ウォッチドッグタイマが起動されると、プログラム中で定期的に 2 ビットのウォッチドッグカウンタをクリアする必要があります。具体的には、WDTC レジスタの WTE ビットに定期的に "0" を書き込む必要があります。ウォッチドッグカウンタは、タイムベースタイマの桁上り信号をクロックソースとする 2 ビットのカウンタにより構成されています。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より長くなることがあります。

< 注意事項 >

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 TBR ビットへの "0" 書き込みのほか、メインクロックモードから PLL クロックモードへの遷移時にもクリアされますので、ご注意ください。

図 9.3-1 ウォッチドッグタイマ動作



■ ウォッチドッグ停止

ウォッチドッグタイマは、一度起動すると、パワーオン、ハードウェアスタンバイ、ウォッチドッグによるリセットでのみ初期化され停止状態となります。

外部端子、ソフトウェアによるリセットではウォッチドッグのカウンタはクリアしますがウォッチドッグ機能は、停止されません。

■ ウォッチドッグタイマのクリア

ウォッチドッグタイマは、WTE ビットへの書込みによるクリアのほかに、リセットの発生、スリープモード、ストップモードへの遷移、ホールドアクノリッジ信号によってクリアされます。

第10章

16 ビット入出力タイマ

16 ビット入出力タイマの機能と動作について説明します。

- 10.1 16 ビット入出力タイマの概要
- 10.2 16 ビット入出力タイマのブロックダイアグラム
- 10.3 16 ビット入出力タイマのレジスタ
- 10.4 16 ビットフリーランタイマの動作
- 10.5 16 ビットアウトプットコンペアの動作
- 10.6 16 ビットインプットキャプチャの動作

10.1 16 ビット入出力タイマの概要

16 ビット入出力タイマは、16 ビットフリーランタイム 1 本、アウトプットコンペア 4 本およびインプットキャプチャ 4 本のモジュールから構成されています。

本機能を使用すると、16 ビットフリーランタイムをベースに 2 本の独立した波形出力が可能であり、入力パルス幅測定、外部クロック周期の測定が可能となります。

■ 16 ビットフリーランタイム (× 1)

16 ビットフリーランタイムは 16 ビットのアップカウンタ、コントロールレジスタ、およびプリスケアラより構成されています。本タイマカウンタ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) として使用されます。

- カウンタ動作クロック (4 種類から選択可能)

内部クロック 4 種類 : $\phi/4$, $\phi/16$, $\phi/64$, $\phi/256$

ϕ : マシンクロック

- 割込み

割込みはカウンタ値のオーバーフロー、コンペアレジスタ 0 とのコンペアマッチにより発生することができます (コンペアマッチはモード設定が必要です)。

- カウンタ値

カウンタ値はリセット、ソフトクリア、コンペアレジスタ 0 とのコンペアマッチにより "0000_H" に初期化することができます。

■ アウトプットコンペア (× 4)

アウトプットコンペアは 2 本の 16 ビットコンペアレジスタ、コンペア出力用ラッチ、コントロールレジスタより構成されています。16 ビットフリーランタイム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共に割込みを発生することができます。

- 2 本のコンペアレジスタを独立して動作させる。
 - 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
- 2 本のコンペアレジスタをペアにして出力端子を制御することができます。
 - コンペアレジスタ 2 本を使用して出力端子を反転する。
- 出力端子の初期値を設定することが可能です。
- 割込みはコンペア一致により発生可能です。

■ インプットキャプチャ (× 4)

インプットキャプチャは独立した 4 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により構成されています。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 16 ビットフリーランタイム値をキャプチャレジスタに保持し、同時に割込みを発生することができます。

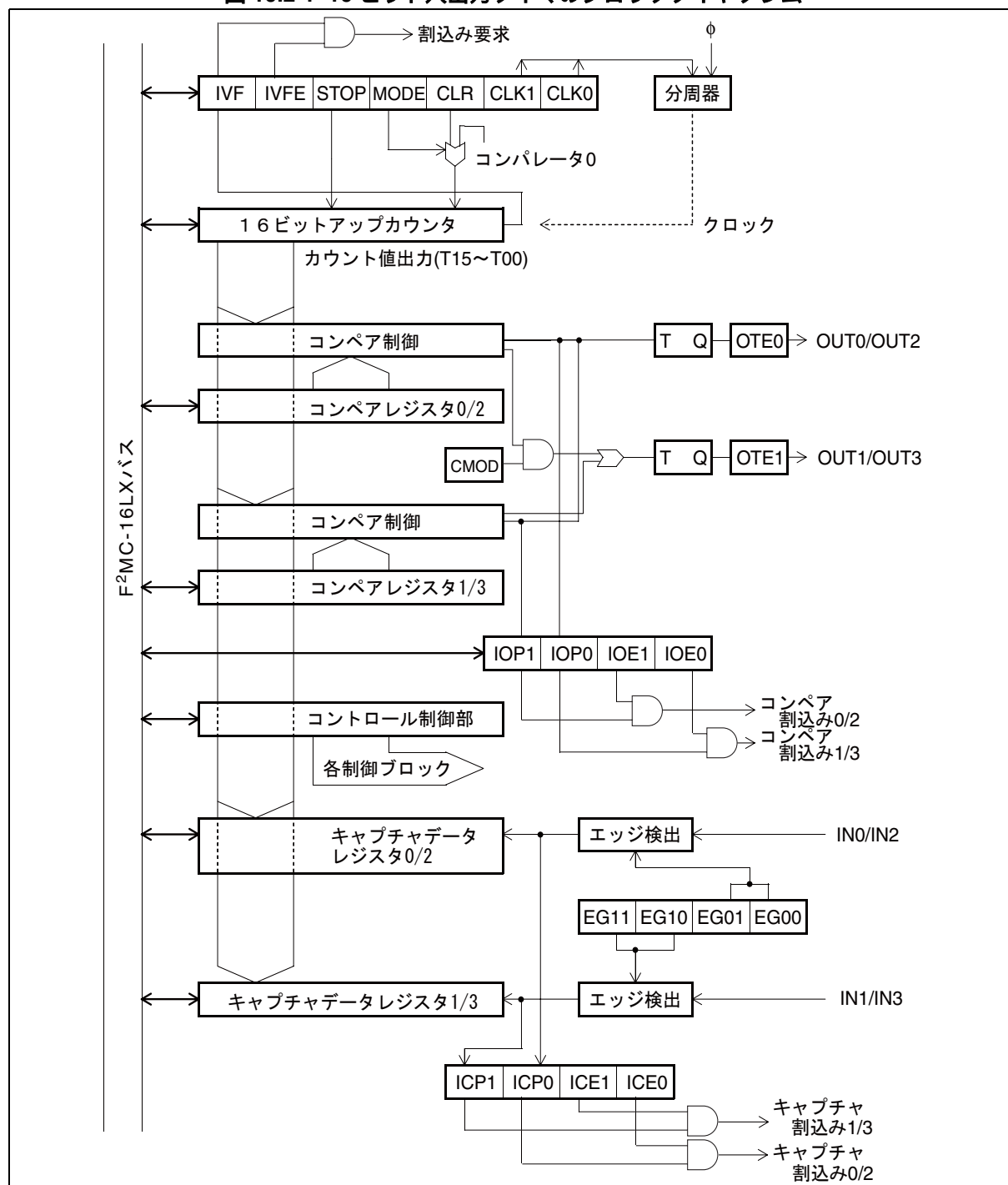
- 外部入力信号のエッジを選択可能です。
 - 立上りエッジ、立下りエッジ、両エッジから選択可能
- 4 本のインプットキャプチャは独立して動作可能です。
- 割込みは外部入力信号の有効エッジにより発生可能です。
 - インプットキャプチャの割込みにより拡張インテリジェント I/O サービスを起動できます。

10.2 16 ビット入出力タイマのブロックダイアグラム

図 10.2-1 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。

■ 16 ビット入出力タイマのブロックダイアグラム

図 10.2-1 16 ビット入出力タイマのブロックダイアグラム



10.3 16 ビット入出力タイマのレジスタ

16 ビット入出力タイマのレジスタには、次の 6 つがあります。

- タイマデータレジスタ (TCDT)
- タイマコントロールステータスレジスタ (TCCS)
- コンペアレジスタ (OCCP0/OCCP1)
- コンペアコントロールステータスレジスタ (OCS0 ~ OCS3)
- インプットキャプチャレジスタ (IPCP0 ~ IPCP3)
- コントロールステータスレジスタ (ICS01, ICS23)

■ 16 ビット入出力タイマのレジスタ

図 10.3-1 16 ビット入出力タイマのレジスタ

タイマデータレジスタ上位	
bit 15 14 13 12 11 10 9 8	
アドレス:00006D _H	T15 T14 T13 T12 T11 T10 T09 T08
リード/ライト⇒	(R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W)
初期値⇒	(0) (0) (0) (0) (0) (0) (0) (0)
タイマデータレジスタ下位	
bit 7 6 5 4 3 2 1 0	
アドレス:00006C _H	T07 T06 T05 T04 T03 T02 T01 T00
リード/ライト⇒	(R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W)
初期値⇒	(0) (0) (0) (0) (0) (0) (0) (0)
タイマコントロールステータスレジスタ	
bit 7 6 5 4 3 2 1 0	
アドレス:00006E _H	予約 IVF IVFE STOP MODE CLR CLK1 CLK0
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) (0)
コンペアレジスタ上位	
bit 15 14 13 12 11 10 9 8	
アドレス: ch.0 000071 _H ch.1 000073 _H ch.2 000075 _H ch.3 000077 _H	C15 C14 C13 C12 C11 C10 C09 C08
リード/ライト⇒	(R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W)
初期値⇒	(X) (X) (X) (X) (X) (X) (X) (X)
コンペアレジスタ下位	
bit 7 6 5 4 3 2 1 0	
アドレス: ch.0 000070 _H ch.1 000072 _H ch.2 000074 _H ch.3 000076 _H	C07 C06 C05 C04 C03 C02 C01 C00
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)

(続く)

(続き)

コンペアコントロールステータスレジスタ1/3

		bit 15	14	13	12	11	10	9	8	
アドレス: ch.1 000079 _H ch.3 00007B _H		—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	OCS1/OCS3
リード/ライト⇒	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(-)	(-)	(-)	(0)	(0)	(0)	(0)	(0)	(0)	

コンペアコントロールステータスレジスタ0/2

		bit	7	6	5	4	3	2	1	0	
アドレス:ch.0 000078 _H ch.2 00007A _H			IOP1	IOP0	IOE1	IOE0	—	—	CST1	CST0	OCS0/OCS2
リード/ライト⇒			(R/W)	(R/W)	(R/W)	(R/W)	(-)	(-)	(R/W)	(R/W)	
初期値⇒			(0)	(0)	(0)	(0)	(-)	(-)	(0)	(0)	

インプットキャプチャデータレジスタ上位

アドレス:ch.0 000063 _H		bit 15		14		13		12		11		10		9		8		IPCO0/1/2/3
ch.1 000065 _H																		
ch.2 000067 _H		CP15		CP14		CP13		CP12		CP11		CP10		CP09		CP08		
ch.3 000069 _H																		
リード/ライト⇒		(R)		(R)		(R)		(R)		(R)		(R)		(R)		(R)		
初期値⇒		(X)		(X)		(X)		(X)		(X)		(X)		(X)		(X)		

インプットキャプチャデータレジスタ下位

		bit 7 6 5 4 3 2 1 0																
アドレス: ch.0 000062 _H		<table border="1"><tr><td>CP07</td><td>CP06</td><td>CP05</td><td>CP04</td><td>CP03</td><td>CP02</td><td>CP01</td><td>CP00</td></tr></table>								CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	IPCO0/1/2/3
CP07	CP06									CP05	CP04	CP03	CP02	CP01	CP00			
ch.1 000064 _H																		
ch.2 000066 _H																		
ch.3 000068 _H																		
リード/ライト⇒		(R)	(R)	(R)	(R)	(R)	(R)	(R)										
初期値⇒		(X)	(X)	(X)	(X)	(X)	(X)	(X)										

コントロールステータスレジスタ上位

bit 15 14 13 12 11 10 9 8								ICS23
アドレス: 00006B _H								
	ICP3	ICP2	ICE3	ICE2	EG31	EG30	EG21	EG20
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)

コントロールステータスレジスタ下位

bit 7 6 5 4 3 2 1 0									
アドレス: 00006A _H	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	ICS01
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)	

10.3.1 16 ビットフリーランタイム

16 ビットフリーランタイムのレジスタには、次の 2 つがあります。

- データレジスタ (TCDT)
- コントロールステータスレジスタ (ICCS)

■ データレジスタ

データレジスタは、16 ビットフリーランタイムのカウント値を読み出すことができるレジスタです。カウンタ値は、リセット時に "0000_H" にクリアされます。このレジスタに書き込むことで、タイマ値を設定できますが、必ず停止 (STOP=1) 状態で行ってください。

16 ビットフリーランタイム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
- コントロールステータスレジスタのクリアビット (CLR) による初期化
- アウトプットコンペアのコンペアレジスタ 0 とタイマカウンタ値の一致による初期化 (モードの設定が必要です)

図 10.3-2 データレジスタ (TCDT)

タイマデータレジスタ上位									
	bit 15	14	13	12	11	10	9	8	
アドレス:00006D _H	T15	T14	T13	T12	T11	T10	T09	T08	TCDT
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
タイマデータレジスタ下位									
	bit 7	6	5	4	3	2	1	0	
アドレス:00006C _H	T07	T06	T05	T04	T03	T02	T01	T00	TCDT
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)	

< 注意事項 >

本レジスタはワードアクセスしてください。

■ コントロールステータスレジスタ (TCCS)

図 10.3-3 コントロールステータスレジスタ (TCCS)

タイマコントロールステータスレジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:00006E _H	予約	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0
リード/ライト⇒ 初期値⇒	(-) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)
TCCS								

[bit7] 予約ビット

bit7 は、予約ビットです。TCCS を設定するときは、必ず "0" に設定してください。

[bit6] IVF

IVF は、16 ビットフリーランタイムの割込み要求フラグです。

16 ビットフリーランタイムがオーバーフローを起こしたときまたはモード設定によりコンペアレジスタ0とコンペアマッチしてカウンタクリアされたときに本ビットは "1" にセットされます。このとき、IVFE 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" 書込みによりクリアされます。"1" 書込みは意味を持ちません。リードモディファイ系命令では "1" が読めます。

表 10.3-1 IVF (割込み要求フラグ) の機能

IVF	機能
0	割込み要求なし (初期値)
1	割込み要求あり

[bit5] IVFE

IVFE は、16 ビットフリーランタイムの割込み許可ビットです。

本ビットが "1" のとき、IVF ビットが "1" にセットされると割込みが発生します。

表 10.3-2 IVFE (割込み許可ビット) の機能

IVFE	機能
0	割込み禁止 (初期値)
1	割込み許可

[bit4] STOP

STOP は、16 ビットフリーランタイマのカウントを停止するためのビットです。"1" 書込み時にタイマのカウント停止。"0" 書込み時にタイマのカウントを開始。

表 10.3-3 STOP（カウントを停止ビット）の機能

STOP	機能
0	カウント許可（動作）（初期値）
1	カウント禁止（停止）

16 ビット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。

[bit3] MODE

MODE は、16 ビットフリーランタイマの初期化条件を設定するビットです。"0" のときはリセットと CLR ビットでカウンタ値を初期化可能です。"1" のときはリセットと CLR ビットのほかにアウトプットコンペアのコンペアレジスタ 0 の値との一致によりカウンタ値を初期化することができます。

表 10.3-4 MODE（初期化条件設定ビット）の機能

MODE	機能
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、コンペアレジスタ 0 による初期化

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。

[bit2] CLR

CLR は、動作中の 16 ビットフリーランタイマ値を "0000_H" に初期化するビットです。"1" を書込み時にカウンタ値を "0000_H" に初期化します。"0" を書き込んでも意味を持ちません。リード値は常に "0" です。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。"1" を書き込んだあと、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。

表 10.3-5 CLR（初期化ビット）の機能

CLR	機能
0	意味を持ちません（初期値）
1	カウンタ値を "0000 _H " に初期化します

タイマ停止中に初期化する場合はデータレジスタに "0000_H" を書き込んでください。

[bit1, bit0] CLK1, CLK0

CLK1, CLK0 は、16 ビットフリーランタイマのカウントクロックを選択するビットです。本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。

表 10.3-6 CLK1, CLK0 (カウントクロック選択ビット)

CLK1	CLK0	カウントクロック	$\phi=16\text{MHz}$	$\phi=8\text{MHz}$	$\phi=4\text{MHz}$	$\phi=1\text{MHz}$
0	0	$\phi/4$	$0.25\mu\text{s}$	$0.5\mu\text{s}$	$1\mu\text{s}$	$4\mu\text{s}$
0	1	$\phi/16$	$1\mu\text{s}$	$2\mu\text{s}$	$4\mu\text{s}$	$16\mu\text{s}$
1	0	$\phi/64$	$4\mu\text{s}$	$8\mu\text{s}$	$16\mu\text{s}$	$64\mu\text{s}$
1	1	$\phi/256$	$16\mu\text{s}$	$32\mu\text{s}$	$64\mu\text{s}$	$256\mu\text{s}$

ϕ = マシンクロック

10.3.2 アウトプットコンペア

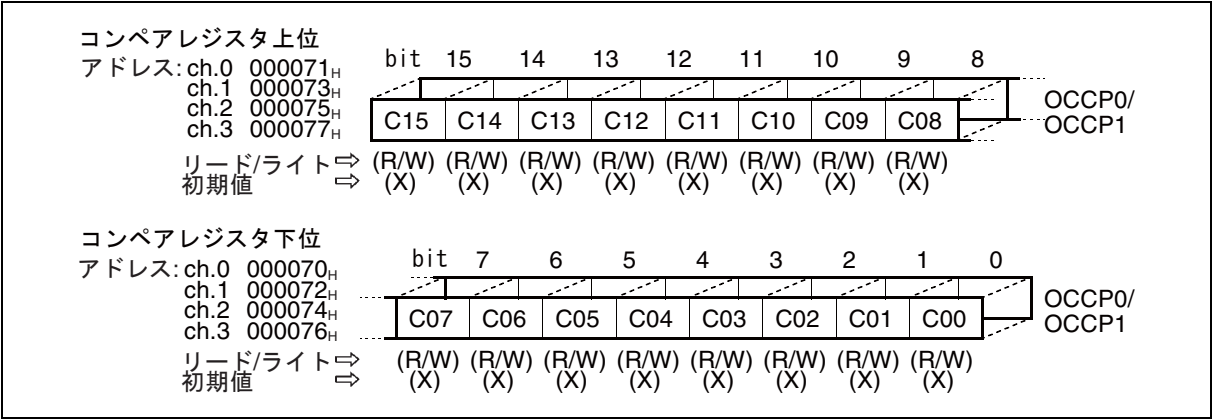
アウトプットコンペアには、次の 2 つのレジスタがあります。

- コンペアレジスタ
- コントロールステータスレジスタ

ここでは、ch.0, ch.1 について説明しますので、ch.2, ch.3 の場合は、ch.0 ch.2, ch.1 ch.3 と読み換えてください。

■ コンペアレジスタ（OCCP0/OCCP1）

図 10.3-4 コンペアレジスタ



コンペアレジスタは、16 ビットフリーランタイムと比較する 16 ビット長のコンペアレジスタです。レジスタ値は初期値不定ですので設定してから起動許可してください。本レジスタ値と 16 ビットフリーランタイム値が一致したとき、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をセットします。また、出力許可をしている場合はコンペアレジスタに対応した出力レベルを反転します。

< 注意事項 >

本レジスタはワードアクセスしてください。

■ コントロールステータスレジスタ (OCS0 ~ OCS2)

図 10.3-5 コントロールステータスレジスタ

コンペアコントロールステータスレジスタ 1/3									
bit 15 14 13 12 11 10 9 8									
アドレス: ch.1 000079 _H	—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	OCS1/OCS3
ch.3 00007B _H									
リード/ライト ⇒	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(-)	(-)	(0)	(0)	(0)	(0)	(0)	
コンペアコントロールステータスレジスタ 0/2									
bit 7 6 5 4 3 2 1 0									
アドレス: ch.0 000078 _H	IOP1	IOP0	IOE1	IOE0	—	—	CST1	CST0	OCS0/OCS2
ch.2 00007A _H									
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(-)	(-)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(-)	(-)	(0)	(0)	

[bit12] CMOD

CMOD では、端子出力を許可した場合 (OTE1=1 または OTE0=1) のコンペア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)

CMOD=0 のとき (初期値) は、コンペアレジスタに対応した端子の出力レベルを反転します。

- OUT0: コンペアレジスタ 0 の一致によりレベルを反転します。
- OUT1: コンペアレジスタ 1 の一致によりレベルを反転します。

● CMOD=1 のとき

CMOD=1 のときは、コンペアレジスタ 0 に対応した端子 (OUT0) は、CMOD=0 時と同じく出力レベルを反転しますが、コンペアレジスタ 1 に対応した端子 (OUT1) の出力レベルは、コンペアレジスタ 0 の一致とコンペアレジスタ 1 の一致の両方で出力レベルを反転します。コンペアレジスタ 0 と 1 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: コンペアレジスタ 0 の一致によりレベルを反転します。
- OUT1: コンペアレジスタ 0 と 1 の一致によりレベルを反転します。

[bit11, bit10] OTE1, OTE0

OTE1, OTE0 は、アウトプットコンペアの端子出力を許可するビットです。

表 10.3-7 OTE1, OTE0 (端子出力許可ビット) の機能

OTE1, OTE0	機能
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になります。

OTE1: アウトプットコンペア 1 に対応、OTE0: アウトプットコンペア 0 に対応

[bit9, bit8] OTD1, OTD0

OTD1, OTD0 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" となります。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出力値が読めます。

表 10.3-8 OTD1, OTD0 (端子出力レベル変更ビット) の機能

OTD1, OTD0	機能
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 に対応, OTD0: アウトプットコンペア 0 に対応

[bit7, bit6] IOP1, IOP0

IOP1, IOP0 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 16 ビット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 (IOE1, IOE0) が許可されているときに本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" 書込みによりクリアされ, "1" 書込みでは意味をもちません。リードモディファイ系命令では "1" が読めます。

表 10.3-9 IOP1, IOP0 (アウトプットコンペア割込みビット) の機能

IOP1, IOP0	機能
0	コンペアー致なし [初期値]
1	コンペアー致あり

IOP1: アウトプットコンペア 1 に対応, IOP0: アウトプットコンペア 0 に対応

[bit5, bit4] IOE1, IOE0

IOE1, IOE0 は、アウトプットコンペアの割込み許可ビットです。本ビットが "1" のとき割込みフラグ (IOP0, IOP1) 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。

表 10.3-10 IOE1, IOE0 (アウトプットコンペアの割込み許可ビット) の機能

IOE1, IOE0	機能
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

IOE1: アウトプットコンペア 1 に対応, IOE0: アウトプットコンペア 0 に対応

[bit1, bit0] CST1, CST0

CST1, CST0 は、16 ビットフリーランタイムとの一致動作を許可するビットです。

表 10.3-11 CST1, CST0 (16 ビットフリーランタイムとの一致動作許可ビット)

CST1, CST0	設定
0	コンペア動作禁止 [初期値]
1	コンペア動作許可

- ・コンペア動作許可をする前にコンペアレジスタ値を設定してください。
CST1: アウトプットコンペア 1 に対応, CST0: アウトプットコンペア 0 に対応

< 注意事項 >

アウトプットコンペアは、16 ビットフリーランタイムのクロックと同期させているため、16 ビット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。

10.3.3 インプットキャプチャ

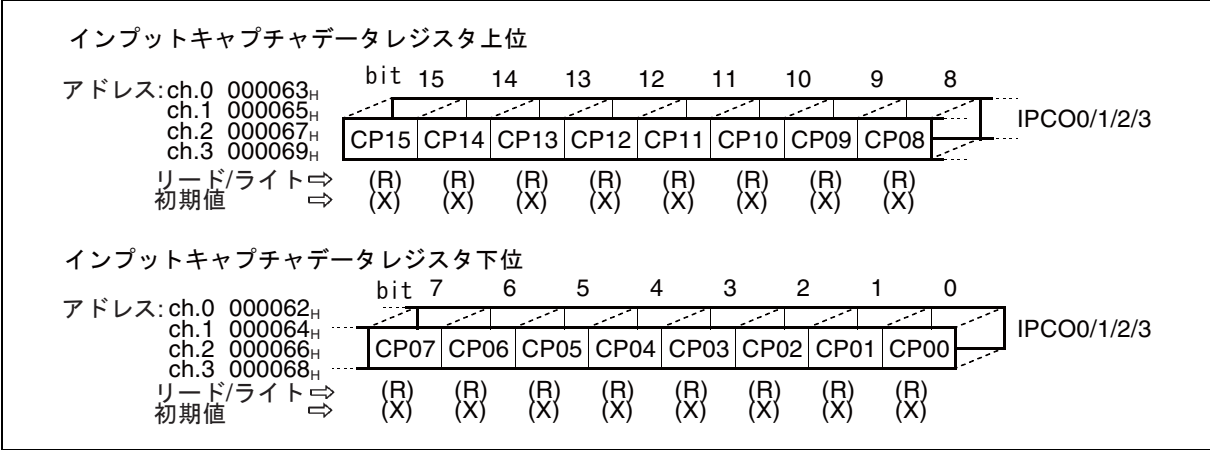
インプットキャプチャには、次の 2 つのレジスタがあります。

- インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3)
- コントロールステータスレジスタ (ICS23/ICS01)

■ インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3)

インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3) 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、16 ビットフリーランタイム値を保持するレジスタです。

図 10.3-6 インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3)



< 注意事項 >

本レジスタはワードアクセスしてください。書込みはできません。

■ コントロールステータスレジスタ (ICS23/ICS01)

図 10.3-7 コントロールステータスレジスタ (ICS23/ICS01)

コントロールステータスレジスタ上位							
	bit 15	14	13	12	11	10	9 8
アドレス: 00006B _H	ICP3	ICP2	ICE3	ICE2	EG31	EG30	EG21 EG20
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)
ICS23							
コントロールステータスレジスタ下位							
	bit 7	6	5	4	3	2	1 0
アドレス: 00006A _H	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01 EG00
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)
ICS01							

< 注意事項 >

本レジスタはバイトアクセスしてください。

[bit15, bit14, bit7, bit6] ICP_x (x: ch 番号)

ICP_x は、インプ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本ビットを "1" にセットします。割込み許可ビット (ICE0, ICE1) 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することができます。

本ビットは "0" 書込みによりクリアされます。"1" 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" が読めます。

表 10.3-12 ICP_x (インプットキャプチャ割込みフラグ) の機能

ICP _x	機能
0	有効エッジ検出なし (初期値)
1	有効エッジ検出あり

[bit13, bit12, bit5, bit4] ICE_x (x: ch 番号)

ICE_x は、インプットキャプチャ割込み許可ビットです。本ビットが "1" のとき割込みフラグ (ICP0, ICP1) がセットされるとインプットキャプチャ割込みが発生します。

表 10.3-13 ICE_x (インプットキャプチャ割込み許可ビット) の機能

ICE _x	機能
0	割込み禁止 (初期値)
1	割込み許可

[bit11 ~ bit8, bit3 ~ bit0] EGx1, EGx0 (x:ch 番号)

EGx1, EGx0 ビットは , 外部入力の有効エッジ極性を指定します。インプットキャプチャ動作許可も兼用しています。

表 10.3-14 EGx1, EGx0 (外部入力の有効エッジ極性指定ビット) の機能

EGx1	EGx0	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

10.4 16 ビットフリーランタイマの動作

16 ビットフリーランタイマはリセット解除後にカウンタ値 "0000_H" よりカウントを開始します。このカウンタ値が 16 ビットアウトプットコンペアと 16 ビットインプットキャプチャの基準時間となります。

■ 16 ビットフリーランタイマの動作

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レジスタ 0 値とコンペアマッチしたとき(モード設定が必要)
- 動作中に TCCS レジスタの CLR ビットに "1" を書き込んだとき。
- 停止中に TCDDT レジスタに "0000_H" 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、コンペアレジスタ 0 値とコンペアマッチしてカウンタクリアされたとき発生することができます(コンペアマッチ割込みはモード設定が必要です)。

図 10.4-1 オーバフローによるカウンタクリア

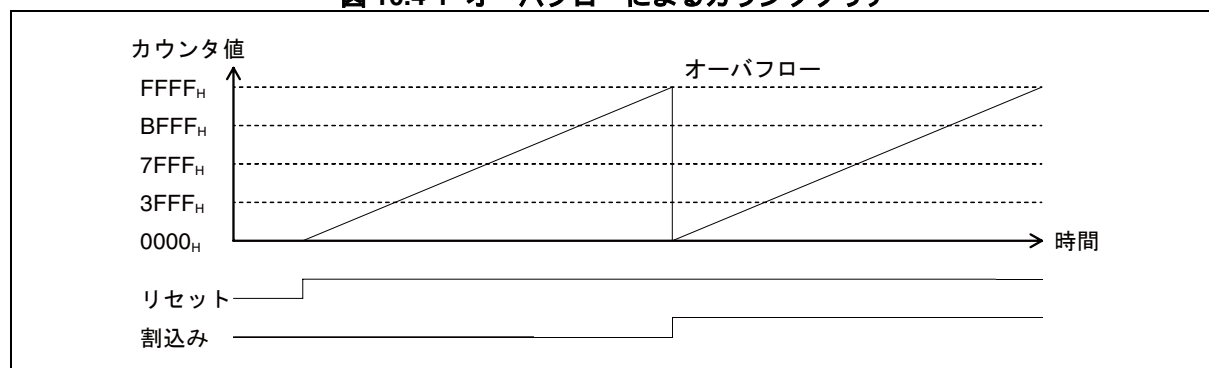
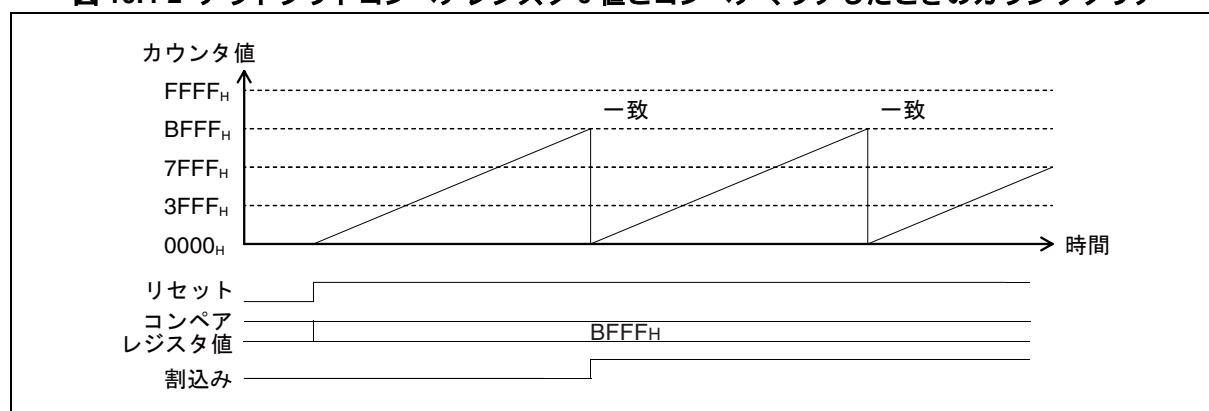


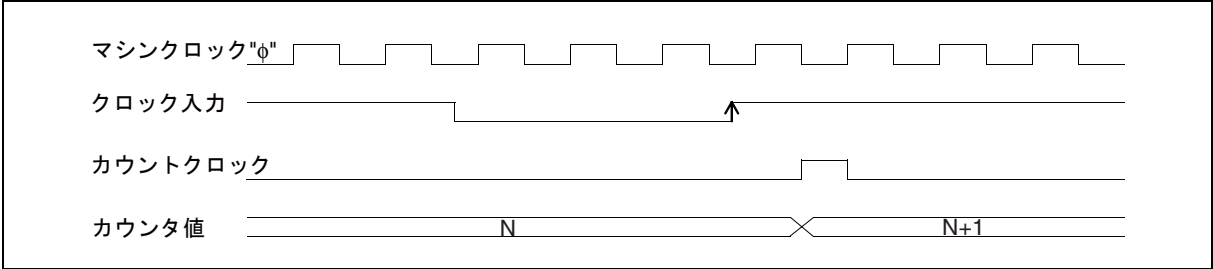
図 10.4-2 アウトプットコンペアレジスタ 0 値とコンペアマッチしたときのカウンタクリア



■ 16 ビットフリーランタイマのカウンタタイミング

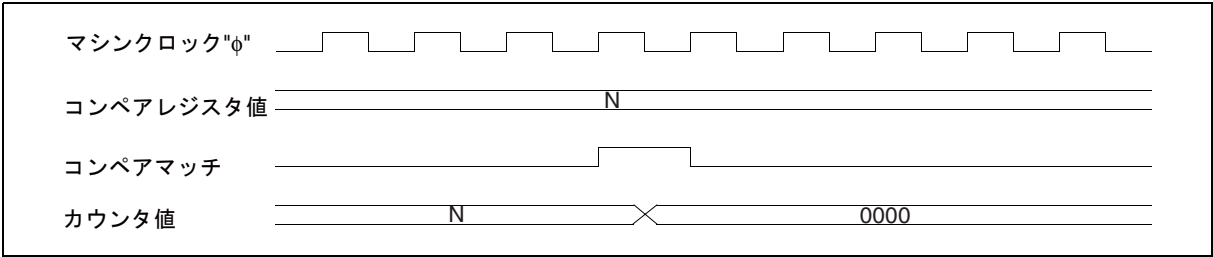
16 ビットフリーランタイマは、入力されたクロックによりカウントアップされます。

図 10.4-3 フリーランタイマのカウンタタイミング



カウンタのクリアはリセット、ソフト、コンペアレジスタ 0 との一致で行うことができます。リセットとソフトでのカウンタクリアはクリア発生とともに行われますが、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タタイミングに同期して行われます。

図 10.4-4 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）



10.5 16 ビットアウトプットコンペアの動作

16 ビット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 16 ビットフリーランタイムとの値を比較して一致したら割込み要求フラグをセットするとともに、出力レベルを反転することができます。

■ 16 ビットアウトプットコンペアの動作

図 10.5-1 コンペアレジスタ 0, 1 を使用したときの出力波形例 (CMOD=0 のとき)

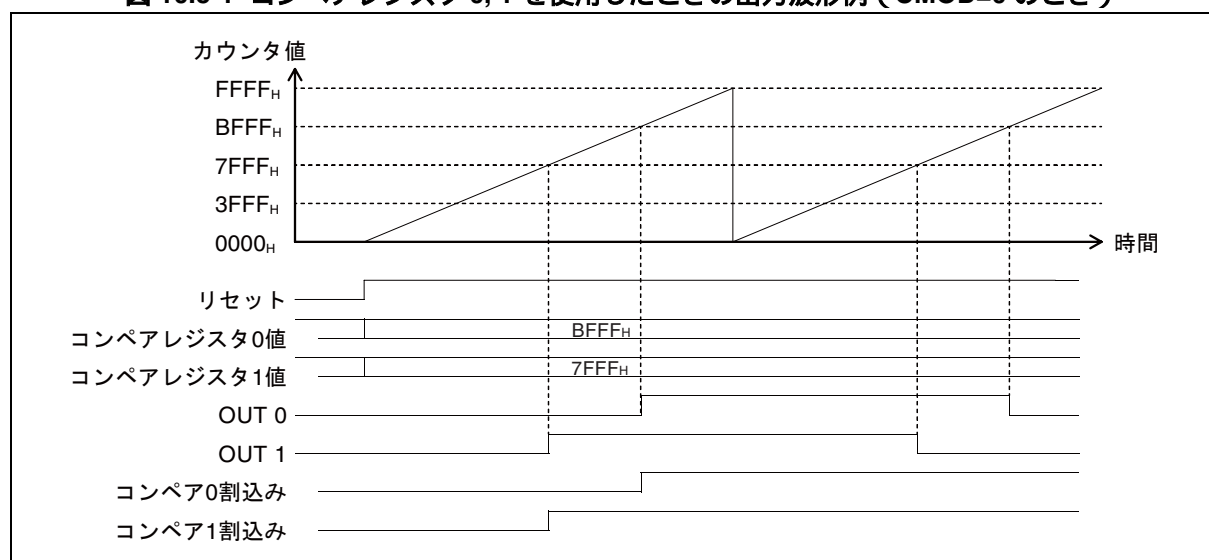
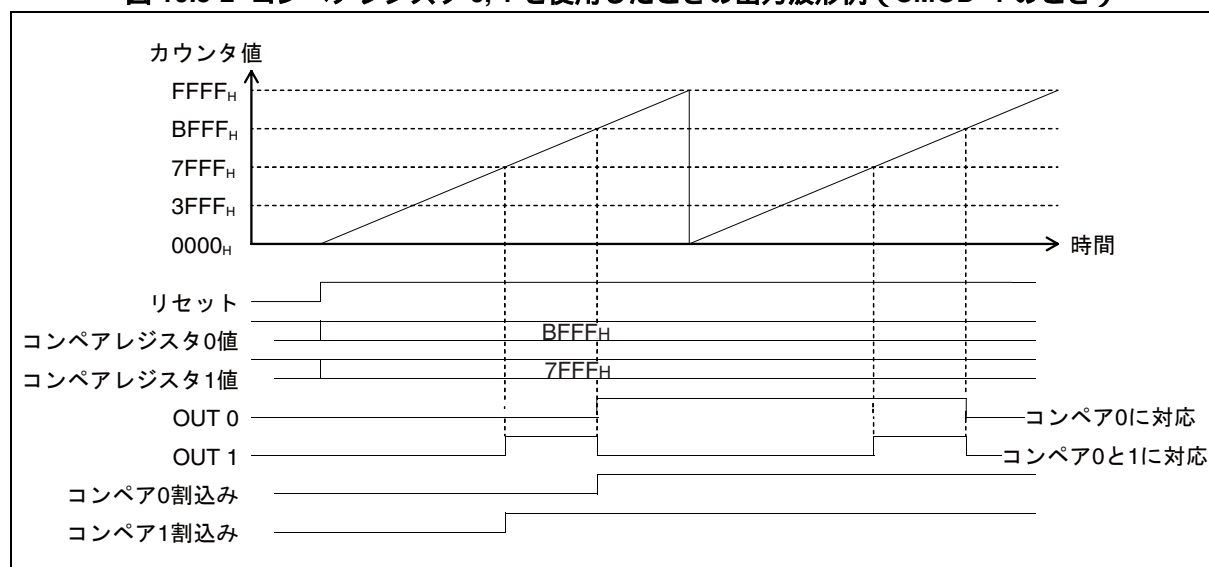


図 10.5-2 コンペアレジスタ 0, 1 を使用したときの出力波形例 (CMOD=1 のとき)



■ 16 ビット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、フリーランタイマ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を発生することができます。

コンペアマッチ時の出力反転タイミングはカウンタのカウントタイミングに同期して行われます。コンペアレジスタ書き換え時のカウンタ値とは、コンペアしません。

図 10.5-3 コンペアレジスタ書き換え時のコンペア動作

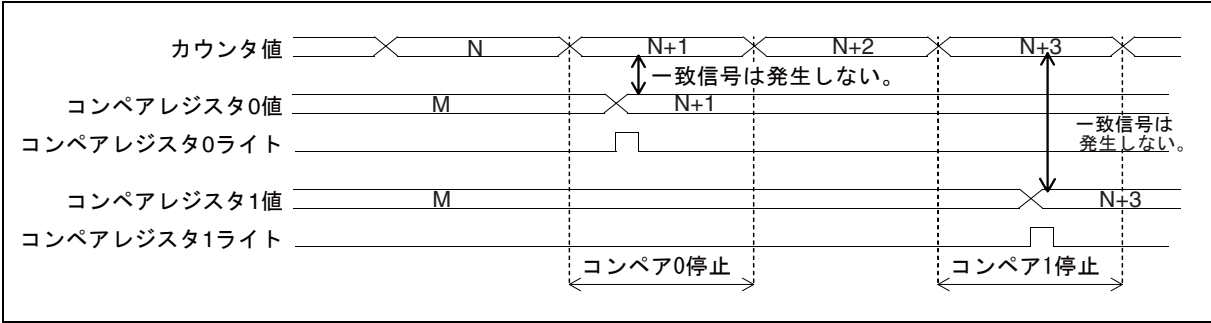


図 10.5-4 割込みタイミング

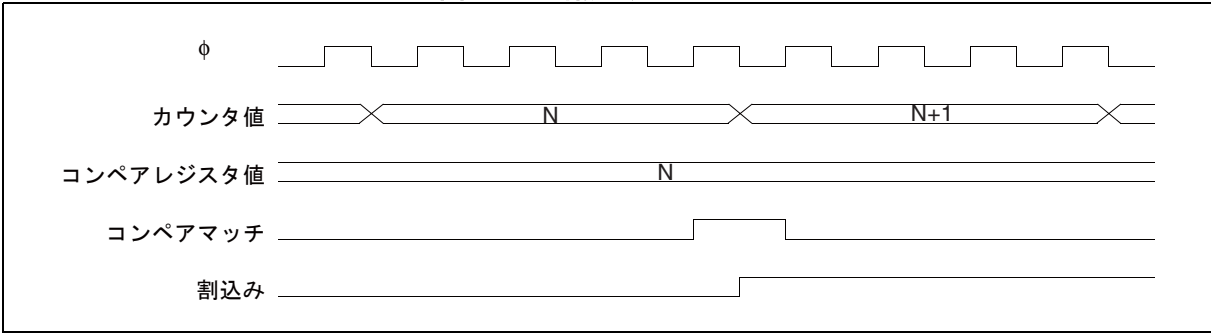
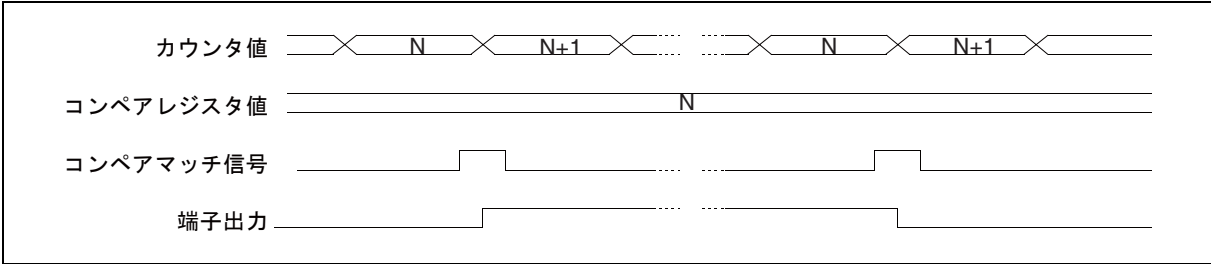


図 10.5-5 出力端子変化タイミング



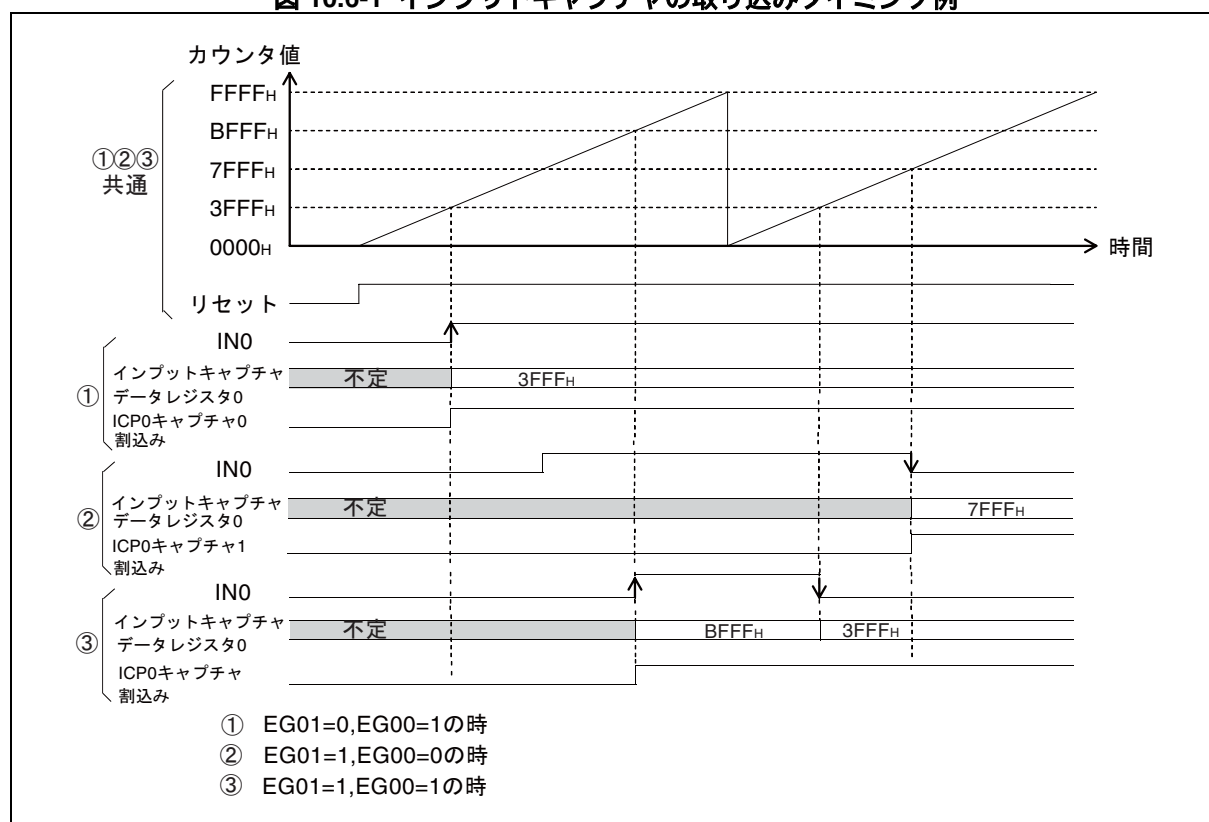
10.6 16 ビットインプットキャプチャの動作

16 ビットインプットキャプチャは、設定された有効エッジを検出すると 16 ビットフリーランタイムの値をキャプチャレジスタに取り込んで割込みを発生することができます。

■ 16 ビットインプットキャプチャの動作

図 10.6-1 では、0ch の場合のインプットキャプチャ取り込みタイミング例を示します。ほかのチャンネルも同様の動作となります。

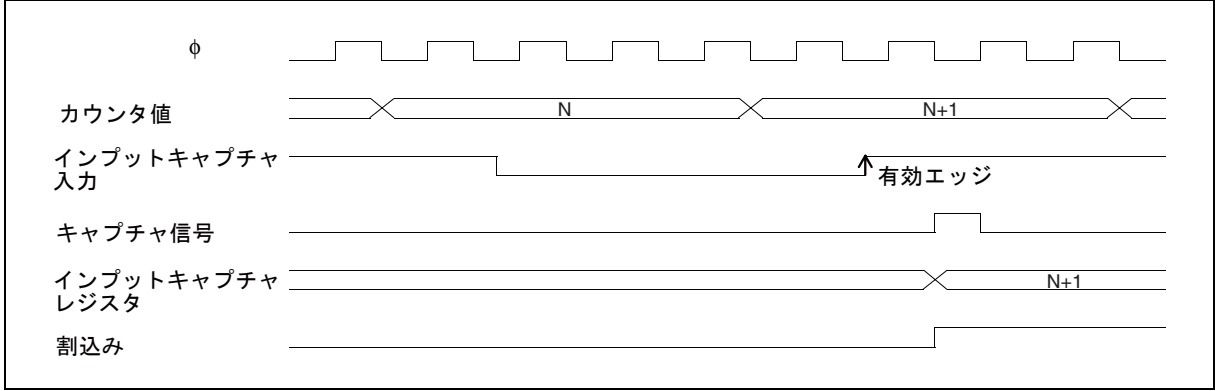
図 10.6-1 インプットキャプチャの取り込みタイミング例



■ インพุットキャプチャ入力タイミング

図 10.6-2 では , EGx1=0, EGx0=1 の場合の , 入力信号に対するキャプチャタイミングを示します。

図 10.6-2 入力信号に対するキャプチャタイミング



第11章

16 ビットリロードタイマ (イベントカウント機能付)

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

- 11.1 16 ビットリロードタイマ (イベントカウント機能付) の概要
- 11.2 16 ビットリロードタイマ (イベントカウント機能付) のレジスタ
- 11.3 クロック動作
- 11.4 アンダフロー動作
- 11.5 入出力端子機能
- 11.6 カウンタの動作状態

11.1 16 ビットリロードタイマ (イベントカウント機能付) の概要

16 ビットリロードタイマ 1 は, 16bit のダウンカウンタ, 16bit のリロードレジスタ, 入力端子 1 本 (TIN), 出力端子 1 本 (TOT), コントロールレジスタで構成されています。入力クロックとして, 内部クロック 3 種類と外部クロックから選択できます。

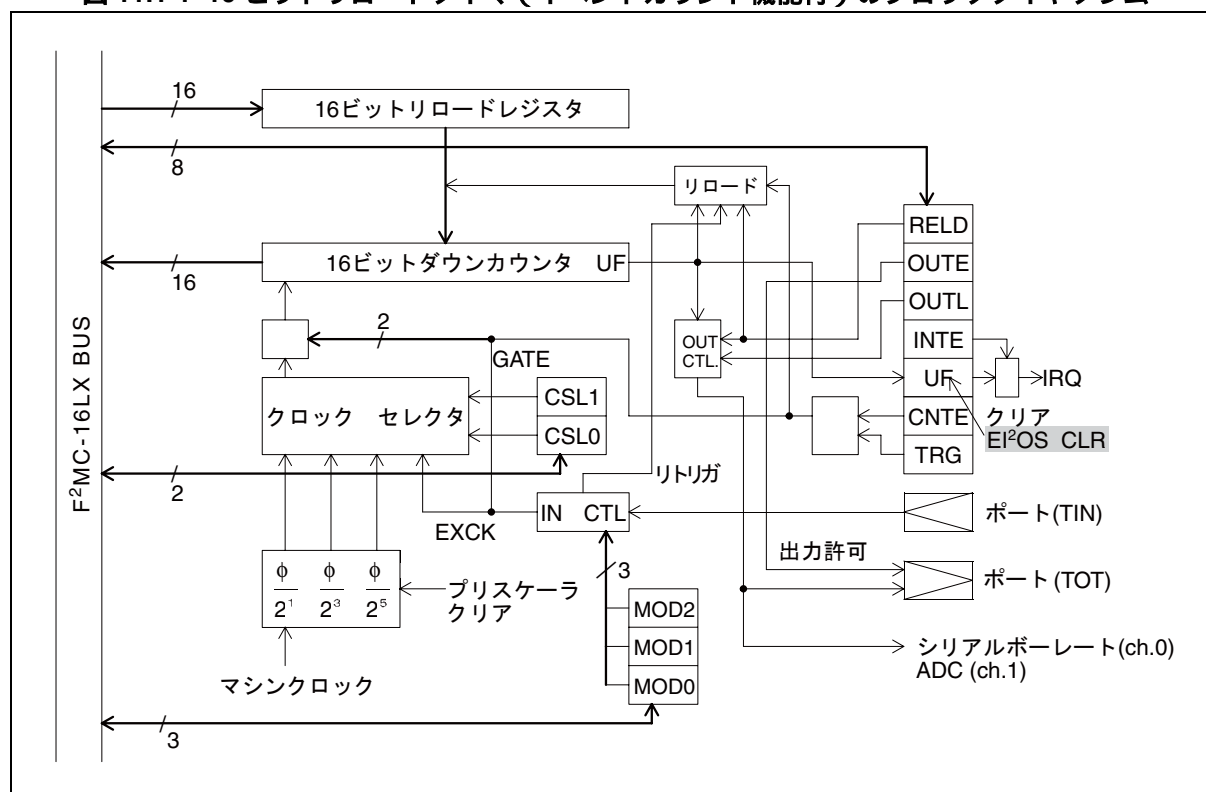
■ 16 ビットリロードタイマ (イベントカウント機能付) の概要

出力端子 (TOT) には, リロードモード時にはトグル出力波形を出力し, ワンショットモード時にはカウント中を示す矩形波を出力します。入力端子 (TIN) は, イベントカウントモード時にイベント入力となり, 内部クロックモード時にはトリガ入力またはゲート入力に使用することができます。

本シリーズでは, 16 ビットリロードタイマを 2ch 内蔵しています。

■ 16 ビットリロードタイマ (イベントカウント機能付) のブロックダイアグラム

図 11.1-1 16 ビットリロードタイマ (イベントカウント機能付) のブロックダイアグラム



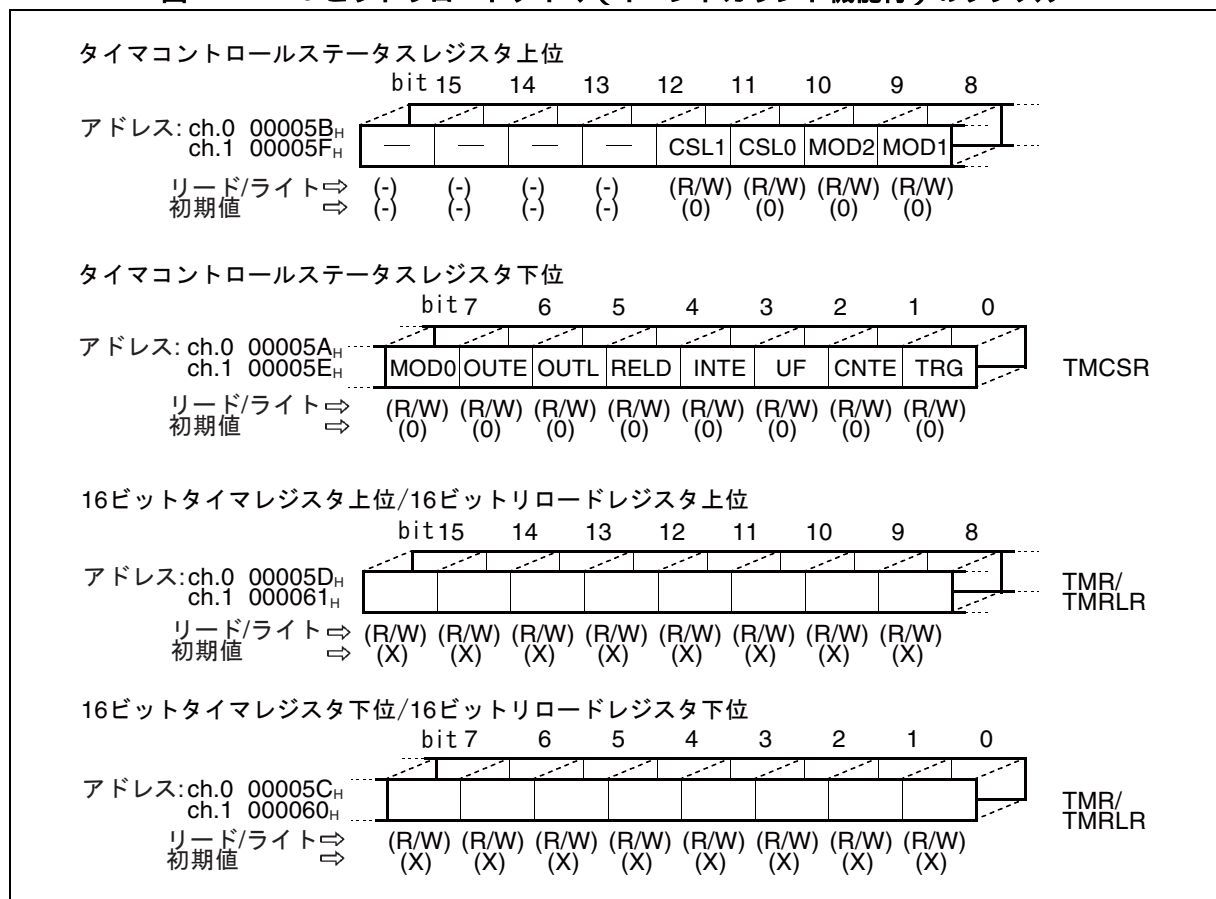
11.2 16 ビットリロードタイマ (イベントカウント機能付) のレジスタ

16 ビットリロードタイマ (イベントカウント機能付) には、次の 4 種類のレジスタがあります。

- タイマコントロールステータスレジスタ上位
- タイマコントロールステータスレジスタ下位
- 16 ビットタイマレジスタ上位 / 16 ビットリロードレジスタ上位
- 16 ビットタイマレジスタ下位 / 16 ビットリロードレジスタ下位

■ 16 ビットリロードタイマ (イベントカウント機能付) のレジスタ

図 11.2-1 16 ビットリロードタイマ (イベントカウント機能付) のレジスタ

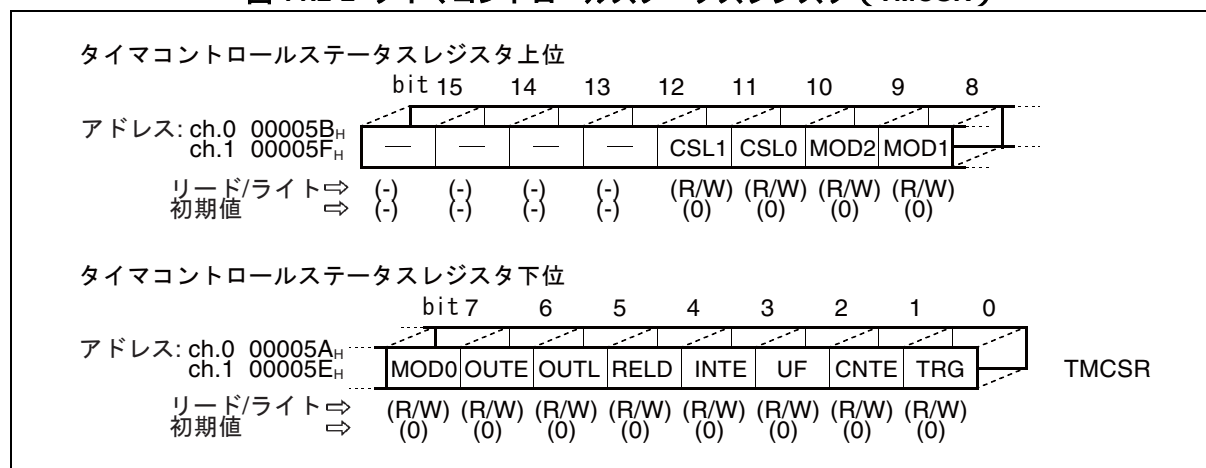


11.2.1 タイマコントロールステータスレジスタ（TMCSR）

タイマコントロールステータスレジスタ（TMCSR）は、16 ビットタイマの動作モードおよび割込みを制御します。

■ タイマコントロールステータスレジスタ（TMCSR）

図 11.2-2 タイマコントロールステータスレジスタ（TMCSR）



< 注意事項 >

UF, CNTE, TRG ビット以外のビットの書換えは、CNTE=0 のときに行ってください。

[bit11, bit10] CSL1, CSL0 (Clock Select0, 1)

CSL1, CSL0 ビットは、カウントクロックセレクトビットです。選択されるクロックソースを、以下に示します。

表 11.2-1 CSL1, CSL0 (カウントクロックセレクトビット) の機能

CSL1	CSL0	クロックソース (マシンサイクル $\phi=16\text{MHz}$)
0	0	$\phi/2^1$ (0.125 μs) [初期値]
0	1	$\phi/2^3$ (0.5 μs)
1	0	$\phi/2^5$ (2.0 μs)
1	1	外部イベントカウントモード

[bit9 ~ bit7] MOD2, MOD1, MOD0

MOD2, MOD1, MOD0 ビットは、動作モードおよび入出力端子の機能を設定するビットです。

MOD2 ビットは、入出力の機能を選択するビットです。"0" の場合、入力端子 (TIN) はトリガ入力端子となり、有効エッジが入力されるとリロードレジスタの内容をカウンタへロードし、カウント動作を継続します。"1" の場合、ゲートカウンタモードになり、入力端子 (TIN) はゲート入力となり、有効レベルが入力されている間のみカウントをします。

MOD1, MOD0 ビットは、各モードにおける端子の機能を設定します。

表 11.2-2 MOD2, MOD1, MOD0（動作モード / 入出力端子機能設定ビット）の機能

モード	MOD2	MOD1	MOD0	入力端子機能	有効エッジ, レベル
内部クロックモード時 (CSL0, CSL1=00, 01, 10)	0	0	0	トリガ禁止	-
	0	0	1	トリガ入力	立上りエッジ
	0	1	0		立下りエッジ
	0	1	1		両エッジ
	1	x	0	ゲート入力	"L" レベル
	1	x	1		"H" レベル
イベントカウントモード時 (CSL0, CSL1=11)	x	0	0	-	-
		0	1	トリガ入力	立上りエッジ
		1	0		立下りエッジ
		1	1		両エッジ

x : 任意の値

[bit6] OUTE

OUTE ビットは、出力許可ビットです。

- "0" のときは、TOT 端子は汎用ポートになります。
- "1" のときは、TOT 端子はタイマ出力端子になります。

[bit5] OUTL

OUTL ビットは、TOT 端子の出力レベルを設定するビットです。

[bit4] RELD（RELoad）

RELD ビットは、リロード動作を許可するビットです。

- "0" のときは単発動作モードとなり、カウンタの値が 0000_H ～ FFFF_H へのアンダフローによりカウント動作を停止します。
- "1" のときはリロードモードになり、カウンタの値が 0000_H ～ FFFF_H へのアンダフローと同時にリロードレジスタの内容をカウンタへロードしてカウント動作を続けます。

表 11.2-3 RELD（リロード動作許可ビット）の機能

OUTE	OUTL	RELD	出力波形
0	×	×	汎用ポート
1	0	0	カウント中 "H" の矩形波
1	0	1	カウントスタート時 "L" のトグル出力
1	1	0	カウント中 "L" の矩形波
1	1	1	カウントスタート時 "H" のトグル出力

×：任意の値

[bit3] INTE（INTerrupt Enable）

INTE ビットは、タイマ割込み要求許可ビットです。

表 11.2-4 INTE（タイマ割込み要求許可ビット）の機能

INTE	機能
0	割込み禁止
1	割込み許可

[bit2] UF（Under Flow）

UF ビットは、タイマ割込み要求フラグです。

カウンタの値が、0000_H ～ FFFF_H へのアンダフローにより "1" にセットされます。

"0" の書込みまたはインテリジェント I/O サービスによってクリアされます。このビットへの "1" 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における読出し時には、"1" が読み出されます。

[bit1] CNTE（CouNT Enable）

CNTE ビットは、タイマのカウントイネーブルビットです。

このビットに "1" を書き込むと、起動トリガ待ち状態になります。"0" 書込みによりカウント動作は停止します。

[bit0] TRG（TRiG）

TRG ビットは、ソフトウェアトリガビットです。

"1"書込みによりソフトウェアトリガがかかり、リロードレジスタの内容をカウンタへロードしてカウント動作を開始します。

"0" 書込み動作は、意味を持ちません。読出し値は、常に "0" です。このレジスタによるトリガ入力、CNTE=1 のときのみ有効となります。CNTE=0 のときには、何にも起こりません。

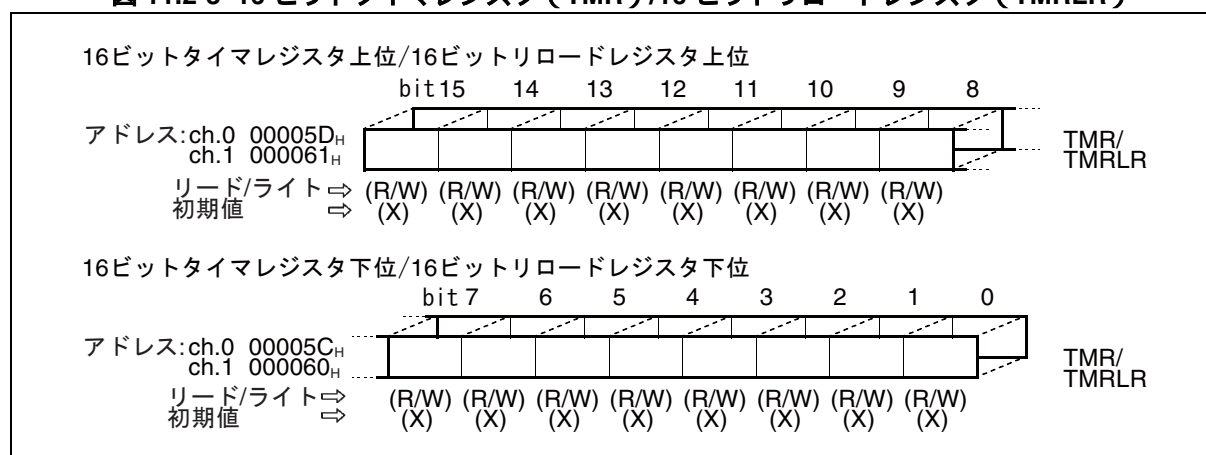
11.2.2 16 ビットタイマレジスタ (TMR) /16 ビットリロードレジスタ (TMRLR)

16 ビットタイマレジスタ (TMR) (読み出し時) は, 16 ビット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。初期値は不定です。

16 ビットリロードレジスタ (TMRLR) (書き込み時) は, 16 ビットリロードレジスタは, カウントの初期値を保持しておくレジスタです。初期値は不定です。

■ 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)

図 11.2-3 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)



< 注意事項 >

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) は, ワードアクセスしてください。

11.3 クロック動作

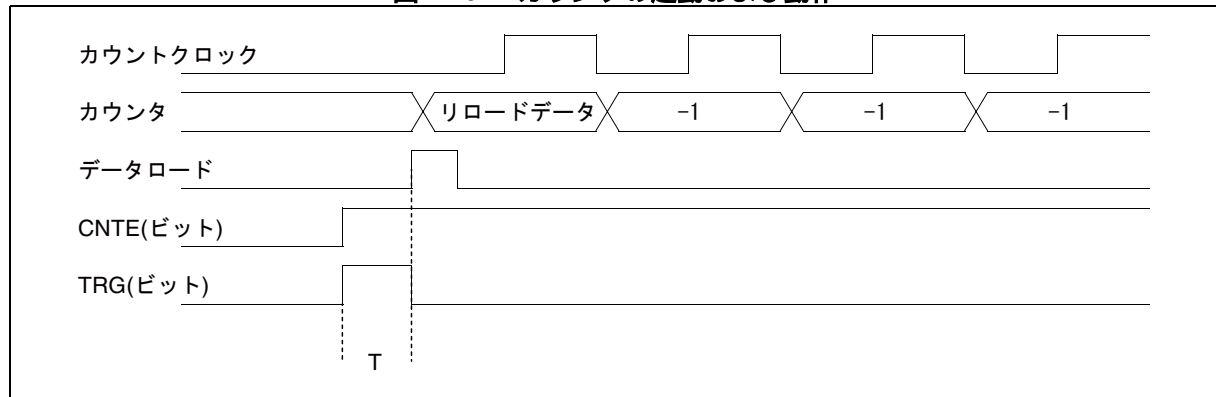
内部クロックの分周クロックでタイマを動作させる場合、クロックソースとしてマシンのクロックの 2^1 , 2^3 , 2^5 分周のクロックから選択することができます。外部入力端子は、レジスタの設定により、トリガ入力またはゲート入力にすることができます。

■ 内部クロック動作

カウント許可と同時にカウント動作を開始したい場合は、コントロールレジスタの CNTE ビットと TRG ビットの両方に "1" を書き込んでください。TRG ビットによるトリガ入力は、タイマが起動状態のとき（CNTE=1）動作モードにかかわらず常に有効です。

カウンタスタートのトリガが入力されてからリロードレジスタのデータがカウンタへロードされるまでに、 T （ T : マシンサイクル）の時間がかかります。

図 11.3-1 カウンタの起動および動作



■ 外部イベントカウント

外部クロックをセレクトすると TIN 端子は、外部イベント入力端子となりレジスタで設定された有効エッジをカウントします。TIN 端子のパルス幅は、 $4 \cdot T$ （ T はマシンサイクル）以上にしてください。

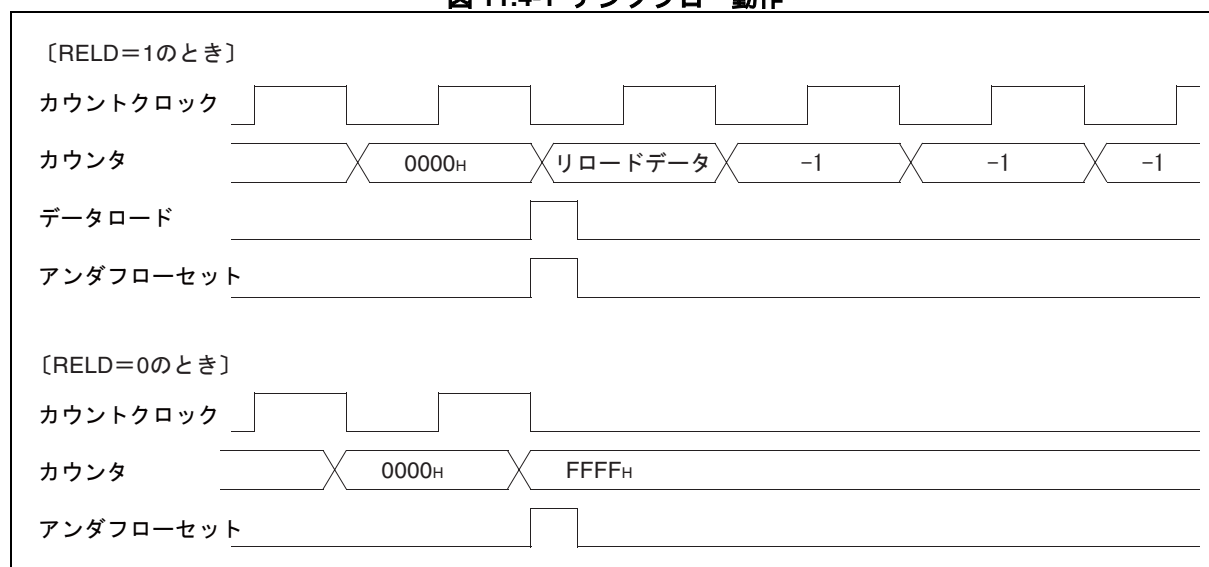
11.4 アンダフロー動作

16 ビットリロードタイマ（イベントカウント機能付）は、カウンタの値が 0000_H から $FFFF_H$ になるときをアンダフローと定義しています。したがって、〔リロードレジスタの設定値 +1〕カウントでアンダフローが発生することになります。

■ アンダフロー動作

アンダフロー発生時、コントロールレジスタの RELD ビットが "1" のときリロードレジスタの内容をカウンタへロードしてカウント動作を継続し、"0" のときカウンタを $FFFF_H$ で停止させます。アンダフローが発生すると、コントロールレジスタの UF ビットをセットし、このときに INTE ビットが "1" であると割り込み要求が発生します。

図 11.4-1 アンダフロー動作



■ 拡張インテリジェント I/O サービス (EI²OS) 機能と割り込み

本タイマは EI²OS 対応の回路を有しています。このため、本タイマのアンダフローにより EI²OS を起動させることができます。本製品では、2 つのタイマすべてで EI²OS を使用することができます。

11.5 入出力端子機能

クロックソースとして、内部クロックを選択した場合、TIN 端子はトリガ入力またはゲート入力として使用することができます。

出力極性はレジスタの OUTL ビットにより設定できます。TOT 端子は、リロードモード時には、アンダフローにより反転するトグル出力として、ワンショットモード時には、カウント中を示すパルス出力として機能します。

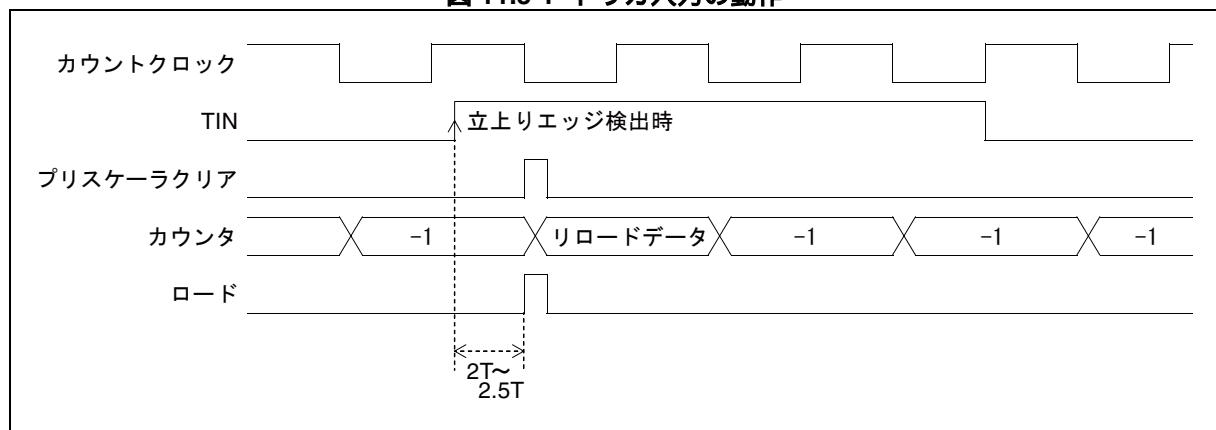
■ 入力端子機能（内部クロックモード時）

クロックソースとして、内部クロックを選択した場合、TIN 端子はトリガ入力またはゲート入力として使用することができます。

トリガ入力として使用した場合、図 11.5-1 に示すように、有効エッジが入力されるとリロードレジスタの内容をカウンタにロードして、内部プリスケータをクリアした後、カウント動作を開始します。

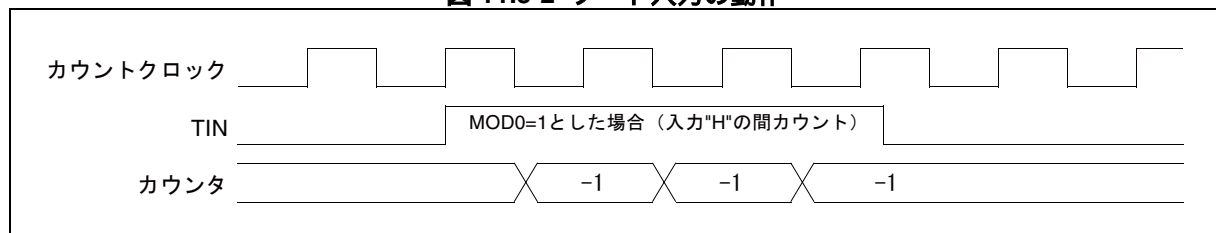
TIN は、 $2 \cdot T$ （ T はマシンサイクル）以上のパルスを入力してください。

図 11.5-1 トリガ入力の動作



ゲート入力として使用する場合、図 11.5-2 に示すように、コントロールレジスタの MOD0 ビットにより設定される有効レベルが、TIN 端子から入力されている間のみカウントをします。このときカウントクロックは、止まらずに動き続けます。ゲートモード時のソフトウェアトリガは、ゲートレベルにかかわらず可能です。TIN 端子のパルス幅は、 $2 \cdot T$ （ T はマシンサイクル）以上にしてください。

図 11.5-2 ゲート入力の動作



■ 出力端子機能

出力極性はレジスタの OUTL ビットにより設定できます。TOT 端子は、リロードモード時には、アンダフローにより反転するトグル出力として、ワンショットモード時には、カウント中を示すパルス出力として機能します。

OUTL が "0" のときトグル出力は、初期値が "0" で、ワンショットパルス出力は、カウント中 "1" を出力します。OUTL を "1" にすると出力波形は、反転します。

図 11.5-3 出力端子機能（1）

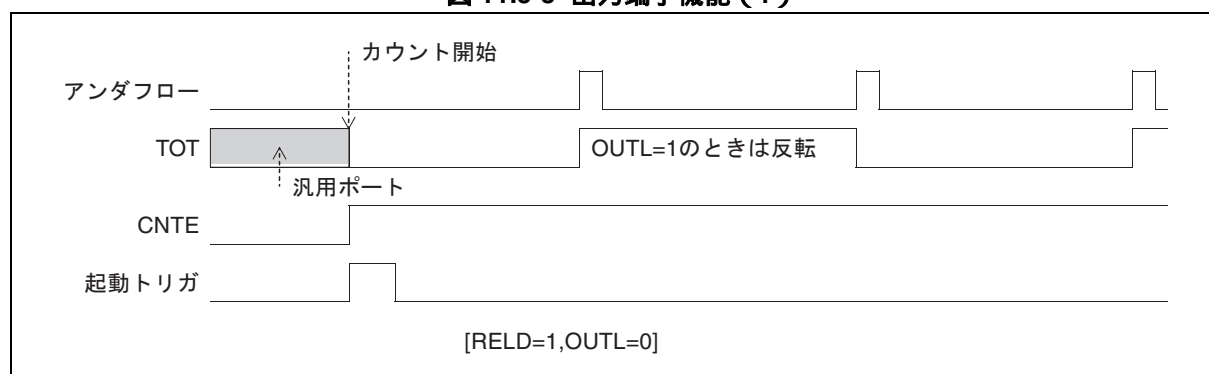
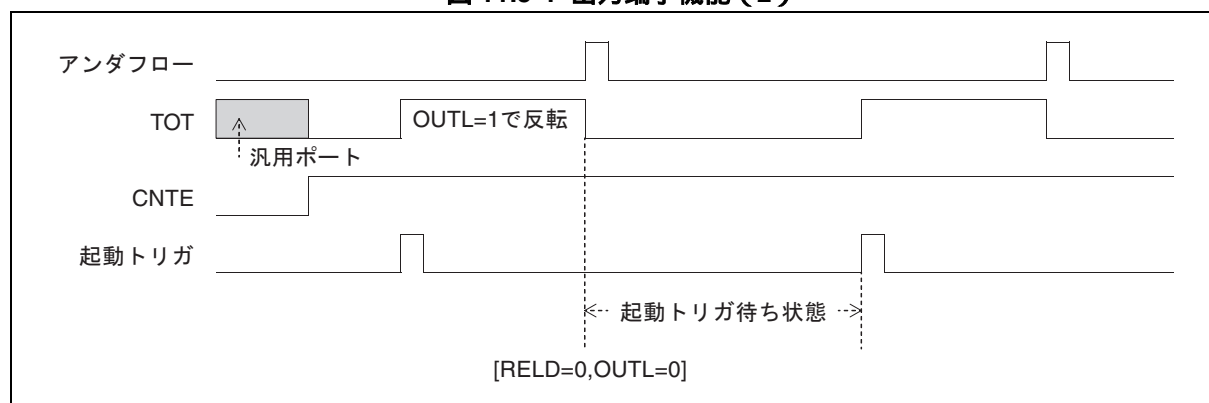


図 11.5-4 出力端子機能（2）



11.6 カウンタの動作状態

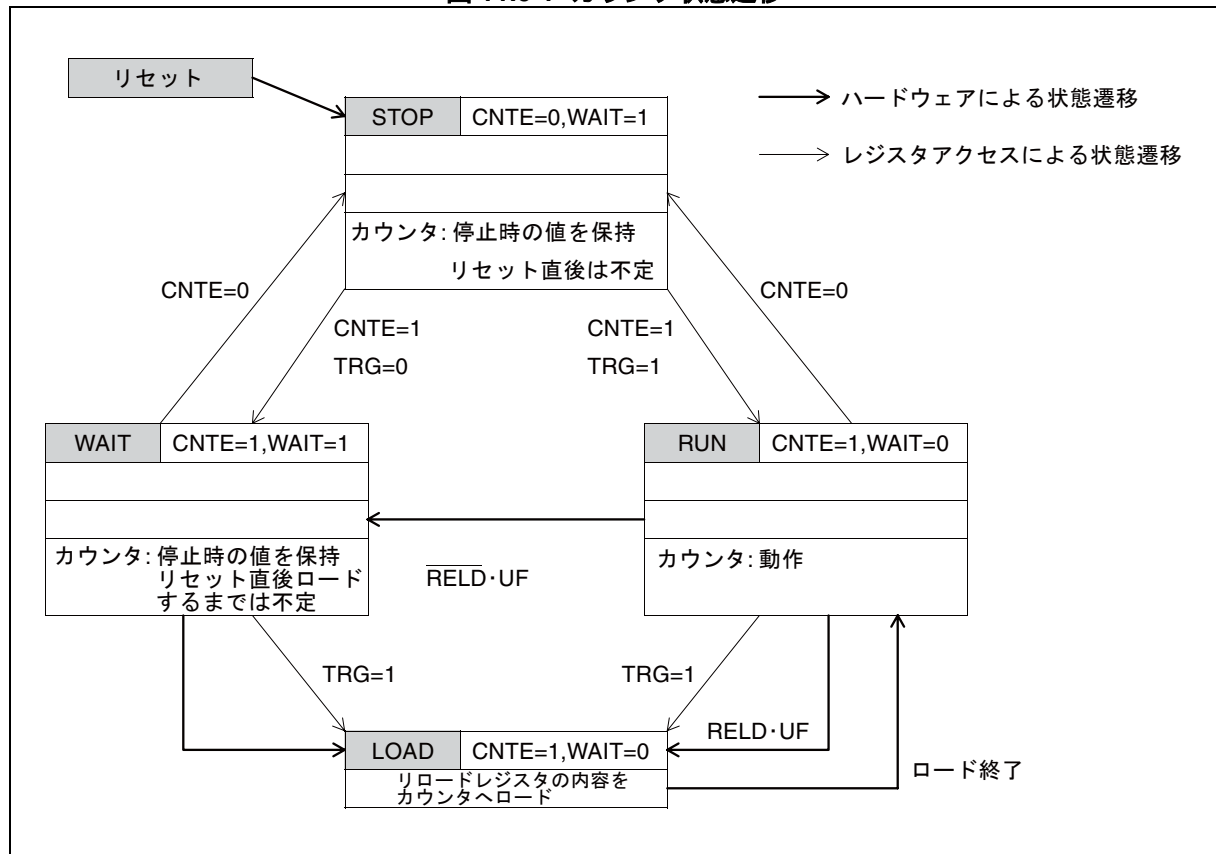
カウンタの状態は、コントロールレジスタの CNTE ビットと内部信号の WAIT 信号によって決まります。設定可能な状態として次の 3 つがあります。

- CNTE=0, WAIT=1 の停止状態（STOP 状態）
- CNTE=1, WAIT=1 の起動トリガ待ち状態（WAIT 状態）
- CNTE=1, WAIT=0 の動作状態（RUN 状態）

■ カウンタの動作状態

図 11.6-1 に、カウンタの動作状態を示します。

図 11.6-1 カウンタ状態遷移



第12章

8/16 ビット PPG

8/16 ビット PPG の機能と動作について説明します。

12.1 8/16 ビット PPG の概要

12.2 8 ビット PPG のブロックダイアグラム

12.3 8/16 ビット PPG のレジスタ

12.4 8/16 ビット PPG の動作

12.1 8/16 ビット PPG の概要

8/16 ビット PPG は、8/16 ビットのリロードタイマモジュールで、タイマ動作に応じたパルス出力制御により、PPG 出力を行います。MB90550A/B シリーズでは、8/16 ビット PPG を 3ch 内蔵しています。

■ 8/16 ビット PPG の概要

8/16 ビット PPG は、8bit ダウンカウンタ 2 個、8bit リロードレジスタ 4 個、8bit 制御レジスタ 3 個、外部パルス出力端子 2 本および割込み出力 2 本で構成されており、以下の機能を実現しています。

- 8 ビット PPG 出力 6ch 独立動作モード

6ch 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

3ch の 16bit の PPG 出力動作が可能です。

- 8+8 ビット PPG 出力動作モード

ch.0/ch.2/ch.4 の出力を、ch.1/ch.3/ch.5 のクロック入力とすることにより、任意周期の 8bit PPG 出力動作が可能です。

- PPG 出力動作

任意周期・デューティ比のパルス波を出力します。外付け回路により D/A コンバータとしても使用可能です。

12.2 8 ビット PPG のブロックダイアグラム

図 12.2-1 に, 8 ビット PPG (ch.0/ch.2/ch.4) のブロックダイアグラムを, 図 12.2-2 に, 8 ビット PPG (ch.1/ch.3/ch.5) のブロックダイアグラムを示します。

■ 8 ビット PPG のブロックダイアグラム

図 12.2-1 8 ビット PPG (ch.0/ch.2/ch.4) のブロックダイアグラム

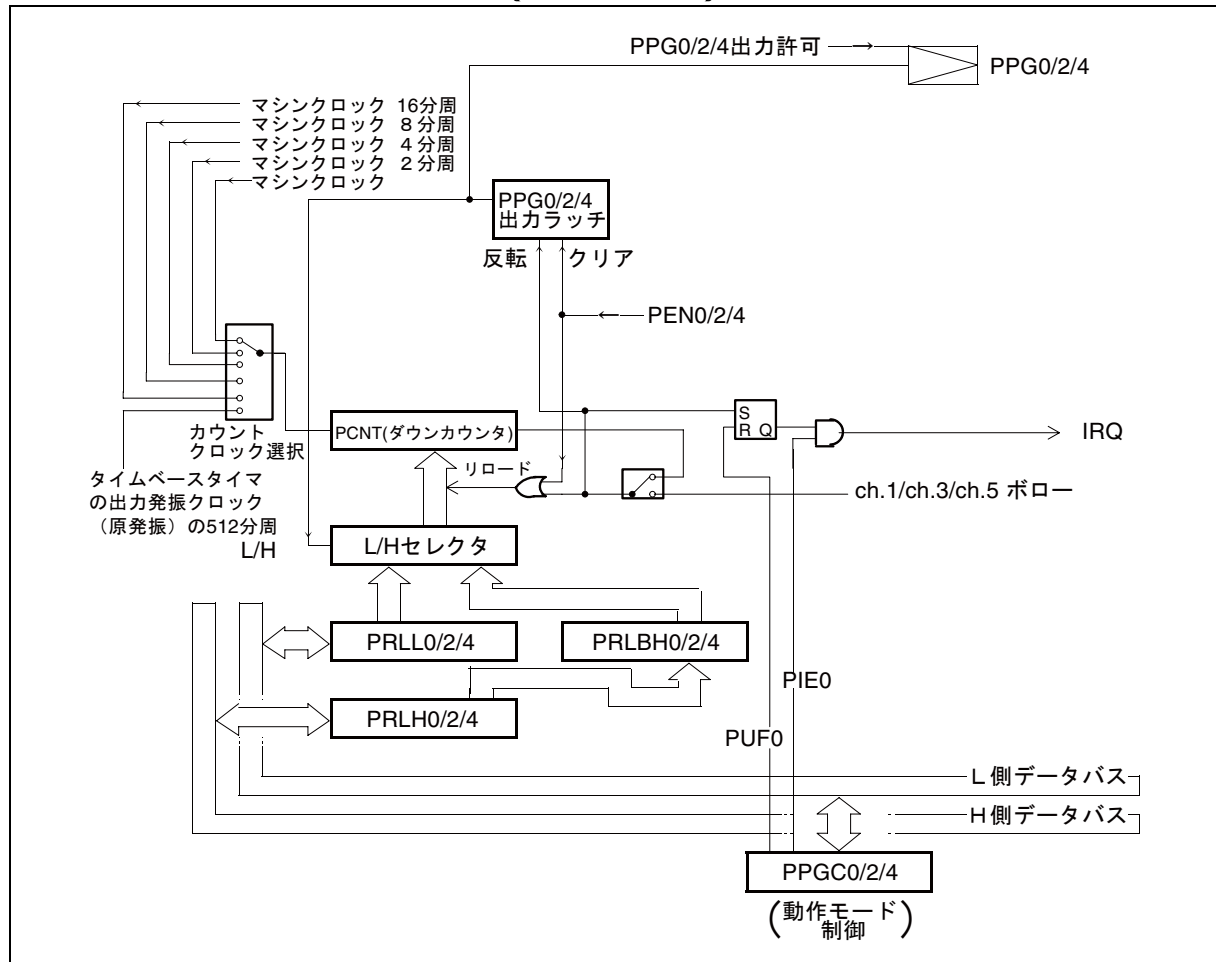
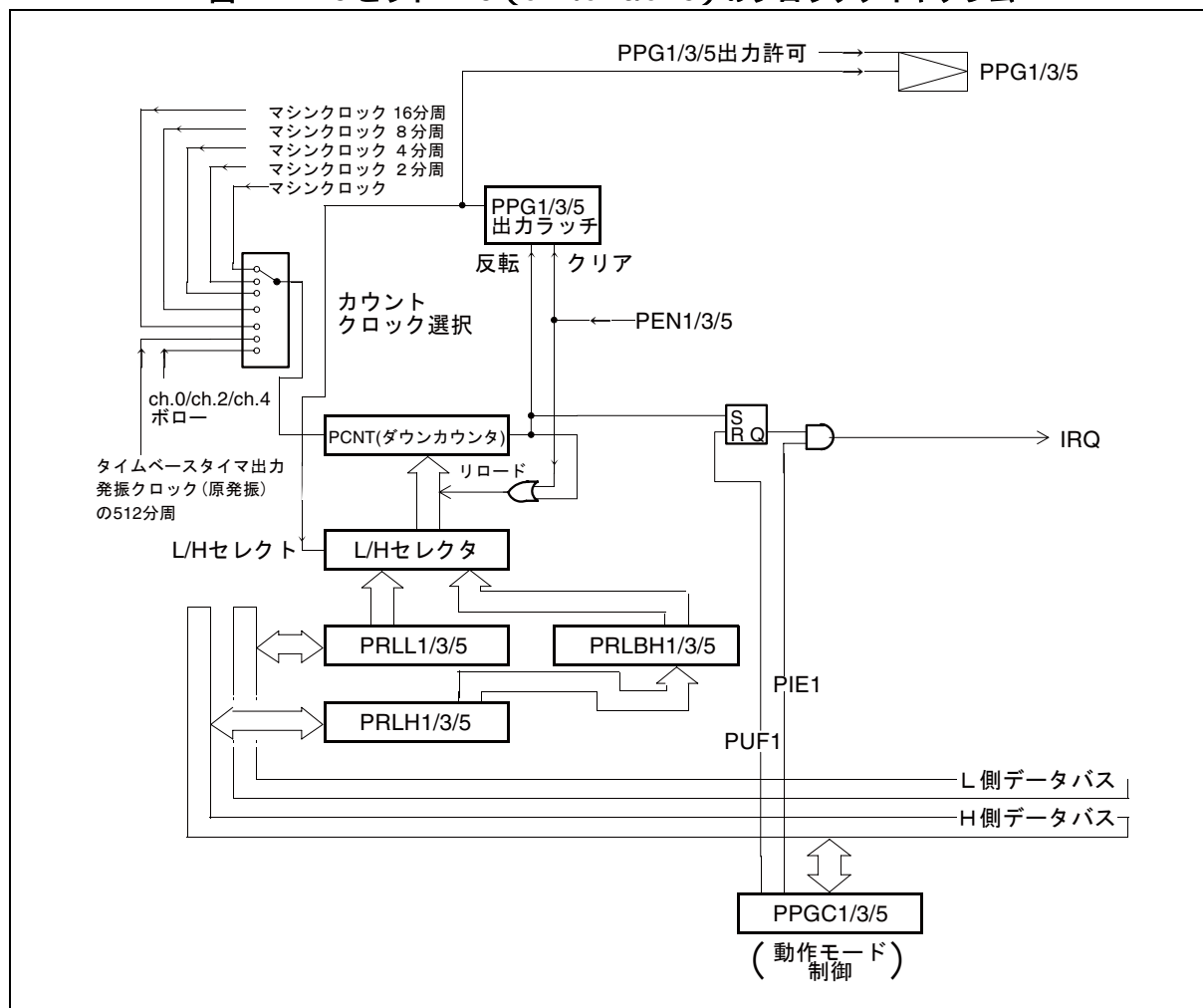


図 12.2-2 8 ビット PPG (ch.1/ch.3/ch.5) のブロックダイアグラム



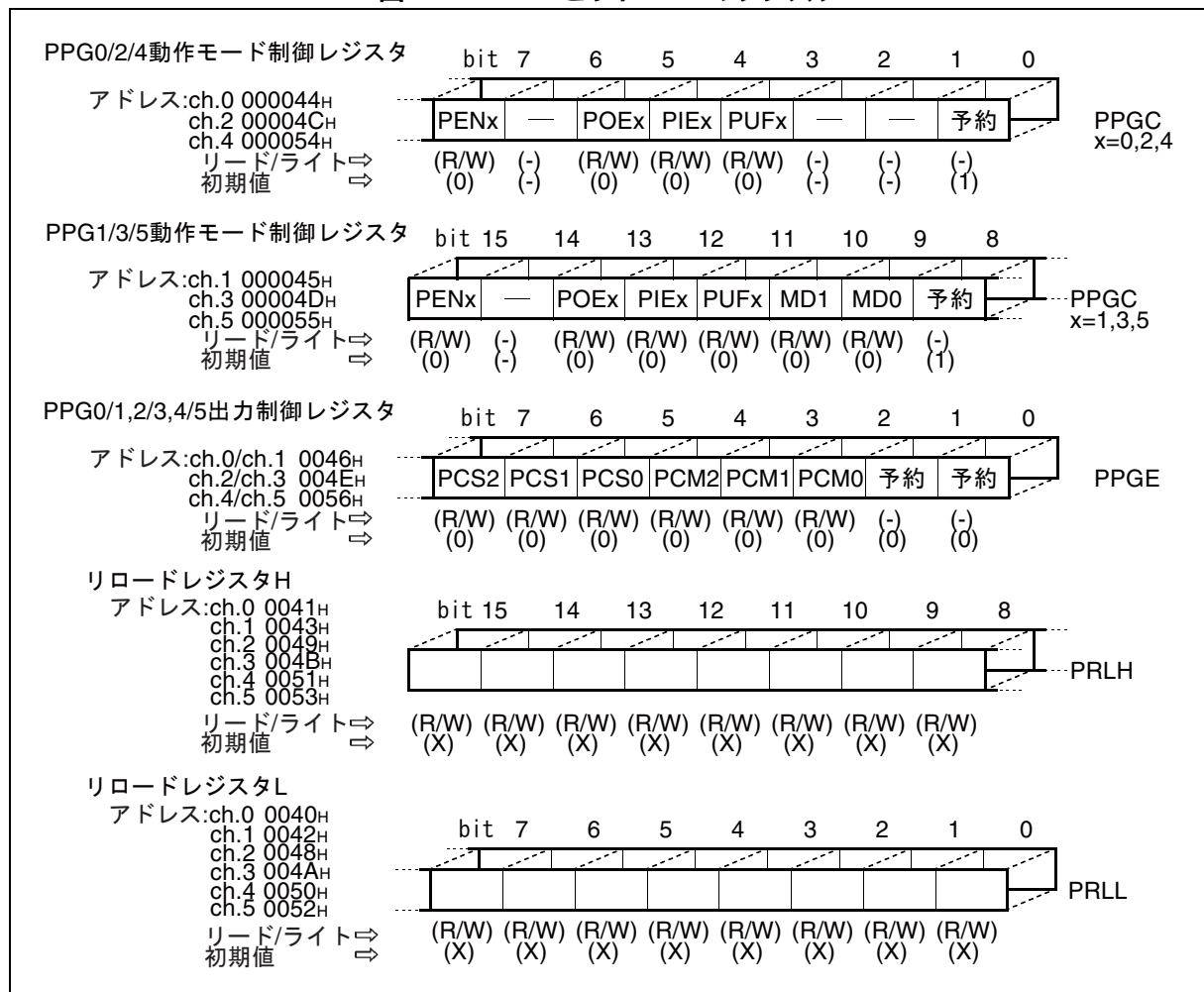
12.3 8/16ビット PPG のレジスタ

8/16ビット PPG のレジスタには、次の3種類があります。

- PPG 動作モード制御レジスタ
- PPG 出力制御レジスタ
- リロードレジスタ

■ 8/16ビット PPG のレジスタ

図 12.3-1 8/16ビット PPG のレジスタ



12.3.1 PPG0 動作モード制御レジスタ (PPGC0)

PPG0 動作モード制御レジスタ (PPGC0) は、8/16 ビット PPG の動作モードの選択、端子出力制御、カウントクロック選択、トリガを制御します。

ここでは、ch.0 について説明します。ch.2 および ch.4 については、ch.0=ch.2=ch.4, ch.1=ch.3=ch.5 に読み換えてください。

■ PPG0 動作モード制御レジスタ (PPGC0)

図 12.3-2 PPG0 動作モード制御レジスタ (PPGC0)

PPG0動作モード制御レジスタ							
アドレス:ch.0 000044H							
ch.2 00004CH							
ch.4 000054H							
リード/ライト⇒							
初期値⇒							
<div> <div>bit 7</div> <div>6</div> <div>5</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> <div>0</div> </div> <div> <div>PENx</div> <div>—</div> <div>POEx</div> <div>PIEx</div> <div>PUFx</div> <div>—</div> <div>—</div> <div>予約</div> </div> <div> <div>(R/W)</div> <div>(-)</div> <div>(R/W)</div> <div>(R/W)</div> <div>(R/W)</div> <div>(-)</div> <div>(-)</div> <div>(-)</div> </div> <div> <div>(0)</div> <div>(-)</div> <div>(0)</div> <div>(0)</div> <div>(0)</div> <div>(-)</div> <div>(-)</div> <div>(1)</div> </div> <div>PPGCx=0,2,4</div>							

[bit7] PEN0 (Ppg ENable)

PEN0 ビットは、PPG の動作開始および動作モードを、表 12.3-1 で示すように選択します。本ビットに "1" を書き込むことで PPG はカウントを開始します。

表 12.3-1 PEN0 (動作許可ビット) の機能

PEN0	機能
0	動作停止 ("L" レベル出力保持) [初期値]
1	PPG 動作イネーブル

[bit5] POE0 (Ppg Output Enable)

POE0 ビットは、パルス出力外部端子 PPG0 を、表 12.3-2 で示すように制御します。

表 12.3-2 POE0 (PPG0 端子出力許可ビット) の機能

POE0	機能
0	PPG 出力禁止 (汎用ポート端子) [初期値]
1	PPG0 出力許可 (PPG 出力端子)

[bit4] PIE0 (Ppg Interrupt Enable)

PIE0 ビットは、PPG の割込み許可を、表 12.3-3 で示すように制御します。

本ビットが "1" のとき、PUF0=1 になると割込み要求を発生します。

本ビットが "0" のときは割込み要求を発生しません。

表 12.3-3 PIE0 (PPG の割込み許可ビット) の機能

PIE0	機能
0	割込み禁止 [初期値]
1	割込み許可

[bit3] PUF0 (Ppg Underflow Flag)

PUF0 ビットは、PPG のカウンタアンダフロービットです。8 ビットダウンカウンタ "PCNT" との関係を表 12.3-4 に示します。

表 12.3-4 PUF0 (PPG カウンタアンダフロービット) の機能

PUF0	機能
0	ダウンカウンタ "PCNT" のアンダフロー未検出 [初期値]
1	ダウンカウンタ "PCNT" のアンダフロー検出

8 ビット PPG 2ch モードおよび 8 ビットプリスケラ +8 ビット PPG モードのときには、ch.0 のカウンタの値が 00_H ~ FF_H へ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 PPG 1ch モードのときには、ch.1/ch.0 のカウンタの値が 0000_H ~ FFFF_H へ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" 書込みは、意味がありません。

リードモディファイライトの読出し時には "1" が読まれます。

[bit0] 予約ビット

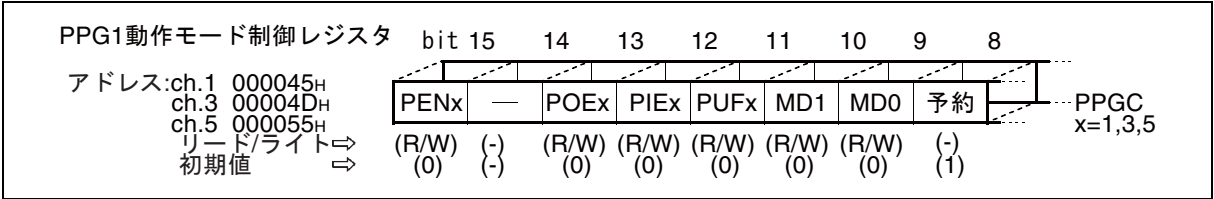
bit0 は、予約ビットです。PPGC0 を設定するときは必ず "1" に設定してください。

12.3.2 PPG1 動作モード制御レジスタ (PPGC1)

PPG1 動作モード制御レジスタ (PPGC1), 8/16 ビット PPG の動作モードの選択・端子出力制御・カウントクロック選択, トリガの制御を行うレジスタです。
ここでは, ch.1 について説明します。ch.3, ch.5 については, ch.0=ch.2=ch.4, ch.1=ch.3=ch.5 と読み換えてください。

■ PPG1 動作モード制御レジスタ (PPGC1)

図 12.3-3 PPG1 動作モード制御レジスタ (PPGC1)



[bit15] PEN1 (Ppg ENable)
PEN1 ビットは, PPG の動作開始および動作モードを, 表 12.3-5 で示すように選択します。本ビットに "1" を書き込むことで PWM はカウントを開始します。

表 12.3-5 動作許可ビット (PEN1) の機能

PEN1	機能
0	動作停止 ("L" レベル出力保持) [初期値]
1	PPG 動作イネーブル

[bit13] POE1 (Ppg Output ENable)
POE1 は, パルス出力外部端子 PPG1 を, 表 12.3-6 で示すように制御します。

表 12.3-6 POE1 (PPG1 端子出力許可ビット) の機能

POE1	機能
0	汎用ポート端子 (パルス出力禁止) [初期値]
1	PPG1= パルス出力端子 (パルス出力許可)

[bit12] PIE1 (Ppg Interrupt Enable)

PIE1 ビットは、PPG の割込み許可を、表 12.3-7 で示すように制御します。本ビットが "1" のとき、PUF1=1 になると割込み要求を発生します。本ビットが "0" のときは割込み要求を発生しません。

リセットにより、本ビットは "0" に初期化されます。本ビットは、読出し / 書込みが可能です。

表 12.3-7 PIE1(PPG の割込み許可ビット) の機能

PIE1	機能
0	割込み禁止 [初期値]
1	割込み許可

[bit11] PUF1 (Ppg Underflow Flag)

PUF1 は、PPG のカウンタアンダフロービットです。8 ビットダウンカウンタ "PCNT" との関係性を、表 12.3-8 に示します。

8 ビット PPG2ch モードおよび 8 ビットプリスケラ +8 ビット PPG モードのときには、ch.1 のカウンタの値が 00_H ~ FF_H へ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 PPG1ch モードのときには、ch.1/ch.0 のカウンタの値が 0000_H ~ FFFF_H へ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" 書込みは、意味がありません。リードモディファイライトの読出し時には "1" が読まれます。

リセットにより、本ビットは "0" に初期化されます。本ビットは、読出し / 書込みが可能です。

表 12.3-8 PUF1(PPG カウンタアンダフロービット) の機能

PUF1	機能
0	ダウンカウンタ "PCNT" のアンダフロー未検出 [初期値]
1	ダウンカウンタ "PCNT" のアンダフロー検出

[bit10, bit9] MD2, MD1 (ppg count MoDe)

MD2, MD1 ビットは、PPG タイマの動作モードを、表 12.3-9 のように選択します。

リセットにより、本ビットは "00" に初期化されます。

本ビットは、読出し / 書込みが可能です。

表 12.3-9 MD2, MD1(動作モード選択ビット) の機能

MD1	MD0	動作モード [初期値]
0	0	8 ビット PPG 2ch 独立モード
0	1	8 ビットプリスケラ +8 ビット PPG 1ch モード
1	0	予約 (設定禁止)
1	1	16 ビット PPG 1ch モード

< 注意事項 >

- 本ビットを "10" に設定しないでください。
 - 本ビットを "01" に設定する場合には、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。
 - 本ビットを "11" に設定する場合には、PPGC0/PPGC1 をワード転送で書き換え、PEN0 ビットおよび PEN1 ビットを同時に "11" または "00" にセットしてください。
-

[bit8] 予約ビット

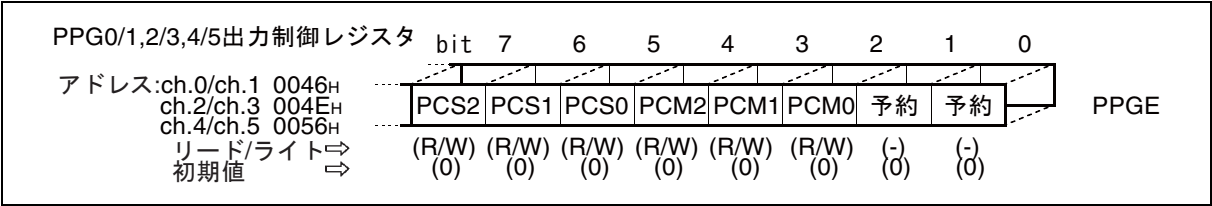
bit8 は、予約ビットです。PPGC1 を設定するときは必ず "1" に設定してください。

12.3.3 PPG0/1 出力端子制御レジスタ (PPGE)

PPG0/1 出力端子制御レジスタ (PPGE) は , 8/16 ビット PPG の端子出力制御を行う 8 ビットの制御レジスタです。

■ PPG0/1 出力端子制御レジスタ (PPGE)

図 12.3-4 PPG0/1 出力端子制御レジスタ (PPGE)



[bit7 ~ bit5] PCS2 ~ PCS0 (Ppg Count Select)

PCS2 ~ PCS0 は , ch.1 のダウンカウンタの動作クロックを表 12.3-10 に示すように選択します。

表 12.3-10 PCS2 ~ PCS0(カウントクロック選択ビット) の機能

PCS2	PCS1	PCS0	動作モード
0	0	0	マシントック (62.5ns マシントック 16MHz 時)
0	0	1	マシントック /2 (125 ns マシントック 16MHz 時)
0	1	0	マシントック /4 (250 ns マシントック 16MHz 時)
0	1	1	マシントック /8 (500 ns マシントック 16MHz 時)
1	0	0	マシントック /16 (1μs マシントック 16MHz 時)
1	1	1	タイムベースタイマからの入力クロック (128μs 原発振 4MHz 時)

< 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 モードのときには , ch.1 の PPG は , ch.0 からカウントクロックを受けて動作するため , PCS2 ~ PCS0 の指定は , 無効になります。

[bit4 ~ bit2] PCM2 ~ PCM0 (Ppg Count Mode)

PCM2 ~ PCM0 ビットは, ch.0 のダウンカウンタの動作クロックを表 12.3-11 のように選択します。

表 12.3-11 PCM2 ~ PCM0(カウントクロック選択ビット) の機能

PCM2	PCM1	PCM0	動作モード
0	0	0	マシンクロック (62.5ns マシンクロック 16MHz 時)
0	0	1	マシンクロック /2 (125 ns マシンクロック 16MHz 時)
0	1	0	マシンクロック /4 (250 ns マシンクロック 16MHz 時)
0	1	1	マシンクロック /8 (500 ns マシンクロック 16MHz 時)
1	0	0	マシンクロック /16 (1μs マシンクロック 16MHz 時)
1	1	1	タイムベースタイマからの入力クロック (128μs 原発振 4MHz 時)

[bit1, bit0] 予約ビット

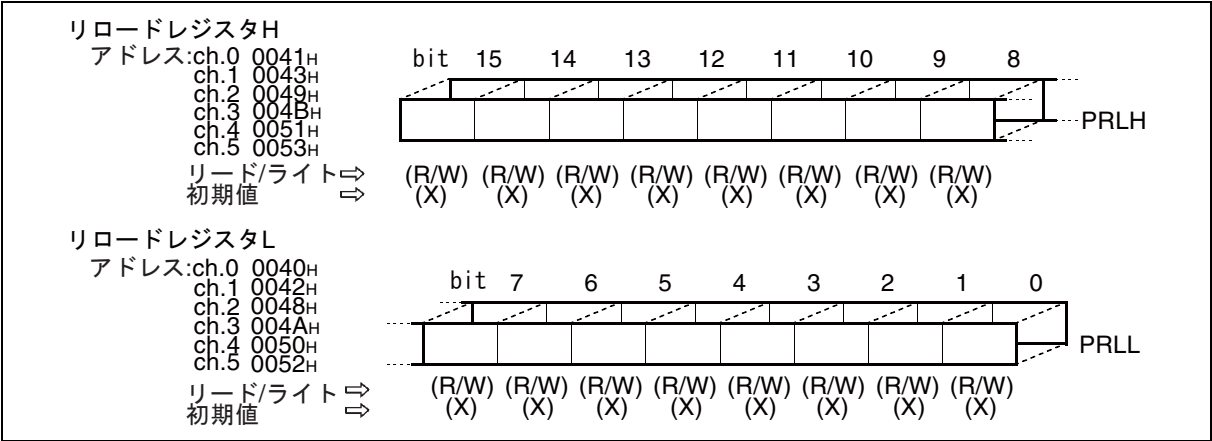
bit1, bit0 は, 予約ビットです。PPGE を設定するときは, 必ず "0" に設定してください。

12.3.4 リロードレジスタ (PRLH/PRLH)

リロードレジスタ (PRLH/PRLH) は、ダウンカウンタ 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。

■ リロードレジスタ (PRLH/PRLH)

図 12.3-5 リロードレジスタ (PRLH/PRLH)



リロードレジスタ (PRLH/PRLH) は、それぞれ表 12.3-12 に示す機能をもっています。どちらのレジスタも、リード/ライトが可能です。

表 12.3-12 リロードレジスタ (PRLH/PRLH)

レジスタ名	機能
PRLH	L 側リロード値保持
PRLH	H 側リロード値保持

< 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 の PRLH と PRLH に異なる値を設定すると、ch.1 の PPG 波形がサイクルごとに異なる場合がありますので、ch.0 の PRLH と PRLH は、同じ値に設定することを推奨します。

12.4 8/16 ビット PPG の動作

8/16 ビット PPG には、8 ビット長の PPG ユニットが 2ch あります。2ch 独立モード以外に、連結動作させることにより、8 ビットプリスケアラ +8 ビット PPG モードと、16 ビット PPG 1ch モードの計 3 種類の動作を行うことができます。
なお、ここでは、ch.0/ch.1 について説明します。ch.2/ch.3 および ch.4/ch.5 については、ch.0=ch.2=ch.4 および ch.1=ch.3=ch.5 に読み換えてください。

■ 8/16 ビット PPG の動作

8bit 長の PPG ユニットそれぞれは、8bit 長のリロードレジスタが L 側と H 側の 2 本あります (PRLH, 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) に 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、カウンタのボロー発生によるリロード時に、端子出力 (PPG) の値を反転させます。この動作により、端子出力 (PPG) 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、表 12.4-1 に示します。

表 12.4-1 リロード動作とパルス出力の関係

リロード動作	出力端子 PPG0, PPG1 の状態遷移
PRLH PCNT	0 1
PRLH PCNT	1 0

また、PPGC0 レジスタの PIE0 ビットが "1" のときおよび PPGC1 レジスタの PIE1 ビットが "1" のとき、それぞれのカウンタの 00_H から FF_H へのボロー (16 ビット PPG モードの場合には、0000_H から FFFF_H へのボロー) によって割込み要求が出力されます。

■ 8/16 ビット PPG の割込み

8/16 ビット PPG の割込みは、リロード値がカウントアウトし、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 PPG 2ch モードおよび 8 ビットプリスケアラ +8 ビット PPG モードのときには、それぞれのカウンタのボローにより、それぞれの割込みの要求が行われますが、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り、PUF0 ビットと PUF1 ビットが同時にセットされます。このため、割込み要因を一本化するために、PIE0 ビットまたは PIE1 ビットのどちらか一方のみを許可にすることを推奨します。また、割込み要因のクリアも PUF0 ビットと PUF1 ビットを同時に行うことを推奨します。

■ 各ハードウェアの初期値

8/16 ビット PPG の各ハードウェアは、リセット時に以下のように初期化されます。

● レジスタ

- PPGC0 0X000001_B
- PPGC1 00000001_B
- PPGOE XXXXXX00_B

● パルス出力

- PPG0 "L"
- PPG1 "L"
- PE00 PPG0 出力禁止
- PE10 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は初期化されません。

12.4.1 8/16 ビット PPG の動作モード

8/16 ビット PPG には、以下の 3 種類の動作モードがあります。

- 2ch 独立モード
 - 8 ビットプリスケアラ +8 ビット PPG モード
 - 16 ビット PPG 1ch モード
-

■ 8/16 ビット PPG の動作モード

● 2ch 独立モード

8 ビット PPG として 2ch 独立に動作させる動作モードです。

PPG0 端子は、ch.0 の PPG 出力が接続され、PPG1 端子は、ch.1 の PPG 出力が接続されます。

● 8 ビットプリスケアラ +8 ビット PPG モード

ch.0 を 8 ビットプリスケアラとして動作させ、ch.1 を ch.0 のボロー出力でカウントすることにより、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 端子は、ch.0 のプリスケアラ出力が接続され、PPG1 端子は、ch.1 の PPG 出力が接続されます。

● 16 ビット PPG 1ch モード

ch.0 と ch.1 を連結させて、16 ビットの PPG として動作させる動作モードです。PPG0 端子と PPG1 端子は、両方とも 16 ビット PPG 出力が接続されます。

12.4.2 PPG 出力動作

8/16 ビット PPG は, ch.0 の PPG については, PPGC0 レジスタの PEN0 ビットを, ch.1 の PPG については, PPGC1 レジスタの PEN1 ビットを "1" にセットすることによって起動され, カウントを開始します。動作を開始した後は, PPGC0 レジスタの PEN0 ビットまたは PPGC1 レジスタの PEN1 ビットに "0" を書き込むことによってカウント動作を停止し, 停止した後, パルス出力は L レベルを保持します。

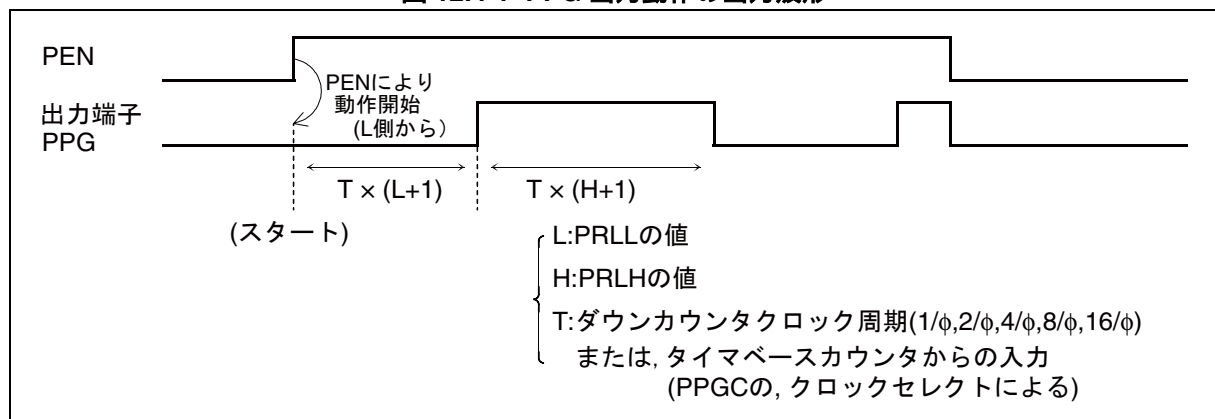
■ PPG 出力動作

PPG の出力動作時には, 次の 2 つの点について注意してください。

- 8 ビットプリスケラ +8 ビット PPG モードのときには, ch.0 を停止状態で, ch.1 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 PEN0 ビットと PPGC1 レジスタの PEN1 ビットは, 同時に開始・停止の制御を行ってください。

PPG 動作時は, 任意周波数・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, 動作停止を設定するまで停止しません。

図 12.4-1 PPG 出力動作の出力波形



■ リロード値とパルス幅の関係

表 12.4-1 に示すように、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。すなわち、8 ビット PPG 動作時のリロードレジスタ値が 00_H のときおよび 16 ビット PPG 動作時のリロードレジスタ値が 0000_H のときは、カウントクロック 1 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。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、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。

$$Pl = T \times (L+1)$$

$$Ph = T \times (H+1)$$

L : PRLH の値

H : PRLH の値

T : 入力クロック周期

Ph: H パルスの幅

Pl: L パルスの幅

12.4.3 カウントクロックの選択

8/16 ビット PPG の動作に使用するカウントクロックは、マシンの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おり、6 種類のカウントクロック入力を選択できます。PPGE レジスタの PCM2 ~ PCM0 ビットで ch.0 のクロックを、PCS2 ~ PCS0 ビットで ch.1 のクロックを選択します。クロックは、マシンのクロックの 1/16 ~ 1 倍のマシンのクロックとタイムベースタイマからの入力より選択できます。

ただし、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 モードのときには、ch.1 の PPG は、ch.0 からカウントクロックを受けて動作するため、PPGC1 レジスタ中の PCS1 ビットの値は、無効になります。

■ カウントクロックの選択に関する注意事項

タイムベースタイマの入力を使用した場合、トリガにより起動がかかる最初およびストップ後の最初のカウント周期がずれる可能性があります。また、本モジュール動作中にタイムベースカウンタのクリアを行うと周期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・ch.0 が動作状態・ch.1 が停止状態であるときに ch.1 の起動を行うと、最初のカウント周期がずれる可能性があります。

12.4.4 パルスの端子出力の制御

本モジュールの動作によって生成されたパルス出力は、外部端子 PPG0/PPG1 より出力させることができます。

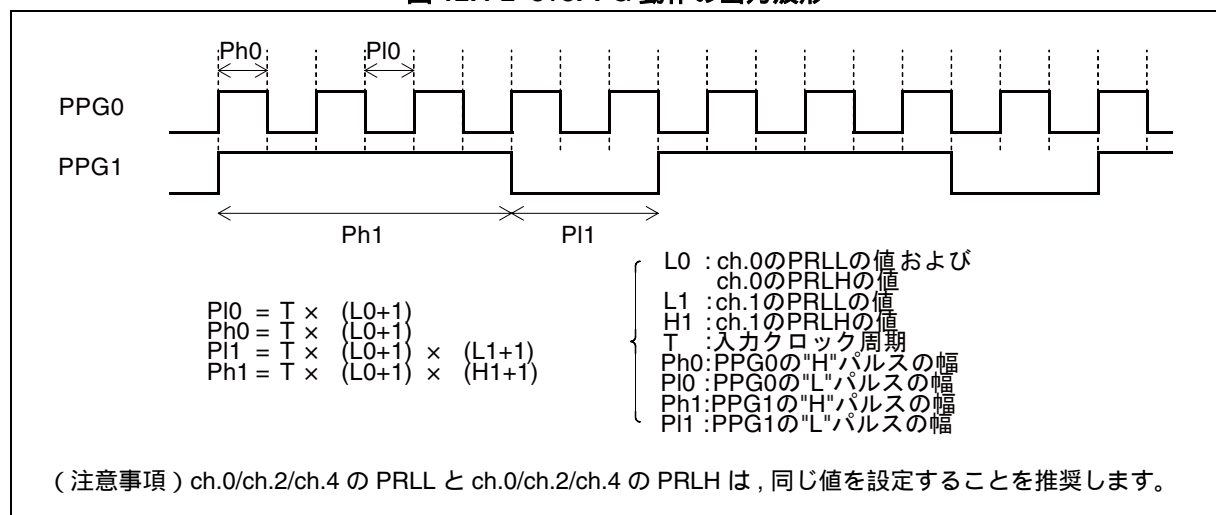
外部端子出力の許可は、PPG0 端子を PPGC0 レジスタの POE0 ビットにて、PPG1 端子を PPGC1 レジスタの POE1 ビットによって行われます。本ビットが "0" のとき（初期値）は、パルス出力は外部端子より出力されずに、汎用ポートとして機能します。本ビットに "1" を設定すると、パルス出力が外部端子より出力されます。

■ パルスの端子出力の制御

16 ビット PPG モードでは、PPG0, PPG1 は同じ波形が出力されるので、どちらの外部端子出力を許可しても同じ出力を得ることができます。

8 ビットプリスケアラ +8 ビット PPG モードでは、PPG0 は 8 ビットプリスケアラのトグル波形が出力され、PPG1 は 8 ビット PPG の波形が出力されます。このモード時の出力波形の例を、図 12.4-2 に示します。

図 12.4-2 8+8PPG 動作の出力波形

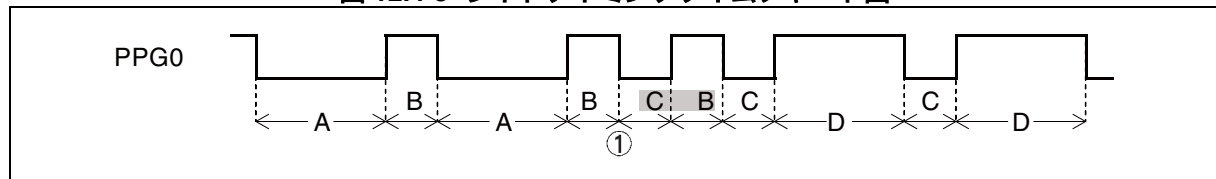


12.4.5 リロードレジスタへのライトタイミング

16 ビット PPG モード以外のモードの場合には、リロードレジスタ PRL_L, PRL_H への書込みはワード転送命令を使用することを推奨します。バイト転送命令 2 回にて書き込んだ場合、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。

■ リロードレジスタへのライトタイミング

図 12.4-3 ライトタイミングタイムチャート図

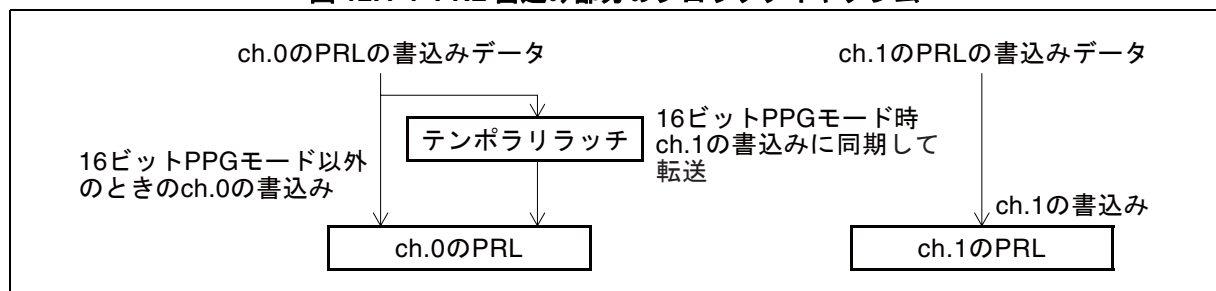


タイムチャートにおいて ① のタイミングの前で PRL_L を A から C に書き換えて、その後 PRL_H の値を B から D に書き換えた場合、① のタイミングでの PRL の値は PRL_L=C, PRL_H=B のため 1 回だけ L 側のカウンタ数 C, H 側のカウンタ数 B のパルスが発生されます。

同様に、16 ビット PPG モードで使用する場合には、ch.0 と ch.1 の PRL はロングワード転送で書き込むかまたは ch.0 → ch.1 の順にワード転送で書き込んでください。このモードのときには、ch.0 の PRL への書込みは、一時的にテンポラリに書き込まれ、その後 ch.1 の PRL に書込みを行った時点で、実際に ch.0 の PRL への書込みが行われます。

なお、16 ビット PPG モード以外では、ch.0 と ch.1 の PRL の書込みは独立して行えます。

図 12.4-4 PRL 書込み部分のブロックダイヤグラム



第13章

DTP/ 外部割込み

この章には、DTP/ 外部割込みの機能と動作について示します。

13.1 DTP/ 外部割込みの概要

13.2 DTP/ 外部割込みのレジスタ

13.3 DTP/ 外部割込みの動作

13.4 DTP/ 外部割込みの使用上の注意

13.1 DTP/ 外部割込みの概要

DTP(Data Transfer Peripheral)/ 外部割込み回路は，デバイス外部に存在するペリフェラルと F²MC-16LX CPU との間において，外部ペリフェラルが発生する DMA 要求あるいは割込み要求を受け取り，これを F²MC-16LX CPU に伝えて拡張インテリジェント I/O サービスあるいは割込み処理を起動させるための周辺です。要求レベルとして，拡張インテリジェント I/O サービスの場合は "H", "L" の 2 種が，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 種が選択可能です。

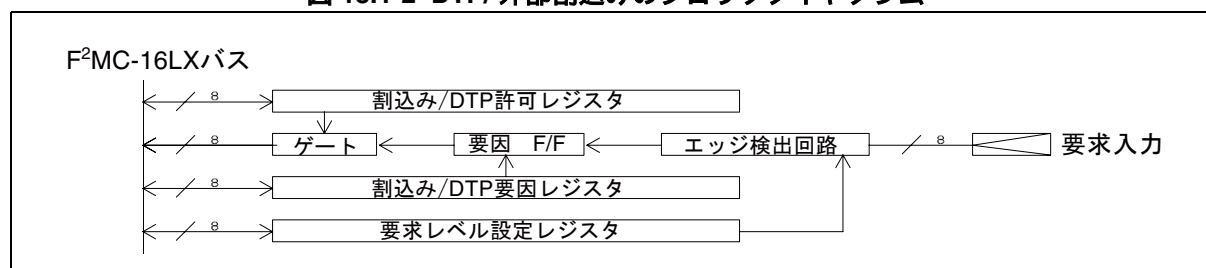
■ DTP/ 外部割込みのレジスタ

図 13.1-1 DTP/ 外部割込みのレジスタ

割込み/DTP要因レジスタ	
bit 15 14 13 12 11 10 9 8	
アドレス:000039 _H	ER7 ER6 ER5 ER4 ER3 ER2 ER1 ER0
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)
割込み/DTP許可レジスタ	
bit 7 6 5 4 3 2 1 0	
アドレス:000038 _H	EN7 EN6 EN5 EN4 EN3 EN2 EN1 EN0
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)
要求レベル設定レジスタ上位	
bit 15 14 13 12 11 10 9 8	
アドレス:00003B _H	LB7 LA7 LB6 LA6 LB5 LA5 LB4 LA4
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)
要求レベル設定レジスタ下位	
bit 7 6 5 4 3 2 1 0	
アドレス:00003A _H	LB3 LA3 LB2 LA2 LB1 LA1 LB0 LA0
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)

■ DTP/ 外部割込みのブロックダイアグラム

図 13.1-2 DTP/ 外部割込みのブロックダイアグラム



13.2 DTP/ 外部割込みのレジスタ

ENIR レジスタはデバイス端子を外部割込み /DTP 要求入力として使用し、割込みコントローラに対して要求を発生させる機能を動作させることを決定するレジスタです。

■ 割込み /DTP 許可レジスタ (ENIR)

図 13.2-1 割込み /DTP 許可レジスタ (ENIR)

割込み/DTP許可レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:000038 _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
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)

割込み /DTP 許可レジスタ (ENIR) の "1" を書かれたビットの対応する端子は外部割込み /DTP 要求入力として使用され、割込みコントローラに対して要求を発生する機能を動作させます。"0" が書かれたビットの対応する端子は外部割込み /DTP 要求入力要因は保持しますが、割込みコントローラに対しては要求を発生しません。

EN_x が IRQ_x に対応しています。

< 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)

図 13.2-2 割込み /DTP 要因レジスタ (EIRR)

割込み/DTP要因レジスタ								
	bit 15	14	13	12	11	10	9	8
アドレス:000039 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
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)

EIRR レジスタは読出し時には対応する外部割込み /DTP 要求があることを示し、書込み時にはこの要求を示すフリップフロップ内容をクリアするレジスタです。このレジスタを読出したときに "1" であったとき、このビットに対応する端子に外部割込み /DTP 要求があることを示します。

また、このレジスタに "0" を書き込むと対応するビットの要求フリップフロップはクリアされます。"1" の書込みでは何の操作も行いません。リードモディファイライトの読出し時には "1" が読まれます。

ER_x が IRQ_x に対応しています。

< 注意事項 >

- 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)

図 13.2-3 要求レベル設定レジスタ (ELVR)

要求レベル設定レジスタ上位		bit 15	14	13	12	11	10	9	8	
アドレス:00003B _H		LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	ELVR
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)	
要求レベル設定レジスタ下位		bit 7	6	5	4	3	2	1	0	
アドレス:00003A _H		LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR
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)	

ELVR レジスタは要求検出の選択を行うレジスタです。1 つの端子あたり 2bit が割り当てられていて, 以下のような対応となっています。要求入力レベルのとき, 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。

LA_x, LB_x が IRQ_x に対応しています。

表 13.2-1 要求レベル設定レジスタ (ELVR) の動作

LB _x	LA _x	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	エッジで要求あり
1	1	エッジで要求あり

< 参考 >

DTP/ 外部割込み端子に選択された検出信号が入力されると, DTP/ 外部割込み許可レジスタ (ENIR) の設定などに関係なく外部割込み要求フラグビット (ER7 ~ ER0) に "1" がセットされます。

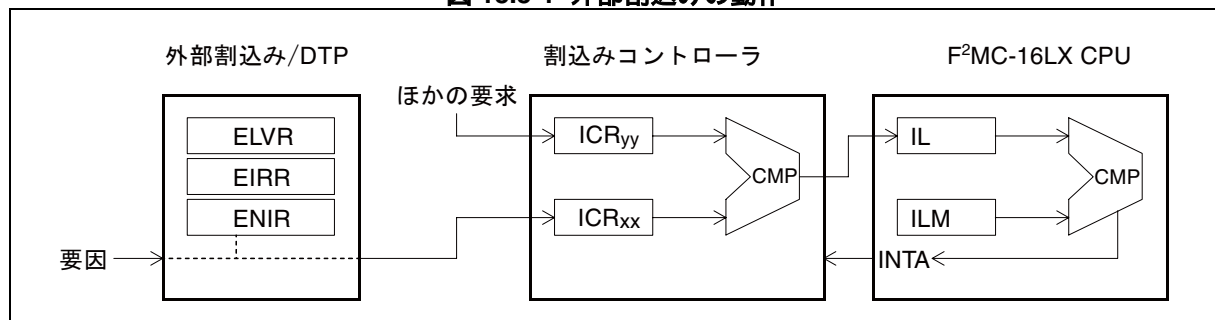
13.3 DTP/ 外部割込みの動作

外部割込み要求の設定ののち，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。

■ 外部割込みの動作

F²MC-16LX CPU は自分の内部にある CCR レジスタ中の ILM ビットと割込み要求を比較し，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 13.3-1 外部割込みの動作



ハードウェア割込み処理マイクロプログラムにおいて，CPU は割込みコントローラから ISE ビットの情報を読み出し，これにより当該要求が割込み処理であることを識別し，割込み処理マイクロプログラムへ分岐します。割込み処理マイクロでは割込みベクタ領域の読み出しと割込みコントローラへの割込みアクノリッジを発生し，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

初期化としてユーザープログラム内では拡張インテリジェント I/O サービスを起動するに当たって，拡張インテリジェント I/O サービス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ほぼ同じで，CPU がハードウェア割込み処理マイクロプログラムを起動するまでは全く同じです。DTP の場合は CPU がハードウェア割込み処理マイクロプログラム内で読み出す ISE ビットの内容が DTP を示しているので，拡張インテリジェント I/O サービス処理マイクロプログラムへ制御を移します。拡張インテリジェント I/O サービスが起動されると，アドレッシングされている外部ペリフェラルにリードまたはライト信号が送られ本チップとの転送が行われます。外部ペリフェラル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てください。転送が終了するとディスクリプタの更新などが行われ，その後転送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。転送

要因をクリアする信号を受けと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 13.3-2 DTP 動作終了時の外部割込み取り下げタイミング

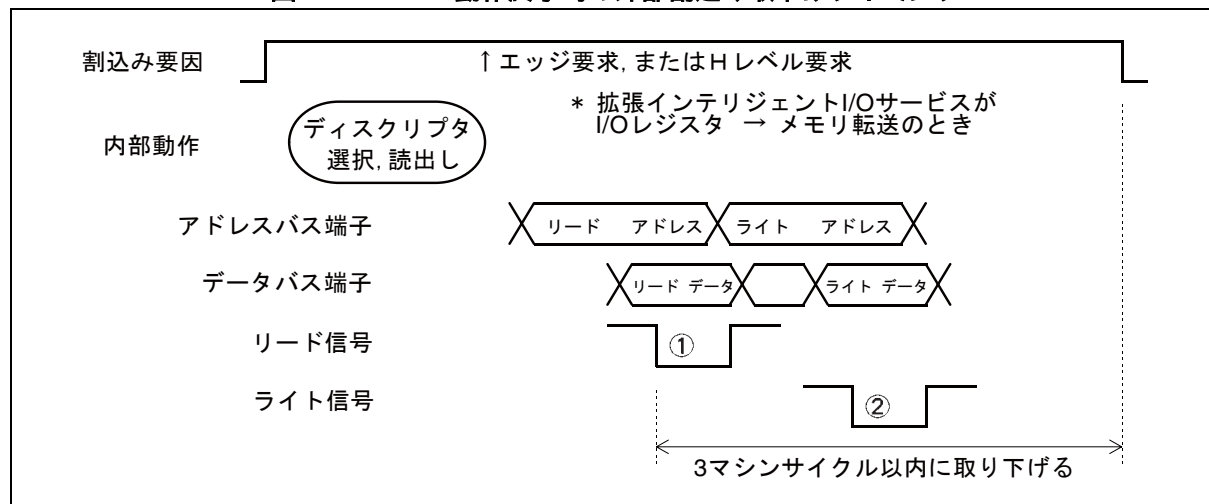
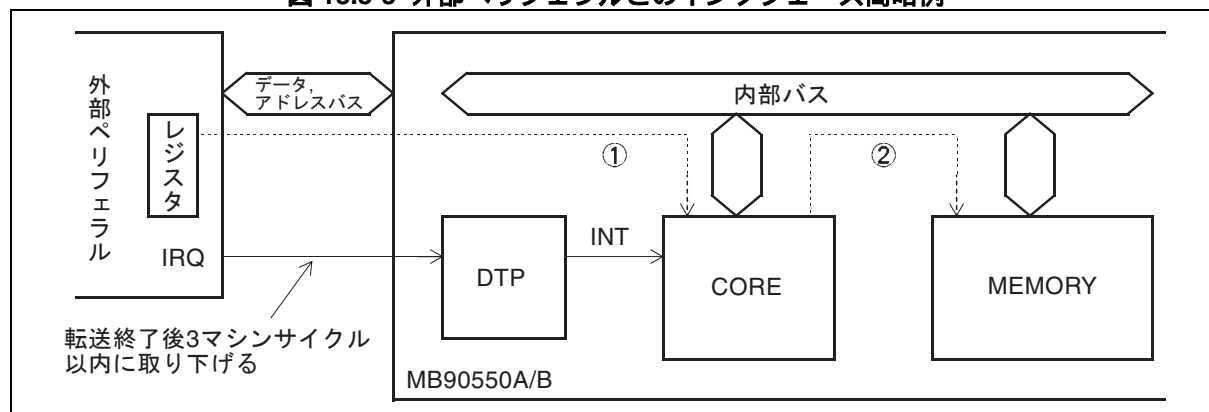


図 13.3-3 外部ペリフェラルとのインタフェース簡略例

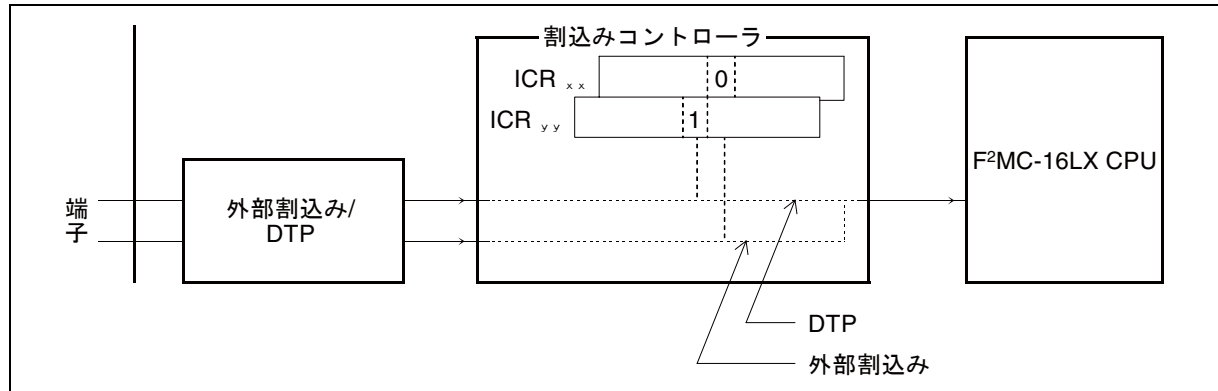


■ 外部割込み要求と DTP 要求の切換え

外部割込み要求と DTP 要求の切換えは、割込みコントローラの中にある、本リソースに対応する ICR レジスタ中の ISE ビットの設定によって行います。

各端子に対応して個々に ICR が割り当てられていますので、対応する ICR の ISE ビットに "1" が書かれた端子が DTP 要求として、また、"0" が書かれた場合には外部割込み要求として、それぞれ動作することとなります。

図 13.3-4 外部割り込み要求と DTP 要求の切換え



13.4 DTP/ 外部割込みの使用上の注意

DTP/ 外部割込みを使用するには、次の 4 つの点について特に注意が必要です。

- DTP を用いたときの外部に接続するペリフェラルの条件
 - スタンバイからの復帰
 - 外部割込み /DTP の動作手順
 - 外部割込み要求レベル
-

■ DTP を用いたときの外部に接続するペリフェラルの条件

DTP がサポートできる外部ペリフェラルは、転送が行われたことにより自動的に要求をクリアするものでなければなりません。また、転送動作が開始してから 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扱ってしまいます。

■ スタンバイからの復帰

クロック停止モードのスタンバイ状態からの復帰に外部割込みを使う場合は、入力要求を "H" レベル要求としてください。"L" レベル要求では誤動作を起こす可能性があります。エッジ要求ではクロック停止モードのスタンバイ状態からの復帰は行われません。

■ 外部割込み /DTP の動作手順

外部割込み/DTP内に存在するレジスタの設定を行う際、次の手順で設定してください。

- 1) 許可レジスタの対象となるビットを禁止状態にする。
- 2) 要求レベル設定レジスタの対象となるビットを設定する。
- 3) 要因レジスタの対象となるビットをクリアする。
- 4) 許可レジスタの対象となるビットを許可状態にする（ただし、3）と 4）はワード指定による同時書込み可）。

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ければなりません。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起こっ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検出のとき，エッジが入力され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とします。
- 要求入力レベルがレベル検出のとき，外部より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，外部割込み要求フラグビットをクリアして要因保持回路をクリアする必要があります。

図 13.4-1 レベル設定時の要因保持回路のクリア

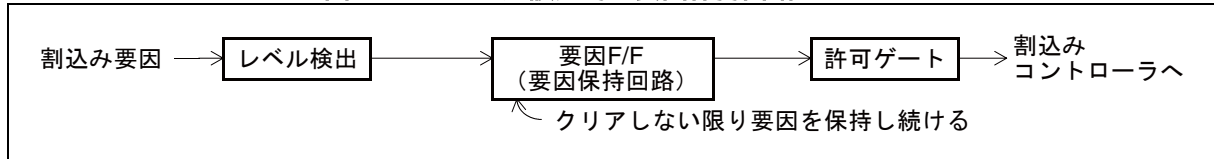
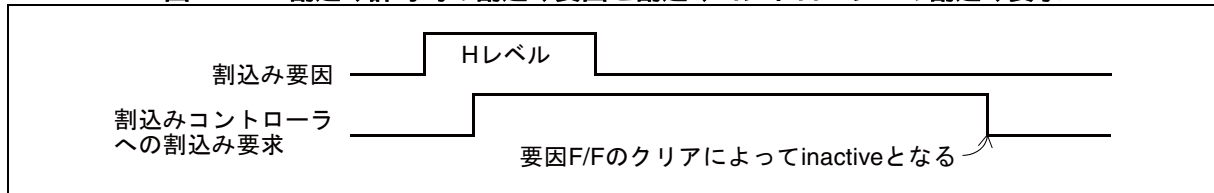


図 13.4-2 割込み許可時の割込み要因と割込みコントローラへの割込み要求



第14章

遅延割込み発生モジュール

遅延割込み発生モジュールの機能と動作について説明します。

14.1 遅延割込み発生モジュールの概要

14.2 遅延割込み発生モジュールの動作

14.1 遅延割込み発生モジュールの概要

遅延割込み発生モジュールは、タスク切換え用の割込みを発生するためのモジュールです。本モジュールを使用することで、ソフトウェアで F²MC-16LX CPU に対して割込み要求の発生 / 取消を行うことができます。

■ 遅延割込み発生モジュールのレジスタ

遅延割込み要因発生 / 解除レジスタ (DIRR) は、遅延割込み要求の発生 / 解除を制御するレジスタで、このレジスタに対して "1" を書き込み時には遅延割込みの要求を発生させ、"0" を書き込み時には遅延割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。

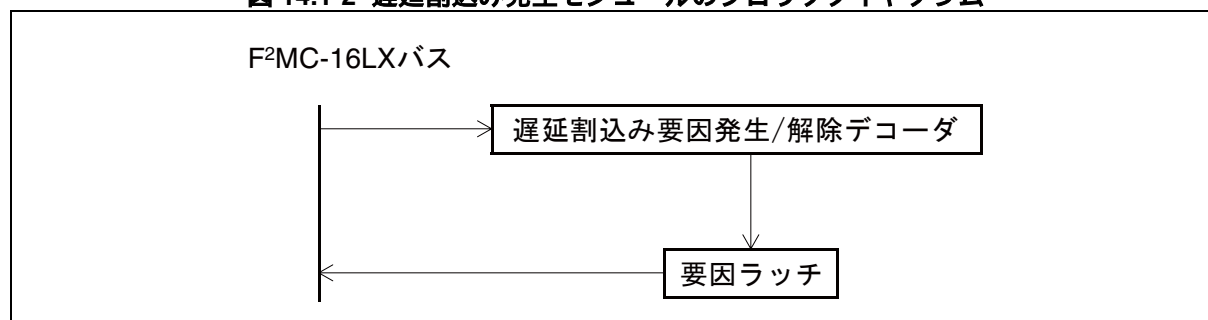
予約ビット領域は "0", "1" どちらの書き込みも結構ですが、将来の拡張を考慮してこのレジスタをアクセスする際にはセットビット、クリアビット命令を使用することをお勧めします。

図 14.1-1 遅延割込み要因発生 / 解除レジスタ (DIRR)

遅延割込み要因発生 / 解除レジスタ							
	bit 15	14	13	12	11	10	9 8
アドレス: 00009F _H	—	—	—	—	—	—	R0
リード/ライト ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)
初期値 ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(0)

■ 遅延割込み発生モジュールのブロックダイアグラム

図 14.1-2 遅延割込み発生モジュールのブロックダイアグラム



14.2 遅延割り込み発生モジュールの動作

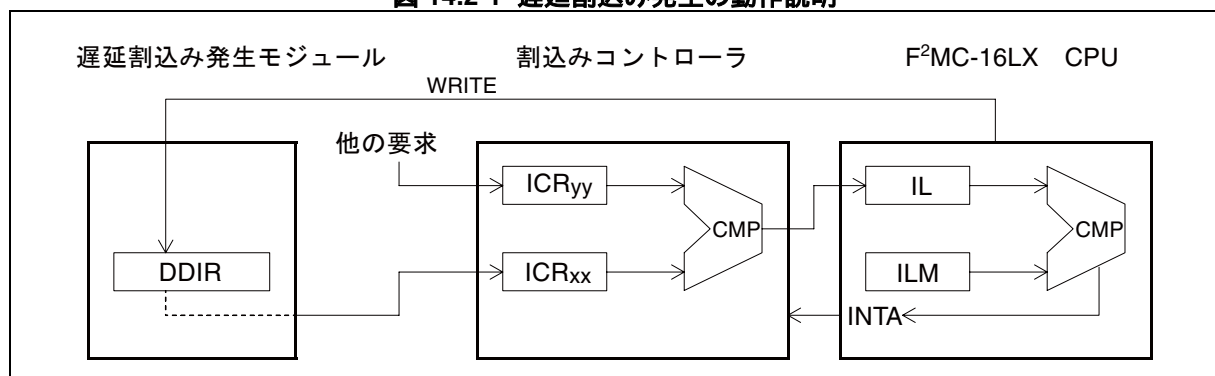
ソフトウェアにて CPU が DIRR レジスタの該当ビットに "1" を書き込むと、遅延割り込み発生モジュールの中にある要求ラッチがセットされ、割り込みコントローラに割り込み要求が発生します。

ほかの割り込み要求が本割り込みより優先順位が低い、あるいはほかの割り込み要求がない場合に、割り込みコントローラは F²MC-16LX CPU に対して割り込み要求が発生します。

■ 遅延割り込み発生モジュールの動作

F²MC-16LX CPU は自分の内部にある CCR レジスタの ILM ビットと割り込み要求を比較し、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、ハードウェア割り込み処理マイクロプログラムを起動します。この結果、本割り込みに対する割り込み処理ルーチンが実行されます。

図 14.2-1 遅延割り込み発生動作説明



割り込み処理ルーチン内で DDIR レジスタの該当ビットの "0" を書き込むことで本割り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。

■ 遅延割り込み要求ラッチの使用上の注意

遅延割り込み要求ラッチは、DIRR レジスタの該当するビットに "1" を書き込むことでセットされ、同じビットに "0" を書き込むことでクリアされます。したがって、割り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り込み処理から復帰した途端に再割り込み処理を起動することになりますので注意してください。

第15章

A/D コンバータ

A/D コンバータの機能と概要について説明します。

- 15.1 A/D コンバータの概要
- 15.2 A/D コンバータのレジスタ
- 15.3 A/D コンバータの動作
- 15.4 変換データ保護機能

15.1 A/D コンバータの概要

A/D コンバータは、アナログ入力電圧をデジタル値に変換するものです。

■ A/D コンバータの概要

A/D コンバータには、次の特長があります。

- 変換時間
1 チャンネル当たり最小 26.3 μ s
- サンプリング時間
1 チャンネル当たり 64 マシンサイクル/4096 マシンサイクル(最小 4 μ s/256 μ s)の選択が可能
なお、変換時間、サンプリング時間は、マシクロック 16MHz 時のものです。
- コンペア時間
1 チャンネル当たり 176 マシンサイクル /352 マシンサイクル
なお、176 マシンサイクルは、マシクロック 8MHz 以下での使用になります。
- サンプル&ホールド回路付 RC 型逐次比較変換方式を採用
- 8 ビット /10 ビットの分解能
- アナログ入力は 8 チャンネルからプログラムで選択
- シングル変換モード
1 チャンネルを選択変換
- スキャン変換モード
連続した複数のチャンネルを変換。最大 8ch プログラム可能
- 連続変換モード
指定チャンネルを繰返し変換
- 停止変換モード
1 チャン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 変換終了の割り込み要求を発生させることができます。この割り込み発生で EI²OS を起動することができ、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、外部トリガ(立下りエッジ)、タイマ(立上りエッジ)の選択。

■ A/D コンバータ使用上の注意

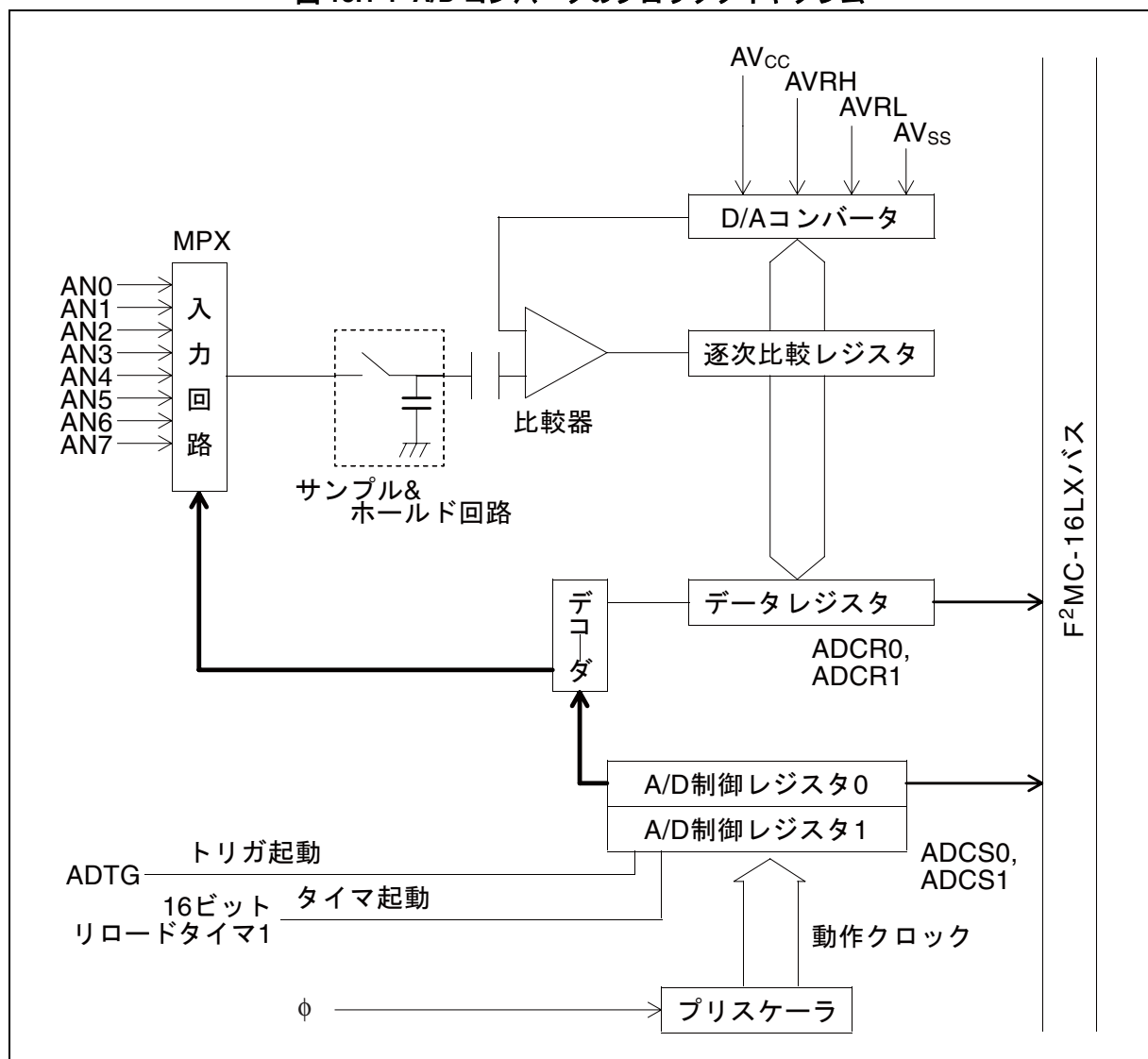
A/D コンバータを外部トリガまたは内部タイマを使って起動する場合 ADCS1 レジスタの A/D 起動要因ビット STS1, STS0 で設定しますが、このときに外部トリガおよび内部タイマの入力値はインアクティブ側の状態で設定してください。アクティブ側にしておくと動作しはじめる場合があります。

STS1, STS0 設定時は、ADTG=1 入力、内部タイマ (16 ビットリロードタイマ)=0 出力の状態で行ってください。

アナログ入力に使用する端子は必ずその対応する ADER レジスタのビットを "1" にしてください。詳細は、「7.3.5 アナログ入力許可レジスタ(ADER)」を参照してください。

■ A/D コンバータのブロックダイアグラム

図 15.1-1 A/D コンバータのブロックダイアグラム



15.2 A/D コンバータのレジスタ

図 15.2-1 に , A/D コンバータのレジスタを示します。

■ A/D コンバータのレジスタ

図 15.2-1 A/D コンバータのレジスタ

コントロールステータスレジスタ上位								
	bit 15	14	13	12	11	10	9	8
アドレス:00003D _H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	予約
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(-)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
コントロールステータスレジスタ下位								
	bit 7	6	5	4	3	2	1	0
アドレス:00003C _H	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
データレジスタ上位								
	bit 15	14	13	12	11	10	9	8
アドレス:00003F _H	S10	ST1	ST0	CT1	CT0	—	D9	D8
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(-)	(R)	(R)
初期値⇒	(0)	(0)	(0)	(0)	(1)	(-)	(X)	(X)
データレジスタ下位								
	bit 7	6	5	4	3	2	1	0
アドレス:00003E _H	D7	D6	D5	D4	D3	D2	D1	D0
リード/ライト⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

15.2.1 コントロールステータスレジスタ (ADCS0, ADCS1)

コントロールステータスレジスタ (ADCS0, 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 (ADCS0, ADCS1)

ADCS0 は、A/D 変換動作中に書換えないようにしてください。

図 15.2-2 コントロールステータスレジスタ (ADCS0, ADCS1)

コントロールステータスレジスタ上位								
	bit 15	14	13	12	11	10	9	8
アドレス:00003D _H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	予約
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(-)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
コントロールステータスレジスタ下位								
	bit 7	6	5	4	3	2	1	0
アドレス:00003C _H	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

< 注意事項 >

ADCS1 は A/D 変換動作中に書き換えないようにしてください。

[bit15] BUSY (Busy flag and stop)

BUSY ビットは、A/D コンバータ動作表示用のビットです。

● 読出し時：

A/D 変換起動でセットされ終了でクリアされます。すなわち、このビットが "0" であれば、A/D 変換停止中であることを示し、"1" であれば、A/D 変換動作中であることを示します。

● 書込み時

A/D 動作中に本ビットに "0" を書き込むと強制的に動作を停止します。連続、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。リードモディファイライト (RMW) 系命令では "1" が読まれます。単発モードでは A/D 変換終了でクリアされます。連発、停止モードでは "0" 書込みで停止するまでクリアされません。

< 注意事項 >

強制停止とソフト起動を同時に行わないでください (BUSY=0, STRT=1)。

[bit14] INT (Interrupt)

INT 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。

INTE ビットが "1" のときに本ビットがセットされると割込み要求が発生します。また EI²OS 起動を許可している場合 EI²OS が起動されます。"1" 書込みは意味を持ちません。

クリアは "0" 書込みと EI²OS 割込みクリア信号で行われます。

リードモディファイライト (RMW) 系命令では "1" が読み出されます。

< 注意事項 >

本ビットの "0" 書込みクリアは A/D 停止中に行ってください。

[bit13] INTE (INTerrupt Enable)

INTE 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。

EI²OS を使用するときは本ビットをセットしてください。EI²OS は割込み要求発生で起動するようになっています。

表 15.2-1 INTE (割込みの許可・不許可指定ビット) の機能

INTE	機能
0	割込みの禁止 [初期値]
1	割込みの許可

[bit12] PAUS (A/D converter PAUSE)

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 EI²OS で転送しなければ前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 EI²OS で転送しなければ次の変換データが格納されないようになっています。この間 A/D 変換動作は停止します。EI²OS で転送を終了すると A/D は変換を再開します。

クリアは、"0" 書込みとリセットにより行われます。

< 注意事項 >

このビットは EI²OS を使用したときのみ有効です。動作説明の変換データ保護機能を参照してください。

[bit11, bit10] STS1, STS0 (Start Source select)

STS1, STS0 ビットの設定により, A/D 起動要因を選択します。

表 15.2-2 STS1, STS0 (A/D 起動要因選択ビット) の機能

STS1	STS0	機能
0	0	ソフト起動 [初期値]
0	1	外部端子トリガでの起動とソフト起動。
1	0	タイマでの起動とソフト起動。
1	1	外部端子トリガ, タイマでの起動とソフト起動

起動が兼用になっているモードでは, いずれかの要因で起動します。起動要因は書換えと同時に変わりますので, A/D 動作中に書換えるときは目的とする変換起動要因がない状態で切り換えてください。

外部端子トリガ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" のときに本ビットを書き換えて外部端子トリガ起動に設定すると A/D が起動する場合があります。

タイマ選択時は, 16 ビットリロードタイマ 1 の出力が選択されます。

[bit9] STRT (StaRT)

STRT ビットに "1" を書き込むことにより 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。

< 注意事項 >

強制停止とソフト起動を同時に行わないでください。(BUSY=0, STRT=1)

[bit8] 予約ビット

bit8 は, 予約ビットです。ADCS1 を設定するときは, 必ず "0" を設定してください。

[bit7, bit6] MD1, MD0 (A/D converter MoDe set)

MD1, MD0 ビットは, 動作モードを設定します。

表 15.2-3 MD1, MD0 の動作モード

MD1	MD0	動作モード
0	0	単発モード, 動作中の再起動は全て可能 [初期値]
0	1	単発モード, 動作中の再起動不可能
1	0	連続モード, 動作中の再起動不可能
1	1	停止モード, 動作中の再起動不可能

● 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 回変換が終了したら停止する。

● 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 変換を繰り返して行います。

● 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 1ch ごとに A/D 変換を行って一時停止する。一時停止中の変換再開は起動要因発生によって行われます。

< 注意事項 >

- ・ 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。
- ・ 停止は BUSY ビットに "0" を書き込むことにより行われます。
- ・ 単発，連続，停止の各モードの再起動の不可能はタイマ，外部トリガ，ソフトすべての起動に適用されます。

[bit5 ~ bit3] ANS2, ANS1, ANS0 (Avalog Start channel set)

ANS2, ANS1, ANS0 ビットは，A/D 変換の開始チャンネルを設定します。

A/D コンバータを起動すると，このビットで選択されたチャンネルから A/D 変換を開始します。

表 15.2-4 ANS2, ANS1, ANS0 ビットの開始チャンネル

ANS2	ANS1	ANS0	開始チャンネル
0	0	0	AN0 [初期値]
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

< 注意事項 >

本ビット群は，A/D 変換中は，変換チャンネル番号が読めますが，A/D 変換停止中は，前に A/D 変換したチャンネル番号が読めます。本ビットの読出し値は，A/D 変換動作が開始するまでは，前回の変換チャンネル番号が読めます。リセット時は，"000" に初期化されます。

[bit2 ~ bit0] ANE2, ANE1, ANE0 (ANalog End channel set)

ANE2, ANE1, ANE0 ビット群より A/D 変換の終了チャンネルを設定します。

表 15.2-5 ANE2, ANE1, ANE0 ビットの終了チャンネル

ANE2	ANE1	ANE0	終了チャンネル
0	0	0	AN0 [初期値]
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

< 注意事項 >

- ANS2 ~ ANS0 と同じチャンネルを設定すると 1ch 変換になります。(シングル変換)
- 連続モード , 停止モードを設定している場合は本ビット群で設定されたチャンネルの変換が終わると ANS2 ~ ANS0 で設定された開始チャンネルに戻ります。
- 設定チャンネルが ANS > ANE の場合は , ANS より変換が始まり , 7ch まで変換したら 0ch に戻り ANE まで変換します。

例 : チャンネル設定 ANS = 6ch, ANE = 3ch で単発モードのとき

動作 変換チャンネル 6ch 7ch 0ch 1ch 2ch 3ch

15.2.2 データレジスタ (ADCR1, ADCR0)

データレジスタ (ADCR1, ADCR0) では , 分解能の選択やマシンサイクルを設定します。

■ データレジスタ (ADCR1, ADCR0)

図 15.2-3 データレジスタ (ADCR1, ADCR0)

データレジスタ上位	bit 15	14	13	12	11	10	9	8	
アドレス:00003F _H	S10	ST1	ST0	CT1	CT0	—	D9	D8	ADCR1
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(-)	(R)	(R)	
初期値⇒	(0)	(0)	(0)	(0)	(1)	(-)	(X)	(X)	
データレジスタ下位	bit 7	6	5	4	3	2	1	0	
アドレス:00003E _H	D7	D6	D5	D4	D3	D2	D1	D0	ADCR0
リード/ライト⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

< 注意事項 >

ADCR0 および ADCR1 の bit9, bit8 のリード値は不定ですが , ADCR1 の bit15 ~ bit11 は常に "0" が読み出されます。

[bit15] S10

S10 は , A/D 分解能選択ビットです。S10 ビットの書換えは必ず変換動作前の A/D 動作が停止の状態でするようにしてください。変換後の書換えをしたとき ADCR の内容は不定となります。

表 15.2-6 S10 の機能

S10	機能
0	10 ビットモード [初期値]
1	8 ビットモード

[bit14, bit13] ST1, ST0 (Sampling Time)

ST1, ST0 ビットでサンプリング時のマシンサイクル数を設定します。

表 15.2-7 ST1, ST0 (サンプリング時のマシンサイクル数設定ビット)

ST1	ST0	サンプリング時マシンサイクル	サンプリング時間
0	0	64 マシンサイクル	4μs/ マシンクロック 16MHz
0	1	予約	
1	0	予約	
1	1	4096 マシンサイクル	256μs/ マシンクロック 16MHz

[bit12, bit11] CT1, CT0 (Compare Time)

CT1, CT0 ビットでコンペア時のマシンサイクル数を設定します。

表 15.2-8 CT1, CT0 (コンペア時のマシンサイクル数設定ビット)

CT1	CT0	コンペア時マシンサイクル	コンペア時間
0	0	176 マシンサイクル	22 μ s/ マシンクロック 8MHz
0	1	352 マシンサイクル	22 μ s/ マシンクロック 16MHz
1	0	予約	
1	1	予約	

< 注意事項 >

- 本ビットを "00" に設定するときは、マシンクロックが 8MHz 以下のときにしてください。
 - 8MHz より高速の場合には変換精度が保証されません。
-

[bit9 ~ bit0] D9 ~ D0

D9 ~ D0 は、A/D 変換格納レジスタで、変換結果であるデジタル値が格納されます。

このレジスタの値は一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。変換データ保護機能が有ります。詳細は「15.3 A/D コンバータの動作」を参照してください。

このレジスタは、リセット時は不定です。

< 注意事項 >

- 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。
 - 8 ビットモード時は、D9, D8 は使用しません。D9, D8 の読出し値は不定です。
-

15.3 A/D コンバータの動作

A/D コンバータは、逐次比較方式で動作し、8 ビット /10 ビットの分解能をもっています。この A/D コンバータは変換結果格納用のレジスタが 1 つ (8 ビット /10 ビット) しかないため、1 回の変換終了とともに変換データレジスタ (ADCR1/ADCR0) が更新されてしまいます。このため、A/D コンバータ単独では連続変換処理には適しませんので EI²OS 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いたします。

■ 単発モード

単発モードは、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 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とき (ANS=ANE) は ANS で指定したチャンネルのみの変換動作になります。

【例】

ANS=000, ANE=011

開始 AN0 AN1 AN2 AN3 終了

ANS=010, ANE=010

開始 AN2 終了

■ 連続モード

連続モードは、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とき (ANS=ANE) は ANS で指定したチャンネルのみの変換を続けます。

【例】

ANS=000, ANE=011

開始 AN0 AN1 AN2 AN3 AN0 繰返し

ANS=010, ANE=010

開始 AN2 AN2 AN2 繰返し

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を繰返し続けます (BUSY ビットに "0" を書き込む 強制動作停止)。

強制動作停止を行うと変換中のものは途中で止まってしまうため、注意してください (強制動作停止した場合、変換レジスタは変換完了した前データが格納されています)。

■ 停止モード

停止モードは、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 1ch 変換する度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起動をかけることにより行われます。

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 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とき (ANS=ANE) は ANS で指定したチャンネルの変換をします。

【例】

ANS=000, ANE=011

開始 AN0 停止 起動 AN1 停止 起動 AN2 停止 起動 AN3 停止
起動 AN0 繰返し

ANS=010, ANE=010

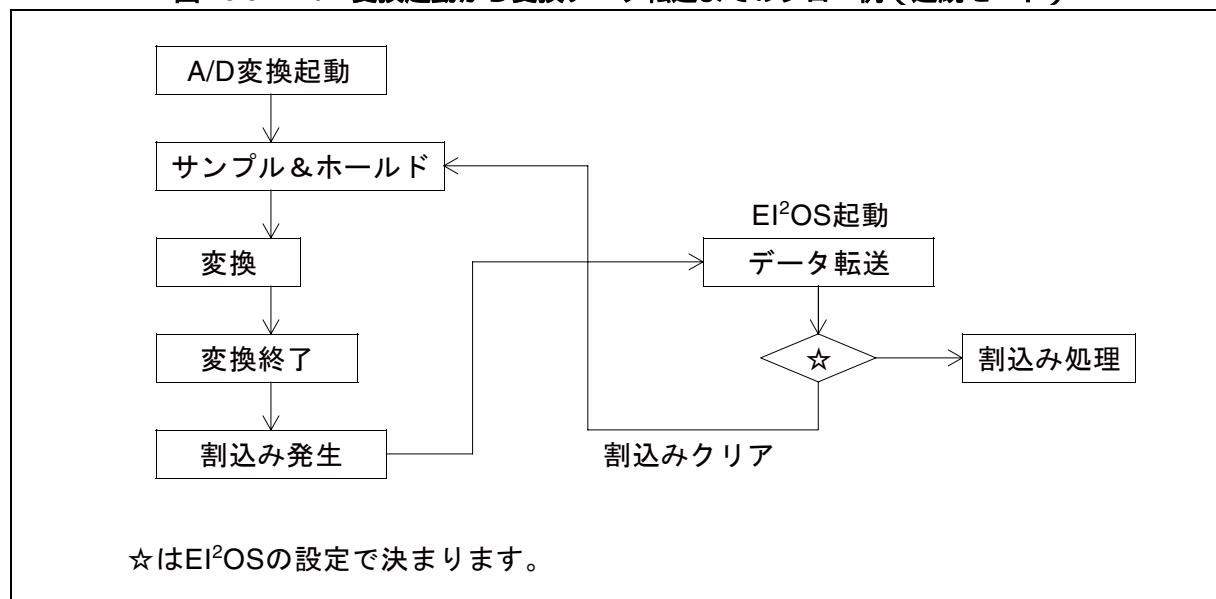
開始 AN2 停止 起動 AN2 停止 起動 AN2 繰返し

このときの起動要因は STS1, STS0 ビット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り変換開始の同期をかけることが可能です。

■ EI²OS を使った変換動作

図 15.3-1 A/D 変換起動から変換データ転送までのフロー例 (連続モード)



15.3.1 単発モード時の EI²OS の起動例

単発モード時の EI²OS は、次の手順で起動します。

- アナログ入力 (AN1 ~ AN3) の変換をして終了する
- 変換データは、200_H ~ 206_H 番地に順に転送する
- 起動はソフトで行う
- 割込みレベル最強

■ 単発モード時の EI²OS の起動例

表 15.3-1 単発モード時の EI²OS の起動例

設定項目	プログラム例	動作説明
EI ² OS の設定	MOV ICR0, #08H	最強割込み設定、割込み時 EI ² OS 起動、ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #02H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送、転送後転送先番地をインクリメントする。I/O からメモリへ転送。
	MOV IOAL, #3EH	A/D コンバータの結果レジスタの設定
	MOV IOAH, #00H	
	MOV DCTL, #03H	EI ² OS 転送を 3 回行う。変換回数と同じにする。
	MOV DCTH, #00H	
A/D コンバータ設定	MOV ADCS0, #0BH	単発モード、開始チャネル AN1、終了チャネル AN3
	MOV ADCS1, #A2H	ソフト起動、A/D 変換開始
その他の処理	-	-
EI ² OS 終了割込みシーケンス	MOV ADCS1, #80H	-
	RETI	割込みからの復帰

ICR3 : 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 : EI²OS ステータスレジスタ
 IOAL : I/O アドレスレジスタ下位
 IOAH : I/O アドレスレジスタ上位
 DCTL : データカウンタ下位
 DCTH : データカウンタ上位

図 15.3-2 単発モード時の EI²OS の起動例

15.3.2 連続モード時の EI²OS の起動例

連続モード時の EI²OS は、次の手順で起動します。

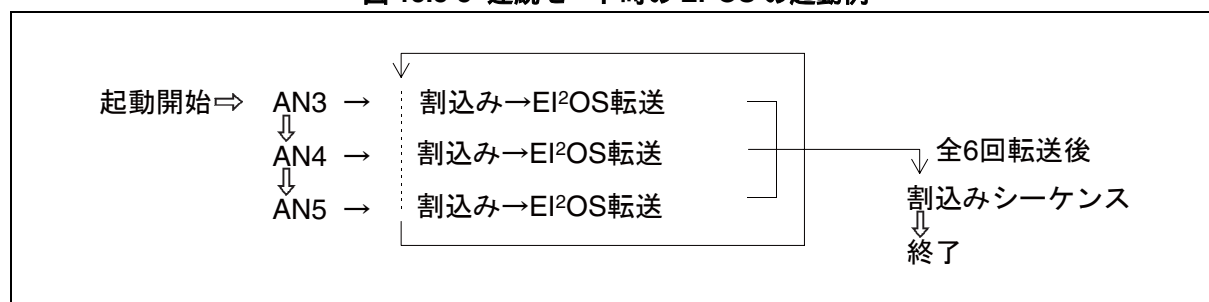
- アナログ入力 (AN3 ~ AN5) の変換をして各チャンネルの変換データを 2 つ取得する。
- 変換データは 600_H ~ 60C_H 番地に順に転送する。
- 起動は外部エッジ入力で行う。
- 割込みレベル最強

■ 連続モード時の EI²OS の起動例

表 15.3-2 連続モード時の EI²OS の起動例

設定項目	プログラム例	動作説明
EI ² OS の設定	MOV ICR0, #08H	最強割込み設定、割込み時 EI ² OS 起動、ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送、転送後転送先番地をインクリメントする。I/O からメモリへ転送。リソースからの要求で終了する。
	MOV IOAL, #3EH	転送元アドレス
	MOV IOAH, #00H	
	MOV DCTL, #06H	EI ² OS 転送を 6 回行う。3ch × 2 分のデータ転送を行う。
	MOV DCTH, #00H	
A/D コンバータ設定	MOV ADCS0, #9DH	単発モード、開始チャンネル AN3、終了チャンネル AN5
	MOV ADCS1, #A4H	外部エッジ起動、A/D 変換開始
その他の処理	-	-
EI ² OS 終了割込みシーケンス	MOV ADCS1, #80H	割込みからの復帰
	RETI	

ICR3 : 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 : EI²OS ステータスレジスタ
 IOAL : I/O アドレスレジスタ下位
 IOAH : I/O アドレスレジスタ上位
 DCTL : データカウンタ下位
 DCTH : データカウンタ上位

図 15.3-3 連続モード時の EI²OS の起動例

15.3.3 停止モード時の EI²OS の起動例

停止モード時の EI²OS は、次の手順で起動します。

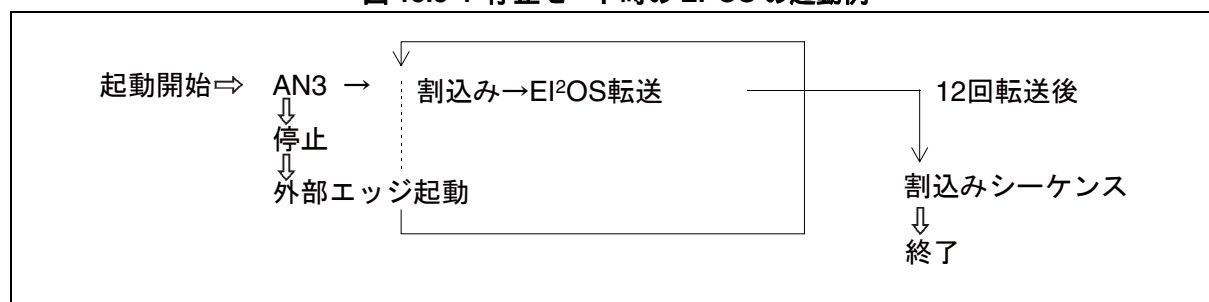
- アナログ入力 (AN3) を一定期間において 12 回変換する。
- 変換データは 600_H ~ 618_H 番地に順に転送する。
- 起動は外部エッジ入力で行う。
- 割込みレベル最強

■ 停止モード時の EI²OS の起動例

表 15.3-3 停止モード時の EI²OS の起動例

設定項目	プログラム例	動作説明
EI ² OS の設定	MOV ICR0, #08H	最強割込み設定, 割込み時 EI ² OS 起動, ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #19H	ワードデータ転送, 転送後転送先番地をインクリメントする。I/O からメモリへ転送。リソースからの要求で終了する。
	MOV IOAL, #3EH	転送元アドレス
	MOV IOAH, #00H	
	MOV DCTL, #0CH	EI ² OS 転送を 12 回行う。
	MOV DCTH, #00H	
A/D コンバータ設定	MOV ADCS0, #DBH	連続モード, 開始チャネル AN3, 終了チャネル AN3 (1ch 変換)
	MOV ADCS1, #A4H	外部エッジ起動, A/D 変換開始
その他の処理	-	-
EI ² OS 終了割込みシーケンス	MOV ADCS1, #80H	割込みからの復帰
	RETI	

ICR3 : 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 : EI²OS ステータスレジスタ
 IOAL : I/O アドレスレジスタ下位
 IOAH : I/O アドレスレジスタ上位
 DCTL : データカウンタ下位
 DCTH : データカウンタ上位

図 15.3-4 停止モード時の EI²OS の起動例

15.4 変換データ保護機能

本 A/D コンバータは、変換データ保護機能を持ち EI²OS を使って連続変換と複数のデータを確保できることを特長としています。

■ 変換データ保護機能

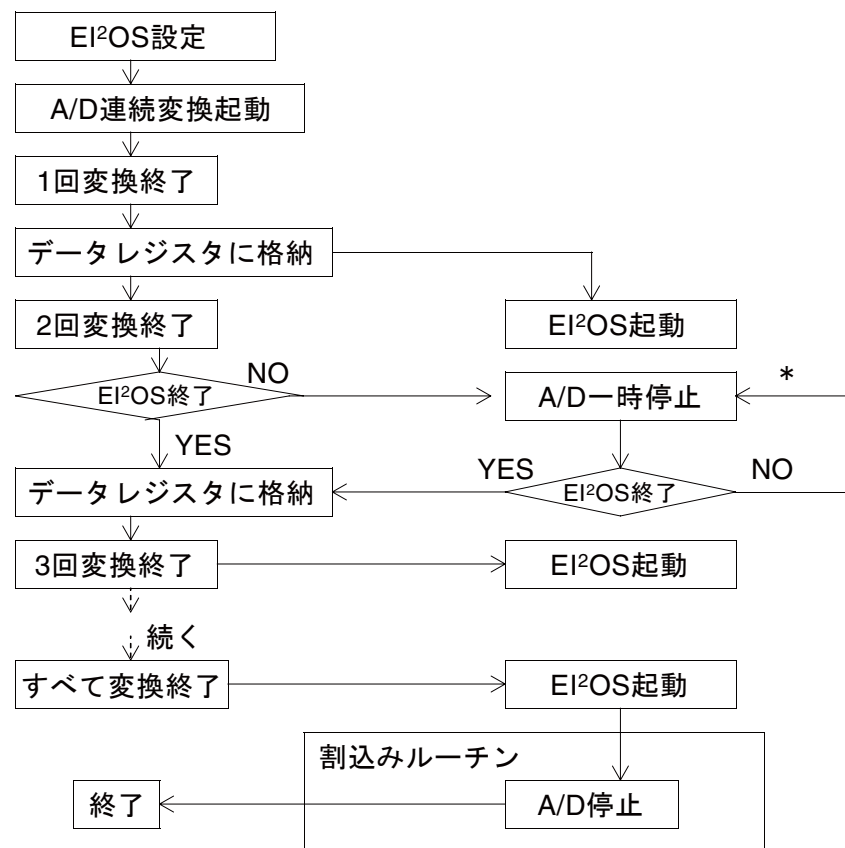
変換データレジスタは 1 つですので連続で A/D 変換をすると 1 回の変換終了と共に変換データが格納されて前データが失われます。これを保護するために、本 A/D コンバータは変換が終了しても前データが EI²OS を使ってメモリへ転送されていないと変換データはレジスタに格納されず A/D は一時停止する機能を持っています。

一時停止の解除は EI²OS でメモリへ転送された後、行われます。

前データが転送されていれば一時停止することなく A/D は連続して変換します。

< 注意事項 >

- この機能は ADCS1 レジスタの INT ビット、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) 状態でしか動作しないようになっています。割込み禁止 (INTE=0) の場合、本機能は動作せず連続で A/D 変換を行った場合は次々に変換データは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 EI²OS 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 A/D 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。
 - EI²OS 動作中で A/D 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、旧データの転送前に新データが書き込まれるときがあります。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。
-

図 15.4-1 データ保護機能フロー (EI²OS を使用時)

- *: ・ 一時停止中に再起動をかけると待機中の変換データは壊れてしまいます。
 ・ A/D コンバータ動作停止時のフローは省略します。

第16章

通信プリスケーラレジスタ

通信プリスケーラレジスタの機能と概要について説明します。

通信プリスケーラの出力は、UART, I/O 拡張シリアルインタフェースで使用されます。

16.1 通信プリスケーラレジスタの概要

16.2 通信プリスケーラレジスタの動作

16.1 通信プリスケアラレジスタの概要

通信プリスケアラレジスタは、マシクロックの分周を制御するレジスタで、種々のマシクロックに対して一定のボーレートが得られるように設計されています。通信プリスケアラの出力は、UART, I/O 拡張シリアルインタフェースで使用されます。

■ 通信プリスケアラレジスタ (CDCR)

図 16.1-1 通信プリスケアラレジスタ (CDCR)

通信プリスケアラレジスタ								
	bit 15	14	13	12	11	10	9	8
アドレス:000027 _H	MD	—	—	—	DIV3	DIV2	DIV1	DIV0
リード/ライト⇒	(R/W)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(0)	(-)	(-)	(-)	(1)	(1)	(1)	(1)

[bit15] MD (Machine clock devide moDe select)

MD は、通信プリスケアラの動作許可ビットです。

表 16.1-1 MD (Machine clock devide moDe select) ビットの機能

MD	機能
0	通信プリスケアラは停止します。[初期値]
1	通信プリスケアラは動作します。

[bit11 ~ bit8] DIV3 ~ DIV0 (DIVide 3 ~ DIVide0)

DIV3 ~ DIV0 は、マシクロックの分周比を決定します。

表 16.1-2 DIV3 ~ DIV0 (DIVide 3 ~ DIVide0) ビットの機能

DIV3	DIV2	DIV1	DIV0	分周比
1	1	1	1	設定禁止 [初期値]
1	1	1	0	2 分周
1	1	0	1	3 分周
1	1	0	0	4 分周
1	0	1	1	5 分周
1	0	1	0	6 分周
1	0	0	1	7 分周
1	0	0	0	8 分周

(注意事項)

- ・ 実際の使用時は、"1111" 以外に設定してください。
- ・ 分周比を変えた場合はクロックの安定時間として 2 周期分の時間を置いてから通信してください。

16.2 通信プリスケアラレジスタの動作

使用するマシクロック ϕ によって、通信プリスケアラレジスタの設定を以下のようになしてください。詳細は、「17.4 UART の動作」、「18.3 I/O 拡張シリアルインタフェースの動作」を参照してください。

■ 通信プリスケアラレジスタの動作

表 16.2-1 通信プリスケアラレジスタの動作

マシクロック ϕ	div	DIV3	DIV2	DIV1	DIV0	$\phi \div \text{div}$
4MHz	4	1	1	0	0	1MHz
6MHz	6	1	0	1	0	
8MHz	8	1	0	0	0	
6MHz	3	1	1	0	1	2MHz
8MHz	4	1	1	0	0	
10MHz	5	1	0	1	1	
12MHz	6	1	0	1	0	
14MHz	7	1	0	0	1	
16MHz	8	1	0	0	0	
8MHz	2	1	1	1	0	4MHz
12MHz	3	1	1	0	1	
16MHz	4	1	1	0	0	

上記以外のマシクロックと div の設定で使用する場合、 $\phi \div \text{div}$ は、最大 4.25MHz を超えないようにしてください。

第17章

UART

UART の機能と動作について説明します。

- 17.1 UART の概要
- 17.2 UART のブロックダイアグラム
- 17.3 UART のレジスタ
- 17.4 UART の動作
- 17.5 UART(モード 1 使用時) の応用例

17.1 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 には、以下の特長があります。

- 全二重ダブルバッファ
- 非同期 (調歩同期), CLK 同期 (I/O 拡張シリアルインタフェース) 通信が可能
- 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 (通信プリスケアラ) 内蔵

表 17.1-1 ボーレート

動作	ボーレート *
非同期	62500/31250/19230/9615/4808/2404/1202 bps
CLK 同期	2M/1M/500k/250k/125k/62.5k bps

*: 内部マシクロックが 6, 8, 10, 12, 16MHz 時の値です。

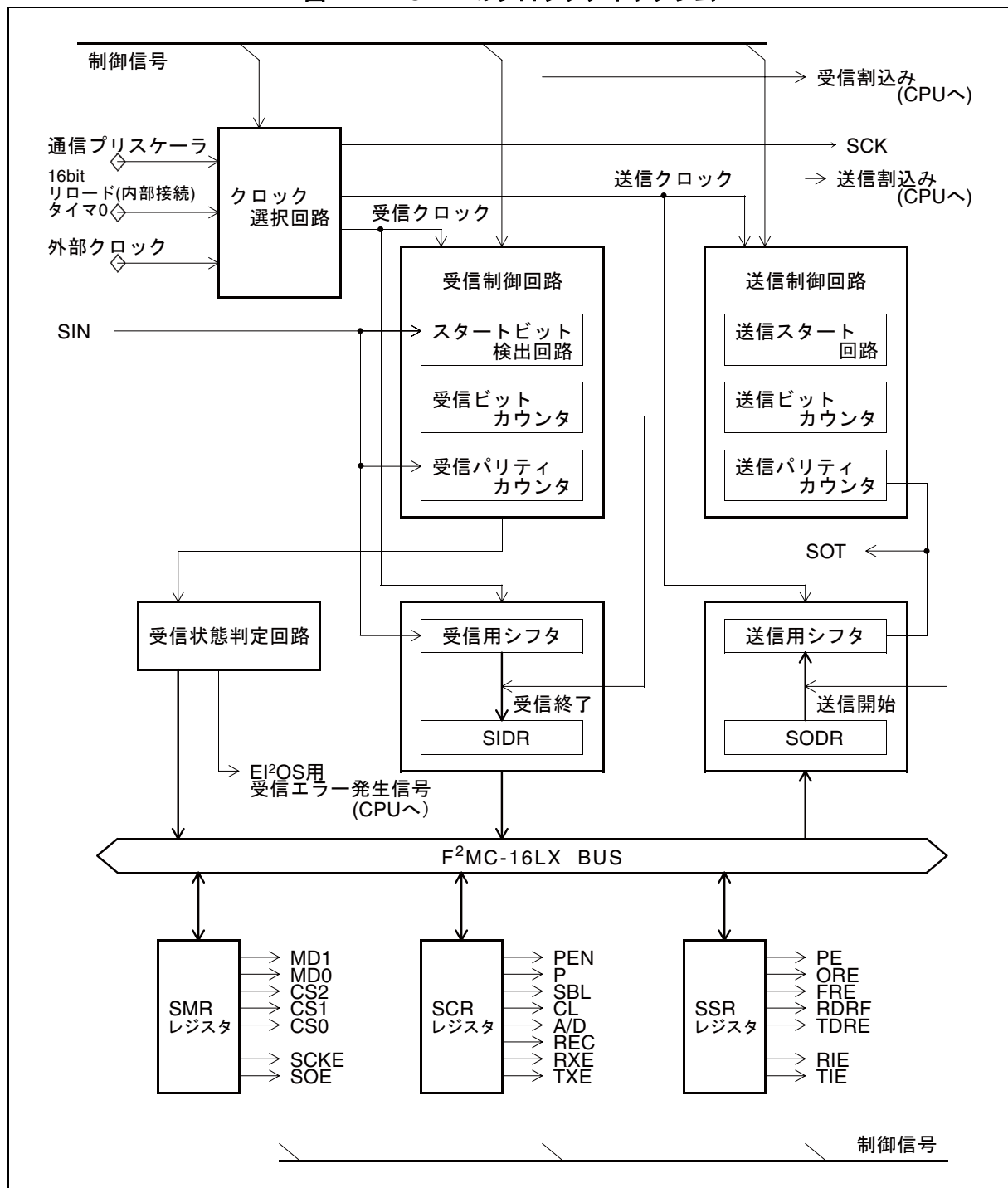
- 外部クロックによる自由なボーレートの設定が可能
- エラー検出機能 (パリティ , フレーミング , オーバラン)
- 転送信号は NRZ 符号
- 拡張インテリジェント I/O サービスのサポート

17.2 UART のブロックダイアグラム

図 17.2-1 に , UART のブロックダイアグラムを示します。

■ UART のブロックダイアグラム

図 17.2-1 UART のブロックダイアグラム



17.3 UART のレジスタ

UART のレジスタには、次の 4 つの種類があります。

- 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
- シリアルインプットレジスタ/シリアルアウトプットレジスタ
- シリアルステータスレジスタ

■ UART のレジスタ

図 17.3-1 UART のレジスタ

シリアルモードレジスタ									
	bit 7	6	5	4	3	2	1	0	
アドレス:000020 _H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	SMR
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)	
シリアルコントロールレジスタ									
	bit 15	14	13	12	11	10	9	8	
アドレス:000021 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	SCR
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)	(1)	(0)	(0)	
シリアルインプットレジスタ/シリアルアウトプットレジスタ									
	bit 7	6	5	4	3	2	1	0	
アドレス:000022 _H	D7	D6	D5	D4	D3	D2	D1	D0	SIDR(read) SODR(write)
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)	
シリアルステータスレジスタ									
	bit 15	14	13	12	11	10	9	8	
アドレス:000023 _H	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	SSR
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)	(1)	(-)	(0)	(0)	

17.3.1 シリアルモードレジスタ (SMR)

SMR レジスタは、UART の動作モードを指定します。
動作モードの設定は動作停止中に行い、動作中には、このレジスタへ書き込まないでください。

■ シリアルモードレジスタ (SMR)

図 17.3-2 シリアルモードレジスタ (SMR) の構成

シリアルモードレジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:000020 _H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE
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)

[bit7, bit6] MD1, MD0 (MoDe select)
MD1, MD0 ビットは、UART の動作モードを選択します。

表 17.3-1 MD1, MD0(動作モードを選択ビット)

MD1	MD0	モード	動作モード
0	0	0	非同期 (調歩同期) ノーマルモード [初期値]
0	1	1	非同期 (調歩同期) マルチプロセッサモード
1	0	2	CLK 同期モード
1	1	-	設定禁止

< 注意事項 >
モード 1 の CLK 非同期モード (マルチプロセッサ) とは、1 台のホスト CPU に数台のスレーブ CPU が接続される使用法です。
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 ~ bit3] CS2, CS1, CS0 (Clock Select)

CS2 ~ CS0 ビットは、ボーレートクロックソースを選択します。

通信プリスケアラを選択した場合には、同時にボーレートも決定されます。

リセットにより、"000" に初期化されます。

表 17.3-2 CS2 ~ CS0 (ボーレートクロックソース選択ビット)

CS2	CS1	CS0	クロック入力
000 _B ~ 100 _B			通信プリスケアラ
1	0	1	予約
1	1	0	内部タイマ (16 ビットリロードタイマ 0)
1	1	1	外部クロック

< 注意事項 >

内部タイマを選択した場合、MB90550A/B では、16 ビットリロードタイマ 0 の出力が選択されます。

[bit2] 予約ビット

bit2 は、予約ビットです。SMR を設定するときは、必ず "0" に設定してください。

[bit1] SCKE (SCIK Enable)

SCKE ビットは、CLK 同期モード (モード 2) で通信する場合、SCK 端子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。

表 17.3-3 SCKE (SCIK Enable) ビットの機能

SCKE	機能
0	クロック入力端子として機能します。[初期値]
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うには、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)

SOE 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 (SOT) 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。

表 17.3-4 SOE (Serial Output Enable) ビットの機能

SOE	機能
0	汎用 I/O ポート端子として機能します。[初期値]
1	シリアルデータ出力端子 (SOT) として機能します。

17.3.2 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) は、シリアル通信する場合の転送プロトコルを制御します。

■ シリアルコントロールレジスタ (SCR)

図 17.3-3 シリアルコントロールレジスタ (SCR) の構成

シリアルコントロールレジスタ									
	bit 15	14	13	12	11	10	9	8	
アドレス:000021 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	SCR
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)	(1)	(0)	(0)	

[bit15] PEN (Parity ENnable)

PEN ビットは、シリアル通信において、パリティを付加してデータ通信を行うかどうかを指定します。

表 17.3-5 PEN (Parity ENable) ビットの機能

PEN	機能
0	パリティなし [初期値]
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、パリティを付加することはできません。

[bit14] P (Parity)

P ビットは、パリティを付加してデータ通信を行うとき、偶数 / 奇数パリティを指定します。

表 17.3-6 P (偶数 / 奇数パリティ指定ビット)

P	機能
0	偶数パリティ [初期値]
1	奇数パリティ

[bit13] SBL (Stop Bit Length)

SBL 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。

表 17.3-7 SBL (ストップビット長指定ビット)

SBL	機能
0	1 ストップビット [初期値]
1	2 ストップビット

[bit12] CL (Character Length)

CL 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。

表 17.3-8 CL (送受信データ長指定ビット)

CL	機能
0	7 ビットデータ [初期値]
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、8 ビットデータとしてください。

[bit11] A/D (Address/Data)

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。

表 17.3-9 A/D (Address/Data) ビットの機能

A/D	機能
0	データフレーム [初期値]
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、読出し値は常に "1" になります。

[bit9] RXE (Receiver Enable)

UART の受信動作を制御します。

表 17.3-10 RXE (Receiver Enable) ビット

RXE	機能
0	受信動作を禁止します。[初期値]
1	受信動作を許可します。

< 注意事項 >

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を完了し受信データバッファ SISR レジスタに受信データをストアしたところで受信動作を停止します。

[bit8] TXE (Transmitter Enable)

TXE ビットは , UART の送信動作を制御します。

表 17.3-11 送信動作制御ビット (TXE)

TXE	機能
0	送信動作を禁止します。[初期値]
1	送信動作を許可します。

< 注意事項 >

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は , 送信データバッファ SODR レジスタにデータがなくなった後に送信動作を停止します。

17.3.3 シリアルインプットデータレジスタ (SIDR) / シリアルアウトプットデータレジスタ (SODR)

シリアルインプットデータレジスタ (SIDR)/ シリアルアウトプットデータレジスタ (SODR) は、受信 / 送信用のデータバッファレジスタです。

■ シリアルインプットデータレジスタ (SIDR)/シリアルアウトプットデータレジスタ (SODR)

SIDR, SODR のデータ長が 7bit の場合は、上位 1bit (D7) は無効データとなります。SODR レジスタへの書込みは、SSR レジスタの TDRE が "1" のときに書き込んでください。

図 17.3-4 シリアルインプットデータレジスタ (SIDR)/ シリアルアウトプットデータレジスタ (SODR) の構成

シリアルインプットレジスタ/シリアルアウトプットレジスタ									
	bit	7	6	5	4	3	2	1	0
アドレス: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)
									SIDR(read) SODR(write)

< 注意事項 >

このアドレスへのライトは SODR レジスタへの書込みを、リードは SIDR レジスタの読出しを意味します。

17.3.4 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) は , UART の動作状態を表すフラグで構成されています。

■ シリアルステータスレジスタ (SSR)

図 17.3-5 シリアルステータスレジスタ (SSR) の構成

シリアルステータスレジスタ									
bit 15 14 13 12 11 10 9 8									
アドレス:000023 _H	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	SSR
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)	(1)	(-)	(0)	(0)	

[bit15] PE (Parity Error)

PE 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。

本ビットがセットされた場合には , SISR レジスタのデータは無効データとなります。

表 17.3-12 PE (Parity Error) ビットの機能

PE	機能
0	パリティエラーなし [初期値]
1	パリティエラーが発生

[bit14] ORE (Over Run Error)

ORE 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。

本ビットがセットされた場合には , SISR レジスタのデータは無効データとなります。

表 17.3-13 ORE (Over Run Error) の機能

ORE	機能
0	オーバランエラーなし [初期値]
1	オーバランエラー発生

[bit13] FRE (FRaming Error)

FRE 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。本ビットがセットされた場合には、SIDR レジスタのデータは無効データとなります。

表 17.3-14 FRE (FRaming Error) の機能

FRE	機能
0	フレーミングエラーなし [初期値]
1	フレーミングエラー発生

[bit12] RDRF (Receiver Data Register Full)

RDRF 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。

表 17.3-15 RDRF (Receiver Data Register Full) の機能

RDRF	機能
0	受信データなし [初期値]
1	受信データあり

[bit11] TDRE (Transmitter Data Register Empty)

TDRE 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、次の送信データデータを書き込めることを表します。

表 17.3-16 TDRE (Transmitter Data Register Empty) の機能

TDRE	設定
0	送信データの書込み禁止
1	送信データの書込み許可 [初期値]

[bit9] RIE (Receiver Interrupt Enable)

RIE ビットは、受信割込みを制御します。

表 17.3-17 RIE (Receiver Interrupt Enable) の機能

RIE	機能
0	割込みを禁止します。 [初期値]
1	割込みを許可します。

< 注意事項 >

受信割込み要因は , PE, ORE, FRE によるエラー発生のほか , RDRF による正常受信があります。

[bit8] TIE (Transmitter Interrupt Enable)

TIE ビットは , 送信割込みを制御します。

表 17.3-18 TIE (Transmitter Interrupt Enable) の機能

TIE	機能
0	割込みを禁止します。[初期値]
1	割込みを許可します。

< 注意事項 >

送信割込み要因は , TDRE による送信要求があります。

17.4 UART の動作

UART は、表 17.4-1 に示す動作モードを持ち、SMR レジスタ、SCR レジスタに値を設定することによりモードを切り換えることができます。

■ UART の動作

表 17.4-1 UART の動作モード

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	有 / 無	7 ビット	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
	有 / 無	8 ビット		
1	無	8 ビット +1 ビット	非同期（調歩同期）マルチプロセッサ	
2	無	8 ビット	CLK 同期モード	無

< 注意事項 >

- 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。上記モード以外では動作しませんので、設定しないでください。
- CLK 同期モードは、クロック制御（I/O 拡張シリアルインタフェース）方式を採用しています。CLK 同期モードでは、スタート・ストップビットはデータに付加されません。
- 通信モードの設定は、動作停止中に行ってください。モード設定時の送受信したデータは保証されません。

■ 拡張インテリジェント I/O サービス (EI²OS)

EI²OS につきましては、「3.6 拡張インテリジェント I/O サービス (EI²OS)」を参照してください。

17.4.1 UART のクロック選択

UART のクロックは、次の 3 種類から選択できます。

- 通信プリスケアラ
- 内部タイマ
- 外部クロック

■ 通信プリスケアラ

通信プリスケアラを選択したときのボーレートは、次のようになります。

表 17.4-2 ボーレート（非同期時（調歩同期））

CS2	CS1	CS0	$\phi \div \text{div}=2\text{MHz}$	$\phi \div \text{div}=4\text{MHz}$	算出式
0	0	0	9615 bps	19230 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2)$
0	0	1	4808 bps	9615 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^2)$
0	1	0	2404 bps	4808 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^3)$
0	1	1	1202 bps	2404 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^4)$
1	0	0	31250 bps	62500 bps	$(\phi \div \text{div}) / 2^6$

ϕ : マシンクロック

表 17.4-3 ボーレート（CLK 同期時）

CS2	CS1	CS0	$\phi \div \text{div}=2\text{MHz}$	$\phi \div \text{div}=4\text{MHz}$	算出式
0	0	0	1M bps	2M bps	$(\phi \div \text{div}) / 2$
0	0	1	500 kbps	1M bps	$(\phi \div \text{div}) / 2^2$
0	1	0	250 kbps	500 kbps	$(\phi \div \text{div}) / 2^3$
0	1	1	125 kbps	250 kbps	$(\phi \div \text{div}) / 2^4$
1	0	0	62.5 kbps	125 kbps	$(\phi \div \text{div}) / 2^5$

ϕ : マシンクロック

div: 通信プリスケアラの設定（詳細は、「第 16 章 通信プリスケアラレジスタ」を参照）

■ 内部タイマ

SMR レジスタの CS2 ~ CS0 ビットを "110" に設定して内部タイマを選択した場合は、16 ビットタイマ (タイマ 0) をリロードモードで動作させます。このときのボーレートの算出式は、次のとおりです。

非同期 (調歩同期) $(\phi \div N) / (16 \times 2 \times (n+1))$

CLK 同期 $(\phi \div N) / (2 \times (n+1))$

ϕ : マシニングロック

N: タイマのカウントクロックソース

n: タイマのリロード値

マシニングロックを 7.3728MHz としたときのボーレートとリロード値の関係を、表 17.4-4 に示します。

表 17.4-4 ボーレートとリロード値

ボーレート (bps)		リロード値	
非同期	CLK 同期	N=2 ¹ (マシニングロックの 2 分周)	N=2 ³ (マシニングロックの 8 分周)
38400	614400	2	-
19200	307200	5	-
9600	153600	11	2
4800	76800	23	5
2400	38400	47	11
1200	19200	95	23
600	9600	191	47
300	4800	383	95

ボーレートクロックソースとして内部タイマ (16 ビットリロードタイマ 0) を選択した場合には、16 ビットタイマ 0 の出力 TOT0 は本コントローラ内部で既に接続されています。したがって、16 ビットタイマ 0 の外部端子 TOT0 から本 UART の外部クロック入力端子 SCK へ外部で接続する必要はありません。また、タイマ 0 の出力端子をほかに使用していなければ I/O ポート端子として使用することができます。

■ 外部クロック

SMR レジスタの CS2 ~ CS0 ビットを "111" に設定して外部クロックを選択した場合のボーレートは、外部クロックの周波数を f とすると、次のようになります。

非同期 (調歩同期) : $f/16$

CLK 同期 : f

ただし、f は最大 2MHz までです。

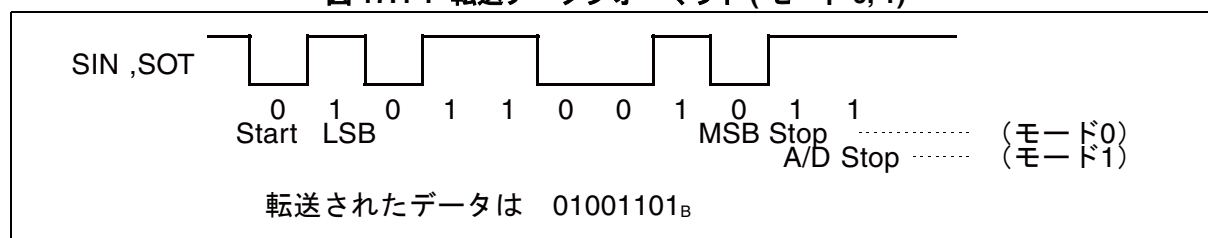
17.4.2 非同期 (調歩同期) モード

非同期 (調歩同期) モードでは、転送データは必ずスタートビット ("L" レベルデータ) から始まり、ストップビット ("H" レベルデータ) で終了します。

■ 転送データフォーマット

UART は、NRZ (Non Return to Zero) 形式のデータのみを扱います。図 17.4-1 に、転送データフォーマットを示します。

図 17.4-1 転送データフォーマット (モード 0, 1)



転送データは必ずスタートビット ("L" レベルデータ) より始まり、LSB ファーストで指定されたデータビット長転送が行われ、ストップビット ("H" レベルデータ) で終了します。外部クロックを選択している場合は、常にクロックを入力してください。

ノーマルモード (モード 0) ではデータ長を 7 ビットまたは 8 ビットに設定することができますが、マルチプロセッサモード (モード 1) では 8 ビットでなければなりません。また、マルチプロセッサモードではパリティを付加することはできません。そのかわり、A/D ビットが必ず付加されます。

■ 受信動作

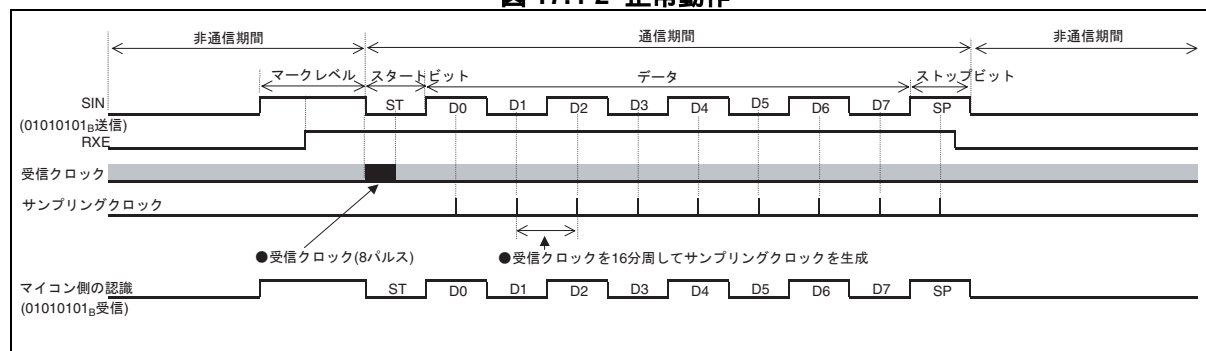
SCR レジスタの RXE ビットが "1" ならば、常に受信動作が行われています。スタートビットを検出すると、SCR レジスタで決められたデータフォーマットに従い 1 フレームデータの受信が行われます。1 フレームの受信が終わると、エラーが発生した場合にはエラーフラグのセットが行われた後 SSR レジスタの RDRF フラグがセットされます。このとき同じ SSR レジスタの RIE ビットが "1" にセットされていれば CPU に対して受信割り込みが発生します。SSR レジスタの各フラグを調べ、正常受信なら SIDR レジスタをリードして、エラーが発生していれば必要なエラー処理を行うようにしてください。RDRF フラグは、SIDR レジスタをリードするとクリアされます。

■ スタートビット検出方法

スタートビットを検出するには、以下のように設定してください。

- 通信期間直前は通信線を必ず "H" (マークレベルを付加) にしてください。
- 通信線が "H" (マークレベル) の期間に、受信許可 (RXE=H) にしてください。
- 非通信期間中 (マークレベルを除く) は、受信許可 (RXE=H) にしないでください。正しいデータが受信できません。
- ストップビット検出後 (RDRF フラグが "1" にセットされた後)、通信線が "H" (マークレベル) の間に受信禁止 (RXE=L) にしてください。

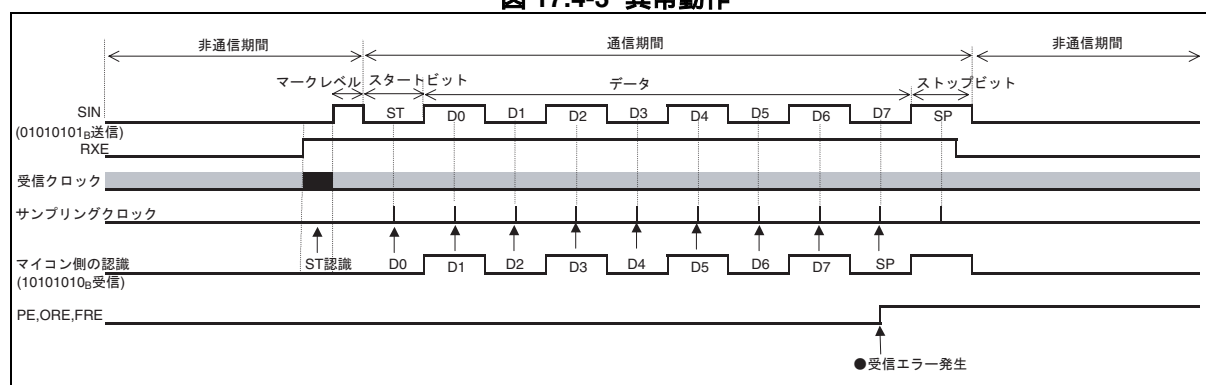
図 17.4-2 正常動作



以下の例のようなタイミングで受信許可に設定しますとマイコン側で入力データ (SIN) が正しく認識されませんので注意してください。

- 通信線が "L" の期間に、受信許可 (RXE=H) に設定した場合の動作例

図 17.4-3 異常動作



■ 送信動作

SSR レジスタの TDRE フラグが "1" のとき、SODR レジスタに送信データを書き込みます。ここで、SCR レジスタの TXE ビットが "1" なら送信が行われます。

SODR レジスタにセットしたデータが送信用シフトレジスタにロードされて送信が開始されると TDRE フラグが再びセットされ、次の送信データをセットできるようになります。このとき同じ SSR レジスタの TIE ビットが "1" にセットされていれば CPU に対して送信割り込みが発生して、SODR レジスタ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 レジスタにデータをセットすると一度クリアされます。

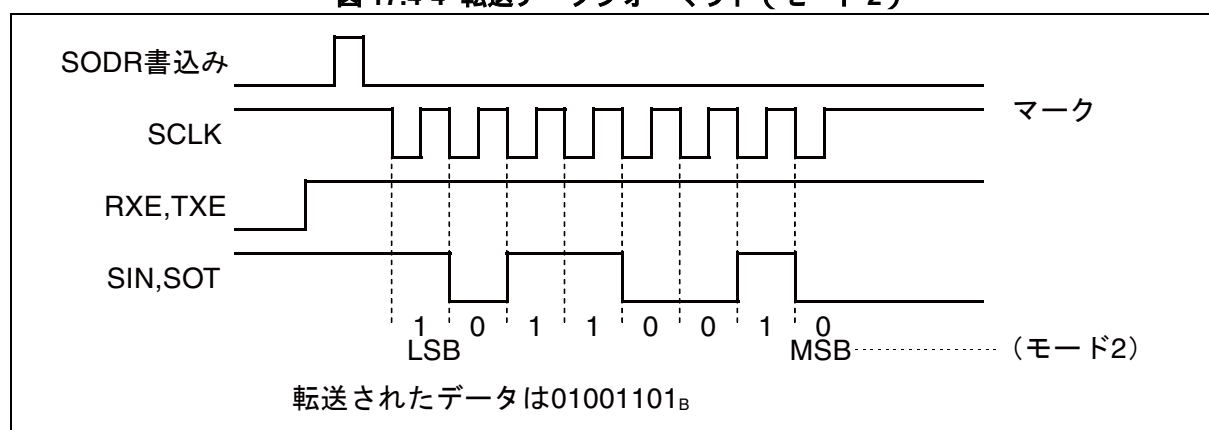
17.4.3 CLK 同期モード

CLK 同期モードでは、内部クロックを選択した場合は、データ受信用同期クロックが自動的に生成され、外部クロックを選択場合は、1 バイト分のクロックを供給する必要があります。

■ 転送データフォーマット

UART は、NRZ (Non Return to Zero) 形式のデータのみを扱います。図 17.4-4 に、送受信クロックとデータとの関係を示します。

図 17.4-4 転送データフォーマット (モード 2)



内部クロック (通信プリスケラまたは内部タイマ) を選択している場合は、データを送信するとデータ受信用同期クロックが自動的に生成されます。

外部クロックを選択している場合は、送信側 UART の送信用データバッファ SODR レジスタにデータがあること (TDRE フラグが "0") を確かめた後、正確に 1 バイト分のクロックを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" にしてください。

データは 8 ビットのみとなり、パリティを付加することはできません。また、スタート / ストップビットがないのでオーバーランエラー以外のエラー検出は行われません。

■ 初期化

CLK 同期モードを使用する場合の、各制御レジスタの設定値を示します。

表 17.4-5 CLK 同期モードを使用する場合の、各制御レジスタの設定値

レジスタ名	ビット名	設定
SMR レジスタ	MD1, MD0	"10 _B "
	CS2, CS1, CS0	クロック入力を指定
	SCKE	通信プリスケータまたは内部タイマの場合 "1" 外部クロックの場合 "0"
	SOE	送信を行う場合 "1", 受信のみの場合 "0"
SCR レジスタ	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)
	RXE, TXE	少なくとも、どちらか一方を "1"
SSR レジスタ	RIE	割込みを使用する場合 "1", 割込みを使用しない場合 "0"
	TIE	"0"

■ 通信開始

SODR レジスタへの書き込みによって通信を開始します。受信のみの場合でも、必ず仮の送信データを SODR レジスタに書き込む必要があります。

■ 通信終了

SSR レジスタの RDRF フラグが "1" に変化したことにより確認できます。

SSR レジスタの ORE ビットによって、通信が正常に行われたかを判断してください。

17.4.4 割り込み発生およびフラグのセットタイミング

UART には、5 つのフラグと 2 つの割り込み要因を持ちます。

5 つのフラグとは、PE/ORE/FRE/RDRF/TDRE で、2 つの割り込み要因は、受信用のものと送信用のものです。

■ 5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割り込み要因

● PE(パリティエラー)/ORE(オーバランエラー)/FRE(フレーミングエラー)

受信時エラーが発生したときにセットされ、SCR レジスタの REC ビットに "0" を書き込むとクリアされます。

● RDRF

受信データが SDR レジスタにロードされるとセットされ、SDR レジスタを読み出すことでクリアされます。ただし、モード 1 ではパリティ検出機能、モード 2 ではパリティ検出機能とフレーミングエラー検出機能はありません。

● TDRE

SODR レジスタが空になり書き込み可能な状態になるとセットされ、SODR レジスタへ書き込むとクリアされます。

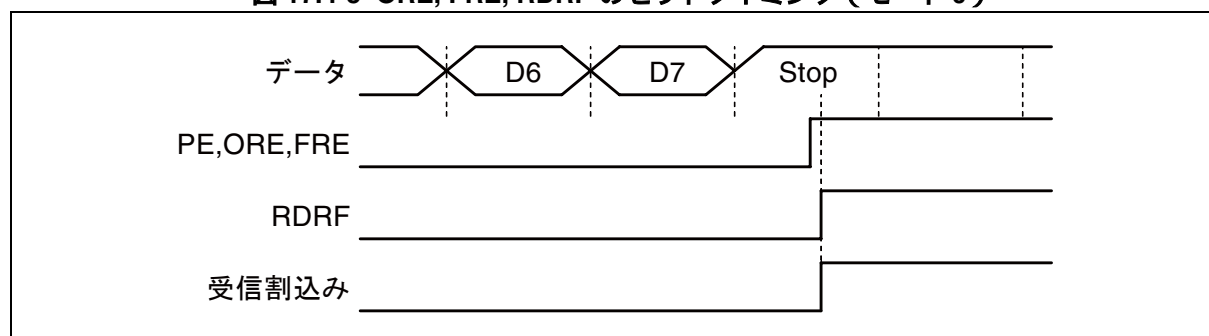
2 つの割り込み要因は、受信用のものと送信用のものがあります。受信時は、PE/ORE/FRE/RDRF により割り込みを要求します。送信時は、TDRE により割り込みを要求します。

■ 各動作モードによる割り込みフラグのセットタイミング

● モード 0 の受信動作時

PE, ORE, FRE, RDRF は受信転送が終了し、最後のストップビットを検出するときにフラグがセットされ、CPU への割り込み要求が発生します。PE, ORE, FRE がアクティブ時は、SDR のデータは無効データとなります。

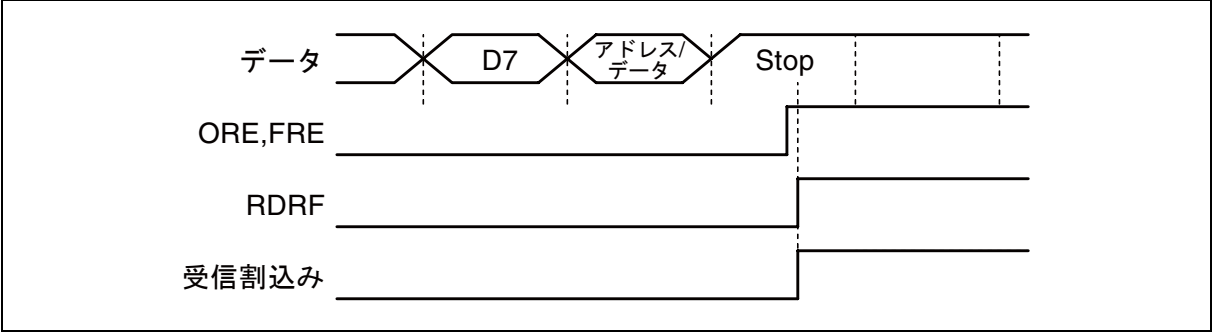
図 17.4-5 ORE, FRE, RDRF のセットタイミング (モード 0)



● モード 1 の受信動作時

ORE, FRE, RDRF は受信転送が終了し、最後のストップビットを検出するときにフラグがセットされ、CPU への割込み要求が発生します。また、受信可能なデータ長は 8bit のため、最後の 9bit 目のアドレス / データを示すデータは無効データとなります。ORE, FRE がアクティブ時は、SIDR レジスタのデータは無効データとなります。

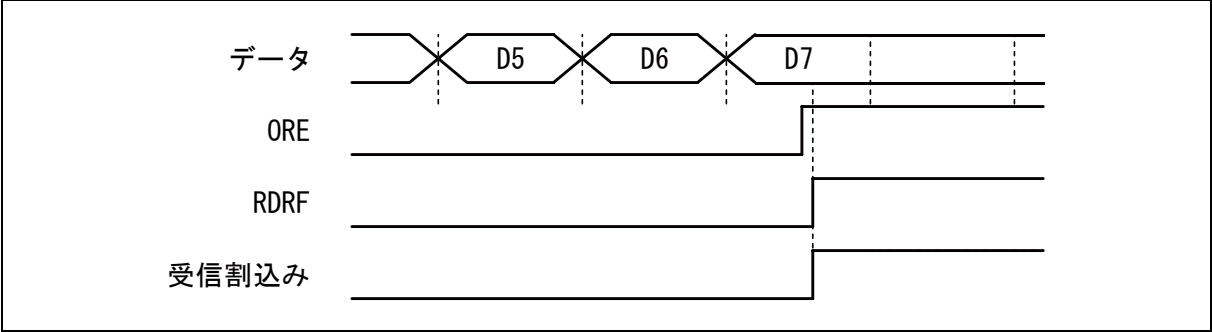
図 17.4-6 ORE, FRE, RDRF のセットタイミング (モード 1)



● モード 2 の受信動作時

ORE, RDRF は受信転送が終了し、最後のデータ (D7) を検出するときにフラグがセットされ、CPU への割込み要求が発生します。ORE がアクティブ時は、SIDR レジスタのデータは無効データとなります。

図 17.4-7 ORE, RDRF のセットタイミング (モード 2)



● モード 0, モード 1, モード 2 の送信動作時

TDRE は SODR レジスタへ書き込まれるとクリアされ、内部のシフトレジスタに転送され次のデータ書込みが可能になるとセットされ、CPU への割込み要求が発生します。送信動作中に SCR レジスタの TXE に "0" (モード 2 のときは RXE も含む) を書き込むと、SSR レジスタの TDRE が "1" となり、送信用のシフトレジスタが停止してから UART の送信動作を禁止します。送信動作中に SCR レジスタの TXE に "0" (モード 2 のときは RXE も含む) を書き込んだ後、送信が停止する前に SODR レジスタへ書き込まれたデータは送信されます。

図 17.4-8 TDRE のセットタイミング (モード 0, 1)

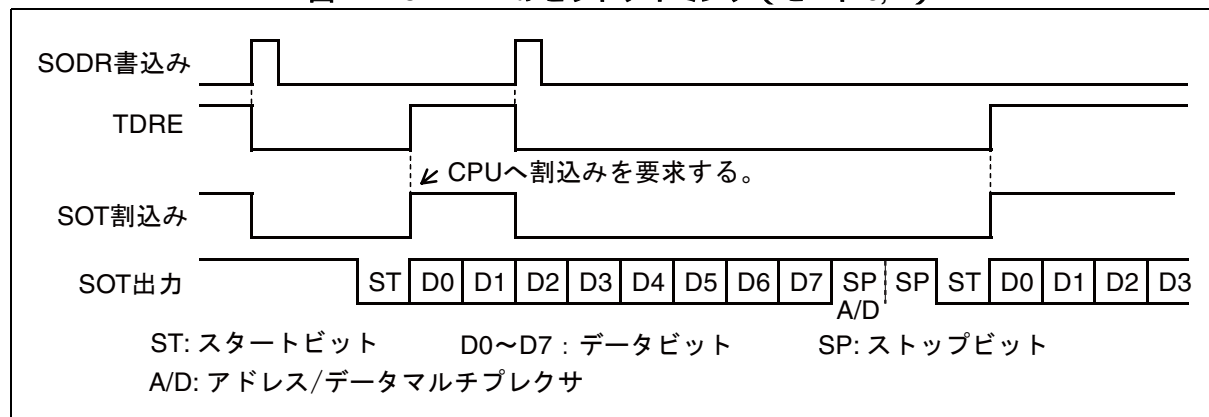
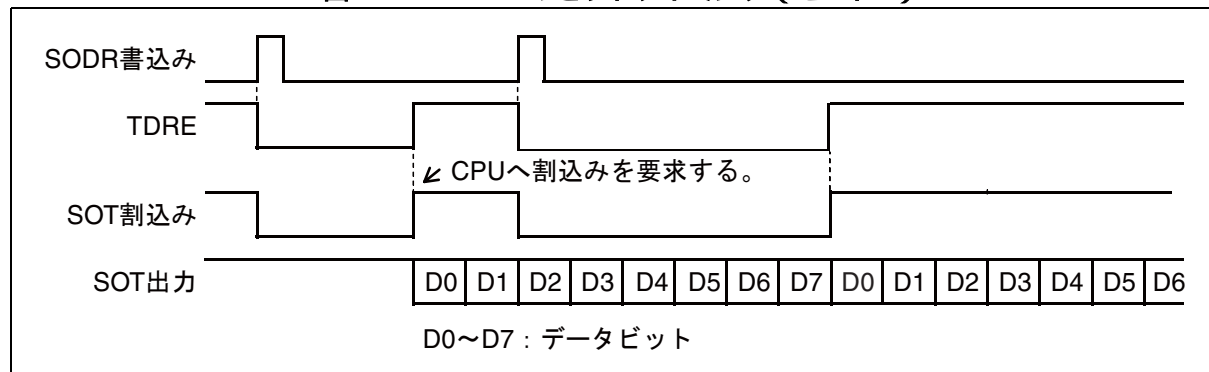


図 17.4-9 TDRE のセットタイミング (モード 2)



17.5 UART(モード 1 使用時) の応用例

モード 1 は、1 台のホスト CPU に数台のスレーブ CPU が接続されるような場合に使用されます。このリソースでは、ホスト側の通信インタフェースのみサポートします (図 17.5-1 を参照)。

■ UART(モード 1 使用時) の応用例

図 17.5-1 モード 1 を使用時のシステム構築例

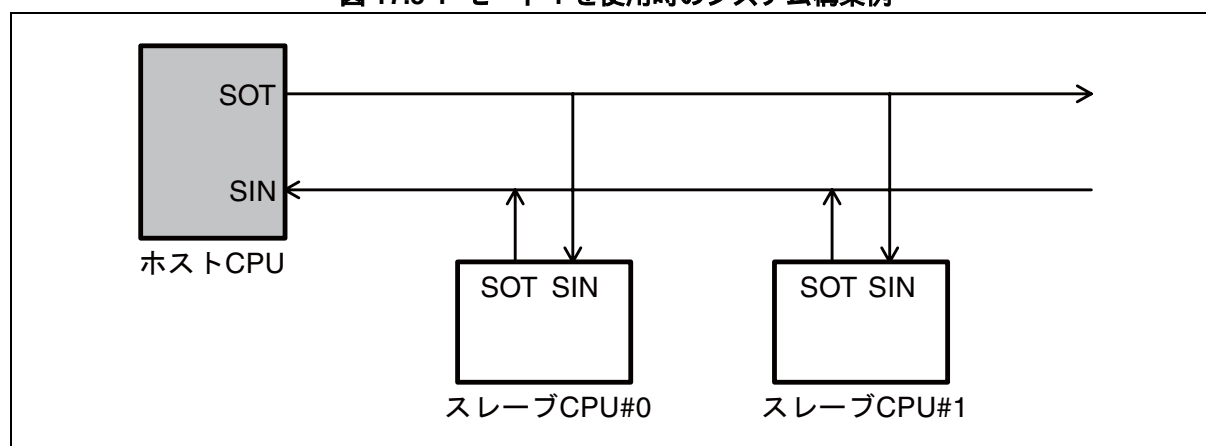
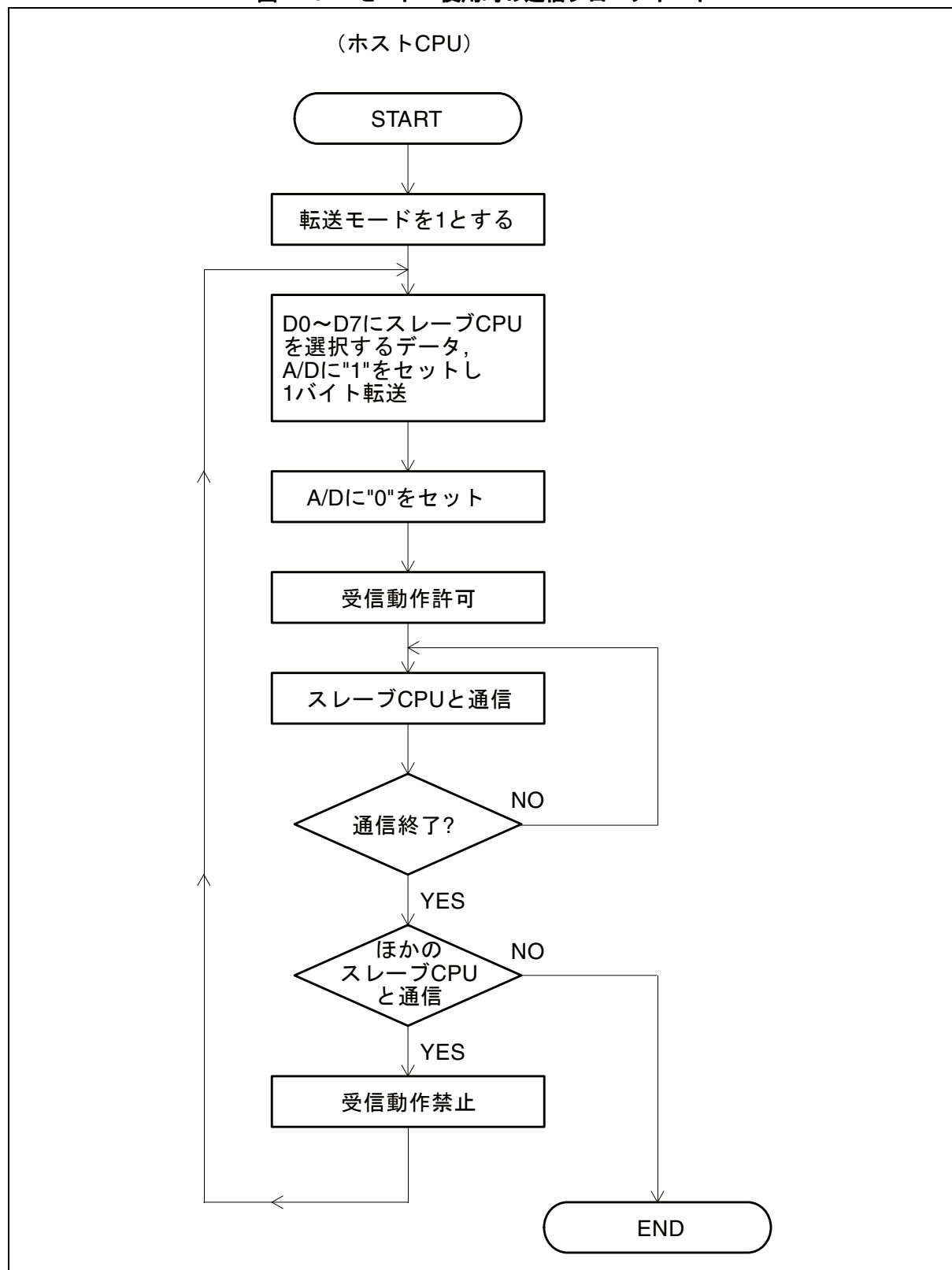


図 17.5-2 に示すように、通信は、ホスト CPU がアドレスデータを転送することによって始まります。アドレスデータとは、SCR レジスタの A/D が "1" のときのデータで、それにより通信先となるスレーブ CPU が選択され、ホスト CPU との通信が可能になります。通常データは、SCR レジスタの A/D が "0" のときのデータです。

このモードにおいては、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 ビットは "0" としてください。

図 17.5-2 モード 1 使用時の通信フローチャート



第18章

I/O 拡張シリアルインタ フェース

I/O 拡張シリアルインタフェースの機能と動作について説明します。

18.1 I/O 拡張シリアルインタフェースの概要

18.2 I/O 拡張シリアルインタフェースのレジスタ

18.3 I/O 拡張シリアルインタフェースの動作

18.1 I/O 拡張シリアルインタフェースの概要

I/O 拡張シリアルインタフェースは、8bit × 1ch. 構成のクロック同期式によるデータ転送可能なシリアル 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可能です。

■ I/O 拡張シリアルインタフェースの概要

I/O 拡張シリアルインタフェースの動作モードには、以下の 2 種類があります。

- 内部シフトクロックモード

内部クロック（通信プリスケアラ）に同期してデータを転送します。

- 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を転送します。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。

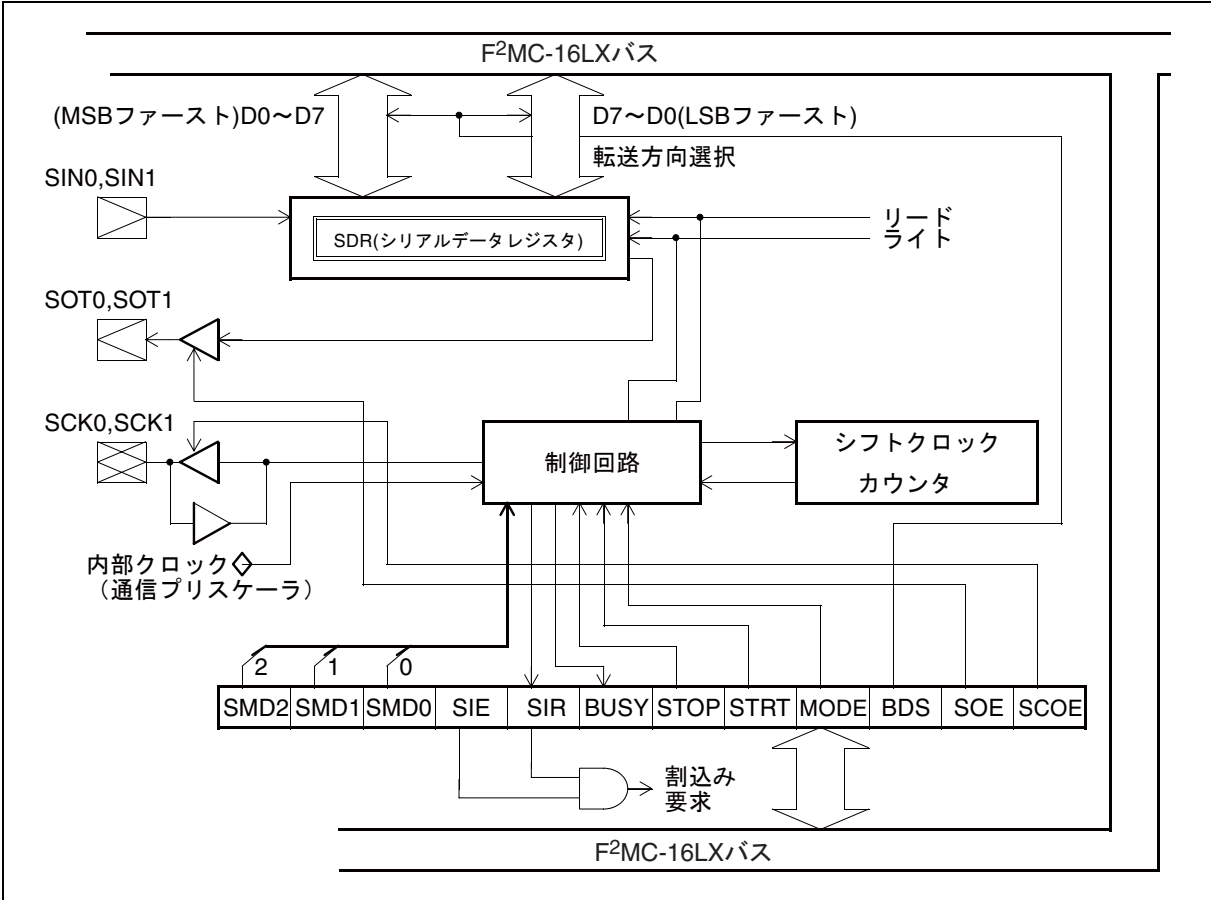
本シリーズは、I/O 拡張シリアルインタフェースを 2ch 内蔵しています。

< 注意事項 >

I/O 拡張シリアルインタフェースの ch.0 を使用時は、I²C インタフェースの ch.0 は、I/O ポートを共有しているため使用できません。

■ I/O 拡張シリアルインタフェースのブロックダイアグラム

図 18.1-1 I/O 拡張シリアルインタフェースブロックダイアグラム



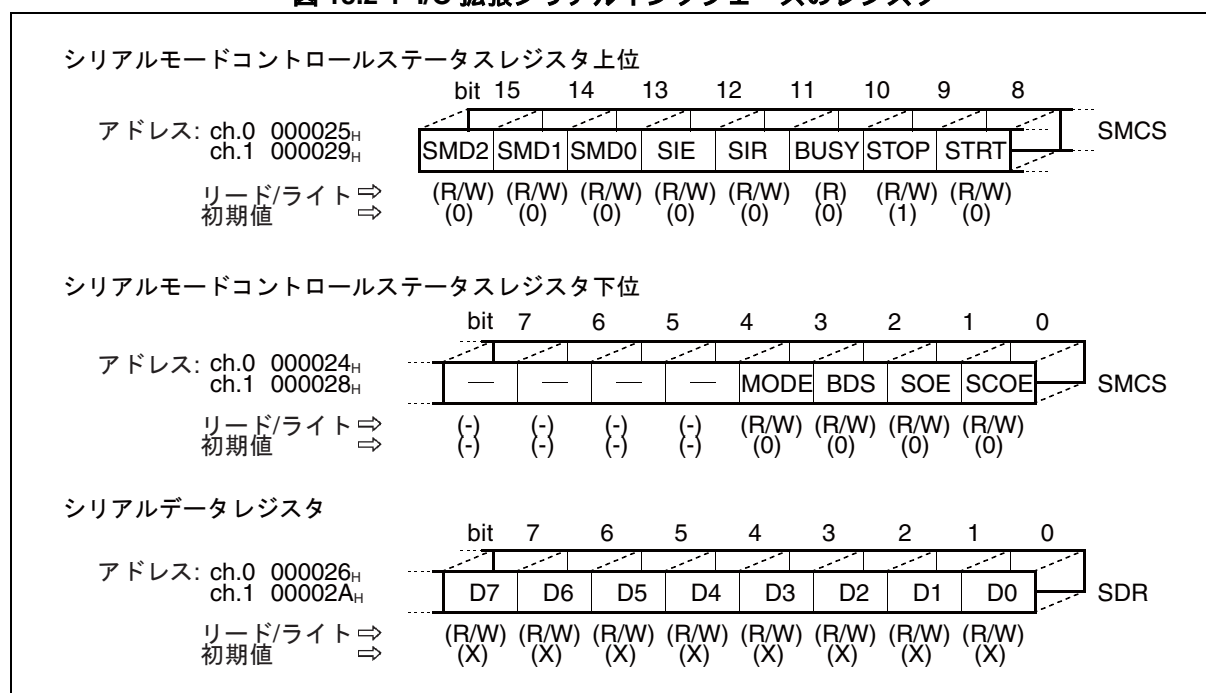
18.2 I/O 拡張シリアルインタフェースのレジスタ

I/O 拡張シリアルインタフェースには、次の 3 種類のレジスタがあります。

- シリアルモードコントロールステータスレジスタ上位
- シリアルモードコントロールステータスレジスタ下位
- シリアルデータレジスタ

■ I/O 拡張シリアルインタフェースのレジスタ

図 18.2-1 I/O 拡張シリアルインタフェースのレジスタ

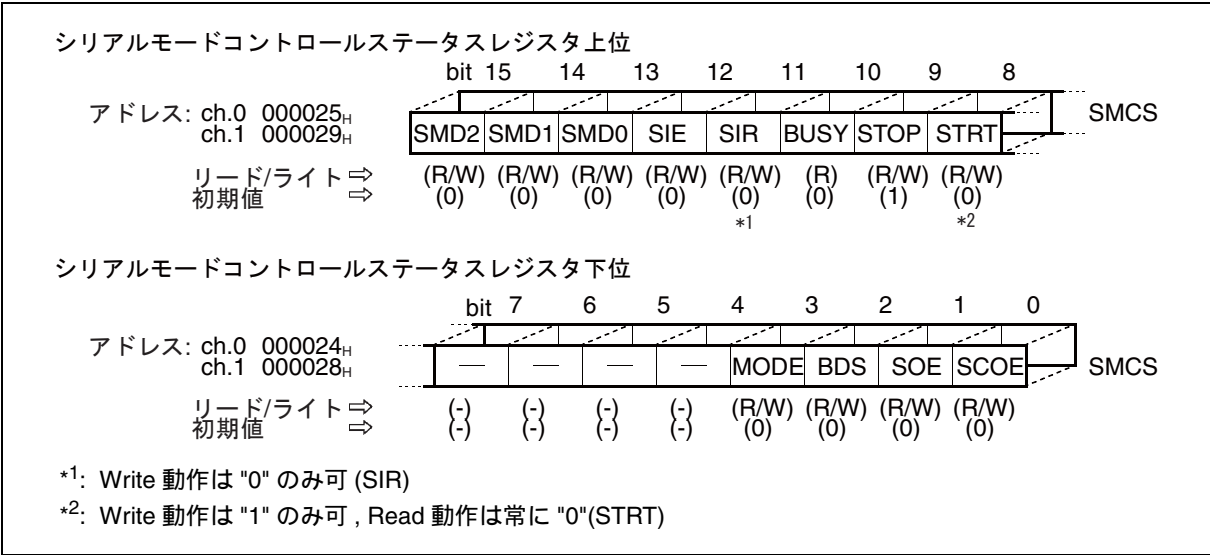


18.2.1 シリアルモードコントロールステータスレジスタ (SMCS)

シリアルモードコントロールステータスレジスタ (SMCS) 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。

■ シリアルモードコントロールステータスレジスタ (SMCS)

図 18.2-2 シリアルモードコントロールステータスレジスタ (SMCS)



[bit15 ~ bit13] SMD2, SMD1, SMD0 (Serial Shift Clock Mode)

SMD2, SMD1, SMD0 ビットは、シリアルシフトクロックモードを表 18.2-1 に示すように選択します。

リセットにより "000" に初期化されます。本ビットの転送中の書換えは禁止します。

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, 111 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、SCK0, SCK1 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可能です。

表 18.2-1 SMD2 ~ SMD0 (シリアルシフトクロックモード選択ビット) の機能

SMD2	SMD1	SMD0	シリアルシフトクロックモード
000 _B ~ 100 _B			内部シフトクロックモード (通信プリスケアラ)
1	0	1	外部シフトクロックモード
1	1	0	予約
1	1	1	予約

[bit12] SIE (Serial I/O Interrupt Enable)

SIE ビットは , シリアル I/O の割込み要求を表 18.2-2 に示すように制御します。

本ビットは , リード / ライトが可能です。

表 18.2-2 SIE (シリアル I/O 割込み要求制御ビット) の機能

SIE	機能
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

[bit11] SIR (Serial i/o Interrupt Request)

SIR 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 へ割込み要求を発生します。クリア条件は MODE ビット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" 書込みによりクリアされます。MODE ビットが "1" のとき , SDR レジスタの読出または書込み動作によりクリアされます。

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" 書込み動作によりクリアされます。

本ビットに "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" がリードされます。

[bit10] BUSY

BUSY ビットは , シリアル転送が実行中か否かを示します。本ビットは , 読出しのみ可能です。

表 18.2-3 BUSY ビットの機能

BUSY	機能
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

[bit9] STOP

STOP 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。

本ビットを "1" にすると STOP=1 による停止状態となります。

本ビットは , リード / ライトが可能です。

表 18.2-4 STOP ビットの機能

STOP	機能
0	通常動作
1	STOP=1 による転送停止 [初期値]

[bit8] STRT

STRT ビットは、シリアル転送を起動します。停止状態で "1" を書き込むことによって転送を開始します。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、"0" のライト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

MODE 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します。

本ビットは、リード / ライトが可能です。

拡張インテリジェント I/O サービスを起動する際には "1" にしておいてください。

表 18.2-5 MODE (起動条件選択ビット) の機能

MODE	機能
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リード / ライトにより起動します。

< 注意事項 >

MODE=1 で使用する場合でも STRT=1 にする必要があります。すなわち、STRT=1 の状態でデータレジスタにデータを書き込むことにより出力されます。

[bit2] BDS (Bit Direction Select)

シリアルデータの入出力時に、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、表 18.2-6 に示すように選択します。

本ビットは、リード / ライト可能です。

表 18.2-6 BDS (Bit Direction Select) ビットの機能

BDS	機能
0	LSB ファースト [初期値]
1	MSB ファースト

< 注意事項 >

転送方向の選択は SDR レジスタ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)

SOE ビットは、シリアル I/O 用出力外部端子 (SOT0, SOT1) の出力を表 18.2-7 に示すように制御します。

本ビットは、リード / ライトが可能です。

表 18.2-7 SOE (Serial Out Enable) ビットの機能

SOE	機能
0	汎用ポート端子 [初期値]
1	シリアルデータ出力

[bit0] SCOE (SCLK Output Enable)

シフトクロック用入出力外部端子 (SCK0, SCK1) の出力を表 18.2-8 で示すように制御します。

外部シフトクロックモードで命令毎に転送を行うときは "0" に設定してください。
本ビットは、リード / ライトが可能です。

表 18.2-8 SCOE (SCLK Output Enable) ビットの機能

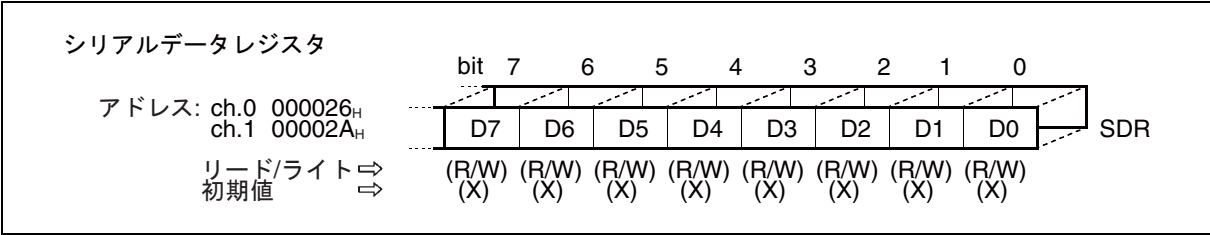
SCOE	機能
0	汎用ポート端子 [初期値]
1	シフトクロック出力端子

18.2.2 シリアルシフトデータレジスタ (SDR)

SDR レジスタは、シリアル I/O の転送データを保持するレジスタです。
転送中の SDR レジスタへの書き込みおよび読出しは禁止です。

■ シリアルシフトデータレジスタ (SDR)

図 18.2-3 シリアルシフトデータレジスタ (SDR)



18.3 I/O 拡張シリアルインタフェースの動作

I/O 拡張シリアルインタフェースは, SMCS レジスタ, SDR レジスタにより構成され, 8bit のシリアルデータの入出力に使用します。

■ I/O 拡張シリアルインタフェースの動作

シリアルデータの入出力は, 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ビット直列にシリアル出力端子 (SOT0, SOT1 端子) に出力され, 立上りに同期してシリアル入力端子 (SIN0, SIN1 端子) からビット直列に SDR 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 レジスタの BDS ビットにより指定することができます。

転送が終了すると, SMCS レジスタの 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, それぞれ次のことを行います。

- 停止状態からの復帰の時は STOP ビットに "0", STRT ビットに "1" を書き込みます。(STOP と STRT は同時設定が可能です。)
- SDR レジスタ R/W 待機状態からの復帰時はデータレジスタを読出したりは書込みします。

18.3.1 シフトクロック

シフトクロックは内部シフトクロックモード，外部シフトクロックモードの 2 種類があり，SMCS レジスタ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。停止状態の確認は，BUSY ビットを読み出すことで可能です。

■ 内部シフトクロックモード

通信プリスケアラの出力により動作し，同期タイミング出力としてデューティ比 50% のシフトクロックが SCK 端子より出力可能です。データが 1 クロックごとに 1bit 転送されます。
転送の速度は，次の式で表せます。

$$\text{転送速度 (s)} = \frac{A}{\text{内部クロックのマシンサイクル (Hz)}}$$

A は SMCS レジスタの SMD ビットで示す分周比で $2^1, 2^2, 2^4, 2^5, 2^6$ です。

表 18.3-1 SMD0 ~ SMD2(シリアルシフトクロックモード選択ビット)の設定値とシフトクロック設定例

SMD2	SMD1	SMD0	$\phi \div \text{div}=4\text{MHz}$	$\phi \div \text{div}=2\text{MHz}$	$\phi \div \text{div}=1\text{MHz}$	算出式
0	0	0	2MHz	1MHz	500kHz	$(\phi \div \text{div}) / 2^1$
0	0	1	1MHz	500kHz	250kHz	$(\phi \div \text{div}) / 2^2$
0	1	0	250kHz	125kHz	62.5kHz	$(\phi \div \text{div}) / 2^4$
0	1	1	125kHz	62.5kHz	31.25kHz	$(\phi \div \text{div}) / 2^5$
1	0	0	62.5kHz	31.25kHz	15.625kHz	$(\phi \div \text{div}) / 2^6$

div は，通信プリスケアラの設定値です。詳細説明は「第 16 章 通信プリスケアラレジスタ」を参照してください。

■ 外部シフトクロックモード

外部シフトクロックモードでは、SCK0, SCK1 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 1bit 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、2MHz まで可能です。

命令ごとに転送することもでき、次のような設定をすることによって実現します。

- 1) 外部シフトクロックモードを選択し、SMCS レジスタの SCOE ビットを "0" に設定します。
- 2) SCK0, SCK1 端子を共有しているポートの方向レジスタに "1" を書き込み、ポートを出力モードに設定します。

以上の設定をした後、ポートのデータレジスタ(PDR)に "1", "0" を書き込むと、SCK0, SCK1 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。シフトクロックの開始は "H" からスタートさせてください。

< 注意事項 >

シリアル I/O 動作中の SMCS レジスタ、SDR レジスタへの書き込みは禁止です。

18.3.2 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、次の 4 種類の状態があります。

- STOP 状態
 - 停止状態
 - SDR レジスタの R/W 待機状態
 - 転送状態
-

■ STOP 状態

RESET 時または SMCS の STOP ビットに "1" を書き込んだときの状態で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 ビットより優先ビットですので 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 レジスタが BUSY=0, SIR=1 となり、カウンタが初期化され停止状態に入ります。停止状態からの復帰は STRT=1 にすると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 レジスタの 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 SDR 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ならば本ブロックより割込み信号を出します。

R/W 待機状態からの復帰は SDR 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 でシリアル転送を行っている状態です。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 18.3-1 に、I/O 拡張シリアルインタフェース動作遷移図を、図 18.3-2 にシリアルデータレジスタへのリード、ライト概念図を示します。

図 18.3-1 I/O 拡張シリアルインタフェース動作遷移図

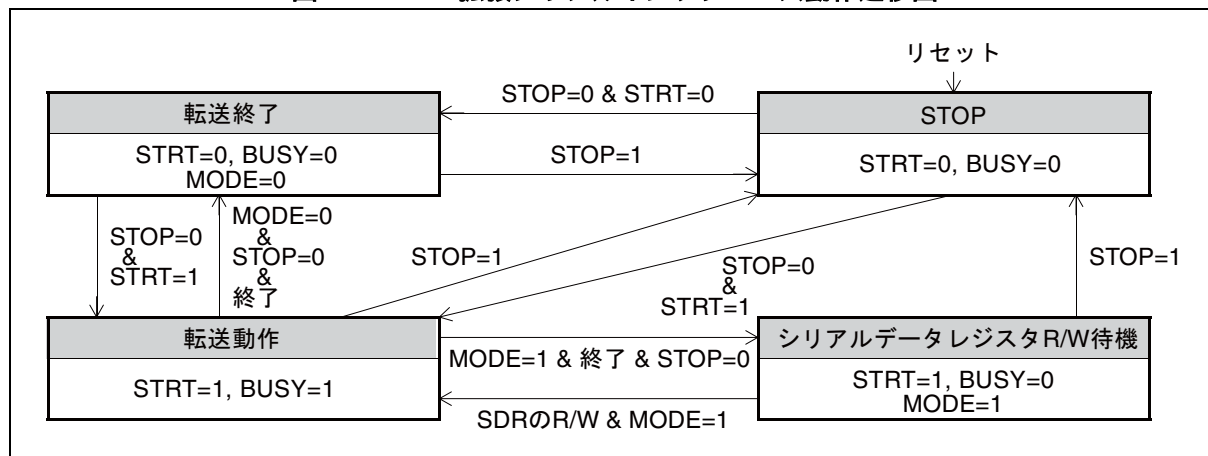


図 18.3-2 シリアルデータレジスタへのリード、ライト概念図

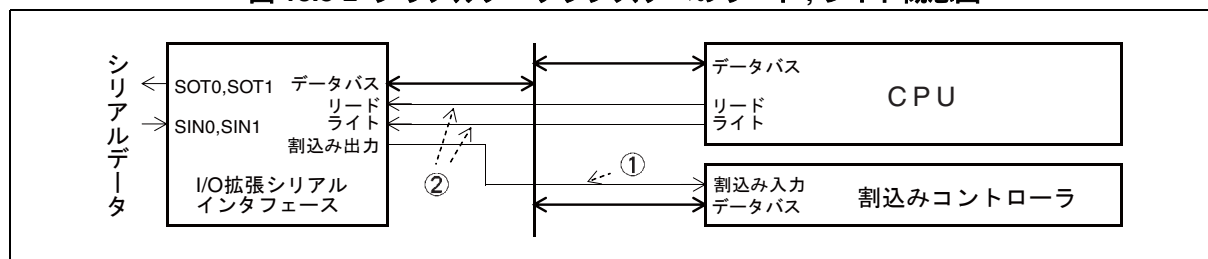


図 18.3-2 の図中における、①と②について説明します。

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 R/W 待機状態に入ります。SIE ビットが "1" なら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 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。

SDR 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を開始します。

18.3.3 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

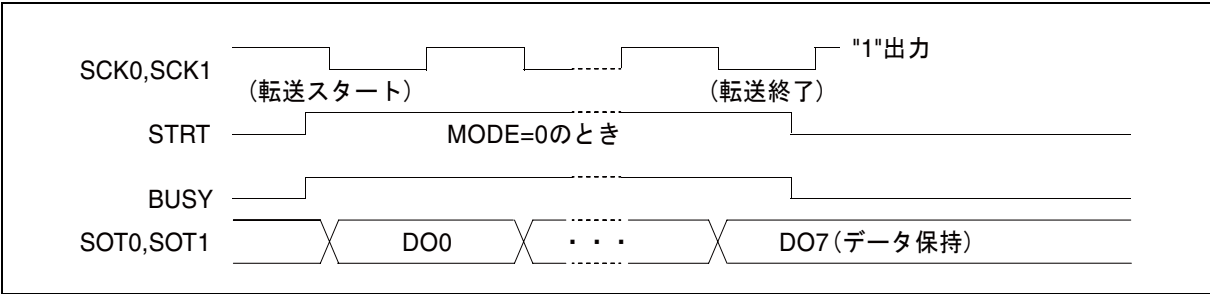
スタート:SMCS レジスタの STOP ビットを "0", STRT ビットを "1" に設定します。
ストップ: 転送終了によって停止する時と STOP=1 によって停止する時があります。
STOP=1 によって停止 MODE ビットにかかわらず SIR=0 のまま停止します。
転送終了で停止 MODE ビットにかかわらず SIR=1 となり停止します。

■ 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

BUSY ビットは MODE ビットにかかわらずシリアル転送状態のとき "1" となり、停止状態またはR/W待機状態のとき "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。

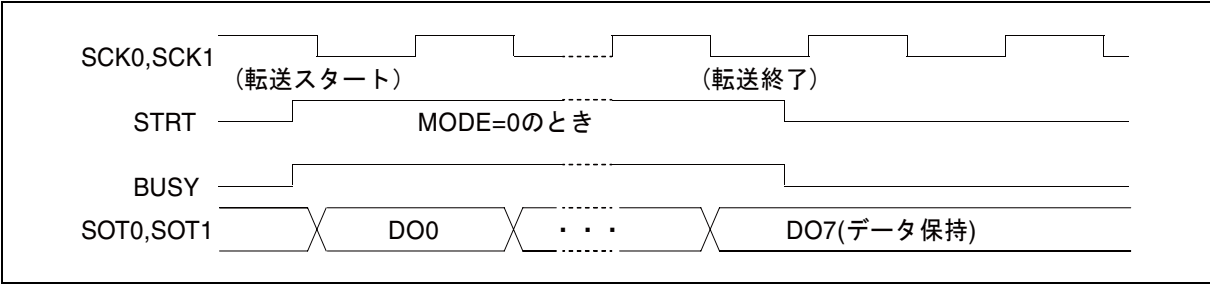
- 内部シフトクロックモード (LSB ファースト)

図 18.3-3 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)



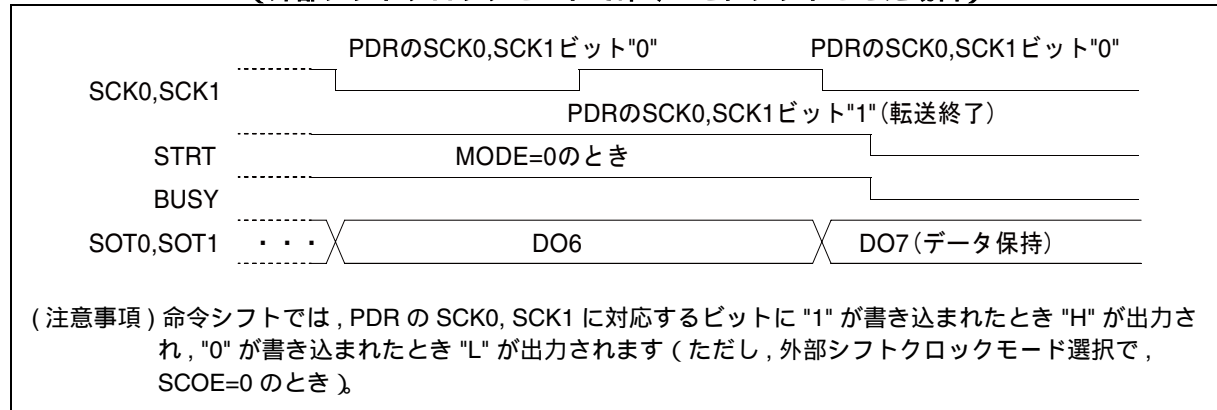
- 外部シフトクロックモード (LSB ファースト)

図 18.3-4 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)



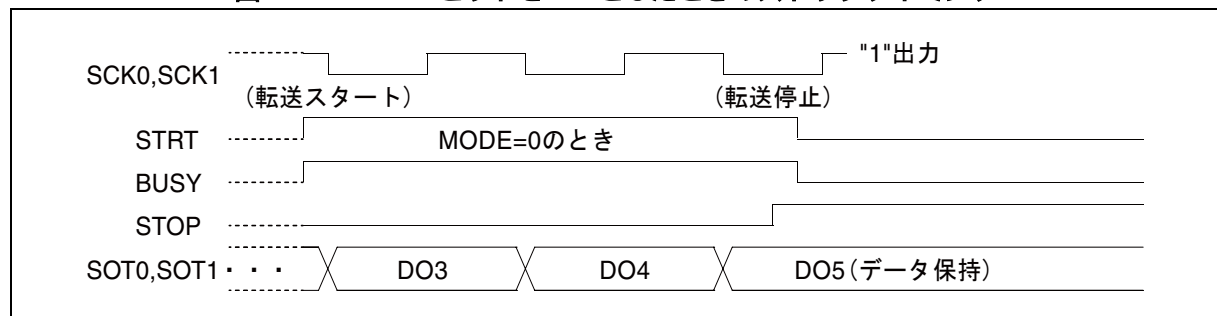
- 外部シフトクロックモードで命令シフトを行ったとき (LSB ファースト)

図 18.3-5 シフト動作のスタート/ストップタイミング
(外部シフトクロックモードで命令ごとにシフトさせた場合)



- STOP=1 による停止 (LSB ファースト, 内部クロック時)

図 18.3-6 STOP ビットを "1" としたときのストップタイミング

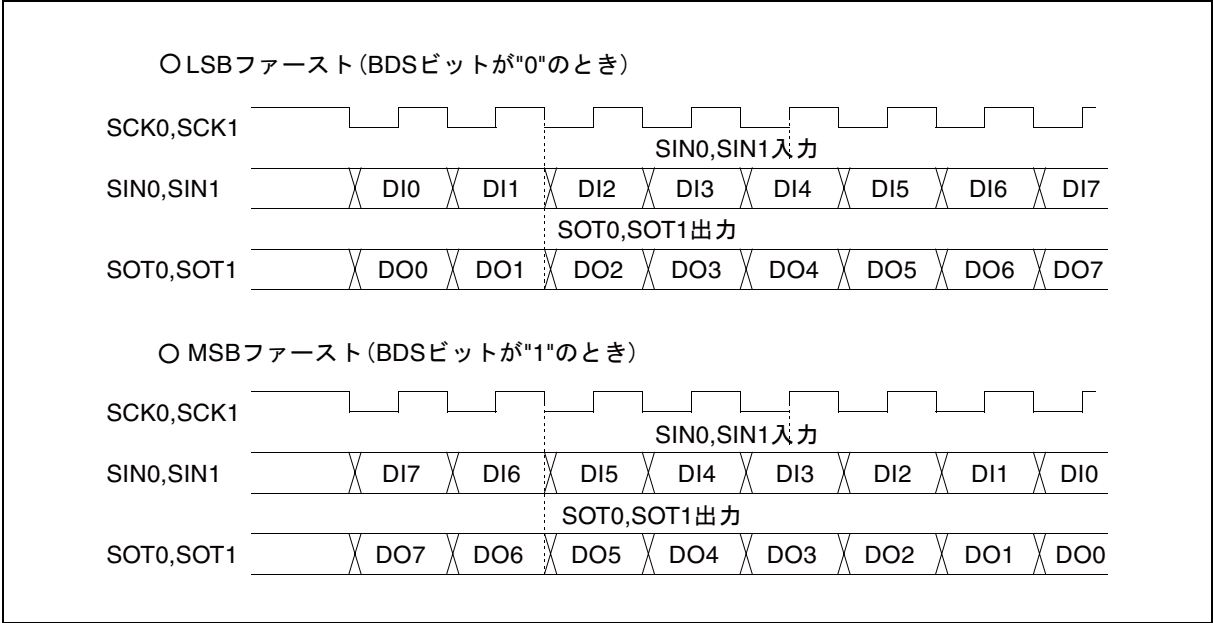


< 注意事項 >

DO7 ~ DO0 は出力データを示します。

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 (SOT0, SOT1) からのデータが出力され，立上りでシリアル入力端子 (SIN0, SIN1) のデータが入力されます。

図 18.3-7 入出力のシフトタイミング

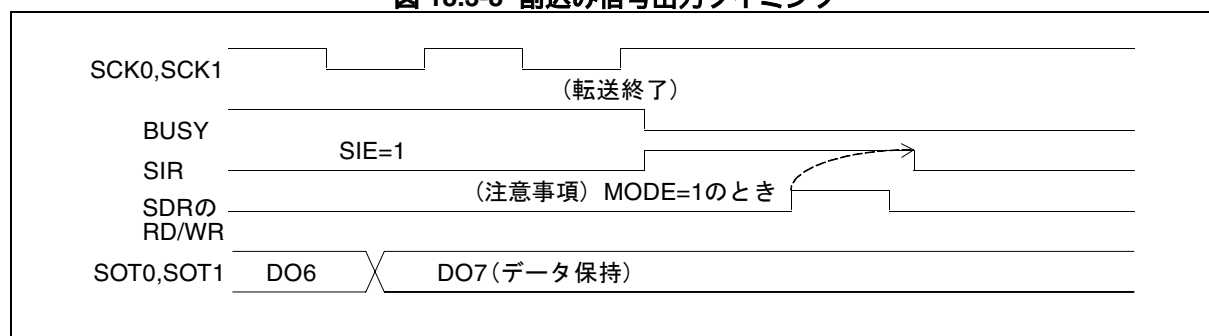


18.3.4 I/O 拡張シリアルインタフェースの割込み機能

I/O 拡張シリアルインタフェースは、CPU に対し割込み要求を発生することができます。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 レジスタの SIE ビットが "1" のとき、CPU へ割込み要求を出力します。

■ I/O 拡張シリアルインタフェースの割込み機能

図 18.3-8 割込み信号出力タイミング



第19章

I²C インタフェース

I²C インタフェースの機能と概要について説明します。

19.1 I²C インタフェースの概要

19.2 I²C インタフェースのブロックダイアグラムと構成図

19.3 I²C インタフェースのレジスタ

19.4 I²C インタフェースの動作

19.1 I²C インタフェースの概要

I²C インタフェースは、Inter IC BUS をサポートするシリアル I/O ポートで、I²C バス上のマスター / スレーブデバイスとして動作し、以下の特長があります。

■ I²C インタフェースの特長

I²C インタフェースには次の特長があります。

- マスター / スレーブ送受信
- 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返し発生および検出機能
- バスエラー検出機能

MB90550A/B シリーズは I²C インタフェースを 2 回路 3ch で構成しています。

< 注意事項 >

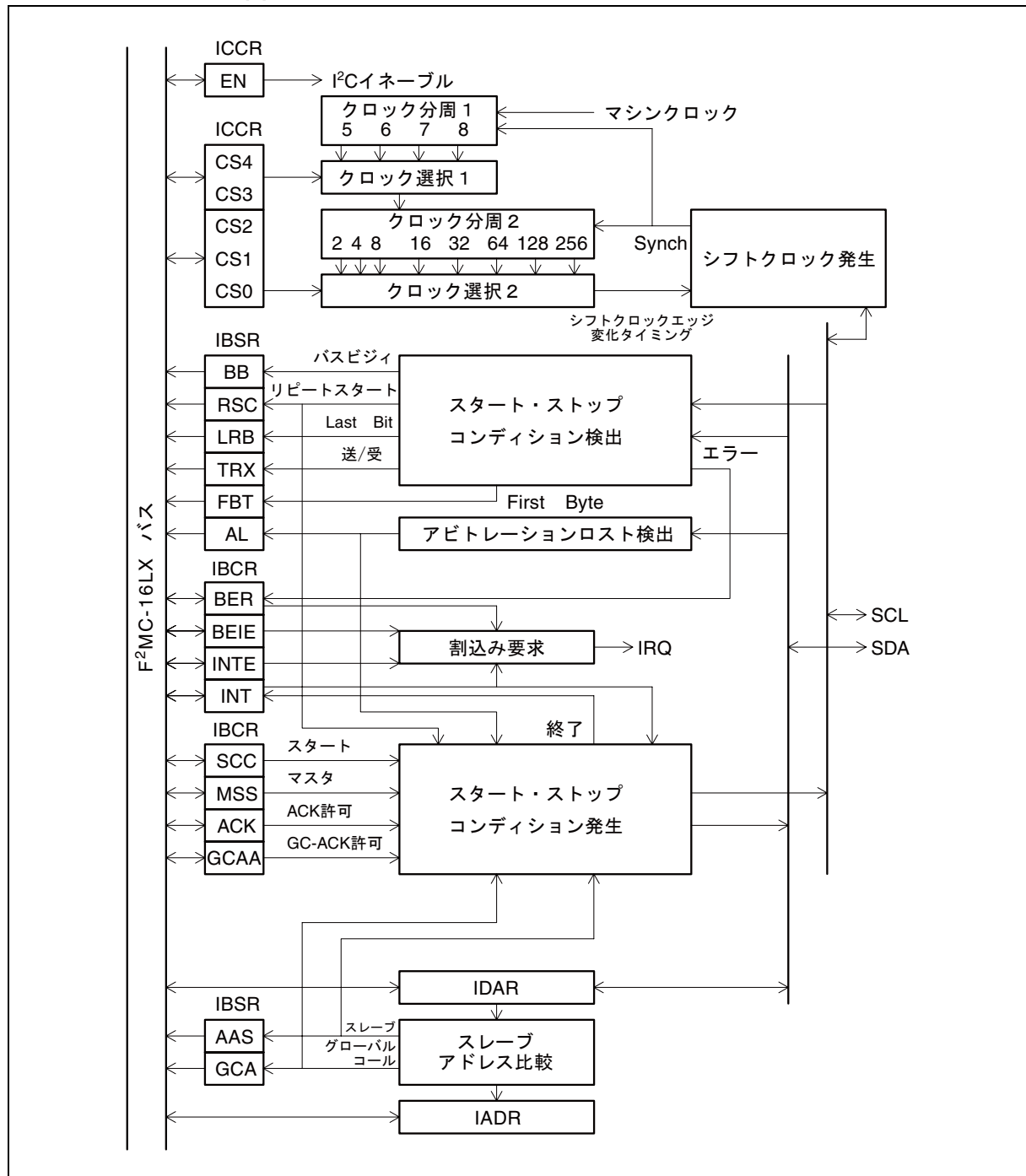
I²C インタフェースの ch.0 を使用時は、I/O 拡張シリアルインタフェースの ch.0 は、I/O ポートを共有しているため使用できません。

19.2 I²C インタフェースのブロックダイアグラムと構成図

図 19.2-1 に I²C インタフェースのブロックダイアグラムを、図 19.2-2 に I²C インタフェースの構成図を示します。

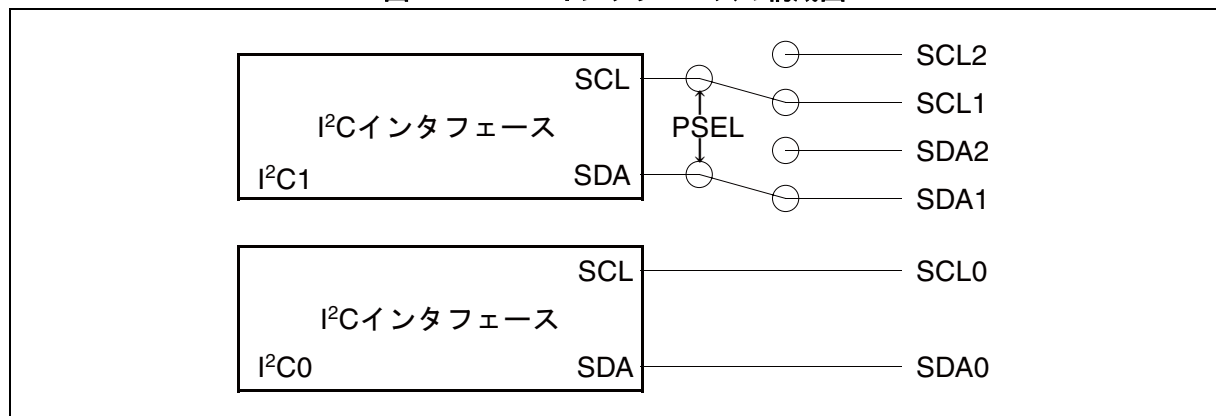
■ I²C インタフェースのブロックダイアグラム

図 19.2-1 I²C インタフェースのブロックダイアグラム



■ I²C インタフェースの構成図

図 19.2-2 I²C インタフェースの構成図



19.3 I²C インタフェースのレジスタ

I²C インタフェースのレジスタには、次の 6 種類があります。

- バスステータスレジスタ
- バスコントロールレジスタ
- クロックコントロールレジスタ
- アドレスレジスタ
- データレジスタ
- ポート選択レジスタ

■ I²C インタフェースのレジスタ

図 19.3-1 I²C インタフェースのレジスタ

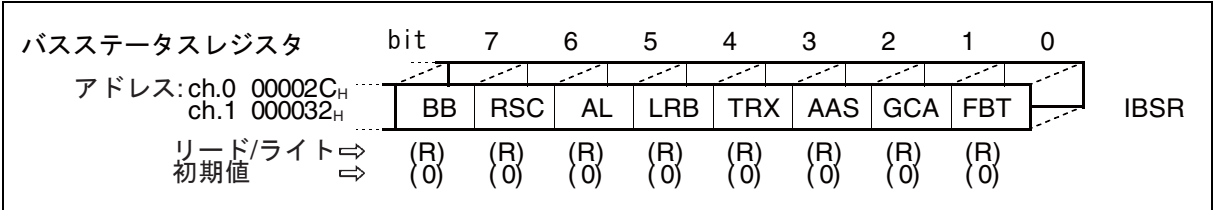
バスステータスレジスタ		bit	7	6	5	4	3	2	1	0	
アドレス: ch.0	00002C _H										IBSR
ch.1	000032 _H		BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	
リード/ライト ⇒			(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値 ⇒			(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
バスコントロールレジスタ		bit	15	14	13	12	11	10	9	8	
アドレス: ch.0	00002D _H										IBCR
ch.1	000033 _H		BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	
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		bit	7	6	5	4	3	2	1	0	
アドレス: ch.0	00002E _H										ICCR
ch.1	000034 _H		—	—	EN	CS4	CS3	CS2	CS1	CS0	
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		bit	15	14	13	12	11	10	9	8	
アドレス: ch.0	00002F _H										IADR
ch.1	000035 _H		—	A6	A5	A4	A3	A2	A1	A0	
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		bit	7	6	5	4	3	2	1	0	
アドレス: ch.0	000030 _H										IDAR
ch.1	000036 _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)	
ポート選択レジスタ		bit	15	14	13	12	11	10	9	8	
アドレス: ch.1	000037 _H		—	—	—	—	—	—	—	PSEL	ISEL
リード/ライト ⇒			(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
初期値 ⇒			(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	

19.3.1 バスステータスレジスタ (IBSR)

バスステータスレジスタ (IBSR) では、I²C インタフェースの各機能の状態を示します。

■ バスステータスレジスタ (IBSR)

図 19.3-2 バスステータスレジスタ (IBSR)



[bit7] BB (Bus Busy)
BB は、I²C バスの状態を示すビットです。

表 19.3-1 BB (Bus Busy) ビットの機能

BB	I ² C バスの状態
0	ストップコンディションを検出した。[初期値]
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)
RSC は、繰返しスタートコンディションを検出するビットです。INT ビットへの "0" 書込み、スレープ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でクリアされます。

表 19.3-2 RSC (Repeated Start Condition) ビットの機能

RSC	スタートコンディション検出の状態
0	繰返しスタートコンディションは検出されていない。[初期値]
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

[bit5] AL (Arbitration Lost)

AL は、アービトレーションロストを検出するビットです。

INT ビットへの "0" 書込みでクリアされます。

表 19.3-3 AL (Arbitration Lost) ビットの機能

AL	アービトレーションロスト検出の状態
0	アービトレーションロスト検出されていない。[初期値]
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であるときに MSS ビットに "1" 書込みを行った場合

[bit4] LRB (Last Received Bit)

LRB は、アクノリッジ格納ビットで、受信側からのアクノリッジを格納します。

表 19.3-4 LRB (Last Received Bit) ビットの機能

LRB	アクノリッジの受信状態
0	受信を確認した。
1	受信が確認されない。

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でクリアされます。

[bit3] TRX (Transfer/Receive)

TRX は、データ転送の送受信を示すビットです。

表 19.3-5 TRX (Transfer/Receive) ビットの機能

TRX	データ転送の送受信状態
0	受信状態 [初期値]
1	送信状態

[bit2] AAS (Addressed As Slave)

AAS は、アドレッシング検出ビットです。

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でクリアされます。

表 19.3-6 AAS (Addressed As Slave) ビットの機能

AAS	機能
0	スレーブ時に、アドレッシングされていない。[初期値]
1	スレーブ時に、アドレッシングされた。

[bit1] GCA (General Call Address)

GCA は、ゼネラルコールアドレス (00_H) 検出ビットです。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でクリアされます。

表 19.3-7 GCA (General Call Address) ビットの機能

GCA	機能
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。[初期値]
1	スレーブ時に、ゼネラルコールアドレスを受信した。

[bit0] FBT (First Byte Transfer)

FBT は、第 1 バイト検出ビットです。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" 書込みまたはスレーブ時にアドレスされなかった場
合クリアされます。

表 19.3-8 FBT (First Byte Transfer) ビットの機能

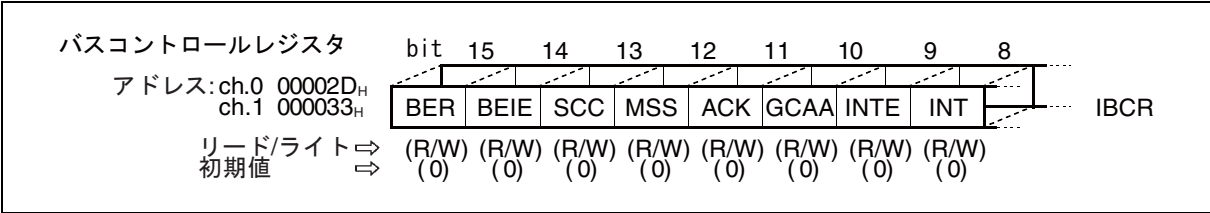
FBT	機能
0	受信データが第 1 バイト以外である [初期値]
1	受信データが第 1 バイト (アドレスデータ) である

19.3.2 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) では、割込みや、I²C インタフェースの各機能を制御します。

■ バスコントロールレジスタ (IBCR)

図 19.3-3 IBCR (バスコントロールレジスタ)



[bit15] BER (Bus ERror)

BER は、バスエラー割込み要求フラグビットです。このビットがセットされた場合、CCR レジスタの EN ビットはクリアされ、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。

表 19.3-9 BER (Bus ERror) ビットの機能

BER		機能
書込み時	0	バスエラー割込み要求フラグをクリアする。[初期値]
	1	関係なし
読出し時	0	バスエラーは検出されていない。[初期値]
	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。

[bit14] BEIE (Bus Error Interrupt Enable)

BEIE は、バスエラー割込み許可ビットです。このビットが "1" のとき、BER ビットが "1" なら割込みを発生します。

表 19.3-10 BEIE (Bus Error Interrupt Enable) ビットの機能

BEIE	機能
0	バスエラー割込み禁止 [初期値]
1	バスエラー割込み許可

[bit13] SCC (Start Condition Continue)

SCC ビットは、スタートコンディション発生ビットです。このビットの読出し値は常に "0" です。

表 19.3-11 SCC (Start Condition Continue) ビットの書込み時の機能

SCC	機能
0	関係なし [初期値]
1	マスタ転送時に再びスタートコンディションを発生し、アドレスデータ転送を開始します。

[bit12] MSS (Master Slave Select)

MSS ビット、マスタ / スレーブの選択ビットです。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クリアされ、スレーブモードになります。

表 19.3-12 MSS (Master Slave Select) の機能

MSS	機能
0	ストップコンディションを発生し転送終了後スレーブモードとなります。 [初期値]
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アドレスデータ転送を開始します。

[bit11] ACK (ACKnowledge)

データを受信した場合のアクノリッジ発生許可ビットです。このビットは、スレーブ時のアドレスデータ受信時には無効となります。

表 19.3-13 ACK (ACKnowledge) の機能

ACK	機能
0	アクノリッジ発生しない [初期値]
1	アクノリッジ発生する

[bit10] GCAA (General Call Address Acknowledge)

ゼネラルコールアドレスを受信した場合のアクノリッジ発生許可ビットです。

表 19.3-14 GCAA (General Call Address Acknowledge) の機能

GCAA	機能
0	アクノリッジ発生しない。 [初期値]
1	アクノリッジ発生する。

[bit9] INTE (INTerrupt Enable)

INTE は、割込み許可ビットです。このビットが "1" のとき、INT ビットが "1" なら割込みを発生します。

表 19.3-15 INTE (INTerrupt Enable) の機能

INTE	機能
0	割込み禁止 [初期値]
1	割込み許可

[bit8] INT (INTerrupt)

INT は、転送終了割込み要求フラグビットです。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" を書き込みによりクリアされ、SCL ラインを解放し、次バイトの転送を行います。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り "0" にリセットされます。

表 19.3-16 INT (INTerrupt) の機能

INT		機能
書き込み時	0	転送終了割込み要求フラグをクリアする。[初期値]
	1	関係なし
読出し時	0	転送が終了していない。[初期値]
	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。

■ SCC, MSS, INT ビットの競合についての注意

SCC, MSS, INT ビットの同時書き込みにより、次バイト転送、スタートコンディション発生、ストップコンディション発生の競合が起こります。このときの優先度は次のようになります。

1) 次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" 書き込みが優先され、ストップコンディションが発生されます。

2) 次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" 書き込みが優先され、スタートコンディションが発生されます。

3) スタートコンディション発生とストップコンディション発生

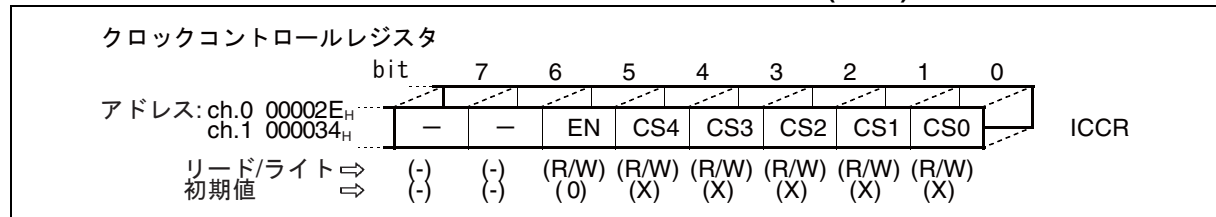
SCC ビットに "1", MSS ビットに "0" の同時書き込みは禁止します。

19.3.3 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) では、I²C インタフェース動作の制御や、シリアルクロックの周波数を設定します。

■ クロックコントロールレジスタ (ICCR)

図 19.3-4 クロックコントロールレジスタ (ICCR)



[bit5] EN (ENable)

EN は、I²C インタフェース動作許可ビットです。このビットが "0" のとき、BSR レジスタ、BCR レジスタ (BER, BEIE ビットを除く) の各ビットはクリアされます。BER ビットがセットされた場合、このビットはクリアされます。

表 19.3-17 EN (Enable) ビットの機能

EN	動作状態
0	動作禁止 [初期値]
1	動作許可

[bit4 ~ bit0] CS4 ~ CS0 (Clock Period Select4 ~ 0)

シリアルクロックの周波数を設定するビットです。

シフトクロックの周波数 f_{sck} は次式のように設定されます。

$$f_{sck} = \frac{\phi}{m \times n + 4}$$

ϕ : マシンクロック

表 19.3-18 シリアルクロック周波数設定 (CS4, CS3)

m	CS4	CS3
5	0	0
6	0	1
7	1	0
8	1	1

表 19.3-19 シリアルクロック周波数設定 (CS2 ~ CS0)

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

例えば、 $\phi=16\text{MHz}$ のとき、 $m=5$, $n=32$ を選択した場合、シリアルクロック周波数は、 97.561kHz になります。

< 注意事項 >

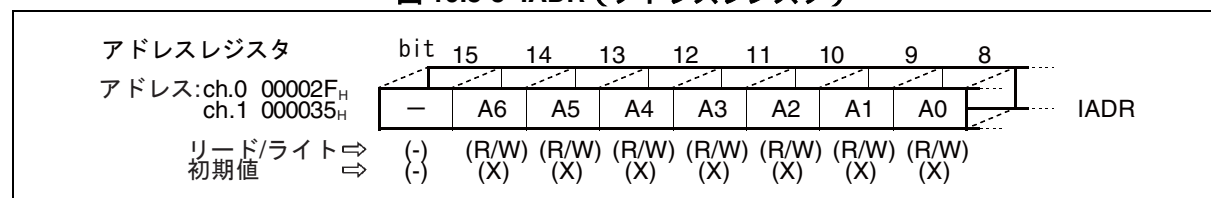
+4 のサイクルは、SCL 端子の出力レベルが変化したことをチェックしているための最小のオーバーヘッドです。SCL 端子の立上りのディレイが大きい場合やスレーブデバイスでクロックを引き延ばしている場合には、この値より大きくなります。

19.3.4 アドレスレジスタ (IADR)

アドレスレジスタ (IADR) では, スレーブアドレスを指定します。

■ アドレスレジスタ (IADR)

図 19.3-5 IADR (アドレスレジスタ)



[bit14 ~ bit8] A6 ~ A0 (スレーブアドレスビット)

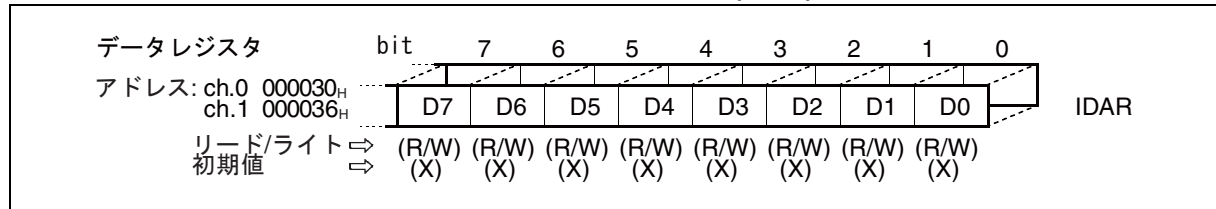
A6 ~ A0 は, スレーブアドレスを指定するレジスタです。スレーブ時, アドレスデータ受信後に DAR レジスタとの比較が行われ, 一致している場合はマスタに対してアクノリッジを送信します。

19.3.5 データレジスタ (IDAR)

データレジスタ (IDAR) では, シリアル転送に使用されるデータの読出しや書込みを行います。

■ データレジスタ (IDAR)

図 19.3-6 データレジスタ (IDAR)



[bit7 ~ bit0] D7 ~ D0 (データビット)

シリアル転送に使用されるデータレジスタであり, MSB から転送されます。データ受信時 (TRX=0) は, 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, 書込みデータは各バイト転送時に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INTビットがセットされている場合のみ有効です。

19.3.6 ポート選択レジスタ (ISEL)

ポート選択レジスタ (ISEL) では , I²C インタフェースの入出力ポートを選択します。

■ ポート選択レジスタ (ISEL)

図 19.3-7 ポート選択レジスタ (ISEL)

ポート選択レジスタ	bit 15	14	13	12	11	10	9	8	
アドレス: ch.1 000037 _H	—	—	—	—	—	—	—	PSEL	ISEL
リード/ライト ⇒ 初期値	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W) (0)	

[bit8] PSEL

PSEL ビットは , I²C インタフェースの入出力ポートを選択するビットです。入出力ポートを切換え (書込み) 時は , ICCR レジスタの EN ビットが "0" であることを確認してから行ってください。

選択されていないポート (P52, P53 または P54, P55) は汎用ポートとして使用できません。本ビットは , I²C インタフェース 1 のみに有効です。I²C インタフェース 0 には関係ありません。

表 19.3-20 PSEL ビットの機能

PSEL	機能
0	SCL1, SDA1 を選択 [初期値]
1	SCL2, SDA2 を選択

19.4 I²C インタフェースの動作

I²C バスは、1 本のシリアルデータライン (SDA) と 1 本のシリアルクロックライン (SCL) の 2 本の双方向バスラインにより通信が行われます。I²C インタフェースはそれに対して 2 本のオープンドレイン入出力端子 (SDA, SCL) を有し、ワイヤード論理を可能にします。

■ 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" を書き込むことで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 とおりがあります。

- 1) 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" 書込み。
- 2) 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" 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" 書込みを行うと、AL ビットが "1" にセットされます。1), 2) 以外での MSS ビット、SCC ビットへの "1" 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" 書込み。

上記以外での、MSS ビットへの "0" 書込みは、無視されます。

■ 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 レジスタの内容を MSB から出力します。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 bit0 (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を IDAR レジスタへ受信します。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、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見なし、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することができます。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。

スレーブ送信時に、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。これにより、マスタはスレーブが SCL ラインを解放したときに、ストップコンディションを発生することができます。

■ 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
- バスアイドル時の I²C バス上の基本規定違反の検出

19.4.1 I²C インタフェースの転送フロー

図 19.4-1 に、マスタからスレーブへの 1 バイト転送のフローを、図 19.4-2 に、スレーブからマスタへの 1 バイト転送のフローを示します。

■ I²C インタフェースの転送フロー

図 19.4-1 マスタからスレーブへの 1 バイト転送のフロー

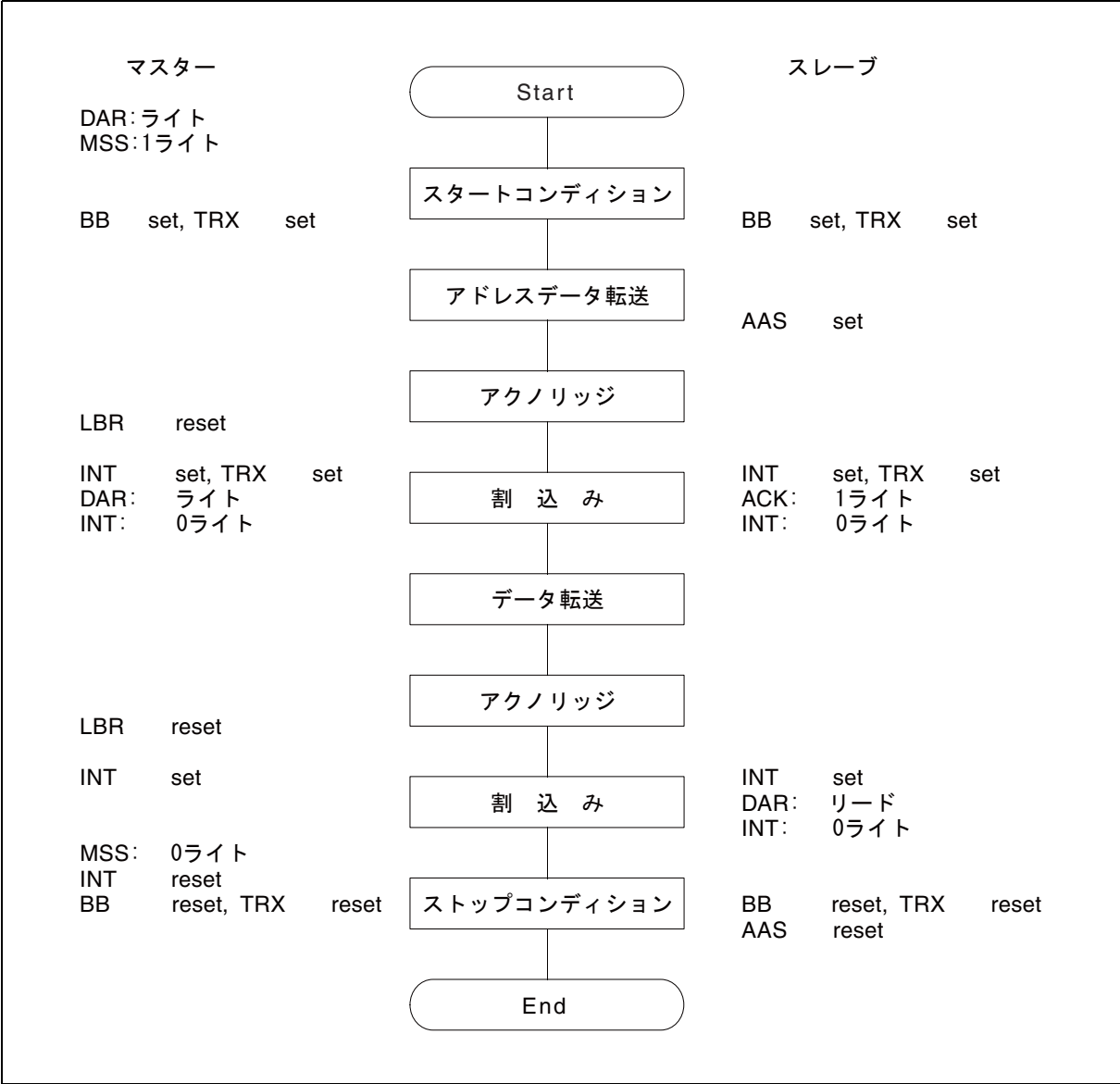
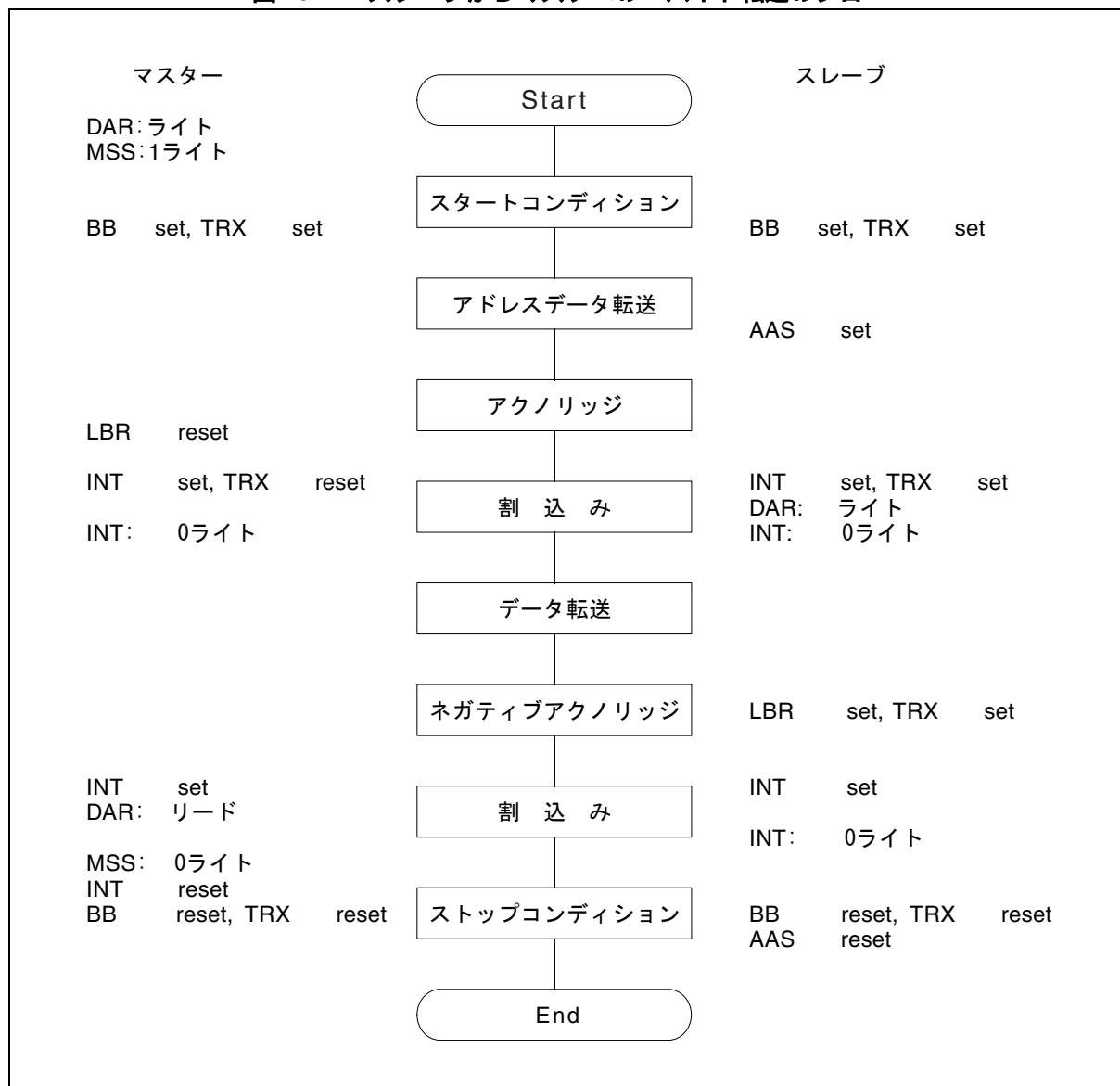


図 19.4-2 スレーブからマスタへの 1 バイト転送のフロー

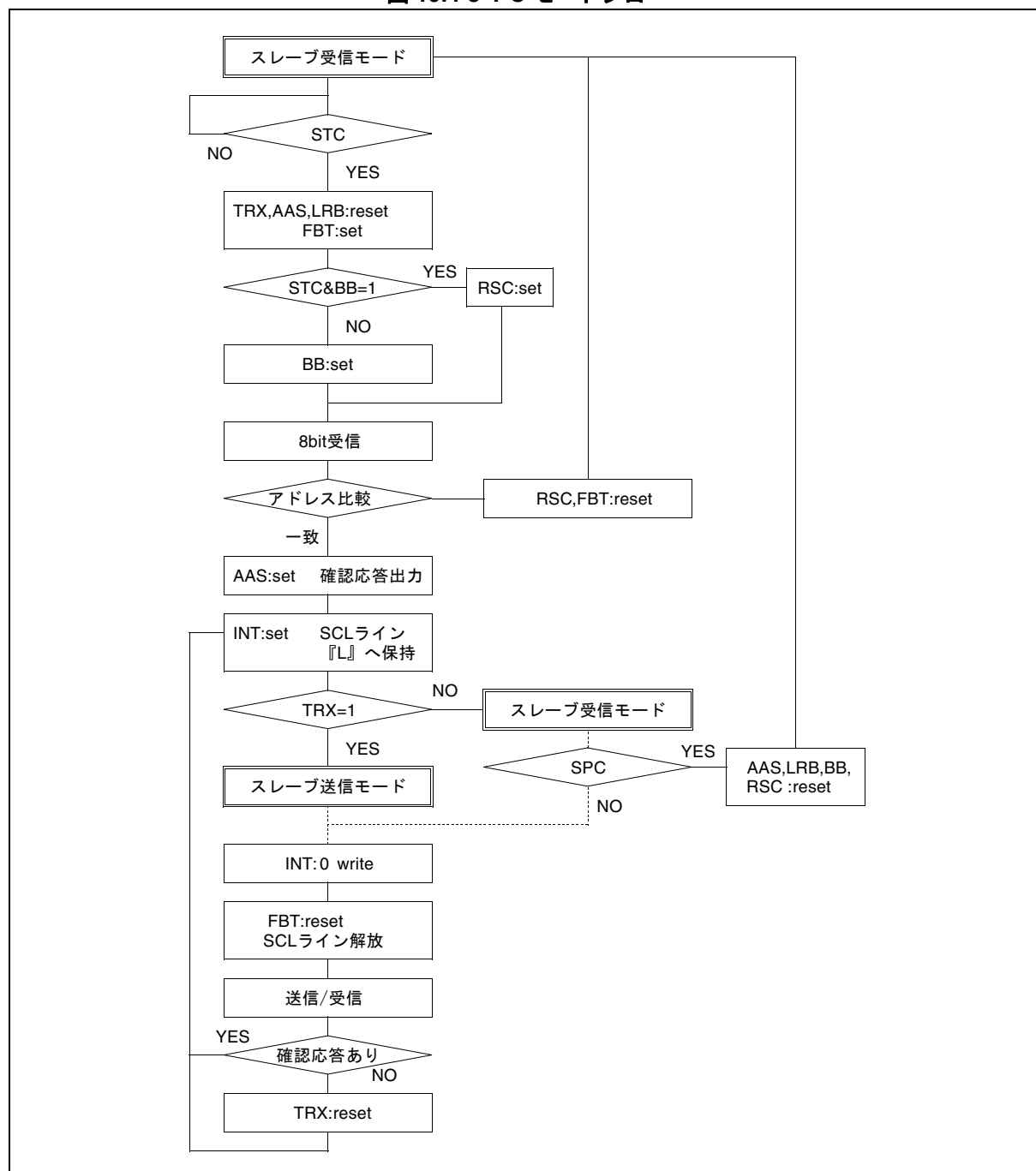


19.4.2 I²C インタフェースのモードフロー

図 19.4-3 に、I²C インタフェースのモードフローを示します。

■ I²C インタフェースのモードフロー

図 19.4-3 I²C モードフロー



第20章

クロックモニタ機能

クロックモニタの機能と動作について説明します。

20.1 クロックモニタ機能の概要

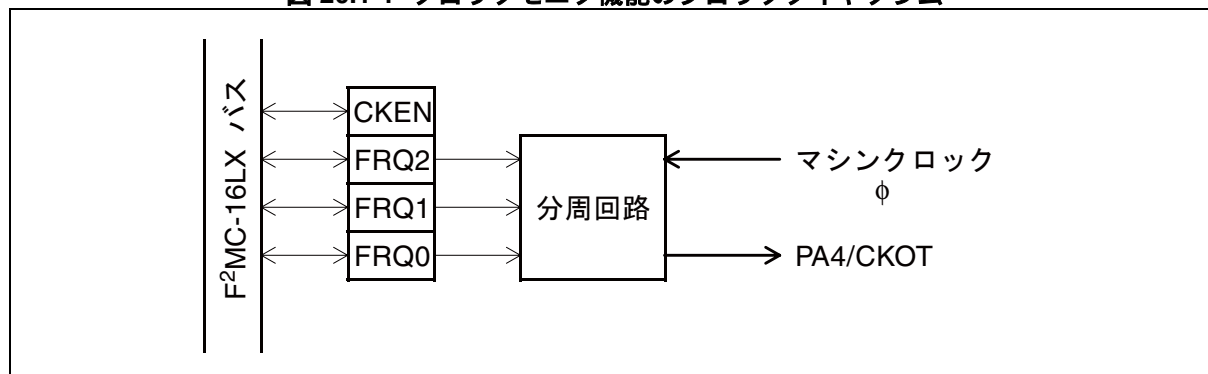
20.2 クロック出力許可レジスタ (CLKR)

20.1 クロックモニタ機能の概要

クロックモニタ機能は、CKOT 端子からマシンのクロックの分周クロック (モニタ用のクロック) を出力します。

■ クロックモニタ機能のブロックダイアグラム

図 20.1-1 クロックモニタ機能のブロックダイアグラム



20.2 クロック出力許可レジスタ (CLKR)

クロック出力許可レジスタ (CLKR) の各ビットを使用して, CKOT 出力許可やクロック出力周波数を選択します。

■ クロック出力許可レジスタ (CLKR)

図 20.2-1 クロック出力許可レジスタ (CLKR)

クロック出力許可レジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:00058 _H	—	—	—	—	CKEN	FRQ2	FRQ1	FRQ0
リード/ライト⇒	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(-)	(-)	(-)	(-)	(0)	(0)	(0)	(0)

[bit3] CKEN

CKEN は, CKOT 出力許可ビットです。

表 20.2-1 CKEN ビットの機能

CKEN	機能
0	通常ポートとなります。
1	CKOT 出力となります。

[bit2 ~ bit0] FRQ2, FRQ1, FRQ0

FRQ2, FRQ1, FRQ0 ビットは, クロック出力周波数を選択するビットです。

表 20.2-2 FRQ2, FRQ1, FRQ0 ビットの機能

FRQ2	FRQ1	FRQ0	出力クロック	$\phi = 16\text{MHz}$	$\phi = 8\text{MHz}$	$\phi = 4\text{MHz}$
0	0	0	$\phi / 2^1$	125	250	500
0	0	1	$\phi / 2^2$	250	500	1
0	1	0	$\phi / 2^3$	500	1	2
0	1	1	$\phi / 2^4$	1	2	4
1	0	0	$\phi / 2^5$	2	4	8
1	0	1	$\phi / 2^6$	4	8	16
1	1	0	$\phi / 2^7$	8	16	32
1	1	1	$\phi / 2^8$	16	32	64

第21章

アドレス一致検出機能

この章は、アドレス一致検出の機能と動作について説明します。

- 21.1 アドレス一致検出機能の概要
- 21.2 アドレス一致検出機能のレジスタ
- 21.3 アドレス一致検出機能の動作
- 21.4 アドレス一致検出機能の使用例
- 21.5 アドレス一致検出機能のプログラム例

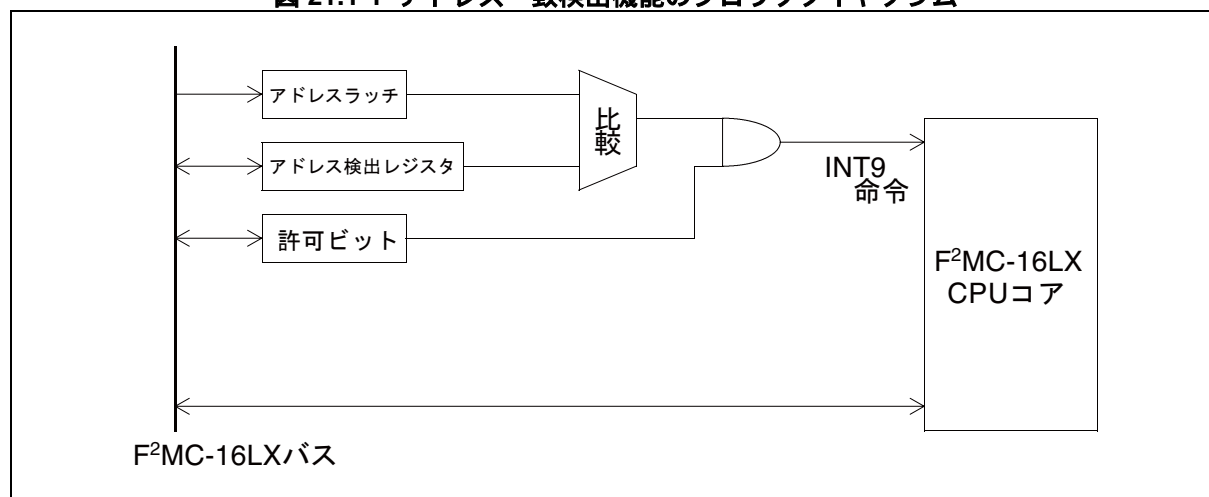
21.1 アドレス一致検出機能の概要

アドレスが、アドレス検出レジスタに設定された値と等しい場合に、CPU に読み込まれる命令コードを強制的に INT9 命令のコード (01_H) に置き換えます。その結果、CPU が設定された命令を実行するときに、INT9 命令を実行します。INT #9 割り込みルーチンで処理を行うことにより、プログラムのパッチ当て機能を実現することができます。

アドレス検出レジスタは 2 本用意されており、各レジスタごとに割り込み許可ビットがあります。アドレス検出レジスタに設定された値とアドレスが一致して、かつ割り込み許可ビットが "1" の場合、CPU に読み込まれる命令コードを強制的に INT9 命令のコードに置き換えます。

■ アドレス一致検出機能のブロックダイアグラム

図 21.1-1 アドレス一致検出機能のブロックダイアグラム



21.2 アドレス一致検出機能のレジスタ

アドレス一致検出機能には、次の 2 種類のレジスタがあります。

- 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コントロール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 (PADR0/PADR1)

プログラムアドレス検出レジスタ (PADR0/PADR1:Program Address Detect Register 0/1) は、各レジスタに書き込まれた値と、アドレスを比較します。一致した場合は、ADCSR の対応する割込み許可ビットが "1" の場合、CPU に対して INT9 命令の発生を要求します。

対応する割込み許可ビットが "0" の場合は、一致しても何も行いません。

図 21.2-1 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			アクセス	初期値
	byte	byte		
PADR0	1FF2H/1FF1H/1FF0H		R/W	不定
PADR1	1FF5H/1FF4H/1FF3H		R/W	不定

表 21.2-1 に、プログラムアドレス検出レジスタ (PADR0/PADR1) と PACSR との対応を示します。

表 21.2-1 PADR0/PADR1 レジスタと PACSR との対応

アドレス検出レジスタ	割込み許可ビット
PADR0	AD0E
PADR1	AD1E

■ プログラムアドレス検出コントロールステータスレジスタ (PACSR)

プログラムアドレス検出コントロールステータスレジスタ (PACSR: Program Address detect Control Register) は、アドレス検出機能の動作を制御します。

図 21.2-2 プログラムアドレス検出コントロールステータスレジスタ (PACSR)

プログラムアドレス検出コントロールステータスレジスタ								
	bit 7	6	5	4	3	2	1	0
アドレス:009E _H	予約	予約	予約	予約	AD1E	予約	AD0E	予約
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)

[bit7 ~ bit4] 予約ビット

bit7 ~ bit4 は、予約ビットです。PACSR を設定するときは、必ず "0" を設定してください。

[bit3] AD1E (Address Detect register 1 Enable)

AD1E は、PADR1 の動作許可ビットです。

このビットが "1" のとき PADR1 レジスタとアドレスの比較を行い、一致した場合に CPU は INT9 命令を発生します。

[bit2] 予約ビット

bit2 は、予約ビットです。PACSR を設定するときは、必ず "0" を設定してください。

[bit1] AD0E (Address Detect register 0 Enable)

AD0E ビットは、PADR0 の動作許可ビットです。

このビットが "1" のとき PADR0 レジスタとアドレスの比較を行い、一致した場合に CPU は INT9 命令を発生します。

[bit0] 予約ビット

bit0 は、予約ビットです。PACSR を設定するときは、必ず "0" を設定してください。

21.3 アドレス一致検出機能の動作

アドレス一致検出機能は、アドレスが、アドレス検出レジスタに設定された値と等しい場合に、CPU に読み込まれる命令コードを強制的に INT9 命令のコード (01_H) に置き換えます。その結果、CPU が設定された命令を実行するときに、INT9 命令を実行します。INT #9 割込みルーチンで処理を行うことにより、プログラムのパッチ当て機能を実現することができます。

■ アドレス一致検出機能の動作

アドレス検出レジスタは2本用意されており、各レジスタごとに割込み許可ビットがあります。アドレス検出レジスタに設定された値とアドレスが一致して、かつ割込み許可ビットが "1" の場合、CPU に読み込まれる命令コードを強制的に INT9 命令のコードに置き換えます。

■ アドレス一致検出機能動作上の注意

アドレス一致検出機能は、内蔵 ROM のアドレスに対してのみ有効です。外部メモリ領域のアドレスを設定しても、INT9 命令は発生いたしません。

アドレス検出レジスタに、命令の 1 バイト目のアドレス以外を設定した場合は、この機能は正しく動作しません。設定したアドレスのデータが "01_H" に替わることになるので、違う命令を実行したり、違うアドレスをアクセスすることになります。

なお、アドレス検出レジスタの変更は割込み許可ビットを "0" にした状態で行ってください。割込み許可ビットが "1" の状態で書き込んだ場合は、書込みの途中で誤ってアドレス検出を行ってしまい、誤動作を引き起こす可能性が有ります。

21.4 アドレス一致検出機能の使用例

図 21.4-1 にアドレス一致検出機能のシステム構成例を，表 21.4-1 に E²PROM メモリマップを示します。

■ アドレス一致検出機能の使用例

図 21.4-1 アドレス一致検出機能のシステム構成例

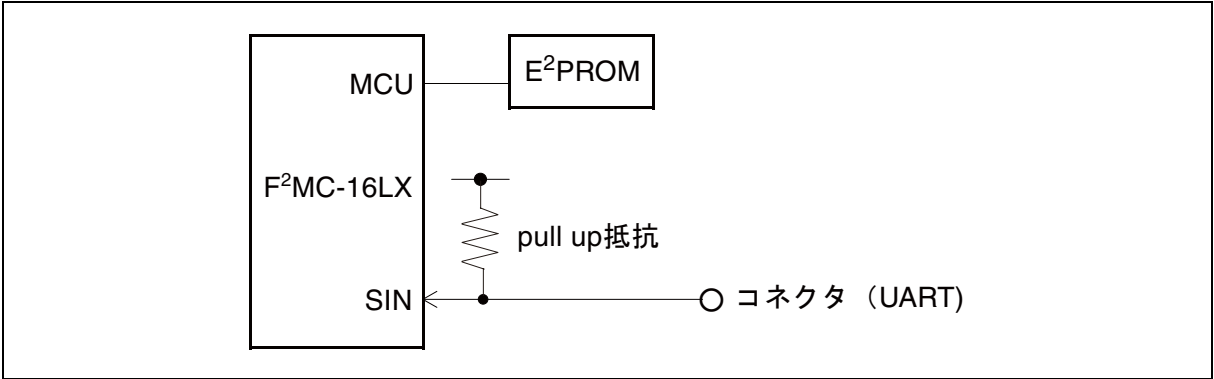


表 21.4-1 E²PROM メモリマップ

アドレス	意味
0000 _H	パッチプログラム No.0 バイト数（0 のときプログラムミスなし）
0001 _H	プログラムアドレス No.0 bit7-0
0002 _H	プログラムアドレス No.0 bit15-8
0003 _H	プログラムアドレス No.0 bit24-16
0004 _H	パッチプログラム No.1 バイト数（0 のときプログラムミスなし）
0005 _H	プログラムアドレス No.1 bit7-0
0006 _H	プログラムアドレス No.1 bit15-8
0007 _H	プログラムアドレス No.1 bit24-16
～ 0010 _H + パッチプログラム No. 0 バイト数	パッチプログラム No.0 本体

- 初期状態
E²PROM はすべて "0" とします。
- プログラムミスが発生した場合
コネクタ (UART) をととして，MCU にパッチプログラムの本体およびプログラムアドレスを転送します。MCU はその情報を E²PROM に書き込みます。

- リセットシーケンス

MCU はリセット後 E²PROM の値を読み出し、もしパッチプログラムのバイト数が "0" でなかった場合は、パッチプログラムの本体を読み出し RAM に書き込みます。プログラムアドレスを PADR0/ADR1 のいずれかに設定して動作を許可します。RAM に書かれたプログラムの先頭番地は、各アドレス検出レジスタごとに定まった RAM に保存します。

- INT9 割込み

割込みルーチン中では、どのアドレス検出で割込みがかかったかをスタックに退避されたプログラムカウンタ (PC) の値より判断し、割込み要求が出力されたプログラムへ分岐します。プログラムへ分岐した場合は、割込みでスタックされた情報は無効となります。

21.5 アドレス一致検出機能のプログラム例

図 21.5-1 に、プログラムパッチ処理例を、図 21.5-2 に、プログラムパッチ処理フローを示します。

■ アドレス一致検出機能のプログラム例

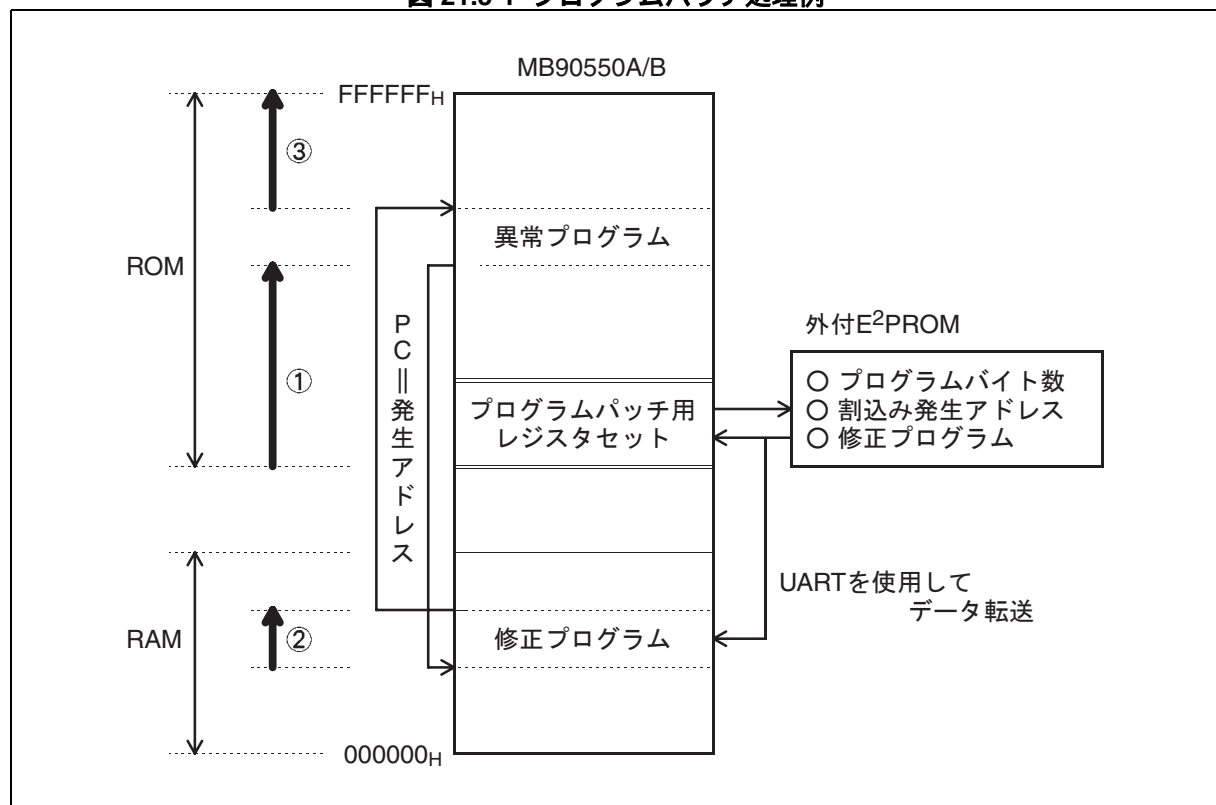
```

movw    a,      sp
movw    rw4,    a      ;スタックポインタの値をRW4へ保存
spb
movl    a,      @rw4+2
movl    r12,    a      ;スタックされているPC+PCB+DPBの値を得る
decl    r12          ;PCの値を1つ補正する(INT9命令の分)
mov     r3,     #00H   ;DPBの部分を00hに初期化
addsp   #12         ;スタックポインタの値を元に戻す
or      ccr,    #040H  ;割込みを許可
mov     a,      padr0+2
movw    a,      padr0  ;アドレス検出レジスタの値をアキュムレータへ
cmpl    a,      r12
beq     check1       ;アドレス検出レジスタとスタックされているPCの値を比較する

```

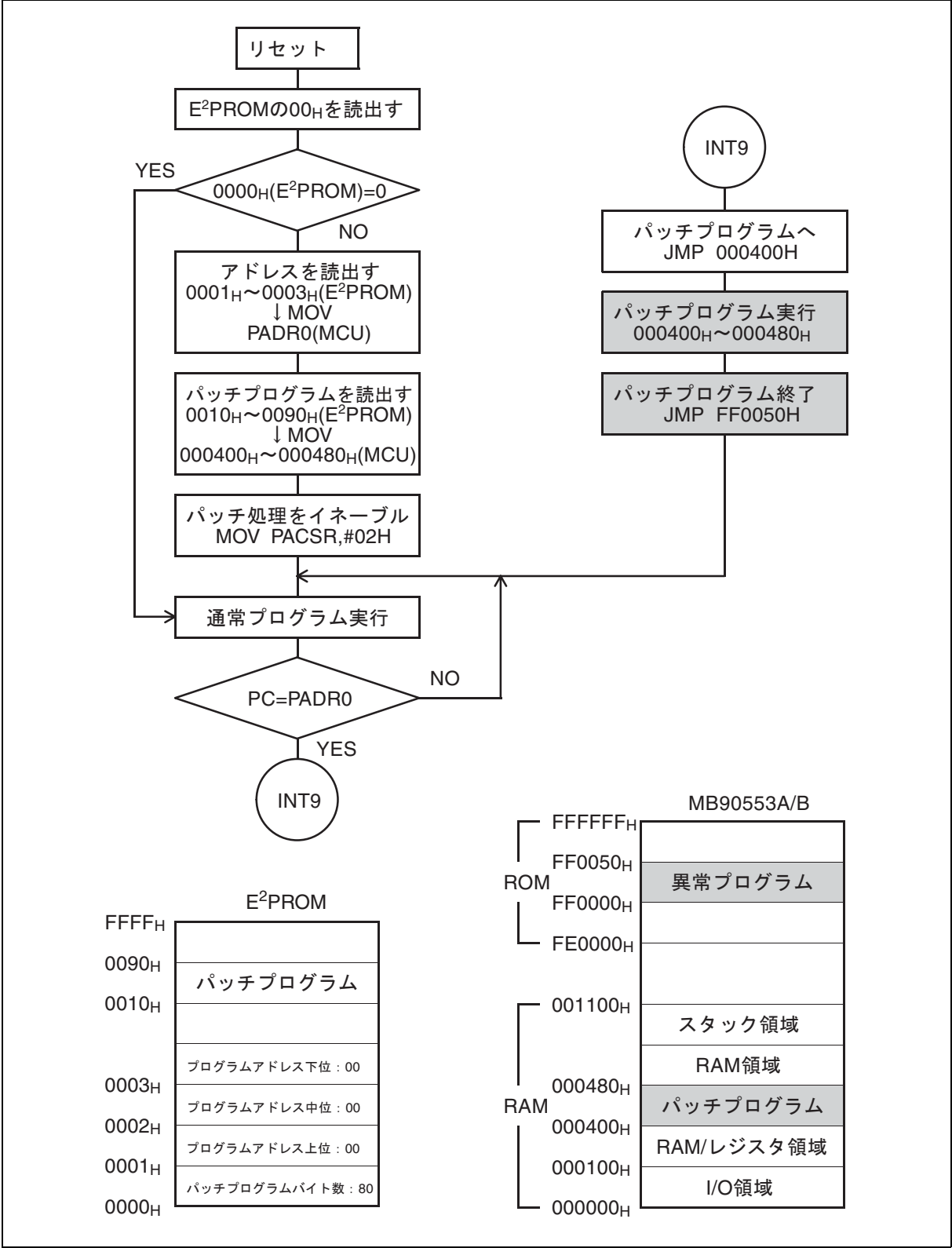
■ プログラムパッチ処理例

図 21.5-1 プログラムパッチ処理例



■ プログラムパッチ処理フロー

図 21.5-2 プログラムパッチ処理フロー



第22章

ROM ミラー機能選択 モジュール

ROM ミラー機能選択モジュールの機能と動作について説明します。

22.1 ROM ミラー機能選択モジュールの概要

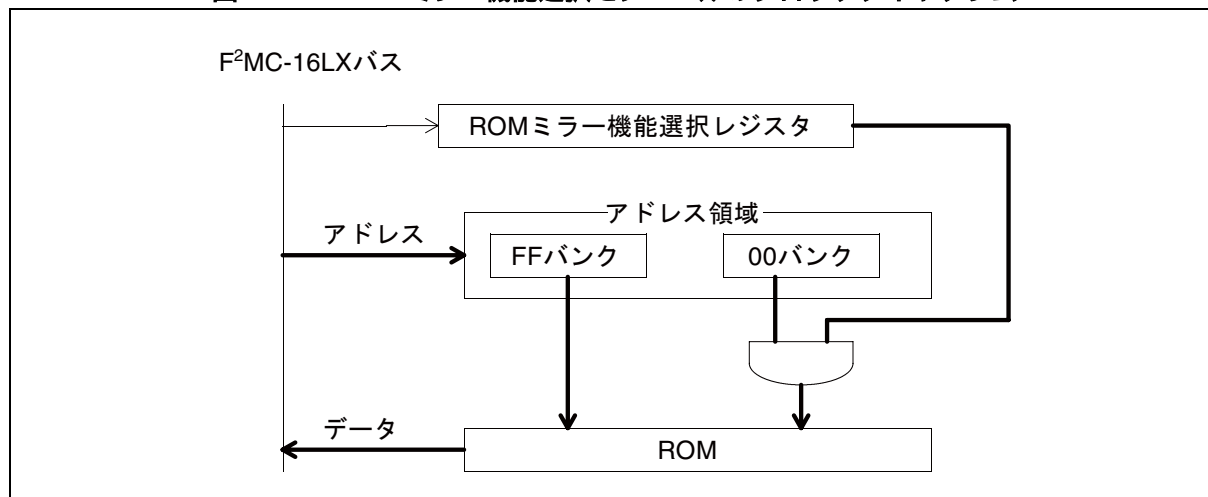
22.2 ROM ミラー機能選択レジスタ (ROMM)

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、ROM が配置されている FF バンクが 00 バンクでみえることをレジスタの設定で選択することができます。

■ ROM ミラー機能選択モジュールのブロックダイアグラム

図 22.1-1 ROM ミラー機能選択モジュールのブロックダイアグラム

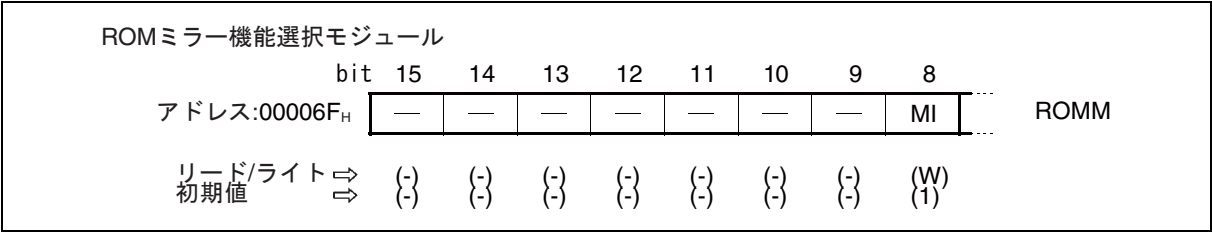


22.2 ROM ミラー機能選択レジスタ (ROMM)

ROM ミラー機能選択レジスタ (ROMM) の MI ビットに "1" を書き込むことにより、FF バンクの ROM データが、00 バンクで読み出せるようになり、"0" を書き込むとこの機能は無効になります。

■ ROM ミラー機能選択レジスタ (ROMM)

図 22.2-1 ROM ミラー機能選択レジスタ (ROMM)



< 注意事項 >
このレジスタをアドレス004000_H ~ 00FFFF_H番地の動作中にアクセスしないでください。

[bit8] MI

MI ビットでは、"1" を書き込み時には FF バンクの ROM のデータが 00 バンクでも読み出せるようになります。"0" を書き込み時には、00 バンクではこの機能は働きます。本ビットは、書き込みのみ可能です。

図 22.2-2 に、シングルチップモード時のメモリ空間を、図 22.2-3 に、内 ROM 外バスモード時のメモリ空間を示します。

< 注意事項 >

ROM のミラー機能を起動している場合、00 バンクの 004000_H ~ 00FFFF_H 番地へ、FF4000_H ~ FFFFF_H 番地がミラーされますので、FF3FFF_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。

表 22.2-1 メモリ空間のアドレス

	MB90552A/B	MB90553A/B	MB90P553A	MB90V550A
アドレス 1	FF0000 _H	FE0000 _H	FE0000 _H	FE0000 _H
アドレス 2	000900 _H	001100 _H	001100 _H	001900 _H

図 22.2-2 シングルチップモード時メモリ空間

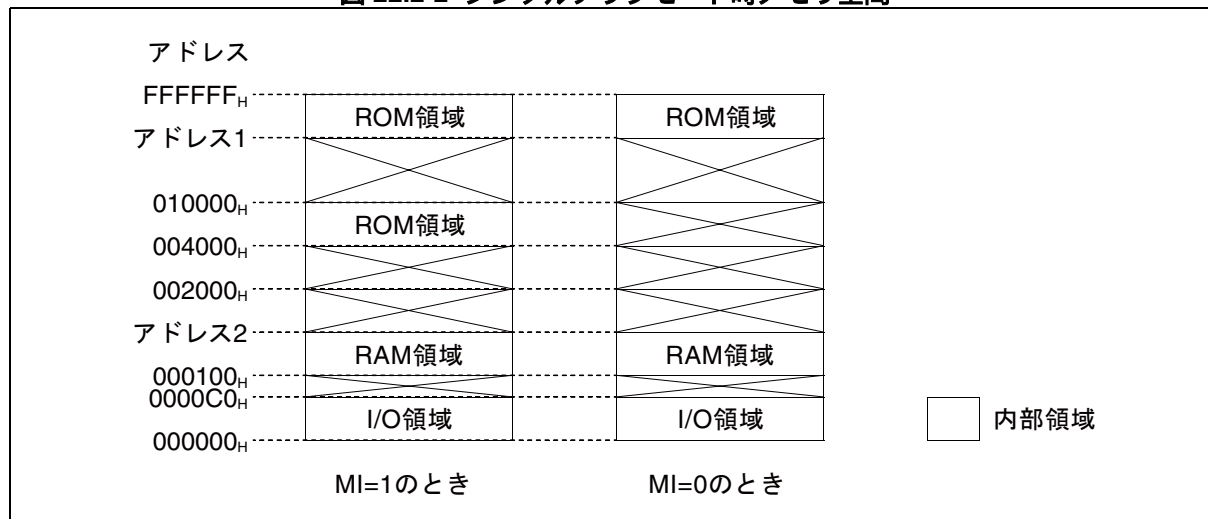
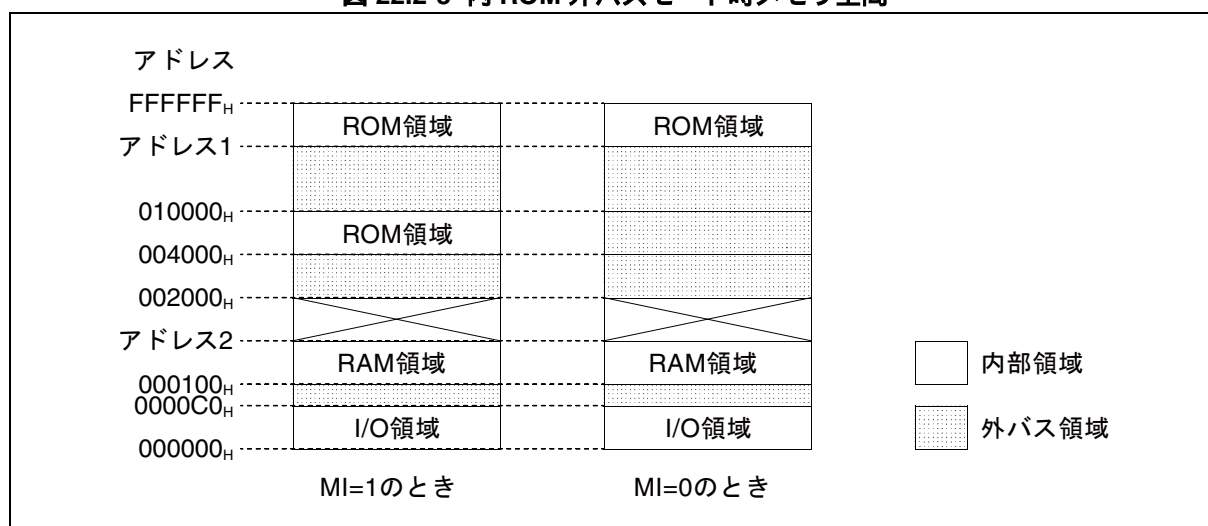


図 22.2-3 内 ROM 外バスモード時メモリ空間



第23章

1M ビットフラッシュ メモリ

1M 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。

フラッシュメモリへのデータ書込み / 消去の方法には、下記の3とおりの方法があります。

1. パラレルライタ

2. シリアル専用ライタ

3. プログラム実行による書込み / 消去

ここでは、"3. プログラム実行による書込み / 消去" について解説します。

23.1 1M ビットフラッシュメモリの概要

23.2 フラッシュメモリ全体のブロックダイアグラムとセクタ構成

23.3 書込み / 消去モード

23.4 フラッシュメモリコントロールステータスレジスタ (FMCS)

23.5 フラッシュメモリ自動アルゴリズム起動方法

23.6 自動アルゴリズム実行状態の確認

23.7 フラッシュメモリ書込み / 消去の詳細説明

23.8 1M ビットフラッシュメモリのプログラム例

23.1 1M ビットフラッシュメモリの概要

1M ビットフラッシュメモリは,CPU メモリマップ上の FE ~ 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り,マスク ROM と同様に CPU からのリード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 からの命令動作で行えます。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行えます。
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。

■ 1M ビットフラッシュメモリの特長

- 128K ワード × 8/64K ワード × 16 ビット (16K+512 × 2+7K+8K+32K+64K)セクタ構成
- 自動プログラムアルゴリズム (Embedded Alogrithm: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 (セクタ組み合わせ自由)
- 書込み / 消去回数 (最小) 10,000 回

Embedded AlogrithmTM は Advanced Micro Devices 社の商標です。

■ フラッシュメモリ書込み / 消去の方法

フラッシュメモリは書込み / 消去と読出しを同時に行うことはできません。すなわち,フラッシュメモリにデータ書込み/消去動作を行う際には,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 実行することで,フラッシュメモリからプログラムアクセスをせず書込み動作だけを行うことが可能となります。

■ フラッシュメモリのレジスタ

- フラッシュメモリコントロールステータスレジスタ (FMCS)

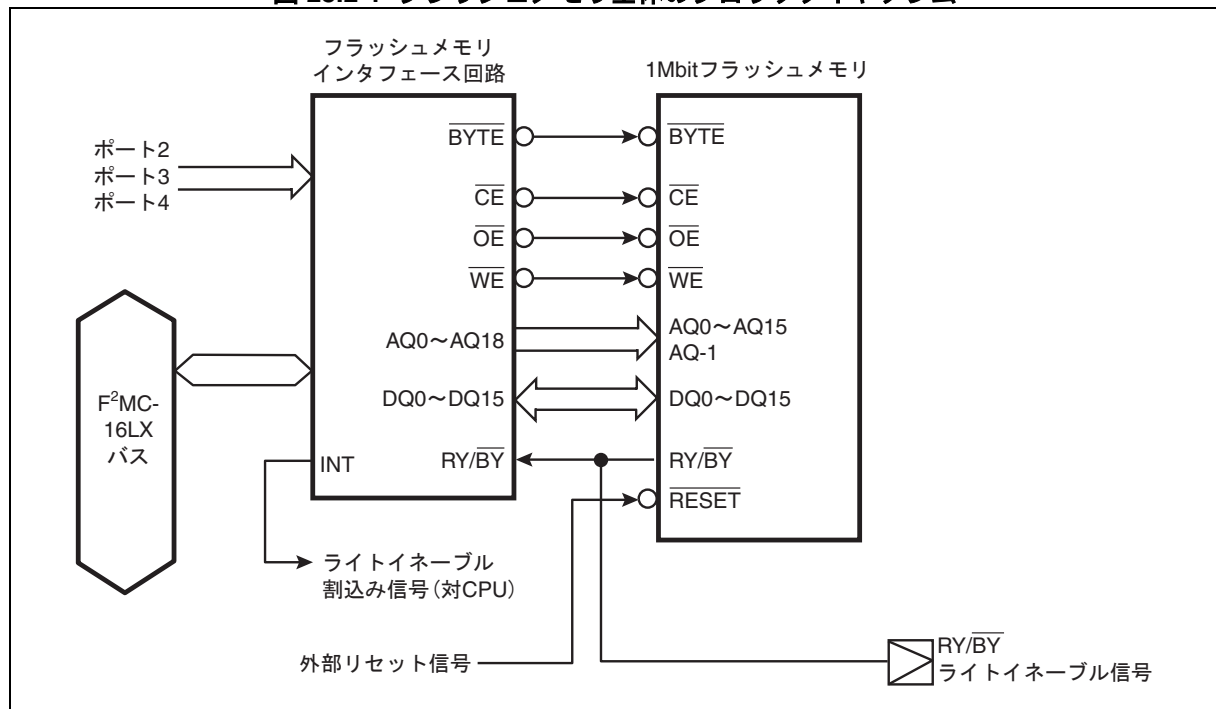
bit	7	6	5	4	3	2	1	0
アドレス:0000AE _H	INTE	RDYINT	WE	RDY	予約	-	-	LPM
リード/ライト	(R/W)	(R/W)	(R/W)	(R)	(W)	(W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(1)	(0)	(-)	(-)	(0)

23.2 フラッシュメモリ全体のブロックダイアグラムとセクタ構成

図 23.2-1 に、フラッシュメモリインタフェース回路付きのフラッシュメモリの全体ブロックダイアグラムを、図 23.2-2 に、フラッシュメモリのセクタ構成を示します。

■ フラッシュメモリ全体のブロックダイアグラム

図 23.2-1 フラッシュメモリ全体のブロックダイアグラム



■ 1M ビットフラッシュメモリのセクタ構成

図 23.2-2 に、1M ビットフラッシュメモリのセクタ構成を示します。図中アドレスには、各セクタの上位アドレスと下位アドレスを示します。

CPU からアクセスする場合 FE バンクレジスタに SA0, FF バンクレジスタに SA1 ~ SA6 が配置されています。

図 23.2-2 1M ビットフラッシュメモリのセクタ構成

フラッシュメモリ	CPUアドレス	ライターアドレス*
SA6 (16Kバイト)	FFFFFF _H	7FFFF _H
	FFC000 _H	7C000 _H
SA5 (512バイト)	FFBFFF _H	7BFFF _H
	FFBE00 _H	7BE00 _H
SA4 (512バイト)	FFBDFF _H	7BDFF _H
	FFBC00 _H	7BC00 _H
SA3 (7Kバイト)	FFBBFF _H	7BBFF _H
	FFA000 _H	7A000 _H
SA2 (8Kバイト)	FF9FFF _H	79FFF _H
	FF8000 _H	78000 _H
SA1 (32Kバイト)	FF7FFF _H	77FFF _H
	FF0000 _H	70000 _H
SA0 (64Kバイト)	FEFFFF _H	6FFFF _H
	FE0000 _H	60000 _H

*:ライターアドレスとは、フラッシュメモリにパラレルライターでデータ書込みを行う際、CPUアドレスに対して相当するアドレスです。汎用ライターを使用し書込み/消去を行う際は、このアドレスで書込み/消去を実行します。

23.3 書込み / 消去モード

フラッシュメモリのアクセスは、フラッシュメモリモードとその他モードの 2 つの異なる方法で行われます。フラッシュメモリモードでは、外部端子から直接書込み / 消去が行え、その他モードでは、内部バスを介して CPU から書込み / 消去が行えます。モードの選択はモード外部端子で行います。

■ フラッシュメモリモード

リセット信号が発生中にモード端子を "111_B" にセットすると、CPU が停止します。フラッシュメモリインタフェース回路は直接ポート 0, 2, 3, 4 に接続しているので、外部端子から直接制御することができます。このモードでは、MCU が外部端子中の標準フラッシュメモリと同様の動作を行い、フラッシュメモリプログラムを使用して書込み / 消去が行えます。

フラッシュメモリモードでは、フラッシュメモリ自動アルゴリズムでサポートされるすべての動作を使用できます。

■ その他モード

フラッシュメモリは、CPU メモリ空間の FC ~ FF バンクに配置されており、通常のマスク ROM と同様にフラッシュメモリインタフェース回路を介して CPU から読出しアクセスおよびプログラムアクセスすることができます。

フラッシュメモリへの書込み / 消去は、フラッシュメモリインタフェース回路を介して CPU からの命令で実行されます。このため、このモードでは MCU が対象ボードにはんだ付けされていても再書込みが可能です。

これらのモードでは、セクタプロテクト動作を実行することはできません。

■ フラッシュメモリの制御信号

表 23.3-1 に、フラッシュメモリモードでのフラッシュメモリ制御信号を示します。

フラッシュメモリ制御信号と MBM29F400TA の外部端子にはほぼ 1 対 1 の対応関係が存在します。セクタプロテクト動作で必要となる V_{ID} (12V) 端子は、MBM29F400TA における A9, \overline{RESET} , \overline{OE} の代わりに、MD0, MD1, MD2 となります。

MB90F553A のメモリ容量は MBM29F400TA の 4 分の 1 ですので、MBM29F400TA のアドレス信号 A17 と A16 に対応する AQ18 と AQ17 端子は冗長となります。これらは常に "1" にセットしてください。

フラッシュメモリモードでは、外部データバス幅は 8 ビットに制限されており、1 バイトのアクセスしか許可されません。DQ15 ~ DQ8 はサポートされていません。 \overline{BYTE} 端子は常に "0" にセットしてください。

表 23.3-1 フラッシュ制御信号

MB90F553A			MBM29F400TA
端子番号	通常機能	フラッシュメモリモード	
1 ~ 8	P20 ~ P27	AQ0 ~ AQ7	A-1, A0 ~ A6
9	P30	AQ16	A15
10	P31	$\overline{\text{CE}}$	$\overline{\text{CE}}$
12	P32	$\overline{\text{OE}}$	$\overline{\text{OE}}$
13	P33	$\overline{\text{WE}}$	$\overline{\text{WE}}$
14, 15	P34, P35	AQ17, AQ18	A16, A17
16	P36	$\overline{\text{BYTE}}$	$\overline{\text{BYTE}}$
17	P37	RY/ $\overline{\text{B}}$	RY/ $\overline{\text{B}}$
18 ~ 22	P40 ~ P44	AQ8 ~ AQ12	A7 ~ A11
24 ~ 26	P45 ~ P47	AQ13 ~ AQ15	A12 ~ A14
49	MD0	MD0	A9 (V_{ID})
50	MD1	MD1	$\overline{\text{RESET}}$ (V_{ID})
51	MD2	MD2	$\overline{\text{OE}}$ (V_{ID})
85 ~ 92	P00 ~ P07	DQ0 ~ DQ7	DQ0 ~ DQ7
77	$\overline{\text{RST}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
使用不可			DQ8 ~ DQ15

23.4 フラッシュメモリコントロールステータスレジスタ (FMCS)

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。

■ コントロールステータスレジスタ (FMCS)

bit	7	6	5	4	3	2	1	0
アドレス: 0000AE _H	INTE	RDYINT	WE	RDY	予約	—	—	LPM
リード/ライト	(R/W)	(R/W)	(R/W)	(R)	(W)	(W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(1)	(0)	(—)	(—)	(0)

[bit7] INTE (INTerrupt Enable)

フラッシュメモリの書込み / 消去の終了で CPU に割り込みを発生させるビットです。INTE ビットが "1" でかつ RDYINT ビットが "1" のときに CPU へ割り込みが発生します。INTE ビットが "0" であれば割り込みは発生しません。

0: 書込み / 消去終了での割り込み禁止

1: 書込み / 消去終了での割り込み許可

[bit6] RDYINT (ReaDY INTerrupt)

フラッシュメモリの動作状態を表すビットです。

フラッシュメモリの書込み / 消去が終わって "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" 書込みによって "0" にクリアされ、"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 (「23.5 フラッシュメモリ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" が読めます。

0: 書込み / 消去動作実行中

1: 書込み / 消去動作終了 (割り込み要求発生)

[bit5] WE (Write Enable)

フラッシュメモリ領域へのライト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（「23.5 フラッシュメモリ自動アルゴリズム起動方法」参照）発行後のライトは、フラッシュメモリ領域へのライトになります。本ビットが "0" のとき、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)

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等のサスペンドコマンドは受け付けられます。

0: 書込み / 消去動作実行中（次データ書込み / 消去不可）

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

試験用予約ビットです。通常使用時は必ず "0" に設定してください。

[bit2, bit1] 空きビット

通常使用時は必ず "0" に設定してください。

[bit0] LPM (Low Power Mode)

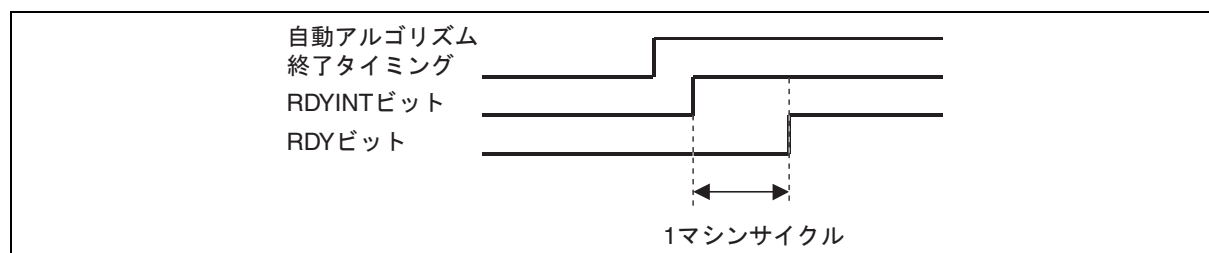
このビットを "1" に設定すると、フラッシュメモリアクセス時のフラッシュメモリへのセレクト信号が最小となり、フラッシュメモリ本体の消費電力が抑えられます。しかし、アクセスタイムは LPM=0 のときに比べて大幅に大きくなりますので、CPU の高速動作時はメモリアクセスが不可能になります。このモードを使用するときは、CPU を 4MHz 以下の周波数で動作させてください。

0: 通常消費電力モード

1: 低電力消費モード（内部動作周波数 4MHz 以下で動作）

< 注意事項 >

RDYINT ビットと RDY ビットは同時には変化しません。どちらか片方のビットで判定するようプログラムを作成してください。



23.5 フラッシュメモリ自動アルゴリズム起動方法

フラッシュメモリの自動アルゴリズムを起動するコマンドには、読出し / リセット、書込み、チップ消去の 4 種類があり、セクタ消去については一時停止と再開の制御が可能です。

■ コマンドシーケンス表

表 23.5-1 に、フラッシュメモリの書込み / 消去時に使用するコマンドの一覧を示します。コマンドレジスタにライトするデータはすべてバイトですが、ワードアクセスでライトする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 23.5-1 コマンドシーケンス表

コマンド シーケンス	バス ライト アクセス	1st バスライト サイクル		2nd バスライト サイクル		3th バスライト サイクル		4th バスライト サイクル		5th バスライト サイクル		6th バスライト サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し / リセット*	1	FxXXXX	XXF0	-	-	-	-	-	-	-	-	-	-
読出し / リセット*	4	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XXF0	RA	RD	-	-	-	-
書込み プログラム	4	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XXA0	PA (even)	PD (word)	-	-	-	-
チップ 消去	6	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX80	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX10
セクタ 消去	6	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX80	FxAAAA	XXAA	Fx5554	XX55	SA (even)	XX30
セクタ消去一時停止		Address "FxXXXX" 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" Data (xx30 _H) の入力で、セクタ消去中の消去一時停止後、消去開始											

(注意事項)

- ・表中のアドレス Fx は、FF、FE 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としてください。
 - ・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。
 - ・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、「フラッシュメモリ全体のブロックダイアグラムとセクタ構成」の 1M ビットフラッシュメモリのセクタ構成」を参照願います。
 - ・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。

23.6 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完了したことを知らせるハードウェアを持ちます。この自動アルゴリズムは、下記のハードウェアシーケンスによって内蔵フラッシュメモリの動作状態の確認ができます。

■ ハードウェアシーケンスフラグ

ハードウェアシーケンスフラグは、DQ7, DQ6, DQ5, DQ3 の 4 ビットの出力で構成されます。それぞれが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) およびセクタ消去タイマフラグ (DQ3) の機能を持ちます。これにより、書込み/チップ・セクタ消去終了、消去コードライトが有効かの確認をすることができます。

ハードウェアシーケンスフラグを参照するには、コマンドシーケンス (「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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にリードアクセスすることで参照できます。表 23.6-1 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 23.6-1 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	-	-	-

自動書込み/チップ・セクタ消去が実行中であるかの判断は、ハードウェアシーケンスフラグを確認するか、フラッシュメモリコントロールレジスタ (FMCS) の RDY ビットを確認することで、書込みが終了しているかを知ることができます。書込み/消去終了後は、読出し/リセット状態に戻ります。実際にプログラムを作成する際には、いずれかのフラグで自動書込み/消去終了を確認後に、データの読出しなどの次処理を行ってください。また、2 回目以降のセクタ消去コードライトが有効であるかについても、ハードウェアシーケンスフラグによって確認することができます。事項に各ハードウェアシーケンスフラグの説明を示します。表 23.6-2 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 23.6-2 ハードウェアシーケンスフラグ機能一覧

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)	$\overline{\text{DQ7}}$ 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	$\overline{\text{DQ7}}$	Toggle	1	0
	チップ・セクタ消去動作	0	Toggle	1	1

23.6.1 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もしくは終了状態であることをデータポーリング機能によって知らせるためのフラグです。

■ データポーリングフラグ (DQ7)

表 23.6-3 と表 23.6-4 に、データポーリングフラグの状態遷移を示します。

表 23.6-3 データポーリングフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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	$\overline{\text{DQ7}}$ DATA:7	0 1	0	0 1	1 0	DATA:7

表 23.6-4 データポーリングフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ消去動作
DQ7	$\overline{\text{DQ7}}$	0

● 書込み動作時

自動書込みアルゴリズム実行中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指し示す番地によらず、最後書き込まれたデータの bit7 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リードアクセスすると、フラッシュメモリはアドレスの指し示す番地の読出し値の bit7 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指し示す番地に関係なく、リード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リードアクセスすると、フラッシュメモリはアドレスの指し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指し示す番地の読出し値の bit7 (DATA:7) を出力します。トグルビットフラグ (DQ6) とともに参照することで、現在セクタ一時停止状態であるか、どのセクタが消去中であるかの判定が可能です。

< 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。データの読出しは、データポーリングフラグ (DQ7) の終了を受けてほかのビットの出力が可能となります。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の次に行うようにしてください。

23.6.2 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)

表 23.6-5 と表 23.6-6 に、トグルビットフラグの状態遷移を示します。

表 23.6-5 トグルビットフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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

表 23.6-6 トグルビットフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	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、連続したリードアクセスを行うと、フラッシュメモリはアドレスの指し示す番地によらず、リードごとに "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、アドレスの指し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指し示す番地が消去中のセクタに属するならば、"1" を出力します。消去中のセクタに属しないのであれば、アドレスの指し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き換え保護されているセクタの場合は、約 2μs 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き換え保護されている場合トグルビットは約 100μs のトグル動作をし、その後データを書き換えしないで読出し / リセット状態に戻ります。

23.6.3 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)

表 23.6-7 と表 23.6-8 に、タイミングリミット超過フラグの状態遷移を示します。

表 23.6-7 タイミングリミット超過フラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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

表 23.6-8 タイミングリミット超過フラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	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、規定時間を超えてしまっている場合は "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無関係ですので、書込み / 消去が成功したか失敗したかの判定が可能です。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もしくはトグルビット機能により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することができます。

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。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、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。

23.6.4 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)

表 23.6-9 と表 23.6-10 に、セクタ消去タイマフラグの状態遷移を示します。

表 23.6-9 セクタ消去タイマフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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

表 23.6-10 セクタ消去タイマフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	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リードアクセスすると、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指し示す番地によらず、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り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ライトまたは消去一時停止以外のコマンドは、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ライトを受け付けます。このことを確認するために、引き続くセクタ消去コードのライトに先立ちこのフラグの状態をチェックすることを推奨します。もし 2 回目の状態チェックで "1" であったなら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。

● セクタ消去動作時

セクタ消去一時停止中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指し示す番地が、消去中のセクタに属するならば "1" を出力します。消去中のセクタに属しないのであれば、アドレスの指し示す番地の読出し値の bit3(DATA:3) を出力します。

23.7 フラッシュメモリ書込み / 消去の詳細説明

自動アルゴリズムを起動するコマンドを発行し、フラッシュメモリに読出し / リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、セクタ消去再開のそれぞれの動作を行う手順を説明します。

■ フラッシュメモリ書込み / 消去の詳細説明

フラッシュメモリは読出し / リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、消去再開の動作がコマンドシーケンス(「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 を参照)のバスへのライトサイクルを行うことで自動アルゴリズムを実行することが可能です。それぞれのバスへのライトサイクルは必ず続けて行う必要があります。また、自動アルゴリズムはデータ・ポーリング機能などで終了時を知ることができます。正常終了後は読出し / リセット状態に戻ります。

各動作について、次項より下記の順に示します。

- 読出し / リセット状態にする
- データを書き込む
- 全データを消去する(チップ全消去)
- 任意のデータを消去する(セクタ消去)
- セクタ消去を一時停止する
- セクタ消去を再開する

23.7.1 フラッシュメモリを讀出し / リセット状態にする

讀出し / リセットコマンドを発行し、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

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（「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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可能です。

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、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リードアクセスでデータを読み出せます。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。

23.7.2 フラッシュメモリヘデータを書き込む

書込みコマンドを発行し、フラッシュメモリヘデータを書き込む手順について説明します。

■ フラッシュメモリヘデータを書き込む

フラッシュメモリの、データ書込み自動アルゴリズムを起動するには、コマンドシーケンス表（「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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可能です。4 サイクル目に目的のアドレスへのデータライトが終了した時点で、自動アルゴリズムが起動され自動書込みが開始します。

● 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。

すなわち、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

書込みによって、データ "0" をデータ "1" に戻すことはできません。データ "0" にデータ "1" を書き込むと、データポーリングアルゴリズム (DQ7) またはトグル動作 (DQ6) が終了せず、フラッシュメモリ素子が不良と判定され、書込み規定時間を超えタイミングリミット超過フラグ (DQ6) がエラーと判定するか、あるいは見かけ上データ "1" が書き込まれたように見えるかのどちらかとなります。しかし、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。消去動作のみが "0" データを "1" にすることができます。

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ので注意が必要です。

■ フラッシュメモリ書込み手順

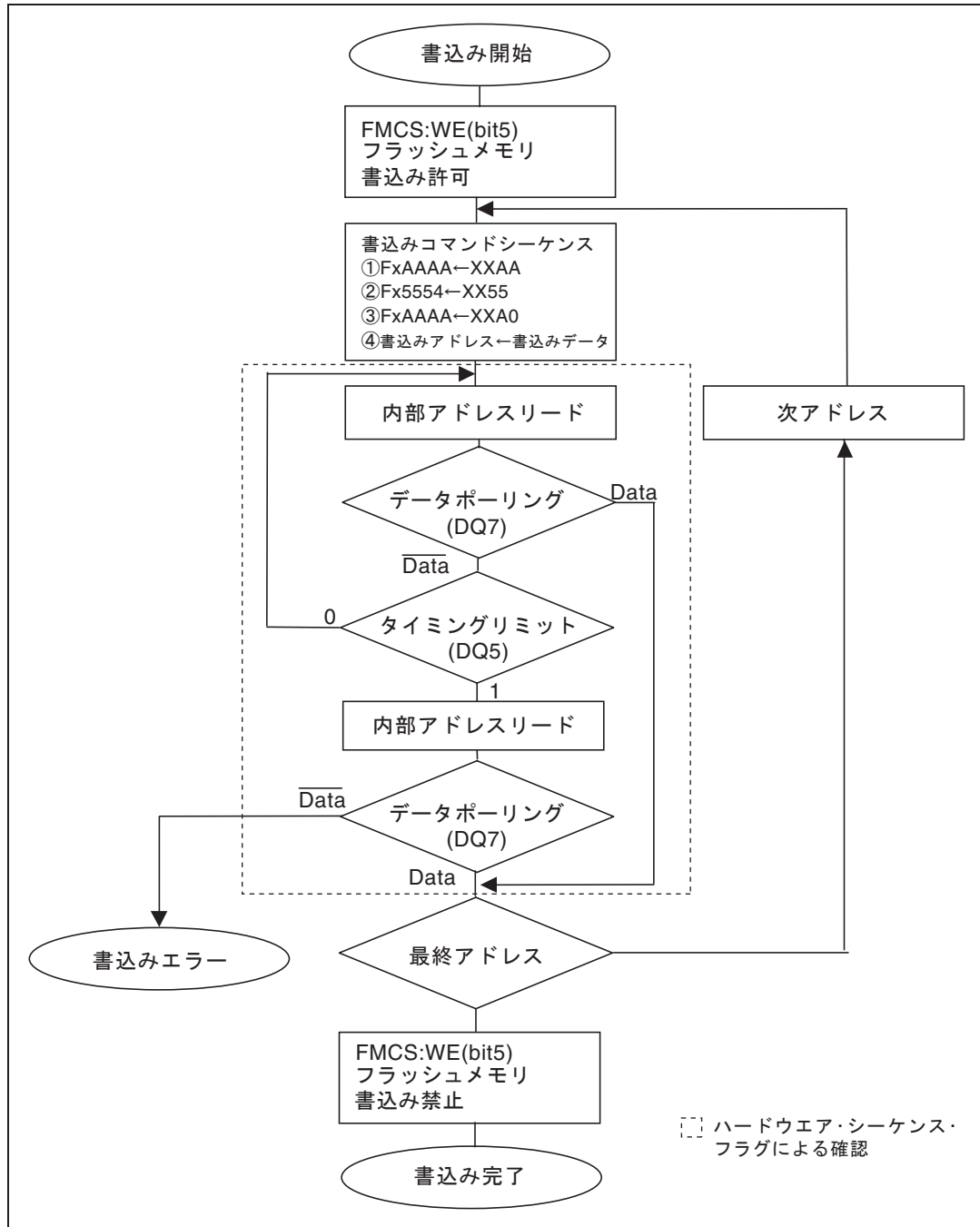
図 23.7-1 にフラッシュメモリ書込みの手順の例を示します。ハードウェアシーケンスフラグ（「23.6 自動アルゴリズム実行状態の確認」参照）を用いることでフラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。ここでは、書込み終了の確認にデータポーリングフラグ (DQ7) を用いています。

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) は再チェックする必要があります。

トグルビットフラグ (DQ6) で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 23.7-1 フラッシュメモリ書き込み手順の例



23.7.3 フラッシュメモリの全データを消去する (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データを消去する (チップ消去)

フラッシュメモリから全てのデータを消去するには、コマンドシーケンス表(「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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実行可能です。

チップ消去コマンドは6回のバス動作で行われます、6サイクル目のライト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。

23.7.4 フラッシュメモリの任意のデータを消去する (セクタ消去)

セクタ消去コマンドを発行し、フラッシュメモリの任意のデータを消去する (セクタ消去) 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可能です。

■ フラッシュメモリの任意のデータを消去する (セクタ消去)

フラッシュメモリの任意のセクタを消去するためには、コマンドシーケンス表 (「23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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で実行可能です。

● 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) を書き込みすることにより 50 μ s のセクタ消去ウェイトが開始します。複数のセクタ消去を行う場合は、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) を書き込みします。

● 複数のセクタを指定するときの注意

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。すなわち、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。それ以降では受け付けられないことがあります。引き続くセクタ消去コードのライト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リードするアドレスは、消去しようとしているセクタを指すようにします。

■ フラッシュメモリのセクタ消去手順

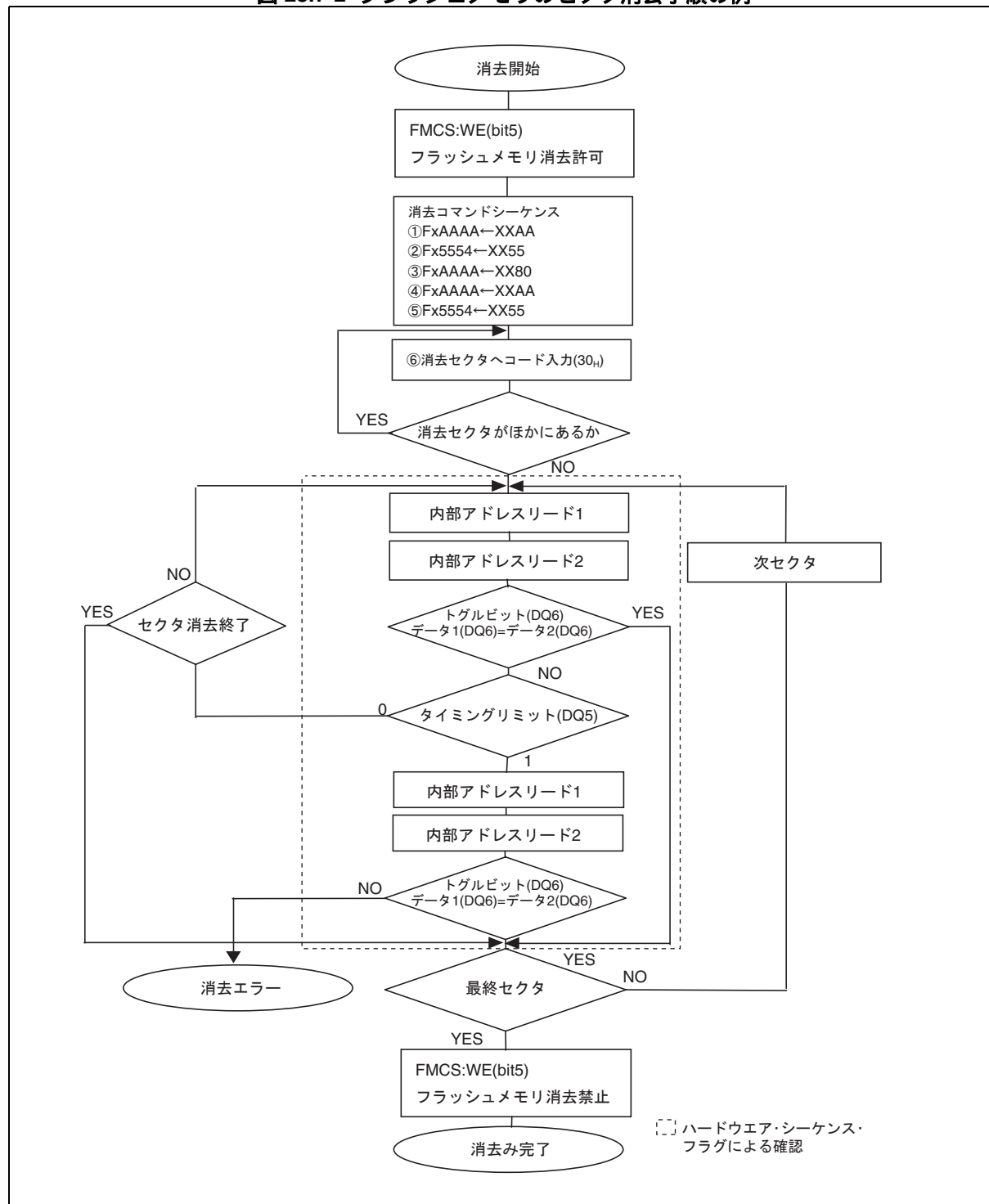
ハードウェアシーケンスフラグ (「23.6 自動アルゴリズム実行状態の確認」を参照) を用いることでフラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。図 23.7-2 にフラッシュメモリセクタ消去の手順の例を示します。ここでは、消去終了の確認にトグルビットフラグ (DQ6) を用いています。

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) は再チェックする必要があります。

データポーリングフラグ (DQ7) でも同様に、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を再チェックする必要があります。

図 23.7-2 フラッシュメモリのセクタ消去手順の例



23.7.5 フラッシュメモリのセクタ消去を一時停止する

セクタ消去一時停止コマンドを発行し、フラッシュメモリセクタ消去の一時停止を行う手順について説明します。消去中でないセクタから、データを読みだすことが可能です。

■ フラッシュメモリのセクタ消去を一時停止する

フラッシュメモリのセクタ消去を一時停止するには、コマンドシーケンス表（23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 を参照）のセクタ消去一時停止コマンドを、フラッシュメモリ内に送ることで実行可能です。

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するものです。この状態では読出しのみが可能で書込みはできません。この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。

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が、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指すようにします。消去一時停止で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 15 μ s の時間の後、消去一時停止状態に入ります。

23.7.6 フラッシュメモリのセクタ消去を再開する

セクタ消去再開コマンドを発行し、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

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（23.5 フラッシュメモリ自動アルゴリズム起動方法」の表 23.5-1 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可能です。

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指すようにします。

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。

23.8 1M ビットフラッシュメモリのプログラム例

ここでは、1M ビットフラッシュメモリのプログラム例を掲載します。

■ 1M ビットフラッシュメモリのプログラム例

```

NAME      FLASHWE
TITLE     FLASHWE
;-----
;1Mbit-FLASH サンプルプログラム
;
;1: FLASHにあるプログラム (アドレス FFBC00H セクタ SA4) をRAM (アドレス 000700H)
に転送する。
;2: RAM上でプログラムを実行する。
;3: PDR1の値をFLASH (アドレス FE0000Hセクタ SA0) に書き込む。
;4: 書込んだ値 (アドレス FE0000H セクタ SA0) を読み出しPDR2に出力する。
;5: 書込んだセクタ (SA0) を消去する。
;6: 消去データ確認の出力
; 条件
;
;   ・ RAM転送バイト数: 100H (256B)
;   ・ 書込み, 消去の終了判定
;       DQ5 (タイミングリミット超過フラグ) での判定
;       DQ6 (トグルビットフラグ) での判定
;       RDY (FMCS) での判定
;   ・ エラー時の処理
;       P00~P07にHiを出力する
;       リセットコマンド発行
;-----
;
;
RESOUS    IOSEG    ABS=00    ;"RESOUS" I/Oセグメントの定義
          ORG      0000H
PDR0      RB       1
PDR1      RB       1
PDR2      RB       1
PDR3      RB       1
          ORG      0010H
DDR0      RB       1
DDR1      RB       1
DDR2      RB       1
DDR3      RB       1
          ORG      00A1H
CKSCR     RB       1
          ORG      00AEH
FMCS      RB       1
          ORG      006FH
ROMM      RB       1
RESOUS    ENDS
;
SSTA      SSEG
          RW       0127H
STA_T     RW       1
SSTA      ENDS
;
DATA      DSEG     ABS=0FFH      ;FLASHコマンドアドレス
          ORG      5554H
COMADR2   RW       1

```

第 23 章 1M ビットフラッシュメモリ

```

ORG      0AAAAH
COMADR1  RW      1
DATA     ENDS
;//////////////////////////////////////
;メインプログラム (FFA000H)
;//////////////////////////////////////
CODE     CSEG
START:
;//////////////////////////////////////
;初期化
;//////////////////////////////////////
MOV      CKSCR, #0BAH      ;3通倍に設定
MOV      RP, #0
MOV      A, #!STA_T
MOV      SSB, A
MOVW     A, #STA_T
MOVW     SP, A
MOV      ROMM, #00H        ;ミラーOFF
MOV      PDR0, #00H        ;エラー確認用
MOV      DDR0, #0FFH
MOV      PDR1, #00H        ;データ入力用ポート
MOV      DDR1, #00H
MOV      PDR2, #00H        ;データ出力用ポート
MOV      DDR2, #0FFH
;//////////////////////////////////////
;RAM (700H番地) に"FLASH書き込み消去プログラム (FFBC00H)"を転送する
;//////////////////////////////////////
MOVW     A, #0700H          ;転送先RAM領域
MOVW     A, #0BC00H          ;転送元アドレス (プログラムのある位置)
MOVW     RW0, #100H          ;転送するバイト数
MOVS     ADB, PCB            ;FFBC00Hから000700Hへ100H転送
CALLP    000700H             ;転送したプログラムのあるアドレスへジャンプ
;//////////////////////////////////////
;データ出力
;//////////////////////////////////////
OUT      MOV      A, #0FEH
        MOV      ADB, A
        MOVW     RW2, #0000H
        MOVW     A, @RW2+00
        MOV      PDR2, A
END      JMP      *
CODE     ENDS
;//////////////////////////////////////
;FLASH書き込み消去プログラム (SA4)
;//////////////////////////////////////
RAMPRG   CSEG      ABS=0FFH
        ORG      0BC00H
;
;
;
;
        MOVW     RW0, #0500H      ;RW0:入力データ確保用RAM空間 00:0500~
        MOVW     RW2, #0000H      ;RW2:フラッシュメモリ書き込みアドレス FD:0000~
        MOV      A, #00H          ;DTB変更
        MOV      DTB, A           ;@RW0用バンク指定
        MOV      A, #0FEH        ;ADB変更1

```

```

MOV     ADB,  A           ;書き込みモード指定アドレス用バンク指定
MOV     PDR3,  #00H       ;スイッチ初期化
MOV     DDR3,  #00H
;
WAIT1   BBC     PDR3:0, WAIT1   ;PDR3:0 Hiで書き込みスタート
;
;////////////////////////////////////
; 書き込み (SA0)
;////////////////////////////////////
MOV     A,      PDR1
MOVW    @RW0+00,  A         ;RAMにPDR1データを確保
MOV     FMCS,   #20H       ;書き込みモード設定
MOVW    ADB:COMADR1,  #00AAH ;フラッシュ書き込みコマンド1
MOVW    ADB:COMADR2,  #0055H ;フラッシュ書き込みコマンド2
MOVW    ADB:COMADR1,  #00A0H ;フラッシュ書き込みコマンド3
;
MOVW    A,      @RW0+00     ;入力データ (RW0) をフラッシュメモリ (RW2)
                           ;に書き込む
;
MOVW    @RW2+00,  A
WRITE   ;待ち時間チェック
;
;////////////////////////////////////
; タイムリミット超過チェック-フラグが立ちトグル動作中であるときERROR
;////////////////////////////////////
MOVW    A,      @RW2+00
AND     A,      #20H       ;DQ5タイムリミットチェック
BZ      NTOW             ;タイムリミットオーバー
MOVW    A,      @RW2+00     ;AH
MOVW    A,      @RW2+00     ;AL
XORW    A
AND     A,      #40H       ;DQ6トグルビットは違っているか
BNZ     ERROR            ;違えばERRORへ
;
;////////////////////////////////////
; 書き込み終了チェック (FMCS-RDY)
;////////////////////////////////////
NTOW    MOVW    A,  FMCS
AND     A,      #10H       ;FMCS RDYビット (4bit) 抽出
BZ      WRITE            ;書き込み終了か?
MOV     FMCS,   #00H       ;書き込みモード解除
;////////////////////////////////////
; 書き込みデータ出力
;////////////////////////////////////
MOVW    RW2,    #0000H     ;書き込みデータ出力
MOVW    A,      @RW2+00
MOV     PDR2,   A
;
WAIT2   BBC     PDR3:1, WAIT2   ;PDR3:1 Hiでセクタ消去スタート
;
;////////////////////////////////////
; セクタ消去 (SA0)
;////////////////////////////////////
MOV     @RW2+00,  #0000H    ;アドレス初期化
MOV     FMCS,   #20H       ;消去モード設定
MOVW    ADB:COMADR1,  #00AAH ;フラッシュ消去コマンド1
MOVW    ADB:COMADR2,  #0055H ;フラッシュ消去コマンド2
MOVW    ADB:COMADR1,  #0080H ;フラッシュ消去コマンド3
MOVW    ADB:COMADR1,  #00AAH ;フラッシュ消去コマンド4

```

第 23 章 1M ビットフラッシュメモリ

```

MOVW   ADB:COMADR2,    #0055H      ;フラッシュ消去コマンド5
MOV     @RW2+00,        #0030H      ;消すセクタに消去コマンド発行6
ELS
;
; //////////////////////////////////////
;   タイムリミット超過チェック       フラグが立ちトグル動作中であるときERROR
;   //////////////////////////////////////
MOVW    A,              @RW2+00
AND     A,              #20H        ;DQ5タイムリミットチェック
BZ      NTOE            ;タイムリミットオーバー
MOVW    A,              @RW2+00      ;AH   書込み動作中は, DQ6から
MOVW    A,              @RW2+00      ;AL   リード毎Hi Lowが交互出力される
XORW    A               ;AHとALのXOR (DQ6の値が違えば1書込み動作
                           ;中である)
AND     A,              #40H        ;DQ6トグルビットはHiか
BNZ     ERROR           ;HiならERRORへ
;
;   //////////////////////////////////////
;   消去終了チェック (FMCS-RDY)
;   //////////////////////////////////////
NTOE    MOVW    A, FMCS              ;
AND     A,      #10H              ;FMCS RDYビット (4bit) 抽出
BZ      ELS                      ;セクタ消去終了か?
MOV     FMCS,    #00H              ;FLASH消去モード解除
RETP                                ;メインプログラムに戻る
;////////////////////////////////////
;エラー
;////////////////////////////////////
ERROR    MOV     FMCS, #00H          ;FLASHモード解除
MOV      PDRO,   #0FFH             ;エラー処理の確認
MOV      ADB:COMADR1, #0F0H        ;リセットコマンド (読出しが可能になる)
RETP                                ;メインプログラムに戻る
RAMPRG   ENDS
;////////////////////////////////////
VECT     CSEG     ABS=0FFH
ORG      0FFDCH
DSL      START
DB       00H
VECT     ENDS
;
END      START

```

第24章

MB90F553A 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、シリアル書込みの接続例につい
て説明します。

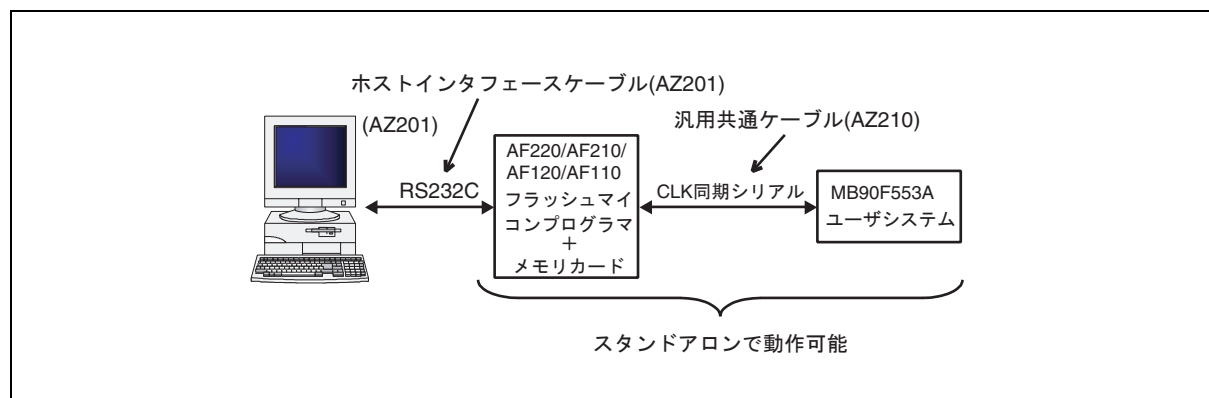
- 24.1 MB90F553A シリアル書込み接続の基本構成
- 24.2 シリアル書込み接続例（ユーザ電源使用時）
- 24.3 シリアル書込み接続例（ライターから電源供給時）
- 24.4 フラッシュマイコンプログラマとの最小限の接続例
（ユーザ電源使用時）
- 24.5 フラッシュマイコンプログラマとの最小限の接続例
（ライターから電源供給時）

24.1 MB90F553A シリアル書込み接続の基本構成

MB90F553A 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。

■ MB90F553A 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 AF220/AF210/AF120/AF110 フラッシュマイコンプログラマを使用します。



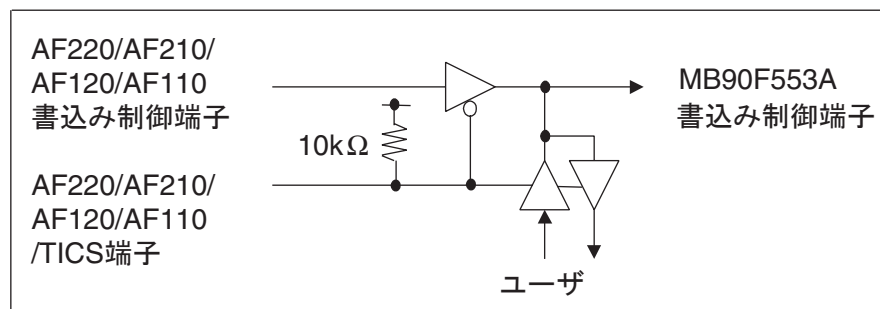
< 注意事項 >

AF220/AF210/AF120/AF110 フラッシュマイコンプログラマの機能，操作方法および接続用汎用共通ケーブル（AZ210），コネクタにつきましては，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。

表 24.1-1 富士通標準シリアルオンボード書込みに使用する端子

端 子	機 能	補足説明
MD2, MD1, MD0	モード端子	フラッシュマイコンプログラマから、書込みモードに制御します。
X0, X1	発振用端子	書込みモード時に、CPU 内部動作クロックは PLL クロック 1 通倍となっております。従いまして、発振クロック周波数が、内部動作クロックとなりますので、シリアル書換え時に使用する発振子は 1MHz ~ 16MHz となります。
P00, P01	書込みプログラム起動端子	-
$\overline{\text{RST}}$	リセット端子	-
SIN	シリアルデータ入力端子	UART を CLK 同期モードとして使用します。
SOT	シリアルデータ出力端子	
SCK	シリアルクロック入力端子	
C	C 端子	電源安定化の容量端子です。外部に 0.1 μ F 程度のセラミックコンデンサを接続してください。
V _{CC}	電源電圧供給端子	書込み電圧 (5V \pm 10%) をユーザシステムから供給する場合には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。接続時にはユーザ側の電源と短絡しないようにしてください。
V _{SS}	GND 端子	フラッシュマイコンプログラマの GND と共通にします。
$\overline{\text{HST}}$	ハードウェアスタンバイ端子	シリアル書込みモード中は "H" レベルを入力してください。

P00, SIN, SOT, SCK 端子をユーザシステムでも使用する場合には、下図の制御回路が必要となります。フラッシュマイコンプログラマの /TICS 信号により、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。



24.2 項 ~ 24.5 項に、以下のシリアル書込み接続例を示しますのでご参照ください。

- シリアル書込み接続例 (ユーザ電源使用時)
- シリアル書込み接続例 (ライターから電源供給時)
- フラッシュマイコンプログラマとの最小限の接続例 (ユーザ電源使用時)
- フラッシュマイコンプログラマとの最小限の接続例 (ライターから電源供給時)

表 24.1-2 フラッシュマイコンプログラマシステム構成（横河デジタルコンピュータ株式会社製）

型 格		機 能
本体	AF200/AC4P	イーサネットインタフェース内蔵モデル /100V ~ 220V 電源アダプタ
	AF210/AC4P	スタンダードモデル /100V ~ 220V 電源アダプタ
	AF120/AC4P	単キーイーサネットインタフェース内蔵モデル /100V ~ 220V 電源アダプタ
	AF110/AC4P	単キーモデル /100V ~ 220V 電源アダプタ
AZ221		ライター専用 PC/AT 用 RS232C ケーブル
AZ210		標準ターゲットプローブ (a) 長さ :1m
FF201		富士通製 F 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		2MB PC Card (Option) FLASH メモリ容量 ~ 128KB 対応
/P4		4MB PC Card (Option) FLASH メモリ容量 ~ 512KB 対応

問い合わせ先：横河デジタルコンピュータ株式会社
電話 :042-333-6224

< 注意事項 >

AF200 フラッシュマイコンプログラマは終息製品ですが，コントロールモジュール FF201 を用いることにより使用できます。シリアル書込み接続例をシリアル書込み接続例（ユーザ電源使用時）を 24.2 節以降に示します。

■ 発振クロック周波数とシリアルクロック入力周波数

MB90F553A の入力可能なシリアルクロック周波数は、以下の計算により求められることができます。ご使用の発振クロック周波数に合わせて、シリアルクロック入力周波数をフラッシュマイコンプログラムに設定してください。

入力可能なシリアルクロック周波数 = $0.125 \times$ 発振クロック周波数

表 24.1-3 入力可能なシリアルクロック周波数の例

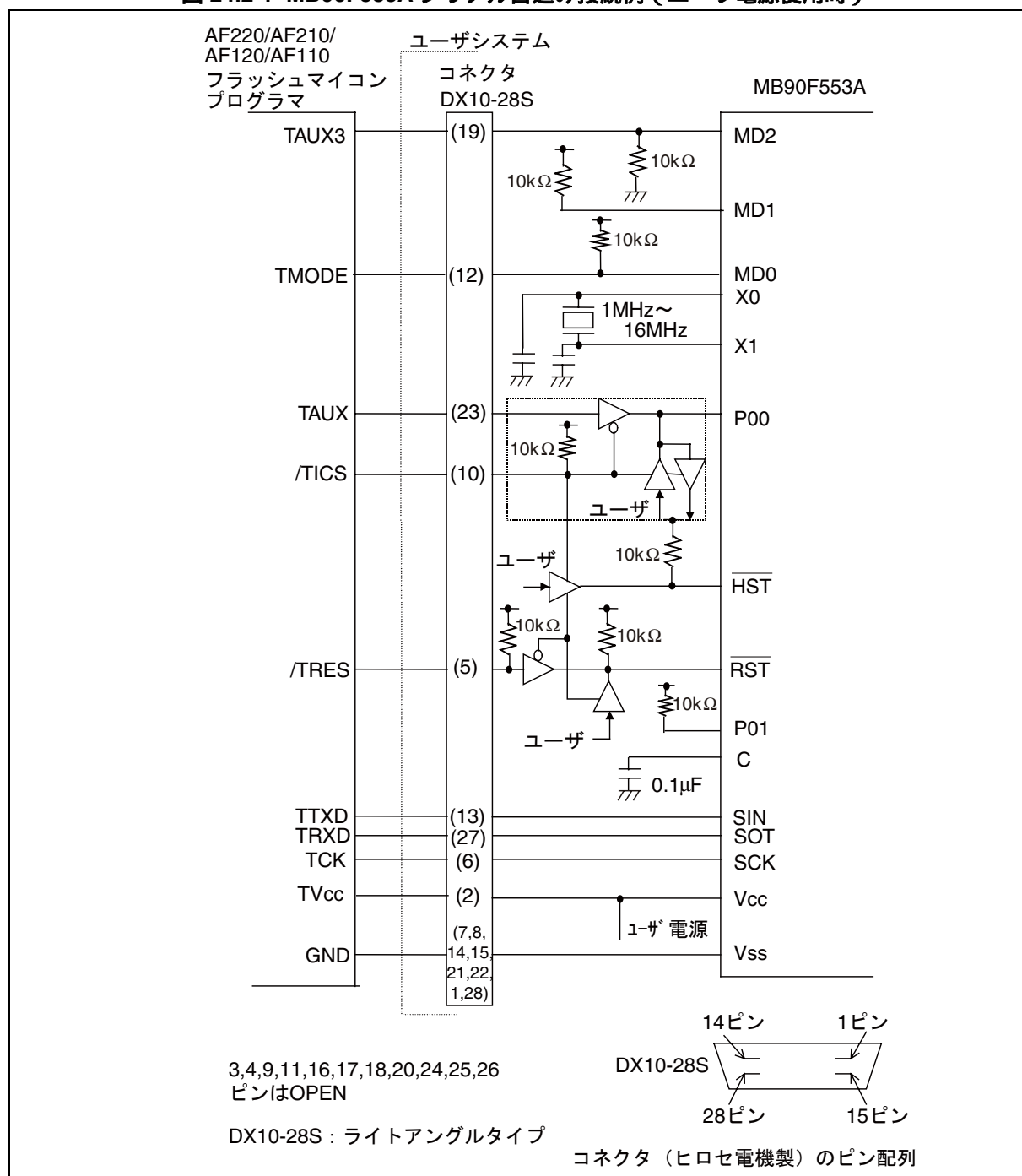
発振クロック周波数	マイコンの入力可能な最大シリアルクロック周波数	AF220/AF210/AF120/AF110 の設定可能な最大シリアルクロック周波数	AF200 の設定可能な最大シリアルクロック周波数
4MHz 時	500kHz	500kHz	500kHz
8MHz 時	1MHz	800kHz	500kHz
16MHz 時	2MHz	1.25MHz	500kHz

24.2 シリアル書き込み接続例（ユーザ電源使用時）

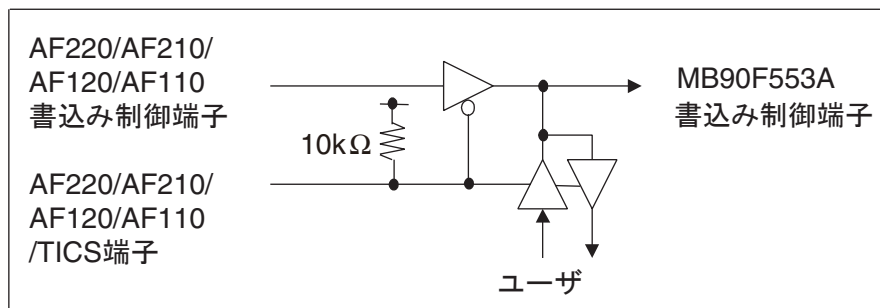
図 24.2-1 に、マイコンの電源電圧をユーザ電源より供給する場合の、シリアル書き込み接続例を示します。なお、モード端子は、内部ベクタモード（シングルチップモード、内 ROM 外バスモード）MD2, MD1, MD0=011 です。

■ シリアル書き込み接続例（ユーザ電源使用時）

図 24.2-1 MB90F553A シリアル書き込み接続例（ユーザ電源使用時）



- SIN, SOT, SCK 端子をユーザシステムでも使用する場合には, P00 と同様下図の制御回路が必要となります (フラッシュマイコンプログラムの /TICS 信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



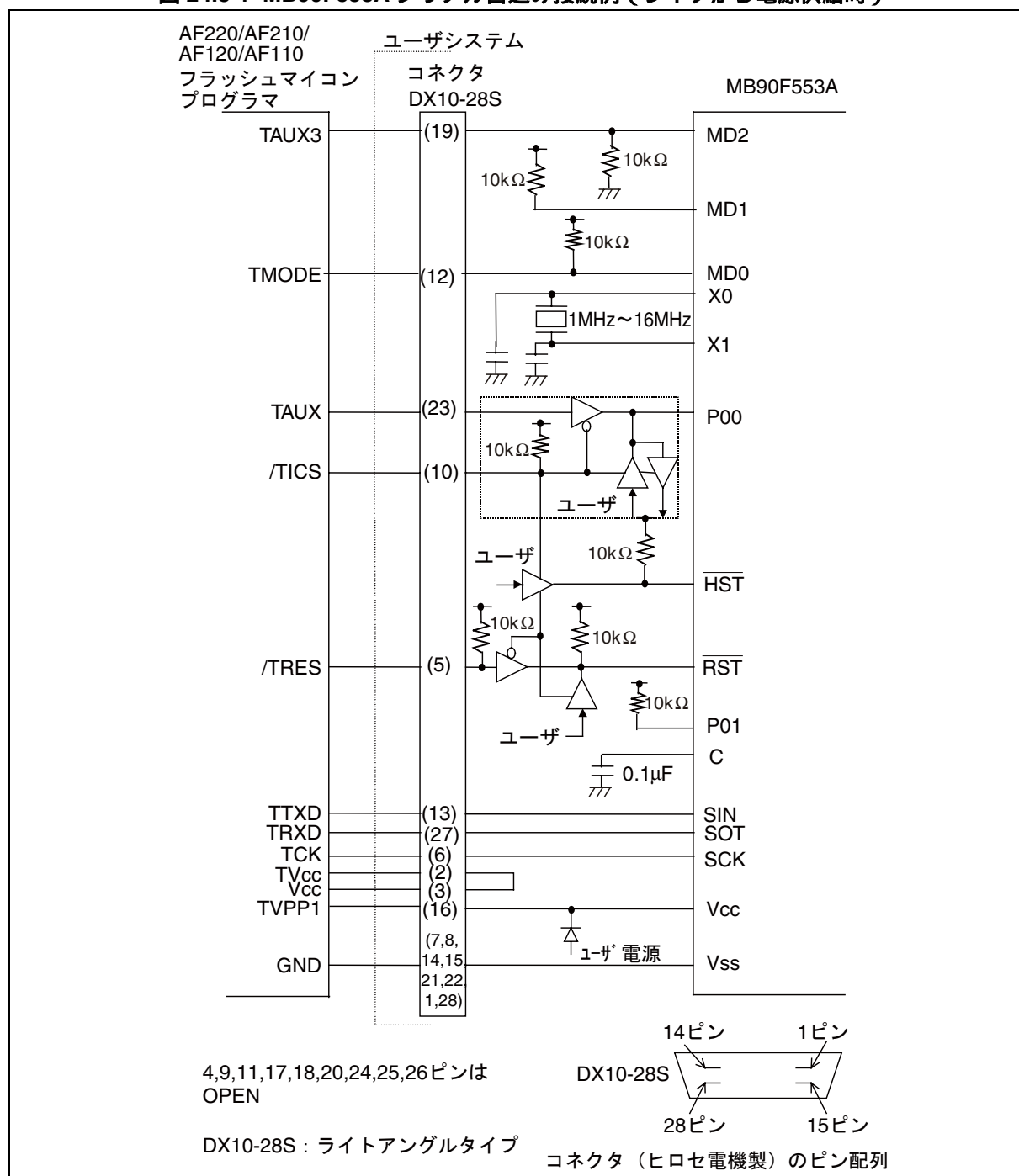
- AF220/AF210/AF120/AF110 との接続はユーザ電源が OFF の状態で行ってください。

24.3 シリアル書き込み接続例（ライターから電源供給時）

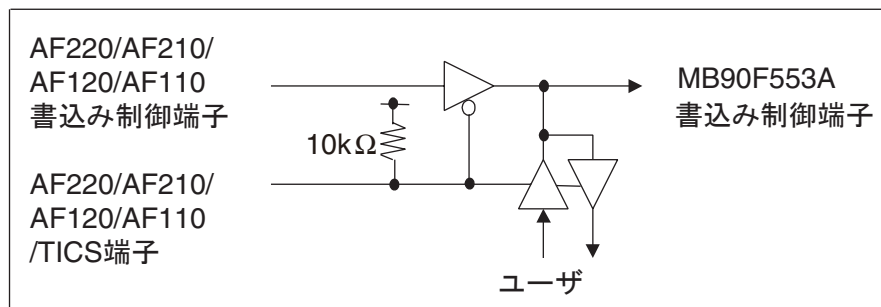
図 24.3-1 に、マイコンの電源電圧をライター電源より供給する場合の、シリアル書き込み接続例を示します。なお、モード端子は、内部ベクタモード（シングルチップモード、内 ROM 外バスモード）MD2, MD1, MD0=011 です。

■ シリアル書き込み接続例（ライターから電源供給時）

図 24.3-1 MB90F553A シリアル書き込み接続例（ライターから電源供給時）



- SIN, SOT, SCK 端子をユーザシステムでも使用する場合には, P00 と同様下図の制御回路が必要となります (フラッシュマイコンプログラムの /TICS 信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110 との接続はユーザ電源が OFF の状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給するときはユーザ電源と短絡しないでください。

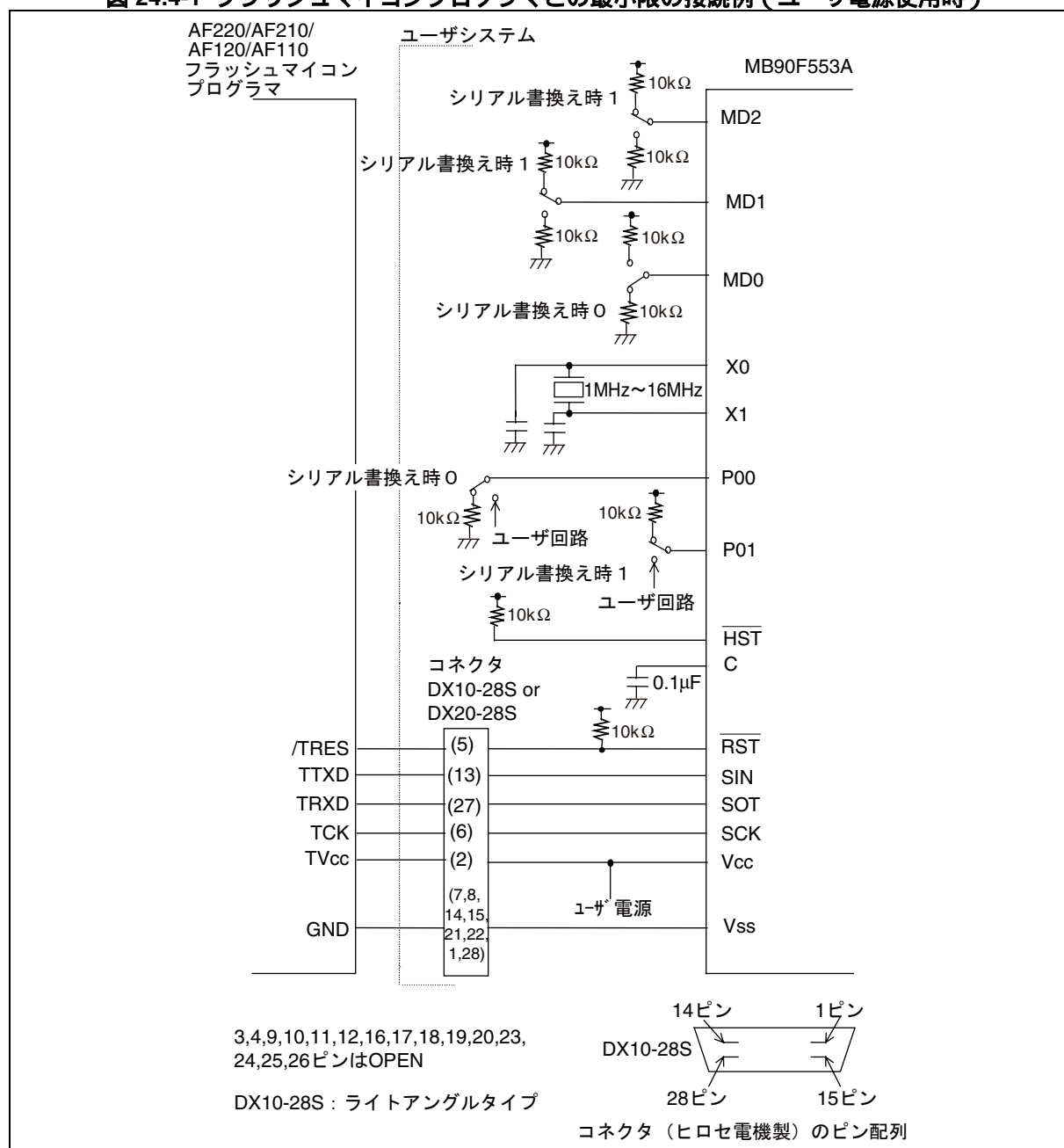
24.4 フラッシュマイコンプログラマとの最小限の接続例 (ユーザ電源使用時)

図 24.4-1 に、マイコンの電源電圧をユーザ電源より供給する場合の、フラッシュマイコンプログラマとの最小限の接続例を示します。

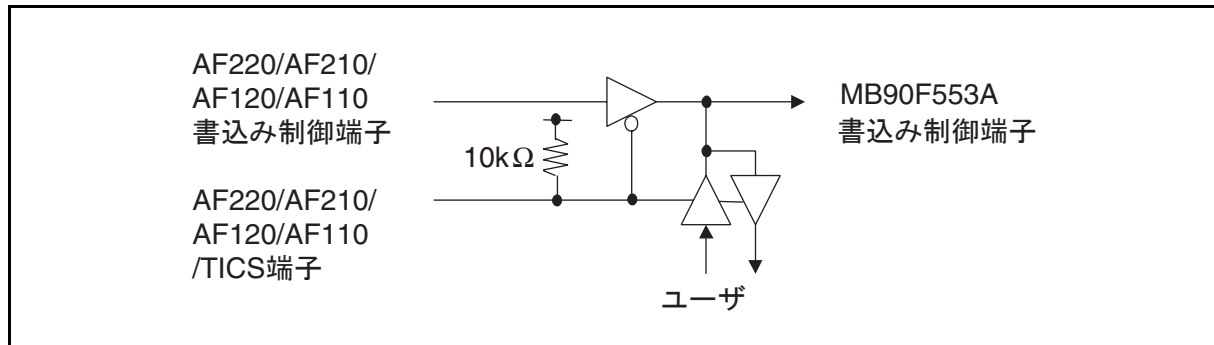
■ フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用時）

フラッシュメモリ書込み時に、各端子を下記のように設定して頂ければ、MD2, MD1, MD0, P00 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。

図 24.4-1 フラッシュマイコンプログラムの最小限の接続例（ユーザ電源使用時）



- SIN, SOT, SCK端子をユーザシステムでも使用する場合には, 下図の制御回路が必要となります (フラッシュマイコンプログラムの /TICS 信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110 との接続はユーザ電源が OFF の状態で行ってください。

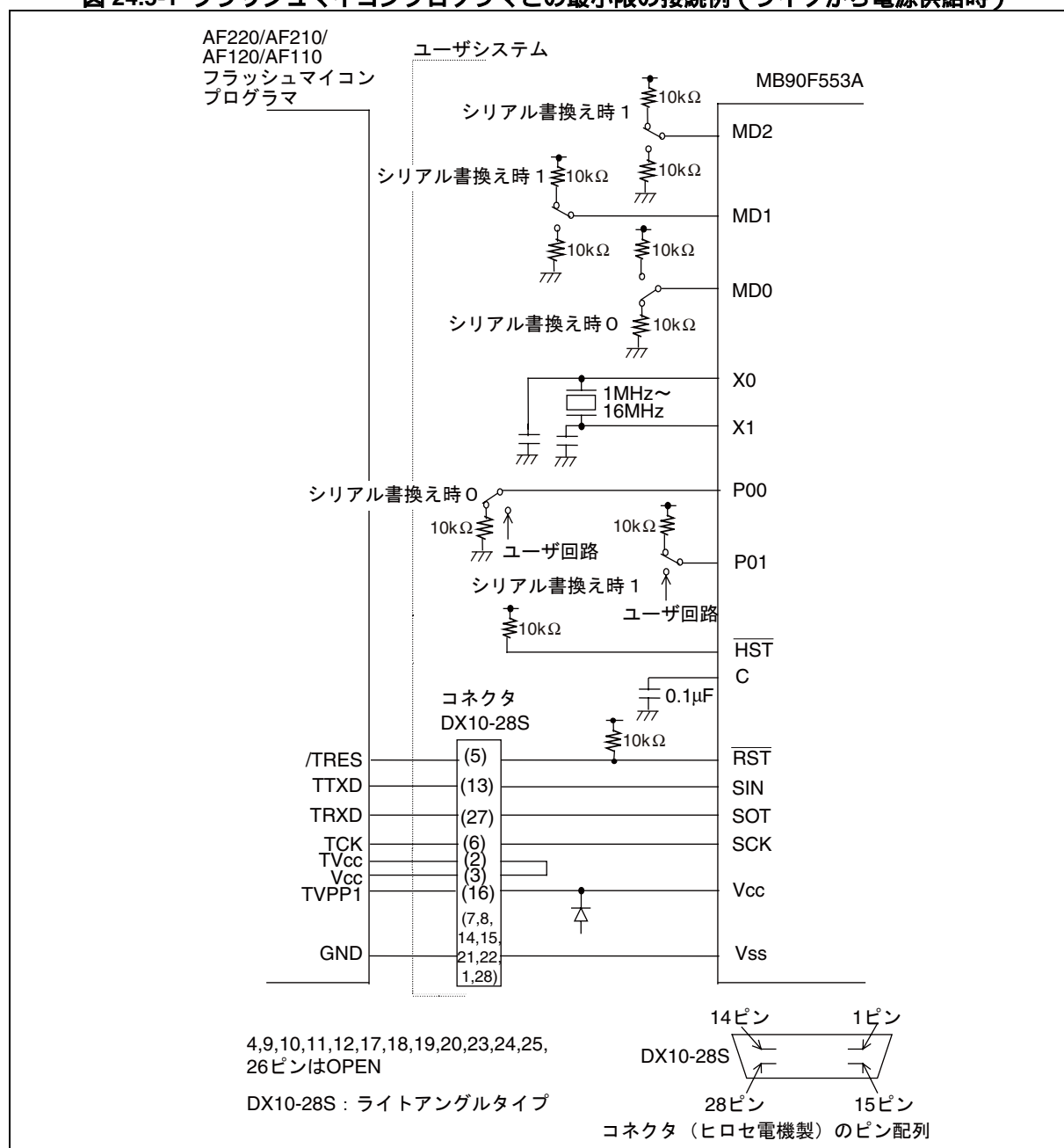
24.5 フラッシュマイコンプログラムの最小限の接続例 (ライターから電源供給時)

図 24.5-1 に、マイコンの電源電圧をライター電源より供給する場合の、フラッシュマイコンプログラムの最小限の接続例を示します。

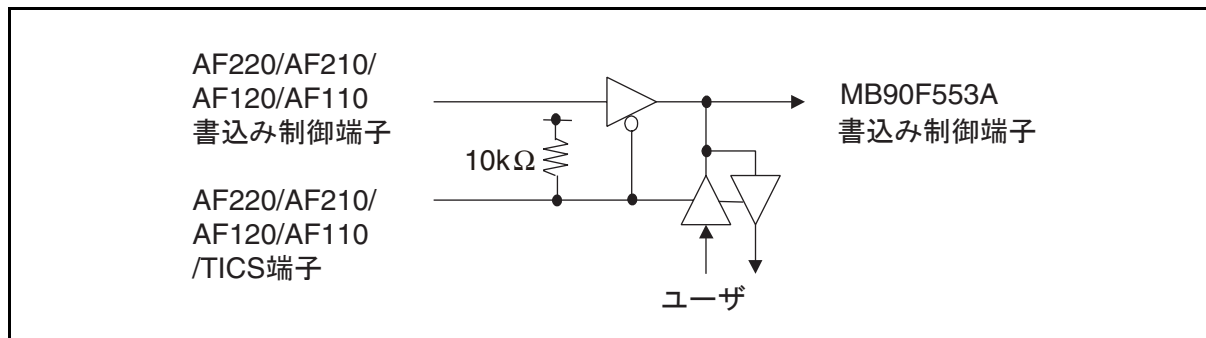
■ フラッシュマイコンプログラムの最小限の接続例 (ライターから電源供給時)

フラッシュメモリ書込み時に各端子を下記のように設定して頂ければ、MD2, MD1, MD0, P00 とフラッシュマイコンプログラムの接続は必要ありません。

図 24.5-1 フラッシュマイコンプログラムの最小限の接続例 (ライターから電源供給時)



- SIN, SOT, SCK端子をユーザシステムでも使用する場合には, 下図の制御回路が必要となります (フラッシュマイコンプログラムの /TICS 信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110 との接続はユーザ電源が OFF の状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給するときはユーザ電源と短絡しないでください。

付録では、I/O マップ、命令および OTPROM 書込みについて示します。

付録 A I/O マップ

付録 B 命令

付録 C OTPROM 書込み

付録 A I/O マップ

本マイクロコントローラの各リソースのレジスタは、以下のようにアドレスが割当てられています。

■ I/O マップ

表 A-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	--11111 _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
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 ~ 0F _H	使用禁止				
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	使用禁止				
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
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	---00000 _B
1B _H	ポート 4 出力端子レジスタ	ODR4	R/W	ポート 4	00000000 _B
1C _H	ポート 0 抵抗レジスタ	RDR0	R/W	ポート 0	00000000 _B
1D _H	ポート 1 抵抗レジスタ	RDR1	R/W	ポート 1	00000000 _B
1E _H	使用禁止				

表 A-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
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	00000000 _B
21 _H	シリアルコントロールレジスタ	SCR	R/W		00000100 _B
22 _H	シリアルインプットレジスタ /シリアル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/W		00001-00 _B
24 _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/W	I/O 拡張 シリアルインタ フェース 0	----0000 _B
25 _H	シリアルモードコントロールステータス レジスタ 0		R/W!		00000010 _B
26 _H	シリアルデータレジスタ 0	SDR0	R/W		XXXXXXXX _B
27 _H	クロック分周制御レジスタ	CDCR	R/W	通信 プリスケラ	0---1111 _B
28 _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/W	I/O 拡張シリア ルインタ フェース 1	----0000 _B
29 _H	シリアルモードコントロールステータス レジスタ 1		R/W!		00000010 _B
2A _H	シリアルデータレジスタ 1	SDR1	R/W		XXXXXXXX _B
2B _H	使用禁止				
2C _H	I ² C バスステータスレジスタ 0	IBSR0	R	I ² C インタ フェース 0	00000000 _B
2D _H	I ² C バスコントロールレジスタ 0	IBCR0	R/W		00000000 _B
2E _H	I ² C バスクロック選択レジスタ 0	ICCR0	R/W		--0XXXXX _B
2F _H	I ² C バスアドレスレジスタ 0	IADR0	R/W		-XXXXXXXX _B
30 _H	I ² C バスデータレジスタ 0	IDAR0	R/W		XXXXXXXX _B
31 _H	使用禁止				
32 _H	I ² C バスステータスレジスタ 1	IBSR1	R	I ² C インタ フェース 1	00000000 _B
33 _H	I ² C バスコントロールレジスタ 1	IBCR1	R/W		00000000 _B
34 _H	I ² C バスクロック選択レジスタ 1	ICCR1	R/W		--0XXXXX _B
35 _H	I ² C バスアドレスレジスタ 1	IADR1	R/W		-XXXXXXXX _B
36 _H	I ² C バスデータレジスタ 1	IDAR1	R/W		XXXXXXXX _B
37 _H	I ² C バスポート選択レジスタ	ISEL	R/W		-----0 _B
38 _H	割込み /DTP 許可レジスタ	ENIR	R/W	DTP/ 外部割込み	00000000 _B
39 _H	割込み /DTP 要因レジスタ	EIRR	R/W		XXXXXXXX _B
3A _H	要求レベル設定レジスタ	ELVR	R/W		00000000 _B
3B _H					00000000 _B

表 A-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
3C _H	コントロールステータスレジスタ	ADCS0	R/W	A/D コンバータ	00000000 _B
3D _H		ADCS1	R/W!		00000000 _B
3E _H	データレジスタ	ADCR0	R/W!		XXXXXXXX _B
3F _H		ADCR1	R/W!		00001-XX _B
40 _H	リロードレジスタ L(ch.0)	PRLLO	R/W	8/16bit PPG0/1	XXXXXXXX _B
41 _H	リロードレジスタ H(ch.0)	PRLH0	R/W		XXXXXXXX _B
42 _H	リロードレジスタ L(ch.1)	PRLLO	R/W		XXXXXXXX _B
43 _H	リロードレジスタ H(ch.1)	PRLH1	R/W		XXXXXXXX _B
44 _H	PPG0 動作モード制御レジスタ	PPGC0	R/W		0-000--1 _B
45 _H	PPG1 動作モード制御レジスタ	PPGC1	R/W		0-000001 _B
46 _H	PPG0, PPG1 出力制御レジスタ	PPGE1	R/W		00000000 _B
47 _H	使用禁止				
48 _H	リロードレジスタ L(ch.2)	PRLLO	R/W	8/16bit PPG2/3	XXXXXXXX _B
49 _H	リロードレジスタ H(ch.2)	PRLH2	R/W		XXXXXXXX _B
4A _H	リロードレジスタ L(ch.3)	PRLLO	R/W		XXXXXXXX _B
4B _H	リロードレジスタ H(ch.3)	PRLH3	R/W		XXXXXXXX _B
4C _H	PPG2 動作モード制御レジスタ	PPGC2	R/W		0-000--1 _B
4D _H	PPG3 動作モード制御レジスタ	PPGC3	R/W		0-000001 _B
4E _H	PPG2, PPG3 出力制御レジスタ	PPGE2	R/W		00000000 _B
4F _H	使用禁止				
50 _H	リロードレジスタ L(ch.4)	PRLLO	R/W	8/16 ビット PPG4/5	XXXXXXXX _B
51 _H	リロードレジスタ H(ch.4)	PRLH4	R/W		XXXXXXXX _B
52 _H	リロードレジスタ L(ch.5)	PRLLO	R/W		XXXXXXXX _B
53 _H	リロードレジスタ H(ch.5)	PRLH5	R/W		XXXXXXXX _B
54 _H	PPG4 動作モード制御レジスタ	PPGC4	R/W		0-000--1 _B
55 _H	PPG5 動作モード制御レジスタ	PPGC5	R/W		0-000001 _B
56 _H	PPG4, PPG5 出力制御レジスタ	PPGE3	R/W		00000000 _B
57 _H	使用禁止				
58 _H	クロック出力許可レジスタ	CLKR	R/W	クロックモニタ 機能	----0000 _B
59 _H	使用禁止				
5A _H	コントロールステータスレジスタ 0	TMCSR0	R/W	16bit リロード タイマ 0	00000000 _B
5B _H					----0000 _B
5C _H	16bit タイマレジスタ 0/	TMR0/ TMRLR0	R/W		XXXXXXXX _B
5D _H	16bit リロードレジスタ 0				XXXXXXXX _B

表 A-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
5E _H	コントロールステータスレジスタ 1	TMCSR1	R/W	16bit リロード タイマ 1	00000000 _B
5F _H					----0000 _B
60 _H	16bit タイマレジスタ 1	TMR1	R/W		XXXXXXXX _B
61 _H	16bit リロードレジスタ 1	TMRLR1			XXXXXXXX _B
62 _H	インプットキャプチャレジスタ ch.0 下位	IPCP0	R	16bit 入出力タイマ インプットキャ プチャ (ch.0-3)	XXXXXXXX _B
63 _H	インプットキャプチャレジスタ ch.0 上位				XXXXXXXX _B
64 _H	インプットキャプチャレジスタ ch.1 下位	IPCP1	R		XXXXXXXX _B
65 _H	インプットキャプチャレジスタ ch.1 上位				XXXXXXXX _B
66 _H	インプットキャプチャレジスタ ch.2 下位	IPCP2	R		XXXXXXXX _B
67 _H	インプットキャプチャレジスタ ch.2 上位				XXXXXXXX _B
68 _H	インプットキャプチャレジスタ ch.3 下位	IPCP3	R		XXXXXXXX _B
69 _H	インプットキャプチャレジスタ ch.3 上位				XXXXXXXX _B
6A _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
6B _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	ICS23	R/W		00000000 _B
6C _H	タイマデータレジスタ下位	TCDT	R/W	16bit 入出力タイマ フリーラン タイマ	00000000 _B
6D _H	タイマデータレジスタ上位		R/W		00000000 _B
6E _H	タイマコントロールステータスレジスタ	TCCS	R/W		00000000 _B
6F _H	ROM ミラー機能選択レジスタ	ROMM	W	ROM ミラー 機能	-----1 _B
70 _H	コンペアレジスタ ch.0 下位	OCCP0	R/W	16bit 入出力タイマ アウトプット コンペア (ch.0-3)	XXXXXXXX _B
71 _H	コンペアレジスタ ch.0 上位				XXXXXXXX _B
72 _H	コンペアレジスタ ch.1 下位	OCCP1	R/W		XXXXXXXX _B
73 _H	コンペアレジスタ ch.1 上位				XXXXXXXX _B
74 _H	コンペアレジスタ ch.2 下位	OCCP2	R/W		XXXXXXXX _B
75 _H	コンペアレジスタ ch.2 上位				XXXXXXXX _B
76 _H	コンペアレジスタ ch.3 下位	OCCP3	R/W		XXXXXXXX _B
77 _H	コンペアレジスタ ch.3 上位				XXXXXXXX _B
78 _H	コンペアコントロールステータス レジスタ ch.0	OCS0	R/W		0000--00 _B
79 _H	コンペアコントロールステータス レジスタ ch.1	OCS1	R/W		---00000 _B
7A _H	コンペアコントロールステータス レジスタ ch.2	OCS2	R/W	0000--00 _B	
7B _H	コンペアコントロールステータス レジスタ ch.3	OCS3	R/W	---00000 _B	
7C _H ~ 9D _H	使用禁止				

表 A-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
9E _H	プログラムアドレス検出コントロール 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	遅延割込み	-----0 _B
A0 _H	低消費電力モードレジスタ	LPMCR	R/W!	低消費電力制御 回路	00011000 _B
A1 _H	クロック選択レジスタ	CKSCR	R/W!		11111100 _B
A2 _H ~ A4 _H	使用禁止				
A5 _H	自動レディ機能選択レジスタ	ARSR	W	外部バス端子 制御回路	0011--00 _B
A6 _H	外部アドレス出力制御レジスタ	HACR	W		00000000 _B
A7 _H	バス制御信号選択レジスタ	ECSR	W		0000000- _B
A8 _H	ウォッチドッグ制御レジスタ	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	R/W!	タイムベース タイマ	1--00100 _B
AA _H ~ AD _H	使用禁止				
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	R/W	フラッシュ メモリインタ フェイス回路	00000--0 _B
AF _H	使用禁止				
B0 _H	割込み制御レジスタ 00	ICR00	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	R/W!		00000111 _B
B2 _H	割込み制御レジスタ 02	ICR02	R/W!		00000111 _B
B3 _H	割込み制御レジスタ 03	ICR03	R/W!		00000111 _B
B4 _H	割込み制御レジスタ 04	ICR04	R/W!		00000111 _B
B5 _H	割込み制御レジスタ 05	ICR05	R/W!		00000111 _B
B6 _H	割込み制御レジスタ 06	ICR06	R/W!		00000111 _B
B7 _H	割込み制御レジスタ 07	ICR07	R/W!		00000111 _B
B8 _H	割込み制御レジスタ 08	ICR08	R/W!		00000111 _B
B9 _H	割込み制御レジスタ 09	ICR09	R/W!		00000111 _B
BA _H	割込み制御レジスタ 10	ICR10	R/W!		00000111 _B
BB _H	割込み制御レジスタ 11	ICR11	R/W!		00000111 _B
BC _H	割込み制御レジスタ 12	ICR12	R/W!		00000111 _B
BD _H	割込み制御レジスタ 13	ICR13	R/W!		00000111 _B
BE _H	割込み制御レジスタ 14	ICR14	R/W!		00000111 _B
BF _H	割込み制御レジスタ 15	ICR15	R/W!		00000111 _B
C0 _H ~ FF _H	外部領域				
100 _H ~ # _H	RAM 領域				
# _H ~ 1FE _H	予約領域				

表 A-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	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 1		R/W		XXXXXXXX _B
1FF2 _H	プログラムアドレス検出レジスタ 2		R/W		XXXXXXXX _B
1FF3 _H	プログラムアドレス検出レジスタ 3	PADR1	R/W		XXXXXXXX _B
1FF4 _H	プログラムアドレス検出レジスタ 4		R/W		XXXXXXXX _B
1FF5 _H	プログラムアドレス検出レジスタ 5		R/W		XXXXXXXX _B
1FF6 _H ~ 1FFF _H	予約領域				
<div>• 初期値について "0": 初期値 "0"。/"1": 初期値 "1"。/"X": 初期値不定。/"-": 初期値不定（なし）。</div> <div>• 00FF_H 以下のアドレスは予約領域です。</div> <div>• RAM 領域と予約領域の境界 #_H は品種ごとに変わります。</div> <div>(注意事項)</div> <div>書込み可能なビットに関しては、リセットにより初期化される値が初期値として記述されています。</div> <div>読出し時の値ではありませんので注意してください。</div> <div>また、LPMCR/CKSCR/WDTC では、リセットの種類により、初期化される場合と、初期化されない場合がありますが、初期化される場合の初期値が記述されています（詳細は、「4.5 リセット入力によって初期化されないレジスタ」を参照してください）。</div> <div>アクセスの欄で R/W! となっているレジスタは、読出しのみ可または書込みのみ可のビットが存在することを示しています。</div> <div>アクセスの欄で R/W!,R/W* または W となっているレジスタをリードモディファイライト系の命令（ビットセット等の命令）でアクセスすると、命令で注目しているビットは所定の値になりますが、それ以外のビットで書込みのみ可のビットがあると誤動作を引き起こしますので、これらの命令でアクセスしないでください。</div>					

付録 B 命令

F²MC-16LX に使用している命令について説明します。

- B.1 命令の種類
- B.2 アドレッシング
- B.3 直接アドレッシング
- B.4 間接アドレッシング
- B.5 実行サイクル数
- B.6 実効アドレスフィールド
- B.7 命令一覧表の読み方
- B.8 F²MC-16LX 命令一覧表
- B.9 命令マップ

B.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

B.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 B.2-1 に示します。

表 B.2-1 実効アドレスフィールド

コード	表記			アドレス形式	デフォルトバンク
00	R0	RW0	RL0	レジスタ直接 ea は左から順に バイト ワード ロングワード の型に対応する。	なし
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	DTB
09	@RW1				DTB
0A	@RW2				ADB
0B	@RW3				SPB
0C	@RW0+			ポストインクリメント付 レジスタ間接	DTB
0D	@RW1+				DTB
0E	@RW2+				ADB
0F	@RW3+				SPB
10	@RW0+disp8			8 ビットディスプレースメント付 レジスタ間接	DTB
11	@RW1+disp8				DTB
12	@RW2+disp8				ADB
13	@RW3+disp8				SPB
14	@RW4+disp8				DTB
15	@RW5+disp8				DTB
16	@RW6+disp8				ADB
17	@RW7+disp8				SPB
18	@RW0+disp16			16 ビットディスプレースメント付 レジスタ間接	DTB
19	@RW1+disp16				DTB
1A	@RW2+disp16				ADB
1B	@RW3+disp16				SPB
1C	@RW0+RW7			インデックス付レジスタ間接	DTB
1D	@RW1+RW7			インデックス付レジスタ間接	DTB
1E	@PC+disp16			16 ビットディスプレースメント付 PC 間接	PCB
1F	addr16			直接アドレス	DTB

B.3 直接アドレッシング

直接アドレッシングでは、オペランド値、レジスタおよびアドレスを直接指定します。

■ 直接アドレッシング

● 即値 (#imm)

オペランドの値を直接指定します。(#imm4/#imm8/#imm16/#imm32)。

図 B.3-1 に例を示します。

図 B.3-1 即値 (#imm) 例

MOVW A, #01212H (A にオペランドの値を格納する命令)		
実行前	A	2 2 3 3 : 4 4 5 5
実行後	A	4 4 5 5 : 1 2 1 2 (命令によっては AL AH に転送が行われる)

● レジスタ直接

オペランドとして、直接レジスタを指定します。指定できるレジスタを表 B.3-1 に示します。

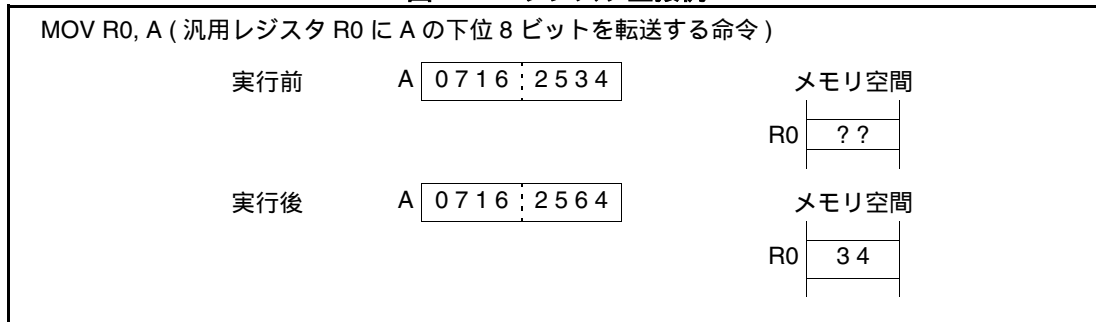
表 B.3-1 レジスタ直接

汎用レジスタ	バイト	R0, R1, R2, R3, R4, R5, R6, R7
	ワード	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
	ロングワード	RL0, RL1, RL2, RL3
専用レジスタ	アキュムレータ	A, AL
	ポインタ	SP *
	バンク	PCB, DTB, USB, SSB, ADB
	ページ	DPR
	制御	PS, CCR, RP, ILM

*: SP は、コンディションコードレジスタ (CCR) 中の S フラグビットの値に応じてユーザスタックポインタ (USP) またはシステムスタックポインタ (SSP) のどちらか一方が選択され、使用されます。分岐系の命令ではプログラムカウンタ (PC) は命令のオペランドには記述されることなく指定されます。

図 B.3-2 に例を示します。

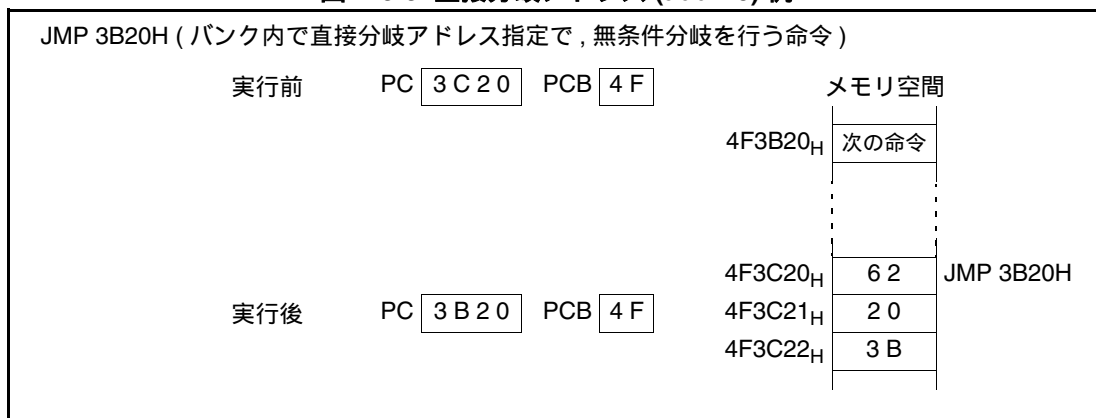
図 B.3-2 レジスタ直接例



● 直接分岐アドレス (addr16)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのデータ長は 16 ビットで、論理空間内での分岐先を示します。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されます。図 B.3-3 に例を示します。

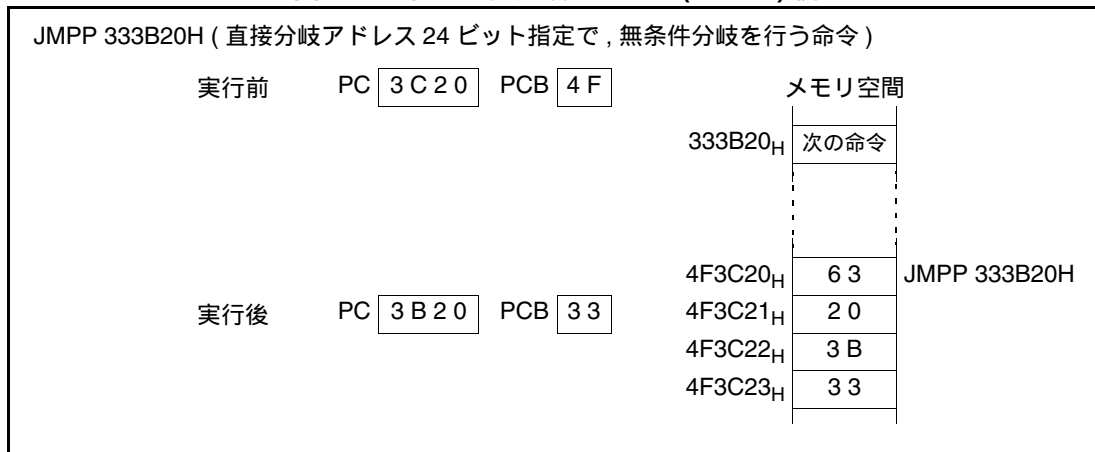
図 B.3-3 直接分岐アドレス (addr16) 例



● 物理直接分岐アドレス (addr24)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのデータ長は 24 ビットです。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。図 B.3-4 に例を示します。

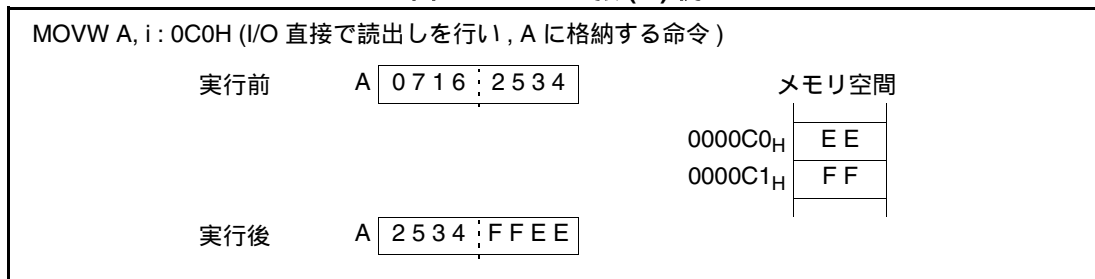
図 B.3-4 物理直接分岐アドレス (addr24) 例



● I/O 直接 (io)

オペランドのメモリアドレスを 8 ビットのディスプレ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 B.3-5 に例を示します。

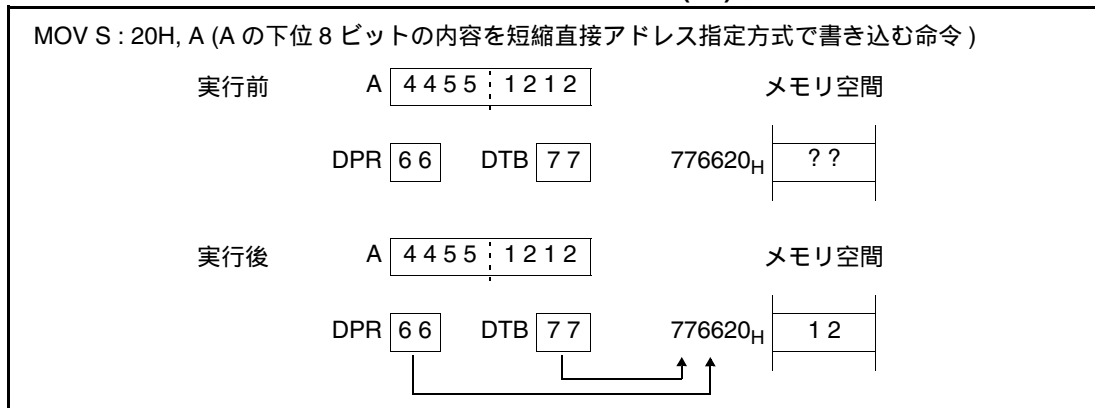
図 B.3-5 I/O 直接 (io) 例



● 短縮直接アドレス (dir)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit15 ~ bit8 はダイレクトページレジスタ (DPR) により指定されます。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) により指定されます。図 B.3-6 に例を示します。

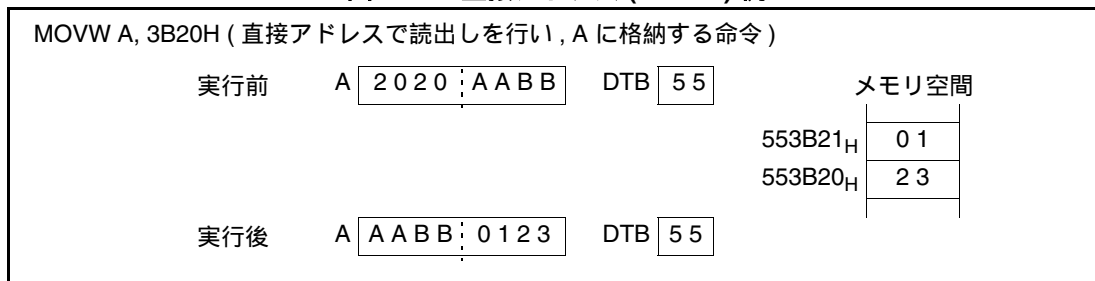
図 B.3-6 短縮直接アドレス (dir) 例



● 直接アドレス (addr16)

オペランドで、メモリアドレス下位 16 ビットを直接指定します。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。直接アドレスのアドレッシングに対しては、アクセス空間指定用のプリフィックス命令は無効です。図 B.3-7 に例を示します。

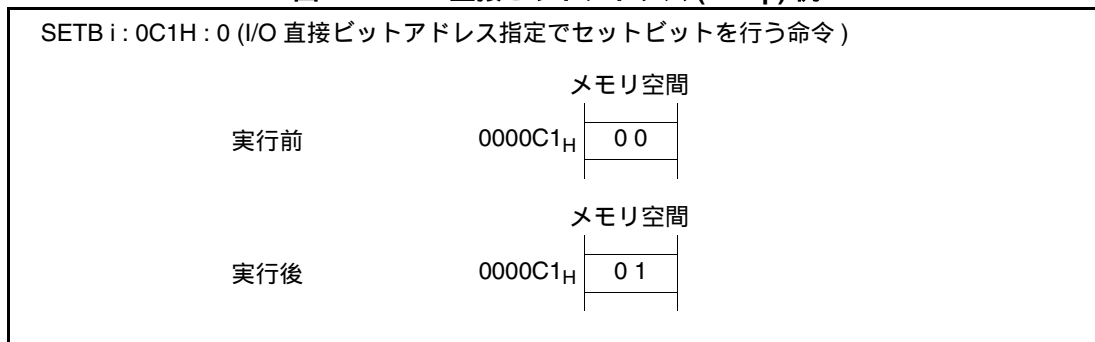
図 B.3-7 直接アドレス (addr16) 例



● I/O 直接ビットアドレス (io: bp)

物理アドレス "000000_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 B.3-8 に例を示します。

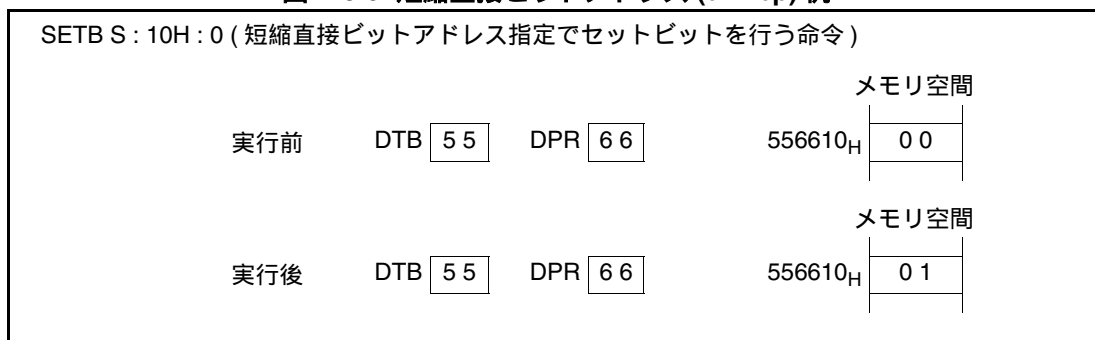
図 B.3-8 I/O 直接ビットアドレス (io: bp) 例



● 短縮直接ビットアドレス (dir: bp)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit15 ~ bit8 はダイレクトページレジスタ (DPR) で指定されます。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され, 数字の大きい方が最上位ビット, 小さい方が最下位ビットとなります。図 B.3-9 に例を示します。

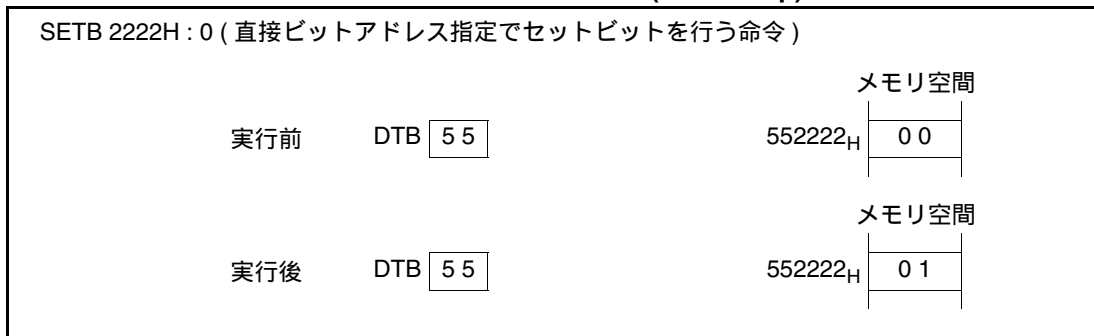
図 B.3-9 短縮直接ビットアドレス (dir: bp) 例



● 直接ビットアドレス (addr16: bp)

64K バイト内の任意のビットに対し直接指定します。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され、数字の大きい方が最上位ビット、小さい方が最下位ビットとなります。図 B.3-10 に例を示します。

図 B.3-10 直接ビットアドレス (addr16: bp) 例



● ベクタアドレス (#vct)

分岐先のアドレスは指定されたベクタの内容となります。ベクタ番号のデータ長には 4 ビットと 8 ビットの 2 種類があります。サブルーチンコール命令、ソフトウェア割り込み命令に使用します。図 B.3-11 に例を示します。

図 B.3-11 ベクタアドレス (#vct) 例



表 B.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE _H	XXFFFF _H
CALLV #1	XXFFFC _H	XXFFFD _H
CALLV #2	XXFFFA _H	XXFFFB _H
CALLV #3	XXFFF8 _H	XXFFF9 _H
CALLV #4	XXFFF6 _H	XXFFF7 _H
CALLV #5	XXFFF4 _H	XXFFF5 _H
CALLV #6	XXFFF2 _H	XXFFF3 _H
CALLV #7	XXFFF0 _H	XXFFF1 _H
CALLV #8	XXFFEE _H	XXFFEF _H
CALLV #9	XXFFEC _H	XXFFED _H
CALLV #10	XXFFEA _H	XXFFEB _H
CALLV #11	XXFFE8 _H	XXFFE9 _H
CALLV #12	XXFFE6 _H	XXFFE7 _H
CALLV #13	XXFFE4 _H	XXFFE5 _H
CALLV #14	XXFFE2 _H	XXFFE3 _H
CALLV #15	XXFFE0 _H	XXFFE1 _H

(注意事項) 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 B.3-2 を参照してください)

B.4 間接アドレッシング

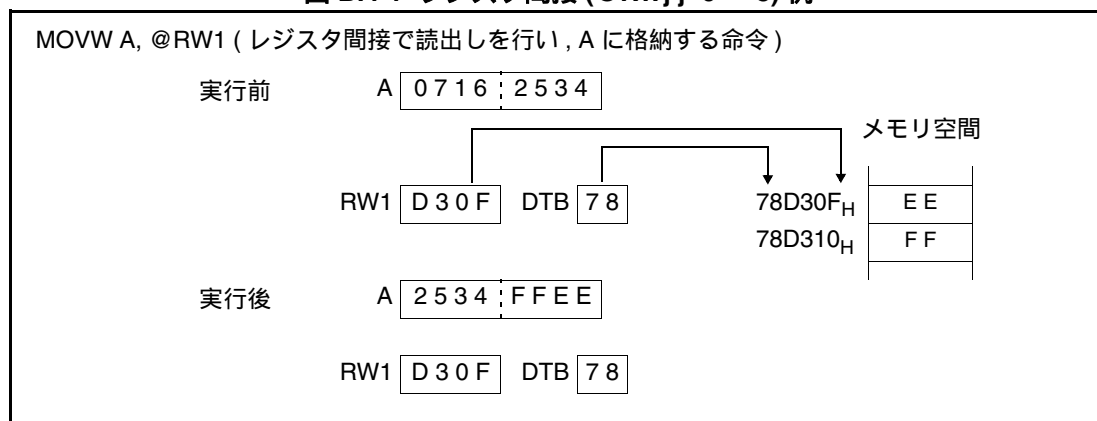
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

■ 間接アドレッシング

● レジスタ間接 (@RWj j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-1 に例を示します。

図 B.4-1 レジスタ間接 (@RWj j=0 ~ 3) 例



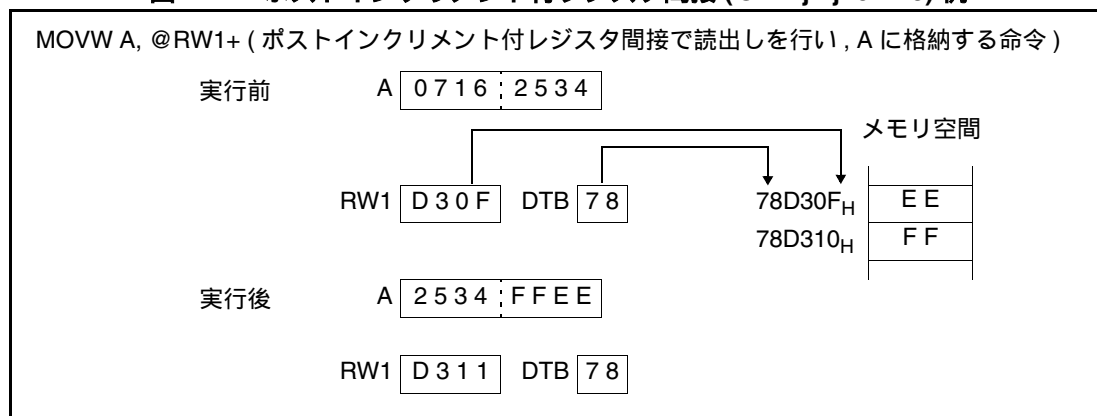
● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。オペランド操作後RWjはオペランドデータ長 (バイトの場合は1, ワードの場合は2, ロングワードの場合は4) 分だけ加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。

ポストインクリメントした結果がインクリメント指定したレジスタ自身のアドレスだった場合は、参照される値はインクリメントした値となり、命令が書込みだった場合は命令による書込みが優先されるので、インクリメントするはずだったレジスタは書込みデータとなります。

図 B.4-2 に例を示します。

図 B.4-2 ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3) 例

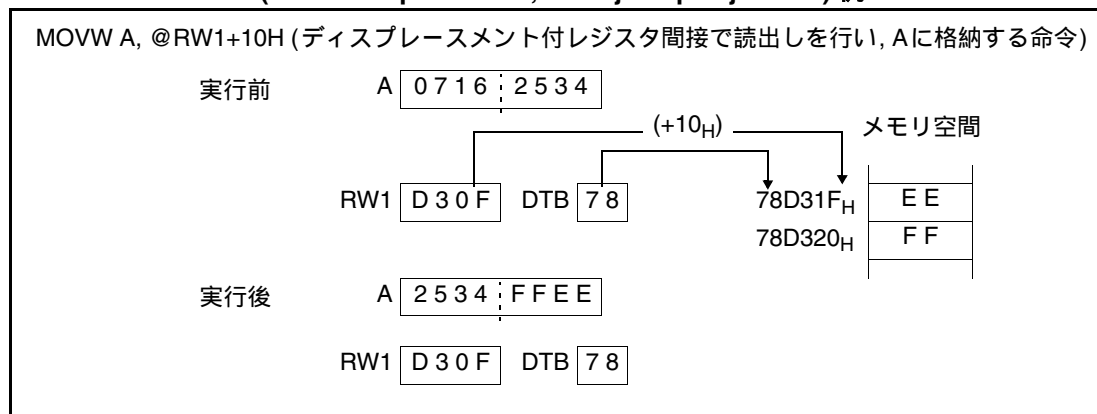


● ディスプレースメント付レジスタ間接

(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)

汎用レジスタ RWj の内容にディスプレースメントを加算したものをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは、バイトとワードの2種類があり、符号付数値として加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1, RW4, RW5 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3, RW7 を用いた場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2, RW6 を用いた場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-3 に例を示します。

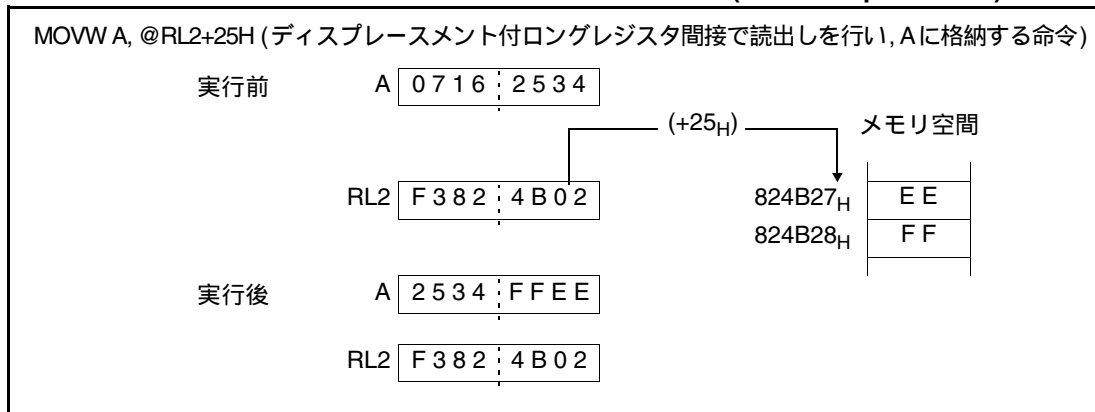
図 B.4-3 ディスプレースメント付レジスタ間接
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● ディスプレースメント付ロングレジスタ間接 (RLi+disp8 i=0 ~ 3)

汎用レジスタ RLi の内容にディスプレースメントを加算した結果の下位 24 ビットをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは 8 ビットで符号付数値として加算されます。図 B.4-4 に例を示します。

図 B.4-4 ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3) 例



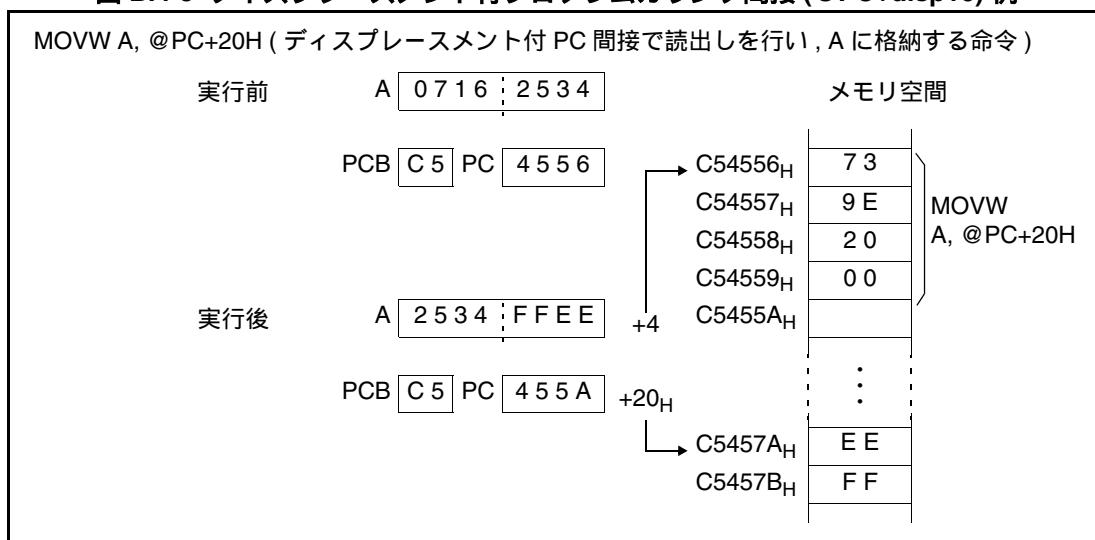
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(命令のアドレス+4+disp16) で示されるアドレスのメモリをアクセスするアドレッシングです。ディスプレースメントはワード長です。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) により指定されます。次に示す命令のオペランドアドレスは, (次の命令のアドレス +disp16) とは, みなしませんでしたので注意してください。

- DBNZ eam, rel
- CBNE eam, #imm8, rel
- MOV eam, #imm8
- DWBNZ eam, rel
- CWBNE eam, #imm16, rel
- MOVW eam, #imm16

図 B.4-5 に例を示します。

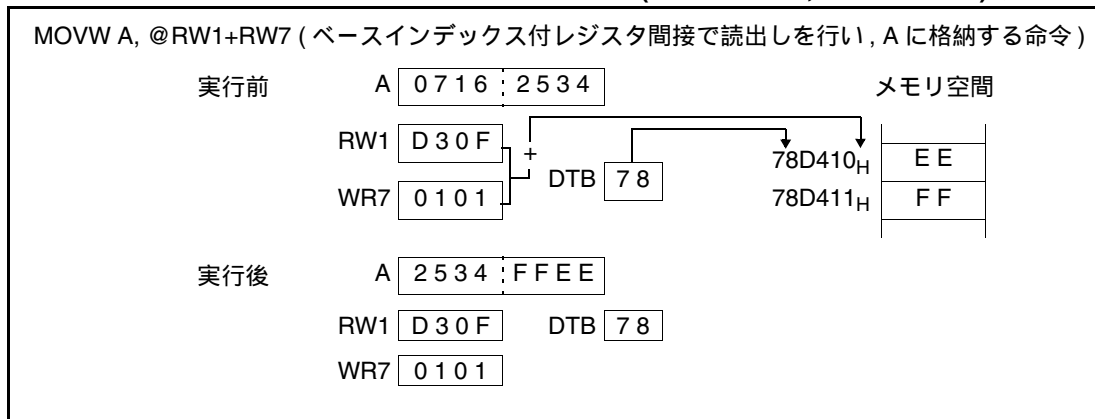
図 B.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例



● ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7)

汎用レジスタ RW7 の内容に, RW0 あるいは RW1 を加算したものをアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で示されます。図 B.4-6 に例を示します。

図 B.4-6 ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7) 例

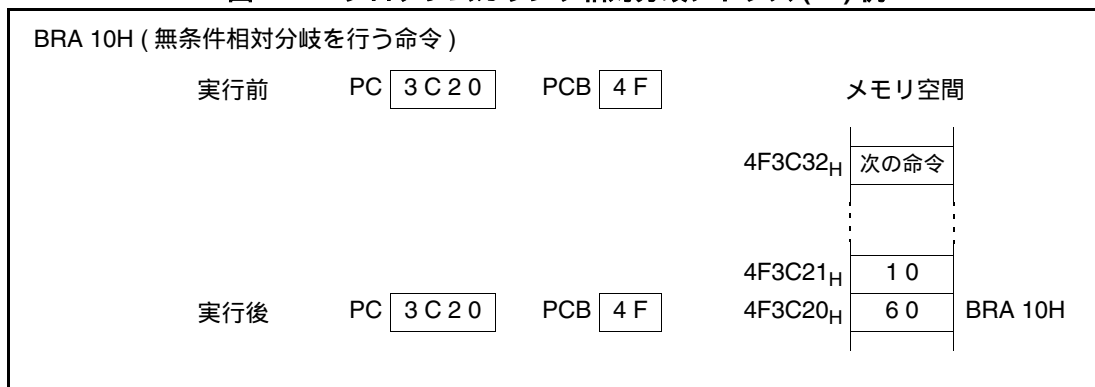


● プログラムカウンタ相対分岐アドレス (rel)

分岐先のアドレスはプログラムカウンタ (PC) の値と 8 ビットのディスプレースメントを加算した値となります。加算の結果が 16 ビットを超えた場合は, バンクレジスタのインクリメントまたはデクレメントをせずに, 超えた分は無視されますので, 64K バイトのバンク内で閉じたアドレスとなります。プログラムカウンタ相対分岐アドレスによるアドレッシングは, 無条件 / 条件分岐命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で示されます。

図 B.4-7 に例を示します。

図 B.4-7 プログラムカウンタ相対分岐アドレス (rel) 例



● レジスタリスト (rlst)

スタックに対するプッシュ / ポップの対象となるレジスタを指定します。図 B.4-8 にレジスタリストの構成、図 B.4-9 に例を示します。

図 B.4-8 レジスタリストの構成

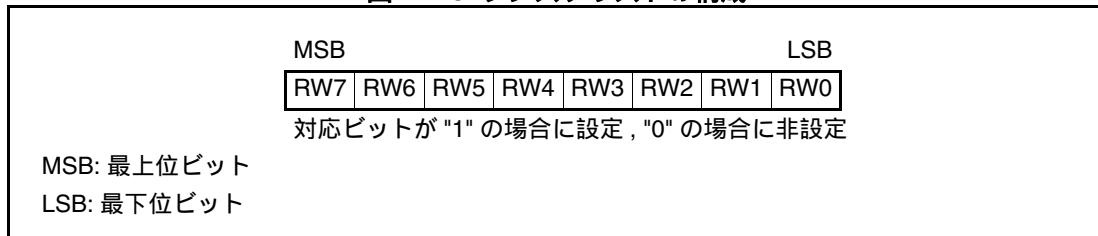
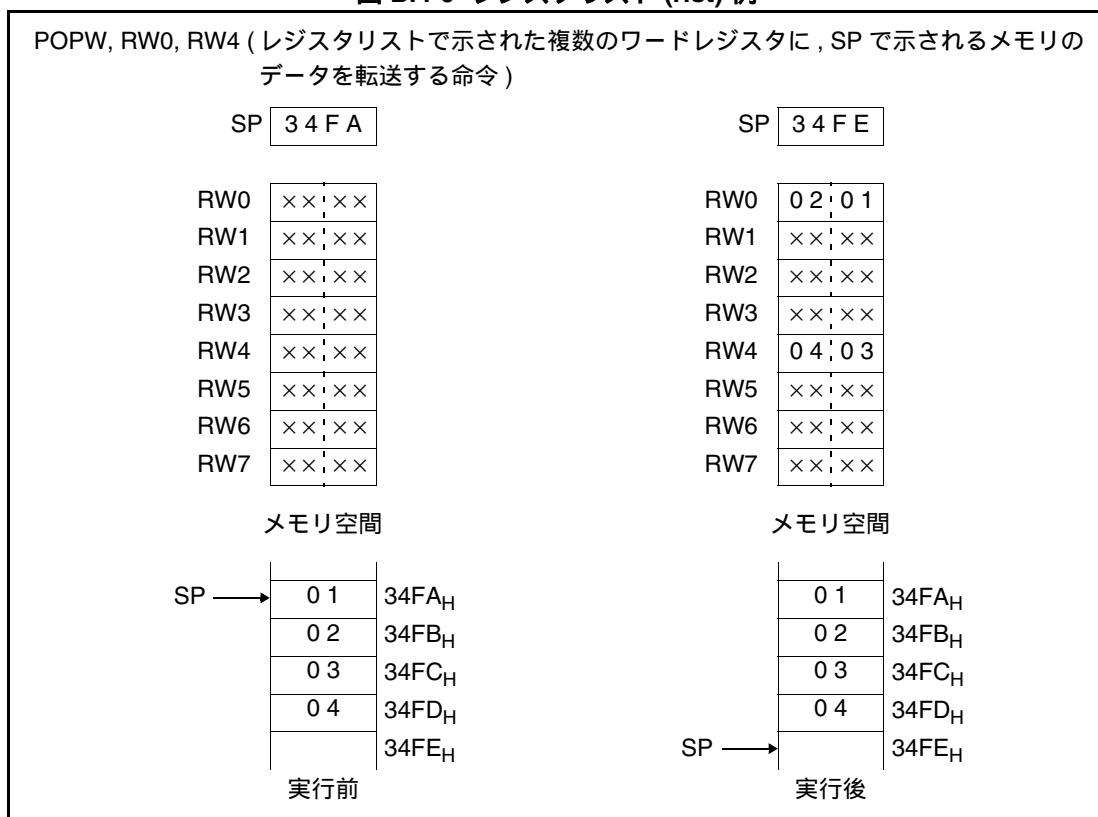


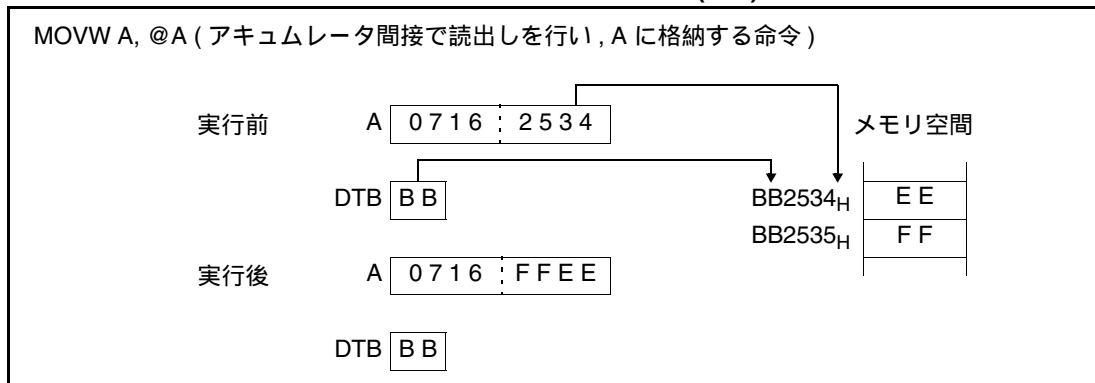
図 B.4-9 レジスタリスト (rlst) 例



● アキュムレータ間接 (@A)

アキュムレータの下位バイト (AL) の内容 (16 ビット) で示されるアドレスのメモリをアクセスするアドレス方式です。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) によりニーモニックで指定されます。図 B.4-10 に例を示します。

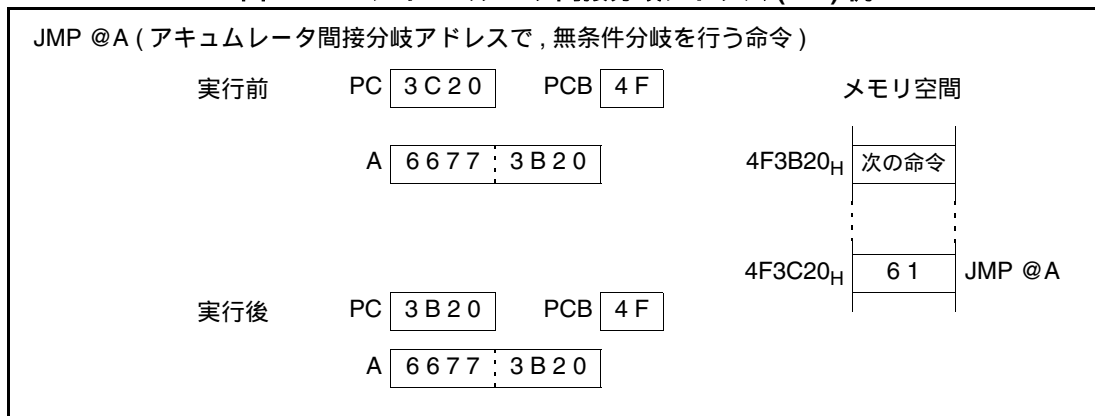
図 B.4-10 アキュムレータ間接 (@A) 例



● アキュムレータ間接分岐アドレス (@A)

分岐先のアドレスは, アキュムレータの下位バイト (AL) の内容 (16 ビット) となります。バンク空間内での分岐先を示し, アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されますが, JCTX (Jump Context) 命令の場合は, アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。アキュムレータ間接分岐アドレスによるアドレッシングは, 無条件分岐命令に使用します。図 B.4-11 に例を示します。

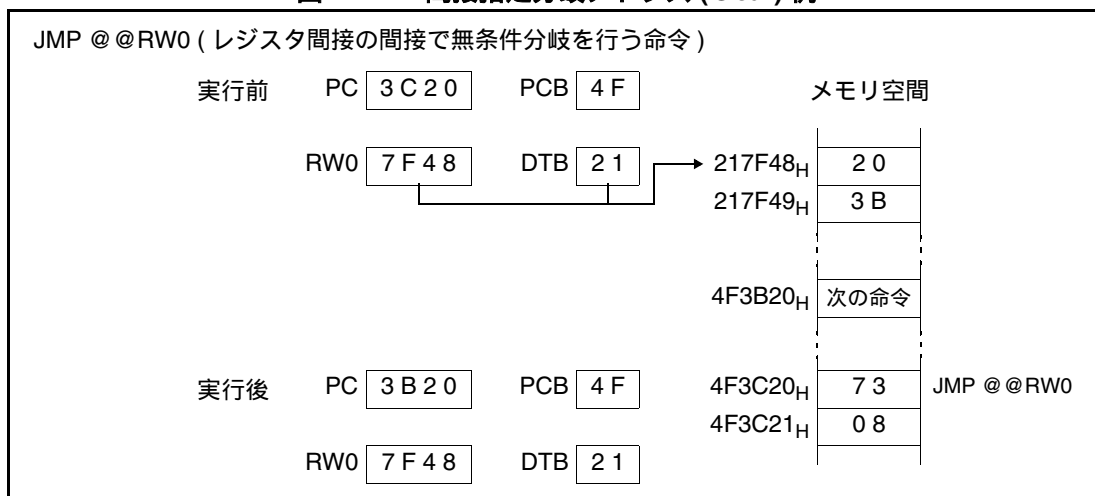
図 B.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 間接指定分岐アドレス (@ear)

ear で示される番地のワードデータが分岐先アドレスとなります。図 B.4-12 に例を示します。

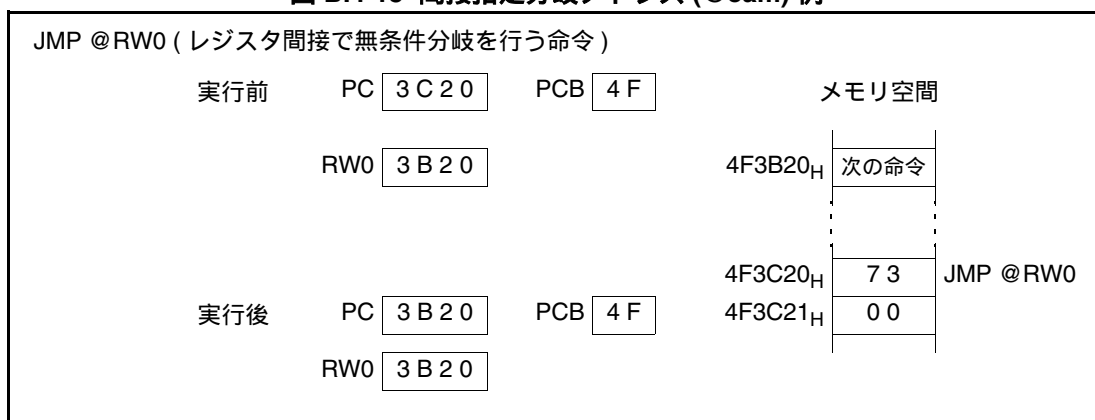
図 B.4-12 間接指定分岐アドレス (@ear) 例



● 間接指定分岐アドレス (@eam)

eam で示される番地のワードデータが分岐先アドレスとなります。図 B.4-13 に例を示します。

図 B.4-13 間接指定分岐アドレス (@eam) 例



B.5 実行サイクル数

命令の実行に要するサイクル数 (実行サイクル数) は、各命令の「サイクル数」の値と、条件で決まる「補正值」の値およびプログラムフェッチの「サイクル数」の値を加算することで得られます。

■ 実行サイクル数

内蔵 ROM などの 16 ビットバスに接続されたメモリ上のプログラムをフェッチする場合には、実行中の命令がワード境界を越えるごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

外部データバスの 8 ビットバスに接続されたメモリ上のプログラムをフェッチする場合は、実行中の命令の 1 バイトごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

CPU 間欠動作時は、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスのアクセスをすると、低消費電力モード制御レジスタの CG0、CG1 ビットで指定されるサイクル数分 CPU へ供給されるクロックが一時停止しますので、CPU 間欠動作の命令の実行に要するサイクル数は、通常の実行サイクル数に、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスの「アクセス回数」×一時停止の「サイクル数」の値を「補正值」として加算してください。

■ 実行サイクル数計算方法

表 B.5-1、表 B.5-2、表 B.5-3 に命令実行サイクル数および補正值のデータを示します。

表 B.5-1 各種アドレッシングに対する実行サイクル数

コード	オペランド	(a) *	各種アドレッシングに対するレジスタアクセス回数
		各種アドレッシングに対する実行サイクル数	
00 ~ 07	Ri Rwi RLi	命令一覧表に記載	命令一覧表に記載
08 ~ 0B	@RWj	2	1
0C ~ 0F	@RWj+	4	2
10 ~ 17	@Rwi+disp8	2	1
18 ~ 1B	@Rwi+disp16	2	1
1C	@RW0+RW7	4	2
1D	@RW1+RW7	4	2
1E	@PC+disp16	2	0
1F	addr16	1	0

*: (a) は「B.8 F²MC-16LX 命令一覧表」の ~ (サイクル数)、B (補正值) で使用されています。" ~ " と "B" の意味については「B.7 命令一覧表の読み方」を参照してください。

表 B.5-2 実サイクル数算出用サイクル数の補正值

オペランド	(b) バイト * 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「B.8 F²MC-16LX 命令一覧表」の～ (サイクル数), B (補正值) で使用されています。

*2: 外部データバスを使用した場合は, レディ入力および自動レディによりウェイトしたサイクル数も加算する必要があります。

表 B.5-3 プログラムフェッチサイクル数算出用サイクル数の補正值

命令	バイト境界	ワード境界
内部メモリ	-	+2
外部データバス 16 ビット	-	+3
外部データバス 8 ビット	+3	-

(注意事項) ・ 外部データバスを使用した場合は, レディ入力および自動レディでウェイトしたサイクル数も加算する必要があります。

・ 実際にはすべてのプログラムフェッチで, 命令実行が遅くなるわけではないので, この補正值は最悪ケースを算出する場合に使用してください。

B.6 実効アドレスフィールド

表 B.6-1 に実効アドレスフィールドを示します。

■ 実効アドレスフィールド

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記			アドレス形式	アドレス拡張部の バイト数*
00	R0	RW0	RL0	レジスタ直接 ea は左から順に バイト ワード ロングワード の型に対応する。	-
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0+			ポストインクリメント付 レジスタ間接	0
0D	@RW1+				
0E	@RW2+				
0F	@RW3+				
10	@RW0+disp8			8 ビットディスプレースメント付 レジスタ間接	1
11	@RW1+disp8				
12	@RW2+disp8				
13	@RW3+disp8				
14	@RW4+disp8				
15	@RW5+disp8				
16	@RW6+disp8				
17	@RW7+disp8				

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記	アドレス形式	アドレス拡張部の バイト数*
18	@RW0+disp16	16 ビットディスプレースメント付 レジスタ間接	2
19	@RW1+disp16		
1A	@RW2+disp16		
1B	@RW3+disp16		
1C	@RW0+RW7	インデックス付レジスタ間接	0
1D	@RW1+RW7	インデックス付レジスタ間接	0
1E	@PC+disp16	16 ビットディスプレースメント付 PC 間接	2
1F	addr16	直接アドレス	2

*: アドレス拡張部のバイト数は、「B.8 F²MC-16LX 命令一覧表」の#(バイト数)の "+" に当てはまります。
 "#" の意味については「B.7 命令一覧表の読み方」を参照してください。

B.7 命令一覧表の読み方

「B.8 F²MC-16LX 命令一覧表」で使用している項目の説明を表 B.7-1 に、記号の説明を表 B.7-2 に示します。

■ 命令の表示記号の説明

表 B.7-1 命令一覧表の項目の説明

項目	説明
ニーモニック	英大文字、記号：アセンブラ上もそのまま表記します。 英小文字：アセンブラ上では、書き替えて記述します。 英小文字の後の数：命令中のビット幅を示します。
#	バイト数を示します。
~	サイクル数を示します。
RG	命令実行時のレジスタアクセス回数を示します。 CPU 間欠動作時の補正値を算出するのに使用します。
B	命令実行時の実サイクル数の算出用補正値を示します。 命令実行時の実サイクルは ~ 欄の数値を加算したものとなります。
オペレーション	命令の動作を示します。
LH	アキュムレータの bit15 ~ bit8 に対する特殊動作を示します。 Z: 0 を転送する。 X: 符号を拡張して転送する。 - : 転送しない。
AH	アキュムレータの上位 16 ビットに対する特殊動作を示します。 *: AL から AH へ転送する。 - : 転送しない。 Z: AH へ 00 _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 (ステッキビット), N (ネガティブ), Z (ゼロ), V (オーバフロー), C (キャリー) の各フラグの状態を示します。 *: 命令の実行で変化する。 - : 変化しない。 S: 命令の実行でセットされる。 R: 命令の実行でリセットされる。
S	
T	
N	
Z	
V	
C	
RMW	
	リードモディファイライト命令 (1 命令でメモリなどからデータを読み出し、メモリへ書き込む) であるかどうかを示します。 *: リードモディファイライト命令である。 - : リードモディファイライト命令ではない。 (注意事項) 読み書きで意味の異なるアドレスには使用できません。

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
A	32 ビットアキュムレータ 命令により、使用されるビット長が変わります。 バイト : AL の下位 8 ビット ワード : AL の 16 ビット ロング : AL: AH の 32 ビット
AH	A の上位 16 ビット
AL	A の下位 16 ビット
SP	スタックポインタ (USP or SSP)
PC	プログラムカウンタ
PCB	プログラムカウンタバンクレジスタ
DTB	データバンクレジスタ
ADB	アディショナルデータバンクレジスタ
SSB	システムスタックバンクレジスタ
USB	ユーザスタックバンクレジスタ
SPB	カレントスタックバンクレジスタ (SSB or USB)
DPR	ダイレクトページレジスタ
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir	短縮直接アドレス指定
addr16	直接アドレス指定
addr24	物理直接アドレス指定
ad24 0 ~ 15	addr24 の bit0 ~ bit15
ad24 16 ~ 23	addr24 の bit16 ~ bit23
io	I/O 領域 (000000 _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 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
disp8	8 ビットディスプレースメント
disp16	16 ビットディスプレースメント
bp	ビットオフセット値
vct4	ベクタ番号 (0 ~ 15)
vct8	ベクタ番号 (0 ~ 255)
() b	ビットアドレス
rel	PC 相対分岐指定
ear	実効アドレス指定 (コード 00 ~ 07)
eam	実効アドレス指定 (コード 08 ~ 1F)
rlst	レジスタ並び

B.8 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 B.8-1 転送系命令 (バイト) 41 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A,dir	2	3	0	(b)	byte (A) (dir)	Z	*	-	-	-	*	*	-	-	-
MOV A,addr16	3	4	0	(b)	byte (A) (addr16)	Z	*	-	-	-	*	*	-	-	-
MOV A,Ri	1	2	1	0	byte (A) (Ri)	Z	*	-	-	-	*	*	-	-	-
MOV A,ear	2	2	1	0	byte (A) (ear)	Z	*	-	-	-	*	*	-	-	-
MOV A,eam	2+	3 + (a)	0	(b)	byte (A) (eam)	Z	*	-	-	-	*	*	-	-	-
MOV A,io	2	3	0	(b)	byte (A) (io)	Z	*	-	-	-	*	*	-	-	-
MOV A,#imm8	2	2	0	0	byte (A) imm8	Z	*	-	-	-	*	*	-	-	-
MOV A,@A	2	3	0	(b)	byte (A) ((A))	Z	-	-	-	-	*	*	-	-	-
MOV A,@RLi+disp8	3	10	2	(b)	byte (A) ((RLi)+disp8)	Z	*	-	-	-	*	*	-	-	-
MOVN A,#imm4	1	1	0	0	byte (A) imm4	Z	*	-	-	-	R	*	-	-	-
MOVX A,dir	2	3	0	(b)	byte (A) (dir)	X	*	-	-	-	*	*	-	-	-
MOVX A,addr16	3	4	0	(b)	byte (A) (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX A,Ri	2	2	1	0	byte (A) (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX A,ear	2	2	1	0	byte (A) (ear)	X	*	-	-	-	*	*	-	-	-
MOVX A,eam	2+	3 + (a)	0	(b)	byte (A) (eam)	X	*	-	-	-	*	*	-	-	-
MOVX A,io	2	3	0	(b)	byte (A) (io)	X	*	-	-	-	*	*	-	-	-
MOVX A,#imm8	2	2	0	0	byte (A) imm8	X	*	-	-	-	*	*	-	-	-
MOVX A,@A	2	3	0	(b)	byte (A) ((A))	X	-	-	-	-	*	*	-	-	-
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX A,@RLi+disp8	3	10	2	(b)	byte (A) ((RLi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOV dir,A	2	3	0	(b)	byte (dir) (A)	-	-	-	-	-	*	*	-	-	-
MOV addr16,A	3	4	0	(b)	byte (addr16) (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,A	1	2	1	0	byte (Ri) (A)	-	-	-	-	-	*	*	-	-	-
MOV ear,A	2	2	1	0	byte (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOV eam,A	2+	3 + (a)	0	(b)	byte (eam) (A)	-	-	-	-	-	*	*	-	-	-
MOV io,A	2	3	0	(b)	byte (io) (A)	-	-	-	-	-	*	*	-	-	-
MOV @RLi+disp8,A	3	10	2	(b)	byte ((RLi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,ear	2	3	2	0	byte (Ri) (ear)	-	-	-	-	-	*	*	-	-	-
MOV Ri,eam	2+	4 + (a)	1	(b)	byte (Ri) (eam)	-	-	-	-	-	*	*	-	-	-
MOV ear,Ri	2	4	2	0	byte (ear) (Ri)	-	-	-	-	-	*	*	-	-	-
MOV eam,Ri	2+	5 + (a)	1	(b)	byte (eam) (Ri)	-	-	-	-	-	*	*	-	-	-
MOV Ri,#imm8	2	2	1	0	byte (Ri) imm8	-	-	-	-	-	*	*	-	-	-
MOV io,#imm8	3	5	0	(b)	byte (io) imm8	-	-	-	-	-	-	-	-	-	-
MOV dir,#imm8	3	5	0	(b)	byte (dir) imm8	-	-	-	-	-	-	-	-	-	-
MOV ear,#imm8	3	2	1	0	byte (ear) imm8	-	-	-	-	-	*	*	-	-	-
MOV eam,#imm8	3+	4 + (a)	0	(b)	byte (eam) imm8	-	-	-	-	-	-	-	-	-	-
MOV @AL,AH	2	3	0	(b)	byte ((A)) (AH)	-	-	-	-	-	*	*	-	-	-
XCH A,ear	2	4	2	0	byte (A) (ear)	Z	-	-	-	-	-	-	-	-	-
XCH A,eam	2+	5 + (a)	0	2 × (b)	byte (A) (eam)	Z	-	-	-	-	-	-	-	-	-
XCH Ri,ear	2	7	4	0	byte (Ri) (ear)	-	-	-	-	-	-	-	-	-	-
XCH Ri,eam	2+	9 + (a)	2	2 × (b)	byte (Ri) (eam)	-	-	-	-	-	-	-	-	-	-

(注意事項) 表中の (a), (b) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-2 転送系命令 (ワード・ロングワード) 38 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A,dir	2	3	0	(c)	word (A) (dir)	-	*	-	-	-	*	*	-	-	-
MOVW A,addr16	3	4	0	(c)	word (A) (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW A,SP	1	1	0	0	word (A) (SP)	-	*	-	-	-	*	*	-	-	-
MOVW A,RWi	1	2	1	0	word (A) (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW A,ear	2	2	1	0	word (A) (ear)	-	*	-	-	-	*	*	-	-	-
MOVW A,eam	2+	3 + (a)	0	(c)	word (A) (eam)	-	*	-	-	-	*	*	-	-	-
MOVW A,io	2	3	0	(c)	word (A) (io)	-	*	-	-	-	*	*	-	-	-
MOVW A,@A	2	3	0	(c)	word (A) ((A))	-	-	-	-	-	*	*	-	-	-
MOVW A,#imm16	3	2	0	0	word (A) imm16	-	*	-	-	-	*	*	-	-	-
MOVW A,@RWi+disp8	2	5	1	(c)	word (A) ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW A,@RLi+disp8	3	10	2	(c)	word (A) ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW dir,A	2	3	0	(c)	word (dir) (A)	-	-	-	-	-	*	*	-	-	-
MOVW addr16,A	3	4	0	(c)	word (addr16) (A)	-	-	-	-	-	*	*	-	-	-
MOVW SP,A	1	1	0	0	word (SP) (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,A	1	2	1	0	word (RWi) (A)	-	-	-	-	-	*	*	-	-	-
MOVW ear,A	2	2	1	0	word (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOVW eam,A	2+	3 + (a)	0	(c)	word (eam) (A)	-	-	-	-	-	*	*	-	-	-
MOVW io,A	2	3	0	(c)	word (io) (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RWi+disp8,A	2	5	1	(c)	word ((RWi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RLi+disp8,A	3	10	2	(c)	word ((RLi)+disp8) (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,ear	2	3	2	0	word (RWi) (ear)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,eam	2+	4 + (a)	1	(c)	word (RWi) (eam)	-	-	-	-	-	*	*	-	-	-
MOVW ear,RWi	2	4	2	0	word (ear) (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW eam,RWi	2+	5 + (a)	1	(c)	word (eam) (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,#imm16	3	2	1	0	word (RWi) imm16	-	-	-	-	-	*	*	-	-	-
MOVW io,#imm16	4	5	0	(c)	word (io) imm16	-	-	-	-	-	-	-	-	-	-
MOVW ear,#imm16	4	2	1	0	word (ear) imm16	-	-	-	-	-	*	*	-	-	-
MOVW eam,#imm16	4+	4 + (a)	0	(c)	word (eam) imm16	-	-	-	-	-	-	-	-	-	-
MOVW @AL,AH	2	3	0	(c)	word ((A)) (AH)	-	-	-	-	-	*	*	-	-	-
XCHW A,ear	2	4	2	0	word (A) (ear)	-	-	-	-	-	-	-	-	-	-
XCHW A,eam	2+	5 + (a)	0	2 × (c)	word (A) (eam)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, ear	2	7	4	0	word (RWi) (ear)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, eam	2+	9 + (a)	2	2 × (c)	word (RWi) (eam)	-	-	-	-	-	-	-	-	-	-
MOVL A,ear	2	4	2	0	long (A) (ear)	-	-	-	-	-	*	*	-	-	-
MOVL A,eam	2+	5 + (a)	0	(d)	long (A) (eam)	-	-	-	-	-	*	*	-	-	-
MOVL A,#imm32	5	3	0	0	long (A) imm32	-	-	-	-	-	*	*	-	-	-
MOVL ear,A	2	4	2	0	long (ear) (A)	-	-	-	-	-	*	*	-	-	-
MOVL eam,A	2+	5 + (a)	0	(d)	long(eam) (A)	-	-	-	-	-	*	*	-	-	-

(注意事項) 表中の (a), (c), (d) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-3 加減算命令 (バイト・ワード・ロングワード) 42 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A,#imm8	2	2	0	0	byte (A) (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD A,dir	2	5	0	(b)	byte (A) (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD A,ear	2	3	1	0	byte (A) (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD A,eam	2+	4 + (a)	0	(b)	byte (A) (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD ear,A	2	3	2	0	byte (ear) (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC A	1	2	0	0	byte (A) (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,ear	2	3	1	0	byte (A) (A) + (ear) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,eam	2+	4 + (a)	0	(b)	byte (A) (A) + (eam) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A	1	3	0	0	byte (A) (AH) + (AL) + (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
SUB A,#imm8	2	2	0	0	byte (A) (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB A,dir	2	5	0	(b)	byte (A) (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB A,ear	2	3	1	0	byte (A) (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB A,eam	2+	4 + (a)	0	(b)	byte (A) (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB ear,A	2	3	2	0	byte (ear) (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC A	1	2	0	0	byte (A) (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,ear	2	3	1	0	byte (A) (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,eam	2+	4 + (a)	0	(b)	byte (A) (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC A	1	3	0	0	byte (A) (AH) - (AL) - (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
ADDW A	1	2	0	0	word (A) (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW A,ear	2	3	1	0	word (A) (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW A,eam	2+	4+(a)	0	(c)	word (A) (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW A,#imm16	3	2	0	0	word (A) (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW ear,A	2	3	2	0	word (ear) (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW A,ear	2	3	1	0	word (A) (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW A,eam	2+	4+(a)	0	(c)	word (A) (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW A	1	2	0	0	word (A) (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW A,ear	2	3	1	0	word (A) (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW A,eam	2+	4+(a)	0	(c)	word (A) (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW A,#imm16	3	2	0	0	word (A) (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW ear,A	2	3	2	0	word (ear) (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW A,ear	2	3	1	0	word (A) (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW A,eam	2+	4+(a)	0	(c)	word (A) (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL A,ear	2	6	2	0	long (A) (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL A,eam	2+	7+(a)	0	(d)	long (A) (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL A,#imm32	5	4	0	0	long (A) (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL A,ear	2	6	2	0	long (A) (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL A,eam	2+	7+(a)	0	(d)	long (A) (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL A,#imm32	5	4	0	0	long (A) (A) - imm32	-	-	-	-	-	*	*	*	*	-

(注意事項) 表中の (a) ~ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-4 増減算命令 (バイト・ワード・ロングワード) 12 命令

ニーモニック		#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	3	2	0	byte (ear) <-- (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	5+(a)	0	2 × (b)	byte (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	3	2	0	byte (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	5+(a)	0	2 × (b)	byte (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	3	2	0	word (ear) (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	5+(a)	0	2 × (c)	word (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	3	2	0	word (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	5+(a)	0	2 × (c)	word (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	7	4	0	long (ear) (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	9+(a)	0	2 × (d)	long (eam) (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	7	4	0	long (ear) (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	9+(a)	0	2 × (d)	long (eam) (eam) - 1	-	-	-	-	-	*	*	*	-	*

(注意事項) 表中の (a) ~ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-5 比較命令 (バイト・ワード・ロングワード) 11 命令

ニーモニック		#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	0	byte (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMP	A,ear	2	2	1	0	byte (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMP	A,eam	2+	3+(a)	0	(b)	byte (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMP	A,#imm8	2	2	0	0	byte (A) - imm8	-	-	-	-	-	*	*	*	*	-
CMPW	A	1	1	0	0	word (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMPW	A,ear	2	2	1	0	word (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPW	A,eam	2+	3+(a)	0	(c)	word (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPW	A,#imm16	3	2	0	0	word (A) - imm16	-	-	-	-	-	*	*	*	*	-
CMPL	A,ear	2	6	2	0	long (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPL	A,eam	2+	7+(a)	0	(d)	long (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPL	A,#imm32	5	3	0	0	long (A) - imm32	-	-	-	-	-	*	*	*	*	-

(注意事項) 表中の (a) ~ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-6 符号なし乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU A	1	*1	0	0	word (AH) / byte (AL) 商 byte (AL) 余り byte (AH)	-	-	-	-	-	-	-	*	*	-
DIVU A,ear	2	*2	1	0	word (A) / byte (ear) 商 byte (A) 余り byte (ear)	-	-	-	-	-	-	-	*	*	-
DIVU A,eam	2+	*3	0	*6	word (A) / byte (eam) 商 byte (A) 余り byte (eam)	-	-	-	-	-	-	-	*	*	-
DIVUW A,ear	2	*4	1	0	long (A) / word (ear) 商 word (A) 余り word (ear)	-	-	-	-	-	-	-	*	*	-
DIVUW A,eam	2+	*5	0	*7	long (A) / word (eam) 商 word (A) 余り word (eam)	-	-	-	-	-	-	-	*	*	-
MULU A	1	*8	0	0	byte (AH) * byte (AL) word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,ear	2	*9	1	0	byte (A) * byte (ear) word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,eam	2+	*10	0	(b)	byte (A) * byte (eam) word (A)	-	-	-	-	-	-	-	-	-	-
MULUW A	1	*11	0	0	word (AH) * word (AL) Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,ear	2	*12	1	0	word (A) * word (ear) Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,eam	2+	*13	0	(c)	word (A) * word (eam) Long (A)	-	-	-	-	-	-	-	-	-	-

*1: ゼロディバイトの場合, 3 オーバフローの場合, 7 正常の場合, 15
 *2: ゼロディバイトの場合, 4 オーバフローの場合, 8 正常の場合, 16
 *3: ゼロディバイトの場合, 6+(a) オーバフローの場合, 9+(a) 正常の場合, 19+(a)
 *4: ゼロディバイトの場合, 4 オーバフローの場合, 7 正常の場合, 22
 *5: ゼロディバイトの場合, 6+(a) オーバフローの場合, 8+(a) 正常の場合, 26+(a)
 *6: ゼロディバイト, またはオーバフローの場合, (b) 正常の場合, 2 × (b)
 *7: ゼロディバイト, またはオーバフローの場合, (c) 正常の場合, 2 × (c)
 *8: byte(AH) がゼロの場合, 3 byte(AH) がゼロでない場合, 7
 *9: byte(ear) がゼロの場合, 4 byte(ear) がゼロでない場合, 8
 *10: byte(eam) がゼロの場合, 5+(a) byte(eam) がゼロでない場合, 9+(a)
 *11: word(AH) がゼロの場合, 3 word(AH) がゼロでない場合, 11
 *12: word(ear) がゼロの場合, 4 word(ear) がゼロでない場合, 12
 *13: word(eam) がゼロの場合, 5+(a) word(eam) がゼロでない場合, 13+(a)
 (注意事項) 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-7 符号付き乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIV A	2	*1	0	0	word (AH) / byte (AL) 商 --> byte (AL) 余り byte (AH)	Z	-	-	-	-	-	-	*	*	-
DIV A,ear	2	*2	1	0	word (A) / byte (ear) 商 --> byte (A) 余り byte (ear)	Z	-	-	-	-	-	-	*	*	-
DIV A,eam	2+	*3	0	*6	word (A) / byte (eam) 商 byte (A) 余り byte (eam)	Z	-	-	-	-	-	-	*	*	-
DIVW A,ear	2	*4	1	0	long (A) / word (ear) 商 --> word (A) 余り word (ear)	-	-	-	-	-	-	-	*	*	-
DIVW A,eam	2+	*5	0	*7	long (A) / word (eam) 商 word (A) 余り word (eam)	-	-	-	-	-	-	-	*	*	-
MUL A	2	*8	0	0	byte (AH) * byte (AL) word (A)	-	-	-	-	-	-	-	-	-	-
MUL A,ear	2	*9	1	0	byte (A) * byte (ear) word (A)	-	-	-	-	-	-	-	-	-	-
MUL A,eam	2+	*10	0	(b)	byte (A) * byte (eam) word (A)	-	-	-	-	-	-	-	-	-	-
MULW A	2	*11	0	0	word (AH) * word (AL) Long (A)	-	-	-	-	-	-	-	-	-	-
MULW A,ear	2	*12	1	0	word (A) * word (ear) Long (A)	-	-	-	-	-	-	-	-	-	-
MULW A,eam	2+	*13	0	(c)	word (A) * word (eam) Long (A)	-	-	-	-	-	-	-	-	-	-

*1: ゼロディバイトの場合, 3 オーバフローの場合, 8 or 18 正常の場合, 18

*2: ゼロディバイトの場合, 4 オーバフローの場合, 11 or 22 正常の場合, 23

*3: ゼロディバイトの場合, 5+(a) オーバフローの場合 12+(a) or 23+(a) 正常の場合, 24+(a)

*4: 被除数が正: ゼロディバイトの場合, 4 オーバフローの場合, 12 or 30 正常の場合, 31

被除数が負: ゼロディバイトの場合, 4 オーバフローの場合, 12 or 31 正常の場合, 32

*5: 被除数が正: ゼロディバイトの場合, 5+(a) オーバフローの場合, 12+(a) or 31+(a) 正常の場合, 32+(a)

被除数が負: ゼロディバイトの場合, 5+(a) オーバフローの場合, 13+(a) or 32+(a) 正常の場合, 33+(a)

*6: ゼロディバイト, またはオーバフローの場合, (b) 正常の場合, 2 × (b)

*7: ゼロディバイト, またはオーバフローの場合, (c) 正常の場合, 2 × (c)

*8: byte(AH) がゼロの場合, 3 結果が正の場合, 12 結果が負の場合, 13

*9: byte(ear) がゼロの場合, 4 結果が正の場合, 13 結果が負の場合, 14

*10: byte(eam) がゼロの場合, 5+(a) 結果が正の場合, 14+(a) 結果が負の場合, 15+(a)

*11: word(AH) がゼロの場合, 3 結果が正の場合, 16 結果が負の場合, 19

*12: word(ear) がゼロの場合, 4 結果が正の場合, 17 結果が負の場合, 20

*13: word(eam) がゼロの場合, 5+(a) 結果が正の場合, 18+(a) 結果が負の場合, 21+(a)

(注意事項) ・ DIV, DIVW 命令でオーバフロー発生時の実行サイクル数に 2 種類ありますが, 演算前と演算後に検出する場合があります。

DIV, DIVW 命令でオーバフロー発生時には, AL の内容は破壊されます。

・ 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-8 論理 1 命令 (バイト・ワード) 39 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
AND A,#imm8	2	2	0	0	byte (A) (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND A,ear	2	3	1	0	byte (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND A,eam	2+	4+(a)	0	(b)	byte (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND ear,A	2	3	2	0	byte (ear) (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR A,#imm8	2	2	0	0	byte (A) (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR A,ear	2	3	1	0	byte (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR A,eam	2+	4+(a)	0	(b)	byte (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR ear,A	2	3	2	0	byte (ear) (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR A,#imm8	2	2	0	0	byte (A) (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR A,ear	2	3	1	0	byte (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR A,eam	2+	4+(a)	0	(b)	byte (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR ear,A	2	3	2	0	byte (ear) (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT A	1	2	0	0	byte (A) not (A)	-	-	-	-	-	*	*	R	-	-
NOT ear	2	3	2	0	byte (ear) not (ear)	-	-	-	-	-	*	*	R	-	-
NOT eam	2+	5+(a)	0	2 × (b)	byte (eam) not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW A	1	2	0	0	word (A) (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW A,#imm16	3	2	0	0	word (A) (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW A,ear	2	3	1	0	word (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW A,eam	2+	4+(a)	0	(c)	word (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW ear,A	2	3	2	0	word (ear) (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW A	1	2	0	0	word (A) (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW A,#imm16	3	2	0	0	word (A) (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW A,ear	2	3	1	0	word (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW A,eam	2+	4+(a)	0	(c)	word (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW ear,A	2	3	2	0	word (ear) (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW A	1	2	0	0	word (A) (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW A,#imm16	3	2	0	0	word (A) (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW A,ear	2	3	1	0	word (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW A,eam	2+	4+(a)	0	(c)	word (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW ear,A	2	3	2	0	word (ear) (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW A	1	2	0	0	word (A) not (A)	-	-	-	-	-	*	*	R	-	-
NOTW ear	2	3	2	0	word (ear) not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW eam	2+	5+(a)	0	2 × (c)	word (eam) not (eam)	-	-	-	-	-	*	*	R	-	*

(注意事項) 表中の (a), (b), (c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-9 論理 2 演算命令 (ロングワード) 6 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A,ear	2	6	2	0	long (A) (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL A,eam	2+	7+(a)	0	(d)	long (A) (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL A,ear	2	6	2	0	long (A) (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL A,eam	2+	7+(a)	0	(d)	long (A) (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL A,ear	2	6	2	0	long (A) (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL A,eam	2+	7+(a)	0	(d)	long (A) (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

(注意事項) 表中の (a), (d) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-10 符号反転命令 (バイト・ワード) 6 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
NEG A	1	2	0	0	byte (A) 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG ear	2	3	2	0	byte (ear) 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG eam	2+	5+(a)	0	2 × (b)	byte (eam) 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW A	1	2	0	0	word (A) 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW ear	2	3	2	0	word (ear) 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW eam	2+	5+(a)	0	2 × (c)	word (eam) 0 - (eam)	-	-	-	-	-	*	*	*	*	*

(注意事項) 表中の (a), (b), (c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-11 ノーマライズ命令 (ロングワード) 1 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
NRML A,R0	2	*1	1	0	long (A) 最初に "1" が立っていた所まで左 シフトする byte (R0) その場合のシフト回数	-	-	-	-	-	-	*	-	-	-

*1: アキュムレータがすべて "0" であった場合は, 4 "0" 以外であった場合は, 6+(R0)

表 B.8-12 シフト命令 (バイト・ワード・ロングワード) 18 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	0	byte (A) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLC A	2	2	0	0	byte (A) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC ear	2	3	2	0	byte (ear) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC eam	2+	5+(a)	0	2 × (b)	byte (eam) 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ROLC ear	2	3	2	0	byte (ear) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLC eam	2+	5+(a)	0	2 × (b)	byte (eam) 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ASR A,R0	2	*1	1	0	byte (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSR A,R0	2	*1	1	0	byte (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSL A,R0	2	*1	1	0	byte (A) ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRW A	1	2	0	0	word (A) 算術右シフト (A,1 ビット)	-	-	-	-	*	*	*	-	*	-
LSRW A/SHRW A	1	2	0	0	word (A) ロジカル右シフト (A,1 ビット)	-	-	-	-	*	R	*	-	*	-
LSLW A/SHLW A	1	2	0	0	word (A) ロジカル左シフト (A,1 ビット)	-	-	-	-	-	*	*	-	*	-
ASRW A,R0	2	*1	1	0	word (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRW A,R0	2	*1	1	0	word (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLW A,R0	2	*1	1	0	word (A) ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRL A,R0	2	*2	1	0	long (A) 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRL A,R0	2	*2	1	0	long (A) ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLL A,R0	2	*2	1	0	long (A) <- ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-

*1 : R0 が "0" であった場合, 6 "0" 以外であった場合は, 5+(R0)

*2 : R0 が "0" であった場合, 6 "0" 以外であった場合は, 6+(R0)

(注意事項) 表中の (a),(b) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-13 分岐命令 1 31 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	0	(Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNZ/BNE rel	2	*1	0	0	(Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BC/BLO rel	2	*1	0	0	(C) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNC/BHS rel	2	*1	0	0	(C) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BN rel	2	*1	0	0	(N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BP rel	2	*1	0	0	(N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BV rel	2	*1	0	0	(V) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNV rel	2	*1	0	0	(V) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BT rel	2	*1	0	0	(T) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNT rel	2	*1	0	0	(T) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLT rel	2	*1	0	0	(V) xor (N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGE rel	2	*1	0	0	(V) xor (N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLE rel	2	*1	0	0	((V) xor (N)) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGT rel	2	*1	0	0	((V) xor (N)) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLS rel	2	*1	0	0	(C) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BHI rel	2	*1	0	0	(C) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BRA rel	2	*1	0	0	無条件分岐	-	-	-	-	-	-	-	-	-	-
JMP @A	1	2	0	0	word (PC) (A)	-	-	-	-	-	-	-	-	-	-
JMP addr16	3	3	0	0	word (PC) addr16	-	-	-	-	-	-	-	-	-	-
JMP @ear	2	3	1	0	word (PC) (ear)	-	-	-	-	-	-	-	-	-	-
JMP @eam	2+	4+(a)	0	(c)	word (PC) (eam)	-	-	-	-	-	-	-	-	-	-
JMPP @ear 注 1)	2	5	2	0	word (PC) (ear), (PCB) (ear+2)	-	-	-	-	-	-	-	-	-	-
JMPP @eam 注 1)	2+	6+(a)	0	(d)	word (PC) (eam), (PCB) (eam+2)	-	-	-	-	-	-	-	-	-	-
JMPP addr24	4	4	0	0	word (PC) ad24 0-15, (PCB) ad24 16-23	-	-	-	-	-	-	-	-	-	-
CALL @ear 注 2)	2	6	1	(c)	word (PC) (ear)	-	-	-	-	-	-	-	-	-	-
CALL @eam 注 2)	2+	7+(a)	0	2 × (c)	word (PC) (eam)	-	-	-	-	-	-	-	-	-	-
CALL addr16 注 3)	3	6	0	(c)	word (PC) addr16	-	-	-	-	-	-	-	-	-	-
CALLV #vct4 注 3)	1	7	0	2 × (c)	ベクタコール命令	-	-	-	-	-	-	-	-	-	-
CALLP @ear 注 4)	2	10	2	2 × (c)	word (PC) (ear)0-15, (PCB) (ear)16-23	-	-	-	-	-	-	-	-	-	-
CALLP @eam 注 4)	2+	11+(a)	0	*2	word (PC) (eam)0-15, (PCB) (eam)16-23	-	-	-	-	-	-	-	-	-	-
CALLP addr24 注 5)	4	10	0	2 × (c)	word (PC) addr0-15, (PCB) addr16-23	-	-	-	-	-	-	-	-	-	-

*1 : 分岐が行われる場合 , 4 行われない場合 , 3

*2 : 3 × (c)+(b)

注 1) 分岐先アドレスの読み込み (word)

注 2) W : スタックへの退避 (word), R : 分岐先アドレスの読み込み (word)

注 3) スタックへの退避 (word)

注 4) W : スタックへの退避 (long), R : 分岐先アドレスの読み込み (long)

注 5) スタックへの退避 (long)

(注意事項) 表中の (a) ~ (d) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-14 分岐命令 2 19 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A,#imm8,rel	3	*1	0	0	byte (A) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE A,#imm16,rel	4	*1	0	0	word (A) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE ear,#imm8,rel	4	*2	1	0	byte (ear) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE eam,#imm8,rel 注 1)	4+	*3	0	(b)	byte (eam) imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE ear,#imm16,rel	5	*4	1	0	word (ear) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE eam,#imm16,rel 注 1)	5+	*3	0	(c)	word (eam) imm16 で分岐	-	-	-	-	-	*	*	*	*	-
DBNZ ear,rel	3	*5	2	0	byte (ear) = (ear) - 1, (ear) 0 で分岐	-	-	-	-	-	*	*	*	-	-
DBNZ eam,rel	3+	*6	2	2 × (b)	byte (eam) = (eam) - 1, (eam) 0 で分岐	-	-	-	-	-	*	*	*	-	*
DWBNZ ear,rel	3	*5	2	0	word (ear) = (ear) - 1, (ear) 0 で分岐	-	-	-	-	-	*	*	*	-	-
DWBNZ eam,rel	3+	*6	2	2 × (c)	word (eam) = (eam) - 1, (eam) 0 で分岐	-	-	-	-	-	*	*	*	-	*
INT #vct8	2	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT addr16	3	16	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INTP addr24	4	17	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT9	1	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
RETI	1	*8	0	*7	割込みからの復帰	-	-	*	*	*	*	*	*	*	-
LINK #imm8	2	6	0	(c)	関数の入口で、旧フレームポインタをスタックに保存し、新フレームポインタの設定、およびローカルポインタの領域を確保する	-	-	-	-	-	-	-	-	-	-
UNLINK	1	5	0	(c)	関数の出口で、旧フレームポインタをスタックから復帰させる	-	-	-	-	-	-	-	-	-	-
RET 注 2)	1	4	0	(c)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-
RETP 注 3)	1	6	0	(d)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-

*1: 分岐が行われる場合, 5 行われない場合, 4

*2: 分岐が行われる場合, 13 行われない場合, 12

*3: 分岐が行われる場合, 7+(a) 行われない場合, 6+(a)

*4: 分岐が行われる場合, 8 行われない場合, 7

*5: 分岐が行われる場合, 7 行われない場合, 6

*6: 分岐が行われる場合, 8+(a) 行われない場合, 7+(a)

*7: 次の割込み要求へ分岐する場合, 3 × (b)+2 × (c) 今の割込みから復帰の場合, 6 × (c)

*8: 次の割込みへ分岐する場合, 15 今の割込みから復帰する場合, 17

注 1) CBNE / CWBNE 命令では, RWj+ のアドレッシングモードは, 使用しないでください。

注 2) スタックからの復帰 (word)

注 3) スタックからの復帰 (long)

(注意事項) 表中の (a) ~ (d) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-15 その他制御命令 (バイト・ワード・ロングワード) 28 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (A)	-	-	-	-	-	-	-	-	-	-
PUSHW AH	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (AH)	-	-	-	-	-	-	-	-	-	-
PUSHW PS	1	4	0	(c)	word (SP) (SP) - 2, ((SP)) (PS)	-	-	-	-	-	-	-	-	-	-
PUSHW rlst	2	*3	*5	*4	(SP) (SP) - 2n, ((SP)) (rlst)	-	-	-	-	-	-	-	-	-	-
POPW A	1	3	0	(c)	word (A) ((SP)), (SP) (SP) + 2	-	*	-	-	-	-	-	-	-	-
POPW AH	1	3	0	(c)	word (AH) ((SP)), (SP) (SP) + 2	-	-	-	-	-	-	-	-	-	-
POPW PS	1	4	0	(c)	word (PS) ((SP)), (SP) (SP) + 2	-	-	*	*	*	*	*	*	*	-
POPW rlst	2	*2	*5	*4	(rlst) ((SP)), (SP) (SP) + 2n	-	-	-	-	-	-	-	-	-	-
JCTX @A	1	14	0	6 × (c)	コンテキストスイッチ命令	-	-	*	*	*	*	*	*	*	-
AND CCR,#imm8	2	3	0	0	byte (CCR) (CCR) and imm8	-	-	*	*	*	*	*	*	*	-
OR CCR,#imm8	2	3	0	0	byte (CCR) (CCR) or imm8	-	-	*	*	*	*	*	*	*	-
MOV RP,#imm8	2	2	0	0	byte (RP) imm8	-	-	-	-	-	-	-	-	-	-
MOV ILM,#imm8	2	2	0	0	byte (ILM) imm8	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,ear	2	3	1	0	word (RWi) ear	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,eam	2+	2+(a)	1	0	word (RWi) eam	-	-	-	-	-	-	-	-	-	-
MOVEA A,ear	2	1	0	0	word (A) ear	-	*	-	-	-	-	-	-	-	-
MOVEA A,eam	2+	1+(a)	0	0	word (A) eam	-	*	-	-	-	-	-	-	-	-
ADDSP #imm8	2	3	0	0	word (SP) (SP) + ext(imm8)	-	-	-	-	-	-	-	-	-	-
ADDSP #imm16	3	3	0	0	word (SP) (SP) + imm16	-	-	-	-	-	-	-	-	-	-
MOV A,brg1	2	*1	0	0	byte (A) (brg1)	Z	*	-	-	-	*	*	-	-	-
MOV brg2,A	2	1	0	0	byte (brg2) (A)	-	-	-	-	-	*	*	-	-	-
NOP	1	1	0	0	ノーオペレーション	-	-	-	-	-	-	-	-	-	-
ADB	1	1	0	0	AD 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
DTB	1	1	0	0	DT 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
PCB	1	1	0	0	PC 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
SPB	1	1	0	0	SP 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
NCC	1	1	0	0	フラグ無変化用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
CMR	1	1	0	0	コモンレジスタバンク用プリフィックス	-	-	-	-	-	-	-	-	-	-

*1 : PCB,ADB,SSB,USB 1 ステート

DTB,DPR 2 ステート

*2 : 7+3 × (POP する回数)+2 × (POP する最後のレジスタ番号), RLST=0(転送レジスタ無し)の場合 7

*3 : 29+3 × (PUSH する回数)-3 × (PUSH する最後のレジスタ番号), RLST=0(転送レジスタ無し)の場合 8

*4 : (POP する回数) × (c), または (PUSH する回数) × (c)

*5 : (POP する回数), または (PUSH する回数)

(注意事項) 表中の (a),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-16 ビット操作命令 21 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB A,dir:bp	3	5	0	(b)	byte (A) (dir:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,addr16:bp	4	5	0	(b)	byte (A) (addr16:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,io:bp	3	4	0	(b)	byte (A) (io:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB dir:bp,A	3	7	0	2×(b)	bit (dir:bp)b (A)	-	-	-	-	-	*	*	-	-	*
MOVB addr16:bp,A	4	7	0	2×(b)	bit (addr16:bp)b (A)	-	-	-	-	-	*	*	-	-	*
MOVB io:bp,A	3	6	0	2×(b)	bit (io:bp)b (A)	-	-	-	-	-	*	*	-	-	*
SETB dir:bp	3	7	0	2×(b)	bit (dir:bp)b 1	-	-	-	-	-	-	-	-	-	*
SETB addr16:bp	4	7	0	2×(b)	bit (addr16:bp)b 1	-	-	-	-	-	-	-	-	-	*
SETB io:bp	3	7	0	2×(b)	bit (io:bp)b 1	-	-	-	-	-	-	-	-	-	*
CLRB dir:bp	3	7	0	2×(b)	bit (dir:bp)b 0	-	-	-	-	-	-	-	-	-	*
CLRB addr16:bp	4	7	0	2×(b)	bit (addr16:bp)b 0	-	-	-	-	-	-	-	-	-	*
CLRB io:bp	3	7	0	2×(b)	bit (io:bp)b 0	-	-	-	-	-	-	-	-	-	*
BBC dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC io:bp,rel	4	*2	0	(b)	(io:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS io:bp,rel	4	*2	0	(b)	(io:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
SBBS addr16:bp,rel	5	*3	0	2×(b)	(addr16:bp) b = 1 で分岐, bit (addr16:bp) b 1	-	-	-	-	-	-	*	-	-	*
WBTS io:bp	3	*4	0	*5	(io:bp) b = 1 になるまで待つ	-	-	-	-	-	-	-	-	-	-
WBTC io:bp	3	*4	0	*5	(io:bp) b = 0 になるまで待つ	-	-	-	-	-	-	-	-	-	-

*1：分岐が行われる場合，8 分岐が行われない場合，7

*2：分岐が行われる場合，7 分岐が行われない場合，6

*3：条件成立の場合，10 未成立の場合，9

*4：不定回数

*5：条件が成立するまで

(注意事項) 表中の (b) は，表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-17 アキュムレータ操作命令 (バイト・ワード) 6 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	0	byte (A)0-7 (A)8-15	-	-	-	-	-	-	-	-	-	-
SWAPW	1	2	0	0	word (AH) (AL)	-	*	-	-	-	-	-	-	-	-
EXT	1	1	0	0	byte 符号拡張	X	-	-	-	-	*	*	-	-	-
EXTW	1	2	0	0	word 符号拡張	-	X	-	-	-	*	*	-	-	-
ZEXT	1	1	0	0	byte ゼロ拡張	Z	-	-	-	-	R	*	-	-	-
ZEXTW	1	1	0	0	word ゼロ拡張	-	Z	-	-	-	R	*	-	-	-

表 B.8-18 スtring命令 10 命令

ニーモニック	#	~	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS / MOVSI	2	*2	*5	*3	byte 転送 @AH+ @AL+, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
MOVSD	2	*2	*5	*3	byte 転送 @AH- @AL-, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
SCEQ / SCEQI	2	*1	*8	*4	byte 検索 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
SCEQD	2	*1	*8	*4	byte 検索 @AH- AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
FILS / FILSI	2	6m+6	*8	*3	byte 充填 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	-	-	-
MOVSW / MOVSWI	2	*2	*5	*6	word 転送 @AH+ @AL+, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
MOVSWD	2	*2	*5	*6	word 転送 @AH- @AL-, カウンタ = RW0	-	-	-	-	-	-	-	-	-	-
SCWEQ / SCWEQI	2	*1	*8	*7	word 検索 @AH+ - AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
SCWEQD	2	*1	*8	*7	word 検索 @AH- - AL, カウンタ = RW0	-	-	-	-	-	*	*	*	*	-
FILSW / FILSWI	2	6m+6	*8	*6	word 充填 @AH+ AL, カウンタ = RW0	-	-	-	-	-	*	*	-	-	-

*1 : RW0 が "0" の場合, 5 カウントアウトの場合, $4+7 \times (RW0)$ 一致した場合, $7n+5$

*2 : RW0 が "0" の場合, 5 "0" 以外の場合, $4+8 \times (RW0)$

*3 : (b) $\times (RW0) + (b) \times (RW0)$ ソースとデスティネーションで異なる領域をアクセスする場合には, それぞれ別々に (b) の項目を算出してください。

*4 : (b)+n

*5 : $2 \times (b) \times (RW0)$

*6 : (c) $\times (RW0) + (c) \times (RW0)$ ソースとデスティネーションで異なる領域をアクセスする場合には, それぞれ別々に (c) の項目を算出してください。

*7 : (c) $\times n$

*8 : (b) $\times (RW0)$

(注意事項) ・ m : RW0 値 (カウンタ値)

n : ループした回数

・ 表中の (b), (c) は, 表 B.5-1 および表 B.5-2 を参照してください。

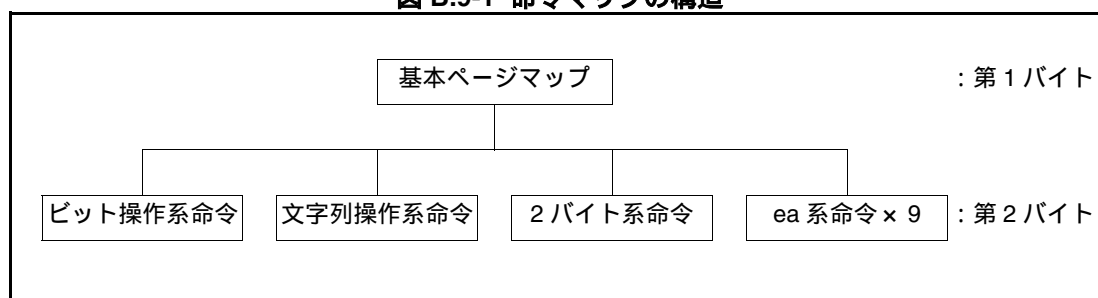
B.9 命令マップ

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

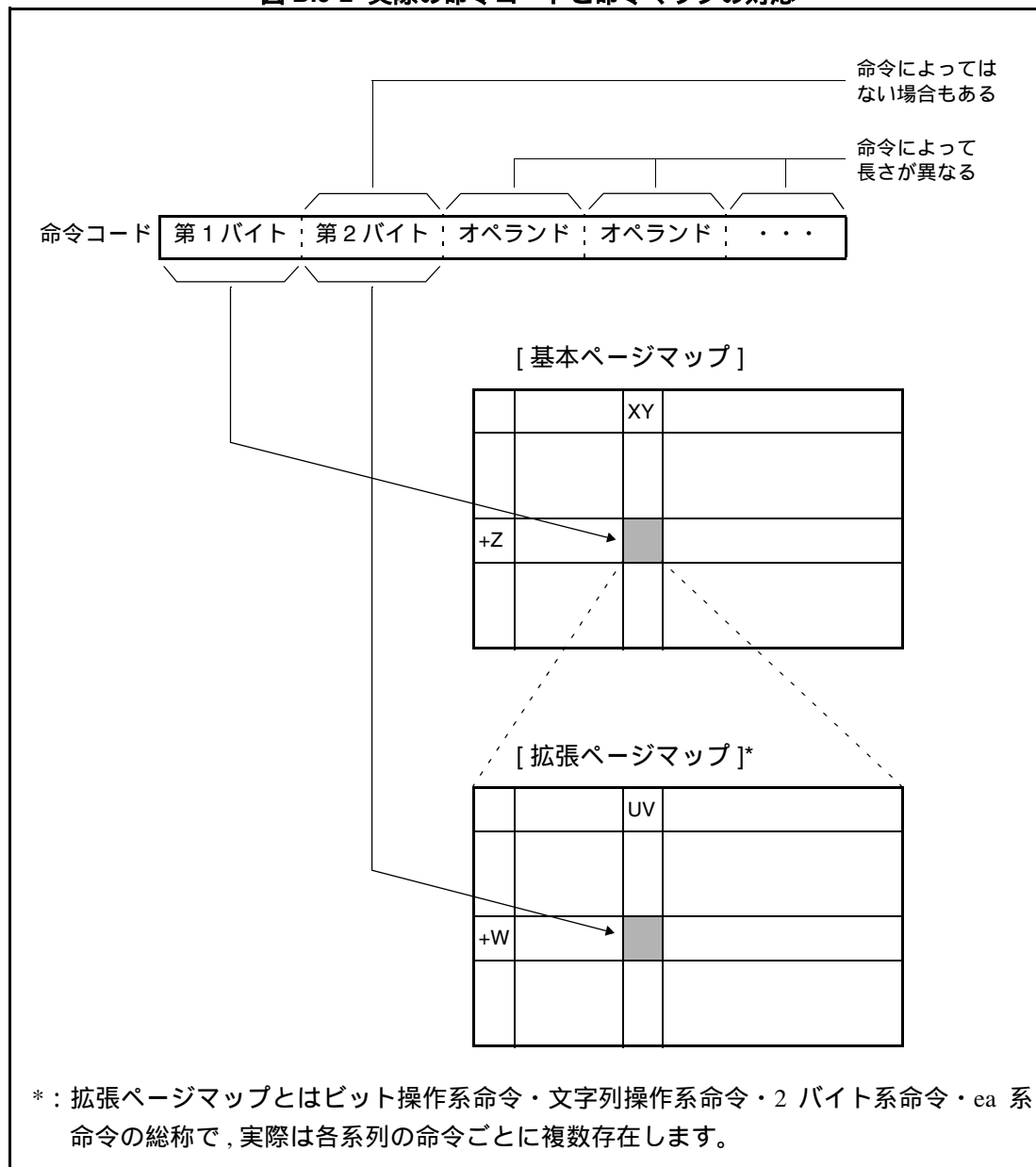
図 B.9-1 命令マップの構造



1 バイトで終わる命令 (NOP 命令など) は基本ページで完結し、2 バイト必要とする命令 (MOVS 命令など) は第 1 バイトを参照した段階で第 2 バイト用のマップの存在がわかり、第 2 バイト用のマップを参照して続く 1 バイトを調べることができます。

実際の命令コードと命令マップの対応を図 B.9-2 に示します。

図 B.9-2 実際の命令コードと命令マップの対応



命令コードの例を表 B.9-1 に示します。

表 B.9-1 命令コードの例

命令	第1バイト (基本ページマップより)	第2バイト (拡張ページマップより)
NOP	$00 + 0 = 00$	
AND A, #8	$30 + 4 = 34$	
MOV A, ADB	$60 + F = 6F$	$00 + 0 = 00$
@RW2+d8, #8, rel	$70 + 0 = 70$	$F0 + 2 = F2$

表 B.9-2 基本ページマップ

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	NOP	CMR	ADD A, dir	ADD A, #8	MOV A, dir	MOV A, io	BRA rel	ea 系命令 その 1	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BZ/BEQ rel
+1	INT9	NCC	SUB A, dir	SUB A, #8	MOV dir, A	MOV io, A	JMP @A	ea 系命令 その 2	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNZ/BNL rel
+2	ADDC A	SUBDC A	ADDC A	SUBC A	MOV A, #8	MOV A, addr16	JMP addr16	ea 系命令 その 3	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BC/BLO rel
+3	NEG A	JCTX @A	CMP A	CMP A, #8	MOVX A, #8	MOV addr16, A	JMPP addr24	ea 系命令 その 4	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNC/BHS rel
+4	PCB	EXT	AND CCR, #8	AND A, #8	MOV dir, #8	MOV io, #8	CALL addr16	ea 系命令 その 5	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BN rel
+5	DTB	ZEXT	OR CCR, #8	OR A, #8	MOVX A, dir	MOVX A, io	CALLP addr24	ea 系命令 その 6	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BP rel
+6	ADB	SWAP	DIVU A	XOR A, #8	MOVW A, SP	MOVW io, #16	RETP	ea 系命令 その 7	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BV rel
+7	SPB	ADDSP #8	MULU #8	NOT A	MOVW SP, A	MOVX A, addr16	RET	ea 系命令 その 8	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNV rel
+8	LINK #imm8	ADDL A, #32	ADDW A	ADDW A, #16	MOVW A, dir	MOVW A, io	INT #vct8	ea 系命令 その 9	MOVW A, RWI	MOVW RWI, A	MOVW RWI, #16	MOVW A, RWI	MOVW @RWI+d8 @RWI+d8,A	MOV A, #4	CALL #4	BT rel
+9	UNLINK	SUBL A, #32	SUBW A	SUBW A, #16	MOVW dir, A	MOVW io, A	INT	MOVEA RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BNT rel
+A	MOV RP, #8	MOV ILM, #8	CBNE A, #8, rel	CWBNL A, #16, rel	MOVW A, #16	MOVW A, addr16	INTP	MOV Ri, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLT rel
+B	NEGW A	CMPL A, #32	CMPW A	CMPW A, #16	MOVL A, #32	MOVW addr16, A	RETI	MOVW RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BGE rel
+C	LSLW A	EXTW	ANDW A	ANDW A, #16	PUSHW A	POPW A	ビット操 作系命令	MOV ea, Ri	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, RWI	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BGT rel
+E	ASRW A	SWAPW A	XORW A	XORW A, #16	PUSHW PS	POPW PS	文字列操 作系命令	XCH Ri, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BLS rel
+F	LSRW A	ADDSP #16	MULW A	NOTW A	PUSHW r1st	POPW r1st	2 バイト 系命令	XCHW RWI, ea	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWI+d8	MOV A, #4	CALL #4	BHI rel

表 B.9-3 ビット操作系命令マップ (第1バイト = 6C_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 B.9-4 文字列操作系命令マップ (第 1 バイト = 6E_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVSI, MOVSD, MOVSWI, MOVSWD								SCWEQI, SCEQD, SCWEQD, FILSI	PCB, DTB, ADB, SPB					FILSI, PCB, DTB, ADB, SPB	
+1	PCB, DTB								PCB, DTB, ADB, SPB							
+2	PCB, ADB								PCB, DTB, ADB, SPB							
+3	PCB, SPB								PCB, DTB, ADB, SPB							
+4	DTB, PCB															
+5	DTB, DTB															
+6	DTB, ADB															
+7	DTB, SPB															
+8	ADB, PCB															
+9	ADB, DTB															
+A	ADB, ADB															
+B	ADB, SPB															
+C	SPB, PCB															
+D	SPB, DTB															
+E	SPB, ADB															
+F	SPB, SPB															

表 B.9-5 2 バイト系命令マップ (第 1 バイト =6F_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	LSL A, R0	LSL A, R0	MOVW @RL2+d8, A	MOVW A, @RL2+d8											
+D	MOVW A, @A	MOVW @AL, AH	NRML A, R0													
+E	ASRW A, R0	ASRL A, R0	ASR A, R0	MOVW @RL3+d8, A	MOVW A, @RL3+d8											
+F	LSRW A, R0	LSRL A, R0	LSR A, R0													

表 B.9-6 ea 系命令 その 1 (第 1 バイト = 70_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	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	R0, @RW0+d8 #8, rel	R0, @RW0+d8 #8, rel
+1	ADDL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	RW1, @RW1+d8 #16, rel	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	R1, @RW1+d8 #8, rel	R1, @RW1+d8 #8, rel
+2	ADDL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	RW2, @RW2+d8 #16, rel	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	R2, @RW2+d8 #8, rel	R2, @RW2+d8 #8, rel
+3	ADDL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	RW3, @RW3+d8 #16, rel	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	R3, @RW3+d8 #8, rel	R3, @RW3+d8 #8, rel
+4	ADDL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	RW4, @RW4+d8 #16, rel	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	R4, @RW4+d8 #8, rel	R4, @RW4+d8 #8, rel
+5	ADDL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	RW5, @RW5+d8 #16, rel	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	R5, @RW5+d8 #8, rel	R5, @RW5+d8 #8, rel
+6	ADDL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	RW6, @RW6+d8 #16, rel	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	R6, @RW6+d8 #8, rel	R6, @RW6+d8 #8, rel
+7	ADDL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	RW7, @RW7+d8 #16, rel	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	R7, @RW7+d8 #8, rel	R7, @RW7+d8 #8, rel
+8	ADDL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #16, rel	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #8, rel	@RW0, @RW0+d16 #8, rel
+9	ADDL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #16, rel	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #8, rel	@RW1, @RW1+d16 #8, rel
+A	ADDL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #16, rel	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #8, rel	@RW2, @RW2+d16 #8, rel
+B	ADDL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #16, rel	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #8, rel	@RW3, @RW3+d16 #8, rel
+C	ADDL A, A, @RW0+R7	SUBL A, A, @RW0+R7	SUBL A, A, @RW0+R7	SUBL A, A, @RW0+R7	@RW0+R7 #16, rel	CMPL A, A, @RW0+R7	CMPL A, A, @RW0+R7	CMPL A, A, @RW0+R7	ANDL A, A, @RW0+R7	ANDL A, A, @RW0+R7	ORL A, A, @RW0+R7	ORL A, A, @RW0+R7	XORL A, A, @RW0+R7	XORL A, A, @RW0+R7	使用禁止	使用禁止
+D	ADDL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	@RW1+R7 #16, rel	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ORL A, A, @RW1+R7	ORL A, A, @RW1+R7	XORL A, A, @RW1+R7	XORL A, A, @RW1+R7	使用禁止	使用禁止
+E	ADDL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	SUBL A, A, @RW2+, @PC+d16	@PC+d16 #16, rel	CMPL A, A, @RW2+, @PC+d16	CMPL A, A, @RW2+, @PC+d16	CMPL A, A, @RW2+, @PC+d16	ANDL A, A, @RW2+, @PC+d16	ANDL A, A, @RW2+, @PC+d16	ORL A, A, @RW2+, @PC+d16	ORL A, A, @RW2+, @PC+d16	XORL A, A, @RW2+, @PC+d16	XORL A, A, @RW2+, @PC+d16	@PC+d16 #8, rel	@PC+d16 #8, rel
+F	ADDL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	SUBL A, A, @RW3+, addr16	addr16 #16, rel	CMPL A, A, @RW3+, addr16	CMPL A, A, @RW3+, addr16	CMPL A, A, @RW3+, addr16	ANDL A, A, @RW3+, addr16	ANDL A, A, @RW3+, addr16	ORL A, A, @RW3+, addr16	ORL A, A, @RW3+, addr16	XORL A, A, @RW3+, addr16	XORL A, A, @RW3+, addr16	使用禁止	使用禁止

表 B.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	MOVL RL0, A, @RW0+d8	MOV R0, #8, @RW0+d8, #8	MOV R1, #8, @RW1+d8, #8	MOVEA A, RW0, @RW0+d8	MOVEA A, RW0, @RW0+d8
+1	JMPP @RL0, @RW1+d8	JMPP @RL0, @RW1+d8	CALLP @RL0, @RW1+d8	CALLP @RL0, @RW1+d8	INCL RL0, @RW1+d8	INCL RL0, @RW1+d8	DECL RL0, @RW1+d8	DECL RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOVL RL0, A, @RW1+d8	MOVL RL0, A, @RW1+d8	MOV R1, #8, @RW1+d8, #8	MOV R2, #8, @RW2+d8, #8	MOVEA A, RW1, @RW1+d8	MOVEA A, RW1, @RW1+d8
+2	JMPP @RL1, @RW2+d8	JMPP @RL1, @RW2+d8	CALLP @RL1, @RW2+d8	CALLP @RL1, @RW2+d8	INCL RL1, @RW2+d8	INCL RL1, @RW2+d8	DECL RL1, @RW2+d8	DECL RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOVL RL1, A, @RW2+d8	MOVL RL1, A, @RW2+d8	MOV R2, #8, @RW2+d8, #8	MOV R3, #8, @RW3+d8, #8	MOVEA A, RW2, @RW2+d8	MOVEA A, RW2, @RW2+d8
+3	JMPP @RL1, @RW3+d8	JMPP @RL1, @RW3+d8	CALLP @RL1, @RW3+d8	CALLP @RL1, @RW3+d8	INCL RL1, @RW3+d8	INCL RL1, @RW3+d8	DECL RL1, @RW3+d8	DECL RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOVL RL1, A, @RW3+d8	MOVL RL1, A, @RW3+d8	MOV R3, #8, @RW3+d8, #8	MOV R4, #8, @RW4+d8, #8	MOVEA A, RW3, @RW3+d8	MOVEA A, RW3, @RW3+d8
+4	JMPP @RL2, @RW4+d8	JMPP @RL2, @RW4+d8	CALLP @RL2, @RW4+d8	CALLP @RL2, @RW4+d8	INCL RL2, @RW4+d8	INCL RL2, @RW4+d8	DECL RL2, @RW4+d8	DECL RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOVL RL2, A, @RW4+d8	MOVL RL2, A, @RW4+d8	MOV R4, #8, @RW4+d8, #8	MOV R5, #8, @RW5+d8, #8	MOVEA A, RW4, @RW4+d8	MOVEA A, RW4, @RW4+d8
+5	JMPP @RL2, @RW5+d8	JMPP @RL2, @RW5+d8	CALLP @RL2, @RW5+d8	CALLP @RL2, @RW5+d8	INCL RL2, @RW5+d8	INCL RL2, @RW5+d8	DECL RL2, @RW5+d8	DECL RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOVL RL2, A, @RW5+d8	MOVL RL2, A, @RW5+d8	MOV R5, #8, @RW5+d8, #8	MOV R6, #8, @RW6+d8, #8	MOVEA A, RW5, @RW5+d8	MOVEA A, RW5, @RW5+d8
+6	JMPP @RL3, @RW6+d8	JMPP @RL3, @RW6+d8	CALLP @RL3, @RW6+d8	CALLP @RL3, @RW6+d8	INCL RL3, @RW6+d8	INCL RL3, @RW6+d8	DECL RL3, @RW6+d8	DECL RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOVL RL3, A, @RW6+d8	MOVL RL3, A, @RW6+d8	MOV R6, #8, @RW6+d8, #8	MOV R7, #8, @RW7+d8, #8	MOVEA A, RW6, @RW6+d8	MOVEA A, RW6, @RW6+d8
+7	JMPP @RL3, @RW7+d8	JMPP @RL3, @RW7+d8	CALLP @RL3, @RW7+d8	CALLP @RL3, @RW7+d8	INCL RL3, @RW7+d8	INCL RL3, @RW7+d8	DECL RL3, @RW7+d8	DECL RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOVL RL3, A, @RW7+d8	MOVL RL3, A, @RW7+d8	MOV R7, #8, @RW7+d8, #8	MOV R8, #8, @RW8+d8, #8	MOVEA A, RW7, @RW7+d8	MOVEA A, RW7, @RW7+d8
+8	JMPP @ @RW0, @RW0+d16	JMPP @ @RW0, @RW0+d16	CALLP @ @RW0, @RW0+d16	CALLP @ @RW0, @RW0+d16	INCL @RW0, @RW0+d16	INCL @RW0, @RW0+d16	DECL @RW0, @RW0+d16	DECL @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOVL @RW0, A, @RW0+d16	MOVL @RW0, A, @RW0+d16	MOV @RW0, #8, @RW0+d16, #8	MOV @RW1, #8, @RW1+d16, #8	MOVEA A, @RW0, @RW0+d16	MOVEA A, @RW0, @RW0+d16
+9	JMPP @ @RW1, @RW1+d16	JMPP @ @RW1, @RW1+d16	CALLP @ @RW1, @RW1+d16	CALLP @ @RW1, @RW1+d16	INCL @RW1, @RW1+d16	INCL @RW1, @RW1+d16	DECL @RW1, @RW1+d16	DECL @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOVL @RW1, A, @RW1+d16	MOVL @RW1, A, @RW1+d16	MOV @RW1, #8, @RW1+d16, #8	MOV @RW2, #8, @RW2+d16, #8	MOVEA A, @RW1, @RW1+d16	MOVEA A, @RW1, @RW1+d16
+A	JMPP @ @RW2, @RW2+d16	JMPP @ @RW2, @RW2+d16	CALLP @ @RW2, @RW2+d16	CALLP @ @RW2, @RW2+d16	INCL @RW2, @RW2+d16	INCL @RW2, @RW2+d16	DECL @RW2, @RW2+d16	DECL @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOVL @RW2, A, @RW2+d16	MOVL @RW2, A, @RW2+d16	MOV @RW2, #8, @RW2+d16, #8	MOV @RW3, #8, @RW3+d16, #8	MOVEA A, @RW2, @RW2+d16	MOVEA A, @RW2, @RW2+d16
+B	JMPP @ @RW3, @RW3+d16	JMPP @ @RW3, @RW3+d16	CALLP @ @RW3, @RW3+d16	CALLP @ @RW3, @RW3+d16	INCL @RW3, @RW3+d16	INCL @RW3, @RW3+d16	DECL @RW3, @RW3+d16	DECL @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOVL @RW3, A, @RW3+d16	MOVL @RW3, A, @RW3+d16	MOV @RW3, #8, @RW3+d16, #8	MOV @RW4, #8, @RW4+d16, #8	MOVEA A, @RW3, @RW3+d16	MOVEA A, @RW3, @RW3+d16
+C	JMPP @ @RW0+, @RW0+RW7	JMPP @ @RW0+, @RW0+RW7	CALLP @ @RW0+, @RW0+RW7	CALLP @ @RW0+, @RW0+RW7	INCL @RW0+, @RW0+RW7	INCL @RW0+, @RW0+RW7	DECL @RW0+, @RW0+RW7	DECL @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOVL @RW0+, A, @RW0+RW7	MOVL @RW0+, A, @RW0+RW7	MOV @RW0+, #8, @RW0+RW7, #8	MOV @RW1+, #8, @RW1+RW7, #8	MOVEA A, @RW0+, @RW0+RW7	MOVEA A, @RW0+, @RW0+RW7
+D	JMPP @ @RW1+, @RW1+RW7	JMPP @ @RW1+, @RW1+RW7	CALLP @ @RW1+, @RW1+RW7	CALLP @ @RW1+, @RW1+RW7	INCL @RW1+, @RW1+RW7	INCL @RW1+, @RW1+RW7	DECL @RW1+, @RW1+RW7	DECL @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOVL @RW1+, A, @RW1+RW7	MOVL @RW1+, A, @RW1+RW7	MOV @RW1+, #8, @RW1+RW7, #8	MOV @RW2+, #8, @RW2+RW7, #8	MOVEA A, @RW1+, @RW1+RW7	MOVEA A, @RW1+, @RW1+RW7
+E	JMPP @ @RW2+, @PC+d16	JMPP @ @RW2+, @PC+d16	CALLP @ @RW2+, @PC+d16	CALLP @ @RW2+, @PC+d16	INCL @RW2+, @PC+d16	INCL @RW2+, @PC+d16	DECL @RW2+, @PC+d16	DECL @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOVL @RW2+, A, @PC+d16, A	MOVL @RW2+, A, @PC+d16, A	MOV @RW2+, #8, @PC+d16, #8	MOV @RW3+, #8, @RW3+d16, #8	MOVEA A, @RW2+, @PC+d16	MOVEA A, @RW2+, @PC+d16
+F	JMPP @ @RW3+, @addr16	JMPP @ @RW3+, @addr16	CALLP @ @RW3+, @addr16	CALLP @ @RW3+, @addr16	INCL @RW3+, @addr16	INCL @RW3+, @addr16	DECL @RW3+, @addr16	DECL @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOVL @RW3+, A, @addr16, A	MOVL @RW3+, A, @addr16, A	MOV @RW3+, #8, @addr16, #8	MOV @RW4+, #8, @addr16, #8	MOVEA A, @RW3+, @addr16	MOVEA A, @RW3+, @addr16

表 B.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	A, R0' @RW0+d8	MOV	MOV	MOVX	A, R0' @RW0+d8	XCH	XCH
+1	ROL	R1' @RW1+d8	RORC	R1' @RW1+d8	INC	R1' @RW1+d8	DEC	R1' @RW1+d8	MOV	A, R1' @RW1+d8	MOV	MOV	MOVX	A, R1' @RW1+d8	XCH	XCH
+2	ROL	R2' @RW2+d8	RORC	R2' @RW2+d8	INC	R2' @RW2+d8	DEC	R2' @RW2+d8	MOV	A, R2' @RW2+d8	MOV	MOV	MOVX	A, R2' @RW2+d8	XCH	XCH
+3	ROL	R3' @RW3+d8	RORC	R3' @RW3+d8	INC	R3' @RW3+d8	DEC	R3' @RW3+d8	MOV	A, R3' @RW3+d8	MOV	MOV	MOVX	A, R3' @RW3+d8	XCH	XCH
+4	ROL	R4' @RW4+d8	RORC	R4' @RW4+d8	INC	R4' @RW4+d8	DEC	R4' @RW4+d8	MOV	A, R4' @RW4+d8	MOV	MOV	MOVX	A, R4' @RW4+d8	XCH	XCH
+5	ROL	R5' @RW5+d8	RORC	R5' @RW5+d8	INC	R5' @RW5+d8	DEC	R5' @RW5+d8	MOV	A, R5' @RW5+d8	MOV	MOV	MOVX	A, R5' @RW5+d8	XCH	XCH
+6	ROL	R6' @RW6+d8	RORC	R6' @RW6+d8	INC	R6' @RW6+d8	DEC	R6' @RW6+d8	MOV	A, R6' @RW6+d8	MOV	MOV	MOVX	A, R6' @RW6+d8	XCH	XCH
+7	ROL	R7' @RW7+d8	RORC	R7' @RW7+d8	INC	R7' @RW7+d8	DEC	R7' @RW7+d8	MOV	A, R7' @RW7+d8	MOV	MOV	MOVX	A, R7' @RW7+d8	XCH	XCH
+8	ROL	R0' @RW0+d16	RORC	R0' @RW0+d16	INC	R0' @RW0+d16	DEC	R0' @RW0+d16	MOV	A, R0' @RW0+d16	MOV	MOV	MOVX	A, R0' @RW0+d16	XCH	XCH
+9	ROL	R1' @RW1+d16	RORC	R1' @RW1+d16	INC	R1' @RW1+d16	DEC	R1' @RW1+d16	MOV	A, R1' @RW1+d16	MOV	MOV	MOVX	A, R1' @RW1+d16	XCH	XCH
+A	ROL	R2' @RW2+d16	RORC	R2' @RW2+d16	INC	R2' @RW2+d16	DEC	R2' @RW2+d16	MOV	A, R2' @RW2+d16	MOV	MOV	MOVX	A, R2' @RW2+d16	XCH	XCH
+B	ROL	R3' @RW3+d16	RORC	R3' @RW3+d16	INC	R3' @RW3+d16	DEC	R3' @RW3+d16	MOV	A, R3' @RW3+d16	MOV	MOV	MOVX	A, R3' @RW3+d16	XCH	XCH
+C	ROL	R0' @RW0+R7	RORC	R0' @RW0+R7	INC	R0' @RW0+R7	DEC	R0' @RW0+R7	MOV	A, R0' @RW0+R7	MOV	MOV	MOVX	A, R0' @RW0+R7	XCH	XCH
+D	ROL	R1' @RW1+R7	RORC	R1' @RW1+R7	INC	R1' @RW1+R7	DEC	R1' @RW1+R7	MOV	A, R1' @RW1+R7	MOV	MOV	MOVX	A, R1' @RW1+R7	XCH	XCH
+E	ROL	R2' @RW2+R7	RORC	R2' @RW2+R7	INC	R2' @RW2+R7	DEC	R2' @RW2+R7	MOV	A, R2' @RW2+R7	MOV	MOV	MOVX	A, R2' @RW2+R7	XCH	XCH
+F	ROL	R3' @RW3+R7	RORC	R3' @RW3+R7	INC	R3' @RW3+R7	DEC	R3' @RW3+R7	MOV	A, R3' @RW3+R7	MOV	MOV	MOVX	A, R3' @RW3+R7	XCH	XCH

表 B.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, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	MOVW RW0, #16, @RW0+d8, #16	XCHW A, RW0, @@RW0+d8	XCHW A, RW0, @@RW0+d8
+1	JMP @RW1, @@RW1+d8	JMP @RW1, @@RW1+d8	CALL RW1, @@RW1+d8	CALL RW1, @@RW1+d8	INCW RW1, @@RW1+d8	INCW RW1, @@RW1+d8	DECW RW1, @@RW1+d8	DECW RW1, @@RW1+d8	MOVW A, RW1, @@RW1+d8	MOVW A, RW1, @@RW1+d8	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	MOVW RW1, #16, @RW1+d8, #16	XCHW A, RW1, @@RW1+d8	XCHW A, RW1, @@RW1+d8
+2	JMP @RW2, @@RW2+d8	JMP @RW2, @@RW2+d8	CALL RW2, @@RW2+d8	CALL RW2, @@RW2+d8	INCW RW2, @@RW2+d8	INCW RW2, @@RW2+d8	DECW RW2, @@RW2+d8	DECW RW2, @@RW2+d8	MOVW A, RW2, @@RW2+d8	MOVW A, RW2, @@RW2+d8	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	MOVW RW2, #16, @RW2+d8, #16	XCHW A, RW2, @@RW2+d8	XCHW A, RW2, @@RW2+d8
+3	JMP @RW3, @@RW3+d8	JMP @RW3, @@RW3+d8	CALL RW3, @@RW3+d8	CALL RW3, @@RW3+d8	INCW RW3, @@RW3+d8	INCW RW3, @@RW3+d8	DECW RW3, @@RW3+d8	DECW RW3, @@RW3+d8	MOVW A, RW3, @@RW3+d8	MOVW A, RW3, @@RW3+d8	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	MOVW RW3, #16, @RW3+d8, #16	XCHW A, RW3, @@RW3+d8	XCHW A, RW3, @@RW3+d8
+4	JMP @RW4, @@RW4+d8	JMP @RW4, @@RW4+d8	CALL RW4, @@RW4+d8	CALL RW4, @@RW4+d8	INCW RW4, @@RW4+d8	INCW RW4, @@RW4+d8	DECW RW4, @@RW4+d8	DECW RW4, @@RW4+d8	MOVW A, RW4, @@RW4+d8	MOVW A, RW4, @@RW4+d8	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	MOVW RW4, #16, @RW4+d8, #16	XCHW A, RW4, @@RW4+d8	XCHW A, RW4, @@RW4+d8
+5	JMP @RW5, @@RW5+d8	JMP @RW5, @@RW5+d8	CALL RW5, @@RW5+d8	CALL RW5, @@RW5+d8	INCW RW5, @@RW5+d8	INCW RW5, @@RW5+d8	DECW RW5, @@RW5+d8	DECW RW5, @@RW5+d8	MOVW A, RW5, @@RW5+d8	MOVW A, RW5, @@RW5+d8	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	MOVW RW5, #16, @RW5+d8, #16	XCHW A, RW5, @@RW5+d8	XCHW A, RW5, @@RW5+d8
+6	JMP @RW6, @@RW6+d8	JMP @RW6, @@RW6+d8	CALL RW6, @@RW6+d8	CALL RW6, @@RW6+d8	INCW RW6, @@RW6+d8	INCW RW6, @@RW6+d8	DECW RW6, @@RW6+d8	DECW RW6, @@RW6+d8	MOVW A, RW6, @@RW6+d8	MOVW A, RW6, @@RW6+d8	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	MOVW RW6, #16, @RW6+d8, #16	XCHW A, RW6, @@RW6+d8	XCHW A, RW6, @@RW6+d8
+7	JMP @RW7, @@RW7+d8	JMP @RW7, @@RW7+d8	CALL RW7, @@RW7+d8	CALL RW7, @@RW7+d8	INCW RW7, @@RW7+d8	INCW RW7, @@RW7+d8	DECW RW7, @@RW7+d8	DECW RW7, @@RW7+d8	MOVW A, RW7, @@RW7+d8	MOVW A, RW7, @@RW7+d8	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	MOVW RW7, #16, @RW7+d8, #16	XCHW A, RW7, @@RW7+d8	XCHW A, RW7, @@RW7+d8
+8	JMP @RW0, @RW0+d16	JMP @RW0, @RW0+d16	CALL @RW0, @RW0+d16	CALL @RW0, @RW0+d16	INCW @RW0, @RW0+d16	INCW @RW0, @RW0+d16	DECW @RW0, @RW0+d16	DECW @RW0, @RW0+d16	MOVW A, @RW0, @RW0+d16	MOVW A, @RW0, @RW0+d16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	MOVW @RW0, #16, @RW0+d16, #16	XCHW A, @RW0, @RW0+d16	XCHW A, @RW0, @RW0+d16
+9	JMP @RW1, @RW1+d16	JMP @RW1, @RW1+d16	CALL @RW1, @RW1+d16	CALL @RW1, @RW1+d16	INCW @RW1, @RW1+d16	INCW @RW1, @RW1+d16	DECW @RW1, @RW1+d16	DECW @RW1, @RW1+d16	MOVW A, @RW1, @RW1+d16	MOVW A, @RW1, @RW1+d16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	MOVW @RW1, #16, @RW1+d16, #16	XCHW A, @RW1, @RW1+d16	XCHW A, @RW1, @RW1+d16
+A	JMP @RW2, @RW2+d16	JMP @RW2, @RW2+d16	CALL @RW2, @RW2+d16	CALL @RW2, @RW2+d16	INCW @RW2, @RW2+d16	INCW @RW2, @RW2+d16	DECW @RW2, @RW2+d16	DECW @RW2, @RW2+d16	MOVW A, @RW2, @RW2+d16	MOVW A, @RW2, @RW2+d16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	MOVW @RW2, #16, @RW2+d16, #16	XCHW A, @RW2, @RW2+d16	XCHW A, @RW2, @RW2+d16
+B	JMP @RW3, @RW3+d16	JMP @RW3, @RW3+d16	CALL @RW3, @RW3+d16	CALL @RW3, @RW3+d16	INCW @RW3, @RW3+d16	INCW @RW3, @RW3+d16	DECW @RW3, @RW3+d16	DECW @RW3, @RW3+d16	MOVW A, @RW3, @RW3+d16	MOVW A, @RW3, @RW3+d16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	MOVW @RW3, #16, @RW3+d16, #16	XCHW A, @RW3, @RW3+d16	XCHW A, @RW3, @RW3+d16
+C	JMP @RW0+, @RW0+RW7	JMP @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	MOVW @RW0+, #16, @RW0+RW7, #16	XCHW A, @RW0+, @RW0+RW7	XCHW A, @RW0+, @RW0+RW7
+D	JMP @RW1+, @RW1+RW7	JMP @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	MOVW @RW1+, #16, @RW1+RW7, #16	XCHW A, @RW1+, @RW1+RW7	XCHW A, @RW1+, @RW1+RW7
+E	JMP @RW2+, @PC+d16	JMP @RW2+, @PC+d16	CALL @RW2+, @PC+d16	CALL @RW2+, @PC+d16	INCW @RW2+, @PC+d16	INCW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	MOVW @RW2+, #16, @PC+d16, #16	XCHW A, @RW2+, @PC+d16	XCHW A, @RW2+, @PC+d16
+F	JMP @RW3+, @addr16	JMP @RW3+, @addr16	CALL @RW3+, @addr16	CALL @RW3+, @addr16	INCW @RW3+, @addr16	INCW @RW3+, @addr16	DECW @RW3+, @addr16	DECW @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	MOVW @RW3+, #16, @addr16, #16	XCHW A, @RW3+, @addr16	XCHW A, @RW3+, @addr16

表 B.9-10 ea 系命令 その5 (第1バイト = 74_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD A, R0, @RW0+d8	ADD A, R0, @RW0+d8	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	CMP A, R0, @RW0+d8	CMP A, R0, @RW0+d8	AND A, R0, @RW0+d8	AND A, R0, @RW0+d8	OR A, R0, @RW0+d8	OR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	DBNZ R0, r, RW0+d8, r	DBNZ R0, r, RW0+d8, r
+1	ADD A, R1, @RW1+d8	ADD A, R1, @RW1+d8	SUB A, R1, @RW1+d8	SUB A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	CMP A, R1, @RW1+d8	CMP A, R1, @RW1+d8	AND A, R1, @RW1+d8	AND A, R1, @RW1+d8	OR A, R1, @RW1+d8	OR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	DBNZ R1, r, RW1+d8, r	DBNZ R1, r, RW1+d8, r
+2	ADD A, R2, @RW2+d8	ADD A, R2, @RW2+d8	SUB A, R2, @RW2+d8	SUB A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	CMP A, R2, @RW2+d8	CMP A, R2, @RW2+d8	AND A, R2, @RW2+d8	AND A, R2, @RW2+d8	OR A, R2, @RW2+d8	OR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	DBNZ R2, r, RW2+d8, r	DBNZ R2, r, RW2+d8, r
+3	ADD A, R3, @RW3+d8	ADD A, R3, @RW3+d8	SUB A, R3, @RW3+d8	SUB A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	CMP A, R3, @RW3+d8	CMP A, R3, @RW3+d8	AND A, R3, @RW3+d8	AND A, R3, @RW3+d8	OR A, R3, @RW3+d8	OR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	DBNZ R3, r, RW3+d8, r	DBNZ R3, r, RW3+d8, r
+4	ADD A, R4, @RW4+d8	ADD A, R4, @RW4+d8	SUB A, R4, @RW4+d8	SUB A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	CMP A, R4, @RW4+d8	CMP A, R4, @RW4+d8	AND A, R4, @RW4+d8	AND A, R4, @RW4+d8	OR A, R4, @RW4+d8	OR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	DBNZ R4, r, RW4+d8, r	DBNZ R4, r, RW4+d8, r
+5	ADD A, R5, @RW5+d8	ADD A, R5, @RW5+d8	SUB A, R5, @RW5+d8	SUB A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	CMP A, R5, @RW5+d8	CMP A, R5, @RW5+d8	AND A, R5, @RW5+d8	AND A, R5, @RW5+d8	OR A, R5, @RW5+d8	OR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	DBNZ R5, r, RW5+d8, r	DBNZ R5, r, RW5+d8, r
+6	ADD A, R6, @RW6+d8	ADD A, R6, @RW6+d8	SUB A, R6, @RW6+d8	SUB A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	CMP A, R6, @RW6+d8	CMP A, R6, @RW6+d8	AND A, R6, @RW6+d8	AND A, R6, @RW6+d8	OR A, R6, @RW6+d8	OR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	DBNZ R6, r, RW6+d8, r	DBNZ R6, r, RW6+d8, r
+7	ADD A, R7, @RW7+d8	ADD A, R7, @RW7+d8	SUB A, R7, @RW7+d8	SUB A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	CMP A, R7, @RW7+d8	CMP A, R7, @RW7+d8	AND A, R7, @RW7+d8	AND A, R7, @RW7+d8	OR A, R7, @RW7+d8	OR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	DBNZ R7, r, RW7+d8, r	DBNZ R7, r, RW7+d8, r
+8	ADD A, @RW0, @RW0+d16	ADD A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	DBNZ @RW0, r, W0+d16, r	DBNZ @RW0, r, W0+d16, r
+9	ADD A, @RW1, @RW1+d16	ADD A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	DBNZ @RW1, r, W1+d16, r	DBNZ @RW1, r, W1+d16, r
+A	ADD A, @RW2, @RW2+d16	ADD A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	DBNZ @RW2, r, W2+d16, r	DBNZ @RW2, r, W2+d16, r
+B	ADD A, @RW3, @RW3+d16	ADD A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	DBNZ @RW3, r, W3+d16, r	DBNZ @RW3, r, W3+d16, r
+C	ADD A, @RW0+, @RW0+RW7	ADD A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	DBNZ @RW0+, r, W0+RW7, r	DBNZ @RW0+, r, W0+RW7, r
+D	ADD A, @RW1+, @RW1+RW7	ADD A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	DBNZ @RW1+, r, W1+RW7, r	DBNZ @RW1+, r, W1+RW7, r
+E	ADD A, @RW2+, @PC+d16	ADD A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	DBNZ @RW2+, r, PC+d16, r	DBNZ @RW2+, r, PC+d16, r
+F	ADD A, @RW3+, A, addr16	ADD A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	DBNZ @RW3+, r, addr16, r	DBNZ @RW3+, r, addr16, r

表 B.9-11 ea 系命令 その6 (第1バイト = 75_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 B.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	DWBZ RW0, r' @RW0+d8, r	DWBZ RW0, r' @RW0+d8, r
+1	ADDW A, RW1, @RW1+d8	ADDW A, RW1, @RW1+d8	SUBW A, RW1, @RW1+d8	SUBW A, RW1, @RW1+d8	ADDCW A, RW1, @RW1+d8	ADDCW A, RW1, @RW1+d8	CMPW A, RW1, @RW1+d8	CMPW A, RW1, @RW1+d8	ANDW A, RW1, @RW1+d8	ANDW A, RW1, @RW1+d8	ORW A, RW1, @RW1+d8	ORW A, RW1, @RW1+d8	XORW A, RW1, @RW1+d8	XORW A, RW1, @RW1+d8	DWBZ RW1, r' @RW1+d8, r	DWBZ RW1, r' @RW1+d8, r
+2	ADDW A, RW2, @RW2+d8	ADDW A, RW2, @RW2+d8	SUBW A, RW2, @RW2+d8	SUBW A, RW2, @RW2+d8	ADDCW A, RW2, @RW2+d8	ADDCW A, RW2, @RW2+d8	CMPW A, RW2, @RW2+d8	CMPW A, RW2, @RW2+d8	ANDW A, RW2, @RW2+d8	ANDW A, RW2, @RW2+d8	ORW A, RW2, @RW2+d8	ORW A, RW2, @RW2+d8	XORW A, RW2, @RW2+d8	XORW A, RW2, @RW2+d8	DWBZ RW2, r' @RW2+d8, r	DWBZ RW2, r' @RW2+d8, r
+3	ADDW A, RW3, @RW3+d8	ADDW A, RW3, @RW3+d8	SUBW A, RW3, @RW3+d8	SUBW A, RW3, @RW3+d8	ADDCW A, RW3, @RW3+d8	ADDCW A, RW3, @RW3+d8	CMPW A, RW3, @RW3+d8	CMPW A, RW3, @RW3+d8	ANDW A, RW3, @RW3+d8	ANDW A, RW3, @RW3+d8	ORW A, RW3, @RW3+d8	ORW A, RW3, @RW3+d8	XORW A, RW3, @RW3+d8	XORW A, RW3, @RW3+d8	DWBZ RW3, r' @RW3+d8, r	DWBZ RW3, r' @RW3+d8, r
+4	ADDW A, RW4, @RW4+d8	ADDW A, RW4, @RW4+d8	SUBW A, RW4, @RW4+d8	SUBW A, RW4, @RW4+d8	ADDCW A, RW4, @RW4+d8	ADDCW A, RW4, @RW4+d8	CMPW A, RW4, @RW4+d8	CMPW A, RW4, @RW4+d8	ANDW A, RW4, @RW4+d8	ANDW A, RW4, @RW4+d8	ORW A, RW4, @RW4+d8	ORW A, RW4, @RW4+d8	XORW A, RW4, @RW4+d8	XORW A, RW4, @RW4+d8	DWBZ RW4, r' @RW4+d8, r	DWBZ RW4, r' @RW4+d8, r
+5	ADDW A, RW5, @RW5+d8	ADDW A, RW5, @RW5+d8	SUBW A, RW5, @RW5+d8	SUBW A, RW5, @RW5+d8	ADDCW A, RW5, @RW5+d8	ADDCW A, RW5, @RW5+d8	CMPW A, RW5, @RW5+d8	CMPW A, RW5, @RW5+d8	ANDW A, RW5, @RW5+d8	ANDW A, RW5, @RW5+d8	ORW A, RW5, @RW5+d8	ORW A, RW5, @RW5+d8	XORW A, RW5, @RW5+d8	XORW A, RW5, @RW5+d8	DWBZ RW5, r' @RW5+d8, r	DWBZ RW5, r' @RW5+d8, r
+6	ADDW A, RW6, @RW6+d8	ADDW A, RW6, @RW6+d8	SUBW A, RW6, @RW6+d8	SUBW A, RW6, @RW6+d8	ADDCW A, RW6, @RW6+d8	ADDCW A, RW6, @RW6+d8	CMPW A, RW6, @RW6+d8	CMPW A, RW6, @RW6+d8	ANDW A, RW6, @RW6+d8	ANDW A, RW6, @RW6+d8	ORW A, RW6, @RW6+d8	ORW A, RW6, @RW6+d8	XORW A, RW6, @RW6+d8	XORW A, RW6, @RW6+d8	DWBZ RW6, r' @RW6+d8, r	DWBZ RW6, r' @RW6+d8, r
+7	ADDW A, RW7, @RW7+d8	ADDW A, RW7, @RW7+d8	SUBW A, RW7, @RW7+d8	SUBW A, RW7, @RW7+d8	ADDCW A, RW7, @RW7+d8	ADDCW A, RW7, @RW7+d8	CMPW A, RW7, @RW7+d8	CMPW A, RW7, @RW7+d8	ANDW A, RW7, @RW7+d8	ANDW A, RW7, @RW7+d8	ORW A, RW7, @RW7+d8	ORW A, RW7, @RW7+d8	XORW A, RW7, @RW7+d8	XORW A, RW7, @RW7+d8	DWBZ RW7, r' @RW7+d8, r	DWBZ RW7, r' @RW7+d8, r
+8	ADDW A, @RW0, @RW0+d16	ADDW A, @RW0, @RW0+d16	SUBW A, @RW0, @RW0+d16	SUBW A, @RW0, @RW0+d16	ADDCW A, @RW0, @RW0+d16	ADDCW A, @RW0, @RW0+d16	CMPW A, @RW0, @RW0+d16	CMPW A, @RW0, @RW0+d16	ANDW A, @RW0, @RW0+d16	ANDW A, @RW0, @RW0+d16	ORW A, @RW0, @RW0+d16	ORW A, @RW0, @RW0+d16	XORW A, @RW0, @RW0+d16	XORW A, @RW0, @RW0+d16	DWBZ @RW0, r' @RW0+d16, r	DWBZ @RW0, r' @RW0+d16, r
+9	ADDW A, @RW1, @RW1+d16	ADDW A, @RW1, @RW1+d16	SUBW A, @RW1, @RW1+d16	SUBW A, @RW1, @RW1+d16	ADDCW A, @RW1, @RW1+d16	ADDCW A, @RW1, @RW1+d16	CMPW A, @RW1, @RW1+d16	CMPW A, @RW1, @RW1+d16	ANDW A, @RW1, @RW1+d16	ANDW A, @RW1, @RW1+d16	ORW A, @RW1, @RW1+d16	ORW A, @RW1, @RW1+d16	XORW A, @RW1, @RW1+d16	XORW A, @RW1, @RW1+d16	DWBZ @RW1, r' @RW1+d16, r	DWBZ @RW1, r' @RW1+d16, r
+A	ADDW A, @RW2, @RW2+d16	ADDW A, @RW2, @RW2+d16	SUBW A, @RW2, @RW2+d16	SUBW A, @RW2, @RW2+d16	ADDCW A, @RW2, @RW2+d16	ADDCW A, @RW2, @RW2+d16	CMPW A, @RW2, @RW2+d16	CMPW A, @RW2, @RW2+d16	ANDW A, @RW2, @RW2+d16	ANDW A, @RW2, @RW2+d16	ORW A, @RW2, @RW2+d16	ORW A, @RW2, @RW2+d16	XORW A, @RW2, @RW2+d16	XORW A, @RW2, @RW2+d16	DWBZ @RW2, r' @RW2+d16, r	DWBZ @RW2, r' @RW2+d16, r
+B	ADDW A, @RW3, @RW3+d16	ADDW A, @RW3, @RW3+d16	SUBW A, @RW3, @RW3+d16	SUBW A, @RW3, @RW3+d16	ADDCW A, @RW3, @RW3+d16	ADDCW A, @RW3, @RW3+d16	CMPW A, @RW3, @RW3+d16	CMPW A, @RW3, @RW3+d16	ANDW A, @RW3, @RW3+d16	ANDW A, @RW3, @RW3+d16	ORW A, @RW3, @RW3+d16	ORW A, @RW3, @RW3+d16	XORW A, @RW3, @RW3+d16	XORW A, @RW3, @RW3+d16	DWBZ @RW3, r' @RW3+d16, r	DWBZ @RW3, r' @RW3+d16, r
+C	ADDW A, @RW0+, @RW0+RW7	ADDW A, @RW0+, @RW0+RW7	SUBW A, @RW0+, @RW0+RW7	SUBW A, @RW0+, @RW0+RW7	ADDCW A, @RW0+, @RW0+RW7	ADDCW A, @RW0+, @RW0+RW7	CMPW A, @RW0+, @RW0+RW7	CMPW A, @RW0+, @RW0+RW7	ANDW A, @RW0+, @RW0+RW7	ANDW A, @RW0+, @RW0+RW7	ORW A, @RW0+, @RW0+RW7	ORW A, @RW0+, @RW0+RW7	XORW A, @RW0+, @RW0+RW7	XORW A, @RW0+, @RW0+RW7	DWBZ @RW0+, r' @RW0+RW7, r	DWBZ @RW0+, r' @RW0+RW7, r
+D	ADDW A, @RW1+, @RW1+RW7	ADDW A, @RW1+, @RW1+RW7	SUBW A, @RW1+, @RW1+RW7	SUBW A, @RW1+, @RW1+RW7	ADDCW A, @RW1+, @RW1+RW7	ADDCW A, @RW1+, @RW1+RW7	CMPW A, @RW1+, @RW1+RW7	CMPW A, @RW1+, @RW1+RW7	ANDW A, @RW1+, @RW1+RW7	ANDW A, @RW1+, @RW1+RW7	ORW A, @RW1+, @RW1+RW7	ORW A, @RW1+, @RW1+RW7	XORW A, @RW1+, @RW1+RW7	XORW A, @RW1+, @RW1+RW7	DWBZ @RW1+, r' @RW1+RW7, r	DWBZ @RW1+, r' @RW1+RW7, r
+E	ADDW A, @RW2+, @PC+d16	ADDW A, @RW2+, @PC+d16	SUBW A, @RW2+, @PC+d16	SUBW A, @RW2+, @PC+d16	ADDCW A, @RW2+, @PC+d16	ADDCW A, @RW2+, @PC+d16	CMPW A, @RW2+, @PC+d16	CMPW A, @RW2+, @PC+d16	ANDW A, @RW2+, @PC+d16	ANDW A, @RW2+, @PC+d16	ORW A, @RW2+, @PC+d16	ORW A, @RW2+, @PC+d16	XORW A, @RW2+, @PC+d16	XORW A, @RW2+, @PC+d16	DWBZ @RW2+, r' @PC+d16, r	DWBZ @RW2+, r' @PC+d16, r
+F	ADDW A, @RW3+, addr 16	ADDW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	DWBZ @RW3+, r' addr 16, r	DWBZ @RW3+, r' addr 16, r

表 B.9-13 ea 系命令 その 8 (第 1 バイト = 77_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBCW A, RW0', @RW0+d8	SUBCW A, RW0', @RW0+d8	NEGW RW0', @RW0+d8	NEGW RW0', @RW0+d8	ANDW RW0, A', @RW0+d8, A	ANDW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	NOTW RW0', @RW0+d8	NOTW RW0', @RW0+d8
+1	ADDW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBCW A, RW1', @RW1+d8	SUBCW A, RW1', @RW1+d8	NEGW RW1', @RW1+d8	NEGW RW1', @RW1+d8	ANDW RW1, A', @RW1+d8, A	ANDW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	NOTW RW1', @RW1+d8	NOTW RW1', @RW1+d8
+2	ADDW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBCW A, RW2', @RW2+d8	SUBCW A, RW2', @RW2+d8	NEGW RW2', @RW2+d8	NEGW RW2', @RW2+d8	ANDW RW2, A', @RW2+d8, A	ANDW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	NOTW RW2', @RW2+d8	NOTW RW2', @RW2+d8
+3	ADDW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBCW A, RW3', @RW3+d8	SUBCW A, RW3', @RW3+d8	NEGW RW3', @RW3+d8	NEGW RW3', @RW3+d8	ANDW RW3, A', @RW3+d8, A	ANDW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	NOTW RW3', @RW3+d8	NOTW RW3', @RW3+d8
+4	ADDW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBCW A, RW4', @RW4+d8	SUBCW A, RW4', @RW4+d8	NEGW RW4', @RW4+d8	NEGW RW4', @RW4+d8	ANDW RW4, A', @RW4+d8, A	ANDW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	NOTW RW4', @RW4+d8	NOTW RW4', @RW4+d8
+5	ADDW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBCW A, RW5', @RW5+d8	SUBCW A, RW5', @RW5+d8	NEGW RW5', @RW5+d8	NEGW RW5', @RW5+d8	ANDW RW5, A', @RW5+d8, A	ANDW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	NOTW RW5', @RW5+d8	NOTW RW5', @RW5+d8
+6	ADDW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBCW A, RW6', @RW6+d8	SUBCW A, RW6', @RW6+d8	NEGW RW6', @RW6+d8	NEGW RW6', @RW6+d8	ANDW RW6, A', @RW6+d8, A	ANDW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	NOTW RW6', @RW6+d8	NOTW RW6', @RW6+d8
+7	ADDW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBCW A, RW7', @RW7+d8	SUBCW A, RW7', @RW7+d8	NEGW RW7', @RW7+d8	NEGW RW7', @RW7+d8	ANDW RW7, A', @RW7+d8, A	ANDW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	NOTW RW7', @RW7+d8	NOTW RW7', @RW7+d8
+8	ADDW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBCW A, @RW0', @RW0+d16	SUBCW A, @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	ANDW @RW0, A', @RW0+d16, A	ANDW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	NOTW @RW0', @RW0+d16	NOTW @RW0', @RW0+d16
+9	ADDW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBCW A, @RW1', @RW1+d16	SUBCW A, @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	ANDW @RW1, A', @RW1+d16, A	ANDW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	NOTW @RW1', @RW1+d16	NOTW @RW1', @RW1+d16
+A	ADDW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBCW A, @RW2', @RW2+d16	SUBCW A, @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	ANDW @RW2, A', @RW2+d16, A	ANDW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	NOTW @RW2', @RW2+d16	NOTW @RW2', @RW2+d16
+B	ADDW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBCW A, @RW3', @RW3+d16	SUBCW A, @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	ANDW @RW3, A', @RW3+d16, A	ANDW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	NOTW @RW3', @RW3+d16	NOTW @RW3', @RW3+d16
+C	ADDW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBCW A, @RW0+', @RW0+RW7	SUBCW A, @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	ANDW @RW0+, A', @RW0+RW7, A	ANDW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	NOTW @RW0+', @RW0+RW7	NOTW @RW0+', @RW0+RW7
+D	ADDW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBCW A, @RW1+', @RW1+RW7	SUBCW A, @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	ANDW @RW1+, A', @RW1+RW7, A	ANDW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	NOTW @RW1+', @RW1+RW7	NOTW @RW1+', @RW1+RW7
+E	ADDW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBCW A, @RW2+', @PC+d16	SUBCW A, @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	ANDW @RW2+, A', @PC+d16, A	ANDW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	NOTW @RW2+', @PC+d16	NOTW @RW2+', @PC+d16
+F	ADDW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBCW A, @RW3+', addr16	SUBCW A, @RW3+', addr16	NEGW @RW3+', addr16	NEGW @RW3+', addr16	ANDW @RW3+, A', addr16, A	ANDW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	NOTW @RW3+', addr16	NOTW @RW3+', addr16

表 B.9-14 ea 系命令 その9 (第1バイト = 78_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, A, RW0', @RW0+d8	MULUW A, RW0', @RW0+d8	MULUW A, A, RW0', @RW0+d8	MUL A, R0', @RW0+d8	MUL A, A, R0', @RW0+d8	MULW A, RW0', @RW0+d8	MULW A, A, RW0', @RW0+d8	DIVU A, R0', @RW0+d8	DIVU A, A, R0', @RW0+d8	DIVUW A, RW0', @RW0+d8	DIVUW A, A, RW0', @RW0+d8	DIV A, R0', @RW0+d8	DIV A, A, R0', @RW0+d8	DIVW A, RW0', @RW0+d8	DIVW A, A, RW0', @RW0+d8
+1	MULU A, R1', @RW1+d8	MULU A, A, RW1', @RW1+d8	MULUW A, RW1', @RW1+d8	MULUW A, A, RW1', @RW1+d8	MUL A, R1', @RW1+d8	MUL A, A, R1', @RW1+d8	MULW A, RW1', @RW1+d8	MULW A, A, RW1', @RW1+d8	DIVU A, R1', @RW1+d8	DIVU A, A, R1', @RW1+d8	DIVUW A, RW1', @RW1+d8	DIVUW A, A, RW1', @RW1+d8	DIV A, R1', @RW1+d8	DIV A, A, R1', @RW1+d8	DIVW A, RW1', @RW1+d8	DIVW A, A, RW1', @RW1+d8
+2	MULU A, R2', @RW2+d8	MULU A, A, RW2', @RW2+d8	MULUW A, RW2', @RW2+d8	MULUW A, A, RW2', @RW2+d8	MUL A, R2', @RW2+d8	MUL A, A, R2', @RW2+d8	MULW A, RW2', @RW2+d8	MULW A, A, RW2', @RW2+d8	DIVU A, R2', @RW2+d8	DIVU A, A, R2', @RW2+d8	DIVUW A, RW2', @RW2+d8	DIVUW A, A, RW2', @RW2+d8	DIV A, R2', @RW2+d8	DIV A, A, R2', @RW2+d8	DIVW A, RW2', @RW2+d8	DIVW A, A, RW2', @RW2+d8
+3	MULU A, R3', @RW3+d8	MULU A, A, RW3', @RW3+d8	MULUW A, RW3', @RW3+d8	MULUW A, A, RW3', @RW3+d8	MUL A, R3', @RW3+d8	MUL A, A, R3', @RW3+d8	MULW A, RW3', @RW3+d8	MULW A, A, RW3', @RW3+d8	DIVU A, R3', @RW3+d8	DIVU A, A, R3', @RW3+d8	DIVUW A, RW3', @RW3+d8	DIVUW A, A, RW3', @RW3+d8	DIV A, R3', @RW3+d8	DIV A, A, R3', @RW3+d8	DIVW A, RW3', @RW3+d8	DIVW A, A, RW3', @RW3+d8
+4	MULU A, R4', @RW4+d8	MULU A, A, RW4', @RW4+d8	MULUW A, RW4', @RW4+d8	MULUW A, A, RW4', @RW4+d8	MUL A, R4', @RW4+d8	MUL A, A, R4', @RW4+d8	MULW A, RW4', @RW4+d8	MULW A, A, RW4', @RW4+d8	DIVU A, R4', @RW4+d8	DIVU A, A, R4', @RW4+d8	DIVUW A, RW4', @RW4+d8	DIVUW A, A, RW4', @RW4+d8	DIV A, R4', @RW4+d8	DIV A, A, R4', @RW4+d8	DIVW A, RW4', @RW4+d8	DIVW A, A, RW4', @RW4+d8
+5	MULU A, R5', @RW5+d8	MULU A, A, RW5', @RW5+d8	MULUW A, RW5', @RW5+d8	MULUW A, A, RW5', @RW5+d8	MUL A, R5', @RW5+d8	MUL A, A, R5', @RW5+d8	MULW A, RW5', @RW5+d8	MULW A, A, RW5', @RW5+d8	DIVU A, R5', @RW5+d8	DIVU A, A, R5', @RW5+d8	DIVUW A, RW5', @RW5+d8	DIVUW A, A, RW5', @RW5+d8	DIV A, R5', @RW5+d8	DIV A, A, R5', @RW5+d8	DIVW A, RW5', @RW5+d8	DIVW A, A, RW5', @RW5+d8
+6	MULU A, R6', @RW6+d8	MULU A, A, RW6', @RW6+d8	MULUW A, RW6', @RW6+d8	MULUW A, A, RW6', @RW6+d8	MUL A, R6', @RW6+d8	MUL A, A, R6', @RW6+d8	MULW A, RW6', @RW6+d8	MULW A, A, RW6', @RW6+d8	DIVU A, R6', @RW6+d8	DIVU A, A, R6', @RW6+d8	DIVUW A, RW6', @RW6+d8	DIVUW A, A, RW6', @RW6+d8	DIV A, R6', @RW6+d8	DIV A, A, R6', @RW6+d8	DIVW A, RW6', @RW6+d8	DIVW A, A, RW6', @RW6+d8
+7	MULU A, R7', @RW7+d8	MULU A, A, RW7', @RW7+d8	MULUW A, RW7', @RW7+d8	MULUW A, A, RW7', @RW7+d8	MUL A, R7', @RW7+d8	MUL A, A, R7', @RW7+d8	MULW A, RW7', @RW7+d8	MULW A, A, RW7', @RW7+d8	DIVU A, R7', @RW7+d8	DIVU A, A, R7', @RW7+d8	DIVUW A, RW7', @RW7+d8	DIVUW A, A, RW7', @RW7+d8	DIV A, R7', @RW7+d8	DIV A, A, R7', @RW7+d8	DIVW A, RW7', @RW7+d8	DIVW A, A, RW7', @RW7+d8
+8	MULU A, @RW0', @RW0+d16	MULU A, A, @RW0', @RW0+d16	MULUW A, @RW0', @RW0+d16	MULUW A, A, @RW0', @RW0+d16	MUL A, @RW0', @RW0+d16	MUL A, A, @RW0', @RW0+d16	MULW A, @RW0', @RW0+d16	MULW A, A, @RW0', @RW0+d16	DIVU A, @RW0', @RW0+d16	DIVU A, A, @RW0', @RW0+d16	DIVUW A, @RW0', @RW0+d16	DIVUW A, A, @RW0', @RW0+d16	DIV A, @RW0', @RW0+d16	DIV A, A, @RW0', @RW0+d16	DIVW A, @RW0', @RW0+d16	DIVW A, A, @RW0', @RW0+d16
+9	MULU A, @RW1', @RW1+d16	MULU A, A, @RW1', @RW1+d16	MULUW A, @RW1', @RW1+d16	MULUW A, A, @RW1', @RW1+d16	MUL A, @RW1', @RW1+d16	MUL A, A, @RW1', @RW1+d16	MULW A, @RW1', @RW1+d16	MULW A, A, @RW1', @RW1+d16	DIVU A, @RW1', @RW1+d16	DIVU A, A, @RW1', @RW1+d16	DIVUW A, @RW1', @RW1+d16	DIVUW A, A, @RW1', @RW1+d16	DIV A, @RW1', @RW1+d16	DIV A, A, @RW1', @RW1+d16	DIVW A, @RW1', @RW1+d16	DIVW A, A, @RW1', @RW1+d16
+A	MULU A, @RW2', @RW2+d16	MULU A, A, @RW2', @RW2+d16	MULUW A, @RW2', @RW2+d16	MULUW A, A, @RW2', @RW2+d16	MUL A, @RW2', @RW2+d16	MUL A, A, @RW2', @RW2+d16	MULW A, @RW2', @RW2+d16	MULW A, A, @RW2', @RW2+d16	DIVU A, @RW2', @RW2+d16	DIVU A, A, @RW2', @RW2+d16	DIVUW A, @RW2', @RW2+d16	DIVUW A, A, @RW2', @RW2+d16	DIV A, @RW2', @RW2+d16	DIV A, A, @RW2', @RW2+d16	DIVW A, @RW2', @RW2+d16	DIVW A, A, @RW2', @RW2+d16
+B	MULU A, @RW3', @RW3+d16	MULU A, A, @RW3', @RW3+d16	MULUW A, @RW3', @RW3+d16	MULUW A, A, @RW3', @RW3+d16	MUL A, @RW3', @RW3+d16	MUL A, A, @RW3', @RW3+d16	MULW A, @RW3', @RW3+d16	MULW A, A, @RW3', @RW3+d16	DIVU A, @RW3', @RW3+d16	DIVU A, A, @RW3', @RW3+d16	DIVUW A, @RW3', @RW3+d16	DIVUW A, A, @RW3', @RW3+d16	DIV A, @RW3', @RW3+d16	DIV A, A, @RW3', @RW3+d16	DIVW A, @RW3', @RW3+d16	DIVW A, A, @RW3', @RW3+d16
+C	MULU A, @RW0+', @RW0+RW7	MULU A, A, @RW0+', @RW0+RW7	MULUW A, @RW0+', @RW0+RW7	MULUW A, A, @RW0+', @RW0+RW7	MUL A, @RW0+', @RW0+RW7	MUL A, A, @RW0+', @RW0+RW7	MULW A, @RW0+', @RW0+RW7	MULW A, A, @RW0+', @RW0+RW7	DIVU A, @RW0+', @RW0+RW7	DIVU A, A, @RW0+', @RW0+RW7	DIVUW A, @RW0+', @RW0+RW7	DIVUW A, A, @RW0+', @RW0+RW7	DIV A, @RW0+', @RW0+RW7	DIV A, A, @RW0+', @RW0+RW7	DIVW A, @RW0+', @RW0+RW7	DIVW A, A, @RW0+', @RW0+RW7
+D	MULU A, @RW1+', @RW1+RW7	MULU A, A, @RW1+', @RW1+RW7	MULUW A, @RW1+', @RW1+RW7	MULUW A, A, @RW1+', @RW1+RW7	MUL A, @RW1+', @RW1+RW7	MUL A, A, @RW1+', @RW1+RW7	MULW A, @RW1+', @RW1+RW7	MULW A, A, @RW1+', @RW1+RW7	DIVU A, @RW1+', @RW1+RW7	DIVU A, A, @RW1+', @RW1+RW7	DIVUW A, @RW1+', @RW1+RW7	DIVUW A, A, @RW1+', @RW1+RW7	DIV A, @RW1+', @RW1+RW7	DIV A, A, @RW1+', @RW1+RW7	DIVW A, @RW1+', @RW1+RW7	DIVW A, A, @RW1+', @RW1+RW7
+E	MULU A, @RW2+', @PC+d16	MULU A, A, @RW2+', @PC+d16	MULUW A, @RW2+', @PC+d16	MULUW A, A, @RW2+', @PC+d16	MUL A, @RW2+', @PC+d16	MUL A, A, @RW2+', @PC+d16	MULW A, @RW2+', @PC+d16	MULW A, A, @RW2+', @PC+d16	DIVU A, @RW2+', @PC+d16	DIVU A, A, @RW2+', @PC+d16	DIVUW A, @RW2+', @PC+d16	DIVUW A, A, @RW2+', @PC+d16	DIV A, @RW2+', @PC+d16	DIV A, A, @RW2+', @PC+d16	DIVW A, @RW2+', @PC+d16	DIVW A, A, @RW2+', @PC+d16
+F	MULU A, @RW3+', addr16	MULU A, A, @RW3+', addr16	MULUW A, @RW3+', addr16	MULUW A, A, @RW3+', addr16	MUL A, @RW3+', addr16	MUL A, A, @RW3+', addr16	MULW A, @RW3+', addr16	MULW A, A, @RW3+', addr16	DIVU A, @RW3+', addr16	DIVU A, A, @RW3+', addr16	DIVUW A, @RW3+', addr16	DIVUW A, A, @RW3+', addr16	DIV A, @RW3+', addr16	DIV A, A, @RW3+', addr16	DIVW A, @RW3+', addr16	DIVW A, A, @RW3+', addr16

表 B.9-15 MOVEA RWi, ea 命令 (第1バイト=79_H)[illegible]

表 B.9-16 MOV Ri, ea 命令 (第1バイト = 7A_H)[illegible]

表 B.9-17 MOVW RWi, ea 命令 (第1バイト=7B_H)[illegible]

表 B.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 B.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, RW2, @RW0-d8, RW2	MOVW RW0, RW3, @RW0-d8, RW3	MOVW RW0, RW4, @RW0-d8, RW4	MOVW RW0, RW5, @RW0-d8, RW5	MOVW RW0, RW6, @RW0-d8, RW6	MOVW RW0, RW7, @RW0-d8, RW7	MOVW RW0, RW4, @RW0-d8, RW4	MOVW RW0, RW5, @RW0-d8, RW5	MOVW RW0, RW6, @RW0-d8, RW6	MOVW RW0, RW7, @RW0-d8, RW7	MOVW RW0, RW8, @RW0-d8, RW8	MOVW RW0, RW9, @RW0-d8, RW9	MOVW RW0, RW10, @RW0-d8, RW10	MOVW RW0, RW11, @RW0-d8, RW11
+1	MOVW RW1, RW0, @RW1-d8, RW0	MOVW RW1, RW1, @RW1-d8, RW1	MOVW RW1, RW2, @RW1-d8, RW2	MOVW RW1, RW3, @RW1-d8, RW3	MOVW RW1, RW4, @RW1-d8, RW4	MOVW RW1, RW5, @RW1-d8, RW5	MOVW RW1, RW6, @RW1-d8, RW6	MOVW RW1, RW7, @RW1-d8, RW7	MOVW RW1, RW4, @RW1-d8, RW4	MOVW RW1, RW5, @RW1-d8, RW5	MOVW RW1, RW6, @RW1-d8, RW6	MOVW RW1, RW7, @RW1-d8, RW7	MOVW RW1, RW8, @RW1-d8, RW8	MOVW RW1, RW9, @RW1-d8, RW9	MOVW RW1, RW10, @RW1-d8, RW10	MOVW RW1, RW11, @RW1-d8, RW11
+2	MOVW RW2, RW0, @RW2-d8, RW0	MOVW RW2, RW1, @RW2-d8, RW1	MOVW RW2, RW2, @RW2-d8, RW2	MOVW RW2, RW3, @RW2-d8, RW3	MOVW RW2, RW4, @RW2-d8, RW4	MOVW RW2, RW5, @RW2-d8, RW5	MOVW RW2, RW6, @RW2-d8, RW6	MOVW RW2, RW7, @RW2-d8, RW7	MOVW RW2, RW4, @RW2-d8, RW4	MOVW RW2, RW5, @RW2-d8, RW5	MOVW RW2, RW6, @RW2-d8, RW6	MOVW RW2, RW7, @RW2-d8, RW7	MOVW RW2, RW8, @RW2-d8, RW8	MOVW RW2, RW9, @RW2-d8, RW9	MOVW RW2, RW10, @RW2-d8, RW10	MOVW RW2, RW11, @RW2-d8, RW11
+3	MOVW RW3, RW0, @RW3-d8, RW0	MOVW RW3, RW1, @RW3-d8, RW1	MOVW RW3, RW2, @RW3-d8, RW2	MOVW RW3, RW3, @RW3-d8, RW3	MOVW RW3, RW4, @RW3-d8, RW4	MOVW RW3, RW5, @RW3-d8, RW5	MOVW RW3, RW6, @RW3-d8, RW6	MOVW RW3, RW7, @RW3-d8, RW7	MOVW RW3, RW4, @RW3-d8, RW4	MOVW RW3, RW5, @RW3-d8, RW5	MOVW RW3, RW6, @RW3-d8, RW6	MOVW RW3, RW7, @RW3-d8, RW7	MOVW RW3, RW8, @RW3-d8, RW8	MOVW RW3, RW9, @RW3-d8, RW9	MOVW RW3, RW10, @RW3-d8, RW10	MOVW RW3, RW11, @RW3-d8, RW11
+4	MOVW RW4, RW0, @RW4-d8, RW0	MOVW RW4, RW1, @RW4-d8, RW1	MOVW RW4, RW2, @RW4-d8, RW2	MOVW RW4, RW3, @RW4-d8, RW3	MOVW RW4, RW4, @RW4-d8, RW4	MOVW RW4, RW5, @RW4-d8, RW5	MOVW RW4, RW6, @RW4-d8, RW6	MOVW RW4, RW7, @RW4-d8, RW7	MOVW RW4, RW4, @RW4-d8, RW4	MOVW RW4, RW5, @RW4-d8, RW5	MOVW RW4, RW6, @RW4-d8, RW6	MOVW RW4, RW7, @RW4-d8, RW7	MOVW RW4, RW8, @RW4-d8, RW8	MOVW RW4, RW9, @RW4-d8, RW9	MOVW RW4, RW10, @RW4-d8, RW10	MOVW RW4, RW11, @RW4-d8, RW11
+5	MOVW RW5, RW0, @RW5-d8, RW0	MOVW RW5, RW1, @RW5-d8, RW1	MOVW RW5, RW2, @RW5-d8, RW2	MOVW RW5, RW3, @RW5-d8, RW3	MOVW RW5, RW4, @RW5-d8, RW4	MOVW RW5, RW5, @RW5-d8, RW5	MOVW RW5, RW6, @RW5-d8, RW6	MOVW RW5, RW7, @RW5-d8, RW7	MOVW RW5, RW4, @RW5-d8, RW4	MOVW RW5, RW5, @RW5-d8, RW5	MOVW RW5, RW6, @RW5-d8, RW6	MOVW RW5, RW7, @RW5-d8, RW7	MOVW RW5, RW8, @RW5-d8, RW8	MOVW RW5, RW9, @RW5-d8, RW9	MOVW RW5, RW10, @RW5-d8, RW10	MOVW RW5, RW11, @RW5-d8, RW11
+6	MOVW RW6, RW0, @RW6-d8, RW0	MOVW RW6, RW1, @RW6-d8, RW1	MOVW RW6, RW2, @RW6-d8, RW2	MOVW RW6, RW3, @RW6-d8, RW3	MOVW RW6, RW4, @RW6-d8, RW4	MOVW RW6, RW5, @RW6-d8, RW5	MOVW RW6, RW6, @RW6-d8, RW6	MOVW RW6, RW7, @RW6-d8, RW7	MOVW RW6, RW4, @RW6-d8, RW4	MOVW RW6, RW5, @RW6-d8, RW5	MOVW RW6, RW6, @RW6-d8, RW6	MOVW RW6, RW7, @RW6-d8, RW7	MOVW RW6, RW8, @RW6-d8, RW8	MOVW RW6, RW9, @RW6-d8, RW9	MOVW RW6, RW10, @RW6-d8, RW10	MOVW RW6, RW11, @RW6-d8, RW11
+7	MOVW RW7, RW0, @RW7-d8, RW0	MOVW RW7, RW1, @RW7-d8, RW1	MOVW RW7, RW2, @RW7-d8, RW2	MOVW RW7, RW3, @RW7-d8, RW3	MOVW RW7, RW4, @RW7-d8, RW4	MOVW RW7, RW5, @RW7-d8, RW5	MOVW RW7, RW6, @RW7-d8, RW6	MOVW RW7, RW7, @RW7-d8, RW7	MOVW RW7, RW4, @RW7-d8, RW4	MOVW RW7, RW5, @RW7-d8, RW5	MOVW RW7, RW6, @RW7-d8, RW6	MOVW RW7, RW7, @RW7-d8, RW7	MOVW RW7, RW8, @RW7-d8, RW8	MOVW RW7, RW9, @RW7-d8, RW9	MOVW RW7, RW10, @RW7-d8, RW10	MOVW RW7, RW11, @RW7-d8, RW11
+8	MOVW @RW0, RW0, @RW0-d16, RW0	MOVW @RW0, RW1, +d16, RW1	MOVW @RW0, RW2, +d16, RW2	MOVW @RW0, RW3, +d16, RW3	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW7, +d16, RW7	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW7, +d16, RW7	MOVW @RW0, RW8, +d16, RW8	MOVW @RW0, RW9, +d16, RW9	MOVW @RW0, RW10, +d16, RW10	MOVW @RW0, RW11, +d16, RW11
+9	MOVW @RW1, RW0, @RW1-d16, RW0	MOVW @RW1, RW1, +d16, RW1	MOVW @RW1, RW2, +d16, RW2	MOVW @RW1, RW3, +d16, RW3	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW7, +d16, RW7	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW7, +d16, RW7	MOVW @RW1, RW8, +d16, RW8	MOVW @RW1, RW9, +d16, RW9	MOVW @RW1, RW10, +d16, RW10	MOVW @RW1, RW11, +d16, RW11
+A	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW8, +d16, RW8	MOVW @RW2, RW9, +d16, RW9	MOVW @RW2, RW10, +d16, RW10	MOVW @RW2, RW11, +d16, RW11
+B	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW8, +d16, RW8	MOVW @RW3, RW9, +d16, RW9	MOVW @RW3, RW10, +d16, RW10	MOVW @RW3, RW11, +d16, RW11
+C	MOVW @RW0+, RW0, +RW7, RW0	MOVW @RW0+, RW1, +RW7, RW1	MOVW @RW0+, RW2, +RW7, RW2	MOVW @RW0+, RW3, +RW7, RW3	MOVW @RW0+, RW4, +RW7, RW4	MOVW @RW0+, RW5, +RW7, RW5	MOVW @RW0+, RW6, +RW7, RW6	MOVW @RW0+, RW7, +RW7, RW7	MOVW @RW0+, RW4, +RW7, RW4	MOVW @RW0+, RW5, +RW7, RW5	MOVW @RW0+, RW6, +RW7, RW6	MOVW @RW0+, RW7, +RW7, RW7	MOVW @RW0+, RW8, +RW7, RW8	MOVW @RW0+, RW9, +RW7, RW9	MOVW @RW0+, RW10, +RW7, RW10	MOVW @RW0+, RW11, +RW7, RW11
+D	MOVW @RW1+, RW0, +RW7, RW0	MOVW @RW1+, RW1, +RW7, RW1	MOVW @RW1+, RW2, +RW7, RW2	MOVW @RW1+, RW3, +RW7, RW3	MOVW @RW1+, RW4, +RW7, RW4	MOVW @RW1+, RW5, +RW7, RW5	MOVW @RW1+, RW6, +RW7, RW6	MOVW @RW1+, RW7, +RW7, RW7	MOVW @RW1+, RW4, +RW7, RW4	MOVW @RW1+, RW5, +RW7, RW5	MOVW @RW1+, RW6, +RW7, RW6	MOVW @RW1+, RW7, +RW7, RW7	MOVW @RW1+, RW8, +RW7, RW8	MOVW @RW1+, RW9, +RW7, RW9	MOVW @RW1+, RW10, +RW7, RW10	MOVW @RW1+, RW11, +RW7, RW11
+E	MOVW @RW2+, RW0, +d16, RW0	MOVW @RW2+, RW1, +d16, RW1	MOVW @RW2+, RW2, +d16, RW2	MOVW @RW2+, RW3, +d16, RW3	MOVW @RW2+, RW4, +d16, RW4	MOVW @RW2+, RW5, +d16, RW5	MOVW @RW2+, RW6, +d16, RW6	MOVW @RW2+, RW7, +d16, RW7	MOVW @RW2+, RW4, +d16, RW4	MOVW @RW2+, RW5, +d16, RW5	MOVW @RW2+, RW6, +d16, RW6	MOVW @RW2+, RW7, +d16, RW7	MOVW @RW2+, RW8, +d16, RW8	MOVW @RW2+, RW9, +d16, RW9	MOVW @RW2+, RW10, +d16, RW10	MOVW @RW2+, RW11, +d16, RW11
+F	MOVW @RW3+, RW0, +d16, RW0	MOVW @RW3+, RW1, +d16, RW1	MOVW @RW3+, RW2, +d16, RW2	MOVW @RW3+, RW3, +d16, RW3	MOVW @RW3+, RW4, +d16, RW4	MOVW @RW3+, RW5, +d16, RW5	MOVW @RW3+, RW6, +d16, RW6	MOVW @RW3+, RW7, +d16, RW7	MOVW @RW3+, RW4, +d16, RW4	MOVW @RW3+, RW5, +d16, RW5	MOVW @RW3+, RW6, +d16, RW6	MOVW @RW3+, RW7, +d16, RW7	MOVW @RW3+, RW8, +d16, RW8	MOVW @RW3+, RW9, +d16, RW9	MOVW @RW3+, RW10, +d16, RW10	MOVW @RW3+, RW11, +d16, RW11

表 B.9-20 XCH Ri, ea 命令 (第1バイト = 7EH)

[illegible]

表 B.9-21 XCHW RWi, ea 命令 (第1バイト = 7FH)

[illegible]

付録 C OTPROM 書込み

MB90P553A の OTPROM は , EPROM モードで MBM27C1000A 相当の機能になります。
専用アダプタソケットを使用することにより , 汎用の EPROM ライタで書込みが可能です。ただし , エレクトロニックシグネチャ (デバイス識別コード) モードには対応しておりません。

■ 専用アダプタソケット

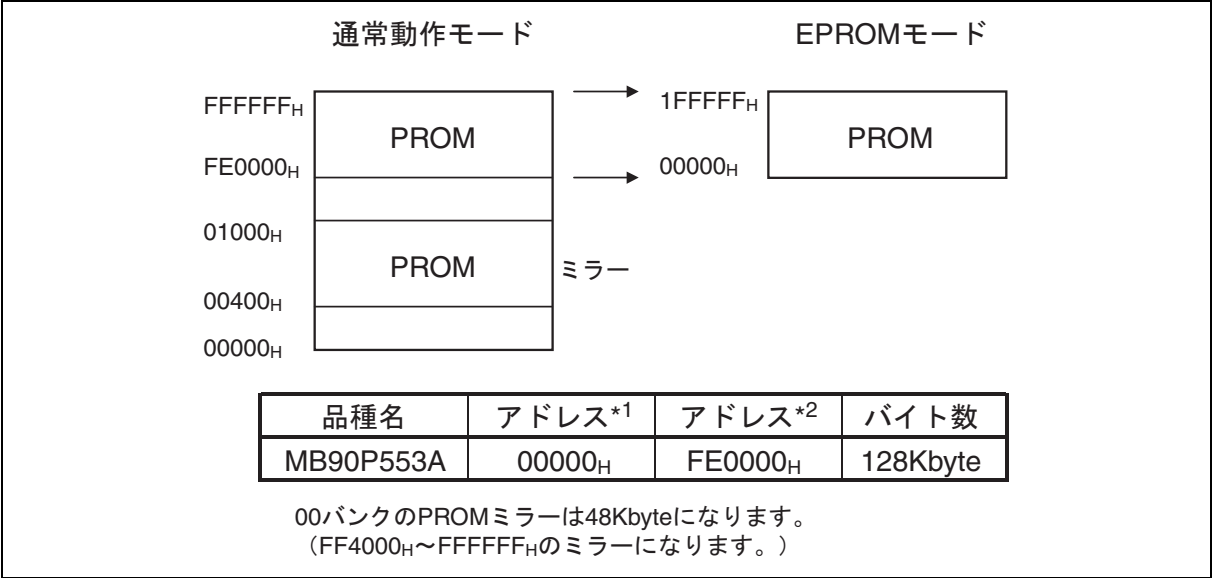
表 C-1 OTPROM 書込み用専用アダプタソケット

パッケージ名称	適合アダプタ型格	供給メーカー
QFP-100	ROM-100QF-32DP-16L	サンハヤト株式会社

■ OTPROM の書込み手順

- 1) EPROM ライタを MBM27C1000A に設定します。
- 2) プログラムデータを EPROM ライタのアドレス ^{*1} ~ 1FFFF_H にロードします
(動作モードでの ROM アドレス ^{*2} ~ FFFFFFF_H が , EPROM 書込みモードでは , アドレス ^{*1} ~ 1FFFF_H に対応します)。
EPROM モードでのメモリ空間を , 図 C-1 に示します。

図 C-1 EPROM モードでのメモリ空間



- 3) アダプタソケットに MB90P553A をセットし , アダプタソケットを EPROM ライタに装着します。このとき , デバイスの向き , アダプタソケットの向きに注意してください。
- 4) OTPROM を書き込みます。

< 注意事項 >

- マスク ROM 品 (MB90553A/B, MB90552A/B) には EPROM モードはありませんので , EPROM ライタでの読出しはできません。
- EPROM ライタ購入の際には , 営業担当部門にご相談ください。

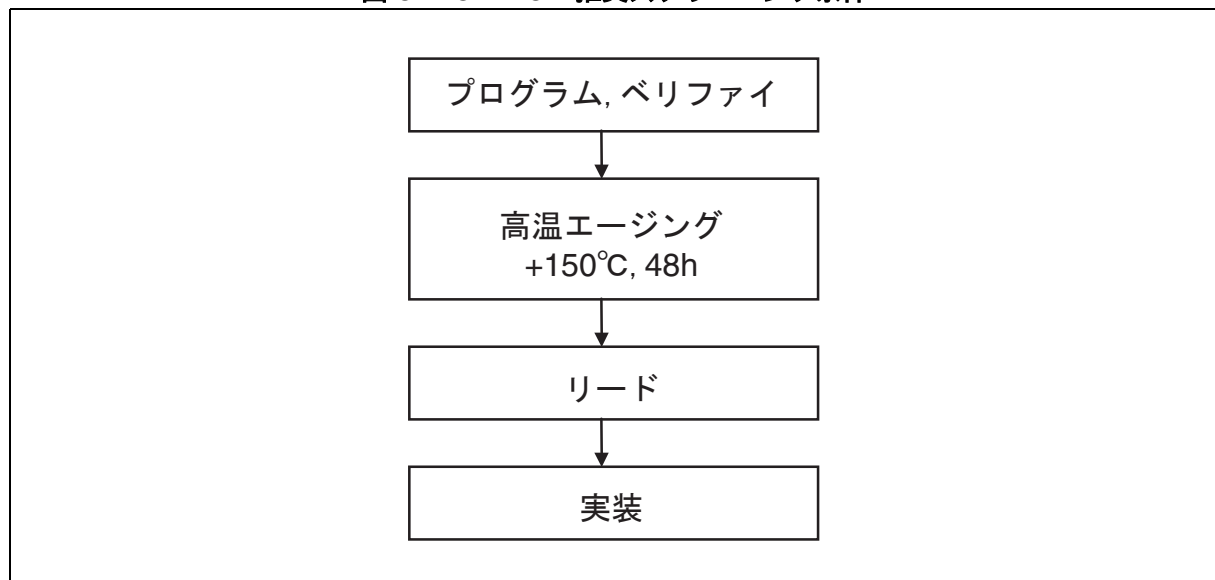
■ プログラムモード

MB90P553A は , 富士通出荷時または消去後すべてのビットが "1" 状態です。情報を導入するには望むビットに "0" を選択的にプログラムします。電氣的に "1" を書き込むことはできません。

■ OTPROM 推奨スクリーニング条件

OTPROM のマイコンプログラム未書込み品は , 実装前のスクリーニング方法として , 高温エージングをお勧めします。

図 C-2 OTPROM 推奨スクリーニング条件



■ OTPROM 書込み歩留まり

OTPROM のマイコンプログラム未書込み品は , その性質上全ビット書込み試験を実施することはできません。したがって , 必ずしも書込み歩留まり 100% は保証できない場合があります。

索引

Numerics

16 ビットアウトプットコンペアのタイミング	
16 ビットアウトプットコンペアのタイミング	175
16 ビットアウトプットコンペアの動作	
16 ビットアウトプットコンペアの動作	174
16 ビットインプットキャプチャ	
16 ビットインプットキャプチャの動作	176
16 ビットタイマレジスタ	
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)	186
16 ビットフリーランタイム	
16 ビットフリーランタイム (× 1)	156
16 ビットフリーランタイムのカウントタイミング	173
16 ビットフリーランタイムの動作	
16 ビットフリーランタイムの動作	172
16 ビットリロードタイム	
16 ビットリロードタイム (イベントカウント機能付) のブロックダイアグラム	180
16 ビットリロードタイム (イベントカウント機能付) のレジスタ	181
16 ビットリロードタイム (イベントカウント機能付) の概要	180
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)	186
16 ビット入出力タイム	
16 ビット入出力タイムのブロックダイアグラム	158
16 ビット入出力タイムのレジスタ	159
1M ビットフラッシュメモリ	
1M ビットフラッシュメモリのセクタ構成	342
1M ビットフラッシュメモリのプログラム例	363
1M ビットフラッシュメモリの特長	340
8/16 ビット PPG	
8/16 ビット PPG のレジスタ	197
8/16 ビット PPG の概要	194
8/16 ビット PPG の割込み	206
8/16 ビット PPG の動作	206
8/16 ビット PPG の動作モード	208
8 ビット PPG	
8 ビット PPG のブロックダイアグラム	195

A

A	
アキュムレータ (A)	31
A/D コンバータ	
A/D コンバータのブロックダイアグラム	231
A/D コンバータのレジスタ	232
A/D コンバータの概要	230
A/D コンバータ使用上の注意	231
ADB	
アディショナルデータバンクレジスタ (ADB)	40
アディショナルデータバンクレジスタ (ADB) リセット時初期値 00H	26
ADCR1, ADCR0	
データレジスタ (ADCR1, ADCR0)	238
ADCS0, ADCS1	
コントロールステータスレジスタ (ADCS0, ADCS1)	233
ADER	
アナログ入力許可レジスタ (ADER)	142
ARSR	
自動レディ機能選択レジスタ (ARSR)	120

B

BAP	
バッファアドレスポインタ (BAP)	73

C

C	
キャリーフラグ (C)	36
CCR	
コンディションコードレジスタ (CCR)	35
CDCR	
通信プリスケールレジスタ (CDCR)	252
CKSCR	
クロック選択レジスタ (CKSCR)	93
CLKR	
クロック出力許可レジスタ (CLKR)	323
CMR	
コモンレジスタバンクプリフィックス (CMR)	44
CPU 間欠動作機能	
CPU 間欠動作機能	106

D

DCT	
データカウンタ (DCT)	71
DDR _x	
ポート方向レジスタ (DDR _x)	139

索引

DPR	
ダイレクトページレジスタ (DPR)	39
DQ3	
セクタ消去タイマフラグ (DQ3)	353
DQ5	
タイミングリミット超過フラグ (DQ5)	352
DQ6	
トグルビットフラグ (DQ6)	351
DQ7	
データボーリングフラグ (DQ7)	350
DTB	
データバンクレジスタ (DTB)	40
データバンクレジスタ (DTB) リセット時初期値 00H	26
DTP	
DTP の動作	219
外部割込み /DTP の動作手順	222
DTP/ 外部割込み	
DTP/ 外部割込みのブロックダイヤグラム	216
DTP/ 外部割込みのレジスタ	216
DTP 要求	
外部割込み要求と DTP 要求の切替え	220
E	
ECSR	
バス制御信号選択レジスタ (ECSR)	123
EI ² OS	
EI ² OS を使った変換動作	241
拡張 I/O シリアル 2 で EI ² OS 機能を使用した場合 の注意事項	53
拡張インテリジェント I/O サービス (EI ² OS)	269
拡張インテリジェント I/O サービス (EI ² OS) の構 造	66
拡張インテリジェント I/O サービス (EI ² OS) の実 行時間	76
拡張インテリジェント I/O サービス (EI ² OS) の 概要	65
拡張インテリジェント I/O サービス (EI ² OS) の 動作フロー	74
単発モード時の EI ² OS の起動例	242
停止モード時の EI ² OS の起動例	246
連続モード時の EI ² OS の起動例	244
EI ² OS ステータスレジスタ	
EI ² OS ステータスレジスタ (ISCS)	71
EIRR	
割込み /DTP 要因レジスタ (EIRR)	217
ELVR	
要求レベル設定レジスタ (ELVR)	218
ENIR	
割込み /DTP 許可レジスタ (ENIR)	217
F	
F ² MC-16LX	
F ² MC-16LX 命令一覧表	413
FMCS	
コントロールステータスレジスタ (FMCS)	345
FPT-100P-M05	
FPT-100P-M05 パッケージ外形寸法図	7
FPT-100P-M06	
FPT-100P-M06 パッケージ外形寸法図	6
FRE	
5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割込み要因	276
FTP-100P-M05	
FTP-100P-M05 端子配列図	9
FTP-100P-M06	
FTP-100P-M06 端子配列図	8
H	
HACR	
外部アドレス出力制御レジスタ (HACR)	122
I	
I	
割込み許可フラグ (I)	35
I/O ポート	
I/O ポートのブロックダイヤグラム	133
I/O ポートのレジスタ	135
I/O ポートの概要	132
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)	71
I/O 拡張シリアルインタフェース	
I/O 拡張シリアルインタフェースのブロックダイ ヤグラム	283
I/O 拡張シリアルインタフェースのレジスタ	284
I/O 拡張シリアルインタフェースの概要	282
I/O 拡張シリアルインタフェースの割込み機能	298
I/O 拡張シリアルインタフェースの動作	290
I ² C インタフェース	
I ² C インタフェースのブロックダイヤグラム	301
I ² C インタフェースのモードフロー	319
I ² C インタフェースのレジスタ	303
I ² C インタフェースの構成図	302
I ² C インタフェースの転送フロー	317
I ² C インタフェースの特長	300
IADR	
アドレスレジスタ (IADR)	312
IBCR	
バスコントロールレジスタ (IBCR)	307
IBSR	
バスステータスレジスタ (IBSR)	304
ICCR	
クロックコントロールレジスタ (ICCR)	310
ICCS	
コントロールステータスレジスタ (ICCS)	162
ICR	
割込み制御レジスタ (ICR)	67
ICS23/ICS01	
コントロールステータスレジスタ (ICS23/ICS01)	170

IDAR	
データレジスタ (IDAR)	313
ILM	
インタラプトレベルマスクレジスタ (ILM)	36
INT	
SCC , MSS , INT ビットの競合についての注意	309
IOA	
I/O レジスタアドレスポインタ (IOA)	71
IPCO0 ~ IPCO3	
インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3)	169
ISCS	
EI ² OS ステータスレジスタ (ISCS)	71
ISD	
拡張インテリジェント I/O サービスディスクリプタ (ISD)	70
ISEL	
ポート選択レジスタ (ISEL)	314
L	
LPMCR	
低消費電力モード制御レジスタ (LPMCR)	91
M	
MSS	
SCC , MSS , INT ビットの競合についての注意	309
N	
N	
ネガティブフラグ (N)	35
NCC	
フラグ変化抑止プリフィックス (NCC)	44
O	
OCCP0/OCCP1	
コンペアレジスタ (OCCP0/OCCP1)	165
OCS0 ~ OCS2	
コントロールステータスレジスタ (OCS0 ~ OCS2)	166
ODR4	
出力端子レジスタ (ODR4)	140
ORE	
5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割込み要因	276
OTPROM	
OTPROM の書込み手順	449
OTPROM 書込み歩留まり	450
OTPROM 推奨スクリーニング条件	450

P	
PACSR	
プログラムアドレス検出コントロールステータスレジスタ (PACSR)	328
PADR0/PADR1	
プログラムアドレス検出レジスタ (PADR0/PADR1)	327
PC	
プログラムカウンタ (PC)	38
PCB	
プログラムバンクレジスタ (PCB)	40
プログラムバンクレジスタ (PCB) リセット時初期値 FFH	26
PDRx	
ポートデータレジスタ (PDRx)	137
PE	
5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割込み要因	276
PLL スリープモード	
PLL スリープモード / メインスリープモード	88
PPG0/1 出力端子制御レジスタ	
PPG0/1 出力端子制御レジスタ (PPGE)	203
PPG0 動作モード制御レジスタ	
PPG0 動作モード制御レジスタ (PPGC0)	198
PPG1 動作モード制御レジスタ	
PPG1 動作モード制御レジスタ (PPGC1)	200
PPGC0	
PPG0 動作モード制御レジスタ (PPGC0)	198
PPGC1	
PPG1 動作モード制御レジスタ (PPGC1)	200
PPGE	
PPG0/1 出力端子制御レジスタ (PPGE)	203
PPG 出力動作	
PPG 出力動作	209
PRL/PRLH	
リロードレジスタ (PRL/PRLH)	205
PS	
プロセッサステータス (PS)	35
R	
RDR0, RDR1	
入力抵抗レジスタ (RDR0, RDR1)	141
RDRF	
5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割込み要因	276
ROMM	
ROM ミラー機能選択レジスタ (ROMM)	337
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ヤグラム	336
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)	337
RP	
レジスタバンクポインタ (RP)	36
S	
S	
スタックフラグ (S)	35

索引

SCC	
SCC , MSS , INT ビットの競合についての注意	309
SCR	
シリアルコントロールレジスタ (SCR)	262
SDR	
シリアルシフトデータレジスタ (SDR)	289
SIDR	
シリアルインプットデータレジスタ (SIDR)/ シリアルアウトプットデータレジスタ (SODR)	265
SMCS	
シリアルモードコントロールステータスレジスタ (SMCS)	285
SMR	
シリアルモードレジスタ (SMR)	259
SODR	
シリアルインプットデータレジスタ (SIDR)/ シリアルアウトプットデータレジスタ (SODR)	265
SSB	
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)	40
ユーザスタックバンクレジスタ (USB) リセット時初期値 00H/ システムスタックバンクレジスタ (SSB) リセット時初期値 00H	26
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)	33
SSR	
シリアルステータスレジスタ (SSR)	266
STOP 状態	
STOP 状態	293

T

T	
スティッキビットフラグ (T)	35
TBTC	
タイムベースタイマ制御レジスタ (TBTC)	145
TDRE	
5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割込み要因	276
TMCSR	
タイマコントロールステータスレジスタ (TMCSR)	182
TMR	
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)	186
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)	186

U

UART	
UART(モード 1 使用時) の応用例	279
UART の動作	269
UART の特長	256
UART のブロックダイアグラム	257

UART のレジスタ	258
USB	
ユーザスタックバンクレジスタ (USB) リセット時初期値 00H/ システムスタックバンクレジスタ (SSB) リセット時初期値 00H	26
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)	40
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)	33

V

V	
オーバフローフラグ (V)	36

W

WDTC	
ウォッチドッグタイマ制御レジスタ (WDTC)	151

Z

Z	
ゼロフラグ (Z)	36

あ

アービトレーション	
アービトレーション	316
アウトプットコンペア	
アウトプットコンペア (× 4)	156
アキュムレータ	
アキュムレータ (A)	31
アクセスモード	
アクセスモード	112
アクノリッジ	
アクノリッジ	316
アディショナルデータバンクレジスタ	
アディショナルデータバンクレジスタ (ADB)	40
アディショナルバンクレジスタ	
アディショナルバンクレジスタ (ADB) リセット 時初期値 00H	26
アドレス指定	
バンク方式によるアドレス指定	26
リニア方式によるアドレス指定	25
アドレスレジスタ	
アドレスレジスタ (IADR)	312
アドレス一致検出機能	
アドレス一致検出機能のブロックダイアグラム	326
アドレス一致検出機能の使用例	330
アドレス一致検出機能の動作	329
アドレス一致検出機能動作上の注意	329
アドレッシング	
アドレッシング	315, 390
間接アドレッシング	399
直接アドレッシング	392
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)	142
アンドフロー動作	
アンドフロー動作	188

い

インターバル割込み機能	
インターバル割込み機能	147
インタラプトレベルマスクレジスタ	
インタラプトレベルマスクレジスタ (ILM)	36
インプットキャプチャ	
インプットキャプチャ (× 4)	157
インプットキャプチャ入力タイミング	177
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 (IPCO0 ~ IPCO3)	169

う

ウォッチドッグ	
ウォッチドッグ停止	153
ウォッチドッグタイマ	
ウォッチドッグタイマの起動方法	153
ウォッチドッグタイマのクリア	154
ウォッチドッグタイマのブロックダイアグラム	150
ウォッチドッグタイマのレジスタ一覧	150

ウォッチドッグタイマリセット	
ウォッチドッグタイマリセットの阻止	153
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)	151

お

オーバフローフラグ	
オーバフローフラグ (V)	36

か

カウンタ	
カウンタの動作状態	191
カウントクロック	
カウントクロックの選択に関する注意事項	211
間接アドレッシング	
間接アドレッシング	399

き

キャリーフラグ	
キャリーフラグ (C)	36

く

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)	310
クロックモニタ機能	
クロックモニタ機能のブロックダイアグラム	322
クロック供給マップ	
クロック供給マップ	81
クロック出力許可レジスタ	
クロック出力許可レジスタ (CLKR)	323
クロック選択レジスタ	
クロック選択レジスタ (CKSCR)	93
クロック発生部	
クロック発生部に関する注意事項	80

こ

コマンドシーケンス	
コマンドシーケンス表	347
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)	44
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)	35
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 (ADCS0,ADCS1)	233
コントロールステータスレジスタ (FMCS)	345
コントロールステータスレジスタ (ICCS)	162
コントロールステータスレジスタ (ICS23/ICS01)	170
コントロールステータスレジスタ (OCS0 ~ OCS2)	166

コンペアレジスタ	
コンペアレジスタ (OCCP0/OCCP1)	165

し

システムスタックバンクレジスタ	
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)	40
ユーザスタックバンクレジスタ (USB) リセット時初期値 00H/ システムスタックバンクレジスタ (SSB) リセット時初期値 00H	26
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)	33
実効アドレス	
実効アドレスフィールド	408
実効アドレスフィールド	
実効アドレスフィールド	391
実行サイクル	
実行サイクル数	406
実行サイクル数計算方法	406
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	295
シリアルアウトデータレジスタ	
シリアルインプットデータレジスタ (SIDR)/ シリアルアウトデータレジスタ (SODR)	265
シリアルインプットデータレジスタ	
シリアルインプットデータレジスタ (SIDR)/ シリアルアウトデータレジスタ (SODR)	265
シリアル書込み接続	
MB90F553A シリアル書込み接続の基本構成	368
シリアル書込み接続例	
シリアル書込み接続例 (ユーザ電源使用时)	372
シリアル書込み接続例 (ライタから電源供給時)	374
シリアルクロック入力周波数	
発振クロック周波数とシリアルクロック入力周波数	371
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)	262
シリアルシフトデータレジスタ	
シリアルシフトデータレジスタ (SDR)	289
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)	266
シリアルデータレジスタ R/W 待機状態	
シリアルデータレジスタ R/W 待機状態	293
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 (SMCS)	285
シリアルモードレジスタ	
シリアルモードレジスタ (SMR)	259
シングルチップモード	
シングルチップモード時の各端子状態	103

す

スタートコンディション	
スタートコンディション	315
スタックフラグ	
スタックフラグ (S)	35
スタンバイ	
スタンバイからの復帰	222
スティッキビットフラグ	
スティッキビットフラグ (T)	35
ストップコンディション	
ストップコンディション	315
ストップモード	
ストップモード/ハードウェアスタンバイモード	88
ストップモードの解除	101
ストップモードへの遷移	101
スリープモード	
スリープモードの解除	98
スリープモードへの遷移	98

せ

セクタ構成	
1M ビットフラッシュメモリのセクタ構成	342
セクタ消去	
フラッシュメモリのセクタ消去を一時停止する	361
フラッシュメモリのセクタ消去を再開する	362
フラッシュメモリのセクタ消去手順	359
フラッシュメモリの任意のデータを消去する (セクタ消去)	359
セクタ消去タイマフラグ	
セクタ消去タイマフラグ (DQ3)	353
ゼロフラグ	
ゼロフラグ (Z)	36

そ

ソフトウェア割込み	
ソフトウェア割込みに関する注意事項	64
ソフトウェア割込みの概要	63
ソフトウェア割込みの構造	63
ソフトウェア割込みの動作	64

た

タイマコントロールステータスレジスタ	
タイマコントロールステータスレジスタ (TMCSR)	182
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)	352
タイムベースタイマ	
タイムベースタイマのブロックダイヤグラム	144
タイムベースタイマのレジスター一覧	144
タイムベースタイマの動作	147

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)	145
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)	39

ち

チップ消去	
フラッシュメモリのデータを消去する (チップ消去)	358
直接アドレッシング	
直接アドレッシング	392

て

データアクセス	
各バンクの設定とデータアクセス	40
データカウンタ	
データカウンタ (DCT)	71
データバンクレジスタ	
データバンクレジスタ (DTB)	40
データバンクレジスタ (DTB) リセット時初期値 00H	26
データポーリングフラグ	
データポーリングフラグ (DQ7)	350
データレジスタ	
データレジスタ	161
データレジスタ (ADCR1, ADCR0)	238
データレジスタ (IDAR)	313

と

トグルビットフラグ	
トグルビットフラグ (DQ6)	351

ね

ネガティブフラグ	
ネガティブフラグ (N)	35

は

ハードウェア	
各ハードウェアの初期値	207
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	348
ハードウェアスタンバイモード	
ストップモード / ハードウェアスタンバイモード	88
ハードウェアスタンバイモードの解除	102
ハードウェアスタンバイモードへの遷移	102
ハードウェア割込み	
ハードウェア割込みの概要	56
ハードウェア割込みの使用上の注意	58
ハードウェア割込みの動作	59
ハードウェア割込みの動作フロー	61
ハードウェア割込み使用手順例	62
ハードウェア割込み処理時間	60
ハードウェア割込みの構造	
ハードウェア割込みの構造	56

ハードウェア割込み要求	
内蔵リソース領域への書込み中のハードウェア割込み要求	57
バスエラー	
バスエラー	316
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)	307
バスステータスレジスタ	
バスステータスレジスタ (IBSR)	304
バスモード	
バスモード	112
バスモード別メモリ空間	115
バスモード別メモリ空間推奨設定例	117
バス制御信号選択レジスタ	
バス制御信号選択レジスタ (ECSR)	123
パッケージ外形寸法図	
FPT-100P-M05 パッケージ外形寸法図	7
FPT-100P-M06 パッケージ外形寸法図	6
発振クロック周波数	
発振クロック周波数とシリアルクロック入力周波数	371
パッチ処理	
プログラムパッチ処理フロー	333
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)	73
パルス	
パルスの端子出力の制御	212
パルス幅	
リロード値とパルス幅の関係	210
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	43
バンク方式	
バンク方式によるアドレス指定	26

ふ

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)	44
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最小限の接続例 (ユーザ電源使用時)	376
フラッシュマイコンプログラマとの最小限の接続例 (ライタから電源供給時)	378
フラッシュメモリ	
フラッシュメモリのセクタ消去を一時停止する	361
フラッシュメモリのセクタ消去を再開する	362
フラッシュメモリのセクタ消去手順	359
フラッシュメモリのデータを消去する (チップ消去)	358
フラッシュメモリのレジスタ	340
フラッシュメモリの制御信号	343
フラッシュメモリの任意のデータを消去する (セクタ消去)	359
フラッシュメモリへデータを書き込む	356
フラッシュメモリを読み出し / リセット状態にする	355
フラッシュメモリ書込み / 消去の詳細説明	354
フラッシュメモリ書込み / 消去の方法	340
フラッシュメモリ書込み手順	356

フラッシュメモリ全体のブロックダイアグラム	341
フラッシュメモリモード	
フラッシュメモリモード	343
ブリフィックスコード	
ブリフィックスコードが連続している場合	46
ブリフィックス命令	
割込み抑止命令とブリフィックス命令に関する制約	46
プログラム	
プログラムパッチ処理フロー	333
プログラムパッチ処理例	332
プログラムアドレス検出コントロールステータスレジスタ	
プログラムアドレス検出コントロールステータスレジスタ (PACSR)	328
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)	327
プログラムカウンタ	
プログラムカウンタ (PC)	38
プログラムパッチ処理例	
プログラムパッチ処理例	332
プログラムバンクレジスタ	
プログラムバンクレジスタ (PCB)	40
プログラムバンクレジスタ (PCB) リセット時初期値 FFH	26
プログラムモード	
プログラムモード	450
プロセッサステータス	
プロセッサステータス (PS)	35
ブロックダイアグラム	
16 ビットリロードタイマ (イベントカウント機能付) のブロックダイアグラム	180
16 ビット入出力タイマのブロックダイアグラム	158
8 ビット PPG のブロックダイアグラム	195
A/D コンバータのブロックダイアグラム	231
DTP/ 外部割込みのブロックダイアグラム	216
I/O ポートのブロックダイアグラム	133
I/O 拡張シリアルインタフェースのブロックダイアグラム	283
I ² C インタフェースのブロックダイアグラム	301
ROM ミラー機能選択モジュールのブロックダイアグラム	336
UART のブロックダイアグラム	257
アドレス一致検出機能のブロックダイアグラム	326
ウォッチドッグタイマのブロックダイアグラム	150
クロックモニタ機能のブロックダイアグラム	322
タイムベースタイマのブロックダイアグラム	144
フラッシュメモリ全体のブロックダイアグラム	341
ブロックダイアグラム	5
外部メモリアクセス (外部バス端子制御回路) のブロックダイアグラム	118

遅延割込み発生モジュールのブロックダイアグラム	226
低消費電力回路のブロックダイアグラム	90

へ

ペリフェラル	
DTP を用いたときの外部に接続するペリフェラルの条件	222

ほ

ポートデータレジスタ	
ポートデータレジスタ (PDRx)	137
ポート選択レジスタ	
ポート選択レジスタ (ISEL)	314
ポート方向レジスタ	
ポート方向レジスタ (DDRx)	139
ホールド機能	
ホールド機能	130

ま

マシクロック	
マシクロックの初期化	108
マシクロック切換え	108

め

命令	
命令の種類	389
命令マップの構造	427
命令一覧表	
F ² MC-16LX 命令一覧表	413
メインクロックモード	
メインクロックモード / メインスリープモード	88
メインスリープモード	
PLL スリープモード / メインスリープモード	88
メインクロックモード / メインスリープモード	88
メモリアクセスモード	
メモリアクセスモードの概要	112
メモリ空間	
バスモード別メモリ空間	115
バスモード別メモリ空間推奨設定例	117
メモリ空間	24
メモリ空間における多バイト長データの配置	28

も

モードデータ	
モードデータ	114
モード 1	
UART (モード 1 使用時) の応用例	279
モード端子	
モード端子	113

ゆ

ユーザスタックバンクレジスタ

ユーザスタックバンクレジスタ (USB) リセット時 初期値 00H/システムスタックバンクレジスタ (SSB) リセット時初期値 00H	26
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)	40

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)	33
---	----

ユーザ電源

シリアル書き込み接続例 (ユーザ電源使用時)	372
フラッシュマイコンプログラマとの最小限の接続 例 (ユーザ電源使用時)	376

ら

ライタから電源

シリアル書き込み接続例 (ライタから電源供給時)	374
フラッシュマイコンプログラマとの最小限の接続 例 (ライタから電源供給時)	378

り

リセット

リセット解除後の動作	84
------------------	----

リセット入力

リセット入力によって初期化されないレジスタ	85
--------------------------------	----

リセット要因

リセット要因	82
--------------	----

リニア方式

リニア方式によるアドレス指定	25
----------------------	----

リロードレジスタ

リロードレジスタ (PRL/PRH)	205
リロードレジスタへのライトタイミング	213

リロード値

リロード値とパルス幅の関係	210
---------------------	-----

れ

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)	186
16 ビットリロードタイマ (イベントカウント機 能付) のレジスタ	181
16 ビット入出力タイマのレジスタ	159
8/16 ビット PPG のレジスタ	197
A/D コンバータのレジスタ	232
DTP/ 外部割込みのレジスタ	216
ET ² OS ステータスレジスタ (ISCS)	71
I/O ポートのレジスタ	135
I/O 拡張シリアルインタフェースのレジスタ	284
I ² C インタフェースのレジスタ	303
PPG0/1 出力端子制御レジスタ (PPGE)	203
PPG0 動作モード制御レジスタ (PPGC0)	198
PPG1 動作モード制御レジスタ (PPGC1)	200

ROM ミラー機能選択レジスタ (ROMM)	337
UART のレジスタ	258
アディショナルデータバンクレジスタ (ADB)	40
アドレスレジスタ (IADR)	312
アナログ入力許可レジスタ (ADER)	142
インタラプトレベルマスクレジスタ (ILM)	36
インプットキャプチャデータレジスタ (IPCO0 ~ IPCO3)	169
ウォッチドッグタイマのレジスタ一覧	150
ウォッチドッグタイマ制御レジスタ (WDTC)	151
クロックコントロールレジスタ (ICCR)	310
クロック出力許可レジスタ (CLKR)	323
クロック選択レジスタ (CKSCR)	93
コンディショニングコードレジスタ (CCR)	35
コントロールステータスレジスタ (ADCS0, ADCS1)	233
コントロールステータスレジスタ (FMCS)	345
コントロールステータスレジスタ (ICCS)	162
コントロールステータスレジスタ (ICS23/ICS01)	170
コントロールステータスレジスタ (OCS0 ~ OCS2)	166
コンペアレジスタ (OCCP0/OCCP1)	165
シリアルインプットデータレジスタ (SIDR)/ シリ アルアウトプットデータレジスタ (SODR)	265
シリアルコントロールレジスタ (SCR)	262
シリアルシフトデータレジスタ (SDR)	289
シリアルステータスレジスタ (SSR)	266
シリアルモードコントロールステータスレジスタ (SMCS)	285
シリアルモードレジスタ (SMR)	259
タイマコントロールステータスレジスタ (TMCSR)	182
タイムベースタイマのレジスタ一覧	144
タイムベースタイマ制御レジスタ (TBTC)	145
ダイレクトページレジスタ (DPR)	39
データバンクレジスタ (DTB)	40
データレジスタ	161
データレジスタ (ADCR1, ADCR0)	238
データレジスタ (IDAR)	313
バスコントロールレジスタ (IBCR)	307
バスステータスレジスタ (IBSR)	304
バス制御信号選択レジスタ (ECSR)	123
フラッシュメモリのレジスタ	340
プログラムアドレス検出コントロールステータ スレジスタ (PACSR)	328
プログラムアドレス検出レジスタ (PADR0/ PADR1)	327
プログラムバンクレジスタ (PCB)	40
ポートデータレジスタ (PDRx)	137
ポート選択レジスタ (ISEL)	314
ポート方向レジスタ (DDRx)	139
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)	40
リセット入力によって初期化されないレジスタ	85
リロードレジスタ (PRL/PRH)	205
レジスタバンク	42

レジスタバンクポインタ (RP)	36
外部アドレス出力制御レジスタ (HACR)	122
外部メモリアクセス (外部バス端子制御回路) の レジスタ	119
割り込み /DTP 許可レジスタ (ENIR)	217
割り込み /DTP 要因レジスタ (EIRR)	217
割り込み制御レジスタ (ICR)	67
自動レディ機能選択レジスタ (ARSR)	120
出力端子レジスタ (ODR4)	140
専用レジスタ	29
遅延割り込み発生モジュールのレジスタ	226
通信プリスケアラレジスタ (CDCR)	252
低消費電力モード制御レジスタ (LPMCR)	91
入力抵抗レジスタ (RDR0, RDR1)	141
汎用レジスタ	41
要求レベル設定レジスタ (ELVR)	218
レジスタバンク レジスタバンク	42
レジスタバンクポインタ レジスタバンクポインタ (RP)	36
レディ機能 レディ機能	128
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)	122
外部イベントカウンタ 外部イベントカウンタ	187
外部クロック 外部クロック	271
外部シフトクロックモード 外部シフトクロックモード	292
外部データバス 16 ビットモード 外部データバス 16 ビットモード時の各端子状態	104
外部データバス 8 ビットモード 外部データバス 8 ビットモード時の各端子状態	105
外部バス端子制御回路 外部メモリアクセス (外部バス端子制御回路)	118
外部メモリアクセス (外部バス端子制御回路) の ブロックダイアグラム	118
外部メモリアクセス (外部バス端子制御回路) の レジスタ	119
外部メモリアクセス 外部メモリアクセス (外部バス端子制御回路)	118
外部メモリアクセス (外部バス端子制御回路) の ブロックダイアグラム	118
外部メモリアクセス (外部バス端子制御回路) の レジスタ	119
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	126
外部割り込み 外部割り込み /DTP の動作手順	222
外部割り込みの動作	219
外部割り込み要求 外部割り込み要求と DTP 要求の切替え	220
外部割り込み要求レベル 外部割り込み要求レベル	223
拡張 I/O シリアル 2 拡張 I/O シリアル 2 で EI ² OS 機能を使用した場合 の注意事項	53
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)	269
拡張インテリジェント I/O サービス (EI ² OS) の構 造	66
拡張インテリジェント I/O サービス (EI ² OS) の実 行時間	76
拡張インテリジェント I/O サービス (EI ² OS) の 概要	65
拡張インテリジェント I/O サービス (EI ² OS) の 動作フロー	74
拡張インテリジェント I/O サービス (EI ² OS) 機能 拡張インテリジェント I/O サービス (EI ² OS) 機能 と割り込み	188
拡張インテリジェント I/O サービスディスクリプタ 拡張インテリジェント I/O サービスディスクリプ タ (ISD)	70
割り込み 5 つのフラグ (PE/ORE/FRE/RDRF/TDRE) と 2 つの割り込み要因	276
各動作モードによる割り込みフラグのセットタイミ ング	276
拡張インテリジェント I/O サービス (EI ² OS) 機能 と割り込み	188
割り込みの概要	50
割り込み時におけるスタック内へのレジスタの回避	57
割り込み /DTP 許可レジスタ 割り込み /DTP 許可レジスタ (ENIR)	217
割り込み /DTP 要因レジスタ 割り込み /DTP 要因レジスタ (EIRR)	217
割り込みハンドリング時間 割り込みハンドリング時間 (割り込み処理準備に要す る時間)	60
割り込みベクタ 割り込みベクタ	54
割り込み許可フラグ 割り込み許可フラグ (I)	35
割り込み制御レジスタ 割り込み制御レジスタ (ICR)	67
割り込み要因 割り込み要因	51
割り込み要求サンプル待ち時間 割り込み要求サンプル待ち時間	60
割り込み抑止命令 割り込み抑止命令	46, 57
割り込み抑止命令とプリフィックス命令に関する制 約	46
使用上の注意 「DIV A,Ri」, 「DIVW A,RWi」命令の使用上の注 意	47
時計モード 時計モード	88
時計モードの解除	99
時計モードへの遷移	99
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)	120
受信動作 受信動作	272

出力端子レジスタ		
出力端子レジスタ (ODR4)	140	
出力端子機能		
出力端子機能	190	
初期化		
初期化	275	
初期値		
各ハードウェアの初期値	207	
専用アダプタソケット		
専用アダプタソケット	449	
専用レジスタ		
専用レジスタ	29	
送信動作		
送信動作	273	
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	28	
多バイト長データのアクセス	28	
多重割込み		
多重割込み	57	
単発モード		
単発モード	240	
単発モード時の EI ² OS の起動例	242	
端子機能説明		
端子機能説明	10	
端子配列図		
FTP-100P-M05 端子配列図	9	
FTP-100P-M06 端子配列図	8	
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ブロックダイアグラム	226	
遅延割込み発生モジュールのレジスタ	226	
遅延割込み発生モジュールの動作	227	
遅延割込み要求ラッチ		
遅延割込み要求ラッチの使用上の注意	227	
注意		
A/D コンバータ使用上の注意	231	
SCC, MSS, INT ビットの競合についての注意	309	
アドレス一致検出機能動作上の注意	329	
カウントクロックの選択に関する注意事項	211	
クロック発生部に関する注意事項	80	
ソフトウェア割込みに関する注意事項	64	
デバイスの取扱いに関する注意事項	19	
ハードウェア割込みの使用上の注意	58	
「DIV A.Ri」, 「DIVW A.RWi」命令の使用上の注意	47	
拡張 I/O シリアル 2 で EI ² OS 機能を使用した場合の注意事項	53	
遅延割込み要求ラッチの使用上の注意	227	
注意事項の回避について	48	
通信プリスケアラ		
通信プリスケアラ	270	
通信プリスケアラレジスタ		
通信プリスケアラレジスタ (CDCR)	252	
通信プリスケアラレジスタの動作	253	
通信開始		
通信開始	275	
通信終了		
通信終了	275	
低消費電力モード制御レジスタ		
低消費電力モード制御レジスタ (LPMCR)	91	
低消費電力回路		
低消費電力回路のブロックダイアグラム	90	
低消費電力回路の概要	88	
低消費電力制御回路		
低消費電力制御回路の動作	96	
停止モード		
停止モード	241	
停止モード時の EI ² OS の起動例	246	
停止状態		
停止状態	293	
転送データフォーマット		
転送データフォーマット	272, 274	
転送状態		
転送状態	294	
動作モード		
動作モード	112	
特長		
MB90550A/B シリーズの特長	2	
内部クロック		
内部クロック動作	187	
内部クロックモード		
入力端子機能 (内部クロックモード時)	189	
内部シフトクロックモード		
内部シフトクロックモード	291	
内部タイマ		
内部タイマ	271	
入出力回路形式		
入出力回路形式	16	
入力端子機能		
入力端子機能 (内部クロックモード時)	189	
入力抵抗レジスタ		
入力抵抗レジスタ (RDR0, RDR1)	141	
発振安定待ち時間		
発振安定待ち時間設定	107	
汎用レジスタ		
汎用レジスタ	41	
品種構成		
品種構成	4	
変換データ保護機能		
変換データ保護機能	248	
未定義命令		
未定義命令の実行による例外発生	77	
要求レベル設定レジスタ		
要求レベル設定レジスタ (ELVR)	218	
連続モード		
連続モード	240	
連続モード時の EI ² OS の起動例	244	

CM44-10103-6

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

MB90550A/B Series

ハードウェアマニュアル

2008 年 7 月 第 6 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
