



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

MB90560/565series

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

MB90560/565series

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。

MB90560/565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一つとして開発された製品です。

本書は、MB90560/565シリーズを使って製品を設計する技術者の方を対象に、MB90560/565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。

■ 本書の全体構成

本書は、以下に示す20の章から構成されています。

第1章 概要

MB90560/565シリーズの特徴や基本的な仕様について説明します。

第2章 CPU

この章では、MB90560/565シリーズの機能と動作について説明します。

第3章 リセット

この章では、MB90560/565シリーズのリセットについて説明します。

第4章 クロック

この章では、MB90560/565シリーズのクロックについて説明します。

第5章 低消費電力モード

この章では、MB90560/565シリーズの低消費電力モードについて説明します。

第6章 割込み

この章では、MB90560/565シリーズの割込みと拡張インテリジェントI/Oサービス (EI²OS) について説明します。

第7章 モード設定

この章では、MB90560/565シリーズの動作モードとメモリアクセスモードについて説明します。

第8章 入出力ポート

この章では、MB90560/565シリーズの入出力ポートの機能と動作について説明します。

第9章 タイムベースタイマ

この章では、MB90560/565シリーズの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10章 ウォッチドッグタイマ

この章では、MB90560/565シリーズの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11章 16ビットリロードタイマ

この章では、MB90560/565シリーズの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 多機能タイマ

この章では、MB90560/565シリーズの多機能タイマの動作について説明します。

第13章 UART

この章では、MB90560/565シリーズのUARTの機能と動作について説明します。

第14章 DTP/外部割り込み回路

この章では、MB90560/565シリーズのDTP/外部割り込み回路の機能と動作について説明します。

第15章 遅延割り込み発生モジュール

この章では、MB90560/565シリーズの遅延割り込み発生モジュールの機能と動作について説明します。

第16章 8/16ビットA/Dコンバータ

この章では、MB90560/565シリーズの8/10ビットA/Dコンバータの機能と動作について説明します。

第17章 アドレス一致検出機能

この章では、MB90560/565シリーズのアドレス一致検出機能と動作について説明します。

第18章 ROMミラー機能選択モジュール

この章では、MB90560/565シリーズの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19章 512Kビットフラッシュメモリ

この章では、MB90560/565シリーズの512K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20章 F²MC-16LX MB90F562T シリアル書込み接続例

この章では、F²MC-16LX MB90F562T 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、命令一覧表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。

Copyright ©2000-2008 FUJITSU MICROELECTRONICS LIMITED All rights reserved.

目次

第1章	概 要	1
1.1	特徴	2
1.2	品種構成	4
1.3	ブロックダイヤグラム	6
1.4	端子配列図	7
1.5	パッケージ外形寸法図	10
1.6	端子機能説明	13
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	18
第2章	CPU	23
2.1	CPUの概要	24
2.2	メモリ空間	25
2.3	メモリマップ	27
2.4	アドレッシング	29
2.4.1	リニア方式によるアドレス指定	30
2.4.2	バンク方式によるアドレス指定	31
2.5	多バイト長データの配置	33
2.6	レジスタ	35
2.7	専用レジスタ	36
2.7.1	アキュムレータ (A)	38
2.7.2	スタックポインタ (USP, SSP)	41
2.7.3	プロセッサステータス (PS)	43
2.7.4	コンディションコードレジスタ (PS : CCR)	44
2.7.5	レジスタバンクポインタ (PS : RP)	46
2.7.6	インタラプトレベルマスクレジスタ (PS : ILM)	47
2.7.7	プログラムカウンタ (PC)	48
2.7.8	ダイレクトページレジスタ (DPR)	49
2.7.9	バンクレジスタ (PCB, DTB, USB, SSB, ADB)	50
2.8	汎用レジスタ	51
2.9	プリフィックスコード	53

2.9.1	バンクセレクトプリフィックス(PCB,DTB,ADB,SPB)	54
2.9.2	コモンレジスタバンクプリフィックス(CMR)	56
2.9.3	フラグ変化抑止プリフィックス(NCC)	57
2.9.4	プリフィックスコードに関する制約	58
2.9.5	「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意	60
第3章	リセット	63
3.1	リセットの概要	64
3.2	リセット要因と発振安定待ち時間	66
3.3	外部リセット端子	67
3.4	リセット動作	68
3.5	リセット要因フラグビット	70
3.6	リセットによる各端子の状態	73
第4章	クロック	75
4.1	クロックの概要	76
4.2	クロック発生部のブロックダイアグラム	78
4.3	クロック選択レジスタ(CKSCR)	80
4.4	クロックモード	82
4.5	発振安定待ち時間	84
4.6	振動子と外部クロックの接続	85
第5章	低消費電力モード	87
5.1	低消費電力モードの概要	88
5.2	低消費電力制御回路のブロックダイアグラム	90
5.3	低消費電力モード制御レジスタ(LPMCR)	92
5.4	CPU間欠動作モード	95
5.5	スタンバイモード	96
5.5.1	スリープモード	97
5.5.2	タイムベースタイマモード	99
5.5.3	ストップモード	101
5.6	状態遷移図	103
5.7	スタンバイモード,ホールド,リセットの端子状態	104
5.8	低消費電力モード使用上の注意	105

第6章	割込み	107
6.1	割込みの概要	108
6.2	割込み要因と割込みベクタ	110
6.3	割込み制御レジスタ (ICR) と周辺機能 (リソース)	113
6.3.1	割込み制御レジスタ (ICR00 ~ ICR15)	115
6.3.2	割込み制御レジスタ (ICR) の機能	117
6.4	ハードウェア割込み	120
6.4.1	ハードウェア割込みの動作	122
6.4.2	割込み動作時の処理	124
6.4.3	ハードウェア割込み使用手順	125
6.4.4	多重割込み	126
6.4.5	ハードウェア割込み処理時間	128
6.5	ソフトウェア割込み	130
6.6	拡張インテリジェントI/Oサービス(EI ² OS)による割込み	132
6.6.1	拡張インテリジェントI/Oサービス (EI ² OS) ディスクリプタ (ISD)	134
6.6.2	拡張インテリジェントI/Oサービス (EI ² OS) ディスクリプタ (ISD) の説明	135
6.6.3	拡張インテリジェントI/Oサービス (EI ² OS) の動作	138
6.6.4	拡張インテリジェントI/Oサービス (EI ² OS) の設定手順	139
6.6.5	拡張インテリジェントI/Oサービス (EI ² OS) 処理時間	140
6.7	例外処理	142
6.8	割込み処理のスタック動作	143
6.9	割込み処理のプログラム例	145
第7章	モード設定	149
7.1	モード設定	150
7.2	モード端子 (MD2 ~ MD0)	151
7.3	モードデータレジスタ	152
第8章	入出力ポート	155
8.1	入出力ポートの概要	156
8.2	入出力ポートのレジスタ説明	158
8.3	ポート0	159
8.3.1	ポート0のレジスタ (PDR0, DDR0, RDR0)	161

8.3.2	ポート0の動作説明	162
8.4	ポート1	164
8.4.1	ポート1のレジスタ (PDR1, DDR1, RDR1)	166
8.4.2	ポート1の動作説明	167
8.5	ポート2	169
8.5.1	ポート2のレジスタ (PDR2, DDR2)	171
8.5.2	ポート2の動作説明	172
8.6	ポート3	174
8.6.1	ポート3のレジスタ (PDR3, DDR3)	176
8.6.2	ポート3の動作説明	177
8.7	ポート4	179
8.7.1	ポート4のレジスタ (PDR4, DDR4)	181
8.7.2	ポート4の動作説明	182
8.8	ポート5	184
8.8.1	ポート5のレジスタ (PDR5, DDR5, ADER)	186
8.8.2	ポート5の動作説明	187
8.9	ポート6	189
8.9.1	ポート6のレジスタ (PDR6, DDR6)	191
8.9.2	ポート6の動作説明	192
8.10	入出力ポートのプログラム例	194
第9章	タイムベースタイマ	195
9.1	タイムベースタイマの概要	196
9.2	タイムベースタイマの構成	197
9.3	タイムベースタイマ制御レジスタ (TBTC)	199
9.4	タイムベースタイマの割込み	201
9.5	タイムベースタイマの動作説明	202
9.6	タイムベースタイマ使用上の注意	205
9.7	タイムベースタイマのプログラム例	206
第10章	ウォッチドッグタイマ	207
10.1	ウォッチドッグタイマの概要	208
10.2	ウォッチドッグタイマの構成	209
10.3	ウォッチドッグタイマ制御レジスタ (WDTC)	211

10.4	ウォッチドッグタイマの動作説明	213
10.5	ウォッチドッグタイマ使用上の注意	215
10.6	ウォッチドッグタイマのプログラム例	216
第11章	16ビットリロードタイマ	217
11.1	16ビットリロードタイマの概要	218
11.2	16ビットリロードタイマの構成	221
11.3	16ビットリロードタイマの端子	223
11.4	16ビットリロードタイマのレジスタ	224
11.4.1	タイマコントロールステータスレジスタ上位 (TMCSR0/TMCSR1 : H)	225
11.4.2	タイマコントロールステータスレジスタ下位 (TMCSR0/TMCSR1 : L)	227
11.4.3	16ビットタイマレジスタ (TMR0/TMR1)	229
11.4.4	16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1)	230
11.5	16ビットリロードタイマの割込み	231
11.6	16ビットリロードタイマの動作説明	232
11.6.1	内部クロックモード (リロードモード)	234
11.6.2	内部クロックモード (ワンショットモード)	237
11.6.3	イベントカウントモード	240
11.7	16ビットリロードタイマ使用上の注意	242
11.8	16ビットリロードタイマのプログラム例	243
第12章	多機能タイマ	247
12.1	多機能タイマの概要	248
12.2	多機能タイマの構成	251
12.3	多機能タイマのレジスタ一覧	254
12.3.1	16ビットフリーランタイマのレジスタ	256
12.3.2	アウトプットコンペアのレジスタ	262
12.3.3	インプットキャプチャのレジスタ	267
12.3.4	8/16ビットPPGタイマのレジスタ	272
12.3.5	波形生成部のレジスタ	278
12.4	多機能タイマの動作説明	283
12.4.1	16ビットフリーランタイマ	284
12.4.2	アウトプットコンペア	286
12.4.3	インプットキャプチャ	288

12.4.4	8/16ビットPPGタイマ	289
12.4.5	波形生成部	294
12.4.6	デッドタイム制御回路部の動作	295
12.4.7	PPG出力およびGATE信号制御回路部の動作	299
12.4.8	DTTI端子入力制御の動作	301
第13章	UART	303
13.1	UARTの概要	304
13.2	UARTの構成	306
13.3	UARTの端子	308
13.4	UARTのレジスタ	309
13.4.1	制御レジスタ (SCR0/SCR1)	310
13.4.2	モードレジスタ (SMR0/SMR1)	312
13.4.3	ステータスレジスタ (SSR0/SSR1)	314
13.4.4	インプットデータレジスタ (SIDR0/SIDR1), アウトプットデータレジスタ (SODR0/SODR1)	316
13.4.5	通信プリスケアラ制御レジスタ (CDCR0/CDCR1)	318
13.5	UARTの割込み	319
13.5.1	受信割込み出力とフラグセットのタイミング	321
13.5.2	送信割込み出力とフラグセットのタイミング	323
13.6	UARTのボーレート	324
13.6.1	専用ボーレートジェネレータによるボーレート	326
13.6.2	内部クロック (16ビットリロードタイマ) によるボーレート	328
13.6.3	外部クロックによるボーレート	330
13.7	UARTの動作説明	331
13.7.1	非同期モード (動作モード0, 1) 時の動作	333
13.7.2	同期モード (動作モード2) 時の動作	336
13.7.3	双方向通信機能 (ノーマルモード)	339
13.7.4	マスタ/スレーブ型通信機能 (マルチプロセッサモード)	341
13.8	UART使用上の注意	344
第14章	DTP / 外部割込み回路	345
14.1	DTP / 外部割込み回路の概要	346
14.2	DTP / 外部割込み回路の構成	347
14.3	DTP / 外部割込み回路の端子	349

14.4	DTP / 外部割込み回路のレジスタ	350
14.4.1	DTP / 割込み要因レジスタ (EIRR)	351
14.4.2	DTP / 割込み許可レジスタ (ENIR)	354
14.4.3	要求レベル設定レジスタ (ELVR)	357
14.5	DTP / 外部割込み回路の動作説明	359
14.5.1	外部割込み機能	362
14.5.2	DTP機能	363
14.6	DTP / 外部割込み回路使用上の注意	364
14.7	DTP / 外部割込み回路のプログラム例	366
第15章	遅延割込み発生モジュール	369
15.1	遅延割込み発生モジュールの概要	370
15.2	遅延割込み要因発生/解除レジスタ (DIRR)	371
15.3	遅延割込み発生モジュールの動作説明	372
15.4	遅延割込み発生モジュール使用上の注意	373
第16章	8/10ビットA/Dコンバータ	375
16.1	8/10ビットA/Dコンバータの概要	376
16.2	8/10ビットA/Dコンバータの構成	377
16.3	8/10ビットA/Dコンバータの端子	379
16.4	8/10ビットA/Dコンバータのレジスタ	381
16.4.1	A/D制御ステータスレジスタ (ADCS1)	382
16.4.2	A/D制御ステータスレジスタ (ADCS0)	384
16.4.3	A/Dデータレジスタ (ADCRO/ADCR1)	386
16.5	8/10ビットA/Dコンバータの割込み	388
16.6	8/10ビットA/Dコンバータの動作説明	389
16.6.1	EI ² OSを使用した変換動作	392
16.6.2	A/D変換データ保護機能	393
16.7	8/10ビットA/Dコンバータ使用上の注意	396
16.8	8/10ビットA/Dコンバータのプログラム例-1 (単発変換モードのEI ² OS起動例) ..	397
16.9	8/10ビットA/Dコンバータのプログラム例-2 (連続変換モードのEI ² OS起動例) ..	399
16.10	8/10ビットA/Dコンバータのプログラム例-3 (停止変換モードのEI ² OS起動例) ..	401

第17章	アドレス一致検出機能	403
17.1	アドレス一致検出機能の概要	404
17.2	アドレス一致検出機能のレジスタ	405
17.2.1	プログラムアドレス検出レジスタ (PADR0/PADR1)	406
17.2.2	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)	407
17.3	アドレス一致検出機能の動作説明	408
17.4	アドレス一致検出機能の使用例	409
第18章	ROMミラー機能選択モジュール	413
18.1	ROMミラー機能選択モジュールの概要	414
18.2	ROMミラー機能選択レジスタ (ROMM)	415
第19章	512Kビット(64Kバイト)フラッシュメモリ	417
19.1	512Kビット(64Kバイト)フラッシュメモリの概要	418
19.2	512Kビット(64Kバイト)フラッシュメモリセクタ構成	419
19.3	フラッシュメモリコントロールステータスレジスタ (FMCS)	420
19.4	フラッシュメモリ自動アルゴリズム起動方法	423
19.5	自動アルゴリズム実行状態の確認	424
19.5.1	データポーリングフラグ (DQ7)	425
19.5.2	トグルビットフラグ (DQ6)	426
19.5.3	タイミングリミット超過フラグ (DQ5)	427
19.5.4	セクタ消去タイマフラグ (DQ3)	428
19.6	フラッシュメモリ書込み/消去の詳細説明	429
19.6.1	読出し/リセット状態にする	430
19.6.2	データ書込みを行う	431
19.6.3	データ消去を行う (チップ消去)	433
19.6.4	データ消去を行う (セクタ消去)	434
19.6.5	セクタ消去を一時停止する	436
19.6.6	セクタ消去を再開する	437
19.7	512Kビット(64Kバイト)フラッシュメモリのコマンドプログラム例	438
第20章	F ² MC-16LX MB90F562/F568シリアル書込み接続例	443
20.1	富士通標準シリアルオンボード書込みの標準構成	444
20.2	シリアル書込み接続例(ユーザー電源使用時)	447

20.3	シリアル書込み接続例(ライター電源から供給時).....	449
20.4	フラッシュマイコンプログラマとの最小限の接続例(ユーザー電源使用時)....	452
20.5	フラッシュマイコンプログラマとの最小限の接続例(ライター電源から供給時)..	454
付録		457
A	I/Oマップ	458
B	命令	464
B.1	命令の種類	465
B.2	アドレッシング	466
B.3	直接アドレッシング	468
B.4	間接アドレッシング	475
B.5	実行サイクル数	482
B.6	実効アドレスフィールド	484
B.7	命令一覧表の読み方	486
B.8	F ² MC-16LX命令一覧表	489
B.9	命令マップ	503
索引		525

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。）
464 ~ 524	「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。

第1章 概 要

この章では，MB90560/565シリーズの特徴と基本的な仕様について説明します。

- 1.1 特徴
- 1.2 品種構成
- 1.3 ブロックダイヤグラム
- 1.4 端子配列図
- 1.5 パッケージ外形寸法図
- 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 特徴

MB90560/565シリーズは、各種産業用、OA機器用、プロセス制御等に適した高速リアルタイム処理が要求される用途向けに開発された汎用の16ビットマイクロコントローラです。任意波形を出力できる多機能タイマを内蔵しています。

命令体系は、F²MC-8L、F²MC-16Lと同じくATアーキテクチャを継承し、C言語対応・アドレッシングモードの拡張・符号付き乗除算命令の強化・ビット処理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、ロングワード処理が可能です。

MB90560/565シリーズの特徴

- クロック
 - 発振回路およびPLL逡倍回路内蔵
 - 発振クロック
発振クロックの2分周（1～16MHz発振時：0.5～8MHz）のメインクロック，
発振クロックの1逡倍～4逡倍（4MHz発振時：4MHz～16MHz）のPLLクロック，マシクロックから設定可能
 - 最小命令実行時間，62.5ns（4MHz発振，発振クロック4逡倍，V_{cc} = 5V動作時）
- 最大メモリ空間：16Mバイト
 - 24ビットアドレッシング
 - バンクアドレッシング
- 命令体系
 - データタイプ（ビット，バイト，ワード，ロングワード）
 - アドレッシングモード（23種類）
 - 32ビットアキュムレータの採用による演算精度の強化
 - 符号付き乗除算命令，RETI命令機能強化
- C言語／マルチタスクに対応する命令体系
 - 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
- プログラムパッチ機能（2アドレスポインタ）
- 4バイトの命令キュー
- 割込み機能
 - プライオリティレベルがプログラマブルに設定可能
 - 32要因の割込み機能
- データ転送機能
 - 拡張インテリジェントI/Oサービス機能：最大16チャンネル

- 低消費電力モード
 - スリープモード（CPU動作クロックが停止します。）
 - タイムベースタイマモード
（発振クロックとタイムベースタイマが動作します。）
 - ストップモード（発振クロックが停止します。）
 - CPU間欠動作モード（設定されたサイクル毎にCPUが動作します。）
- パッケージ
 - QFP-64
（FPT-64P-M09:0.65mmピンピッチ，FPT-64P-M06:1.00mmピンピッチ）
 - SH-DIP（DIP-64P-M01:1.778mmピンピッチ）
- プロセス：CMOSテクノロジー

内蔵周辺機能（リソース）

- I/Oポート：最大51本
- タイムベースタイマ：1チャンネル
- ウォッチドッグタイマ：1チャンネル
- 16ビットリロードタイマ：2チャンネル
- 多機能タイマ：
 - 16ビットフリーランタイマ：1チャンネル
 - アウトプットコンペア：6チャンネル
16ビットフリーランタイマのカウント値とコンペアレジスタへの設定値が一致した場合、割込み要求を出力させることができます。
 - インプットキャプチャ：4チャンネル
外部入力端子から入力された信号の有効エッジを検出することにより、16ビットフリーランタイマのカウント値をインプットキャプチャデータレジスタに取り込み、割込み要求を出力させることができます。
 - 8/16ビットPPGタイマ（8ビット×6チャンネル or 16ビット×3チャンネル）
出力パルスの周期・デューティを任意に変更することができます。
 - 波形生成部（8ビットタイマ：3チャンネル）
- UART：2チャンネル
 - 全二重ダブルバッファ（8bit長）付き
 - 非同期転送またはクロック同期シリアル転送（I/O拡張シリアル）が設定できます。
- DTP/外部割込み回路（8チャンネル）
 - 外部割込み入力による拡張インテリジェントI/Oサービスの起動
 - 外部割込み入力による割込み出力
- 遅延割込み出力モジュール
 - タスク切替え用の割込み要求を出力
- 8/10ビットA/Dコンバータ（8チャンネル）
 - 8ビット分解能または10ビット分解能が設定できます。

1.2 品種構成

表 1.2-1にMB90560シリーズの品種構成表を、表 1.2-2にMB90565シリーズの品種構成表を示します。

品種構成

表 1.2-1 MB90560シリーズの品種構成表

品 種 名	MB90V560	MB90F562/B	MB90562/A	MB90561/A
分類	評価用	フラッシュメモリ 内蔵タイプ	マスクROM内蔵タイプ	
ROM容量	搭載なし	64Kバイト		32Kバイト
RAM容量	4Kバイト	2Kバイト		1Kバイト
エミュレータ専用 電源*	無	-		-
CPU機能	基本命令数 : 351命令 最小命令実行時間 : 62.5ns/4MHz発振(4通倍使用時) アドレッシング種類 : 23種類 プログラムパッチ機能 : 2アドレスポインタ分 最大メモリ空間 : 16Mバイト			
ポート	入出力ポート(CMOS) 51本			
UART	全二重ダブルバッファ付き クロック同期 / 非同期が設定可能 I/Oシリアルとしても使用可能 専用ボーレートジェネレータ内蔵 2チャンネル内蔵			
16ビット リロードタイマ	16ビットリロードタイマ動作 2チャンネル内蔵			
多機能タイマ	16ビットフリーランタイマ×1チャンネル , アウトプットコンペア×6チャンネル インプットキャプチャ×4チャンネル 8/16ビットPPGタイマ (8bitモード×6ch , 16bitモード×3ch) 波形生成部 (8ビットタイマ×3ch) 三相波形出力 , デッドタイム出力			
8/10ビット A/Dコンバータ	8チャンネル (入力マルチプレクス) 8ビット分解能または10ビット分解能が設定可能 変換時間 : 6.13 μs (最大マシクロック16MHz時)			
DTP/外部割込み	8チャンネル (A/D入力と兼用で8チャンネル使用可能) 割込み要因 : "L" "H"エッジ / "H" "L"エッジ / "L"レベル / "H"レベル設定可能			
低消費電力モード プロセス	スリープモード / タイムベースタイマモード / ストップモード / CPU間欠モード CMOS			
パッケージ	PGA-256	QFP-64 (FPT-64P-M09 : 0.65mmピンピッチ) QFP-64 (FPT-64P-M06 : 1.00mmピンピッチ) SH-DIP (FPT-64P-M01 : 1.778mmピンピッチ)		
動作電圧	5V ± 10% (最大マシクロック16MHz時)			

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

表 1.2-2 MB90565シリーズの品種構成表

品 種 名	MB90F568		MB90568	MB90567
分類	フラッシュメモリ 内蔵タイプ		マスクROM内蔵タイプ	
ROM容量	128Kバイト			96Kバイト
RAM容量	4Kバイト			4Kバイト
エミュレータ専用 電源*	-			-
CPU機能	基本命令数 : 351命令 最小命令実行時間 : 62.5ns/4MHz発振(4逓倍使用時) アドレッシング種類 : 23種類 プログラムパッチ機能 : 2アドレスポインタ分 最大メモリ空間 : 16Mバイト			
ポート	入出力ポート(CMOS) 51本			
UART	全二重ダブルバッファ付き クロック同期 / 非同期が設定可能 I/Oシリアルとしても使用可能 専用ボーレートジェネレータ内蔵 2チャンネル内蔵			
16ビット リロードタイマ	16ビットリロードタイマ動作 2チャンネル内蔵			
多機能タイマ	16ビットフリーランタイマ×1チャンネル , アウトプットコンペア×6チャンネル インプットキャプチャ×4チャンネル 8/16ビットPPGタイマ (8bitモード×6ch, 16bitモード×3ch) 波形生成部 (8ビットタイマ×3ch) 三相波形出力, デッドタイム出力			
8/10ビット A/Dコンバータ	8チャンネル (入力マルチプレクス) 8ビット分解能または10ビット分解能が設定可能 変換時間 : 6.13 μs (最大マシクロック16MHz時)			
DTP/外部割込み	8チャンネル (A/D入力と兼用で8チャンネル使用可能) 割込み要因 : "L" "H"エッジ / "H" "L"エッジ / "L"レベル / "H"レベル設定可能			
低消費電力モード プロセス	スリープモード / タイムベースタイマモード / ストップモード / CPU間欠モード CMOS			
パッケージ	PGA-256	QFP-64 (FPT-64P-M09 : 0.65mmピンピッチ) QFP-64 (FPT-64P-M06 : 1.00mmピンピッチ) SH-DIP (FPT-64P-M01 : 1.778mmピンピッチ)		
動作電圧	3V ± 10% (最大マシクロック8MHz時)			

1.3 ブロックダイアグラム

図 1.3-1に、MB90560/565シリーズの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

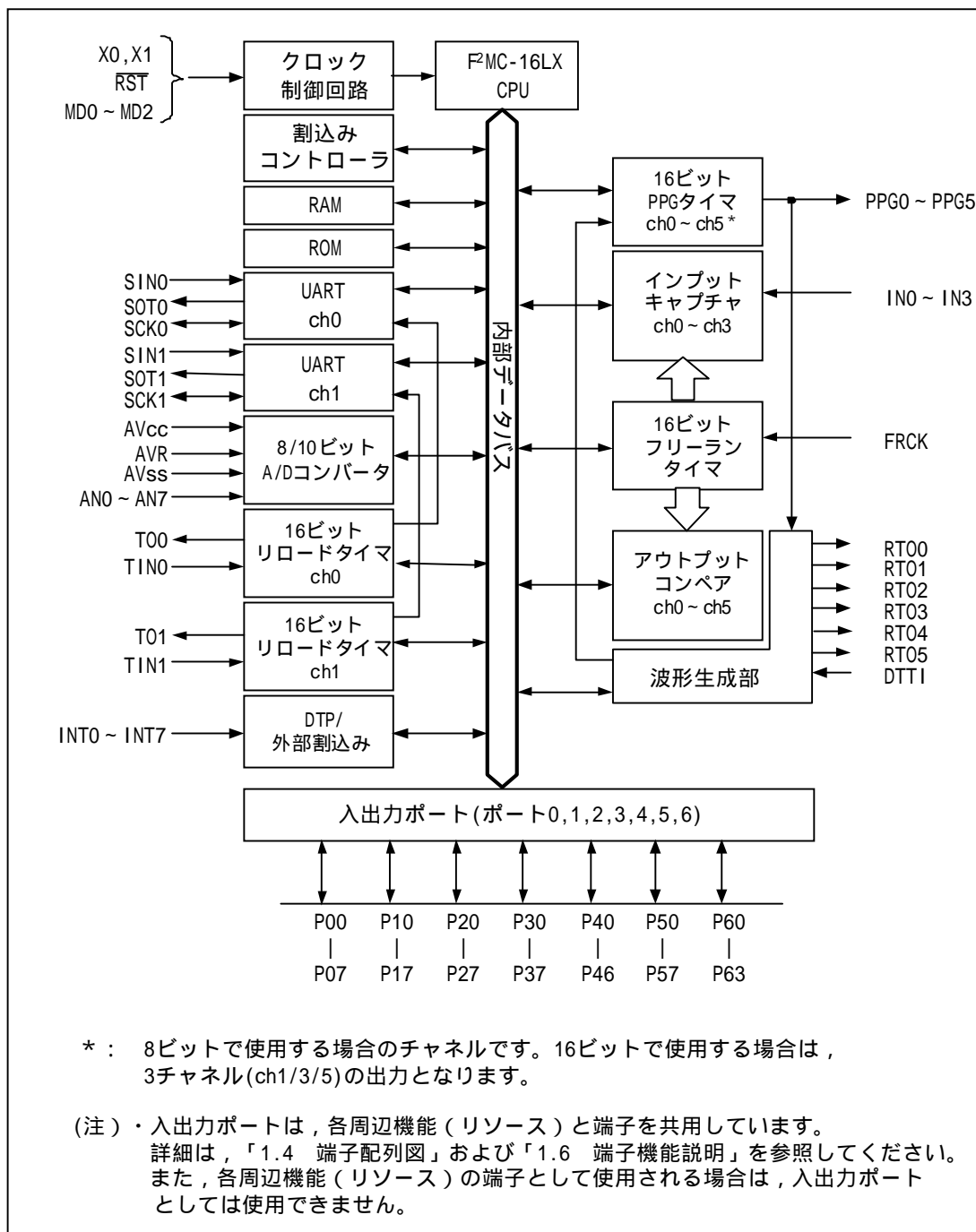


図 1.3-1 ブロックダイアグラム

1.4 端子配列図

MB90560/565シリーズの3つのパッケージについて、端子配列図を示します。

- FPT-64P-M06
- FPT-64P-M09
- DIP-64P-M01

FPT-64P-M06端子配列図

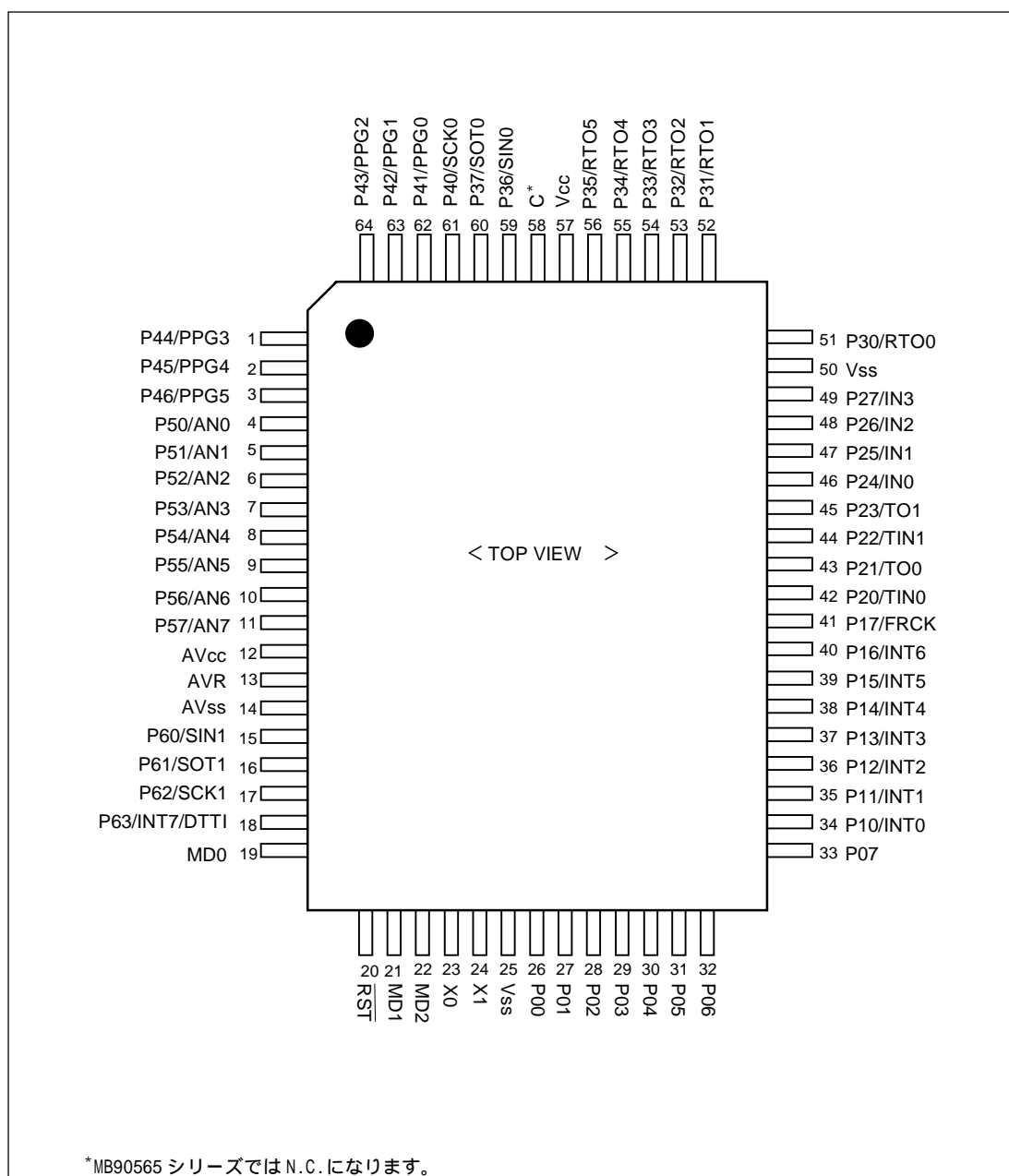


図 1.4-1 FPT-64P-M06端子配列図

FPT-64P-M09端子配列図

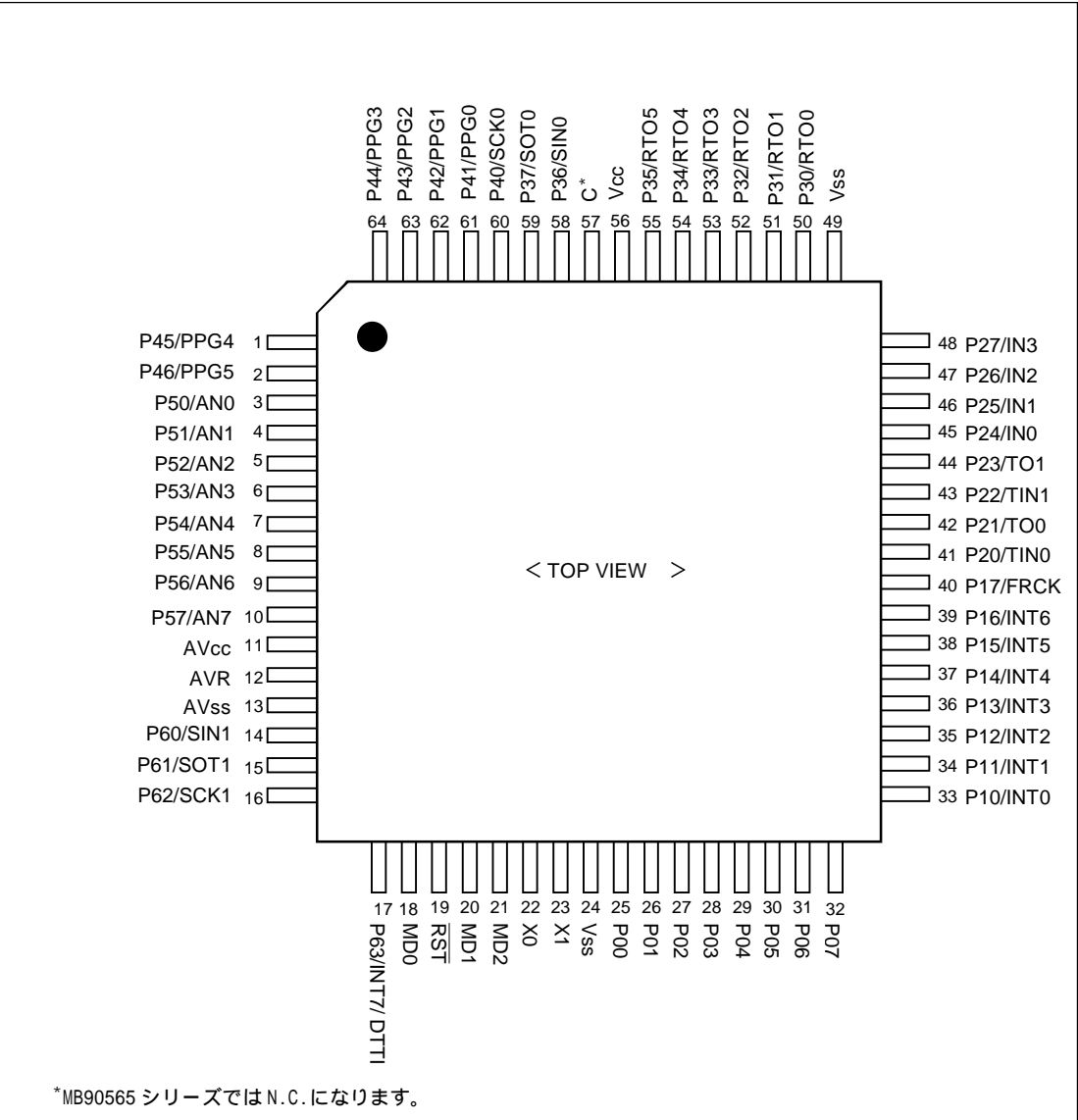


図 1.4-2 FPT-64P-M09端子配列図

DIP-64P-M01端子配列図

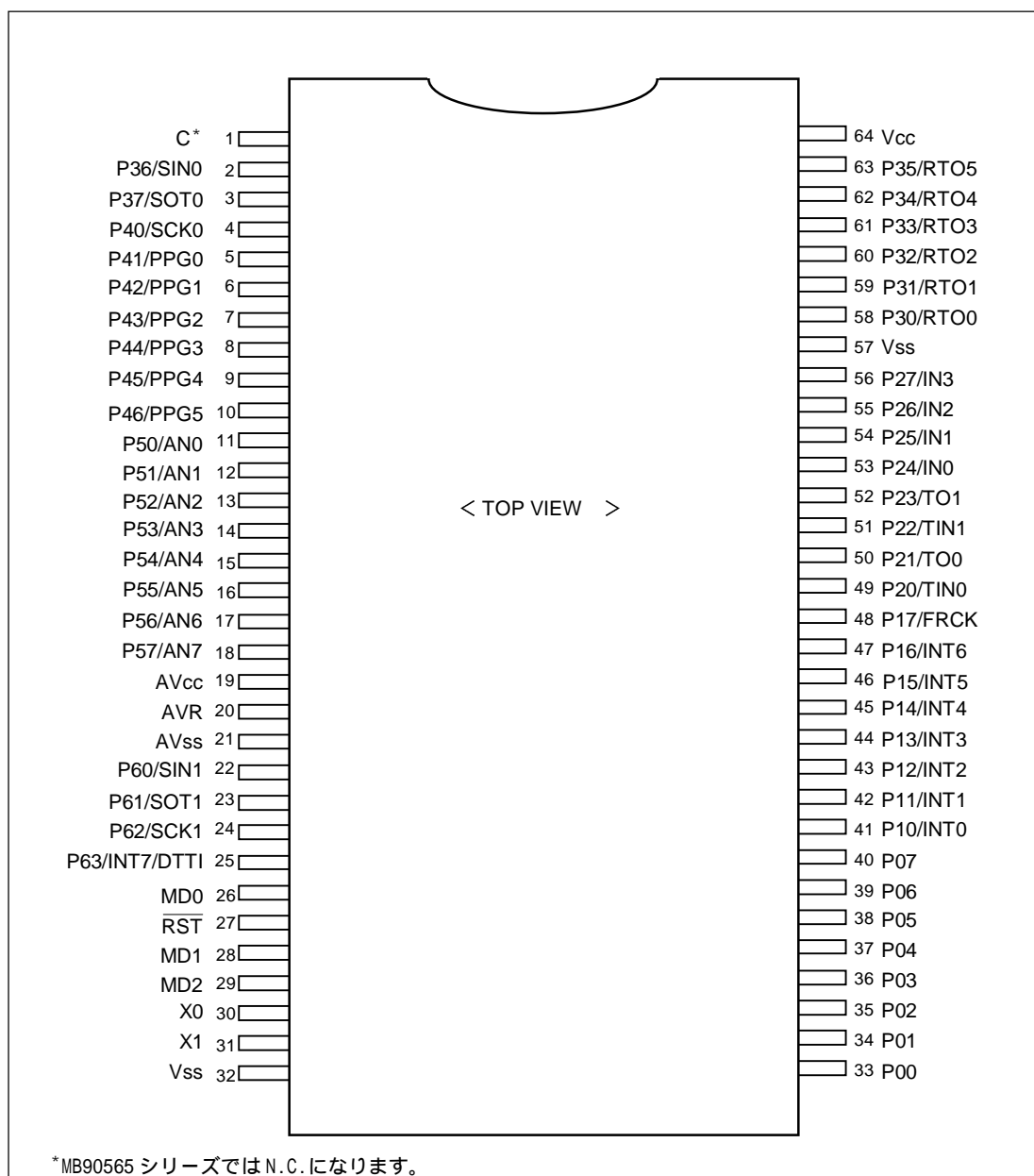


図 1.4-3 DIP-64P-M01端子配列図

1.5 パッケージ外形寸法図

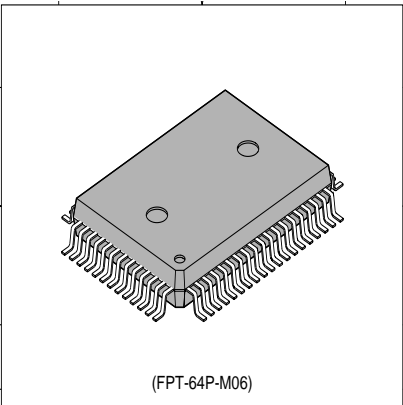
MB90560/565シリーズの3つのパッケージについて外形寸法図を示します。

- ・ FPT-64P-M06
- ・ FPT-64P-M09
- ・ DIP-64P-M01

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-64P-M06パッケージ外形寸法図

EIAJコード：*QFP064-P-1420-3

<p>プラスチック・QFP，64ピン</p>  <p>(FPT-64P-M06)</p>	リードピッチ	1.00mm
	パッケージ幅× パッケージ長さ	14×20mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	端子平坦部長さ	1.20mm

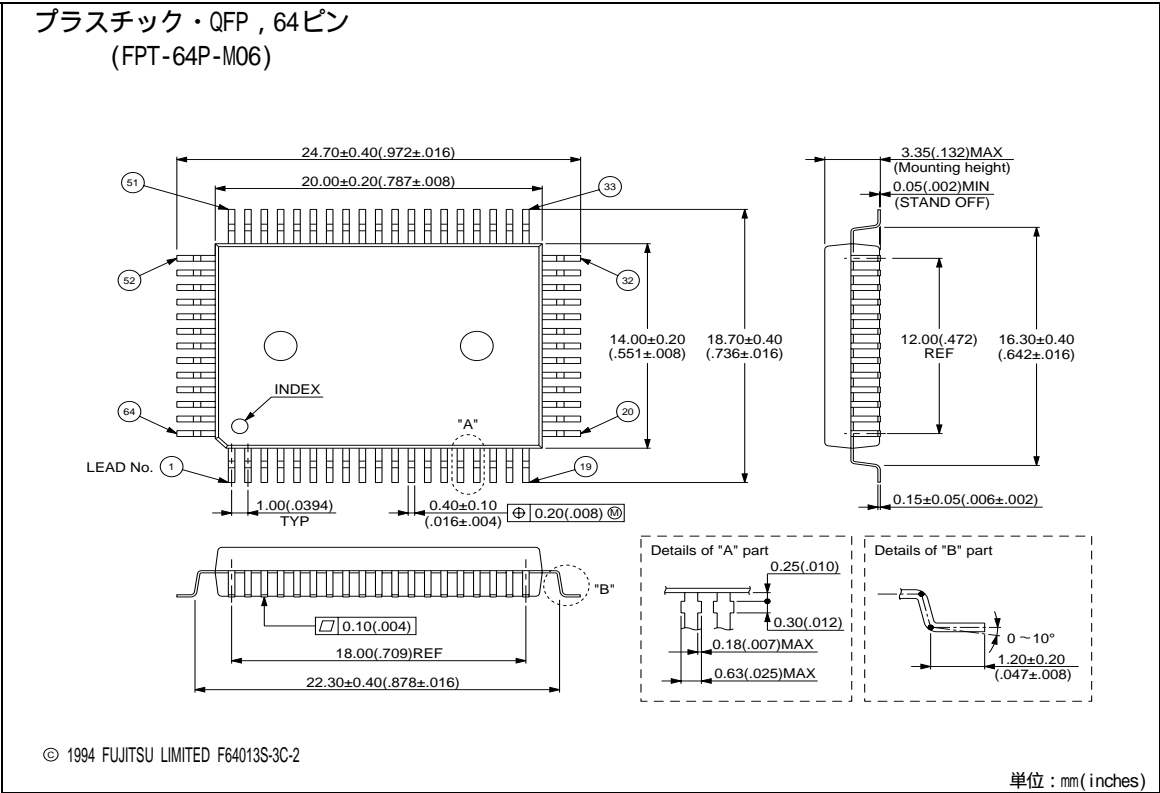
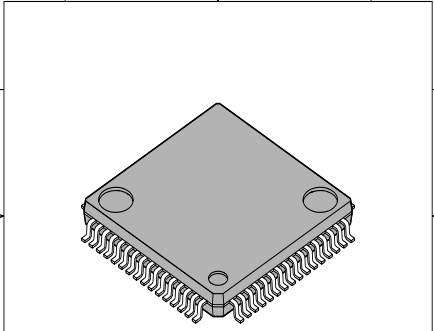
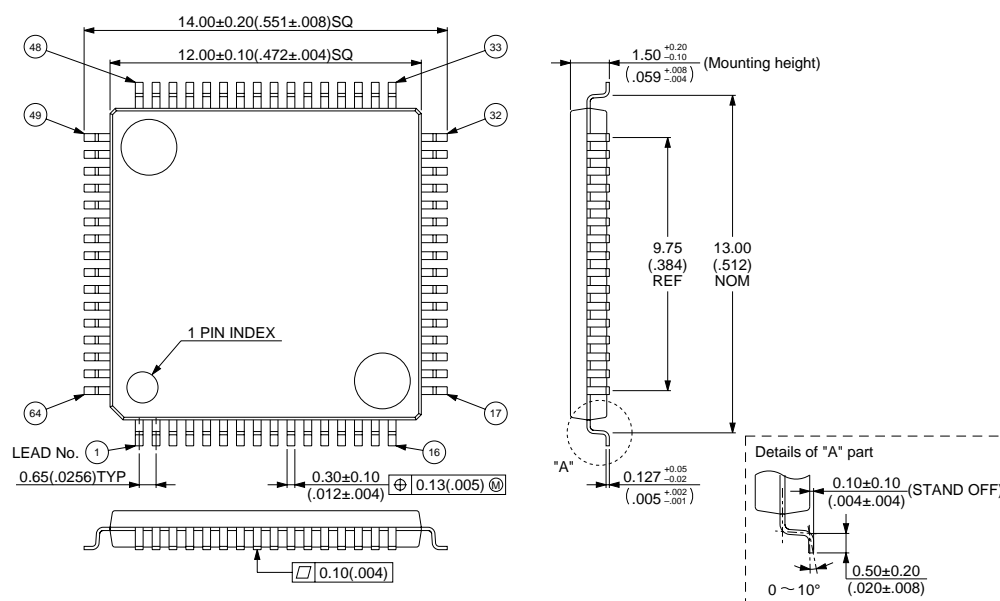


図 1.5-1 FPT-64P-M06パッケージ外形寸法図

FPT-64P-M09パッケージ外形寸法図

EIAJコード：*QFP064-P-1212-1

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

プラスチック・QFP, 64ピン
(FPT-64P-M09)

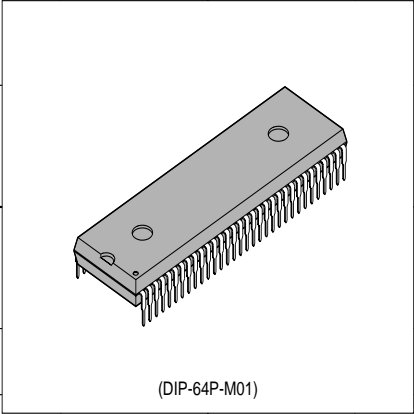
© 1994 FUJITSU LIMITED F64018S-1C-2

単位: mm (inches)

図 1.5-2 FPT-64P-M09パッケージ外形寸法図

DIP-64P-M01パッケージ外形寸法図

EIAJコード：SDIP064-P-0750-1

<p>プラスチック・SH-DIP , 64ピン</p>  <p>(DIP-64P-M01)</p>	リードピッチ	1.778mm(70mil)
	ロースペース	19.05mm(750mil)
	封止方法	プラスチックモールド

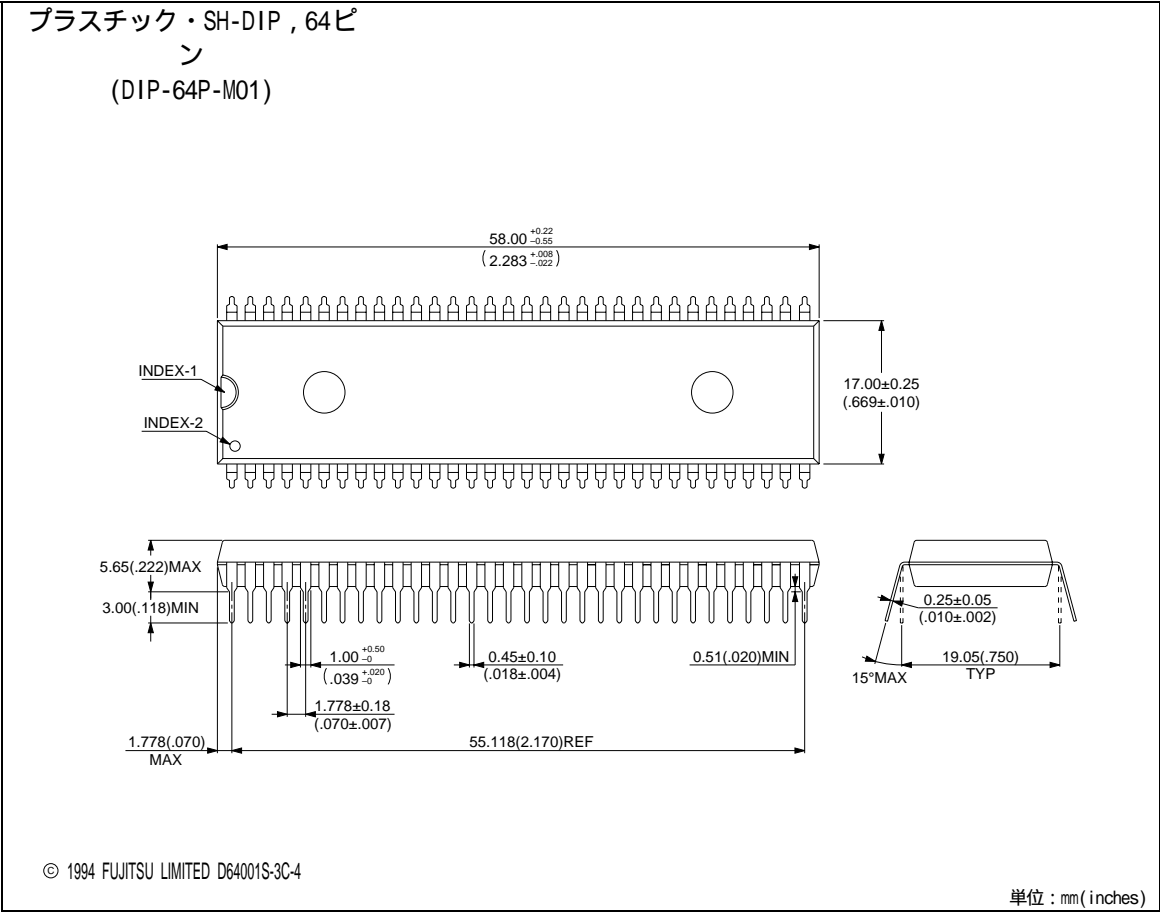


図 1.5-3 DIP-64P-M01パッケージ外形寸法図

1.6 端子機能説明

表 1.6-1に、端子名、機能説明、回路形式、リセット時の状態、機能説明について示します。

端子機能説明

表 1.6-1 端子機能説明（続く）

端子番号			端子名	回路形式*	リセット時の状態/機能	機能説明
QFPM06	QFPM09	SDIP				
23,24	22,23	30,31	X0・X1	A	発振状態	振動子を接続する端子です。 外部クロックを接続する場合はX1端子側を開放してください。
20	19	27	RST	B	リセット入力	外部リセット入力端子です。
26～33	25～32	33～40	P00～P07	C	ポート入力 (Hi-z出力)	入出力ポートです。
34～40	33～39	41～47	P10～P16	C		入出力ポートです。
			INT0～INT6			割込み要求入力ch0～ch6として使用できません。スタンバイ時にはEN0～EN6の対応する各ビットに"1"を設定し、入力ポートを設定した場合に入力できます。使用時にはアナログ入力許可レジスタ（ADER）の対応する各ビットをポート設定として使用してください。
			P17			入出力ポートです。
41	40	48	FRCK	C		フリーランタイムの外部クロック入力端子です。フリーランタイムのクロック入力を設定し、入力ポートを設定した場合に入力できます。使用時にはアナログ入力許可レジスタ（ADER）の対応する各ビットをポート設定として使用してください。
			P20	D		入出力ポートです。
42	41	49	TIN0			リロードタイムch0の外部クロック入力端子です。外部クロック入力を設定し、入力ポートを設定した場合に入力できます。
			P21	D		入出力ポートです。
43	42	50	T00			リロードタイムch0のイベント出力端子です。イベント出力許可を設定した場合に出力されます。
			P22	D		入出力ポートです。
44	43	51	TIN1			リロードタイムch1の外部クロック入力端子です。外部クロック入力を設定し、入力ポートを設定した場合に入力できます。
			P23	D		入出力ポートです。
45	44	52	T01			リロードタイムch1のイベント出力端子です。イベント出力許可を設定した場合に出力されます。
			P24～P27	D		入出力ポートです。
46～49	45～48	53～56	IN0～IN3			インプットキャプチャch0～ch3のトリガ入力端子です。インプットキャプチャトリガ入力を設定し、入力ポートを設定した場合に入力できます。

*：回路形式につきましては、「1.7 入出力回路形式」を参照してください。

表 1.6-1 端子機能説明 (続き)

端子番号			端子名	回路形式	リセット時の状態/機能	機能説明
QFPM06	QFPM09	SDIP				
51 ~ 56	50 ~ 55	58 ~ 63	P30 ~ P35	E	ポート入力 (Hi-z)	入出力ポートです。
			RT05 ~ RT00			アウトプットコンペアイイベント出力端子/波形生成部出力端子です。波形生成部にて指定した波形を出力します。波形生成動作を行わない場合は、アウトプットコンペアのイベント出力許可を設定すると、アウトプットコンペアの出力を行います。使用時にはADERの対応する各ビットをポート設定として使用してください。
59	58	2	P36	D		入出力ポートです。
			SIN0			UART ch0のシリアルデータ入力端子です。UART ch0が入力動作中には随時使用していますので他の入力端子として使用しないでください。
60	59	3	P37	D		入出力ポートです。
			SOT0			UART ch0のシリアルデータ出力端子です。UART ch0の出力許可を設定した場合に出力されます。
61	60	4	P40	D		入出力ポートです。
			SCK0			UART ch0のシリアルクロック入出力端子です。UART ch0のクロック出力許可を設定した場合に出力されます。
62 ~ 64 , 1 ~ 3	61 ~ 64 , 1, 2	5 ~ 10	P41 ~ P46	D		入出力ポートです。
			PPG0 ~ PPG5			PPG ch0 ~ PPG ch5の出力端子です。PPG ch0 ~ PPG ch5の出力許可を設定した場合に出力されます。
4 ~ 11	3 ~ 10	11 ~ 18	P50 ~ P57	F	アナログ入力	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。アナログ入力許可レジスタを設定した場合に入力できます(ADER:bit0 ~ bit7)。
12	11	19	AVcc	-	電源入力	A/DコンバータのVcc電源入力端子です。
13	12	20	AVR	G	リファレンス電圧入力	A/Dコンバータのリファレンス電圧入力端子です。Vccを超えないようにしてください。
14	13	21	AVss	-	電源入力	A/DコンバータのVss電源入力端子です。
15	14	22	P60	D	ポート入力 (Hi-z)	入出力ポートです。
			SIN1			UART ch1のシリアルデータ入力端子です。UART ch1が入力動作中には随時使用していますので他の入力端子として使用しないでください。
16	15	23	P61	D		入出力ポートです。
			SOT1			UART ch1のシリアルデータ出力端子です。UART ch1の出力許可を設定した場合に出力されます。
17	16	24	P62	D		入出力ポートです。
			SCK1			UART ch1のシリアルクロック入出力端子です。UART ch1のクロック出力許可を設定した場合に出力されます。

* : 回路形式につきましては , 「1.7 入出力回路形式」を参照してください。

表 1.6-1 端子機能説明（続き）

端子番号			端子名	回路形式 ^{*1}	リセット時の状態/機能	機能説明
QFPM06	QFPM09	SDIP				
18	17	25	P63	D	ポート入力 (Hi-z)	入出力ポートです。
			INT7			割込み要求入力ch7として使用できます。スタンバイ時には、EN7の対応する各ビットに"1"を設定し、入力ポートを設定した場合に inputs できます。
			DTTI			RT00～RT05端子使用時の端子レベル固定入力端子です。波形生成部で入力許可を設定した場合に inputs が許可されます。
58	57	1	C ^{*2}	-	容量端子 電源入力	電源安定化の容量端子です。外部に0.1μF程度のセラミックコンデンサを接続してください。
19	18	26	MD0	B	モード入力 端子	動作モード指定用入力端子です。VccまたはVssに直結してください。
21	20	28	MD1	B		動作モード指定用入力端子です。VccまたはVssに直結してください。
22	21	29	MD2	B		動作モード指定用入力端子です。Vssに直結してください。
25,50	24,49	32,57	Vss	-	電源入力	電源（GND）入力端子です。
57	56	64	Vcc	-		電源（5V）入力端子です。

*1：回路形式につきましては、「1.7 入出力回路形式」を参照してください。

*2：MB90565シリーズではN.C.になります。

1.7 入出力回路形式

表 1.7-1に,各端子の回路形式について示します。

入出力回路形式

表 1.7-1 入出力回路形式（続く）

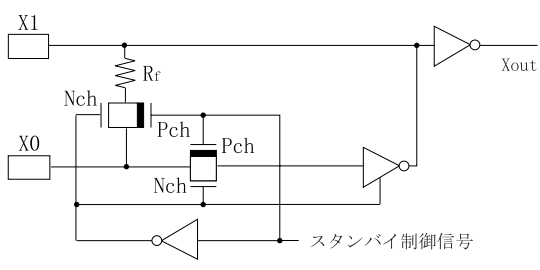
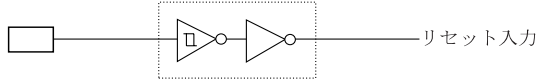
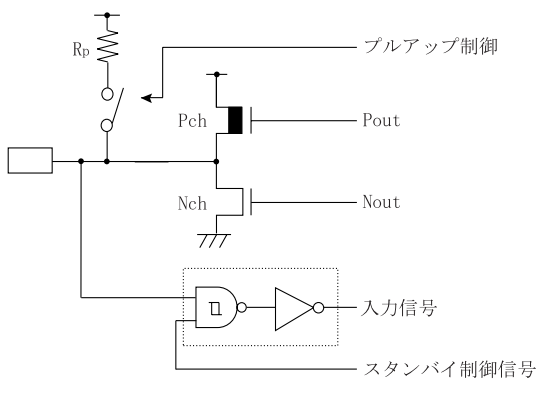
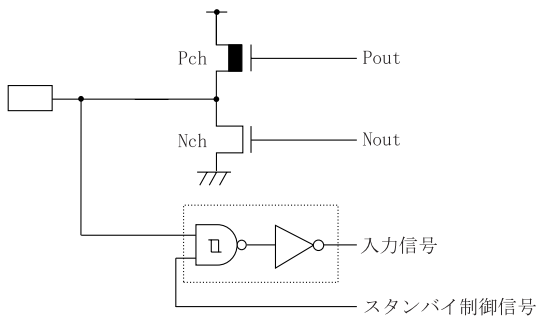
分類	回路	備考
A		・ 発振回路 発振帰還抵抗(R_f)内蔵
B		・ CMOSヒステリシスリセット入力端子
C		・ プルアップコントロールつき CMOSヒステリシス入出力端子 CMOS出力 CMOSヒステリシス入力 （スタンバイ時入力遮断機能つき） プルアップ抵抗(R_p)内蔵 <注記> ・ プルアップ抵抗は，ポートが入力状態の場合に有効となります。
D		・ CMOSヒステリシス入出力端子 CMOS出力 CMOSヒステリシス入力 （スタンバイ時入力遮断機能つき） <注記> ・ 入出力ポートの出力と内蔵リソースの出力は，ひとつの出力バッファを共有しています。 ・ 入出力ポートの入力と，内蔵リソースの入力は，ひとつの入力バッファを共有しています。

表 1.7-1 入出力回路形式（続き）

分類	回路	備考
E		<ul style="list-style-type: none"> CMOS入出力端子 CMOS出力 CMOSヒステリシス入力 (スタンバイ時入力遮断機能つき)
F		<ul style="list-style-type: none"> アナログ / CMOSヒステリシス入出力端子 CMOS出力 CMOSヒステリシス入力 (スタンバイ時入力遮断機能つき) アナログ入力 (アナログ入力許可レジスタ (ADER) の対応するビットが "1" の場合は, A/Dコンバータのアナログ入力が有効になります) 入出力ポートの出力と, 内蔵リソースの出力は, ひとつの出力バッファを共有しています。 入出力ポートの入力と, 内蔵リソースの入力は, ひとつの入力バッファを共有しています。
G		<ul style="list-style-type: none"> A/Dコンバータ (AVR)電圧入力端子

1.8 デバイスの取扱いに関する注意事項

デバイスを取り扱う際には、次の9項目に関して特に注意が必要です。

- ・ 最大定格電圧の厳守（ラッチアップの防止）
- ・ 供給電圧の安定化
- ・ 電源投入時
- ・ 未使用入力端子の処理
- ・ A/Dコンバータの電源端子処理
- ・ 外部クロック使用時
- ・ 電源端子
- ・ A/Dコンバータの電源アナログ入力投入順序
- ・ 「DIV A,Ri」, 「DIVW A,RWi」命令の使用上の注意

デバイスの取扱いに関する注意事項

最大定格電圧の厳守（ラッチアップの防止）

MB90560/565シリーズの入力端子と出力端子に、 V_{CC} より高い電圧および V_{SS} より低い電圧を印加しないでください。また、 $V_{CC} \sim V_{SS}$ 間に定格を超える電圧を印加しないでください。定格を超える電圧を印加した場合、ラッチアップ現象が発生します。ラッチアップを起こした場合、素子が熱破壊を起こします。

アナログ電源（ AV_{CC} , AVR ）とアナログ入力（ $AN0 \sim AN7$ ）を電源投入する場合、および電源を切断する場合は、デジタル電源（ V_{CC} ）を超えないようにしてください。

供給電圧の安定化

V_{CC} 電源電圧の急激な変化があると誤動作を起こすことがありますので、 V_{CC} 電源電圧を安定させてください。電源電圧安定の基準として、商用周波数（50～60Hz）での V_{CC} リプル変動（Peak to Peak値）は、 V_{CC} 電源電圧値の10%以下、電源の切替え時での変化は、過渡変動率が0.1V/ms以下になるように、電圧変動を抑えてください。

電源投入時

内蔵している降圧回路の誤動作を防ぐために、電源投入時の電圧立上げ時間は、50 μ s (0.2V～2.7Vの間)以上としてください。

未使用入力端子の処理

使用していない入力端子を開放状態にしておくと、誤動作及びラッチアップによる永久破壊の原因になることがありますので、2k Ω 以上の抵抗を介して、プルアップ、または、プルダウンの処置をしてください。

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解放とするか、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。

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は、 $AV_{CC}=AVR=V_{CC}$, $AV_{SS}=V_{SS}$ となるよう接続してください。

外部クロック使用時

外部クロックを使用する場合においても、振動子を接続した場合と同様に、パワーオンリセットおよびストップモードから復帰する場合には、発振安定待ち時間をとります。外部クロックを使用する場合には、X0端子を駆動し、X1端子はオープンとしてください。

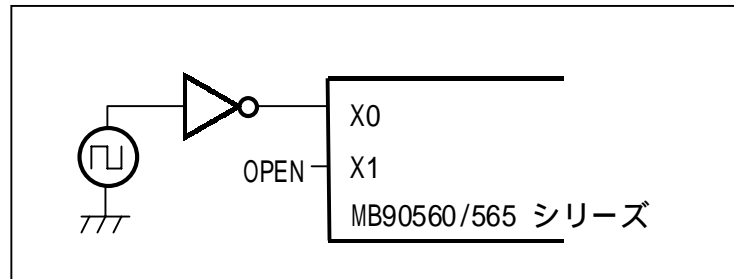


図 1.8-1 外部クロック使用例

電源端子

ラッチアップを防止するために、複数あるVcc,Vss電源端子はデバイス内部で接続しています。しかし、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プ信号の誤動作防止,総出力電流規格を守るために、必ずVcc,Vss電源端子は、外部で同一電源へ接続してください。

また、電流供給源から低インピーダンスで、デバイスのVcc,Vss電源端子に接続してください。デバイスのVcc,Vss電源端子間に0.1 μ F程度のバイパスコンデンサを、Vcc,Vss電源端子の近くに接続することで対策できます。

電源の投入および、切断順序

A/Dコンバータの電源端子(AVcc,AVR,AVss)とアナログ入力端子(AN0~AN7)への電圧印加は、必ずデジタル電源(Vcc)の投入後に行ってください。デバイスの電源切断は、A/Dコンバータの電源、およびアナログ入力電源を切断した後に、デジタル電源(Vcc)を切断してください。AVRは、AVccを超えないように電圧の印加と切断を行ってください。

アナログ入力と兼用している端子を入力ポートとして使用する場合には、入力電圧がAVccを超えないようにしてください（アナログ電源とデジタル電源に電源電圧を同時に印加・切断することは問題ありません）。

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 $\overline{\text{RST}}$ 端子が"H"の場合、ポート0, 1から不定を出力します。RST 端子が"L"の場合、ポート0, 1はハイ・インピーダンス状態になります。

タイミングは、図 1.8-2、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：MB90F562/B, MB90V560）

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：MB90561/A, MB90562/A）

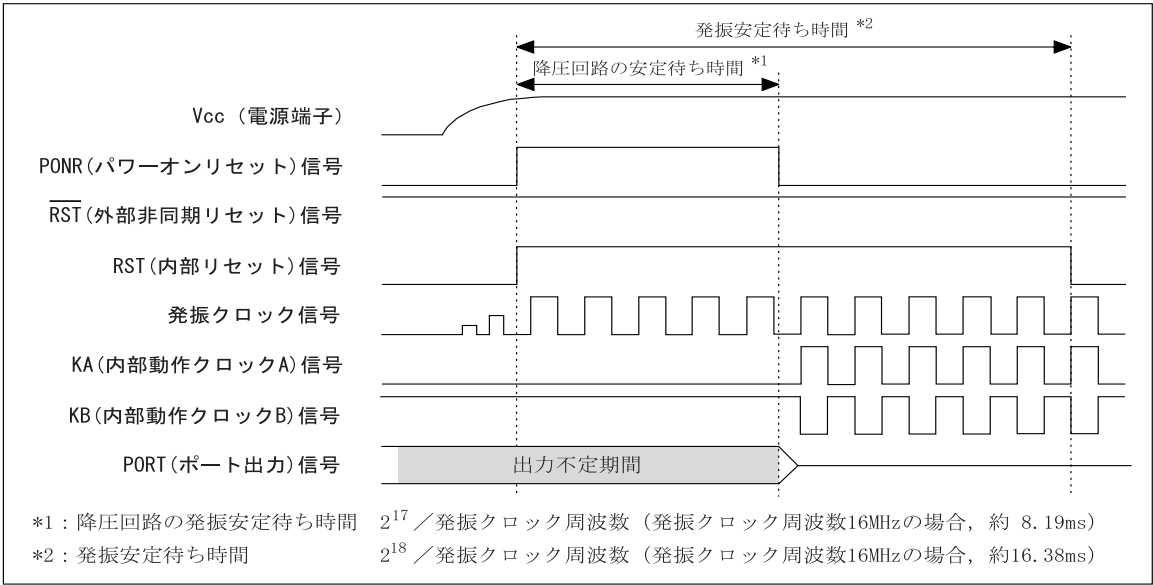


図 1.8-2 ポート0, 1が不定出力になるタイミングチャート（ $\overline{\text{RST}}$ 端子が"H"の場合）

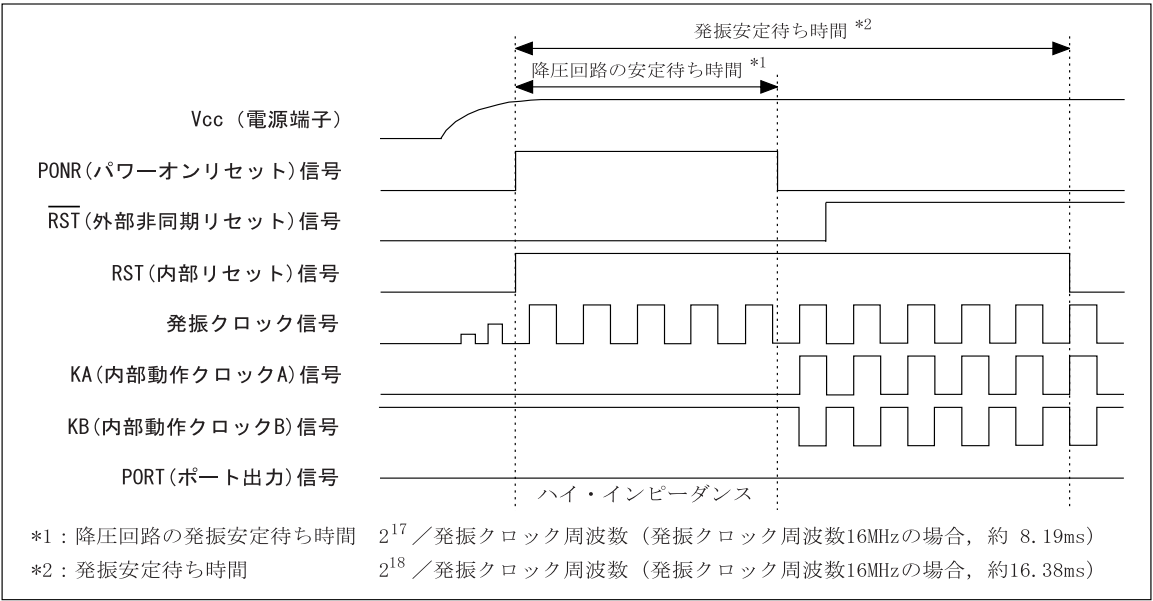


図 1.8-3 ポート0, 1がハイ・インピーダンス状態になるタイミングチャート（ $\overline{\text{RST}}$ 端子が"L"の場合）

「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意

符号付乗除算命令「DIV A,Ri」,「DIVW A,RWi」の,命令実行結果により得られる余りが,バンクレジスタの影響を受け,バンクレジスタで設定されたメモリバンクのあるアドレスに格納されます。

「DIV A,Ri」,「DIVW A,RWi」命令は,バンクレジスタを"00H"にして使用してください。

参考:

詳細は,「2.9.5 「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意」を参照してください。

バンクレジスタの詳細は,「2.7.9 バンクレジスタ (PCB,DTB,USB,SSB,ADB)」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。

第2章 CPU

この章では、MB90560/565シリーズのCPUの機能と動作について説明します。

- 2.1 CPUの概要
- 2.2 メモリ空間
- 2.3 メモリマップ
- 2.4 アドレッシング
- 2.5 多バイト長データの配置
- 2.6 レジスタ
- 2.7 専用レジスタ
- 2.8 汎用レジスタ
- 2.9 プリフィックスコード

2.1 CPUの概要

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16ビットCPUです。F²MC-16LXの命令セットはコントローラ用途向けに設計されており、各種制御の高速・高効率処理が可能です。

F²MC-16LX CPUコアは、内部32ビットアキュムレータを搭載していますので32ビット処理も可能です。メモリ空間は、最大16Mバイト、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およびF²MC-16LのATアーキテクチャをベースに、C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。次に、F²MC-16LX CPUの特徴を示します。

CPUの概要

最小命令実行時間：62.5ns（4MHz発振・4逓倍）

最大メモリ空間：16Mバイト、リニア／バンク方式にてアクセス

命令体系

- データタイプ：ビット／バイト／ワード／ロングワード
- アドレッシングモード：23種類
- 32ビットアキュムレータ採用による演算精度の強化
- 符号付き乗除算・拡張RETI命令

割込み機能

8つのプライオリティレベル（プログラマブル）

CPUに依存しない自動転送機能

最大16チャンネルまでの拡張インテリジェントI/Oサービス

C言語／マルチタスクに対応した命令体系

システムスタックポイントの採用／命令セットの対称性／バレルシフト命令

実行速度の向上：4バイトのキュー

< 注意事項 >

MB90560/565シリーズでは、シングルチップモードの使用となるため、内蔵ROM、内蔵RAM、内蔵周辺の空間しかアクセスできません。

2.2 メモリ空間

F²MC-16LXのI/O，プログラム，およびデータは，16Mバイトのメモリ空間に配置します。RAM領域は，拡張インテリジェントI/Oサービス（EI²OS）ディスクリプタ，汎用レジスタ，およびベクタテーブルに使用されます。

メモリ空間

I/O，プログラム，およびデータは，F²MC-16LX CPUが持つ16Mバイトのメモリ空間に配置され，CPUは24ビットのアドレスバスでメモリ空間のアドレスを示し，各内蔵周辺機能（リソース）にアクセスすることができます。

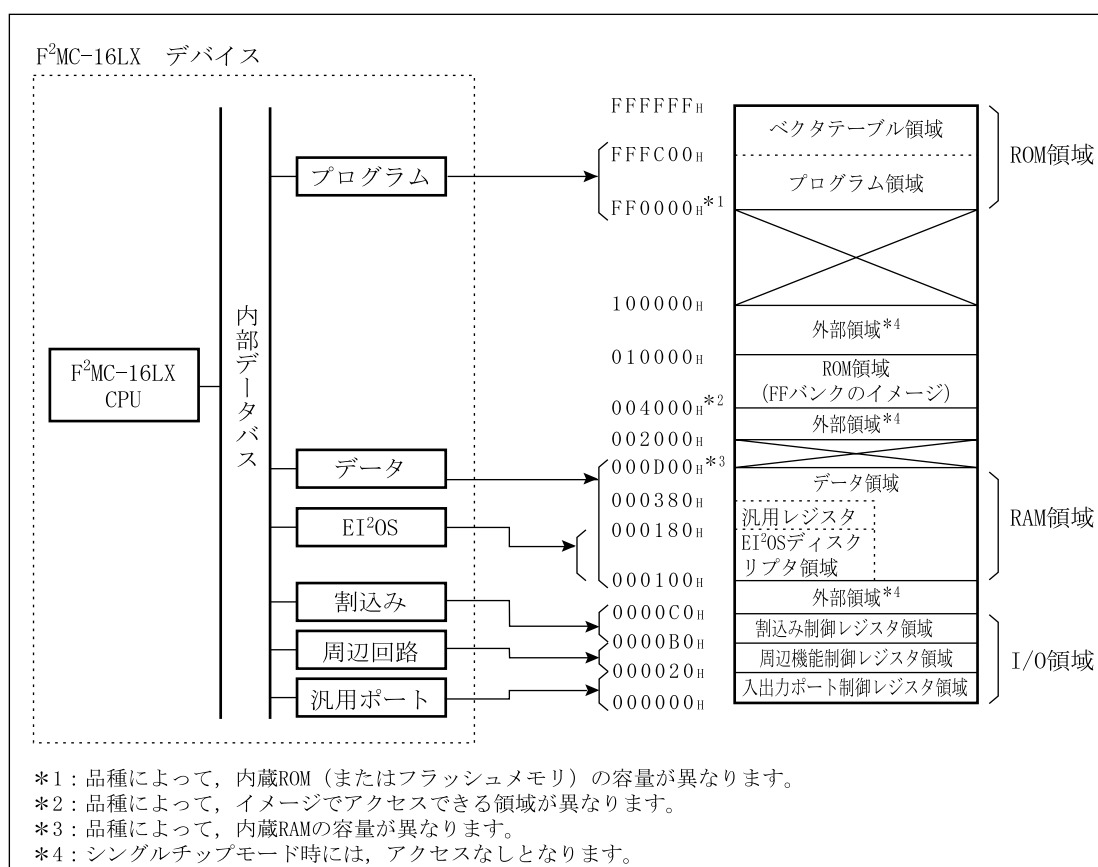


図 2.2-1 MB90560/565シリーズシステムとメモリマップの関係例

ROM領域

ベクタテーブル領域（アドレス："FFFC00H～FFFFFFH"）

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

プログラム領域（アドレス："～FFFBFFH"）

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

RAM領域

データ領域（アドレス："000100H～"）

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

汎用レジスタ領域（アドレス："000180H～00037FH"）

- 8ビット，16ビット，32ビットの演算や転送に使用する，汎用レジスタが配置されています。
- 汎用レジスタとして使用しない場合は，通常のRAMとして使用することができます。
- 汎用レジスタとして使用した場合は，汎用レジスタアドレッシングを使用することができますので，短い命令サイクル数でアクセスできます。

拡張インテリジェントI/Oサービス（EI²OS）ディスクリプタ領域
（アドレス："000100H～00017FH"）

- 拡張インテリジェントI/Oサービス（EI²OS）の転送モード，I/Oのアドレス，転送数，およびバッファアドレスを設定します。
- 拡張インテリジェントI/Oサービス（EI²OS）を使用しない場合は，通常のRAMとして使用することができます。

I/O領域

割込み制御レジスタ領域（アドレス："0000B0H～0000BFH"）

割込み制御レジスタ（ICR00～ICR15）は，割込み機能を持つ内蔵周辺機能（リソース）に対応しています。割込みレベルの設定，および拡張インテリジェントI/Oサービス（EI²OS）の設定ができます。

周辺機能制御レジスタ領域（アドレス："000020H～0000AFH"）

内蔵周辺機能（リソース）の設定ができます。

入出力ポート制御レジスタ領域（アドレス："000000H～00001FH"）

入出力ポートの設定ができます。

2.3 メモリマップ

MB90560/565シリーズの品種ごとのメモリマップを示します。

メモリマップ

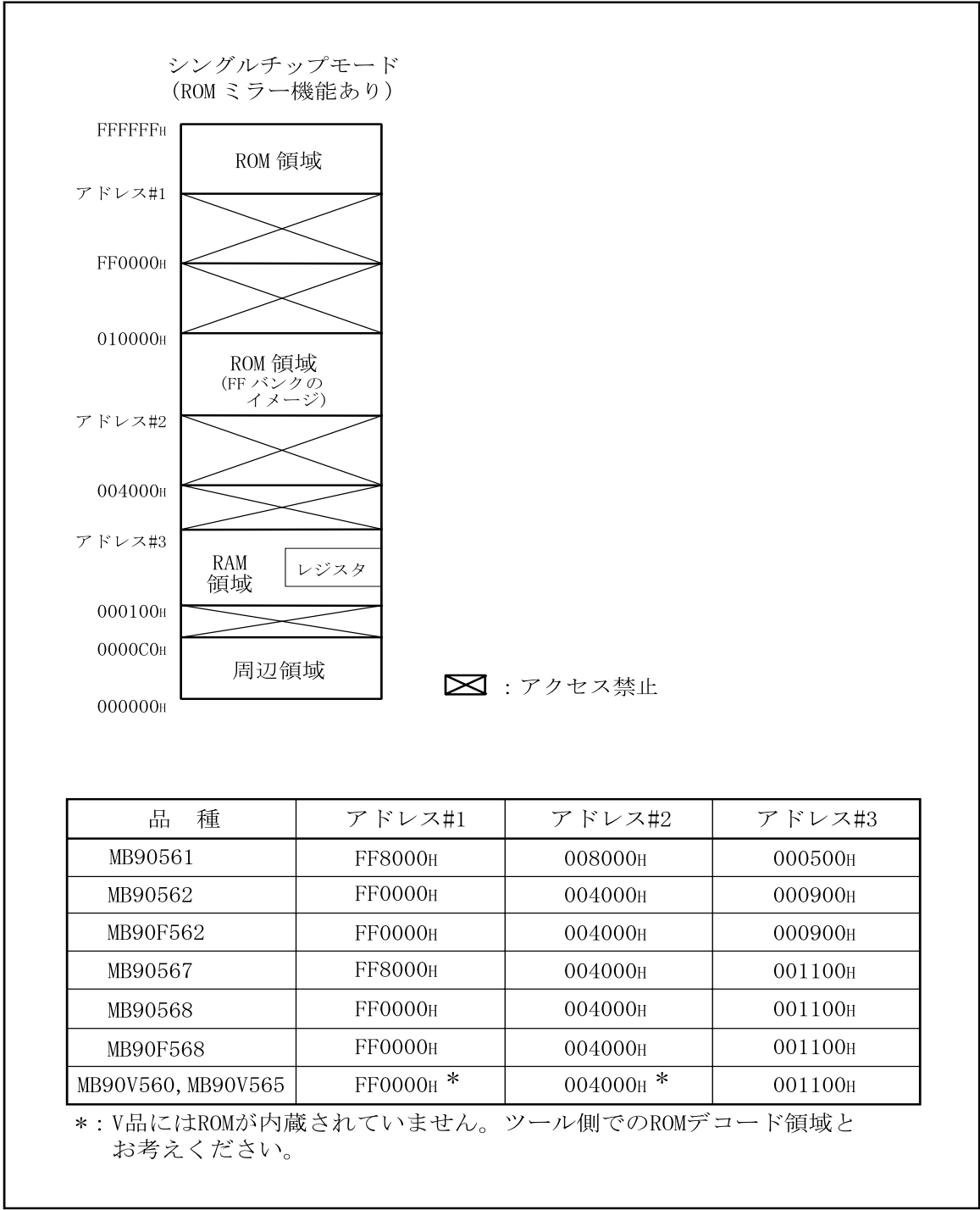


図 2.3-1 MB90560/565シリーズのメモリマップ

< 注意事項 >

- ROMミラー機能レジスタの設定を行った場合、00バンクの上位側 ("004000_H ~ 00FFFF_H") にFFバンクの上位側 ("FF4000_H ~ FFFFFF_H") のデータがミラーイメージで見えるようになります。
- ROMミラー機能の設定は、「第18章 ROMミラー機能選択モジュール」を参照してください。

参考：

ROMミラー機能は、Cコンパイラのスモールモデルを使用するためのものです。

FFバンクの下位16ビットアドレスは、00バンクの下位16ビットアドレスと同じになります。ただし、FFバンクのROM領域は48Kバイトを越えますので、00バンクには、ROM領域の全データをミラーイメージで見せることはできません。

Cコンパイラのスモールモデルをご使用の場合は、データテーブルを"FF4000_H ~ FFFFFF_H"に格納しておくことで、"004000_H ~ 00FFFF_H"にミラーイメージでデータテーブルを見せることができます。したがって、ポインタでfar指定を宣言することなく、ROM領域内のデータテーブルを参照することができます。

2.4 アドレッシング

アドレス生成方式には、リニア方式とバンク方式があります。

リニア方式は、16Mバイトの空間を連続した24ビットアドレスにより直接指定する方法です。

バンク方式は、16Mバイトの空間を64Kバイトごとの256バンクに分割し、上位8ビットアドレスをバンクレジスタにより指定し、下位16ビットアドレスを命令で直接指定する方法です。

F²MC-16LXシリーズは、命令によりアドレス生成方式が異なります。

リニアアドレッシングとバンクアドレッシング

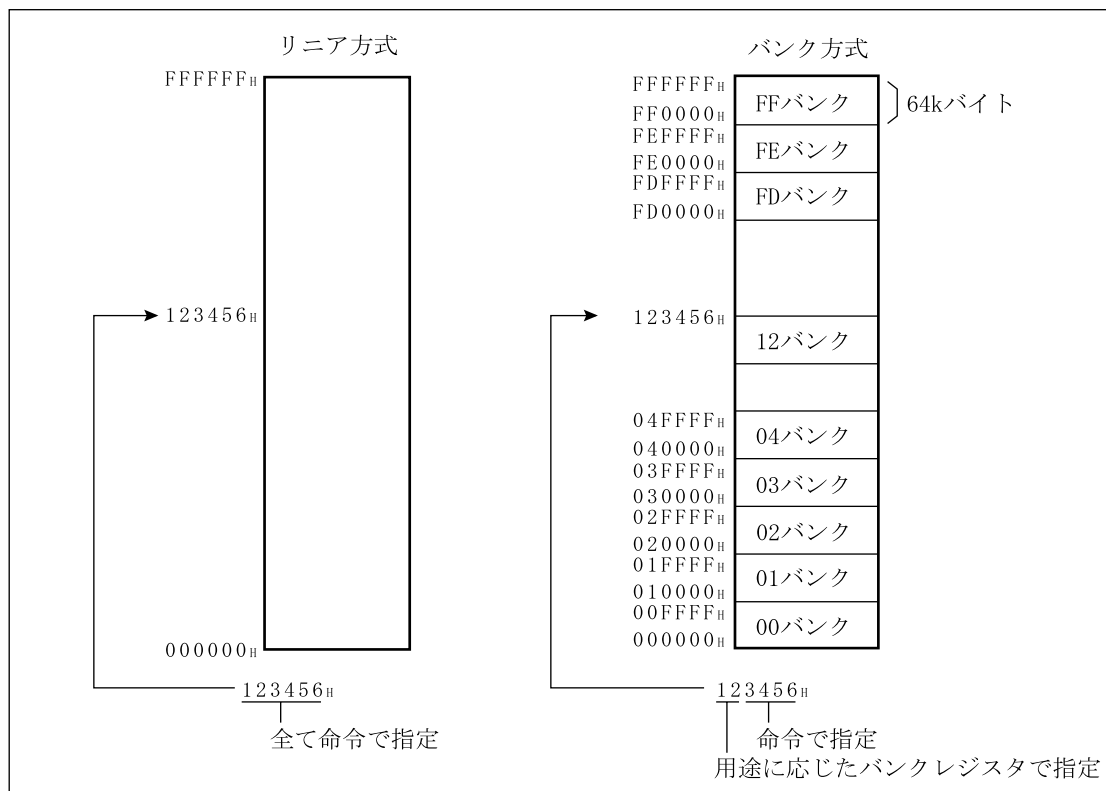


図 2.4-1 リニア方式とバンク方式のメモリ管理

2.4.1 リニア方式によるアドレス指定

リニア方式によるアドレス指定には、オペランドで直接24ビットのアドレスを指定する方法と、32ビットの汎用レジスタの下位24ビットをアドレスとして引用する方法があります。

24ビットオペランド指定によるリニアアドレッシング

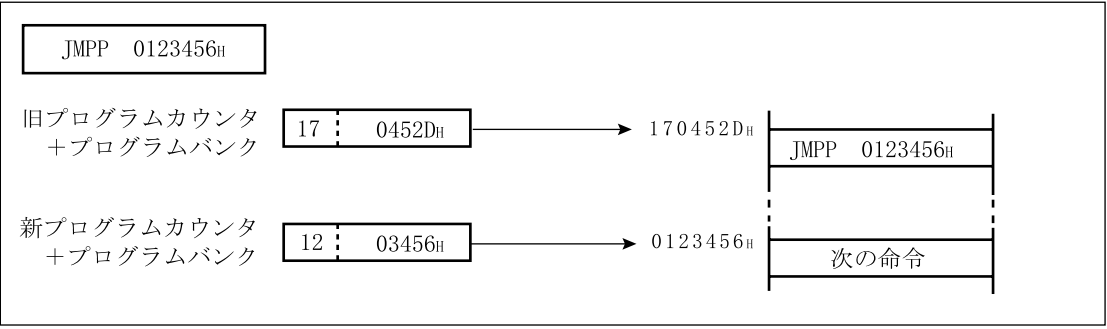


図 2.4-2 リニア方式による24ビット物理直接アドレス指定例

32ビットレジスタ間接指定によるアドレッシング

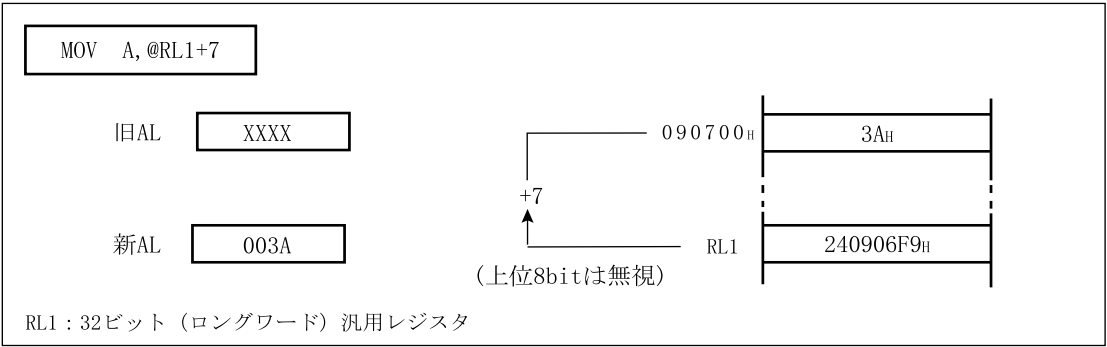


図 2.4-3 リニア方式による32ビット汎用レジスタ間接指定例

2.4.2 バンク方式によるアドレス指定

バンク方式によるアドレス指定は、16Mバイトのメモリ空間を64Kバイトごとの256バンクに分割し、バンクレジスタでアドレスの上位8ビットを指定します。下位16ビットアドレスは命令で直接指定します。

バンクレジスタには、用途別に以下に示す5種類があります。

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

バンクレジスタとアクセス空間

表 2.4-1 各バンクレジスタのアクセス空間と主な用途

バンクレジスタ名	アクセス空間	主な用途	リセット時 初期値
プログラムバンクレジスタ （PCB）	プログラム （PC）空間	命令コード、ベクタテーブル、即 値データを格納します。	FF _H
データバンクレジスタ （DTB）	データ（DT） 空間	読み書き可能なデータの格納や、 内外周辺の制御レジスタ／データ レジスタのアクセスを行います。	00 _H
ユーザスタックバンクレジスタ （USB）	スタック （SP）空間	PUSH / POP命令や、割込みのレジス タ退避などのスタックアクセスに 用いられる領域です。 コンディションレジスタ内のス タックフラグ（CCR：S）が、"1"の 場合にSSBを、"0"の場合にUSBを使 用します。*	00 _H
システムスタックバンクレジスタ （SSB）*			00 _H
アディショナルバンクレジスタ （ADB）	アディショナル （AD）空間	データ（DT）空間に入りきらない データを格納します。	00 _H

*：割込み時のスタックには、必ずSSBが使用されます。

バンクレジスタの詳細は、「2.7.9 バンクレジスタ（PCB,DTB,USB,SSB,ADB）」
を参照してください。

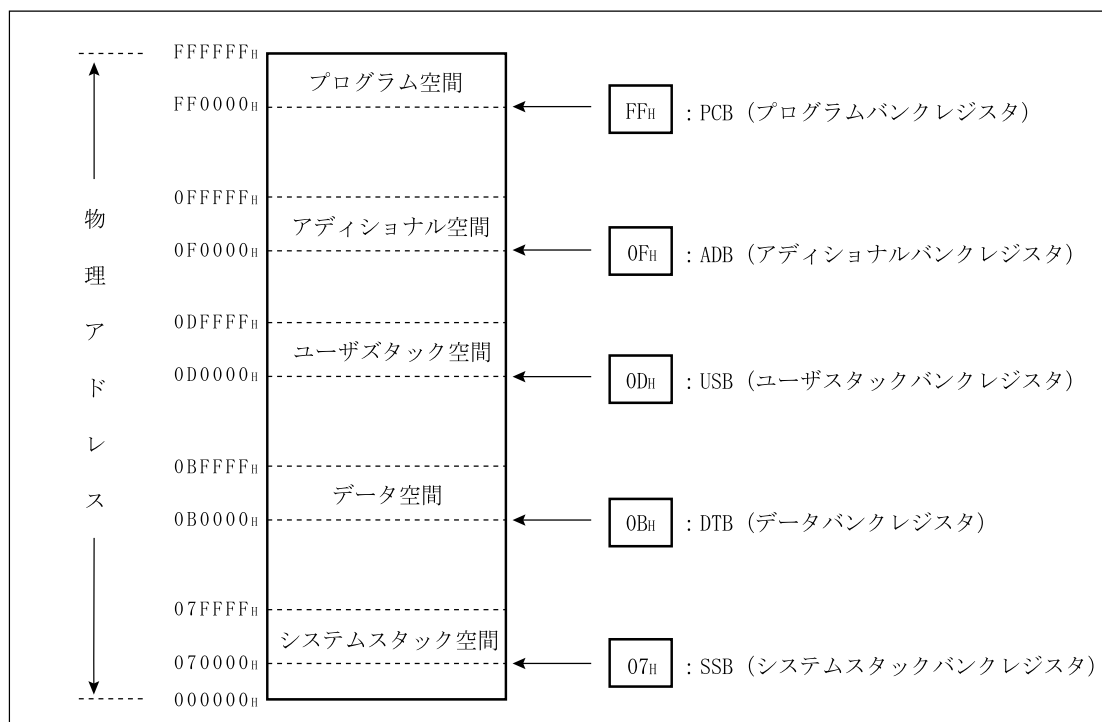


図 2.4-4 バンク方式によるアドレッシング例

バンクアドレッシングとデフォルト空間

命令のコード効率向上のため、各命令には、アドレス指定方式ごとに、表 2.4-2 に示すようなデフォルト空間が決められています。デフォルト以外の空間を使用したい場合は、各バンクに対応しているプリフィックスコードを命令に先行して設定することで、プリフィックスコードに対応したバンク空間をアクセスすることができます。プリフィックスコードの詳細は、「2.9 プリフィックスコード」を参照してください。

表 2.4-2 アドレッシングとデフォルト空間

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	@RWO, @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

2.5 多バイト長データの配置

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

メモリ上の多バイトデータの格納状態

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

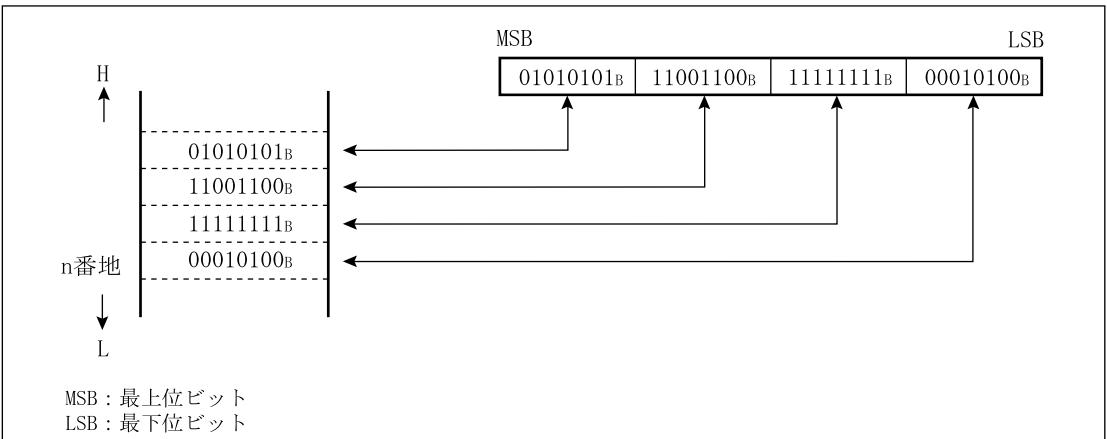


図 2.5-1 メモリ上の多バイト長データ構成

多バイト長オペランドの場合の構成

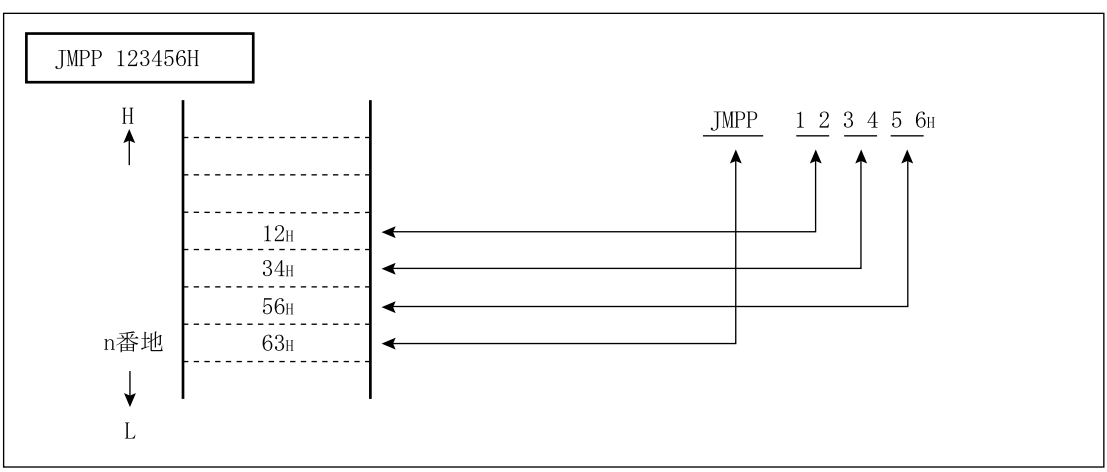


図 2.5-2 メモリ上の多バイト長オペランド構成

スタック上の多バイトデータの構成

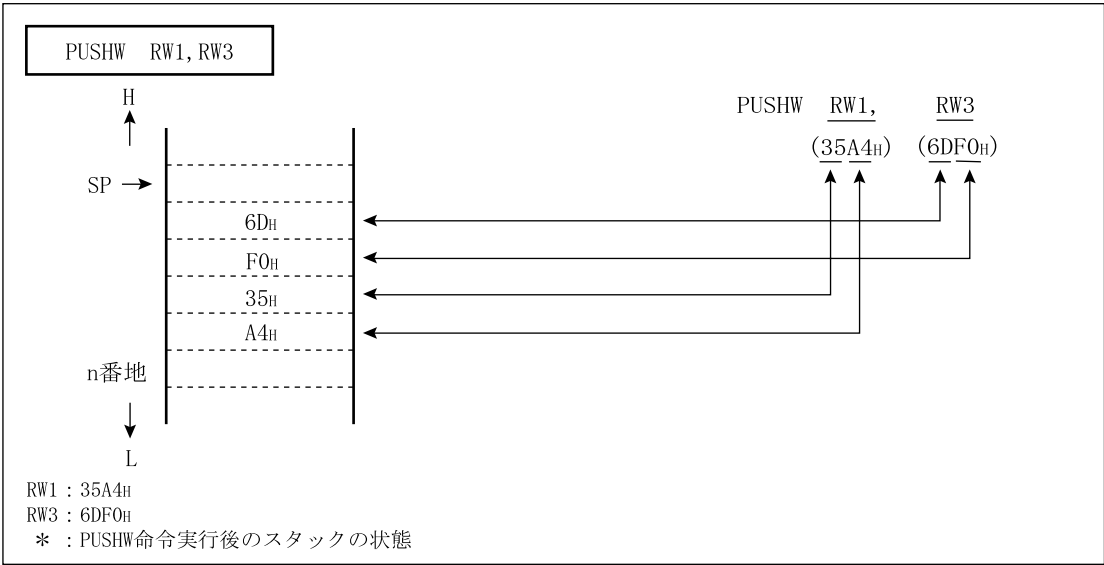


図 2.5-3 スタック上の多バイト長データ構成

多バイト長データのアクセス

多バイト長データのアクセスは、バンク内を基本に行われます。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。

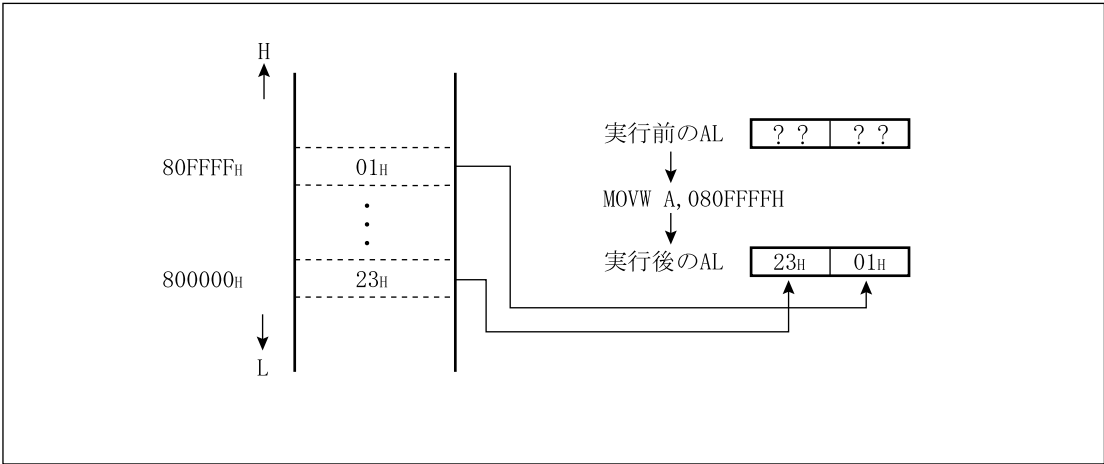


図 2.5-4 バンクアドレス境界での多バイト長データアクセス

2.6 レジスタ

F²MC-16LXのレジスタには，CPUに内蔵される専用レジスタと，内蔵RAM上に配置される汎用レジスタがあります。

専用レジスタと汎用レジスタ

専用レジスタは，CPUに内蔵されているハードウェアです。
汎用レジスタは，CPUのアドレス空間上にRAMと共存します。

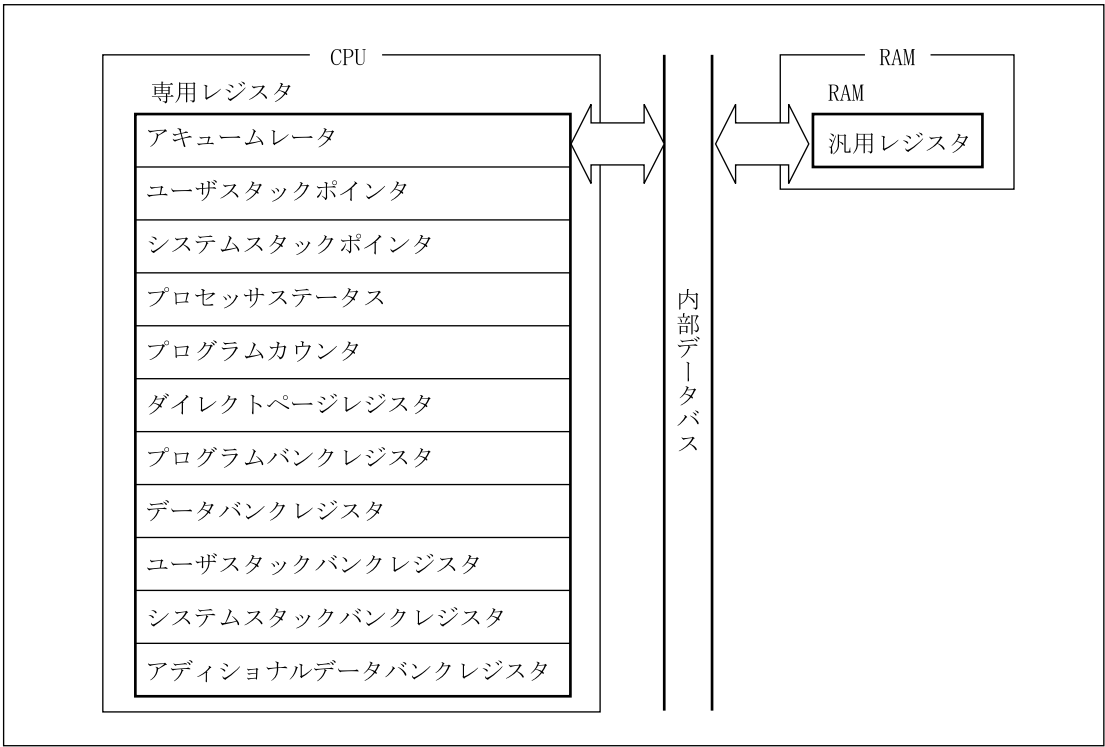


図 2.6-1 専用レジスタと汎用レジスタ

2.7 専用レジスタ

専用レジスタは、以下に示す11種類のレジスタで構成されています。

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

専用レジスタの構成

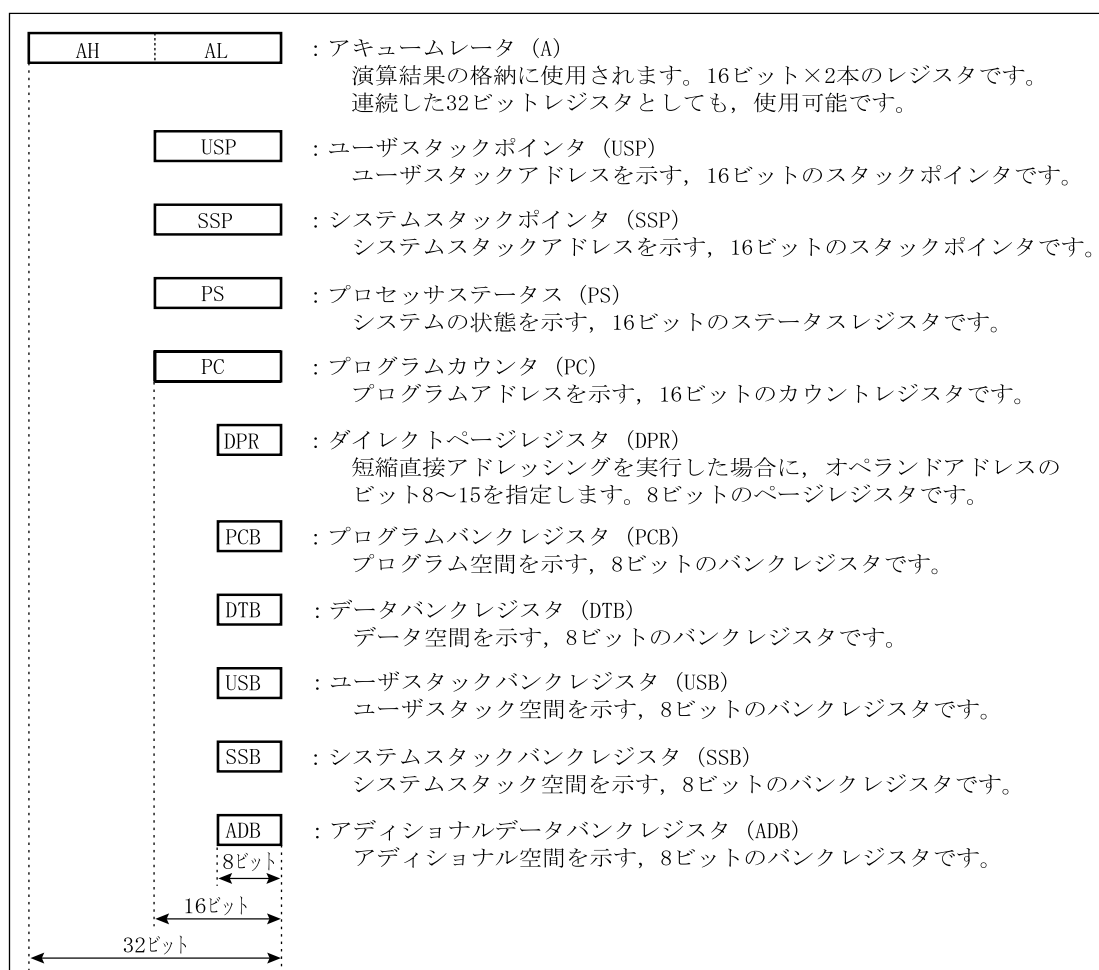


図 2.7-1 専用レジスタの構成

表 2.7-1 専用レジスタの初期値

専用レジスタ	初期値
アキュムレータ (A)	不定
ユーザスタックポインタ (USP)	不定
システムスタックポインタ (SSP)	不定
プロセッサステータス (PS)	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">PS</div> <div style="border: 1px solid black; padding: 2px; text-align: center;"> <div style="display: flex; justify-content: space-between; font-size: 0.8em;"> bit15 ~ bit13 bit12 ~ bit8 bit7 ~ bit0 </div> <div style="display: flex; justify-content: space-between; font-size: 0.8em;"> ILM RP CCR </div> <div style="display: flex; justify-content: space-between; font-size: 0.7em;"> 0 0 0 0 0 0 0 0 x 0 1 x x x x x </div> </div> </div>
プログラムカウンタ (PC)	リセットベクタ中の値 (FFFFDC _H , FFFFDD _H の内容)
ダイレクトページレジスタ (DPR)	01 _H
プログラムバンクレジスタ (PCB)	リセットベクタ中の値 (FFFFDE _H の内容)
データバンクレジスタ (DTB)	00 _H
ユーザスタックバンクレジスタ (USB)	00 _H
システムスタックバンクレジスタ (SSB)	00 _H
アディショナルデータバンクレジスタ (ADB)	00 _H

2.7.1 アキュムレータ (A)

アキュムレータ (A) は、2つの16ビット長演算用レジスタ (AH / AL) から構成されています。演算結果やデータの一時記憶に使用されるレジスタです。

アキュムレータ (A) は、32/16/8ビットのレジスタとして使用できます。メモリと他のレジスタ間、あるいは上位側の16ビット長演算レジスタ (AH) と下位側の16ビット長演算レジスタ (AL) 間で、演算ができます。また、ワード長以下のデータを下位側の16ビット長演算レジスタ (AL) へ転送した場合、転送前の下位側の16ビット長演算レジスタ (AL) に格納されているデータが、上位側の16ビット長演算レジスタ (AH) に転送されるデータ保持機能があります。(一部の命令ではデータ保持を行いません。)

アキュムレータ (A)

アキュムレータへのデータ転送

アキュムレータは、32ビット長 (ロングワード)、16ビット長 (ワード)、8ビット長 (バイト) のデータを処理することができます。例外として、4ビットデータ転送命令 (MOVN) もありますが、8ビットデータと同様に処理されず。

- 32ビットデータ処理の場合は、上位演算用レジスタ (AH) と下位演算用レジスタ (AL) が連結して使用されます。
- 16ビットデータや8ビットデータの場合は、下位演算用レジスタ (AL) が使用され、上位演算用レジスタ (AH) は、下位演算用レジスタ (AL) のデータを保持します。
- 下位演算用レジスタ (AL) へバイト長以下のデータが転送された場合は、符号拡張またはゼロ拡張され、16ビット長となり下位演算用レジスタ (AL) へ格納されます。また、下位演算用レジスタ (AL) の格納されたデータは、16ビットデータまたは8ビットデータとして扱うことができます。

具体的な転送例を図 2.7-3 ~ 図 2.7-6 に示します。

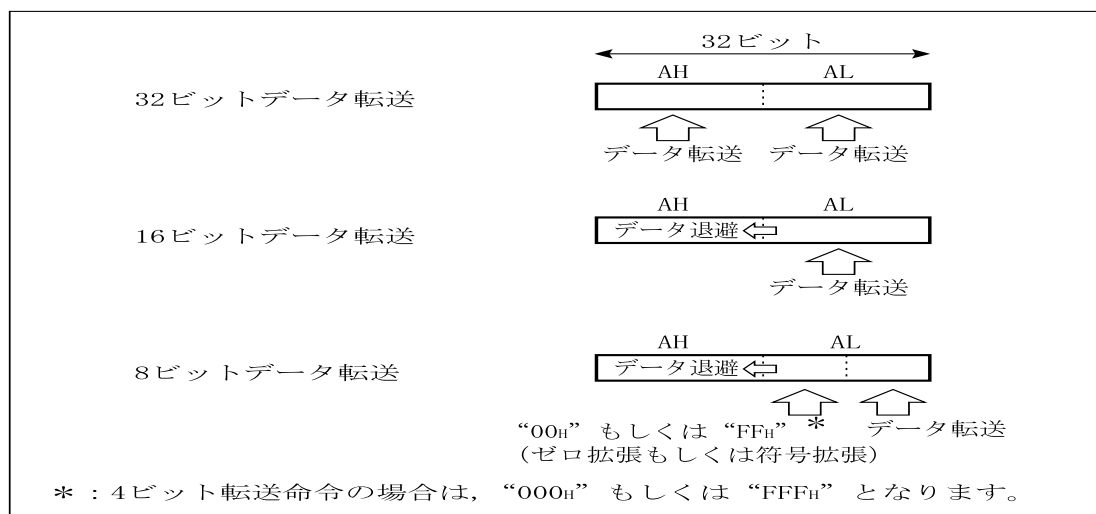


図 2.7-2 アキュムレータへのデータ転送

アキュムレータのバイト処理算術演算

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

アキュムレータの初期値

リセット後の初期値は不定です。

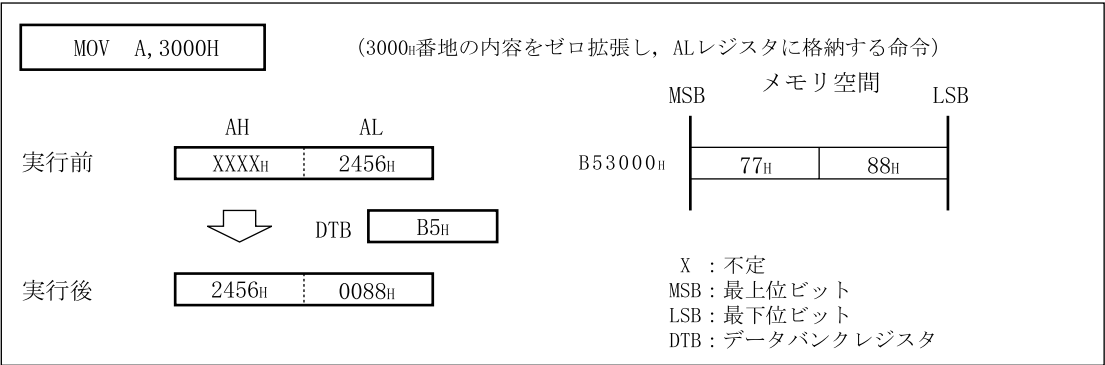


図 2.7-3 アキュムレータ（A）のAL-AH間の転送例（8ビット即値，ゼロ拡張）

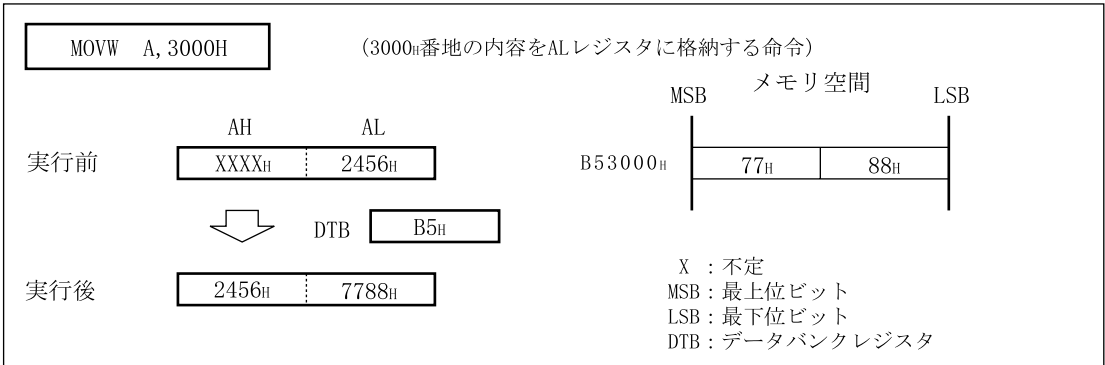


図 2.7-4 アキュムレータ（A）のAL-AH間の転送例（8ビット即値，符号拡張）

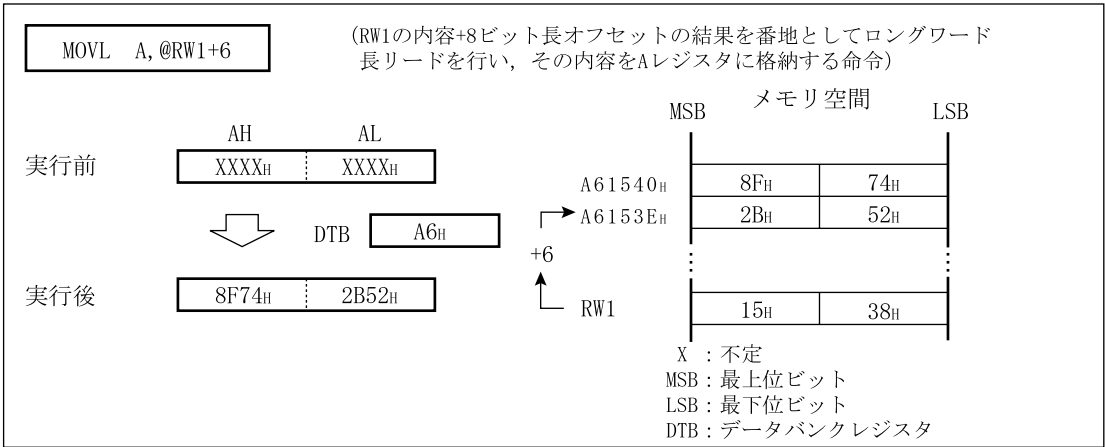


図 2.7-5 アキュムレータ（A）への32ビットデータ転送例（レジスタ間接）

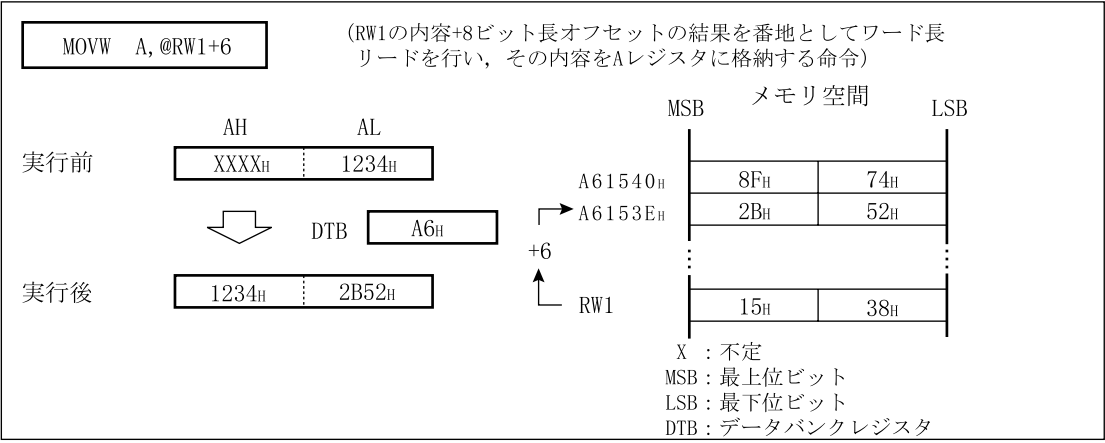


図 2.7-6 アキュムレータ (A) のAL-AH間の転送例 (16ビット , レジスタ間接)

2.7.2 スタックポインタ（USP, SSP）

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


コンディションコードレジスタ（CCR）のSフラグが、“0”の場合はUSPおよびUSBレジスタが有効になり、Sフラグが“1”の場合はSSPおよびSSBレジスタが有効になります。

スタックの設定

F²MC-16LXでは、システムスタックとユーザスタックの2種類のスタックが使用できます。スタックアドレスは、プロセッサステータスレジスタ（PS：CCR）内のSフラグによって、表 2.7-2のように決定されます。

表 2.7-2 スタックアドレスの指定

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

 : 初期値

リセットによって、Sフラグは“1”に初期化されますので、初期設定ではシステムスタックが使用されます。ただし、割込みルーチンのスタック操作は、システムスタックが使用され、割込みルーチン以外のスタック操作ではユーザスタックが使用されます。スタック空間を分けなければ、システムスタックを使用してください。

< 注意事項 >

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

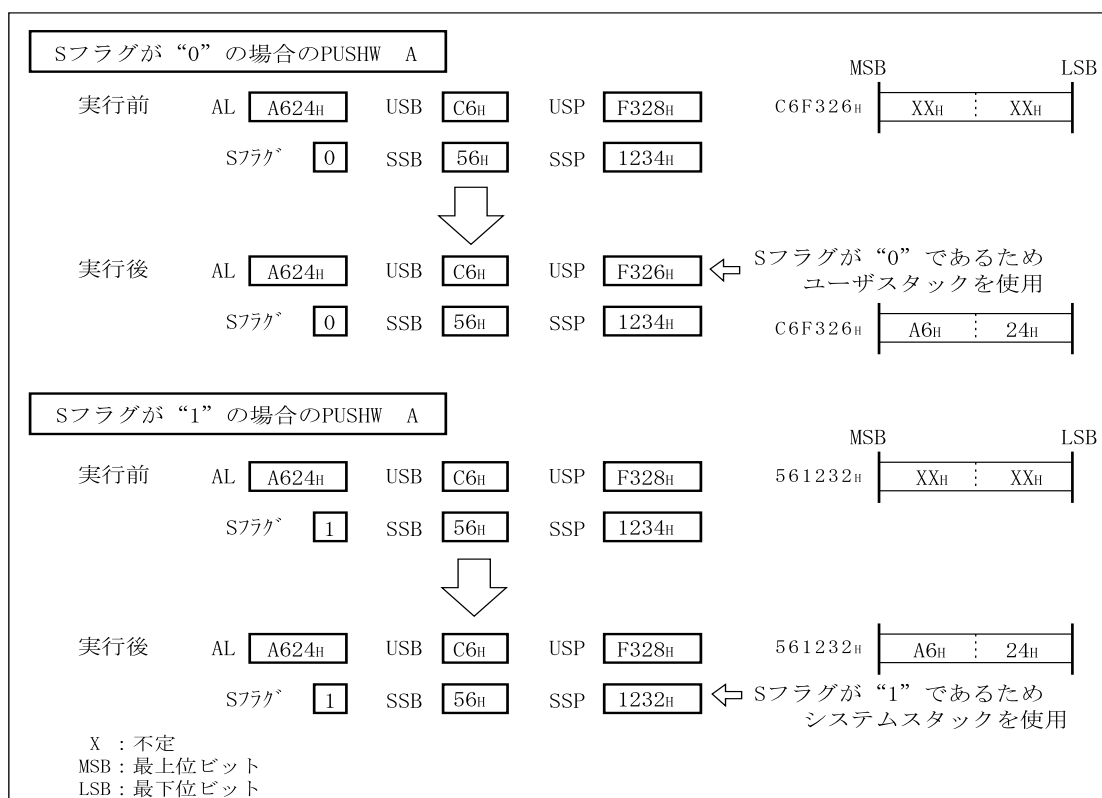


図 2.7-7 スタック操作命令とスタックポインタ

< 注意事項 >

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

システムスタックポインタ (SSP)

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

コンディションコードレジスタ (CCR) の詳細は、「2.7.4 コンディションコードレジスタ (PS : CCR)」を、システムスタックバンクレジスタ (SSB) の詳細は、「2.7.9 バンクレジスタ (PCB,DTB,USB,SSB,ADB)」を参照してください。

ユーザスタックポインタ (USP)

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

コンディションコードレジスタ (CCR) の詳細は、「2.7.4 コンディションコードレジスタ (PS : CCR)」を、システムスタックバンクレジスタ (SSB) の詳細は、「2.7.9 バンクレジスタ (PCB,DTB,USB,SSB,ADB)」を参照してください。

2.7.3 プロセッサステータス (PS)

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

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

プロセッサステータス (PS) の構成

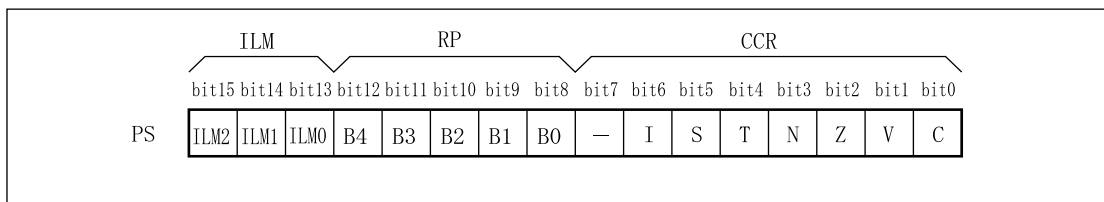


図 2.7-8 プロセッサステータス (PS) の構成

コンディションコードレジスタ (CCR)

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

各種フラグの詳細は、「2.7.4 コンディションコードレジスタ (PS : CCR)」を参照してください。

レジスタバンクポインタ (RP)

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

汎用レジスタは、32バンクあり、レジスタバンクポインタ (RP) に "00H ~ 1FH" の値をセットしてバンクを指定します。

設定方法および詳細は、「2.7.5 レジスタバンクポインタ (PS : RP)」を参照してください。

インタラプトレベルマスクレジスタ (ILM)

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

設定方法および詳細は、「2.7.6 インタラプトレベルマスクレジスタ (PS : ILM)」を参照してください。

2.7.4 コンディションコードレジスタ (PS : CCR)

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

- ・ 演算結果や転送データの内容を示すビット
- ・ 割込み要求の受付を制御するビット

コンディションコードレジスタ (CCR) の構成

命令実行時のコンディションコードレジスタ (CCR) の状態については、『F²MC-16LX プログラミングマニュアル』を参照してください。

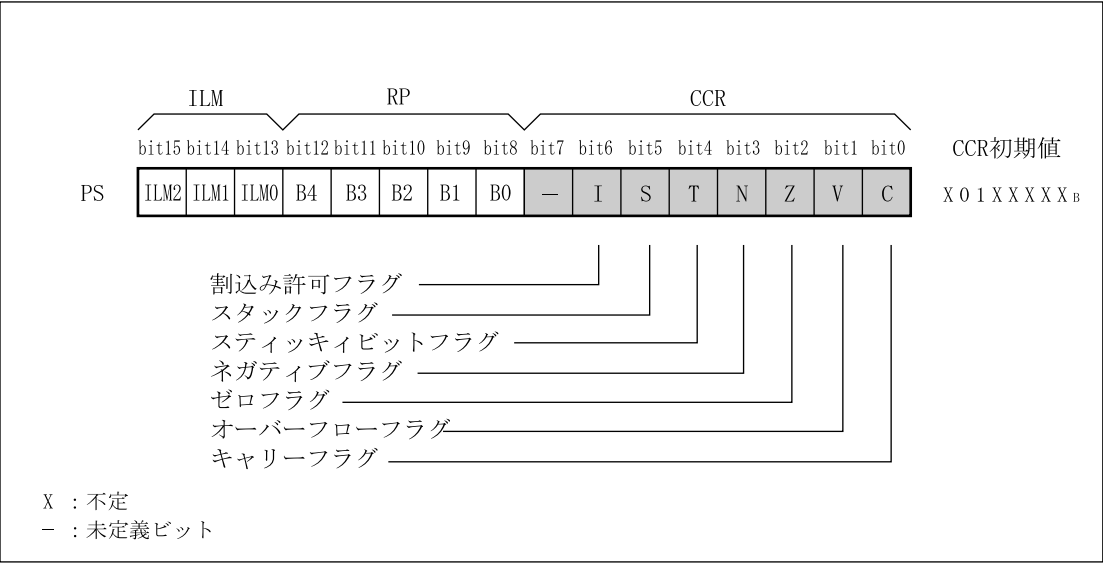


図 2.7-9 コンディションコードレジスタ (CCR) の構成

割込み許可フラグ (I)

ソフトウェア割込み以外の割込み要求に対し、割込み許可フラグ (I) に "1" がセットされた場合は割込み要求が許可され、割込み許可フラグ (I) が "0" にクリアされた場合は割込み要求が禁止されます。外部リセットおよびソフトウェアリセットにより "0" にクリアされます。

スタックフラグ (S)

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

スタックポインタの詳細は、「2.7.2 スタックポインタ (USP, SSP)」を参照してください。

スティッキビットフラグ (T)

論理右シフト命令，または算術右シフト命令を実行した場合に，キャリーからシフトアウトされたデータに"1"があればスティッキビットフラグ (T) に"1"がセットされ，"1"がなければスティッキビットフラグ (T) は"0"にクリアされます。また，シフト量がゼロの場合でも"0"にクリアされます。

ネガティブフラグ (N)

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

汎用レジスタの詳細は，「2.8 汎用レジスタ」を参照してください。

ゼロフラグ (Z)

演算結果が格納される汎用レジスタ (RL0 ~ RL3) の値が"0000H"の場合は，ゼロフラグ (Z) に"1"がセットされ，演算結果が格納される汎用レジスタ (RL0 ~ RL3) の値が"0000H"でない場合は，ゼロフラグ (Z) は"0"にクリアされます。

汎用レジスタの詳細は，「2.8 汎用レジスタ」を参照してください。

オーバーフローフラグ (V)

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

キャリーフラグ (C)

演算の実行で，最上位ビットからの桁上がり，または最上位ビットへの桁下がりが発生した場合は，キャリーフラグ (C) に"1"がセットされ，発生しなかった場合は，キャリーフラグ (C) は"0"にクリアされます。

2.7.5 レジスタバンクポインタ (PS : RP)

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

レジスタバンクポインタ (RP)

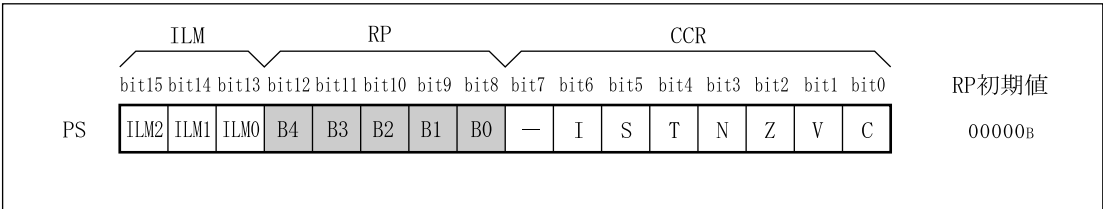


図 2.7-10 レジスタバンクポインタ (RP) の構成

汎用レジスタ領域とレジスタバンクポインタ (RP)

レジスタバンクポインタ (RP) は、F²MC-16LXの持つ汎用レジスタと、内部RAMのアドレスとの関係を示すポインタです。レジスタバンクポインタ (RP) の内容とアドレスの関係は、図 2.7-11に示す変換規則になっています。

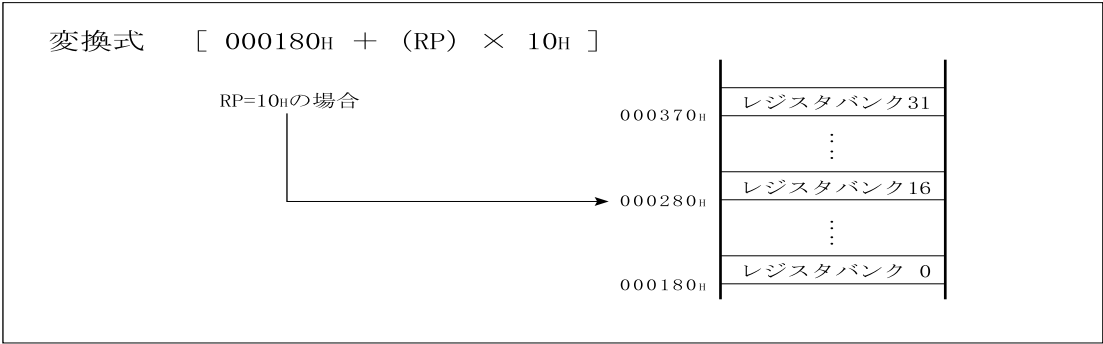


図 2.7-11 汎用レジスタ領域の物理アドレス変換規則

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

2.7.6 インタラプトレベルマスクレジスタ (PS : ILM)

インタラプトレベルマスクレジスタ (ILM) は、CPUが受付ける割込みのレベルを示す、3ビットのレジスタです。

インタラプトレベルマスクレジスタ (ILM)

割込みの詳細は、「第6章 割込み」を参照してください。

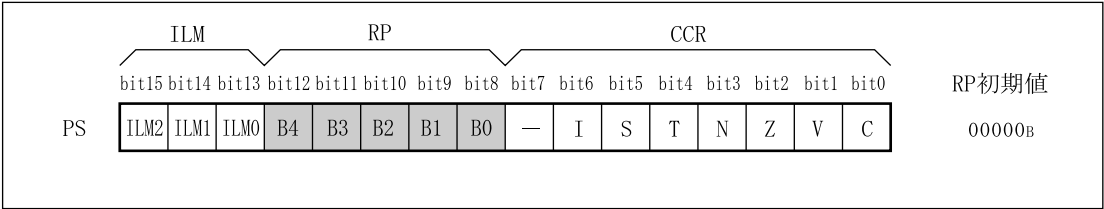


図 2.7-12 インタラプトレベルマスクレジスタ (ILM) の構成

インタラプトレベルマスクレジスタ (ILM) は、現在の割込みレベルを示します。インタラプトレベルマスクレジスタ (ILM) には、受付け可能な割込みレベルを設定することができます。インタラプトレベルマスクレジスタ (ILM) に設定された割込みレベル値より、弱い割込みレベル値のものは、割込みを受け付けません。

- インタラプトレベルマスクレジスタ (ILM) は、リセットにより割込みレベルが最強に設定され、割込みが受け付けられなくなります。
- アセンブラ命令では、インタラプトレベルマスクレジスタ (ILM) に転送する8ビットの即値転送命令が使用できます。ただし、下位3ビットが有効となります。

表 2.7-3 インタラプトレベルマスクレジスタ (ILM) と割込みレベルの強弱

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

2.7.7 プログラムカウンタ (PC)

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

プログラムカウンタ (PC)

CPUが、次に実行する命令アドレスの、上位8ビットをプログラムバンクレジスタ (PCB) で設定し、下位16ビットをプログラムカウンタ (PC) で設定します。次に実行する命令アドレスは、図 2.7-13のようになります。プログラムカウンタ (PC) は条件分岐命令、サブルーチンコール命令、割込み、リセットでも内容が更新されます。また、オペランドを読み出す場合のベースポインタとしても使用できます。

プログラムバンクレジスタ (PCB) の詳細は、「2.7.9 バンクレジスタ (PCB,DTB,USB,SSB,ADB)」を参照してください。

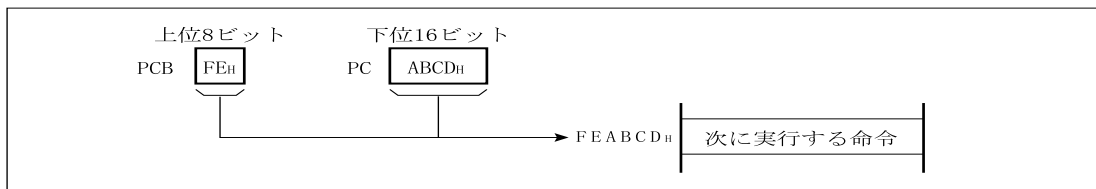


図 2.7-13 プログラムカウンタ (PC)

<注意事項>

プログラムカウンタ (PC)、プログラムバンクレジスタ (PCB) とともにプログラム (MOV PC, #0FFH などの命令) で直接書き換えることはできません。

2.7.8 ダイレクトページレジスタ (DPR)

ダイレクトページレジスタ (DPR) は、短縮直接アドレス指定方式の命令実行時に、オペランドアドレスの bit8 ~ bit15 (addr8 ~ addr15) を指定する 8 ビットレジスタです。リセットで "01H" に初期化されます。

ダイレクトページレジスタ (DPR)

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

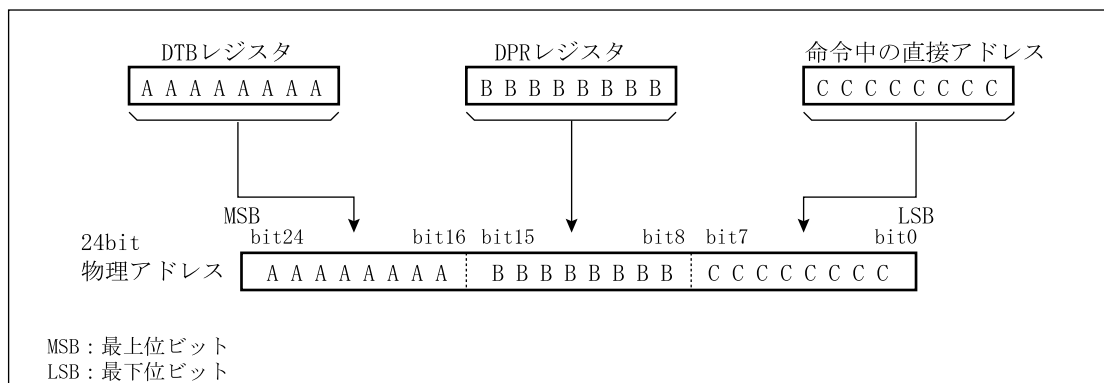


図 2.7-14 ダイレクトページレジスタ (DPR) による物理アドレスの生成

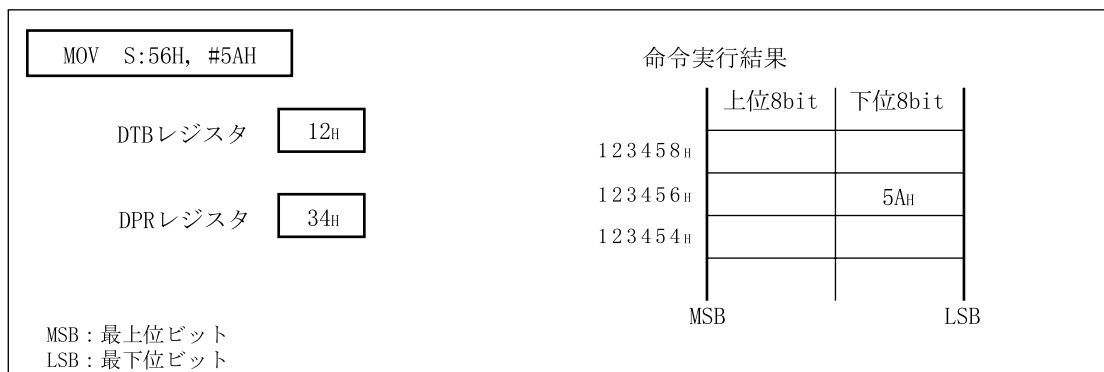


図 2.7-15 ダイレクトページレジスタ (DPR) の設定とデータアクセス例

2.7.9 バンクレジスタ (PCB,DTB,USB,SSB,ADB)

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

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

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

バンクレジスタ (PCB,DTB,USB,SSB,ADB)

プログラムバンクレジスタ (PCB)

プログラム (PC) 空間を指定するバンクレジスタです。

データバンクレジスタ (DTB)

データ (DT) 空間を指定するバンクレジスタです。

ユーザスタックバンクレジスタ (USB)

/ システムスタックバンクレジスタ (SSB)

スタック (SP) 空間を指定するバンクレジスタです。

アディショナルバンクレジスタ (ADB)

アディショナル (AD) 空間を指定するバンクレジスタです。

各バンクの設定とデータアクセス

バンクレジスタは、8ビット長であり、リセットによりプログラムバンクレジスタ (PCB) は"FF_H"に初期化され、データバンクレジスタ (DTB)、ユーザスタックバンクレジスタ (USB)、システムスタックバンクレジスタ (SSB)、アディショナルバンクレジスタ (ADB) 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、読出しはできますが書込みはできません。プログラムバンクレジスタ (PCB) 以外のバンクレジスタは、読書き可能です。

< 注意事項 >

MB90560/565シリーズは、デバイスに内蔵されるメモリ空間までのサポートになります。

各レジスタの動作の詳細は、「2.4.2 バンク方式によるアドレス指定」を参照してください。

2.8 汎用レジスタ

汎用レジスタは，"000180_H ~ 00037F_H"のRAM上に，16ビット×8本を1レジスタバンクとして割当てたメモリブロックです。汎用の8ビットレジスタ（バイトレジスタR0 ~ R7），16ビットレジスタ（ワードレジスタRW0 ~ RW7），もしくは32ビットレジスタ（ロングワードレジスタRL0 ~ RL7）として使用できます。

汎用レジスタは，短い命令で高速にRAMにアクセスできます。レジスタバンクでブロック化されているために，内容の保護や機能単位の分割が容易です。

また，ロングワードレジスタとして使用する場合は，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。

汎用レジスタの構成

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

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

レジスタバンクポインタ (PR)の詳細は，「2.7.5 レジスタバンクポインタ (PS : RP)」を参照してください。

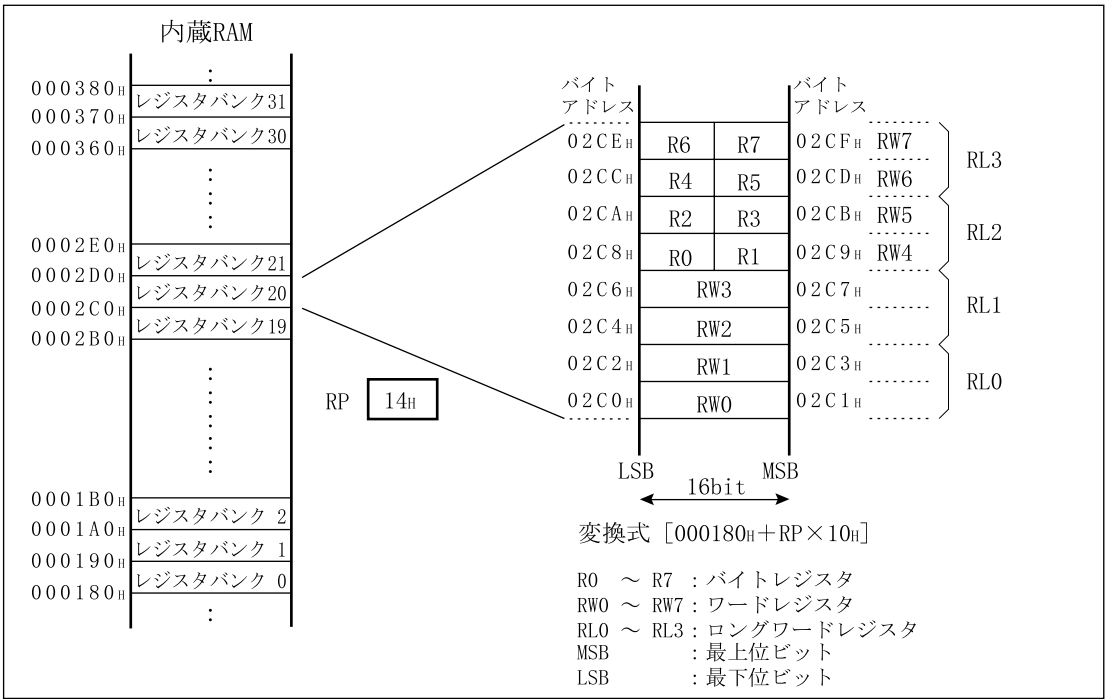


図 2.8-1 汎用レジスタのメモリ空間内での配置と構成

< 注意事項 >

レジスタバンクポインタ（RP）は，リセット後"00_H"に初期化されます。

レジスタバンク

レジスタバンク内の汎用レジスタの内容は，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，パワーオンリセット時には不定となります。

表 2.8-1 汎用レジスタの代表的な機能

レジスタ名	機 能
R0 ~ R7	各種命令のオペランドとして使用 <注記> ・R0はバレルシフトのカウンタおよびノーマライズ（正規化）命令のカウンタとしても使用
RW0 ~ RW7	ポインタとして使用 各種命令のオペランドとして使用 <注記> ・RW0はストリング命令のカウンタとしても使用
RL0 ~ RL3	ロングポインタとして使用 各種命令のオペランドとして使用

2.9 プリフィックスコード

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

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

プリフィックスコード

バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)

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

詳細は、「2.9.1 バンクセレクトプリフィックス(PCB, DTB, ADB, SPB)」を参照してください。

コモンレジスタバンクプリフィックス (CMR)

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

詳細は、「2.9.2 コモンレジスタバンクプリフィックス (CMR)」を参照してください。

フラグ変化抑止プリフィックス (NCC)

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

詳細は、「2.9.3 フラグ変化抑止プリフィックス (NCC)」を参照してください。

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

データアクセスを行う場合のメモリ空間は、アドレス指定方式ごとに定められていますが、バンクセレクトプリフィックスを命令の前に設定した場合は、アクセスするメモリ空間を、アドレス指定方式と無関係に設定できます。

バンクセレクトプリフィックス (PCB,DTB,ADB,SPB)

表 2.9-1 バンクセレクトプリフィックス

バンクセレクト プリフィックス	設定される空間
PCB	プログラム空間
DTB	データ空間
ADB	アディショナル空間
SPB	コンディションコードレジスタ (CCR) のSフラグの値が"0"の場合にユーザスタック空間が、"1"の場合にシステムスタック空間が用いられます。

バンクセレクトプリフィックスを使用した場合、例外的な動作を行う命令があります。

表 2.9-2 バンクセレクトプリフィックスの影響を受けない命令

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

表 2.9-3 バンクセレクトプリフィックス使用時に注意すべき命令

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

2.9.2 コモンレジスタバンクプリフィックス (CMR)

コモンレジスタバンクプリフィックス (CMR) を，レジスタバンクをアクセスする命令の前に設定した場合は，レジスタバンクポインタ (RP) の値に関係なく，レジスタアクセスを，"000180H ~ 00018FH"にあるコモンバンク (RP="00H"の場合に設定されるレジスタバンク) へ変更できます。

コモンレジスタバンクプリフィックス (CMR)

複数のタスク間でのデータ交換を容易にするためにF²MC-16LXでは，各タスクで共通に使用できるコモンバンクを用意しています。コモンバンクは，"000180H ~ 00018FH"番地にあります。

ただし，表 2.9-4に示した命令を使用する場合は，注意してください

表 2.9-4 コモンレジスタバンクプリフィックス (CMR) 使用時に注意すべき命令

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

2.9.3 フラグ変化抑止プリフィックス (NCC)

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

フラグ変化抑止プリフィックス (NCC)

不要なフラグ変化を抑止するために、フラグ変化抑止プリフィックス (NCC) を用います。フラグ変化が抑止されるフラグにはT, N, Z, V, Cがあります。

ただし、表 2.9-5に示した命令を使用する場合は注意してください。

T, N, Z, V, Cの各フラグの詳細は、「2.7.4 コンディションコードレジスタ (PS : CCR)」を参照してください。

表 2.9-5 フラグ変化抑止プリフィックス (NCC) の使用時に注意すべき命令

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

2.9.4 プリフィックスコードに関する制約

- プリフィックスコードの使用には、以下のような制約があります。
- ・プリフィックスコードや割込み／ホールド抑止命令の実行中は、割込み／ホールド要求を受け付けません。
 - ・割込み／ホールド命令の前にプリフィックスコードを置いた場合、プリフィックスコードの効果が遅延します。
 - ・競合するプリフィックスコードが連続した場合、最後のプリフィックスコードが有効となります。

プリフィックスコードと割込み／ホールド抑止命令

表 2.9-6 プリフィックスコードと割込み／ホールド抑止命令

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

割込み／ホールドの抑止

図 2.9-1に示すように、プリフィックスコードや割込み／ホールド命令の実行中は、割込みやホールド要求が発生しても、受け付けられません。受け付けられない場合、割込み／ホールド処理が行われるのは、プリフィックスコードや割込み／ホールド抑止命令以降、初めてプリフィックスコードや割込み／ホールド命令以外の命令が実行された後になります。

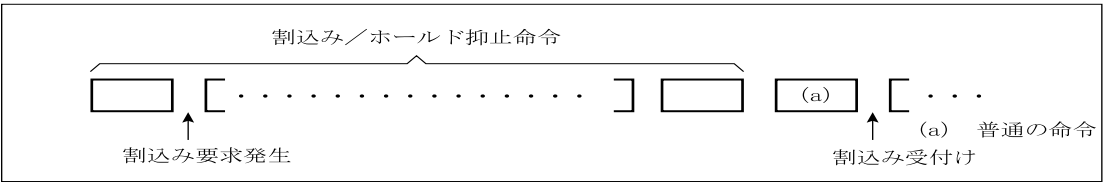


図 2.9-1 割込み／ホールドの抑止

プリフィックスコードの効果遅延

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

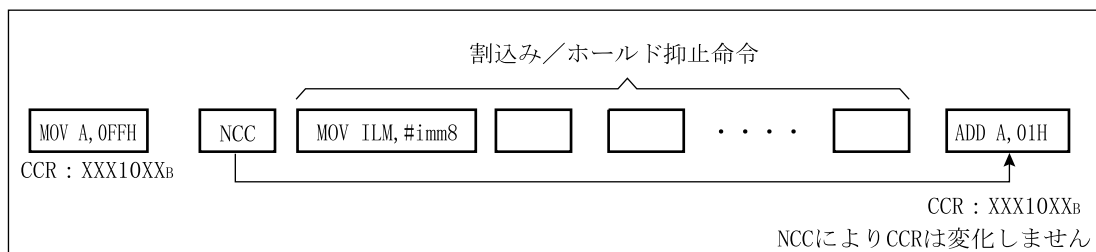


図 2.9-2 割込み／ホールド抑止命令とプリフィックスコード

プリフィックスコードの連続

競合するプリフィックスコード（PCB, ADB, DTB, SPB）が連続していた場合、最後のPCBに設定された値が有効になります。

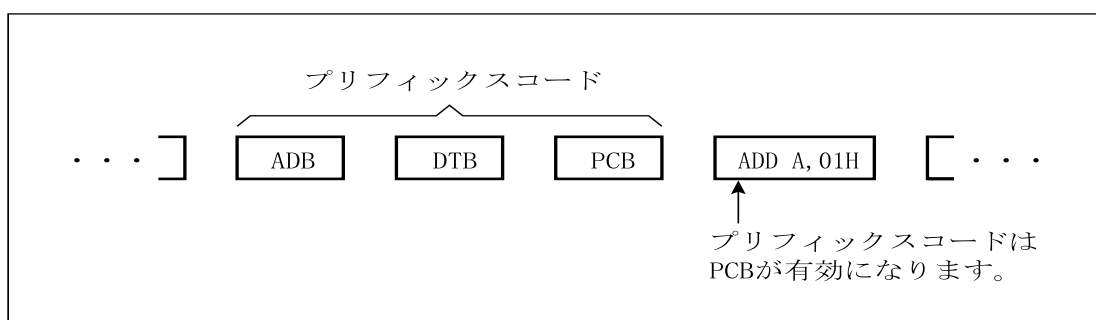


図 2.9-3 プリフィックスコードの連続

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

「DIV A,Ri」,「DIVW A,RWi」命令を使用する場合は、バンクレジスタを"00H"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 2.9-7 「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意(i=0~7)

命令	左記命令実行時に影響を受けるバンクレジスタ名	余りが格納されるアドレス
DIV A,R0	DTB	(DTB: 上位8ビット) + (0180H+RP × 10H+8H: 下位16ビット)
DIV A,R1		(DTB: 上位8ビット) + (0180H+RP × 10H+9H: 下位16ビット)
DIV A,R4		(DTB: 上位8ビット) + (0180H+RP × 10H+C H: 下位16ビット)
DIV A,R5		(DTB: 上位8ビット) + (0180H +RP × 10H +DH: 下位16ビット)
DIVW A,RW0		(DTB: 上位8ビット) + (0180H +RP × 10H +0H: 下位16ビット)
DIVW A,RW1		(DTB: 上位8ビット) + (0180H +RP × 10H +2H: 下位16ビット)
DIVW A,RW4		(DTB: 上位8ビット) + (0180H +RP × 10H +8H: 下位16ビット)
DIVW A,RW5		(DTB: 上位8ビット) + (0180H +RP × 10H +AH: 下位16ビット)
DIV A,R2	ADB	(ADB: 上位8ビット) + (0180H +RP × 10H +AH: 下位16ビット)
DIV A,R6		(ADB: 上位8ビット) + (0180H +RP × 10H +EH: 下位16ビット)
DIVW A,RW2		(ADB: 上位8ビット) + (0180H +RP × 10H +4H: 下位16ビット)
DIVW A,RW6		(ADB: 上位8ビット) + (0180H +RP × 10H +EH: 下位16ビット)
DIV A,R3	USB SSB ^{*1}	(USB ^{*2} : 上位8ビット) + (0180H +RP × 10H +BH: 下位16ビット)
DIV A,R7		(USB ^{*2} : 上位8ビット) + (0180H +RP × 10H +FH: 下位16ビット)
DIVW A,RW3		(USB ^{*2} : 上位8ビット) + (0180H +RP × 10H +6H: 下位16ビット)
DIVW A,RW7		(USB ^{*2} : 上位8ビット) + (0180H +RP × 10H +EH: 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"00H"の場合は、除算結果の余りは命令オペランドのレジスタに格納されます。バンクレジスタの値が"00H"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="053H", RP="03H"の場合で「DIV A,R0」を実行すると、R0のアドレスは"0180H" + RP ("03H") × "10 H" + "08H" (R0相当アドレス)="0001B8H"です。

ここで、「DIV A,R0」で指定されるバンクレジスタはデータバンクレジスタ (DTB) ですので、バンクアドレス"053H"を付加したアドレス"05301B8H"に余りが格納されます (RiおよびRWiのレジスタについては、「2.8 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 2.9-7の命令を生成しないように変更し、アセンブラでは表 2.9-7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。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. 300004)以降のバージョン

第3章 リセット

この章では、MB90560/565シリーズのリセットについて説明します。

- 3.1 リセットの概要
- 3.2 リセット要因と発振安定待ち時間
- 3.3 外部リセット端子
- 3.4 リセット動作
- 3.5 リセット要因フラグビット
- 3.6 リセットによる各端子の状態

3.1 リセットの概要

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

リセットには，以下に示す4種類の要因があります。

- ・ パワーオンリセットの発生（電源投入時）
- ・ ウォッチドッグタイマのオーバーフロー（ウォッチドッグタイマを使用している場合）
- ・ $\overline{\text{RST}}$ 端子から外部リセット入力を行った場合
- ・ 低消費電力モード制御レジスタの内部リセット信号発生ビット（RST）を"0"に設定した場合（ソフトウェアリセット）

リセット要因

表 3.1-1 リセット要因

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

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

外部リセット

外部リセットは，外部リセット端子（ $\overline{\text{RST}}$ 端子）を"L"レベルに設定した場合にリセットを発生します。"L"レベル入力時間は，16マシサイクル（16/ ）以上が必要です。マシクロックで動作している場合には，外部リセット端子を"L"レベルに設定してリセットが発生しても，発振安定待ち時間を取りません。

参考：

外部リセット端子を命令実行中（転送系命令実行中 MOV命令など）にアサートした場合，実行中の命令が終了したあとにリセット入力が有効になります。

ただし，ストリング系命令（MOV命令等）は設定したカウンタ値による転送が完了する前に，リセット入力が有効になる場合があります。

外部リセット端子をアサートした場合，ポート端子は命令実行サイクルに関係なくリセット状態になります。（アサートされた場合は非同期です。）

ソフトウェアリセット

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

ウォッチドッグリセット

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

パワーオンリセット

パワーオンリセットは、電源投入時に発生するリセットです。

MB90V560/MB90F562, MB90V565/MB90F568の発振安定待ち時間は $2^{18}/\text{HCLK}$ (約65.54ms: 発振クロック4MHz時), MB90562, MB90568の発振安定待ち時間は $2^{17}/\text{HCLK}$ (約32.77ms: 4MHz発振時) 固定になります。発振安定待ち時間が経過した後、リセット動作を行います。

備考：クロックの定義

HCLK：発振クロック周波数 (発振端子から供給されるクロック)

MCLK：メインクロック周波数 (発振クロックの2分周クロック)

：マシンクロック周波数 (CPU動作クロック)

1/ ：マシンサイクル (CPU動作クロック周期)

マシンクロックの詳細は、「4.1 クロックの概要」を参照してください。

3.2 リセット要因と発振安定待ち時間

F²MC-16LXには、4種類のリセット要因があり、リセット時の発振安定待ち時間がリセット要因によって異なります。

リセット要因と発振安定待ち時間

表 3.2-1 リセット要因と発振安定待ち時間

リセット要因	発振安定待ち時間 () 内は発振クロック周波数4MHz時
パワーオンリセット	MB90V560/MB90F562 : $2^{18}/\text{HCLK}$ (約65.54ms) MB90V565/MB90F568 : $2^{18}/\text{HCLK}$ (約65.54ms) MB90562, MB90568 : $2^{17}/\text{HCLK}$ (約32.77ms)
ウォッチドッグタイマ	なし (WS1, WS0ビットは"11 _B "に初期化されます。)
外部リセット	なし (WS1, WS0ビットは"11 _B "に初期化されます。)
ソフトウェアリセット	なし (WS1, WS0ビットは"11 _B "に初期化されます。)

HCLK : 発振クロック周波数 (MHz)

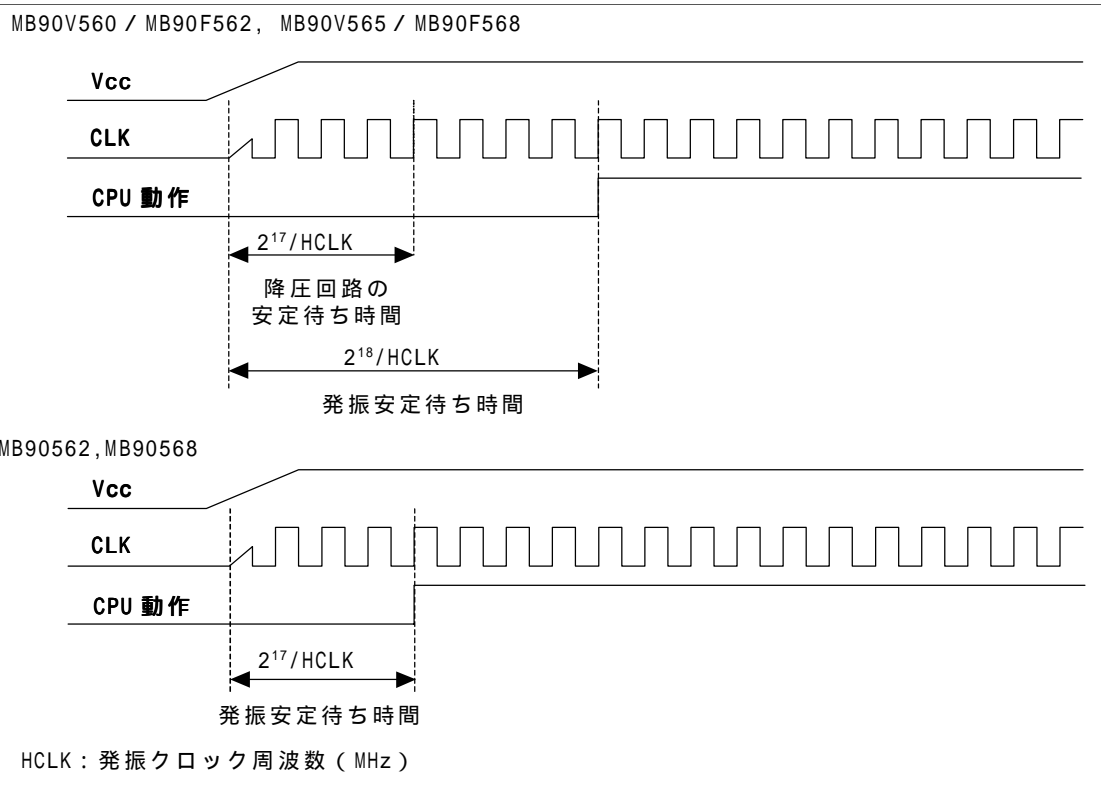


図 3.2-1 パワーオンリセットが発生した場合のMB90560/565シリーズの発振安定待ち時間

< 注意事項 >

発振クロック振動子は、発振を開始してから固有振動数で安定するまでに、振動子固有の発振安定待ち時間が必要です。使用する振動子に合わせた発振安定待ち時間を設定してください。

発振安定待ち時間の詳細は、「4.5 発振安定待ち時間」を参照してください。

3.3 外部リセット端子

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

外部リセット端子のブロックダイアグラム

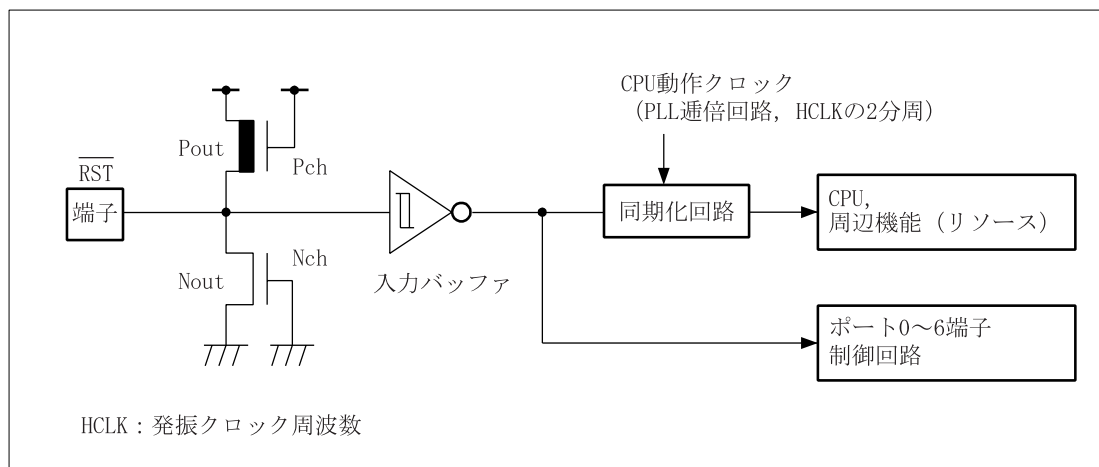


図 3.3-1 リセット発生のブロックダイアグラム

< 注意事項 >

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

3.4 リセット動作

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

リセット動作の概要

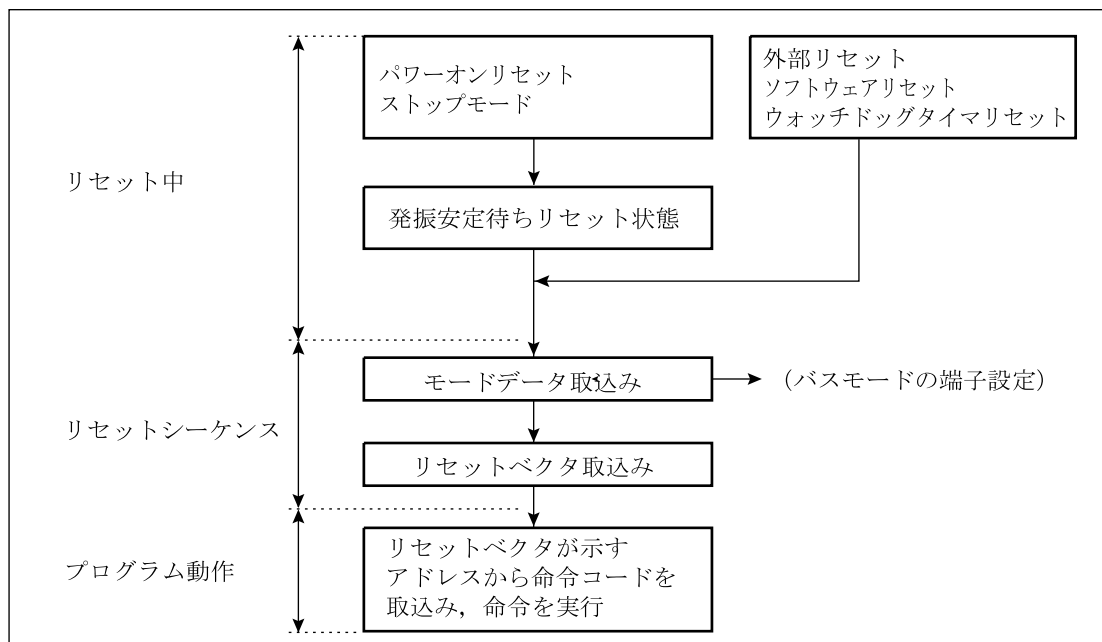


図 3.4-1 リセット動作フロー

モード端子

モード端子（MD2～MD0）は，モードデータとリセットベクタの取込み方法を事前に設定します。モードデータとリセットベクタを取込む場合は，リセットシーケンスを実行します。詳細は，「7.2 モード端子(MD2～MD0)」を参照してください。

モードデータ取込み

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

モードデータレジスタは，バスモードの設定およびバス幅の設定を行うことができます。また，リセットベクタでは，プログラムの開始アドレスを指定することができます。

モードデータ取込みの詳細は，「第7章 モード設定」を参照してください。

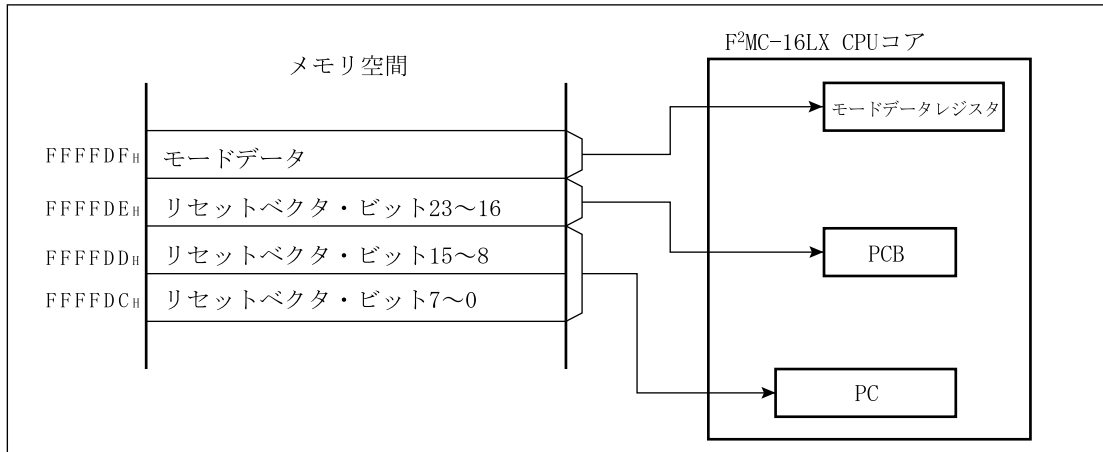


図 3.4-2 モードデータとリセットベクタの転送

参考：

モードデータとリセットベクタを、内蔵ROM（またはフラッシュメモリ）、または外部メモリから読み出すかは、モード端子を設定することにより設定できます。モード端子を外部メモリに設定した場合は、外部メモリからモードデータとリセットベクタを読み出します。また、モード端子を内蔵ROM（またはフラッシュメモリ）に設定した場合は、内蔵ROM（またはフラッシュメモリ）からモードデータとリセットベクタを読み出します。シングルチップモードの場合は、モード端子を内蔵ROM（またはフラッシュメモリ）に設定してください。詳細は、「7.2 モード端子(MD2～MD0)」を参照してください。

モードデータレジスタ（アドレス："FFFFDF_H"）

モードデータレジスタは、リセットシーケンス実行中に設定を変更できます。モードデータレジスタの設定は、リセットベクタ取込み後から有効になります。また、命令を使用して、「FFFFDF_H」にモードデータを設定しても、モードデータレジスタを書き換えることはできません。

詳細は、「7.3 モードデータレジスタ」を参照してください。

リセットベクタ（アドレス："FFFFDC_H～FFFFDE_H"）

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

3.5 リセット要因フラグビット

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

リセット要因フラグビット

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

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

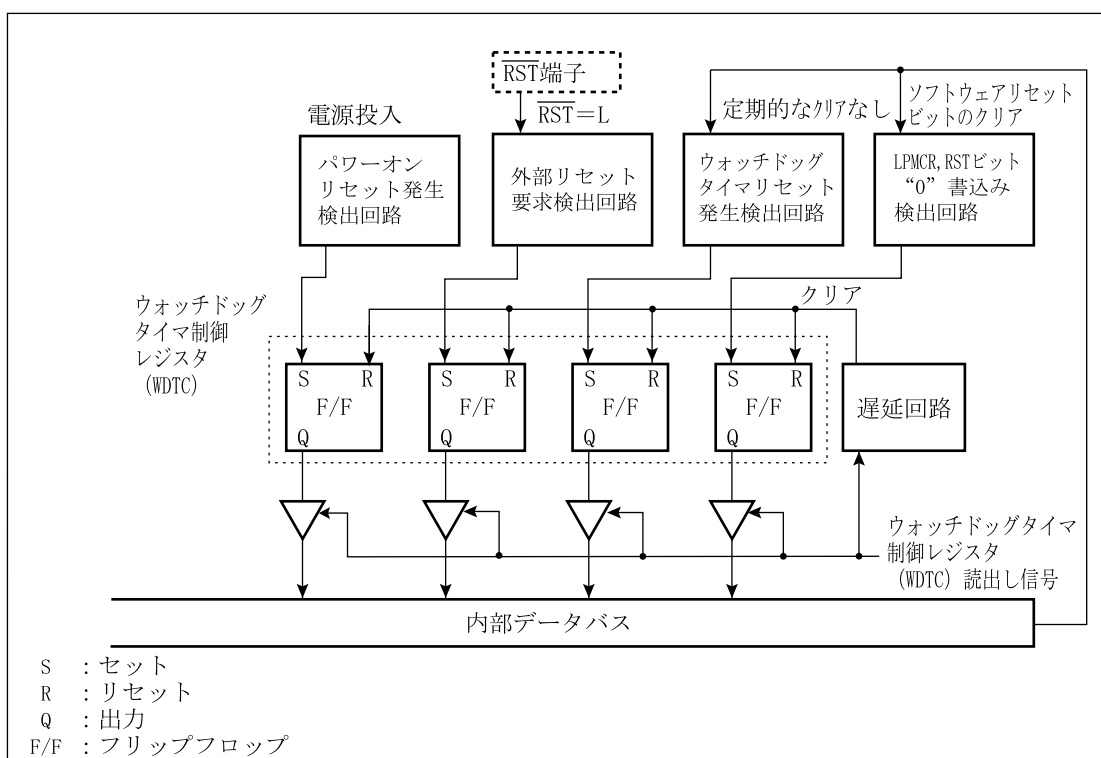


図 3.5-1 リセット要因フラグビットのブロック図

リセット要因フラグビットとリセット要因の対応

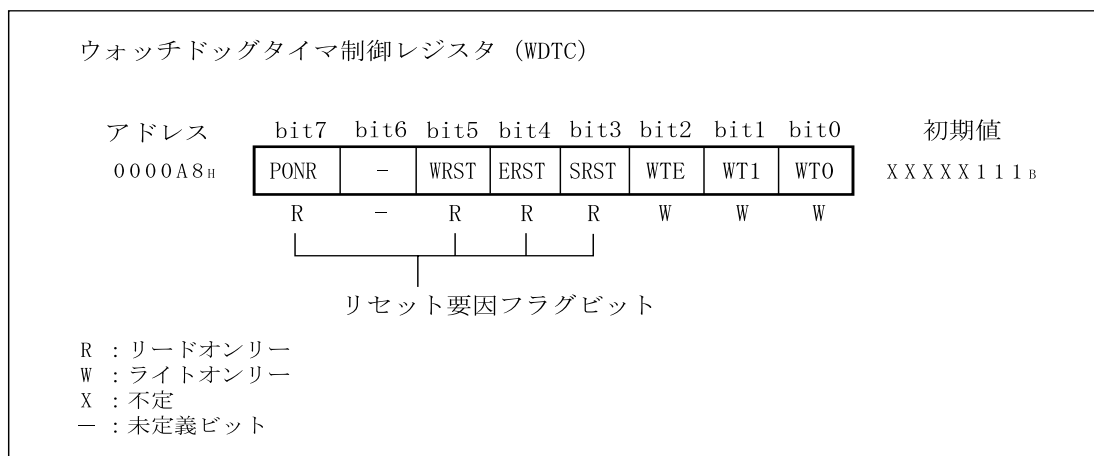


図 3.5-2 リセット要因フラグビットの構成 (ウォッチドッグタイマ制御レジスタ)

表 3.5-1 リセット要因フラグビットの内容とリセット要因の対応

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

* : 前の状態を保持

X : 不定

リセット要因フラグビットの注意事項

複数のリセット要因が発生した場合

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

【例】

- 外部リセットとウォッチドッグタイマリセットが同時に発生した場合は、ウォッチドッグタイマ制御レジスタ (WDTC) のリセット要因フラグビット (ERST, WRST) に "1" がセットされます。

パワーオンリセットが発生した場合

パワーオンリセットが発生した場合は、ウォッチドッグタイマ制御レジスタ（WDTC）のリセット要因フラグビット（PONR）に"1"がセットされますが、リセット要因フラグビット（WRST, ERST, SRST）は不定値となります。

リセット要因フラグビット（PONR）に"1"がセットされている場合は、リセット要因フラグビット（WRST, ERST, SRST）の内容を無視してください。

リセット要因フラグビットのクリア

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

3.6 リセットによる各端子の状態

リセットによる各端子の状態について説明します。

リセットによる各端子の状態

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

内部ベクタモード設定時（MD2="0"，MD1="1"，MD0="1"）

ポート0～6はハイインピーダンス出力となります。

モードデータは内蔵ROM（フラッシュメモリ）から読み出されます。

モードデータ読み出し後の端子状態

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

シングルチップモード設定時（モードデータレジスタ：M1,M0ビット="00_B"）

ポート0～6はハイインピーダンス出力となります。

モードデータは、内蔵ROM（またはフラッシュメモリ）から読み出されます。

< 注意事項 >

外部端子レベルを、外部回路が動作しないように設定してください。

リセット中の各端子状態は、「5.7 スタンバイモード,ホールドモード,リセットの端子状態」の「表 5.7-1」を参照してください。

第4章 クロック

この章では、MB90560/565シリーズのクロックについて説明します。

- 4.1 クロックの概要
- 4.2 クロック発生部のブロックダイアグラム
- 4.3 クロック選択レジスタ（CKSCR）
- 4.4 クロックモード
- 4.5 発振安定待ち時間
- 4.6 振動子と外部クロックの接続

4.1 クロックの概要

クロック発生部は、CPUおよび周辺機能（リソース）の動作クロックを制御します。クロックには、以下の4種類があります。

- ・発振クロック
- ・メインクロック
- ・PLLクロック
- ・マシンのクロック

クロックの概要

クロック発生部は、発振回路およびPLL逡倍回路を内蔵しており、発振安定待ち時間の制御、PLLクロック逡倍制御、およびクロックセクタでのクロック切替えの動作制御を行います。

発振クロック

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

メインクロック

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

PLLクロック

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

マシンのクロック

CPUおよび周辺機能（リソース）の動作クロックです。マシンのクロックの1周期をマシンサイクルといいます。メインクロックおよびPLLクロックから1種類を設定します。

< 注意事項 >

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

クロック供給マップ

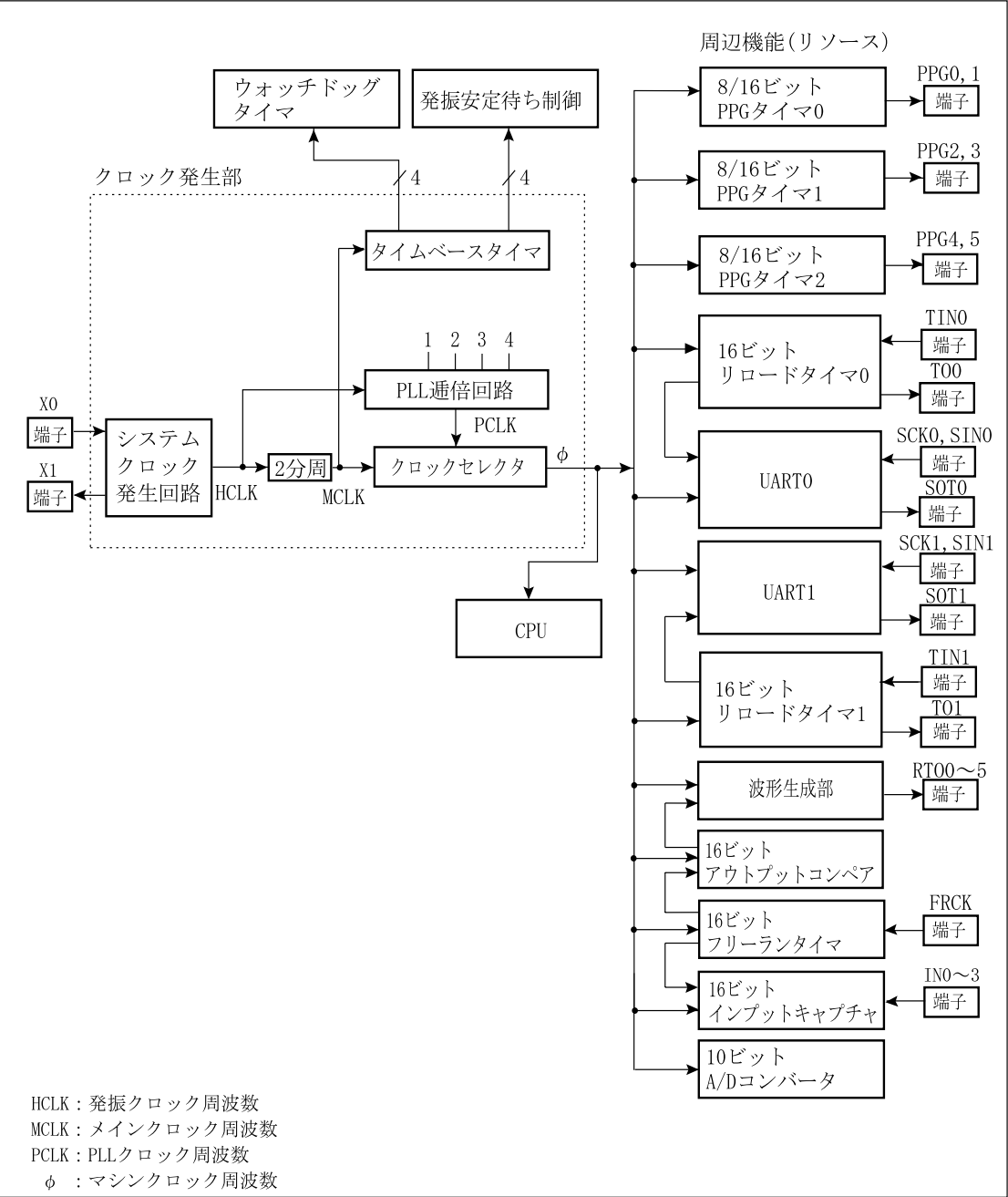


図 4.1-1 クロック供給マップ

4.2 クロック発生部のブロックダイアグラム

クロック発生部は、以下の5種類のブロックから構成されています。

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

クロック発生部のブロックダイアグラム

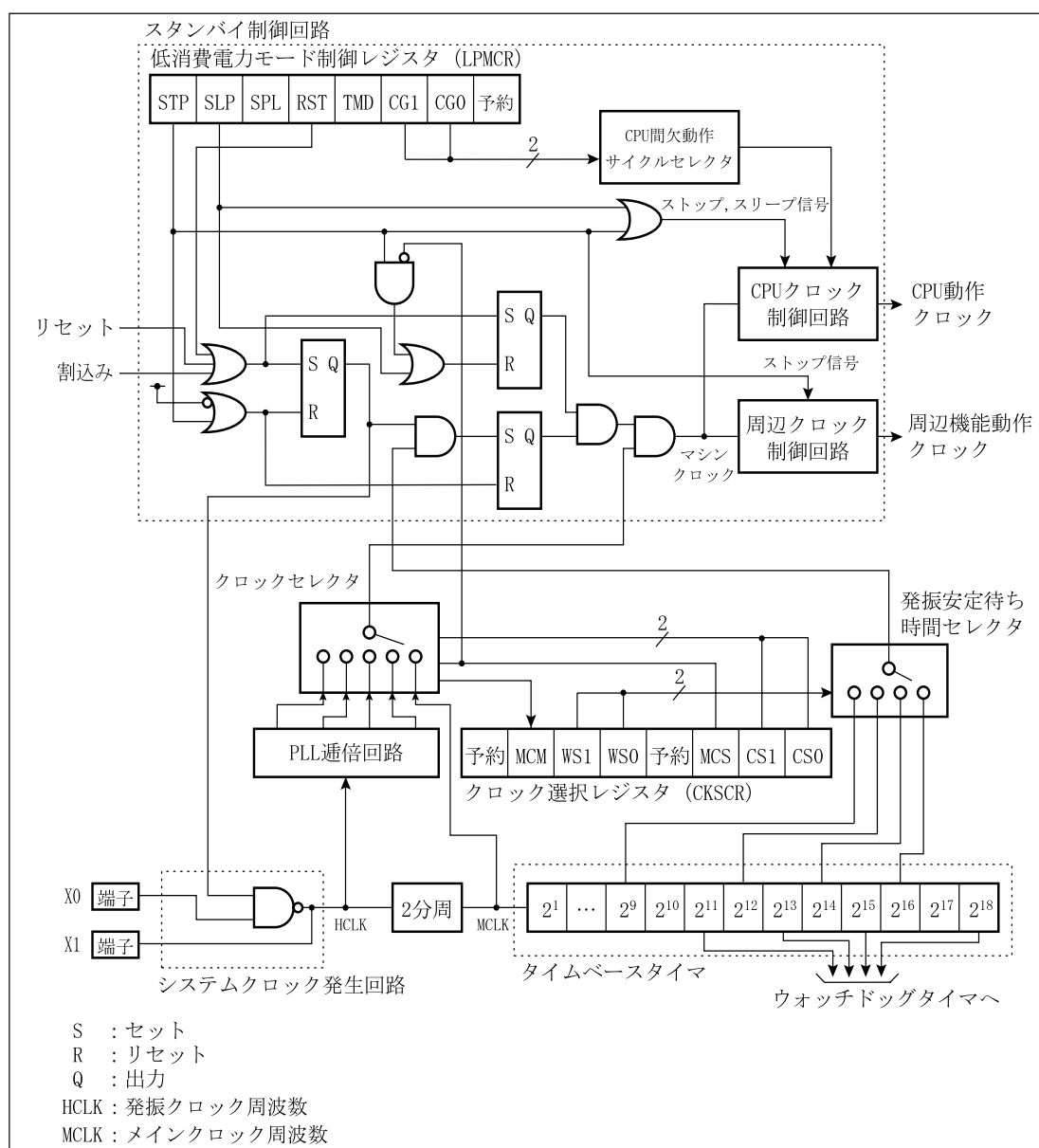


図 4.2-1 クロック発生部のブロックダイアグラム

システムクロック発生回路

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

PLL逡倍回路

発振クロックを逡倍し，クロックセクタへ供給します。

クロックセクタ

メインクロックおよびPLLクロックから，CPUクロック制御回路および周辺クロック制御回路へ供給されるクロックを設定します。

クロック選択レジスタ（CKSCR）

マシンクロックの切替え，発振安定待ち時間の設定，およびPLLクロックの逡倍率の設定などを行います。

発振安定待ち時間セクタ

ストップモードを解除した場合の発振クロックの発振安定待ち時間を設定します。4種類のタイムベースタイマ出力から発振安定待ち時間を設定します。

4.3 クロック選択レジスタ (CKSCR)

クロック選択レジスタ (CKSCR) は、マシンのクロックの切替え、発振安定待ち時間の設定、およびPLLクロックの逡倍率の設定などを行うレジスタです

クロック選択レジスタ (CKSCR) の構成

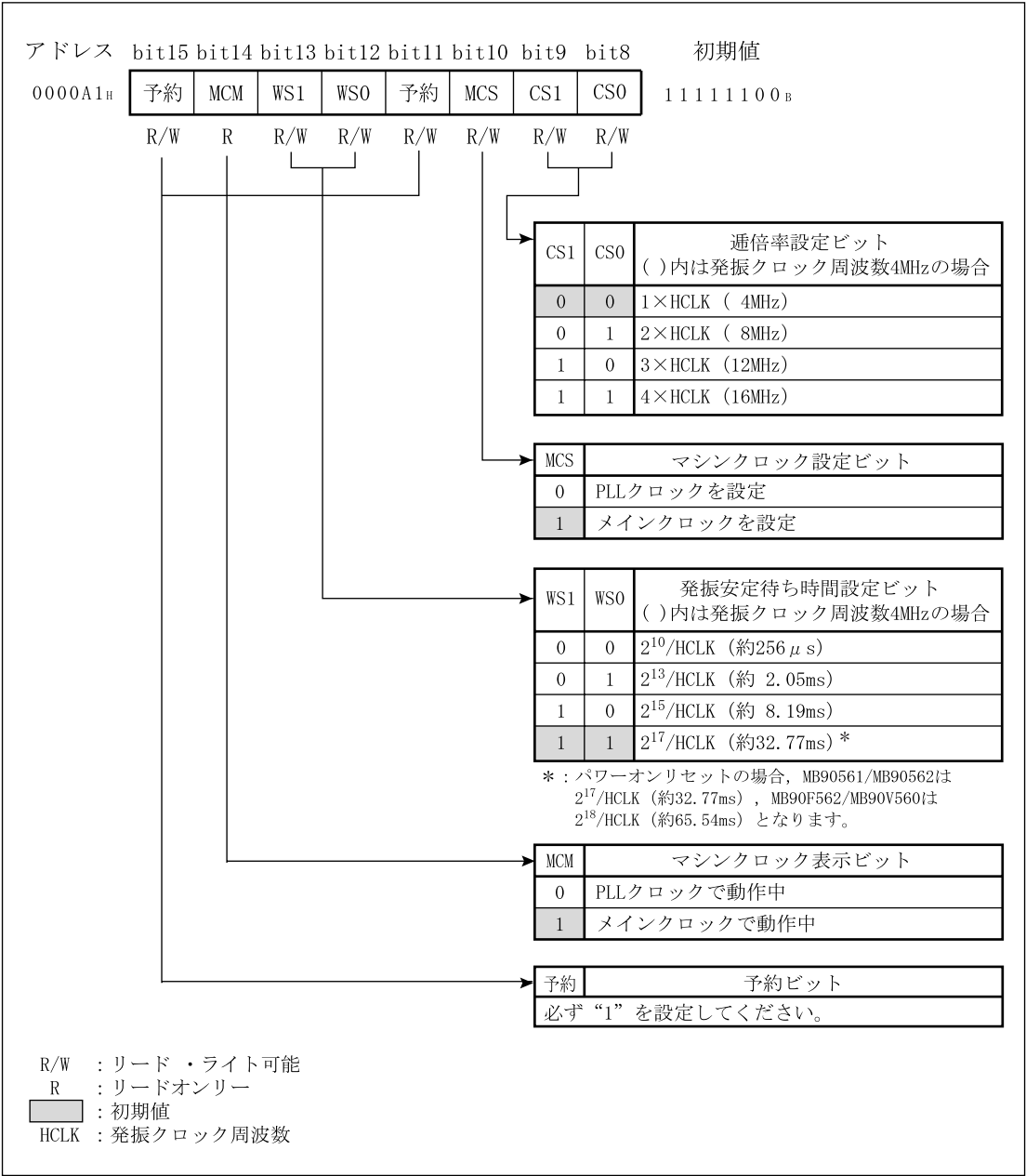


図 4.3-1 クロック選択レジスタ (CKSCR) の構成

表 4.3-1 クロック選択レジスタ (CKSCR) の各ビットの機能説明

ビット名		機 能
bit15 bit11	予約: 予約ビット	<ul style="list-style-type: none"> 必ず"1"を設定してください。
bit14	MCM: マシニングロック 表示ビット	<ul style="list-style-type: none"> マシニングロックとして、メインクロックまたはPLLクロックのどちらが設定されているかを示します。 "0"にクリアされている場合は、PLLクロックが設定されていることを示します。 "1"がセットされている場合は、メインクロックが設定されていることを示します。 マシニングロック設定ビット (MCS) に"0"を設定後に,"1"がセットされている場合は、PLLクロックの発振安定待ち時間中であることを示します。
bit13 bit12	WS1, WS0: 発振安定待ち 時間設定ビット	<ul style="list-style-type: none"> ストップモードを外部割込みにより解除した場合の、発振クロックの発振安定待ち時間を設定します。 リセット要因で"11b"に初期化されます。 発振安定待ち時間は、使用する振動子に合わせて設定してください。
bit10	MCS: マシニングロック 設定ビット	<ul style="list-style-type: none"> マシニングロックとして、メインクロックまたはPLLクロックを設定します。 "0"を設定した場合は、PLLクロックを設定します。 "1"を設定した場合は、メインクロックを設定します。 "1"が設定されている場合に,"0"を設定すると、PLLクロックの発振安定待ち時間が発生するため、タイムベースタイマカウンタおよびタイムベースタイマ制御レジスタ (TBTC) の割込み要求フラグビット (TBOF) が"0"にクリアされます。 PLLクロックの発振安定待ち時間は、$2^{14}/\text{HCLK}$で固定です。発振クロック周波数4MHzの場合は、発振安定待ち時間は約4.1msになります。 メインクロックを設定した場合のマシニングロックは、発振クロックを2分周したクロックとなります。発振クロック周波数4MHzの場合、マシニングクロック周波数は、2MHzになります。 リセットで"1"に初期化されます。 <p><注記> "1"が設定されている場合に,"0"を設定するには、タイムベースタイマ制御レジスタ (TBTC) の割込み要求許可ビット (TBIE) またはインタラプトレベルマスクレジスタ (ILM) により、タイムベースタイマ割込み要求が禁止されている状態で行うようにしてください。</p>
bit9 bit8	CS1, CS0: 逡倍率設定 ビット	<ul style="list-style-type: none"> PLLクロックの逡倍率を設定します。 4種類の逡倍率から設定できます。 リセットで"00b"に初期化されます。 <p><注記> マシニングロック設定ビット (MCS) または、マシニングロック表示ビット (MCM) が"0"に設定されている場合は、設定できません。マシニングロック設定ビット (MCS) に"1"を設定後に、設定してください。</p>

HCLK: 発振クロック周波数

4.4 クロックモード

クロックモードには、メインクロックモードとPLLクロックモードがあります。

メインクロックモードとPLLクロックモード

メインクロックモード

メインクロックモードは、CPUおよび周辺機能（リソース）のマシナクロックとして、メインクロックを使用し、PLLクロックを停止させます。

PLLクロックモード

PLLクロックモードは、CPUおよび周辺機能（リソース）のマシナクロックとしてPLLクロックを使用します。PLLクロックの通倍率は、クロック選択レジスタ（CKSCR）の通倍率設定ビット（CS1, CS0）で設定します。

クロックモードの遷移

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

メインクロックモードからPLLクロックモードへの遷移

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

PLLクロックモードからメインクロックモードへの遷移

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

<注意事項>

マシナクロックの切替え後、周辺機能（リソース）を設定する場合は、クロック選択レジスタ（CKSCR）のマシナクロック表示ビット（MCM）により、マシナクロックが切り替わったことを確認したあとに設定してください。

PLLクロック通倍率の設定

クロック選択レジスタ（CKSCR）の通倍率設定ビット（CS1, CS0）に"00_B"～"11_B"を設定し、1～4通倍の4種類からPLLクロック通倍率を設定します。

マシナクロック

メインクロックまたはPLLクロックが、マシナクロックとなります。マシナクロックはCPUおよび周辺機能（リソース）の動作クロックです。メインクロックまたはPLLクロックは、クロック選択レジスタ（CKSCR）のマシナクロック設定ビット（MCS）で設定します。

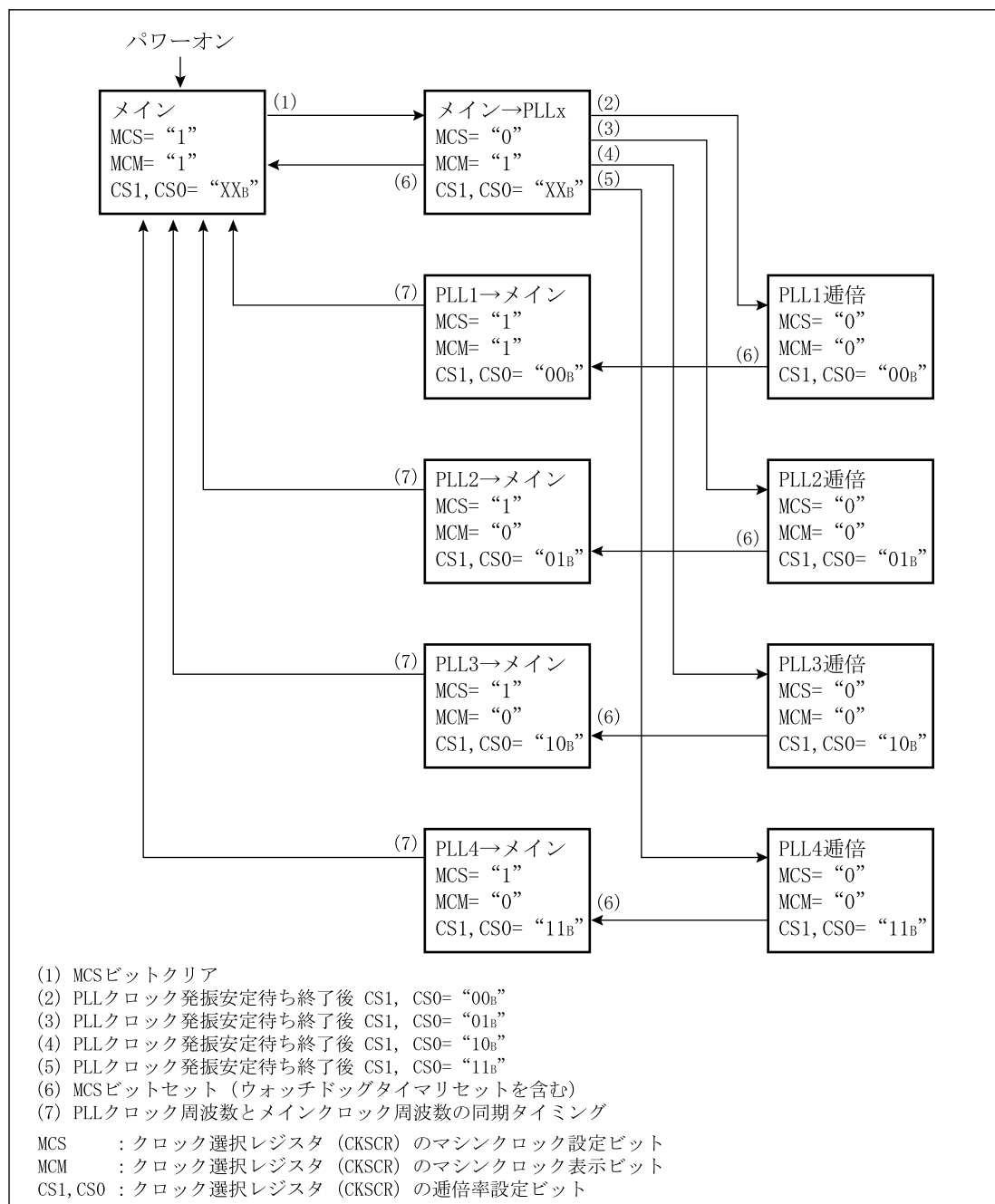


図 4.4-1 マシナクロック選択の状態遷移図

< 注意事項 >

マシナクロックの初期値は、メインクロック (CKSCR : MCS="1") です。

4.5 発振安定待ち時間

電源投入，およびストップモードを解除した場合は，発振停止からの動作となるため，発振開始後に発振安定待ち時間が必要です。また，メインクロックからPLLクロックへ切り替える場合も，PLLクロックの発振開始後に発振安定待ち時間が必要です。

発振安定待ち時間

振動子の種類によって発振が安定するまでの時間が異なるため，使用する振動子に合わせて発振安定待ち時間を設定する必要があります。クロック選択レジスタ（CKSCR）の発振安定待ち時間設定ビット（WS1,WS0）によって適切な発振安定待ち時間を設定してください。

メインクロックからPLLクロックへ切り替えた場合，PLL発振安定待ち時間の間，CPUはメインクロックで動作したあとに，PLLクロックに切り替わります。

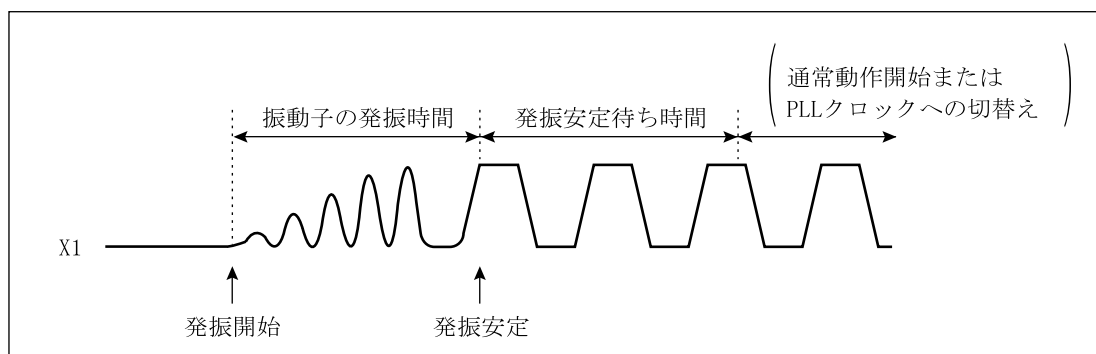


図 4.5-1 発振開始直後の動作

4.6 振動子と外部クロックの接続

MB90560/565シリーズは、システムクロック発生回路を内蔵しており、X0、X1端子に振動子を接続するか、または外部クロックを入力することができます。

振動子と外部クロックの接続

水晶振動子またはセラミック振動子の接続例

水晶振動子またはセラミック振動子は、図 4.6-1の例で示すように接続してください。

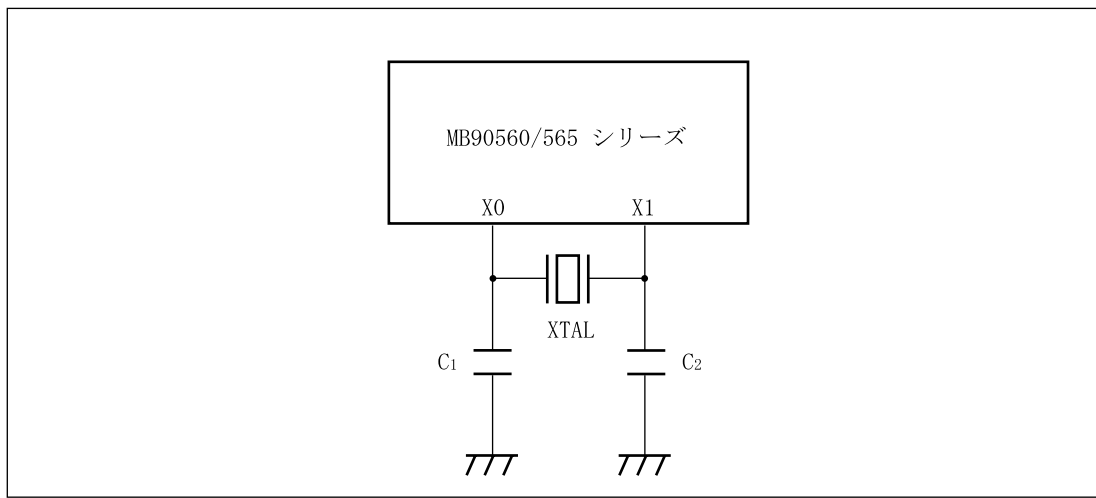


図 4.6-1 水晶振動子またはセラミック振動子の接続例

外部クロックの接続例

図 4.6-2の例で示すように、外部クロックは、X0端子に接続し、X1端子はオープン（開放）にしてください。

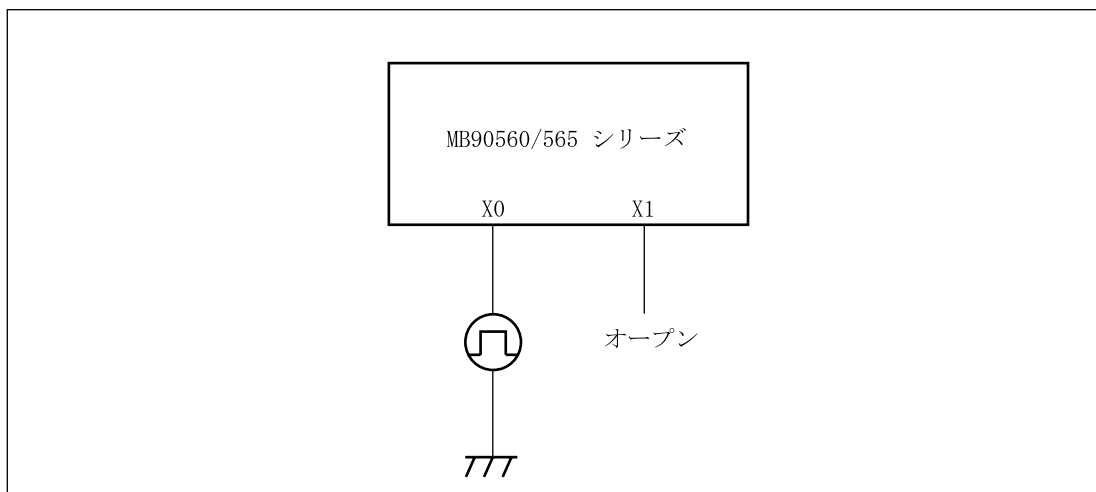


図 4.6-2 外部クロックの接続例

第5章 低消費電力モード

この章では,MB90560/565シリーズの低消費電力モードについて説明します。

- 5.1 低消費電力モードの概要
- 5.2 低消費電力制御回路のブロックダイアグラム
- 5.3 低消費電力モード制御レジスタ (LPMCR)
- 5.4 CPU間欠動作モード
- 5.5 スタンバイモード
- 5.6 状態遷移図
- 5.7 スタンバイモード, ホールド, リセットの端子状態
- 5.8 低消費電力モード使用上の注意

5.1 低消費電力モードの概要

MB90560/565シリーズは、動作クロックの設定とクロックの動作制御により、次の低消費電力モードがあります。

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

CPU動作モードと消費電流

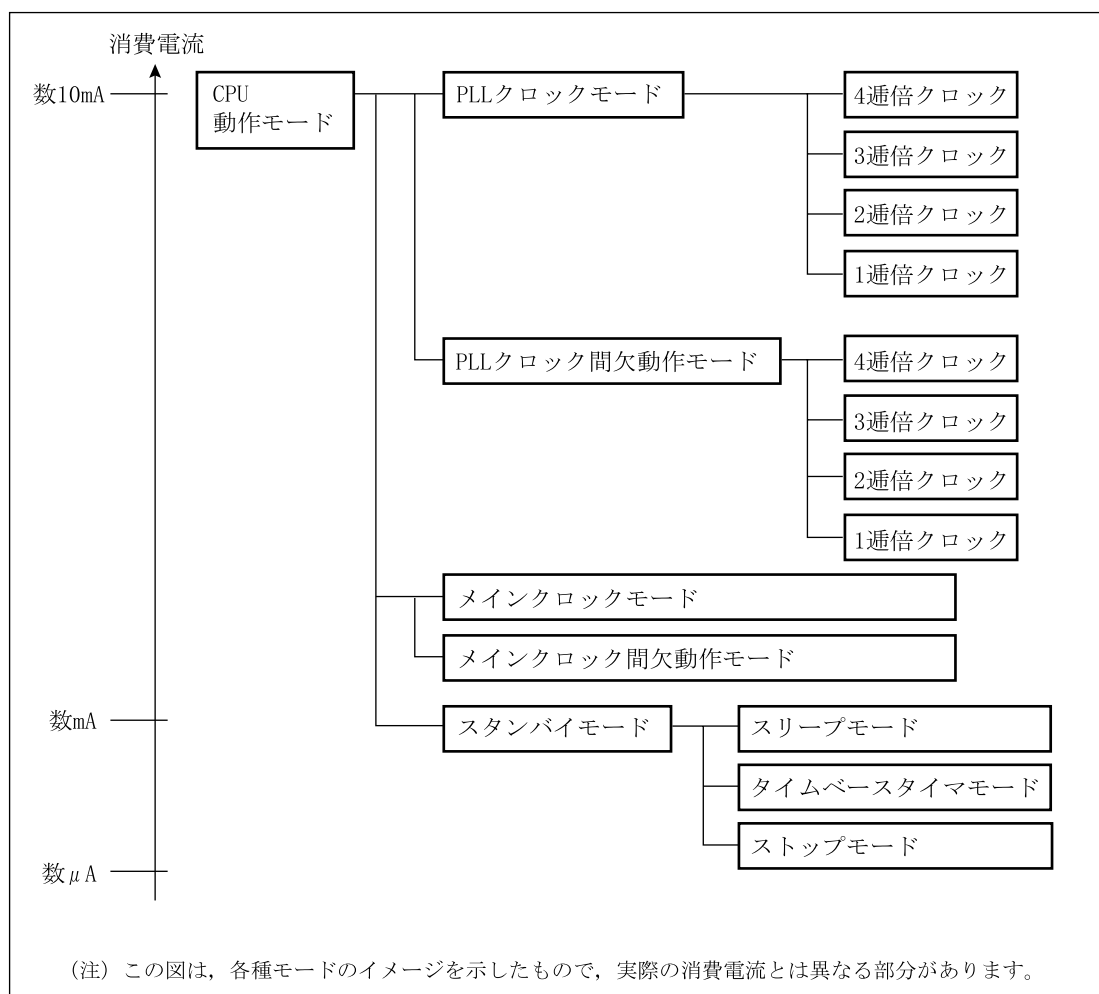


図 5.1-1 CPU動作モードと消費電流

クロックモード

PLLクロックモード

PLLクロックで、CPUおよび周辺機能（リソース）を動作させます。

メインクロックモード

メインクロックで、CPUおよび周辺機能（リソース）を動作させます。メインクロックモードでは、PLL逡倍回路が停止します。

参考：

クロックモードの詳細は、「4.4 クロックモード」を参照してください。

CPU間欠動作モード

周辺機能（リソース）にマシクロックを供給した状態で、CPUを間欠動作させます。

スタンバイモード

PLLスリープモード

CPUの動作クロックを停止させます。CPU以外はPLLクロックで動作します。

メインスリープモード

CPUの動作クロックを停止させます。CPU以外はメインクロックで動作します。

タイムベースタイマモード

発振クロックとタイムベースタイマ以外の動作を停止させます。

ストップモード

発振クロックを停止させます。すべての機能が停止します。

< 注意事項 >

ストップモードでは、発振クロックが停止するため、最も低消費電力でデータを保持できます。MB90560/565シリーズには、ハードウェアスタンバイ機能用の端子がないためストップモードは使用できません。

5.2 低消費電力制御回路のブロックダイアグラム

低消費電力制御回路は、以下の7種類のブロックから構成されています。

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

低消費電力制御回路のブロックダイアグラム

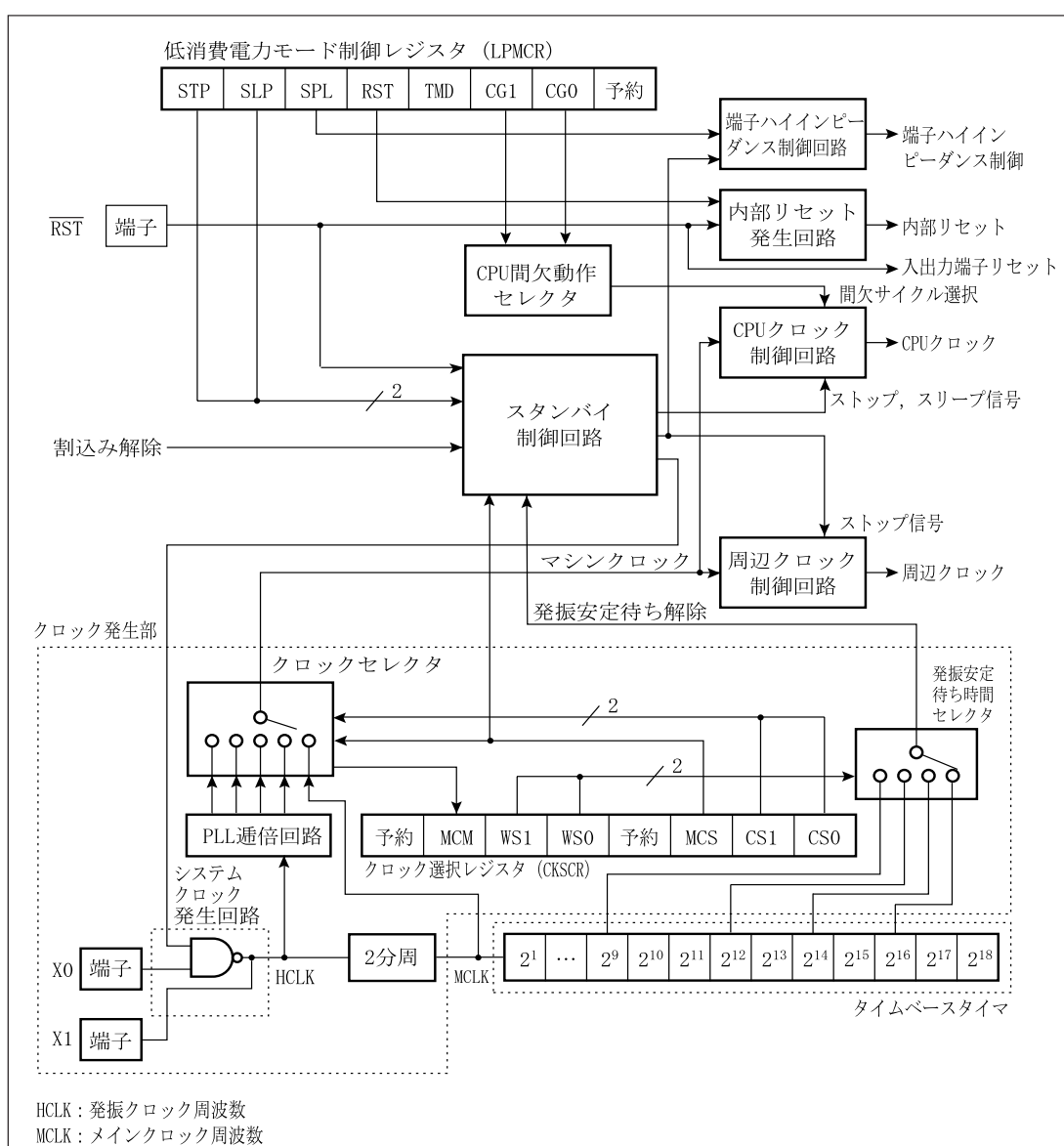


図 5.2-1 低消費電力制御回路のブロックダイアグラム

CPU間欠動作セクタ

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

スタンバイ制御回路

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

CPUクロック制御回路

CPUに供給するクロックを制御します。

周辺クロック制御回路

周辺機能（リソース）に供給するクロックを制御します。

端子ハイインピーダンス制御回路

タイムベースタイマモードおよびストップモードで，設定により入出力端子をハイインピーダンスにします。プルアップ抵抗の接続を設定している入出力端子は，ストップモードでプルアップ抵抗を切り離します。

内部リセット発生回路

内部リセット信号を発生させます。

低消費電力モード制御レジスタ（LPMCR）

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

5.3 低消費電力モード制御レジスタ (LPMCR)

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

低消費電力モード制御レジスタ (LPMCR)

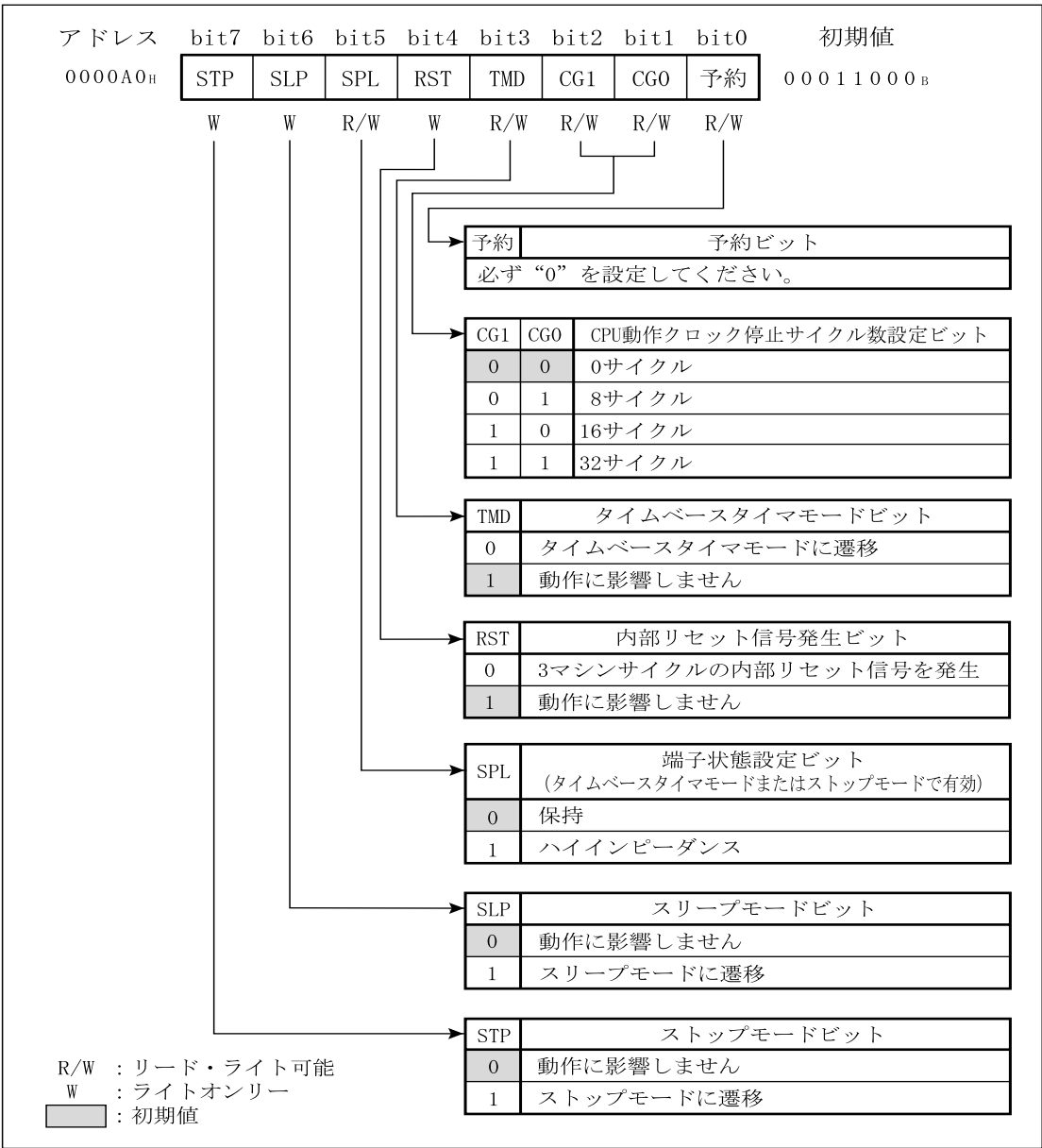


図 5.3-1 低消費電力モード制御レジスタ (LPMCR) の構成

表 5.3-1 低消費電力モード制御レジスタ (LPMCR) の各ビットの機能説明

ビット名		機 能
bit7	STP : ストップモードビット	<ul style="list-style-type: none"> ・ストップモードを設定するビットです。 ・"1"を設定した場合は、ストップモードに遷移します。 ・"0"を設定した場合は、動作に影響しません。 ・外部リセット、またはハードウェア割込み出力で、"0"にクリアされます。 ・読出し値は、"0"です。
bit6	SLP : スリープモードビット	<ul style="list-style-type: none"> ・スリープモードを設定するビットです。 ・"1"を設定した場合は、スリープモードに遷移します。 ・"0"を設定した場合は、動作に影響しません。 ・外部リセット、またはハードウェア割込み出力で、"0"にクリアされます。 ・読出し値は、"0"です。
bit5	SPL : 端子状態設定ビット (タイムベースタイマ モードまたはストップ モードで有効)	<ul style="list-style-type: none"> ・タイムベースタイマモード、またはストップモードの場合での端子状態を設定するビットです。 ・"0"を設定した場合は、入出力端子のレベルを保持します。 ・"1"を設定した場合は、入出力端子をハインピーダンスにします。 ・外部リセットで"0"にクリアされます。
bit4	RST : 内部リセット 信号発生ビット	<ul style="list-style-type: none"> ・内部リセットを設定するビットです。 ・"0"を設定した場合は、3マシンサイクルの内部リセット信号を発生します。 ・"1"を設定した場合は、動作に影響しません。 ・読出し値は、"1"です。
bit3	TMD : タイムベースタイマ モードビット	<ul style="list-style-type: none"> ・タイムベースタイマモードへの遷移を設定するビットです。 ・"0"を設定した場合は、タイムベースタイマモードに遷移します。 ・"1"を設定した場合は、動作に影響しません。 ・外部リセット、またはタイムベースタイマ割込み出力で、"1"に初期化されます。 ・読出し値は、"1"です。
bit2 bit1	CG1, CG0 : CPU動作クロック停止 サイクル数設定ビット	<ul style="list-style-type: none"> ・CPU間欠動作モードで動作中のCPU動作クロック停止サイクル数を設定するビットです。 ・一命令ごとに設定サイクル数のCPUクロック供給を停止します。 ・4種類のサイクル数から設定します。 ・リセットで"00b"に初期化されます。
bit0	予約 : 予約ビット	<ul style="list-style-type: none"> ・必ず"0"を設定してください。

低消費電力モード制御レジスタへのアクセス

低消費電力モード制御レジスタを設定する命令は、表 5.3-2に示す命令を使用してください。表 5.3-2に示す命令以外で低消費電力モードへ遷移した場合は、動作が保証できません。

ワード長で低消費電力モード制御レジスタへの設定を行う場合は、低消費電力制御回路（0000A0_H，0000A1_H）にワードアクセスしてください。

表 5.3-2 低消費電力モードへ遷移する場合に使用する命令一覧

MOV io,#imm8	MOV dir,#imm8	MOV eam,#imm8	MOV eam,Ri
MOV io,A	MOV dir,A	MOV addr,A	MOV eam,A
MOV @RLi+disp8,A	MOVW dir24,A		
MOVW io,#imm16	MOVW dir,#imm16	MOVW eam,#imm16	MOVW eam,RWi
MOVW io,A	MOVW dir,A	MOVW addr16,	AMOVW eam,A
MOVW @RLi+disp8,A	MOVW dir24,A		

スタンバイモードの優先順位

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

5.4 CPU間欠動作モード

CPU間欠動作モードは、周辺機能(リソース)をマシンのクロックで動作させたまま、CPUを間欠動作させます。

CPU間欠動作モード

CPU間欠動作モードは、CPUに供給するマシンのクロックを、1命令実行後、一定期間停止させ、内部バスサイクルの起動を遅らせるモードです。

図 5.4-1中の「停止サイクル数」は、低消費電力モード制御レジスタ(LPMCR)のCPU動作クロック停止サイクル数設定ビット(CG1,CG0)で設定します。

停止サイクル数の詳細は、「5.3 低消費電力モード制御レジスタ(LPMCR)」を参照してください。

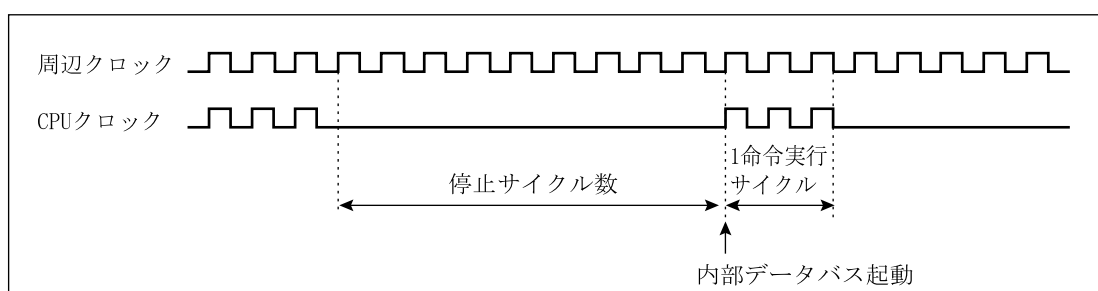


図 5.4-1 CPU間欠動作時のクロック

5.5 スタンバイモード

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

スタンバイモードでの動作状態

表 5.5-1 スタンバイモードでの動作状態

スタンバイモード		遷移条件	発振	クロック	CPU	タイムベースタイマ	周辺	端子	解除方法	
スリープモード	PLL スリープモード	MCS=0 SLP=1	動作	動作	停止	動作	動作	動作	外部 リセット または、 割込み	
	メインスリープ モード	MCS=1 SLP=1								
タイムベース タイマモード	タイムベースタイマ モード (SPL = 0)	TMD=0	停止	動作		動作	動作	保持	外部 リセット または、 割込み ^{*1}	
	タイムベースタイマ モード (SPL = 1)	TMD=0								
ストップ モード	ストップモード (SPL = 0)	STP=1		停止		動作	動作	動作	保持	外部 リセット または、 割込み ^{*2}
	ストップモード (SPL = 1)	STP=1								

*1 : タイムベースタイマ、および外部割込み

*2 : 外部割込み

SPL : 低消費電力モード制御レジスタ(LPMCR)の端子状態設定ビット

SLP : 低消費電力モード制御レジスタ(LPMCR)のスリープモードビット

STP : 低消費電力モード制御レジスタ(LPMCR)のストップモードビット

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

MCS : クロック選択レジスタ(CKSCR)のマシンクロック設定ビット

Hi-z : ハイインピーダンス

5.5.1 スリープモード

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

スリープモードへの遷移

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

データ保持機能

スリープモードは、専用レジスタとRAMの内容を保持します。

専用レジスタの詳細は、「2.7 専用レジスタ」を参照してください。

ホールド機能

スリープモードは、ホールド機能が使用できます。ホールド要求がある場合は、ホールド状態になります。

割込み要求出力中の動作

割込み要求を出力している場合は、低消費電力モード制御レジスタ (LPMCR) のスリープモードビット (SLP) に "1" を設定しても、スリープモードへ遷移せず、次の命令を実行します。

スリープモードの解除

スリープモードは、外部リセット、またはハードウェア割込み出力で解除できます。

外部リセットによる解除

詳細は、「3.4 リセット動作」を参照してください。

ハードウェア割込みによる解除

ハードウェア割込みによりスリープモードを解除する場合は、割込みレベルが7より強い割込み要求（割込み制御レジスタ ICR：IL2, IL1, IL0="000b" ~ "110b"）が必要です。

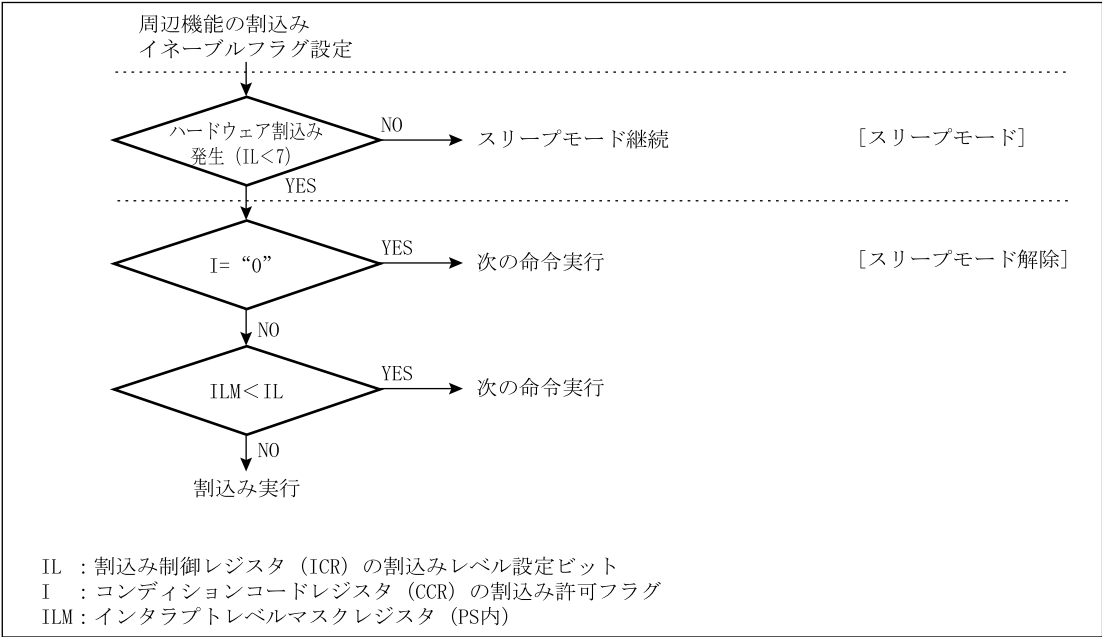


図 5.5-1 ハードウェア割込み出力によるスリープモードの解除フロー

5.5.2 タイムベースタイマモード

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

タイムベースタイマモードへの遷移

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

データ保持機能

タイムベースタイマモードは、専用レジスタとRAMの内容を保持します。
専用レジスタの詳細は、「2.7 専用レジスタ」を参照してください。

ホールド機能

タイムベースタイマモードは、ホールド機能が使用できません。タイムベースタイマモードへ遷移中に、ホールド要求が入力された場合、外部バス端子をハイインピーダンスにし、HAKX端子が"H"になります。

割込み要求出力中の動作

割込み要求を出力している場合は、低消費電力モード制御レジスタ (LPMCR) のタイムベースタイマモードビット (TMD) に "0" を設定しても、タイムベースタイマモードへ遷移しません。

端子状態

タイムベースタイマモードの入出力端子は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL) で、直前のレベルを保持するか、ハイインピーダンスにするかを設定します。

タイムベースタイマモードの解除

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

外部リセットによる解除

詳細は、「3.4 リセット動作」を参照してください。

タイムベースタイマ割込みによる解除

タイムベースタイマ割込みによりタイムベースタイマモードを解除する場合は、割込みレベルが7より強いタイムベースタイマ、および外部割込みによる割込み要求（割込み制御レジスタ ICR：IL2, IL1, IL0="000_B" ~ "110_B"）が必要です。

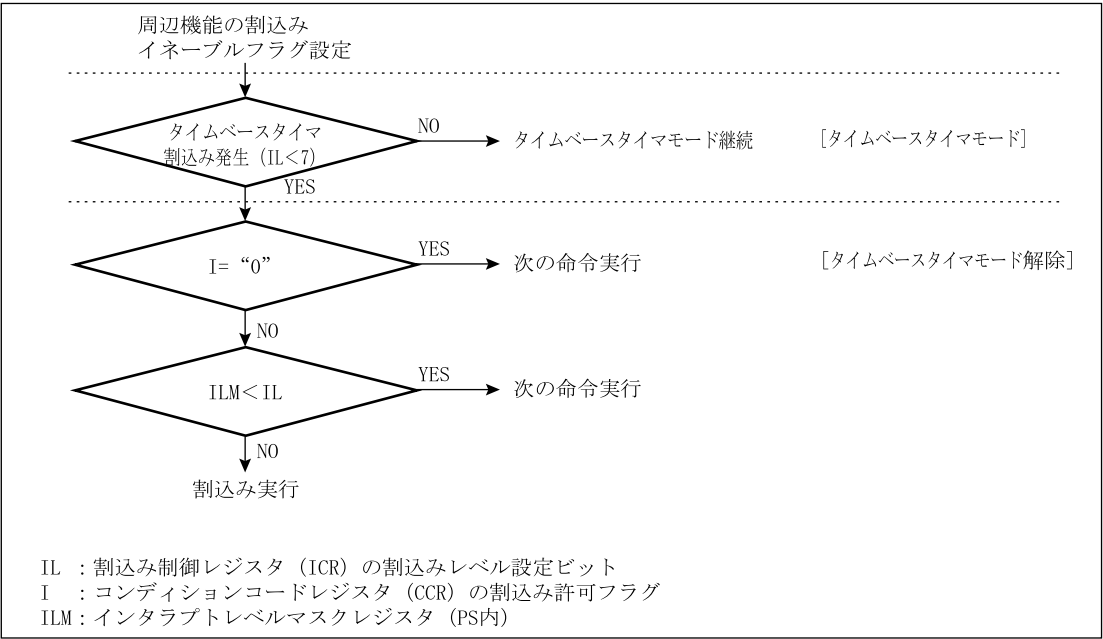


図 5.5-2 外部割込み出力によるタイムベースタイマモードの解除フロー

5.5.3 ストップモード

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

ストップモードへの遷移

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

データ保持機能

ストップモードは、専用レジスタとRAMの内容を保持します。

専用レジスタの詳細は、「2.7 専用レジスタ」を参照してください。

ホールド機能

ストップモードは、ホールド機能が使用できません。ストップモードへ遷移中に、ホールド要求が入力された場合、外部バス端子をハイインピーダンスにし、HAKX端子が"H"になります。

割込み要求出力中の動作

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

端子状態の設定

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

ストップモードの解除

ストップモードは、外部リセット、またはハードウェア割込み出力で解除できます。

外部リセットによる解除

詳細は、「3.4 リセット動作」を参照してください。

外部割込みによる解除

外部割込みによりストップモードを解除する場合は、割込みレベルが7より強い外部割込みによる割込み要求（割込み制御レジスタ ICR：IL2, IL1, IL0="000_B" ~ "110_B"）が必要です。

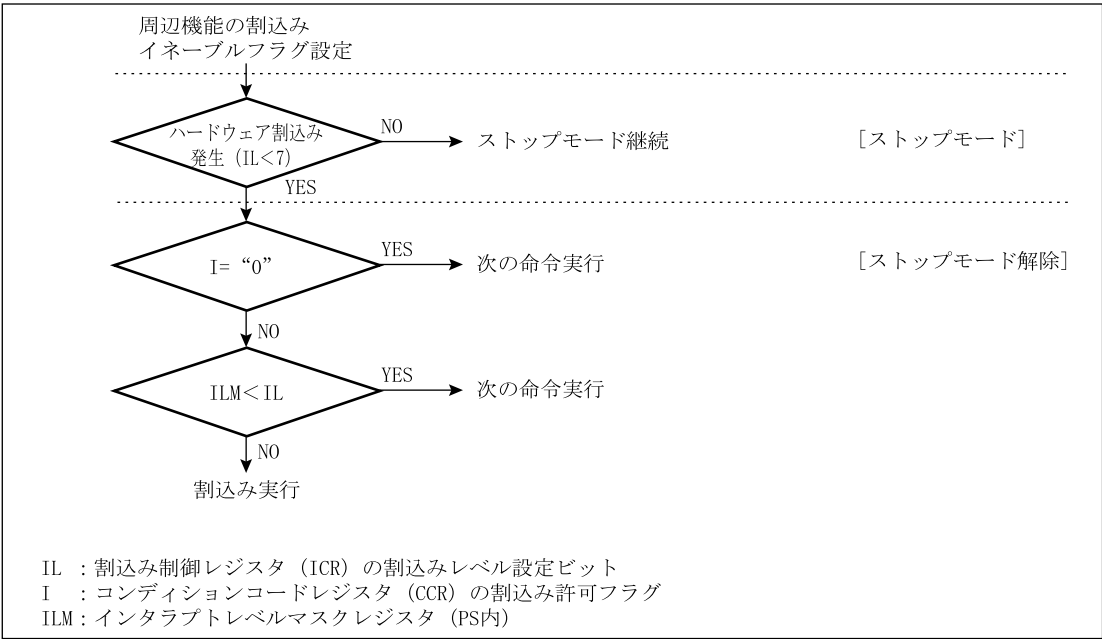


図 5.5-3 外部割込み出力によるストップモードの解除フロー

5.7 スタンバイモード，ホールド，リセットの端子状態

スタンバイモード，ホールド，リセットの端子状態を示します。

ソフトウェアブルアップ抵抗

ソフトウェアでブルアップ抵抗の接続を設定している入出力端子は，出力設定にすることでブルアップ抵抗は切り離されます。

シングルチップモードの端子状態

表 5.7-1 シングルチップモードの各端子状態

端子名	スタンバイモード			ホールド時	リセット時
	スリープモード	ストップモード			
		SPL=0	SPL=1		
P00 ~ P07 P17 P20 ~ P27 P30 ~ P37 P40 P41 ~ P46 P50 ~ P57 P60 ~ P63	直前の状態を保持 ^{*2}	直前の状態を保持 ^{*2}	入力遮断 ^{*3} / 出力Hi-z ^{*4}	出力Hi-z ^{*4}	出力Hi-z ^{*4}
P10 ~ P16		入力可 ^{*1}			

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

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

*3: "入力遮断"とは，端子から入力動作を禁止する状態を意味します。

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

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

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

5.8 低消費電力モード使用上の注意

低消費電力モードを使用する場合は、次の注意が必要です。

- ・スタンバイモードへの遷移と割込み
- ・スタンバイモードの割込みによる解除
- ・ストップモードの外部割込みによる解除時
- ・発振安定待ち時間

スタンバイモードへの遷移と割込み

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

スタンバイモードの割込みによる解除

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

割込み要求に対応する割込みレベル設定ビット (ICR: IL2, IL1, IL0) の優先度が、インタラプトレベルマスクレジスタ (ILM) より強く、コンディションコードレジスタの割込み許可フラグが許可 (CCR: I="1") されている場合は、割込みが受け付けられ、割込み処理ルーチンを実行します。割込みが受け付けられない場合は、スタンバイモードを設定した命令の次の命令から動作を再開します。

< 注意事項 >

スタンバイモード解除直後に割込み処理ルーチンを実行させない場合は、スタンバイモード設定の前に、割込み禁止の設定が必要です。

ストップモードの外部割込みによる解除

外部割込みによりストップモードを解除する場合は、ストップモードへ遷移させる前に、DTP/割込み許可レジスタ (ENIR) および、要求レベル設定レジスタ (ELVR) の設定が必要です。

入力要因は"H"レベル、"L"レベル、立上がりエッジ、立下がりエッジから設定できます。

発振安定待ち時間

発振クロックの発振安定待ち時間

ストップモードは、発振クロックが停止するため、ストップモードを解除した場合は、発振安定待ち時間をとる必要があります。発振安定待ち時間は、クロック選択レジスタ（CKSCR）の発振安定待ち時間設定ビット（WS1, WS0）で設定します。

PLLクロックの発振安定待ち時間

CPUがメインクロックで動作している場合は、PLLクロックが停止しているため、動作クロックをメインクロックからPLLクロックに切り替えた場合は、PLLクロック発振安定待ち時間をとります。PLLクロック発振安定待ち中は、メインクロックで動作します。

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

第6章 割り込み

この章では、MB90560/565シリーズの割り込みと拡張インテリジェントI/Oサービス(EI²OS)について説明します。

- 6.1 割り込みの概要
- 6.2 割り込み要因と割り込みベクタ
- 6.3 割り込み制御レジスタ(ICR)と周辺機能(リソース)
- 6.4 ハードウェア割り込み
- 6.5 ソフトウェア割り込み
- 6.6 拡張インテリジェントI/Oサービス(EI²OS)による割り込み
- 6.7 例外処理
- 6.8 割り込み処理のスタック動作
- 6.9 割り込み処理のプログラム例

6.1 割込みの概要

MB90560/565シリーズには、3種類の割込み機能と例外処理とがあります。

- ・ハードウェア割込み
 - ・ソフトウェア割込み
 - ・拡張インテリジェントI/Oサービス (EI²OS) による割込み
 - ・例外処理
-

割込みの種類と機能

ハードウェア割込み

周辺機能 (リソース) の割込み要求に対して、割込み処理プログラムへ遷移します。詳細は、「6.4 ハードウェア割込み」を参照してください。

ソフトウェア割込み

ソフトウェア割込み命令 (INT命令) をプログラム上にて実行した場合、割込み処理プログラムへ遷移します。詳細は、「6.5 ソフトウェア割込み」を参照してください。

拡張インテリジェントI/Oサービス (EI²OS) による割込み

拡張インテリジェントI/Oサービス (EI²OS) では、事前に割込み制御レジスタ (ICR00 ~ ICR15) と拡張インテリジェントI/Oサービスディスクリプタ (ISD) を設定することにより、周辺機能 (リソース) に内蔵されているレジスタと、内蔵されているメモリの間で、データを転送することができます。

データ転送処理が終了した場合、割込み処理プログラムへ遷移します。詳細は、「6.6 拡張インテリジェントI/Oサービス (EI²OS) による割込み」を参照してください。

例外処理

定義されていない命令コードを実行した場合に例外処理を行います。

例外処理を行った場合は、現在処理中のレジスタ値をシステムスタックに退避させた後に例外処理ルーチンへ分岐します。詳細は、「6.7 例外処理」を参照してください。

6.2 割込み要因と割込みベクタ

MB90560/565シリーズには、256種類の割込み要因に対応する機能があり、メモリの最上位アドレスに256組の割込みベクタテーブルが割り当てられています。

ソフトウェア割込みは、256種類の割込み命令（INT0～INT255）を設定できます。ただし、INT8はリセットベクタ割込みと、INT10は例外処理と共有されています。また、INT11～INT42までは、周辺機能（リソース）の割込みと共有されています。

割込みベクタ

割込み処理を実行する場合に、参照する割込みベクタテーブルは、メモリ領域の最上位アドレス（"FFFC00_H"～"FFFFFF_H"）に割り当てられています。割込みベクタは、拡張インテリジェントI/Oサービス、例外処理、ハードウェア割込み、およびソフトウェア割込みを共有しています。ソフトウェア割込み命令、割込み番号と割込みベクタの割り当てを表 6.2-1に示します。

表 6.2-1 割込みベクター一覧表

ソフトウェア 割込み命令	ベクタ アドレス L	ベクタ アドレス M	ベクタ アドレス H	モード データ	割込み No	ハードウェア割込み
INT0	FFFFC _H	FFFFFD _H	FFFFFE _H	未使用	#0	なし
⋮	⋮	⋮	⋮	⋮	⋮	⋮
INT7	FFFFE0 _H	FFFFE1 _H	FFFFE2 _H	未使用	#7	なし
INT8	FFFFDC _H	FFFFDD _H	FFFFDE _H	FFFFDF _H	#8	リセットベクタ
INT9	FFFFD8 _H	FFFFD9 _H	FFFFDA _H	未使用	#9	なし
INT10	FFFFD4 _H	FFFFD5 _H	FFFFD6 _H	未使用	#10	例外処理
INT11	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	未使用	#11	ハードウェア割込み#0
INT12	FFFFCC _H	FFFFCD _H	FFFFCE _H	未使用	#12	ハードウェア割込み#1
INT13	FFFFC8 _H	FFFFC9 _H	FFFFCA _H	未使用	#13	ハードウェア割込み#2
INT14	FFFFC4 _H	FFFFC5 _H	FFFFC6 _H	未使用	#14	ハードウェア割込み#3
⋮	⋮	⋮	⋮	⋮	⋮	⋮
INT254	FFFC04 _H	FFFC05 _H	FFFC06 _H	未使用	#254	なし
INT255	FFFC00 _H	FFFC01 _H	FFFC02 _H	未使用	#255	なし

参考：

ソフトウェア設計時に設定されない割込みベクタは、例外処理アドレスに設定してください。

割り込み要因と割り込みベクタ・割り込み制御レジスタ

表 6.2-2 ハードウェア割り込み要因と割り込みベクタ・割り込み制御レジスタ

割り込み要因	EI ² OS 対応	割り込みベクタ			割り込み制御レジスタ		優先 強度
		番号 ^{*1}		アドレス	ICR	アドレス	
リセット	×	#08	08 _H	FFFFDC _H			強 ↑

: 使用可能

× : 使用不可

: 使用可能, EI²OS停止機能付

: ICRを共有する割り込み要因を使用しない場合に使用可能

*1: 同時に同じレベルの割り込みを出力した場合は, 割り込みベクタ番号の小さい割り込み要因を優先します。

割り込み制御レジスタ（ICR）と割り込み要因

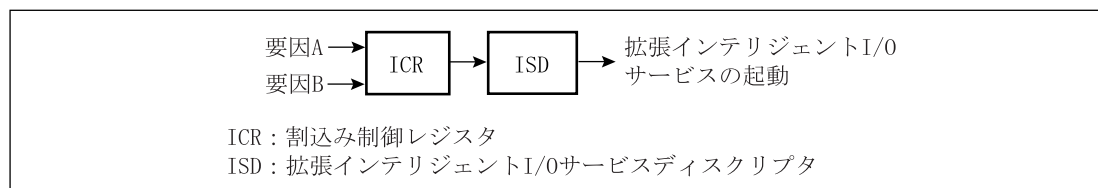


図 6.2-1 割り込み制御レジスタ（ICR）と割り込み要因の関係

図 6.2-1に示したように、割り込みレベルは割り込み制御レジスタ（ICR）単位で設定しますので、同じ割り込み制御レジスタ（ICR）に接続されている要因A，要因Bは同じ割り込みレベルが設定されます。

同じ割り込みレベルの割り込み要因を出力した場合は、割り込みベクタ番号の小さい割り込み要因が優先されます。

拡張インテリジェントI/Oサービスを要因Aで起動する場合は、必ず要因Bの割り込み要求出力を周辺機能（リソース）で禁止に設定してください。

要因Bの割り込み要求を周辺機能（リソース）で許可に設定した場合は、要因Bの割り込み要求を出力した場合でも、拡張インテリジェントI/Oサービスが起動されません。

6.3 割込み制御レジスタ (ICR) と周辺機能 (リソース)

割込み制御レジスタ (ICR00 ~ ICR15) は、割込み機能を持つすべての周辺機能 (リソース) に対応します。割込み制御レジスタは、割込みと拡張インテリジェント I/O サービス (EI²OS) の制御を行います。

割込み制御レジスタ一覧

表 6.3-1 割込み制御レジスタ一覧

アドレス	レジスタ	略称	対応する周辺機能 (リソース)
0000B0 _H	割込み制御レジスタ00	ICR00	A/Dコンバータ
0000B1 _H	割込み制御レジスタ01	ICR01	アウトプットコンペア0 8/16ビットPPGタイマ0
0000B2 _H	割込み制御レジスタ02	ICR02	アウトプットコンペア1 8/16ビットPPGタイマ1
0000B3 _H	割込み制御レジスタ03	ICR03	アウトプットコンペア2 8/16ビットPPGタイマ2
0000B4 _H	割込み制御レジスタ04	ICR04	アウトプットコンペア3 8/16ビットPPGタイマ3
0000B5 _H	割込み制御レジスタ05	ICR05	アウトプットコンペア4 8/16ビットPPGタイマ4
0000B6 _H	割込み制御レジスタ06	ICR06	アウトプットコンペア5 8/16ビットPPGタイマ5
0000B7 _H	割込み制御レジスタ07	ICR07	DTP/外部割込み0, 1, 2, 3
0000B8 _H	割込み制御レジスタ08	ICR08	DTP/外部割込み4, 5, 6, 7
0000B9 _H	割込み制御レジスタ09	ICR09	8ビットタイマ0, 1, 2 16ビットリロードタイマ0
0000BA _H	割込み制御レジスタ10	ICR10	16ビットフリーランタイムオーバーフロー 16ビットリロードタイマ1
0000BB _H	割込み制御レジスタ11	ICR11	インプットキャプチャ0, 1 16ビットフリーランタイムクリア
0000BC _H	割込み制御レジスタ12	ICR12	インプットキャプチャ2, 3 タイムベースタイマ
0000BD _H	割込み制御レジスタ13	ICR13	UART1
0000BE _H	割込み制御レジスタ14	ICR14	UART0
0000BF _H	割込み制御レジスタ15	ICR15	フラッシュメモリ, 遅延割込み出力モジュール

割込み制御レジスタの機能

割込み制御レジスタ（ICR）では，4種類の設定を行うことができます。

- 周辺機能（リソース）の割込みレベルの設定ができます。
- 周辺機能（リソース）の割込みを割込みに設定するか，拡張インテリジェントI/Oサービスに設定するかを設定できます。
- 拡張インテリジェントI/Oサービス（EI²OS）のディスクリプタアドレスの設定ができます。
- 拡張インテリジェントI/Oサービス（EI²OS）処理ステータスの表示ができます。

割込み制御レジスタ（ICR）は，書込みを行う場合と，読出しを行う場合では機能が異なります。

< 注意事項 >

割込み制御レジスタ（ICR）を設定する場合は，SETB,CLRBのリードモディファイライト系命令でのアクセスはできません。

6.3.1 割り込み制御レジスタ (ICR00 ~ ICR15)

割り込み制御レジスタは、割り込み要求を出力した場合に、割り込み処理または拡張インテリジェントI/Oサービス処理を設定することができます。割り込み制御レジスタは書込みを行う場合と読出しを行う場合では、機能が異なります。

割り込み制御レジスタ (ICR00 ~ ICR15)

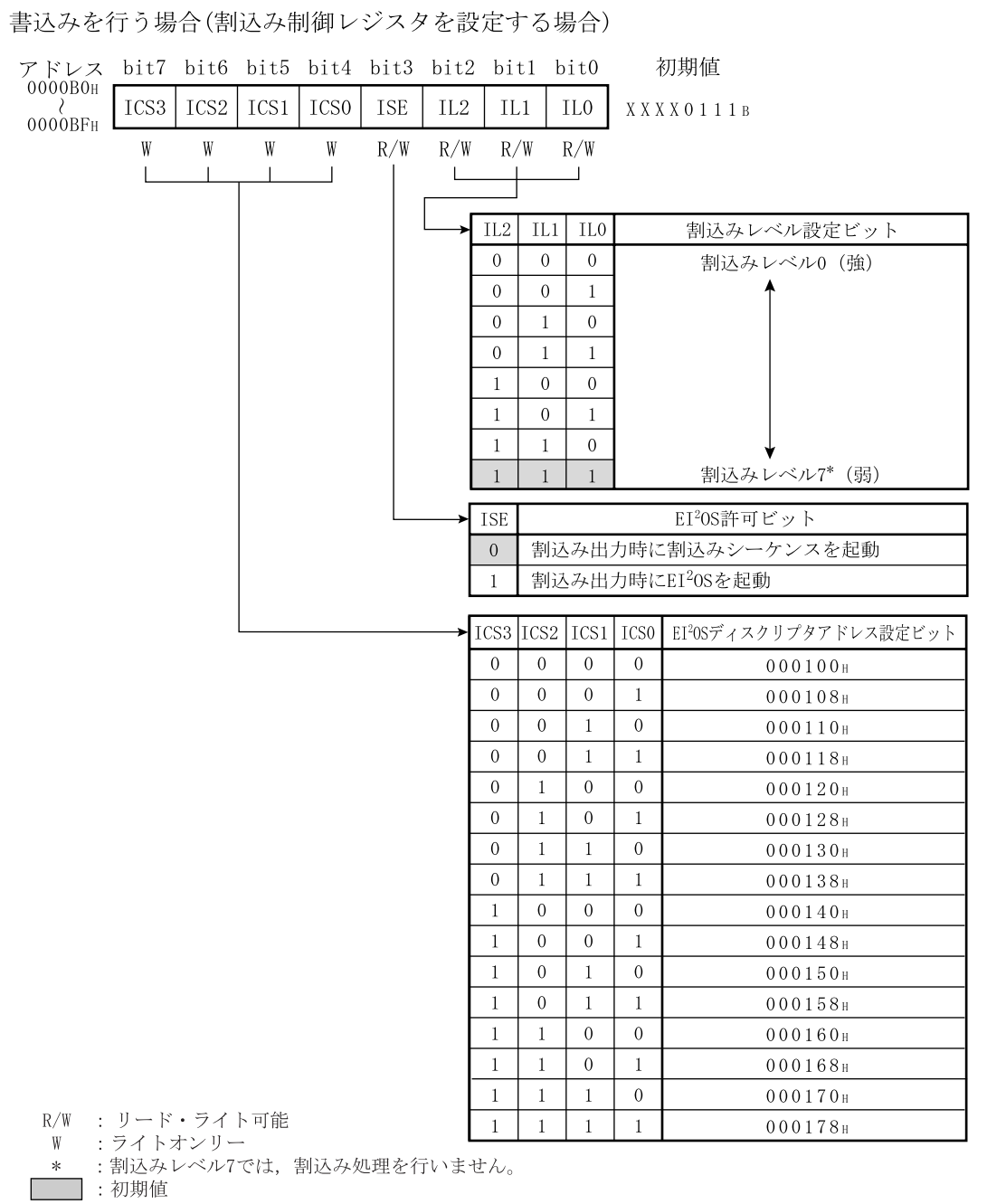


図 6.3-1 割り込み制御レジスタ (ICR00 ~ ICR15) [書込みを行う場合]

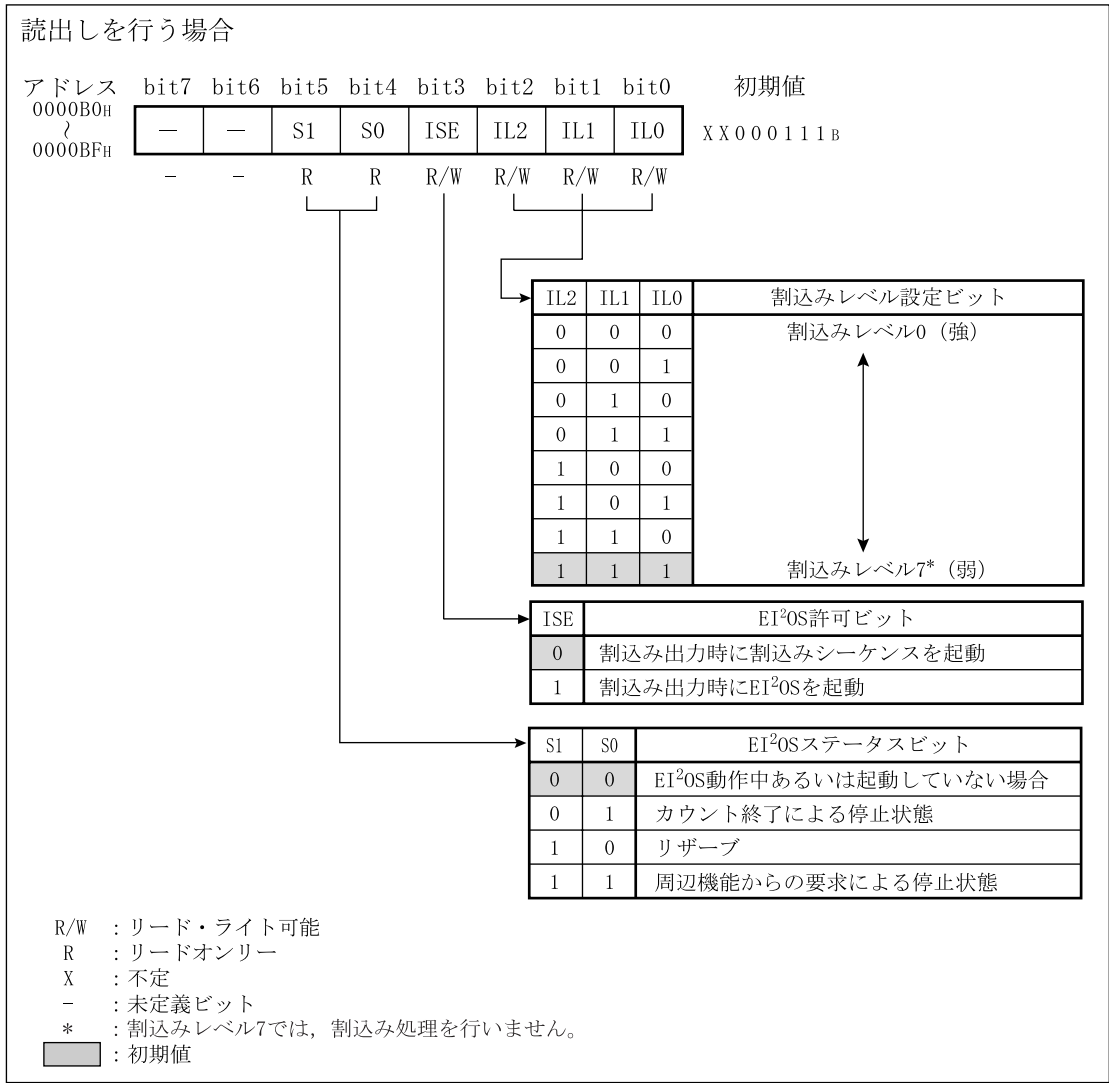


図 6.3-2 割り込み制御レジスタ（ICR00～ICR15）[読出しを行う場合]

6.3.2 割り込み制御レジスタ (ICR) の機能

割り込み制御レジスタ (ICR00 ~ ICR15) は、以下に示す設定を行うことができます。

- ・ 割り込みレベル設定
- ・ 拡張インテリジェントI/Oサービス (EI²OS) 許可設定
- ・ 拡張インテリジェントI/Oサービス (EI²OS) ディスクリプタアドレス設定
- ・ 拡張インテリジェントI/Oサービス (EI²OS) 動作ステータス表示

割り込み制御レジスタ (ICR) の構成

割り込み制御レジスタ (ICR) に書込みを行う場合									初期値
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0000B0H └ 0000BFH	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	XXXX0111B
	W	W	W	W	R/W	R/W	R/W	R/W	
割り込み制御レジスタ (ICR) から読出しを行う場合									初期値
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0000B0H └ 0000BFH	—	—	S1	S0	ISE	IL2	IL1	IL0	XX000111B
	—	—	R	R	R/W	R/W	R/W	R/W	

R/W : リード・ライト可能
 R : リードオンリー
 W : ライトオンリー
 X : 不定
 — : 未定義ビット

図 6.3-3 割り込み制御レジスタ (ICR) の構成

参考：

EI²OS ディスクリプタアドレス設定ビット (ICS3 ~ ICS0) への設定は、拡張インテリジェントI/Oサービス (EI²OS) を起動する場合に有効になります。EI²OS を起動させる場合は、EI²OS 許可ビット (ISE) に"1"を設定し、起動させない場合はEI²OS 許可ビット (ISE) に"0"を設定してください。EI²OSを起動させない場合は、EI²OS ディスクリプタアドレス設定ビット (ICS3 ~ ICS0) を設定する必要はありません。

割り込み制御レジスタの機能

割り込みレベル設定

周辺機能（リソース）の割り込みレベルが設定できます。リセットによりレベル7に初期化されます。

表 6.3-2 割り込みレベル設定ビットと割り込みレベル

IL2	IL1	IL0	割り込みレベル
0	0	0	0（最強割り込み） ↑ ↓ 6（最弱割り込み）
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	7（割り込みなし）
1	1	1	

拡張インテリジェントI/Oサービス（EI²OS）許可設定

割り込み要求を出力した場合にEI²OS許可ビット（ISE）に"1"が設定されている場合は、EI²OSが起動され、EI²OS許可ビット（ISE）に"0"が設定されている場合は、割り込みシーケンスが起動されます。EI²OSの処理が終了した場合は、EI²OS許可ビット（ISE）は"0"にクリアされます。周辺機能（リソース）にEI²OS機能がない場合には、ソフトウェアでEI²OS許可ビット（ISE）を"0"に設定してください。EI²OS許可ビット（ISE）はリセットで"0"にクリアされます。

拡張インテリジェントI/Oサービス（EI²OS）ディスクリプタアドレス設定

EI²OSディスクリプタアドレス設定ビット（ICS3～ICS0）は、書込みを行う場合に有効なビットです。EI²OSのディスクリプタアドレスを設定します。EI²OSディスクリプタアドレス設定ビット（ICS3～ICS0）に値を設定することで、EI²OSディスクリプタのアドレスを設定します。EI²OSディスクリプタアドレス設定ビット（ICS3～ICS0）はリセットで、"0000B"に初期化されます。

表 6.3-3 EI²OSディスクリプタアドレス設定ビットとディスクリプタアドレスの対応

ICS3	ICS2	ICS1	ICS0	ディスクリプタアドレス
0	0	0	0	000100 _H
0	0	0	1	000108 _H
0	0	1	0	000110 _H
0	0	1	1	000118 _H
0	1	0	0	000120 _H
0	1	0	1	000128 _H
0	1	1	0	000130 _H
0	1	1	1	000138 _H
1	0	0	0	000140 _H
1	0	0	1	000148 _H
1	0	1	0	000150 _H
1	0	1	1	000158 _H
1	1	0	0	000160 _H
1	1	0	1	000168 _H
1	1	1	0	000170 _H
1	1	1	1	000178 _H

拡張インテリジェントI/Oサービス (EI²OS) 動作ステータス表示

EI²OSステータスビット (S1, S0) は読出しを行う場合に有効なビットです。EI²OSを起動した場合にEI²OSステータスビット (S1, S0) を読出するとEI²OSが動作中か終了しているかを判別することができます。リセットで"00_B"に初期化されます。

表 6.3-4 EI²OSステータスビットとEI²OSステータスの関係

S1	S0	EI ² OSステータス
0	0	EI ² OS動作中あるいは起動していない場合
0	1	カウント終了による停止状態
1	0	リザーブ
1	1	周辺機能 (リソース) からの要求による停止状態

6.4 ハードウェア割り込み

ハードウェア割り込みは、周辺機能（リソース）にて出力された割り込み要求により、CPUが実行していたプログラム処理を中断し、事前に設定された割り込み処理プログラムに遷移します。拡張インテリジェントI/Oサービス（EI²OS）もハードウェア割り込みとして処理します。

ハードウェア割り込み

ハードウェア割り込み機能

ハードウェア割り込みは、周辺機能（リソース）にて出力された割り込み要求の割り込みレベルと、インタラプトレベルマスクレジスタ（PS：ILM）を比較し、Iフラグ（PS：I）の内容を参照して、割り込みを受け付けつけるか、受け付けないかを判定します。

ハードウェア割り込みが受け付けられた場合は、ダイレクトページレジスタ（DPR）、アキュムレータ（A）、プログラムカウンタ（PC）、プロセッサステータスレジスタ（PS）、および各バンクレジスタ（ADB,DTB,PCB）をシステムスタックへ退避させ、インタラプトレベルマスクレジスタ（ILM）にICRレジスタで設定されている割り込みレベルを格納し、割り込みベクタへ分岐して割り込み処理プログラムを実行します。

多重割り込み

ハードウェア割り込みは割り込み処理プログラム実行中でも起動できます。

拡張インテリジェントI/Oサービス（EI²OS）

EI²OSは、メモリ - I/Oレジスタ間のデータ転送機能です。拡張インテリジェントI/Oサービスディスクリプタに転送を終了した場合に、ハードウェア割り込みを起動します。EI²OSは多重に起動されません。EI²OSを処理している場合は、割り込み要求、EI²OS要求を受け付けません。EI²OSの処理が終了した場合に、割り込み要求、EI²OS要求が受け付けられます。

外部割り込み

外部割り込みは、外部端子から割り込み要求を出力させることのできる回路（DTP / 外部割り込み回路）にて割り込み要求を検出した場合に、ハードウェア割り込みとして受け付けられます。

割り込みベクタ

割り込み処理を実行する場合に参照する割り込みベクタは、メモリの"FFFC00_H" ~ "FFFFFF_H"に割り当てられています。

割り込み番号と割り込みベクタの割当てについては、「6.2 割り込み要因と割り込みベクタ」を参照してください。

ハードウェア割り込みの構造

ハードウェア割り込みに関連する機構は、表 6.4-1に示す4種類（7カ所）に分かれています。ハードウェア割り込みを使用する場合は、事前に、ユーザープログラムにて4種類（7カ所）の設定を行う必要があります。

表 6.4-1 ハードウェア割込みに関連する機構

	ハードウェア割込みに関連する機構	機 能
周辺機能（リソース）	割込み許可ビット 割込み要求フラグビット	周辺機能（リソース）の割込み要求制御
割込みコントローラ	割込み制御レジスタ（ICR）	割込みレベルの設定とEIOSの制御
CPU	割込み許可フラグ（I）	割込み許可 / 禁止の判定制御
	インタラプトレベルマスクレジスタ（ILM）	割込み要求レベルの設定値と出力要因の比較
	マイクロコード	割込み処理ルーチンの実行
メモリ上の "FFFC00H" ～ "FFFFFFH"	割込みベクタテーブル	割込み処理時の分岐先アドレスを格納

ハードウェア割込み禁止

ハードウェア割込みは、下記条件の場合、割込み要求の受け付けを禁止します。

周辺機能(リソース)制御レジスタへ書き込み中のハードウェア割込み受け付け禁止

周辺機能（リソース）制御レジスタへ書き込み中はハードウェア割込み要求を受け付けません。

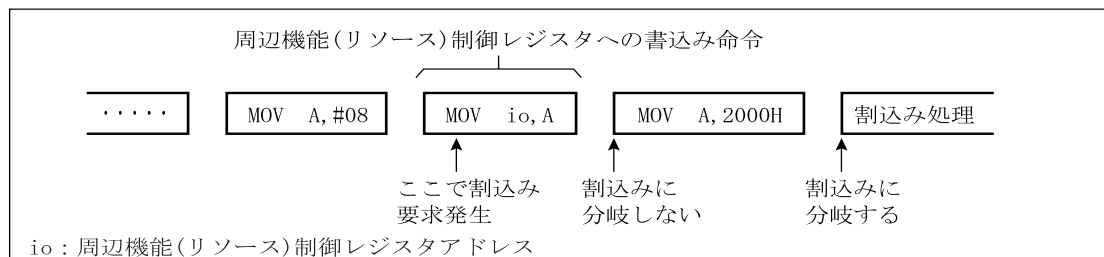


図 6.4-1 周辺機能（リソース）制御レジスタへ書き込み中のハードウェア割込み要求

割込み抑止命令のハードウェア割込み受け付け禁止

表 6.4-2に示す10種類のハードウェア割込み抑止命令は、ハードウェア割込み要求の有無の検出をせず、割込み要求を無視します。ハードウェア割込み抑止命令を実行中にハードウェア割込み要求を出力した場合、割込み処理が行われるのは、ハードウェア割込み抑止命令を実行した後のハードウェア割込み抑止命令以外の命令が実行された後に受け付けられます。

表 6.4-2 ハードウェア割込み抑止命令

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

ソフトウェア割込み実行中のハードウェア割込み受け付け禁止

ソフトウェア割込みを起動した場合、Iフラグを"0"にクリアするため、ハードウェア割込み要求は受け付けません。

6.4.1 ハードウェア割り込みの動作

割り込み要求出力から，割り込み処理完了までの動作について説明します。

ハードウェア割り込みの起動

周辺機能（リソース）の動作（割り込み要求の出力）

ハードウェア割り込み要求の機能を持つ周辺機能（リソース）は，割り込みの要求のあり/なしを示す「割り込み要求フラグビット」と，CPUへの割り込み要求の許可/禁止を設定する「割り込み許可ビット」が各周辺機能（リソース）制御レジスタに定義されています。周辺機能（リソース）にて事前に設定された割り込み要因を検出した場合，割り込み要求フラグビットに"1"がセットされ，割り込み許可ビットの設定が，CPUへの割り込み要求を許可している場合は，割り込みコントローラへ割り込み要求を出力します。

割り込みコントローラの動作（割り込み要求の制御）

割り込みコントローラは，割り込み要求の割り込みレベル（IL）を比較し，最も強いレベルの要求を設定します。同じレベルの割り込み要求を出力した場合は，割り込み番号が小さいものを優先します（表 6.2-1参照）。

CPUの動作（割り込み要求の受け付けと割り込み処理）

CPUは受け取った割り込みレベル値（ICR：IL2～IL0）とインタラプトレベルマスクレジスタ値（ILM）を比較し，IL2～IL0 < ILMで，割り込みが許可（PS：CCR：I="1"）されている場合は，実行中の命令を終了した後に，割り込み処理を行います。また，割り込み制御レジスタ（ICR）のEI²OS許可ビット（ISE）に"0"が設定されている場合は割り込み処理を実行し，"1"に設定されている場合はEI²OSを起動します。

割り込み処理では，システムスタック（SSBとSSPの示すシステムスタック空間）へ専用レジスタ（A,DPR,ADB,DTB,PCB,PC,PSの12バイト）の内容を退避させた後に，割り込みベクタのプログラムカウンタ値（PCB,PC）とILMの更新を行ってから，スタックフラグ（S）に"1"をセットし，システムスタックを有効にします。

ハードウェア割り込みからの復帰

割り込み処理プログラムで，割り込み要因となった周辺機能（リソース）の割り込み要求フラグビットに"0"を設定し，RETI命令を実行させた場合，システムスタックに退避させているデータを専用レジスタに戻し，割り込み分岐前に実行していたプログラム処理へ復帰します。

ハードウェア割込みの動作

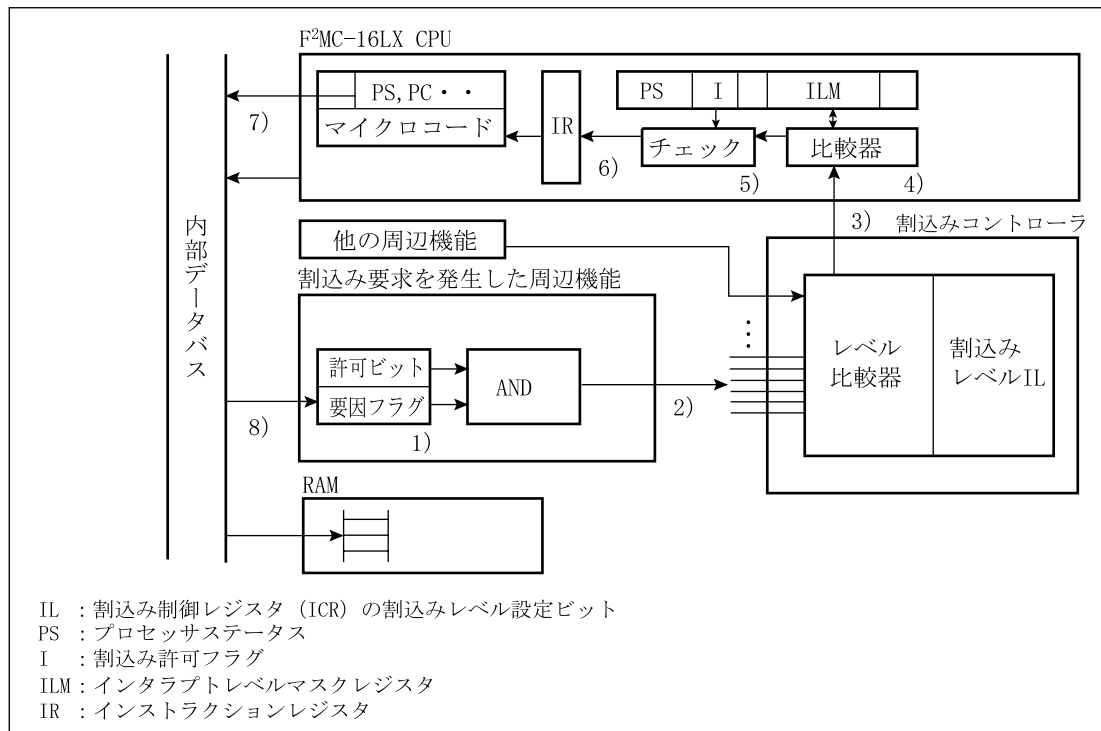


図 6.4-2 ハードウェア割込みの動作

- 1) 周辺機能（リソース）の内部で割込み要因を出力します。
- 2) 周辺機能（リソース）内の割込み許可ビットの設定が、割込みを許可している場合に、周辺機能（リソース）から割込みコントローラへ割込み要求を出力します。
- 3) 割込み要求を周辺機能（リソース）から受け取った割込みコントローラは、割込みの優先順位を判定してから、優先順位の最も高い割込み要求の割込みレベル（IL）をCPUへ転送します。
- 4) CPUは、割込みコントローラから要求のあった割込みレベル（IL）を、インタラプトレベルマスクレジスタ（ILM）と比較します。
- 5) 比較結果が設定されている割込み処理レベルより優先順位が高い場合、コンディションコードレジスタ（CCR）のIフラグの内容をチェックします。
- 6) コンディションコードレジスタのIフラグの内容をチェックした結果、割込み許可（CCR: I="1"）の場合、現在実行中の命令が終了するまで待ち、終了した場合、ILMに要求されたレベル（IL2～IL0）を設定します。
- 7) システムスタックへ専用レジスタの値を退避させ、割込み処理ルーチンへ分岐します。
- 8) 割込み処理ルーチン内のプログラムによって、周辺機能（リソース）の割込み要求フラグビットに"0"を設定して、RET1命令を実行させた場合に、システムスタックから専用レジスタへ退避させた値を戻して、割込み処理を終了します。

6.4.2 割り込み動作時の処理

周辺機能（リソース）から割り込み要求を出力し，CPUが割り込みを受け付けた場合は，現在実行中の命令が終了したあと，割り込み処理を行います。割り込み制御レジスタ（ICR）のEI²OS 許可ビット（ISE）に"0"が設定されている場合は，割り込み処理ルーチンを実行し，EI²OS 許可ビット（ISE）に"1"が設定されている場合は，拡張インテリジェントI/Oサービス（EI²OS）を起動します。INT命令によるソフトウェア割り込みを出力した場合は，現在実行中の命令を一時中断して，割り込み処理ルーチンを実行し，ハードウェア割り込みを禁止します。

割り込み動作時の処理

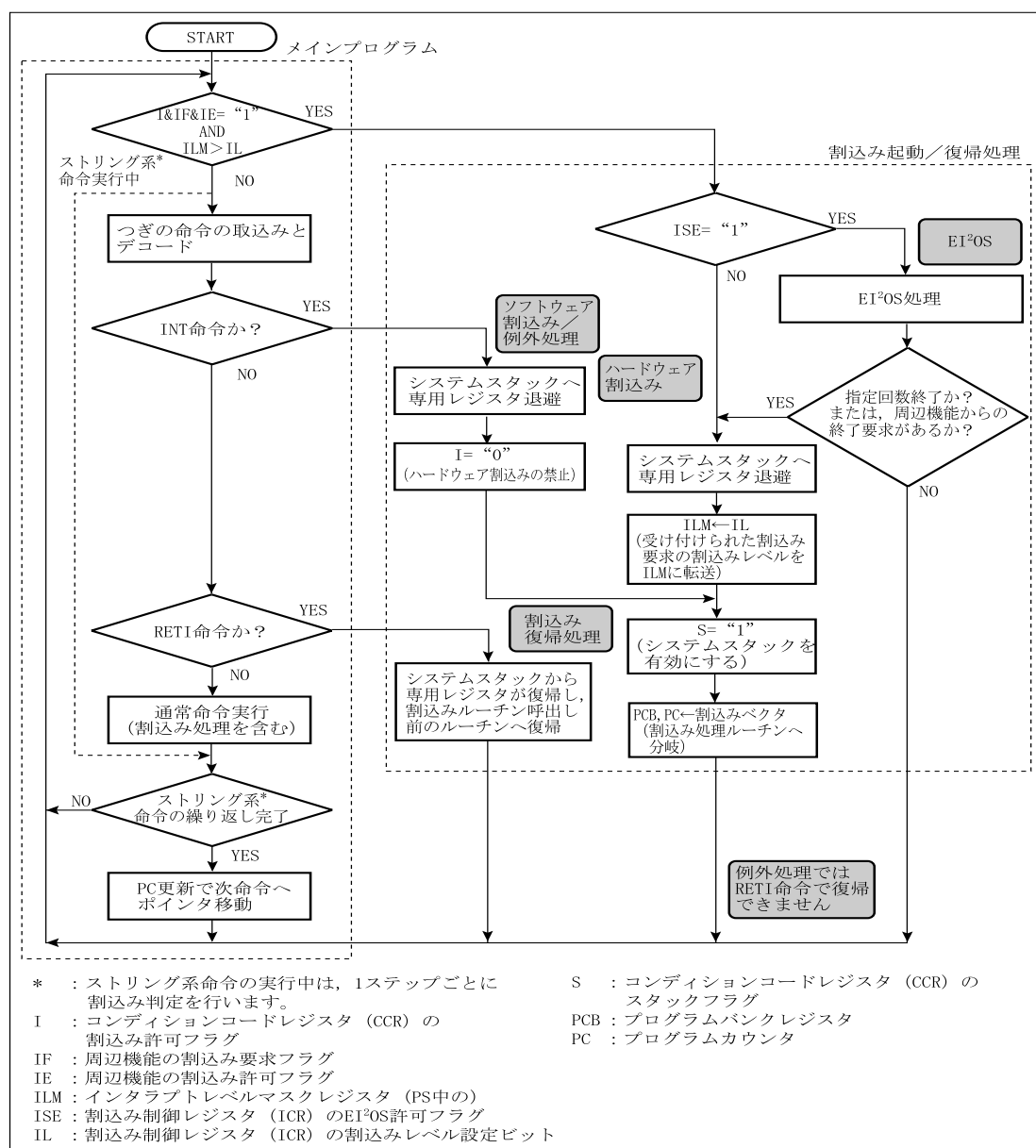


図 6.4-3 割り込み処理のフロー

6.4.3 ハードウェア割込み使用手順

ハードウェア割込みを使用する場合は、システムスタック領域、周辺機能（リソース）、および割込み制御レジスタ（ICR）を設定してください。

ハードウェア割込み使用手順

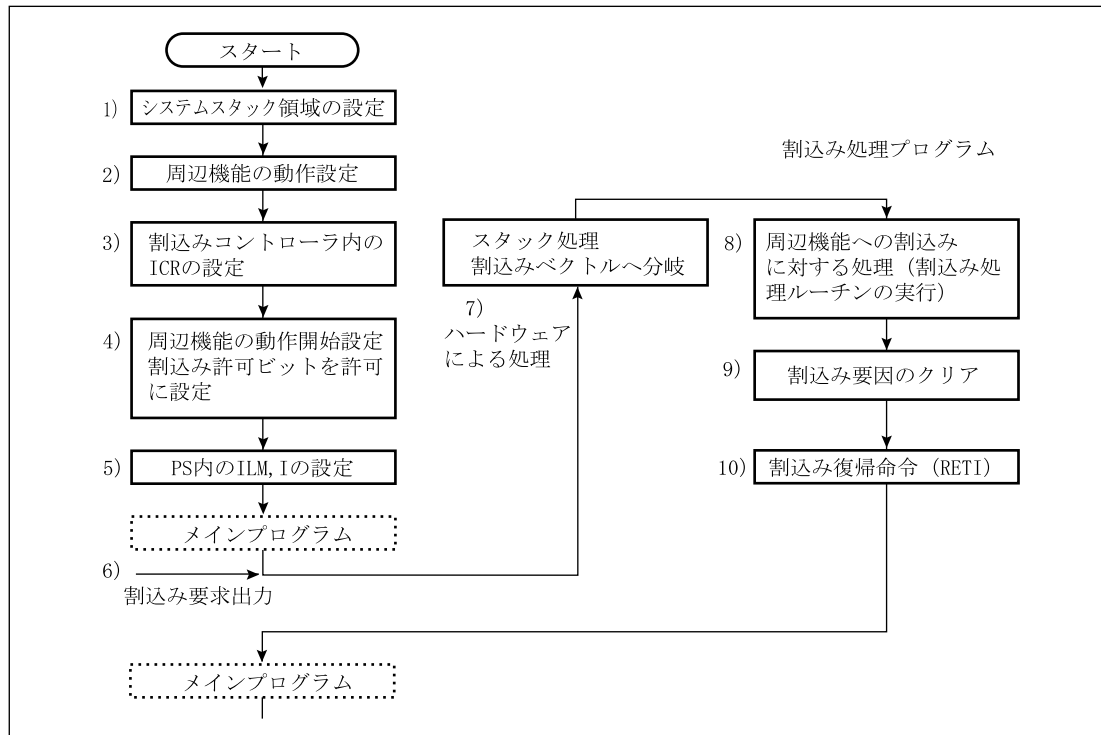


図 6.4-4 ハードウェア割込み使用手順

- 1) システムスタック領域を設定します。
- 2) 周辺機能（リソース）の動作を設定します。
- 3) 割込み制御レジスタ（ICR）の設定を行います。
- 4) 周辺機能（リソース）の割込み許可ビットを、割込み要求が出力できるように設定します。
- 5) 割込みレベルマスクレジスタ（ILM）、および割込み許可フラグ（I）を割込み受付け可能に設定します。
- 6) 周辺機能（リソース）の割込み要求を検出した場合、ハードウェア割込み要求を出力します。
- 7) 割込み処理ハードウェアで、専用レジスタ値をシステムスタックへ退避させ、割込み処理プログラムに分岐します。
- 8) 割込み処理プログラムで、割込み要求出力に対する周辺機能（リソース）への処理を行います。
- 9) 周辺機能（リソース）の割込み要求を解除します。
- 10) 割込み復帰命令 (RETI 命令) を実行し、分岐前のプログラムに復帰します。

6.4.4 多重割り込み

ハードウェア割り込みでは、周辺機能（リソース）からの複数の割り込み要求に対して、多重割り込みを設定することができますが、拡張インテリジェントI/Oサービスの多重起動はできません。

多重割り込み

多重割り込み動作

現在実行中の割り込み要求より強い割り込みレベルの割り込み要求を出力した場合は、現在実行中の割り込み処理を中断し、強い割り込み要求を実行します。現在実行中の割り込み要求より強いレベルの割り込み処理が終了した場合、最初に行っていた割り込み処理を行います。

割り込み処理実行中に、現在実行中の割り込みと同等か、より弱いレベルの割り込み処理を出力した場合、コンディションコードレジスタ（CCR）のIフラグかインタラプトレベルマスクレジスタ（ILM）を変更しないかぎり現在の割り込み処理が終了するまで新しい割り込み要求は保留され、現在の割り込み処理が終了した場合、保留されている割り込み要求を実行します。

割り込み処理ルーチン内でコンディションコードレジスタ（CCR）のIフラグを割り込み禁止（CCR：I="0"）に設定するか、インタラプトレベルマスクレジスタ（ILM）を割り込み禁止（ILM="000b"）に設定すると割り込みの多重起動を禁止することができます。

< 注意事項 >

-
- ・割り込みレベルは0～7まで設定できますが、レベル7に設定した場合、CPUは割り込み要求を受け付けません。
 - ・拡張インテリジェントI/Oサービス（EI²OS）は多重起動できません。拡張インテリジェントI/Oサービス（EI²OS）の処理中に出力された割り込み要求、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。
-

多重割込みの例

多重割込み処理の例として、A/Dコンバータよりタイマ割込みを優先させる場合を想定し、A/Dコンバータの割込みレベルを2に、タイマ割込みレベルを1に設定します。A/Dコンバータの割込み処理中にタイマ割込みを出力した場合は、図 6.4-5に示す割込み処理を行います。

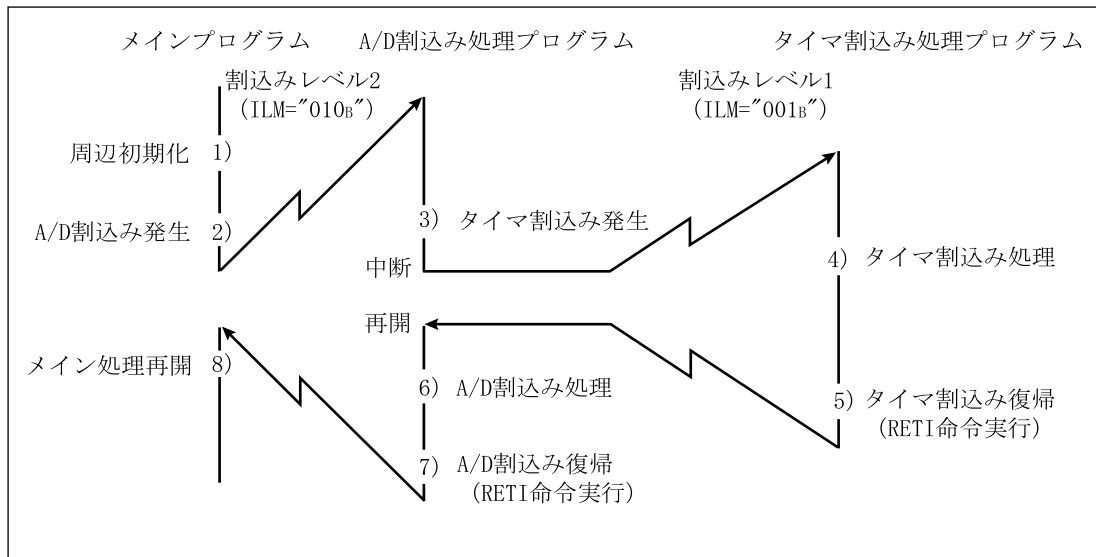


図 6.4-5 多重割込み処理の実行例

6.4.5 ハードウェア割り込み処理時間

ハードウェア割り込み要求を出力して、割り込み処理ルーチンが実行されるまでの処理時間について説明します。

ハードウェア割り込み処理時間

周辺機能（リソース）から割り込み要求を出力し、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（割り込み処理準備に要する時間）が必要です。

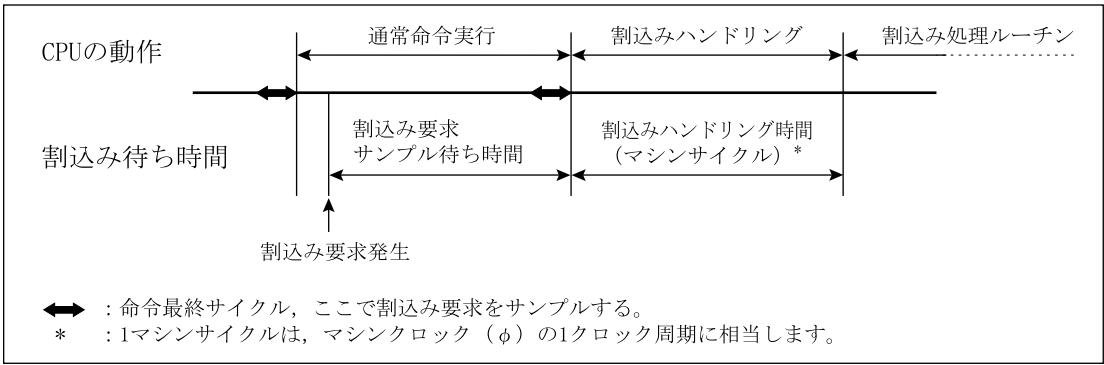


図 6.4-6 割り込み処理時間

割り込み要求サンプル待ち時間

周辺機能（リソース）から割り込み要求を出力し、現在実行中の命令が終了するまでの時間を示します。割り込み要求は、実行している命令の最終サイクルで割り込み要求を出力しているか出力していないかをサンプリングします。命令の実行中は、割り込み要求を認識することができず、割り込み要求サンプル待ち時間が発生します。

割り込みハンドリング時間（マシンサイクル）

CPUは、割り込み要求を受け付けてから、専用レジスタの値をシステムスタックへ退避させ、割り込みベクタの取込みを行うため、割り込みハンドリング時間を必要とします。割り込みハンドリング時間は、以下の式によって求められます。

割り込み起動の場合：	$= 24 + 6 \times Z$	マシンサイクル
割り込み復帰の場合：	$= 11 + 6 \times Z$	マシンサイクル (RETI命令)

割り込みハンドリング時間は、スタックポインタのアドレスによって異なります。

表 6.4-3 割込みハンドリング時間の補正值 (Z)

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

6.5 ソフトウェア割り込み

ソフトウェア割り込み命令を実行した場合、メインプログラムから割り込み処理用プログラムへ遷移します。ソフトウェア割り込みの実行中は、ハードウェア割り込みを受け付けません。

ソフトウェア割り込みの起動

ソフトウェア割り込みの起動

ソフトウェア割り込みの起動には、INT命令を使用します。ソフトウェア割り込み要求にはハードウェア割り込みのような割り込み要求フラグビット、および許可許可ビットはありませんので、INT命令を実行した場合に割り込み要求を出力します。

ハードウェア割り込みの抑止

INT命令には割り込みレベルがありませんので、インタラプトレベルマスクレジスタ（ILM）は更新されません。INT命令実行中は、コンディションコードレジスタ（CCR）のIフラグを"0"に設定し、ハードウェア割り込みをマスクします。ソフトウェア割り込み処理中にハードウェア割り込みを許可する場合は、ソフトウェア割り込み処理ルーチンでコンディションコードレジスタ（CCR）のIフラグに"1"を設定してください。

ソフトウェア割り込みの動作

CPUがINT命令を取込み実行した場合、ソフトウェア割り込み処理用マイクロコードを起動します。ソフトウェア割り込み処理用マイクロコードにより、CPU内部のレジスタ類をシステムスタックに退避させ、ハードウェア割り込みをマスク（CCR：I="0"）したあとに対応する割り込みベクタへ分岐します。

割り込み番号と割り込みベクタの割り当てについては、「6.2 割り込み要因と割り込みベクタ」を参照してください。

ソフトウェア割り込みからの復帰

割り込み処理プログラムの中で割り込み復帰命令（RETI命令）を実行した場合は、システムスタックに退避させているデータを専用レジスタに戻して、割り込み分岐前に実行していた処理に復帰します。

ソフトウェア割込みの動作

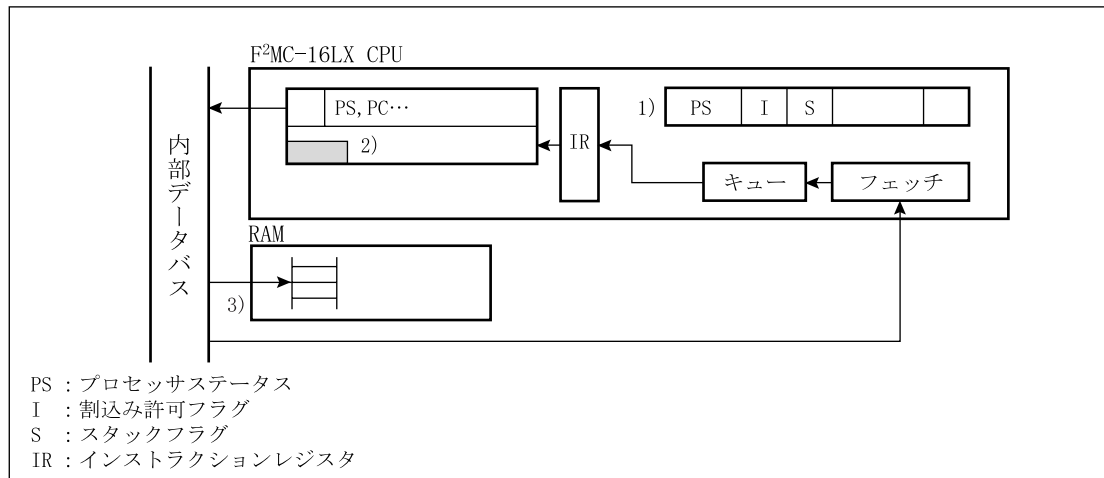


図 6.5-1 ソフトウェア割込みの動作

- 1) ソフトウェア割込み命令 (INT 命令) を実行させます。
- 2) 専用レジスタをシステムスタックに退避させ、ハードウェア割込みをマスクして割込みベクタへ分岐します。
- 3) ユーザの割込み処理ルーチン内の RETI 命令で割込み処理を終了します。

< 注意事項 >

プログラムバンクレジスタ (PCB) が "FF_H" に設定されている場合、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ので、ソフトウェアは CALLV 命令と INT #vct8 命令のアドレスの重複を解消するように設計してください。

6.6 拡張インテリジェントI/Oサービス(EI²OS)による割込み

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

拡張インテリジェントI/Oサービス(EI²OS)

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

拡張インテリジェントI/Oサービス(EI²OS)の利点

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

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

拡張インテリジェントI/Oサービス(EI²OS)の終了割込み

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

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

割込み番号や割込みベクタは、各周辺で固定されています。詳細は「6.2 割込み要因と割込みベクタ」を参照ください。

割込み制御レジスタ(ICR)

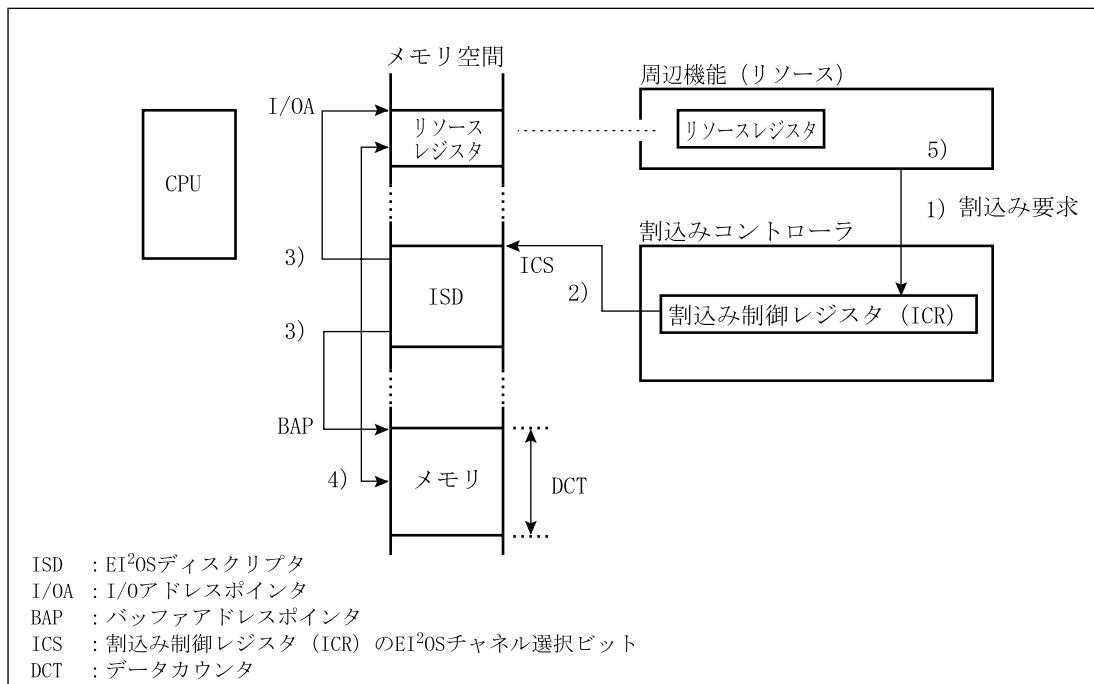
EI²OSの起動、EI²OSのチャンネルを設定することができます。また、EI²OS終了時のEI²OSステータス表示を行います。

拡張インテリジェントI/Oサービス(EI²OS)ディスクリプタ(ISD)

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

< 注意事項 >

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

拡張インテリジェントI/Oサービス (EI²OS) の動作図 6.6-1 拡張インテリジェントI/Oサービス (EI²OS) の動作

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

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

拡張インテリジェントI/Oサービス (EI²OS) ディスクリプタ (ISD) は、RAMの"000100H" ~ "00017FH"に存在し、8バイト×16チャンネルで構成されています。

拡張インテリジェントI/Oサービス(EI²OS)ディスクリプタ (ISD) の構成
ISDは、8バイト×16チャンネルで構成されています。



図 6.6-2 EI²OSディスクリプタ (ISD) の構成

表 6.6-1 チャンネル番号とディスクリプタアドレスの対応

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

* : ISDのアドレスは、8バイトのうちの先頭アドレスを示しています。

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

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

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

各レジスタの初期値は、不定です。

データカウントレジスタ(DCT)

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

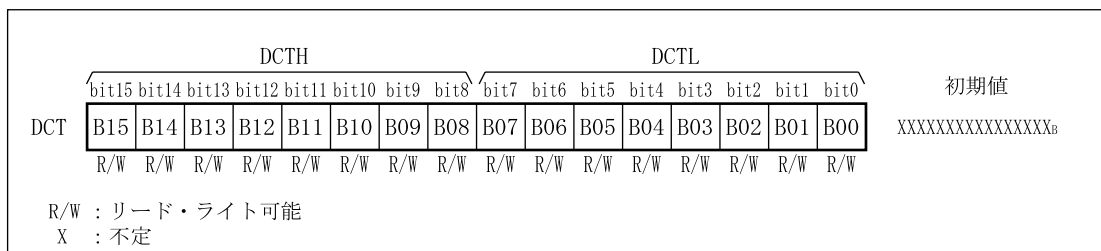


図 6.6-3 データカウントレジスタ(DCT)の構成

I/Oレジスタアドレスポインタレジスタ(I/OA)

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

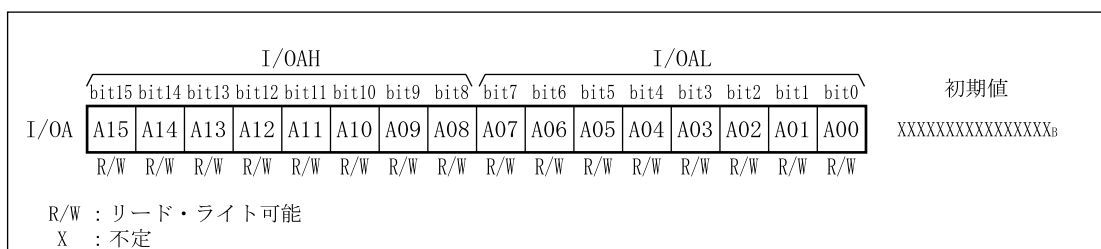


図 6.6-4 I/Oレジスタアドレスポインタレジスタ(I/OA)の構成

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

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

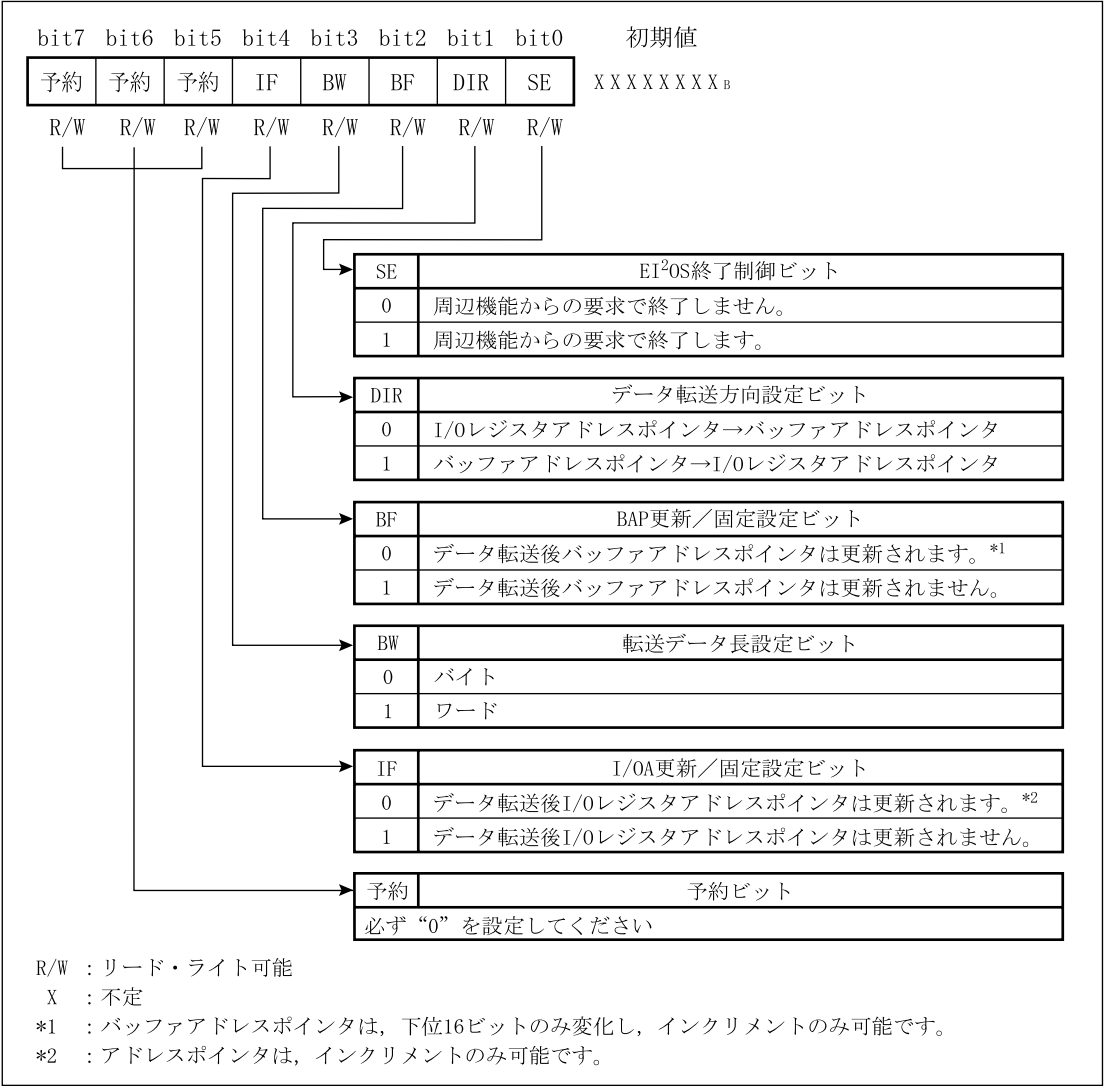


図 6.6-5 EI²OSステータスレジスタ (ISCS) の構成

バッファアドレスポインタレジスタ (BAP)

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

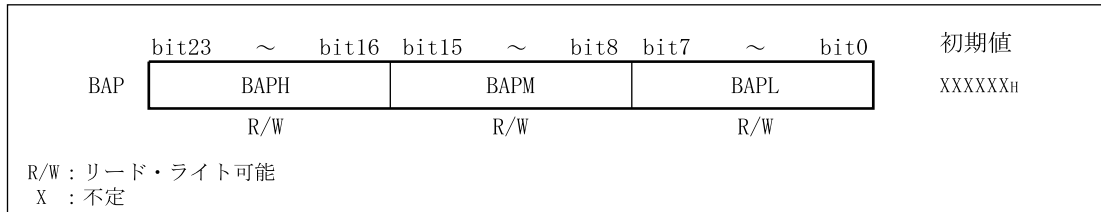


図 6.6-6 バッファアドレスポインタレジスタ (BAP) の構成

参考：

- ・データカウントレジスタ(DCT)で設定できる最大転送回数は、65,536回(64Kバイト)です。
- ・I/Oアドレスポインタレジスタ(I/OA)で設定できる領域は、"000000H" ~ "00FFFFH"です。
- ・バッファアドレスポインタレジスタ(BAP)で設定できる領域は、"000000H" ~ "FFFFFFH"です。

6.6.3 拡張インテリジェントI/Oサービス (EI²OS) の動作

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

拡張インテリジェントI/Oサービス (EI²OS) の処理手順

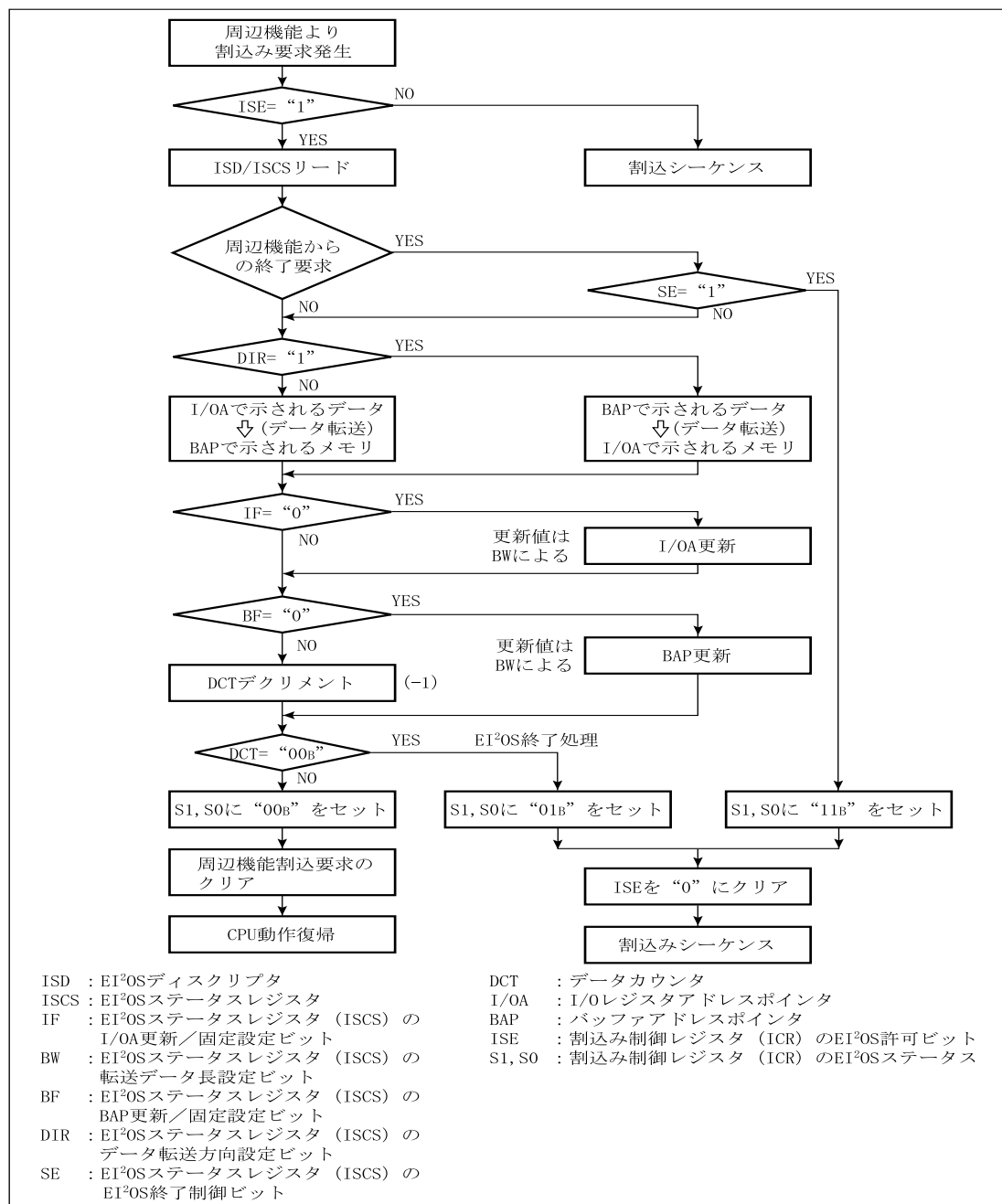


図 6.6-7 拡張インテリジェントI/Oサービス (EI²OS) の動作フロー

6.6.4 拡張インテリジェントI/Oサービス (EI²OS) の設定手順

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

拡張インテリジェントI/Oサービス (EI²OS) の設定手順

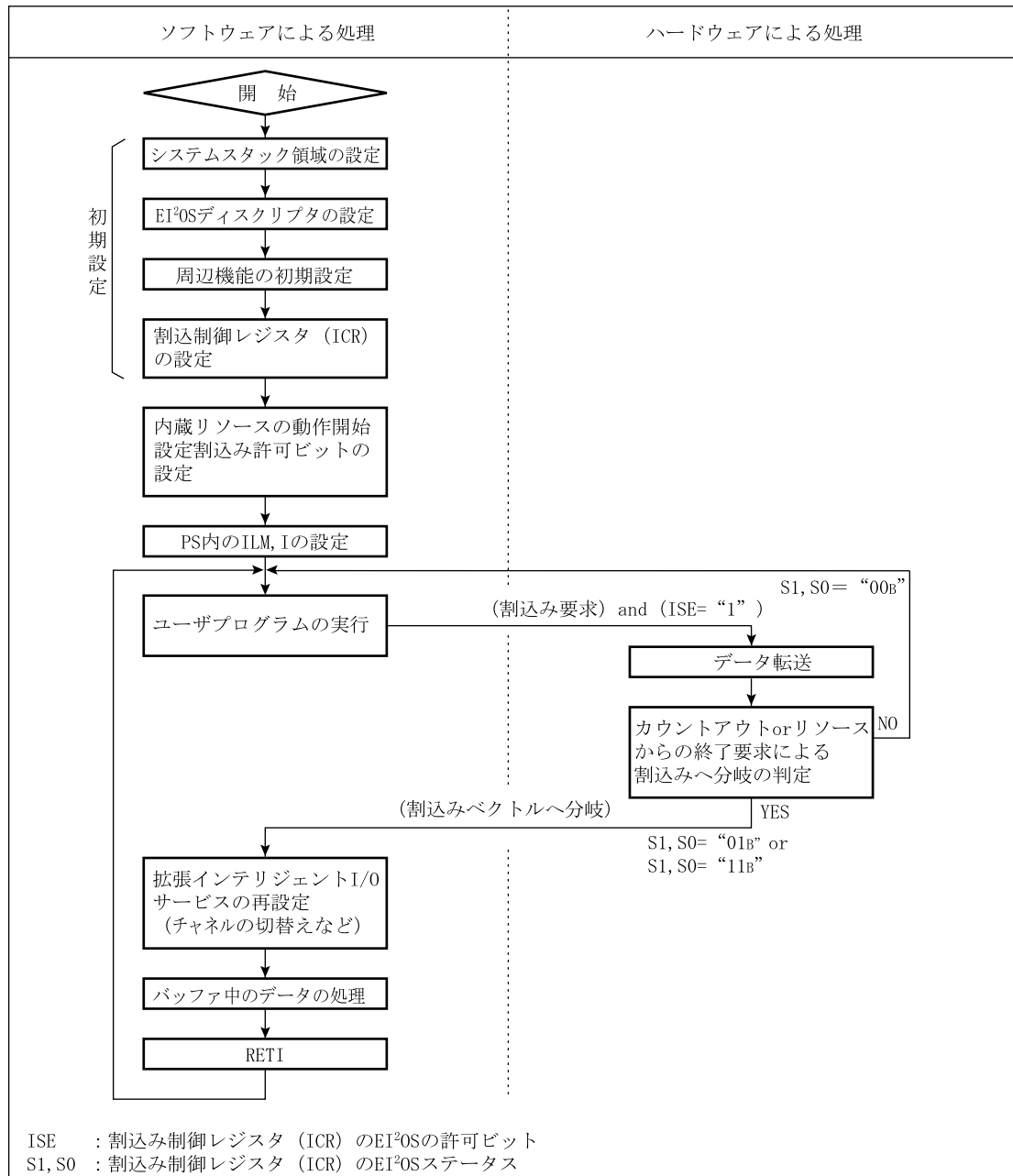


図 6.6-8 拡張インテリジェントI/Oサービス (EI²OS) 設定手順

6.6.5 拡張インテリジェントI/Oサービス (EI²OS) 処理時間

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

- ・EI²OSステータスレジスタ (ISCS) の設定
- ・I/Oレジスタアドレスポインタレジスタ (I/OA) の示すアドレス設定
- ・バッファアドレスポインタレジスタ (BAP) の示すアドレス設定
- ・外部アクセス時の外部データバス幅
- ・転送データのデータ長

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

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

データ転送を継続する場合

データ転送を継続する場合のEI²OS処理時間は、EI²OSステータスレジスタ (ISCS) の設定によって、表 6.6-2のようになります。

表 6.6-2 拡張インテリジェントI/Oサービス実行時間

EI ² OS終了制御ビット (SE) の設定		周辺からの終了要求により終了		周辺からの終了要求を無視	
I/OA更新/固定設定ビット (IF) の設定		固定	更新	固定	更新
BAPアドレス更新/固定 設定ビット (BF) の設定	固定	32	34	33	35
	更新	34	36	35	37

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

データ転送を継続する場合のEI²OS処理時間は、表 6.6-3に示すように、EI²OS実行条件によって補正が必要です。

表 6.6-3 EI²OS実行時間のデータ転送の補正值

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

B：バイトデータ転送

8：外部バス幅8ビット・ワード転送

偶：偶数アドレス・ワード転送

奇：奇数アドレス・ワード転送

データカウンタ (DCT) のカウント終了時 (最終回のデータ転送時)

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

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

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

表 6.6-4 割込みハンドリング時間の補正值 (Z)

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

周辺機能 (リソース) からの終了要求により終了した場合

周辺機能 (リソース) からの終了要求では、EI²OSによるデータ転送を途中で終了した場合 (ICR : S1, S0="11_B") は、データ転送は行わず、ハードウェア割込みを起動します。EI²OS処理時間は、下式で算出されます。式中のZは、割込みハンドリング時間の補正值です (表 6.6-4を参照)。

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

1マシンサイクルは、マシクロック () の1クロック周期に相当します。

6.7 例外処理

MB90560/565シリーズでは、定義されていない命令を実行した場合、例外処理を行います。例外処理は、割込みと同じものであり命令と命令の間で例外事項が発生した場合は、プログラム処理を中断して例外処理ルーチンへ分岐します。

例外処理は、予想外の動作を行った結果によって発生するものでありデバッグ時や未定義命令の実行、CPU暴走状態検出の起動に使用することができます。

例外処理

例外処理の動作

MB90560/565シリーズでは、命令マップで定義されていない命令を実行した場合、ソフトウェア割込み命令の例外処理ルーチンへ分岐します。

例外処理では、割込みルーチンへ分岐する前に、以下に示す処理を行います。

- システムスタックへ専用レジスタ（A,DPR,ADB,DTB,PCB,PC,PS）の内容を退避させます。
- コンディションコードレジスタ（CCR）のIフラグを"0"にクリアし、ハードウェア割込みをマスクします。
- コンディションコードレジスタ（CCR）のSフラグに"1"をセットし、システムスタックを有効にします。

システムスタックへ退避させたプログラムカウンタ（PC）の値は、未定義命令のプログラムアドレスを格納しています。2バイト以上の命令コードの場合は、未定義命令であることが識別できたコードを格納しているプログラムアドレスになります。例外処理ルーチン内で、例外要因の種類を判定する必要がある場合は、システムスタックへ退避させたPC値を使用してください。

例外処理からの復帰

例外処理からRET I命令で復帰させた場合、PCが未定義命令を指しているために、再度、例外処理ルーチンへ分岐します。ソフトウェアリセット、またはRST端子から"L"レベルを入力（外部リセット）してください。

6.8 割込み処理のスタック動作

割込みが受け付けられた場合，割込み処理に分岐する前に，専用レジスタの内容をシステムスタックに退避させます。割込み処理が終了したあとに，割込み復帰命令を実行することにより，システムスタックから専用レジスタへ退避させた値を復帰できます。

割込み処理開始時のスタック動作

割込みが受け付けられた場合は，CPUは，専用レジスタの内容を，以下に示す順番で，システムスタックに退避させます。

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

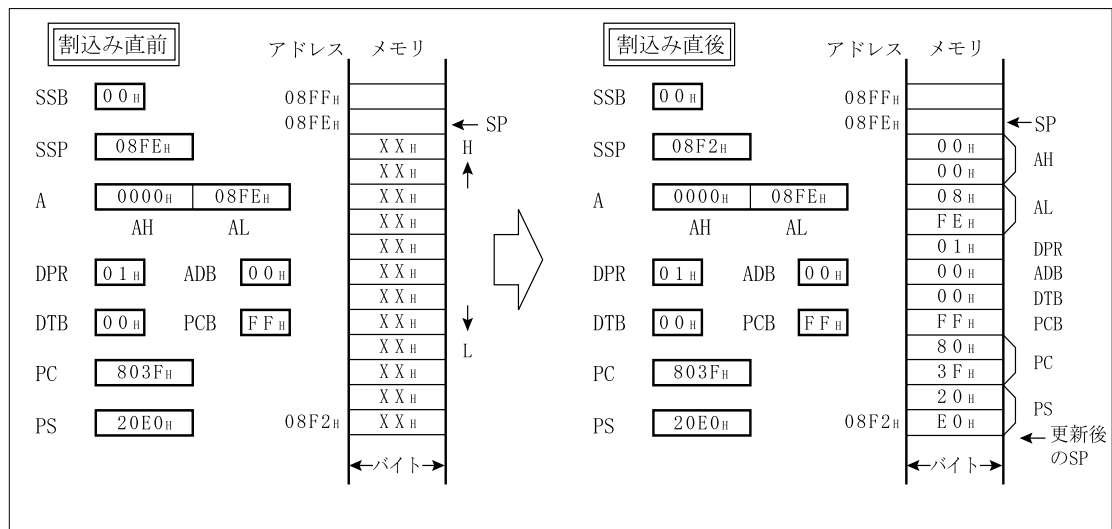


図 6.8-1 割込み処理開始時のスタック動作

割込み処理復帰時のスタック動作

割込み処理が終了してから，割込み復帰命令 (RETI) を実行した場合は，割込み処理開始時と反対の順番で専用レジスタ (PS, PC, PCB, DTB, ADB, DPR, A) の値がスタックから復帰し，専用レジスタは割込み処理へ分岐する前の状態に戻ります。

スタック領域

スタック領域の確保

スタック領域は、割り込み処理、サブルーチンコール命令（CALL）、ベクタコール命令（CALLV）を実行する場合に、プログラムカウンタ（PC）の退避/復帰や、PUSHW、POPW命令によるレジスタ値の退避/復帰にも使用します。スタック領域は、データ領域とともにRAM上に設定します。

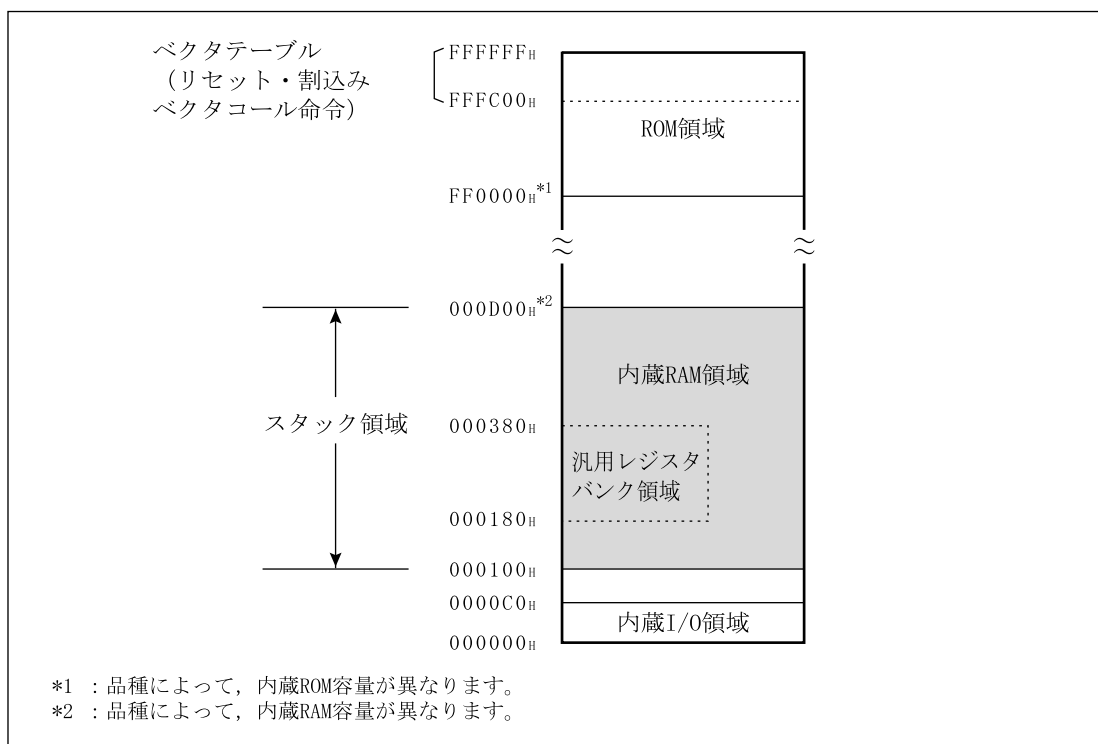


図 6.8-2 スタック領域

< 注意事項 >

- ・スタックポインタ (SSP, USP) にアドレスを設定する場合、偶数アドレスを設定してください。奇数アドレスを設定した場合は、スタックに退避 / 復帰処理時間が1サイクル多くかかります。
- ・システムスタック領域、ユーザスタック領域、およびデータ領域は、重なり合わないよう配置してください。

システムスタックとユーザスタック

割込み処理では、システムスタック領域を使用します。割込み出力時にユーザスタック領域を使用している場合、システムスタックに切り替わります。スタック空間を分ける必要がない場合は、システムスタックを使用してください。

6.9 割込み処理のプログラム例

割込み処理のプログラム例を示します。

割込み処理のプログラム例

処理仕様

外部割込み0 (INT0) を利用した割込みプログラムの一例です。

コーディング例

```

DDR1 EQU 000011H ; ポート1方向レジスタ
ENIR EQU 000030H ; 割込み/DTP許可レジスタ
EIRR EQU 000031H ; 割込み/DTP要因レジスタ
ELVR EQU 000032H ; 要求レベル設定レジスタ
ICR00 EQU 0000B0H ; 割込み制御レジスタ00
STACK SSEG ; スタック
      RW 100
STACK_T RW 1
STACK ENDS
;-----メインプログラム-----
CODE CSEG ;
START:
      MOV RP, #0 ; 汎用レジスタは先頭バンクを使用
      MOV ILM, #07H ; PS内ILMをレベル7に設定
      MOV A, #!STACK_T ; システムスタックの設定
      MOV SSB, A
      MOVW A, #STACK_T ; スタックポインタの設定, この場合は
      MOVW SP, A ; Sフラグ=1のためSSPにセットされる
      MOV DDR1, #00000000B ; P10/INT0端子を入力に設定
      OR CCR, #040H ; PS内CCRのIフラグをセットして割込み許可
      MOV I:ICR00, #00H ; 割り込みレベル0 (最強) とする
      MOV I:ELVR, #00000001B ; INT0をHレベル要求とする
      MOV I:EIRR, #00H ; INT0の割込み要因をクリア
      MOV I:ENIR, #01H ; INT0の入力許可
      :
LOOP: NOP ; ダミーループ
      NOP
      NOP
      NOP
      BRA LOOP ; 無条件ジャンプ
;-----割込みプログラム-----
ED_INT1:
      MOV I:EIRR, #00H ; INT0 の新規受付禁止
      NOP
      NOP
      NOP
      NOP
      NOP
      NOP
      RETI ; 割込みから復帰
CODE ENDS

```



```

;-----ベクタ設定-----
VECT    CSEG      ABS=OFFH
        ORG        OFF98H          ; 割り込み#25 (19H) にベクタを設定
        DSL        ED_INT1
        ORG        OFFDCH          ; リセットベクタ設定
        DSL        START
        DB         00H             ; シングルチップモードに設定
VECT    ENDS
        END          START

```

拡張インテリジェントI/Oサービス (EI²OS) のプログラム例

処理仕様

- 1) INT0端子に入力される, 信号の"H"レベルを検出して拡張インテリジェントI/Oサービス (EI²OS) を起動
- 2) INT0端子に"H"レベルが入力されると, EI²OSが起動されポート0のデータをメモリの"3000H"番地に転送
- 3) 転送データバイト数は100バイトで, 100バイト転送後, EI²OS転送終了による割り込みを出力

コーディング例

```

DDR1    EQU        000011H          ; ポート1方向レジスタ
ENIR    EQU        000030H          ; 割り込み/DTP許可レジスタ
EIRR    EQU        000031H          ; 割り込み/DTP要因レジスタ
ELVR    EQU        000032H          ; 要求レベル設定レジスタ
ICR00   EQU        0000B0H          ; 割り込み制御レジスタ00
BAPL    EQU        000100H          ; バッファアドレスポインタ下位
BAPM    EQU        000101H          ; バッファアドレスポインタ中位
BAPH    EQU        000102H          ; バッファアドレスポインタ上位
ISCS    EQU        000103H          ; EI2OSステータス
I/OAL   EQU        000104H          ; I/Oアドレスポインタ下位
I/OAH   EQU        000105H          ; I/Oアドレスポインタ上位
DCTL    EQU        000106H          ; データカウンタ下位
DCTH    EQU        000107H          ; データカウンタ上位
ERO     EQU        EIRR:0           ; 外部割り込み要求フラグビットの定義
STACK   SSEG
        RW         100             ; スタック
STACK_T RW         1
STACK   ENDS

```

```

;-----メインプログラム-----
CODE    CSEG
START:
        AND        CCR, #0BFH      ; PS内CCRのIフラグをクリアして割り込み禁止
        MOV        RP, #00         ; レジスタバンクポインタを設定
        MOV        A, #!STACK_T    ; システムスタックの設定
        MOV        SSB, A
        MOVW       A, #STACK_T      ; スタックポインタの設定, この場合は
        MOVW       SP, A            ; Sフラグ=1のためSSPにセットされる
        MOV        I:DDR1, #00000000B ; P10/INT0端子を入力に設定
        MOV        BAPL, #00H       ; バッファアドレスをセット (003000H)
        MOV        BAPM, #30H
        MOV        BAPH, #00H
        MOV        ISCS, #00010001B ; I/Oアドレス更新なし, バイト転送,
        ; バッファアドレス更新あり

```

```

; I/O バッファへ転送，周辺機能（リソース）
; による終了あり
MOV      I/OAL,#00H      ; 転送元アドレスをセット（ポート0：000000H）
MOV      I/OAH,#00H
MOV      DCTL,#064H      ; 転送バイト数をセット（100バイト）
MOV      DCTH,#00H
MOV      I:ICR00,#00001000B ; EI2OSチャネル0，EI2OS許可，
                           ; 割り込みレベル0（最強）
MOV      I:ELVR,#00000001B ; INT0を"H"レベル要求とする
MOV      I:EIRR,#00H      ; INT0の割り込み要因クリア
MOV      I:ENIR,#01H      ; INT0の割り込み許可
MOV      ILM,#07H        ; PS内ILMをレベル7に設定
OR       CCR,#040H        ; PS内CCRのIフラグをセットして割り込み許可
;
LOOP:    BRA      LOOP      ; 無限ループ
;-----割り込みプログラム-----
WARI     CLRB      ERO      ; 割り込み/DTP要求フラグのクリア
;
; ユーザ処理
;
; EI2OSの終了要因をチェック，
; バッファ中のデータの処理，EI2OSの再設定
; など
RETI
CODE     ENDS
;-----ベクタ設定-----
VECT     CSEG      ABS=OFFH
ORG      OFFD0H      ; 割り込み #11（0BH）にベクタを設定
DSL      WARI
ORG      OFFDCH      ; リセットベクタ設定
DSL      START
DB       00H          ; シングルチップモードに設定
VECT     ENDS
END      START

```


第7章 モード設定

この章では、MB90560/565シリーズの動作モードについて説明します。

- 7.1 モード設定
- 7.2 モード端子 (MD2 ~ MD0)
- 7.3 モードデータレジスタ

7.1 モード設定

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

モード設定

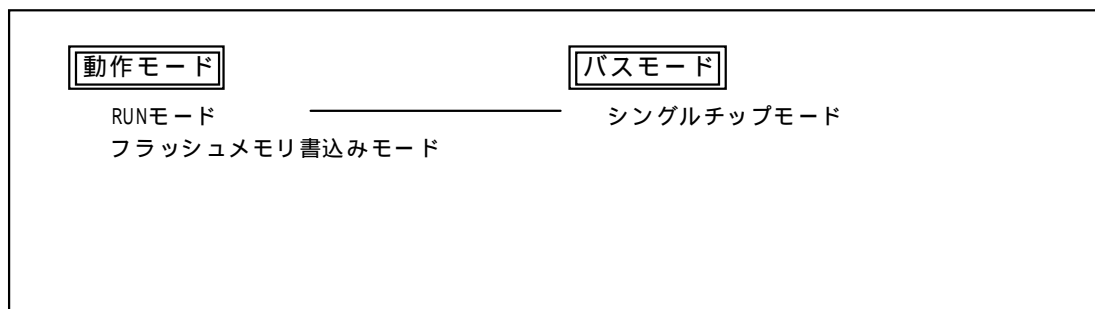


図 7.1-1 モードの分類

動作モード

動作モードは、モード端子(MD2～MD0)とモードデータレジスタのバスモード設定ビット(M1,M0)で設定します。設定した動作モードで、マイコンは起動します。

<注意事項>

MB90560/565シリーズでは、シングルチップモードを設定してください。

シングルチップモードを設定する場合は、MD2～MD0端子を"001_B"に、モードデータレジスタのバスモード設定ビット(M1,M0)は"10_B"に、設定してください。

バスモード

バスモードは、リセットベクタを讀出すメモリが外部にあるか内部にあるかにより異なります。モード設定端子(MD2～MD0)とモードデータレジスタのバスモード設定ビット(M1,M0)を設定することでバスモードを設定できます。モード設定端子(MD2～MD0)で、リセットベクタ、およびモードデータを讀出す場合のバスモードを設定します。また、モードデータレジスタのバスモード設定ビット(M1,M0)で、バスモードを設定します。

詳細は「7.2 モード端子(MD2～MD0)」と「7.3 モードデータレジスタ」を参照してください。

RUNモード

RUNモードは、CPUが動作しているモードのことをいいます。RUNモードには、メインクロックで動作するメインクロックモード、PLLクロックで動作するPLLクロックモード、低消費電力モードがあります。詳細は、「第5章 低消費電力モード」を参照してください。

7.2 モード端子 (MD2 ~ MD0)

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

モード端子 (MD2 ~ MD0)

モード端子で、リセットベクタの読出しを外部メモリとするか、内部メモリとするかを設定します。リセットベクタを外部メモリに設定した場合は、外部データバス幅をモードデータレジスタで設定します。

フラッシュメモリ内蔵品の場合は、内蔵フラッシュメモリにプログラムを書込むために、フラッシュメモリ書込みモードの設定をモード端子で行います。

表 7.2-1 モード端子の設定

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

MD2 ~ MD0 : "0" = Vss, "1" = Vccとしてください。

* : フラッシュメモリのシリアル書込みは、モード端子の設定を行うだけでは書込みできません。他の箇所も設定する必要があります。詳細は、「第20章 シリアル書き込み接続例」を参照してください。

7.3 モードデータレジスタ

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

モードデータレジスタ

リセットシーケンス実行中に，"FFFFDFH"番地のモードデータをモードデータレジスタに取り込みます。モードデータレジスタの内容は，リセットシーケンス実行中に変更できます。命令では変更できません。モードデータの設定は，リセットシーケンス後に有効となります。

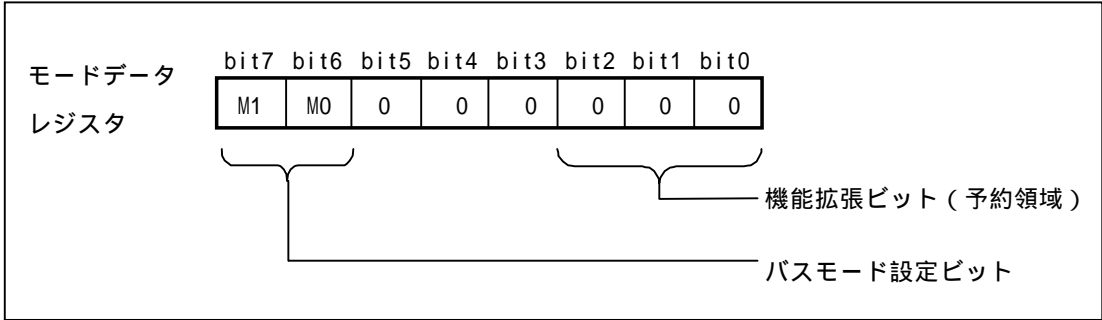


図 7.3-1 モードデータレジスタの構成

バスモード設定ビット

リセットシーケンス終了後のバスモードを設定するビットです。

表 7.3-1 バスモードのビット設定と機能

M1	M0	機 能	備 考
0	0	シングルチップモード	本マニュアルで説明
0	1	(設定禁止)	-
1	0		
1	1		

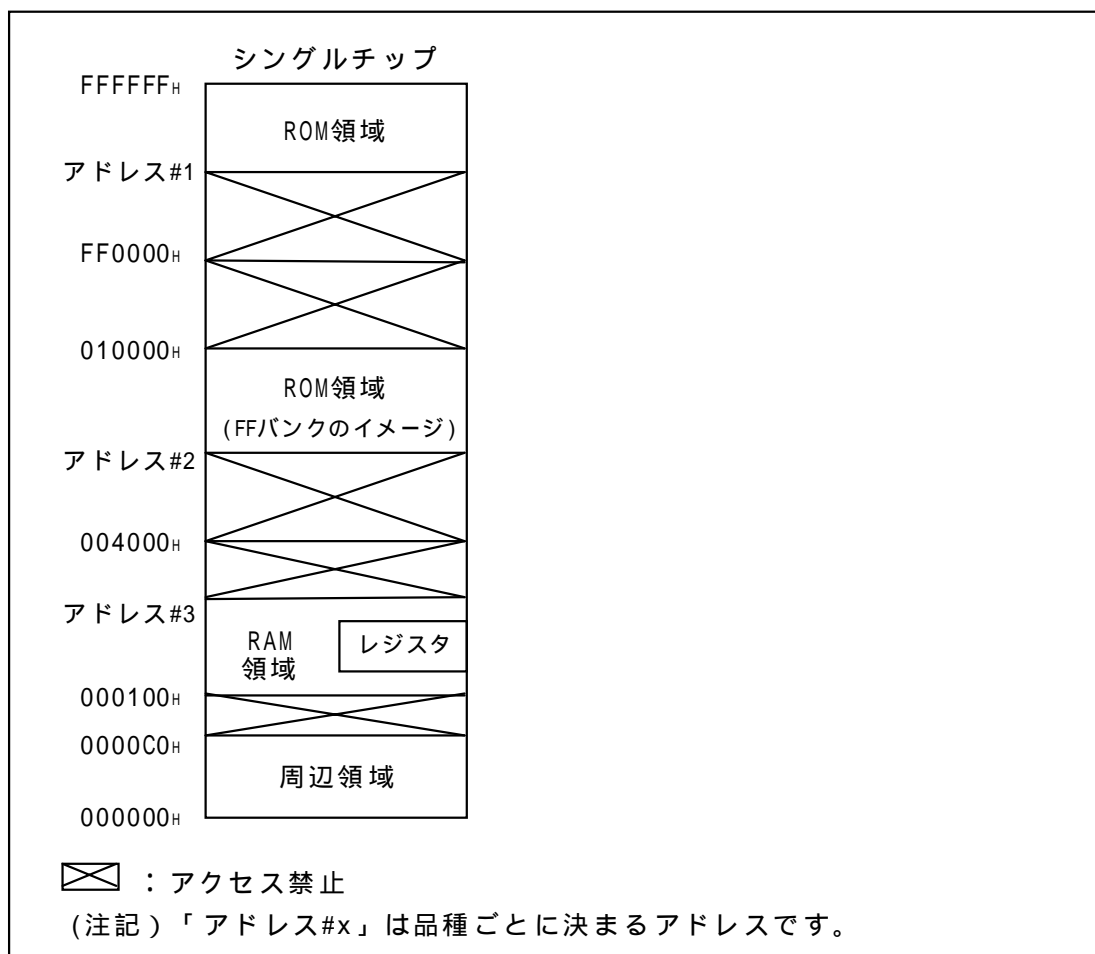


図 7.3-2 シングルチップモードのメモリマップ

モード端子とモードデータの関係

表 7.3-2 モード端子とモードデータの関係

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

第8章 入出力ポート

この章では、MB90560/565シリーズの入出力ポートの機能と動作について説明します。

- 8.1 入出力ポートの概要
- 8.2 入出力ポートのレジスタ説明
- 8.3 ポート0
- 8.4 ポート1
- 8.5 ポート2
- 8.6 ポート3
- 8.7 ポート4
- 8.8 ポート5
- 8.9 ポート6
- 8.10 入出力ポートのプログラム例

8.1 入出力ポートの概要

入出力ポート（パラレルI/Oポート）は，最大51本あり，リソース入出力端子（周辺機能の入出力端子）と兼用しています。

入出力ポート機能

入出力ポートには，ポート方向レジスタ（DDR）とポートデータレジスタ（PDR）があります。ポート方向レジスタ（DDR）では，ポート端子の入出力をビット単位で設定します。ポートデータレジスタ（PDR）では，ポート端子への出力データを設定します。ポート方向レジスタ（DDR）で入出力ポート端子を入力に設定した場合は，ポートデータレジスタ（PDR）をリードすることにより，ポート端子のレベル値を読むことができます。ポート方向レジスタ（DDR）で入出力ポート端子を出力に設定した場合は，ポートデータレジスタ（PDR）の値をポート端子に出力します。以下に各入出力ポートの機能と兼用するリソースを示します。

- ポート0 : 入出力ポート
- ポート1 : 入出力ポート / リソース（外部割込み入力端子）兼用
- ポート2 : 入出力ポート /
リソース（16ビットリロードタイマ，
インプットキャプチャ）兼用
- ポート3 : 入出力ポート /
リソース（アウトプットコンペア，UART0）兼用
- ポート4 : 入出力ポート /
リソース（UART0，8ビット/16ビットPPGタイマ）兼用
- ポート5 : 入出力ポート / リソース（アナログ入力端子）兼用
- ポート6 : 入出力ポート / リソース（UART1）兼用

表 8.1-1 各ポートの機能一覧

入出力 ポート名	端子名	入力形式	出力形式	機 能								
ポート0	P00 ~ P07	CMOS (ヒステリシス)	CMOS プルアップ 抵抗設定可	入出力ポート	P07	P06	P05	P04	P03	P02	P01	P00
ポート1	P10/INT0 ~ P17/FRCK			入出力ポート	P17	P16	P15	P14	P13	P12	P11	P10
				リソース	FRCK	INT6	INT5	INT4	INT3	INT2	INT1	INT0
ポート2	P20/TIN0 ~ P27/IN3		CMOS	入出力ポート	P27	P26	P25	P24	P23	P22	P21	P20
ポート3	P30/RT00 ~ P37/SOT0			リソース	IN3	IN2	IN1	IN0	T01	TIN1	T00	TIN0
				入出力ポート	P37	P36	P35	P34	P33	P32	P31	P30
ポート4	P40/SCK0 ~ P46/PPG5			リソース	SOT0	SIN0	RT05	RT04	RT03	RT02	RT01	RT00
				入出力ポート	-	P46	P45	P44	P43	P42	P41	P40
ポート5	P50/AN0 ~ P57/AN7			リソース	-	PPG5	PPG4	PPG3	PPG2	PPG1	PPG0	SCK0
				入出力ポート	P57	P56	P55	P54	P53	P52	P51	P50
ポート6	P60/SIN1 ~ P63/INT7	アナログ・CMOS (ヒステリシス)		アナログ入力	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
		CMOS (ヒステリシス)		入出力ポート	-	-	-	-	P63	P62	P61	P60
				リソース	-	-	-	-	INT7 DTT1	SCK1	SOT1	SIN1

< 注意事項 >

ポート5は、アナログ入力端子と兼用になっており、入出力ポートとして使用する場合は、ポート5方向レジスタ（DDR5）とポート5データレジスタ（PDR5）の他に、アナログ入力許可レジスタ（ADER）に"00_H"を設定してください。リセットでアナログ入力許可レジスタ（ADER）は"FF_H"に初期化されます。また、端子ごとにアナログ入力と入出力ポートを設定する場合は、アナログ入力許可レジスタ（ADER）の対応するビットに"0"を設定してください。

8.2 入出力ポートのレジスタ説明

入出力ポートの設定に関連するレジスタの一覧を示します。

入出力ポートのレジスタ一覧

表 8.2-1 ポート0～6のレジスタ一覧

レジスタ名	リードライト	アドレス	初期値
ポート0データレジスタ (PDR0)	R/W	000000H	XXXXXXXXB
ポート1データレジスタ (PDR1)	R/W	000001H	XXXXXXXXB
ポート2データレジスタ (PDR2)	R/W	000002H	XXXXXXXXB
ポート3データレジスタ (PDR3)	R/W	000003H	XXXXXXXXB
ポート4データレジスタ (PDR4)	R/W	000004H	XXXXXXXXB
ポート5データレジスタ (PDR5)	R/W	000005H	XXXXXXXXB
ポート6データレジスタ (PDR6)	R/W	000006H	XXXXXXXXB
ポート0方向レジスタ (DDR0)	R/W	000010H	00000000B
ポート1方向レジスタ (DDR1)	R/W	000011H	00000000B
ポート2方向レジスタ (DDR2)	R/W	000012H	00000000B
ポート3方向レジスタ (DDR3)	R/W	000013H	00000000B
ポート4方向レジスタ (DDR4)	R/W	000014H	X0000000B
ポート5方向レジスタ (DDR5)	R/W	000015H	00000000B
ポート6方向レジスタ (DDR6)	R/W	000016H	XXX00000B
アナログ入力許可レジスタ (ADER)	R/W	000017H	11111111B
ポート0プルアップ抵抗設定レジスタ (RDR0)	R/W	00008CH	00000000B
ポート1プルアップ抵抗設定レジスタ (RDR1)	R/W	00008DH	00000000B

R/W : リード・ライト可能

X : 不定

8.3 ポート0

ポート0は、入出力ポートです。ここでは、ポート0の構成、端子のブロックダイヤグラム、レジスタについて説明します。

ポート0の構成

ポート0は、次の要素から構成されます。

- 入出力ポート端子
- ポート0データレジスタ (PDR0)
- ポート0方向レジスタ (DDR0)
- ポート0プルアップ抵抗設定レジスタ (RDR0)

ポート0の端子

表 8.3-1 ポート0の端子設定

入出力 ポート名	端子名	入出力ポート機能	入出力形式	
			入力	出力
ポート0	P00	P00	CMOS (ヒステリシス)	CMOS
	P01	P01		
	P02	P02		
	P03	P03		
	P04	P04		
	P05	P05		
	P06	P06		
	P07	P07		

ポート0端子のブロックダイヤグラム

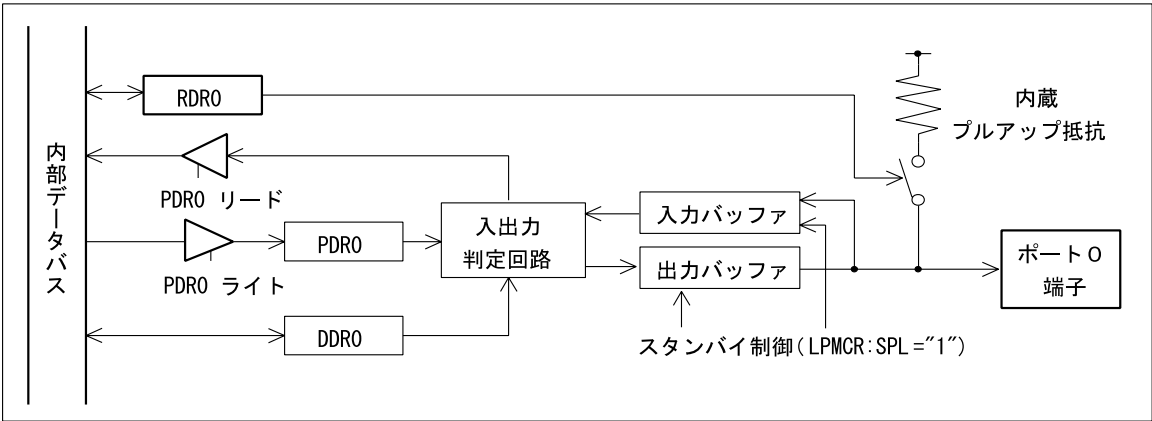


図 8.3-1 ポート0端子のブロックダイヤグラム

ポート0のレジスタ

ポート0のレジスタには、ポート0データレジスタ (PDR0)、ポート0方向レジスタ (DDR0) および、ポート0プルアップ抵抗設定レジスタ (RDR0) があります。各レジスタを構成するビットは、ポート0の端子に1対1で対応しています。

表 8.3-2 ポート0のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
	PDR0,DDR0,RDR0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート0	対応する端子	P07	P06	P05	P04	P03	P02	P01	P00

8.3.1 ポート0のレジスタ (PDR0, DDR0, RDR0)

ポート0のレジスタについて説明します。

ポート0のレジスタの機能

ポート0データレジスタ (PDR0)

ポート0の端子ごとの出力値を設定します。

ポート0方向レジスタ (DDR0)

ポート0の端子ごとの入出力方向が設定できます。端子に対応するビットに"1"を設定した場合は、端子は出力ポートになり、"0"を設定した場合は、端子は入力ポートになります。

ポート0プルアップ抵抗設定レジスタ (RDR0)

端子に対応するビットごとに端子へのプルアップ抵抗を接続・切断するための設定ができます。ポート0の端子に対応するビットに"1"を設定した場合は、プルアップ抵抗が接続状態になり、"0"を設定した場合は、プルアップ抵抗が切断状態になります。

表 8.3-3 ポート0のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート0データ レジスタ (PDR0)	0	端子状態が "L" レベル です。	PDR0レジスタ のビットに "0" が設定さ れています。	PDR0レジスタ のビットに "0"を設定し ます。	端子より"L" レベルを出 力します。	000000 _H	XXXXXXX _B
	1	端子状態が "H" レベル です。	PDR0レジスタ のビットに "1" が設定さ れています。	PDR0レジスタ のビットに "1"を設定し ます。	端子より"H" レベルを出 力します。		
ポート0方向 レジスタ (DDR0)	0	DDR0レジスタのビットに"0" が設定されています。		入力ポートになります。		000010 _H	00000000 _B
	1	DDR0レジスタのビットに"1" が設定されています。		出力ポートになります。			
ポート0 プルアップ抵抗 設定レジスタ (RDR0)	0	RDR0レジスタのビットに"0" が設定されています。		プルアップ抵 抗が切断さ れ、ハイイン ピーダンス状 態になりま す。	プルアップ 抵抗は切断 されます。	00008C _H	00000000 _B
	1	RDR0レジスタのビットに"1" が設定されています。		プルアップ抵 抗が接続さ れ、"H"レベ ル保持しま す。	プルアップ 抵抗は切断 されます。		

X : 不定

8.3.2 ポート0の動作説明

ポート0の動作を説明します。

ポート0の動作

ポート0方向レジスタ（DDR0）で、出力ポートに設定した場合

- ポート0データレジスタ（PDR0）に設定した値が、ポート0端子へ出力されます。
- ポート0データレジスタ（PDR0）を読出した場合は、ポート0データレジスタ（PDR0）に設定されている値が読出されます。

ポート0方向レジスタ（DDR0）で、入力ポートに設定した場合

- ポート0端子はハイインピーダンスになります。ただし、ポート0プルアップ抵抗レジスタ（RDR0）のビットに"1"が設定されている場合は、プルアップ抵抗が接続され、"H"レベルを保持します。
- ポート0データレジスタ（PDR0）に値を設定した場合は、ポート0データレジスタ（PDR0）に設定した値が保持されますが、端子には出力されません。
- ポート0データレジスタ（PDR0）を読出した場合は、端子の入力レベル値（"L"の場合は"0"、"H"の場合は"1"）が読出されます。

< 注意事項 >

リードモディファイライト系命令（ビットセット命令など）を使用して、ポート0データレジスタ（PDR0）をアクセスする場合は、ポート0方向レジスタ（DDR0）で出力に設定されているビットは影響を受けませんが、ポート0方向レジスタ（DDR0）で入力に設定されているビットは、端子の入力レベル値がポート0データレジスタ（PDR0）に書込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート0データレジスタ（PDR0）に出力値を設定した後に、ポート0方向レジスタ（DDR0）を出力に設定してください。

リセット時の動作

- CPUがリセットされた場合は、ポート0方向レジスタ（DDR0）およびポート0データレジスタ（RDR0）の値が"00H"に初期化され、ポート0端子はハイインピーダンスになります。
- ポート0データレジスタ（PDR0）は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート0データレジスタ（PDR0）に出力値を設定した後に、ポート0方向レジスタ（DDR0）を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）に"1"が設定されていると、端子はポート0方向レジスタ（DDR0）の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

< 注意事項 >

プルアップ抵抗が接続されている場合は、低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）に"1"を設定した場合でも、プルアップ抵抗は切断されず、端子レベルは"H"レベルを保持します。

表 8.3-4 ポート0の端子状態

端子名	通常動作	スリープモード	ストップモード, タイムベース タイマモード (SPL="0")	ストップモード, タイムベース タイマモード (SPL="1", RDR="0")	ストップモード, タイムベース タイマモード (SPL="1", RDR="1")
P00 ~ P07	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z	入力遮断 /Hレベル保持

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

8.4 ポート1

ポート1は、リソース入力兼用の入出力ポートです。各端子は、リソースと入出力ポートをビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート1の構成、端子のブロックダイヤグラム、レジスタについて説明します。

ポート1の構成

ポート1は、次の要素から構成されます。

- 入出力ポート端子 / リソース入力端子 (P10/ INT0 ~ P17/ FRCK)
- ポート1データレジスタ (PDR1)
- ポート1方向レジスタ (DDR1)
- ポート1プルアップ抵抗設定レジスタ (RDR1)

ポート1の端子

ポート1の入出力端子は、リソース入力端子と兼用になっていますので、リソース入力端子として使用する場合は、入出力ポートとしては使用できません。

表 8.4-1 ポート1の端子

入出力 ポート名	端子名	入出力ポート機能	リソース機能		入出力形式	
					入力	出力
ポート1	P10/ INT0	P10	INT0	外部割込み入力	CMOS (ヒステリシス)	CMOS
	P11/ INT1	P11	INT1			
	P12/ INT2	P12	INT2			
	P13/ INT3	P13	INT3			
	P14/ INT4	P14	INT4			
	P15/ INT5	P15	INT5			
	P16/ INT6	P16	INT6			
	P17/ FRCK	P17	FRCK	フリーランタイム 外部クロック入力		

ポート1端子のブロックダイヤグラム

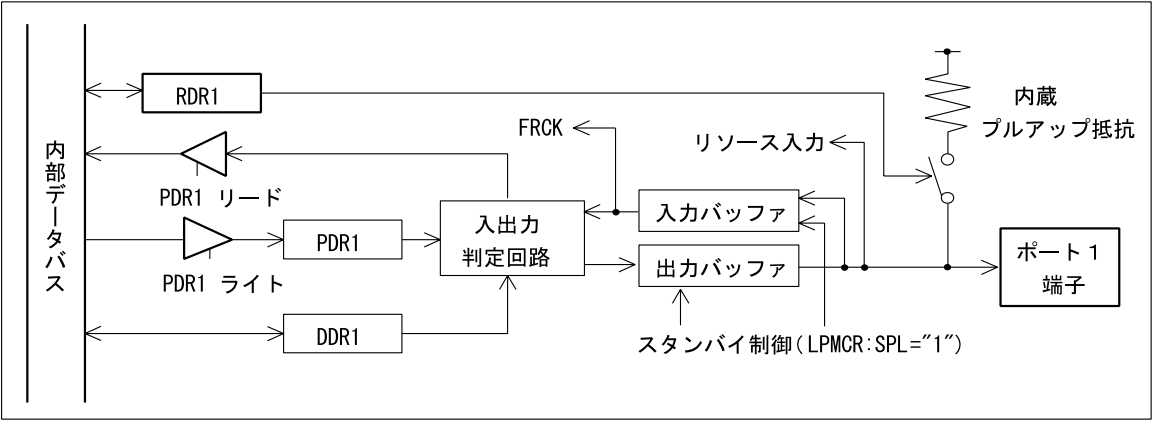


図 8.4-1 ポート1端子のブロックダイヤグラム

ポート1のレジスタ

ポート1のレジスタには、ポート1データレジスタ (PDR1)、ポート1方向レジスタ (DDR1) および、ポート1プルアップ抵抗設定レジスタ (RDR1) があります。各レジスタを構成するビットは、ポート1の端子に1対1で対応しています。

表 8.4-2 ポート1のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
ポート1	PDR1,DDR1,RDR1	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	対応する端子	P17	P16	P15	P14	P13	P12	P11	P10

8.4.1 ポート1のレジスタ (PDR1, DDR1, RDR1)

ポート1のレジスタについて説明します。

ポート1のレジスタの機能

ポート1データレジスタ (PDR1)

ポート1の端子ごとの出力値を設定します。

ポート1方向レジスタ (DDR1)

ポート1の端子ごとの入出力方向を設定できます。端子に対応するビットに"1"を設定した場合は、端子は出力ポートになり、"0"を設定した場合は、端子は入力ポートになります。

ポート1プルアップ抵抗設定レジスタ (RDR1)

端子に対応するビットごとに端子へのプルアップ抵抗を接続・切断するための設定ができます。ポート1の端子に対応するビットに"1"を設定した場合は、プルアップ抵抗が接続状態になり、"0"を設定した場合は、プルアップ抵抗が切断状態になります。

表 8.4-3 ポート1のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート1データ レジスタ (PDR1)	0	端子状態が "L" レベル です。	PDR1レジスタ のビットに "0" が設定さ れています。	PDR1レジスタ のビットに "0" を設定し ます。	端子より "L" レベルを出 力します。	000001 _H	XXXXXXXX _B
	1	端子状態が "H" レベル です。	PDR1レジスタ のビットに "1" が設定さ れています。	PDR1レジスタ のビットに "1" を設定し ます。	端子より "H" レベルを出 力します。		
ポート1方向 レジスタ (DDR1)	0	DDR1レジスタのビットに "0" が設定されています。		入力ポートになります。		000011 _H	00000000 _B
	1	DDR1レジスタのビットに "1" が設定されています。		出力ポートになります。			
ポート1 プルアップ抵抗 設定レジスタ (RDR1)	0	RDR1レジスタのビットに "0" が設定されています。		プルアップ抵 抗が切断さ れ、ハイイン ピーダンス状 態になりま す。	プルアップ 抵抗は切断 されます。	00008D _H	00000000 _B
	1	RDR1レジスタのビットに "1" が設定されています。		プルアップ抵 抗が接続さ れ、"H" レベ ル保持しま す。	プルアップ 抵抗は切断 されます。		

X : 不定

8.4.2 ポート1の動作説明

ポート1の動作を説明します。

ポート1の動作

ポート1方向レジスタ (DDR1) で、出力ポートに設定した場合

- ポート1データレジスタ (PDR1) に設定した値が、ポート1端子へ出力されます。
- ポート1データレジスタ (PDR1) を読出した場合は、ポート1データレジスタ (PDR1) に設定されている値が読出されます。

ポート1方向レジスタ (DDR1) で、入力ポートに設定した場合

- ポート1端子はハイインピーダンスになります。ただし、プルアップ抵抗設定レジスタ (RDR1) のビットに"1"が設定されている場合は、プルアップ抵抗が接続され、"H"レベルを保持します。
- ポート1データレジスタ (PDR1) に値を設定した場合は、ポート1データレジスタ (PDR1) に設定した値が保持されますが、端子には出力されません。
- ポート1データレジスタ (PDR1) を読出した場合は、端子の入力レベル値 ("L"の場合は"0", "H"の場合は"1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート1データレジスタ (PDR1) をアクセスする場合は、ポート1方向レジスタ (DDR1) で出力に設定されているビットは影響を受けませんが、ポート1方向レジスタ (DDR1) で入力に設定されているビットは、端子の入力レベル値がポート1データレジスタ (PDR1) に書込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート1データレジスタ (PDR1) に出力値を設定した後に、ポート1方向レジスタ (DDR1) を出力に設定してください。

リセット時の動作

- CPUがリセットされた場合は、ポート1方向レジスタ (DDR1) およびプルアップ抵抗設定レジスタ (RDR1) の値が"00H"に初期化され、ポート1端子はハイインピーダンスになります。
- ポート1データレジスタ (PDR1) は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート1データレジスタ (PDR1) に出力値を設定した後に、ポート1方向レジスタ (DDR1) を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL) に"1"が設定されていると、端子はポート1方向レジスタ (DDR1) の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力はバッファも強制的に遮断されます。

< 注意事項 >

プルアップ抵抗が接続されている場合は、低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）に"1"を設定した場合でも、プルアップ抵抗は切断されず、端子レベルは"H"レベルを保持します。

表 8.4-4 ポート1の端子状態

端子名	通常動作	スリープモード	ストップモード, タイムベースタイマモード (SPL="0")	ストップモード, タイムベースタイマモード (SPL="1", RDR="0")	ストップモード, タイムベースタイマモード (SPL="1", RDR="1")
P10 ~ P17	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z	入力遮断 /Hレベル保持
INT0 ~ INT6	外部割込み入力 0~6	外部割込み入力 0~6			
FRCK	フリーランタイム 外部クロック入力	フリーランタイム 外部クロック入力			

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

Hi-z : ハイインピーダンス

(注) リソース機能を使用している場合は、INT0~INT6,FRCKとして機能します。

8.5 ポート2

ポート2は、リソース入出力兼用の入出力ポートです。各端子は、リソースと入出力ポートをビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート2の構成、端子のブロックダイアグラム、レジスタを示します。

ポート2の構成

ポート2は、次の要素から構成されます。

- 入出力ポート端子 / リソース入出力端子 (P20/TIN0 ~ P27/IN3)
- ポート2データレジスタ (PDR2)
- ポート2方向レジスタ (DDR2)

ポート2の端子

ポート2の入出力端子は、リソース入出力端子と兼用になっていますので、リソース入出力端子として使用する場合は、入出力ポートとしては使用できません。

表 8.5-1 ポート2の端子

入出力 ポート名	端子名	入出力ポート 機能	リソース機能		入出力形式	
					入力	出力
ポート2	P20/TIN0	P20	TIN0	リロードタイム0の 外部クロック入力	CMOS (ヒステリシス)	CMOS
	P21/T00	P21	T00	リロードタイム0の イベント出力		
	P22/TIN1	P22	TIN1	リロードタイム1の 外部クロック入力		
	P23/T01	P23	T01	リロードタイム1の イベント出力		
	P24/IN0	P24	IN0	インプットキャプチャ0 のトリガ入力		
	P25/IN1	P25	IN1	インプットキャプチャ1 のトリガ入力		
	P26/IN2	P26	IN2	インプットキャプチャ2 のトリガ入力		
	P27/IN3	P27	IN3	インプットキャプチャ3 のトリガ入力		

ポート2端子のブロックダイアグラム

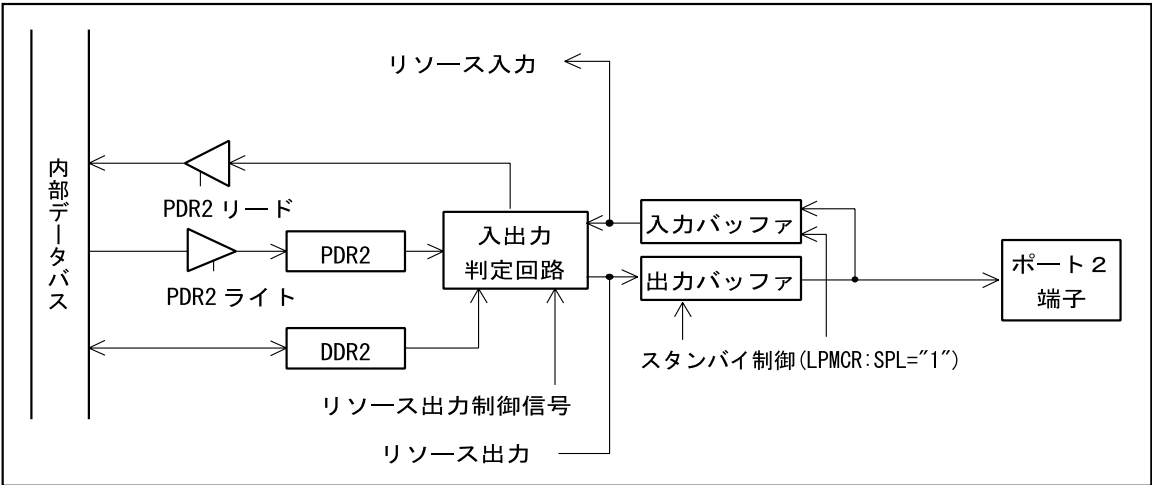


図 8.5-1 ポート2端子のブロックダイアグラム

ポート2のレジスタ

ポート2のレジスタには、ポート2データレジスタ（PDR2）および、ポート2方向レジスタ（DDR2）があります。各レジスタを構成するビットは、ポート2の端子に1対1で対応しています。

表 8.5-2 ポート2のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
	PDR2,DDR2	bit7	bit6	bit5	Bit4	bit3	bit2	bit1	bit0
ポート2	対応する端子	P27	P26	P25	P24	P23	P22	P21	P20

8.5.1 ポート2のレジスタ (PDR2, DDR2)

ポート2のレジスタについて説明します。

ポート2のレジスタの機能

ポート2データレジスタ (PDR2)

ポート2の端子ごとの出力値を設定します。

ポート2方向レジスタ (DDR2)

ポート2の端子ごとの入出力方向が設定できます。端子に対応するビットに "1" を設定した場合は、端子は出力ポートになり、"0" を設定した場合は、端子は入力ポートになります。

表 8.5-3 ポート2のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート2データ レジスタ (PDR2)	0	端子状態が "L" レベル です。	PDR2レジスタ のビットに "0" が設定さ れています。	PDR2レジスタ のビットに "0" を設定し ます。	端子より "L" レベルを出 力します。	000002 _H	XXXXXXXX _B
	1	端子状態が "H" レベル です。	PDR2レジスタ のビットに "1" が設定さ れています。	PDR2レジスタ のビットに "1" を設定し ます。	端子より "H" レベルを出 力します。		
ポート2方向 レジスタ (DDR2)	0	DDR2レジスタのビットに "0" が設定されています。		入力ポートになります。		000012 _H	00000000 _B
	1	DDR2レジスタのビットに "1" が設定されています。		出力ポートになります。			

X : 不定

8.5.2 ポート2の動作説明

ポート2の動作を説明します。

ポート2の動作

ポート2方向レジスタ (DDR2) で、出力ポートに設定した場合

- ポート2データレジスタ (PDR2) に設定した値が、ポート2端子へ出力されます。
- ポート2データレジスタ (PDR2) を読出した場合は、ポート2データレジスタ (PDR2) に設定されている値が読出されます。

ポート2方向レジスタ (DDR2) で、入力ポートに設定した場合

- ポート2端子はハイインピーダンスになります。
- ポート2データレジスタ (PDR2) に値を設定した場合は、ポート2データレジスタ (PDR2) に設定した値が保持されますが、端子には出力されません。
- ポート2データレジスタ (PDR2) を読出した場合は、端子の入力レベル値 ("L" の場合は "0", "H" の場合は "1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート2データレジスタ (PDR2) をアクセスする場合は、ポート2方向レジスタ (DDR2) で出力に設定されているビットは影響を受けませんが、ポート2方向レジスタ (DDR2) で入力に設定されているビットは、端子の入力レベル値がポート2データレジスタ (PDR2) に書込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート2データレジスタ (PDR2) に出力値を設定した後に、ポート2方向レジスタ (DDR2) を出力に設定してください。

リセット時の動作

- CPU がリセットされた場合は、ポート2方向レジスタ (DDR2) の値が "00_H" に初期化され、ポート2端子はハイインピーダンスになります。
- ポート2データレジスタ (PDR2) は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート2データレジスタ (PDR2) に出力値を設定した後に、ポート2方向レジスタ (DDR2) を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL) に "1" が設定されていると、端子はポート2方向レジスタ (DDR2) の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

表 8.5-4 ポート2の端子状態

端子名	通常動作	スリープモード	ストップモード, タイムベース タイマモード (SPL="0")	ストップモード, タイムベース タイマモード (SPL="1")
P20 ~ P27	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z
TIN0, TIN1	リロードタイマ0,1 外部クロック入力	リロードタイマ0,1 外部クロック入力		
T00, T01	リロードタイマ0,1 イベント出力	リロードタイマ0,1 イベント出力		
IN0 ~ IN3	インプット キャプチャ0~3 トリガ入力	インプット キャプチャ0~3 トリガ入力		

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

Hi-z : ハイインピーダンス

(注) リソース機能を使用している場合は, TIN0, TIN1, T00, T01, IN0 ~ IN3として機能します。

8.6 ポート3

ポート3は、リソース入出力兼用の入出力ポートです。各端子は、リソースと入出力ポートをビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート3の構成、端子のブロックダイアグラム、レジスタを示します。

ポート3の構成

ポート3は、次の要素から構成されます。

- 入出力ポート端子 / リソース入出力端子 (P30/RT00 ~ P37/SOT0)
- ポート3データレジスタ (PDR3)
- ポート3方向レジスタ (DDR3)

ポート3の端子

ポート3の入出力端子は、リソース入出力端子と兼用になっていますので、リソース入出力端子として使用する場合は、入出力ポートとしては使用できません。

表 8.6-1 ポート3の端子

入出力 ポート名	端子名	入出力ポート機能	リソース機能		入出力形式	
					入力	出力
ポート3	P30/RT00	P30	RT00	外部割込み要求入力0	CMOS(ヒステリシス)	CMOS
	P31/RT01	P31	RT01	外部割込み要求入力1		
	P32/RT02	P32	RT02	外部割込み要求入力2		
	P33/RT03	P33	RT03	外部割込み要求入力3		
	P34/RT04	P34	RT04	外部割込み要求入力4		
	P35/RT05	P35	RT05	外部割込み要求入力5		
	P36/SIN0	P36	SIN0	UART0・シリアル データ入力		
	P37/SOT0	P37	SOT0	UART0・シリアル データ出力		

ポート3端子のブロックダイヤグラム

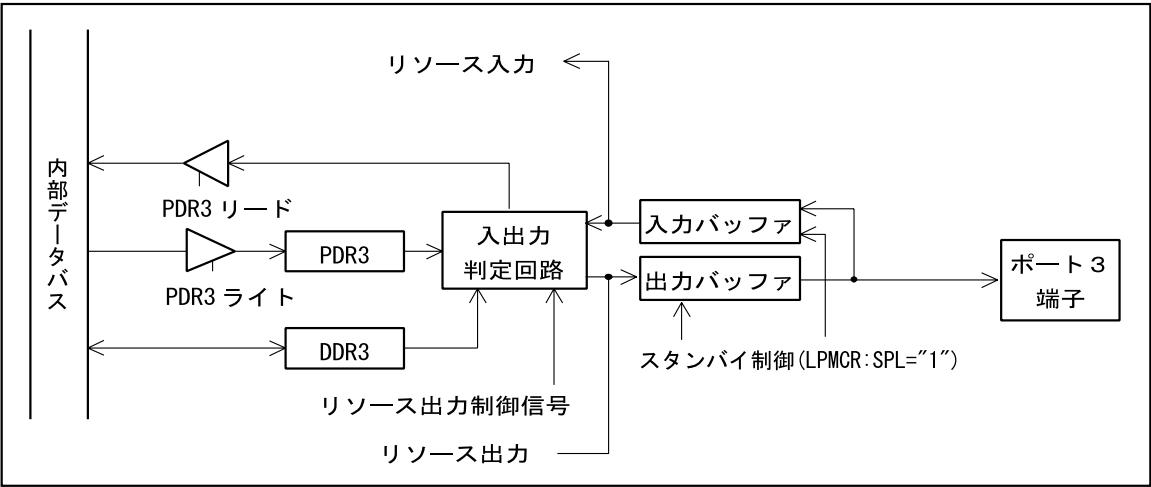


図 8.6-1 ポート3端子のブロックダイヤグラム

ポート3のレジスタ

ポート3のレジスタには、ポート3データレジスタ（PDR3）および、ポート3方向レジスタ（DDR3）があります。各レジスタを構成するビットは、ポート3の端子に1対1で対応しています。

表 8.6-2 ポート3のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
ポート3	PDR3,DDR3	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
	対応する端子	P37	P36	P35	P34	P33	P32	P31	P30

8.6.1 ポート3のレジスタ (PDR3, DDR3)

ポート3のレジスタについて説明します。

ポート3のレジスタの機能

ポート3データレジスタ (PDR3)

ポート3の端子ごとの出力値を設定します。

ポート3方向レジスタ (DDR3)

ポート3の端子ごとの入出力方向が設定できます。端子に対応するビットに "1"を設定した場合は、端子は出力ポートになり、"0"を設定した場合は、端子は入力ポートになります。

表 8.6-3 ポート3のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート3データ レジスタ（PDR3）	0	端子状態が "L" レベル です。	PDR3レジスタ のビットに "0" が設定さ れています。	PDR3レジスタ のビットに "0"を設定し ます。	端子より"L" レベルを出 力します。	000003 _H	XXXXXXXX _B
	1	端子状態が "H" レベル です。	PDR3レジスタ のビットに "1" が設定さ れています。	PDR3レジスタ のビットに "1"を設定し ます。	端子より"H" レベルを出 力します。		
ポート3方向 レジスタ（DDR3）	0	DDR3レジスタのビットに"0" が設定されています。		入力ポートになります。		000013 _H	00000000 _B
	1	DDR3レジスタのビットに"1" が設定されています。		出力ポートになります。			

X : 不定

8.6.2 ポート3の動作説明

ポート3の動作を説明します。

ポート3の動作

ポート3方向レジスタ (DDR3) で、出力ポートに設定した場合

- ポート3データレジスタ (PDR3) に設定した値がポート3端子へ出力されます。
- ポート3データレジスタ (PDR3) を読出した場合は、ポート3データレジスタ (PDR3) に設定されている値が読出されます。

ポート3方向レジスタ (DDR3) で、入力ポートに設定した場合

- ポート3端子はハイインピーダンスになります。
- ポート3データレジスタ (PDR3) に値を設定した場合は、ポート3データレジスタ (PDR3) に設定した値が保持されますが、端子には出力されません。
- ポート3データレジスタ (PDR3) を読出した場合は、端子の入力レベル値 ("L" の場合は "0", "H" の場合は "1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート3データレジスタ (PDR3) をアクセスする場合は、ポート3方向レジスタ (DDR3) で出力に設定されているビットは影響を受けませんが、ポート3方向レジスタ (DDR3) で入力に設定されているビットは、端子の入力レベル値がポート3データレジスタ (PDR3) に書込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート3データレジスタ (PDR3) に出力値を設定した後に、ポート3方向レジスタ (DDR3) を出力に設定してください。

リセット時の動作

- CPU がリセットされた場合は、ポート3方向レジスタ (DDR3) の値は "00H" に初期化され、ポート3端子はハイインピーダンスになります。
- ポート3データレジスタ (PDR3) は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート3データレジスタ (PDR3) に出力値を設定した後に、ポート3方向レジスタ (DDR3) を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL) に "1" が設定されていると、端子はポート3方向レジスタ (DDR3) の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

表 8.6-4 ポート3の端子状態

端子名	通常動作	スリープモード	ストップモード, タイムベース タイマモード (SPL="0")	ストップモード, タイムベース タイマモード (SPL="1")
P30 ~ P37	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z
RT00 ~ RT05	アウトプットコンペア イベント出力0 ~ 5/ 波形生成部出力0 ~ 5	アウトプットコンペア イベント出力0 ~ 5/ 波形生成部出力0 ~ 5		
SIN0	UART0・シリアル データ入力	UART0・シリアル データ入力		
SOT0	UART0・シリアル データ出力	UART0・シリアル データ出力		

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

Hi-z : ハイインピーダンス

(注) リソース機能を使用している場合は, RT00 ~ RT05, SIN0, SOT0として機能します。

8.7 ポート4

ポート4は、リソース入出力兼用の入出力ポートです。各端子は、リソースと入出力ポートを、ビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート4の構成、端子のブロックダイヤグラム、レジスタについて説明します。

ポート4の構成

ポート4は、次の要素から構成されます。

- 入出力ポート端子 / リソース入出力端子 (P40/SCK0 ~ P46/PPG5)
- ポート4データレジスタ (PDR4)
- ポート4方向レジスタ (DDR4)

ポート4の端子

ポート4の入出力端子は、リソース入出力端子と兼用になっていますので、リソース入出力端子として使用する場合は、入出力ポートとしては使用できません。

表 8.7-1 ポート4の端子

入出力 ポート名	端子名	入出力ポート機能	リソース機能		入出力形式	
					入力	出力
ポート4	P40/SCK0	P40	SCK0	UART0・シリアル クロック入出力	CMOS (ヒステリシス)	CMOS
	P41/PPG0	P41	PPG0	PPG0出力		
	P42/PPG1	P42	PPG1	PPG1出力		
	P43/PPG2	P43	PPG2	PPG2出力		
	P44/PPG3	P44	PPG3	PPG3出力		
	P45/PPG4	P45	PPG4	PPG4出力		
	P46/PPG5	P46	PPG5	PPG5出力		

ポート4端子のブロックダイヤグラム

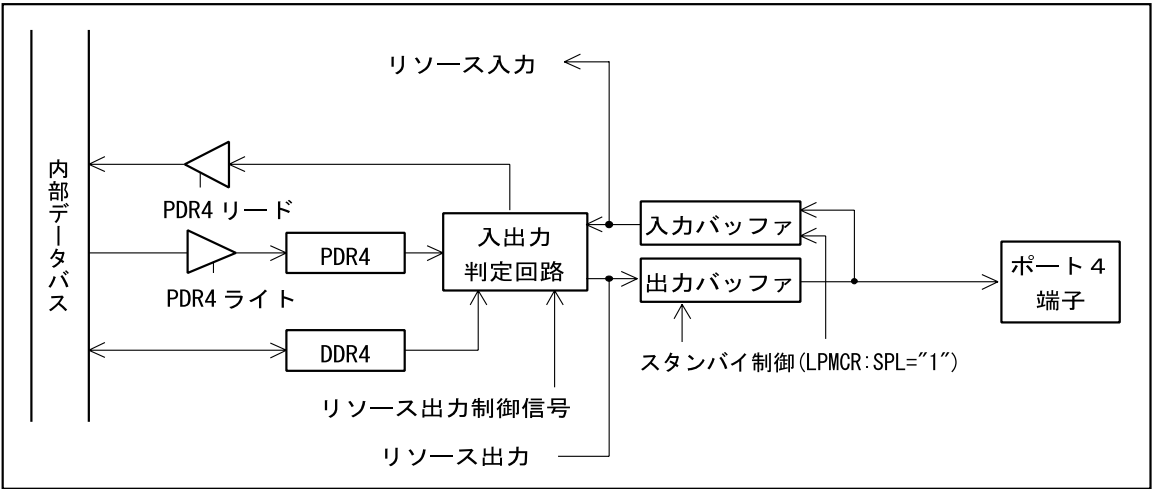


図 8.7-1 ポート4端子のブロックダイヤグラム

< 注意事項 >

リソースの出力許可ビットを許可に設定した場合は、ポート4方向レジスタ (DDR4) の値にかかわらず強制的にリソースの出力端子になります。

ポート4のレジスタ

ポート4のレジスタには、ポート4データレジスタ (PDR4) および、ポート4方向レジスタ (DDR4) があります。各レジスタを構成するビットは、ポート4の端子に1対1で対応しています。

表 8.7-2 ポート4のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
	PDR4,DDR4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ポート4	対応する端子	-	P46	P45	P44	P43	P42	P41	P40

8.7.1 ポート4のレジスタ (PDR4, DDR4)

ポート4のレジスタについて説明します。

ポート4のレジスタの機能

ポート4データレジスタ (PDR4)

ポート4の端子ごとの出力値を設定します。

ポート4方向レジスタ (DDR4)

ポート4の端子ごとの入出力方向が設定できます。端子に対応するビットに "1" を設定した場合は、端子は出力ポートになり、"0" を設定した場合は、端子は入力ポートになります。

参考：

- ・リソース機能の出力端子として使用するには、出力端子に対応するリソースの出力許可ビットを許可に設定してください。ポート4方向レジスタ (DDR4) の設定値に関係なく、リソース機能の出力として使用できます。
- ・リソース機能の入力端子として使用するには、入力端子に対応するポート4方向レジスタ (DDR4) のビットに "0" を設定して入力ポートにしてください。

表 8.7-3 ポート4のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート4データ レジスタ (PDR4)	0	端子状態が "L" レベル です。	PDR4レジスタ のビットに "0" が設定さ れています。	PDR4レジスタ のビットに "0" を設定し ます。	端子より"L" レベルを出 力します。	000004 _H	XXXXXXX _B
	1	端子状態が "H" レベル です。	PDR4レジスタ のビットに "1" が設定さ れています。	PDR4レジスタ のビットに "1" を設定し ます。	端子より"H" レベルを出 力します。		
ポート4方向 レジスタ (DDR4)	0	DDR4レジスタのビットに"0" が設定されています。		入力ポートになります。		000014 _H	00000000 _B
	1	DDR4レジスタのビットに"1" が設定されています。		出力ポートになります。			

X：不定

8.7.2 ポート4の動作説明

ポート4の動作を説明します。

ポート4の動作

ポート4方向レジスタ (DDR4) で、出力ポートに設定した場合

- ポート4データレジスタ (PDR4) に設定した値が、ポート4端子へ出力されます。
- ポート4データレジスタ (PDR4) を読出した場合は、ポート4データレジスタ (PDR4) に設定されている値が読出されます。

ポート4方向レジスタ (DDR4) で、入力ポートに設定した場合

- ポート4端子はハイインピーダンスになります。
- ポート4データレジスタ (PDR4) に値を設定した場合は、ポート4データレジスタ (PDR4) に設定した値が保持されますが、端子には出力されません。
- ポート4データレジスタ (PDR4) を読出した場合は、端子の入力レベル値 ("L" の場合は "0", "H" の場合は "1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート4データレジスタ (PDR4) をアクセスする場合は、ポート4方向レジスタ (DDR4) で出力に設定されているビットは影響を受けませんが、ポート4方向レジスタ (DDR4) で入力に設定されているビットは、端子の入力レベル値がポート4データレジスタ (PDR4) に書き込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート4データレジスタ (PDR4) に出力値を設定した後に、ポート4方向レジスタ (DDR4) を出力に設定してください。

リソース出力時の動作

リソース出力として使用する場合は、リソースの出力許可ビットで出力許可設定を行います。ポート出力とリソース出力の切替えはリソースの出力が優先されますので、ポート4方向レジスタ (DDR4) のリソース兼用端子に対応したビットに "0" が設定されている場合でも、各リソースの出力が許可されていればリソースの出力を行うことができます。また、ポート4方向レジスタ (DDR4) のリソース兼用端子に対応したビットに "0" を設定されている場合に、リソースの出力を許可設定すると、リソースの出力値を読出すことができます。

リソース入力時の動作

リソース入力と兼用の入出力ポートをリソース入力として使用する場合は、ポート4方向レジスタ (DDR4) の対応するビットに "0" を設定してください。

リセット時の動作

- CPUがリセットされた場合は、ポート4方向レジスタ（DDR4）の値が"00_H"に初期化され、ポート4端子はハイインピーダンスになります。
- ポート4データレジスタ（PDR4）は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート4データレジスタ（PDR4）に出力値を設定した後に、ポート4方向レジスタ（DDR4）を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）に"1"が設定されていると、端子はポート4方向レジスタ（DDR4）の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

表 8.7-4 ポート4の端子状態

端子名	通常動作	スリープモード	ストップモード、 タイムベース タイマモード (SPL="0")	ストップモード、 タイムベース タイマモード (SPL="1")
P40～P46	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z
SCK0	UART0・シリアル クロック入出力	UART0・シリアル クロック入出力		
PPG0～5	PPG出力	PPG出力		

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

Hi-z：ハイインピーダンス

(注) リソース機能を使用している場合は、SCK0, PPG0～PPG5として機能します。

8.8 ポート5

ポート5は、A/Dコンバータのアナログ入力兼用の入出力ポートです。各端子は、アナログ入力と入出力ポートを、ビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート5の構成、端子のブロックダイアグラム、レジスタについて説明します。

ポート5の構成

ポート5は、次の要素から構成されます。

- 入出力ポート端子 / A/Dコンバータのアナログ入力端子 (P50/AN0 ~ P57/AN7)
- ポート5データレジスタ (PDR5)
- ポート5方向レジスタ (DDR5)
- アナログ入力許可レジスタ (ADER)

ポート5の端子

ポート5の入出力端子は、A/Dコンバータのアナログ入力と兼用になっていますので、A/Dコンバータのアナログ入力端子として使用する場合は、入出力ポートとして使用できません。また、入出力ポートとして使用する場合は、A/Dコンバータのアナログ入力端子として使用できません。

表 8.8-1 ポート5の端子

入出力 ポート名	端子名	入出力ポート機能	リソース機能		入出力形式	
					入力	出力
ポート5	P50/AN0	P50	AN0	A/Dコンバータ アナログ入力0	CMOS (ヒステリシス)	CMOS
	P51/AN1	P51	AN1	A/Dコンバータ アナログ入力1		
	P52/AN2	P52	AN2	A/Dコンバータ アナログ入力2		
	P53/AN3	P53	AN3	A/Dコンバータ アナログ入力3		
	P55/AN4	P54	AN4	A/Dコンバータ アナログ入力4		
	P55/AN5	P55	AN5	A/Dコンバータ アナログ入力5		
	P56/AN6	P56	AN6	A/Dコンバータ アナログ入力6		
	P57/AN7	P57	AN7	A/Dコンバータ アナログ入力7		

ポート5端子のブロックダイヤグラム

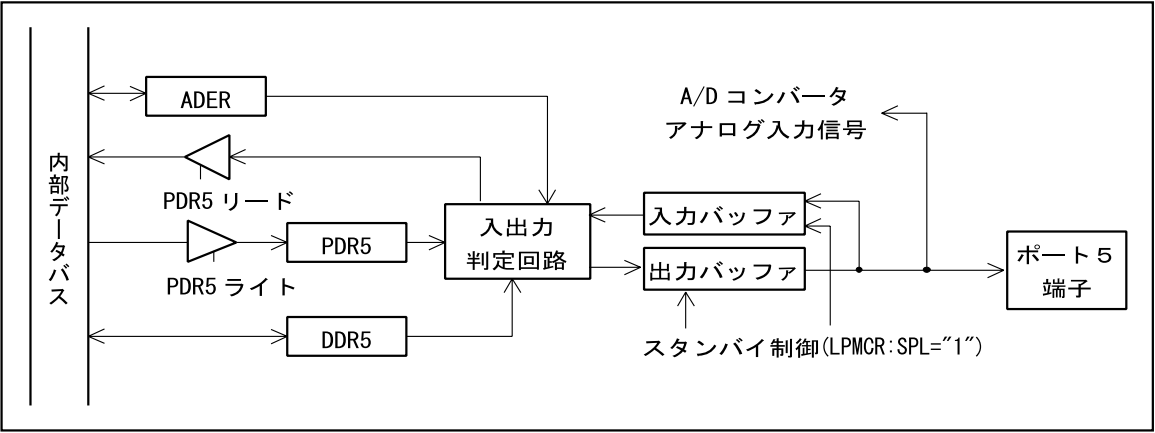


図 8.8-1 ポート5端子のブロックダイヤグラム

< 注意事項 >

入力ポートとして使用する場合は、対応するポート5方向レジスタ (DDR5) のビットに "0" を設定し、更にアナログ入力許可レジスタ (ADER) の対応するビットに "0" を設定してください。

アナログ入力端子として使用する場合は、対応するポート5方向レジスタ (DDR5) のビットに "0" を設定し、更に、アナログ入力許可レジスタ (ADER) の対応するビットに "1" を設定してください。

ポート5のレジスタ

ポート5のレジスタには、ポート5データレジスタ (PDR5)、ポート5方向レジスタ (DDR5) および、アナログ入力許可レジスタ (ADER) があります。各レジスタを構成するビットは、ポート5の端子に1対1で対応しています。

表 8.8-2 ポート5のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子								
	PDR5,DDR5,ADER	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
ポート5	対応する端子	P57	P56	P55	P54	P53	P52	P51	P50

8.8.1 ポート5のレジスタ(PDR5,DDR5,ADER)

ポート5のレジスタについて説明します。

ポート5のレジスタの機能

ポート5データレジスタ (PDR5)

ポート5の端子ごとの出力値を設定します。

ポート5方向レジスタ (DDR5)

ポート5の端子ごとの入出力方向が設定できます。端子に対応するビットに"1"を設定した場合は、端子は出力ポートになり、"0"を設定した場合は、端子は入力ポートになります

アナログ入力許可レジスタ (ADER)

端子ごとに入出力ポートと、A/Dコンバータのアナログ入力を設定できます。ポート（端子）に対応するビットに"1"を設定した場合は、A/Dコンバータのアナログ入力になり、"0"を設定した場合は、入出力ポートになります。

< 注意事項 >

入出力ポートに設定されている場合は、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、A/Dコンバータのアナログ入力を行う場合は、必ずアナログ入力許可レジスタ (ADER) の対応するビットをA/Dコンバータのアナログ入力に設定してください。

参考：

リセットによりポート5方向レジスタ (DDR5)は"00H"，アナログ入力許可レジスタ (ADER)は"FFH"に初期化され、A/Dコンバータのアナログ入力設定になります。

表 8.8-3 ポート5のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート5データ レジスタ (PDR5)	0	端子状態が "L" レベル です。	PDR5レジスタ のビットに "0" が設定さ れています。	PDR5レジスタ のビットに "0"を設定し ます。	端子より"L" レベルを出 力します。	000005 _H	XXXXXXXX _B
	1	端子状態が "H" レベル です。	PDR5レジスタ のビットに "1" が設定さ れています。	PDR5レジスタ のビットに "1"を設定し ます。	端子より"H" レベルを出 力します。		
ポート5方向 レジスタ (DDR5)	0	DDR5レジスタのビットに"0" が設定されています。		入力ポートになります。		000015 _H	00000000 _B
	1	DDR5レジスタのビットに"1" が設定されています。		出力ポートになります。			
アナログ入力許可 レジスタ (ADER)	0	ADERレジスタのビットに"0" が設定されています。		入出力ポート		000017 _H	11111111 _B
	1	ADERレジスタのビットに"1" が設定されています。		A/Dコンバータのアナログ 入力			

X：不定

8.8.2 ポート5の動作説明

ポート5の動作を説明します。

ポート5の動作

ポート5方向レジスタ (DDR5) とアナログ入力許可レジスタ (ADER) で、出力ポートに設定した場合

- ポート5データレジスタ (PDR5) に設定した値が、ポート5端子へ出力されます。
- ポート5データレジスタ (PDR5) を読出した場合は、ポート5データレジスタ (PDR5) に設定されている値が読出されます。

ポート5方向レジスタ (DDR5) とアナログ入力許可レジスタ (ADER) で、入力ポートに設定した場合

- ポート5端子はハイインピーダンスになります。
- ポート5データレジスタ (PDR5) に値を設定した場合は、ポート5データレジスタ (PDR5) に設定した値が保持されますが、端子には出力されません。
- ポート5データレジスタ (PDR5) を読出した場合は、端子の入力レベル値 ("L" の場合は "0", "H" の場合は "1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート5データレジスタ (PDR5) をアクセスする場合は、ポート5方向レジスタ (DDR5) で出力に設定されているビットは影響を受けませんが、ポート5方向レジスタ (DDR5) で入力に設定されているビットは、端子の入力レベル値がポート5データレジスタ (PDR5) に書き込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート5データレジスタ (PDR5) に出力値を設定した後に、ポート5方向レジスタ (DDR5) を出力に設定してください。

A/Dコンバータのアナログ入力に設定する場合

A/Dコンバータのアナログ入力として使用する場合は、A/Dコンバータのアナログ入力端子に対応するアナログ入力許可レジスタ (ADER) のビットに "1" を設定してください。A/Dコンバータのアナログ入力に設定されている場合は、PDR5 の対応するビットをリードすると、読出し値は "0" になります。

リソース出力時の動作

リソース出力として使用する場合は、リソースの出力端子に対応するアナログ入力許可レジスタ (ADER) のビットに "0" を設定し、使用するリソースの出力許可ビットで出力許可設定を行います。ポート出力とリソース出力の切替えはリソースの出力が優先されますので、ポート5方向レジスタ (DDR5) のリソース兼用端子に対応したビットに "0" が設定されている場合でも、各リソースの出力が許可されていればリソースの出力を行うことができます。また、ポート5方向レジスタ (DDR5) のリソース兼用端子に対応したビットに "0" が設定されている場合に、リソースの出力を許可設定すると、リソースの出力値を読出すことができます。

リソース入力時の動作

リソース入力と兼用の入出力ポートをリソース入力として使用する場合は、リソースの入力端子に対応するアナログ入力許可レジスタ (ADER) のビットに "0" を設定し、ポート5方向レジスタ (DDR5) の対応するビットに "0" を設定してください。

リセット時の動作

- CPUがリセットされた場合は、ポート5方向レジスタ (DDR5) の値が "00H" に、アナログ入力許可レジスタ (ADER) の値が "FFH" に初期化され、A/Dコンバータのアナログ入力に設定されます。入出力ポートとして使用する場合は、アナログ入力許可レジスタ (ADER) に "00H" を設定し、ポート入出力モードに設定してください。
- ポート5データレジスタ (PDR5) は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート5データレジスタ (PDR5) に出力値を設定した後に、ポート5方向レジスタ (DDR5) を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ (LPMCR) の端子状態設定ビット (SPL) に "1" が設定されていると、端子はポート5方向レジスタ (DDR5) の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

表 8.8-4 ポート5の端子状態

端子名	通常動作	スリープモード	ストップモード, タイムベース タイマモード (SPL="0")	ストップモード, タイムベース タイマモード (SPL="1")
P50 ~ P57	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z
AN0 ~ AN7	A/Dコンバータ0~7 アナログ入力	A/Dコンバータ0~7 アナログ入力		

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

Hi-z : ハイインピーダンス

(注) リソース機能を使用している場合は、AN0 ~ AN7として機能します。

8.9 ポート6

ポート6は、リソース入出力兼用の入出力ポートです。各端子は、リソースと入出力ポートを、ビット単位で切り替えて使用します。ここでは、入出力ポートを中心に、ポート6の構成、端子のブロックダイヤグラム、レジスタを示します。

ポート6の構成

ポート6は、次の要素から構成されます。

- 入出力ポート端子 / リソース入出力端子 (P60/SIN1 ~ P63/INT7/DTTI)
- ポート6データレジスタ (PDR6)
- ポート6方向レジスタ (DDR6)

ポート6の端子

ポート6の入出力端子は、リソース入出力端子と兼用になっていますので、リソース入出力端子として使用する場合は、入出力ポートとしては使用できません。

表 8.9-1 ポート6の端子

入出力 ポート名	端子名	入出力ポート機能	リソース機能		入出力形式	
					入力	出力
ポート6	P60/SIN1	P60	SIN1	UART1・シリアル データ入力	CMOS (ヒステリシス)	CMOS
	P61/SOT1	P61	SOT1	UART1・シリアル データ出力		
	P62/SCK1	P62	SCK1	UART1・シリアル クロック入出力		
	P63/INT7	P63	INT7	外部割込み入力7		

ポート6端子ブロックダイヤグラム

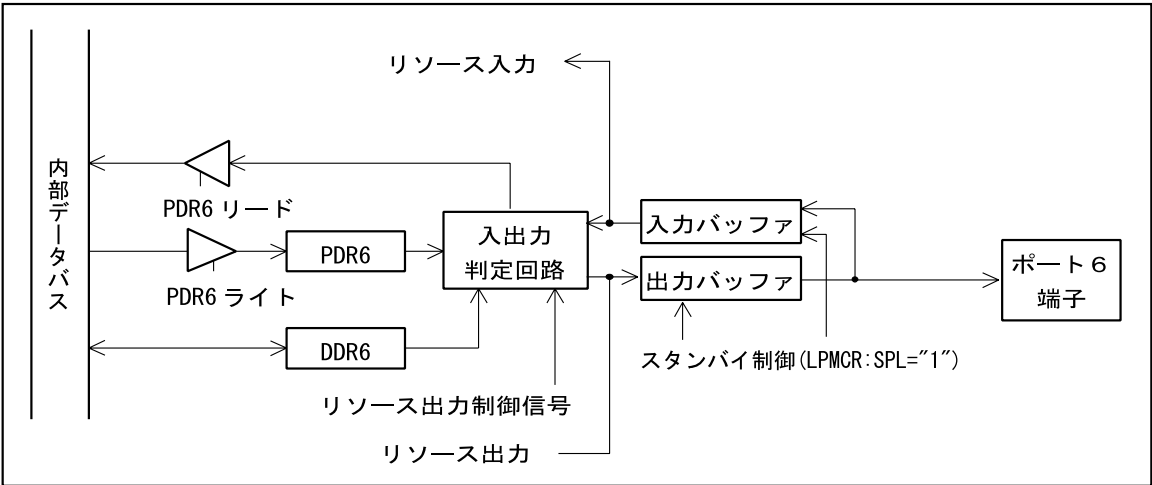


図 8.9-1 ポート6端子のブロックダイヤグラム

< 注意事項 >

リソースの出力許可ビットを許可に設定した場合は、ポート6方向レジスタ (DDR6)の値にかかわらず強制的にリソースの出力端子になります。

ポート6のレジスタ

ポート6のレジスタには、ポート6データレジスタ (PDR6) および、ポート6方向レジスタ (DDR6) があります。各レジスタを構成するビットは、ポート6の端子に1対1で対応しています。

表 8.9-2 ポート6のレジスタと端子の対応

入出力 ポート名	関連するレジスタのビットと対応する端子									
	PDR6,DDR6 対応する端子	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ポート6		-	-	-	-	P63	P62	P61	P60	

8.9.1 ポート6のレジスタ (PDR6, DDR6)

ポート6のレジスタについて説明します。

ポート6のレジスタの機能

ポート6データレジスタ (PDR6)

ポート6の端子ごとの出力値を設定します。

ポート6方向レジスタ (DDR6)

ポート6の端子ごとの入出力方向を設定できます。端子に対応するビットに "1" を設定した場合は、端子は出力ポートになり、"0" を設定した場合は、端子は入力ポートになります。

参考：

- ・リソース機能の出力端子として使用するには、出力端子に対応するリソースの出力許可ビットを許可に設定してください。ポート6方向レジスタ (DDR6) の設定値に関係なく、リソース機能の出力端子として使用できます。
- ・リソース機能の入力端子として使用するには、入力端子に対応するポート6方向レジスタ (DDR6) のビットに "0" を設定して入力ポートにしてください。

表 8.9-3 ポート6のレジスタの機能

レジスタ名	bit 値	リードした場合		ライトした場合		アドレス	初期値
		入力ポート	出力ポート	入力ポート	出力ポート		
ポート6データ レジスタ (PDR6)	0	端子状態が "L" レベル です。	PDR6レジスタ のビットに "0" が設定さ れています。	PDR6レジスタ のビットに "0" を設定し ます。	端子より "L" レベルを出 力します。	000006 _H	XXXXXXX _B
	1	端子状態が "H" レベル です。	PDR6レジスタ のビットに "1" が設定さ れています。	PDR6レジスタ のビットに "1" を設定し ます。	端子より "H" レベルを出 力します。		
ポート6方向 レジスタ (DDR6)	0	DDR6レジスタのビットに "0" が設定されています。		入力ポートになります。		000016 _H	XXXX0000 _B
	1	DDR6レジスタのビットに "1" が設定されています。		出力ポートになります。			

X : 不定

8.9.2 ポート6の動作説明

ポート6の動作を説明します。

ポート6の動作

ポート6方向レジスタ (DDR6) で、出力ポートに設定した場合

- ポート6データレジスタ (PDR6) に設定した値が、ポート6端子へ出力されます。
- ポート6データレジスタ (PDR6) を読出した場合は、ポート6データレジスタ (PDR6) に設定されている値が読出されます。

ポート6方向レジスタ (DDR6) で、入力ポートに設定した場合

- ポート6端子はハイインピーダンスになります。
- ポート6データレジスタ (PDR6) に値を設定した場合は、ポート6データレジスタ (PDR6) に設定した値が保持されますが、端子には出力されません。
- ポート6データレジスタ (PDR6) を読出した場合は、端子の入力レベル値 ("L" の場合は "0", "H" の場合は "1") が読出されます。

< 注意事項 >

リードモディファイライト系命令 (ビットセット命令など) を使用して、ポート6データレジスタ (PDR6) をアクセスする場合は、ポート6方向レジスタ (DDR6) で出力に設定されているビットは影響を受けませんが、ポート6方向レジスタ (DDR6) で入力に設定されているビットは、端子の入力レベル値がポート6データレジスタ (PDR6) に書込まれますので、入力に設定していたビットを出力に設定変更する場合は、ポート6データレジスタ (PDR6) に出力データを設定した後に、ポート6方向レジスタ (DDR6) を出力に設定してください。

リソース出力時の動作

リソース出力として使用する場合は、リソースの出力許可ビットで出力許可設定を行います。ポート出力とリソース出力の切替えはリソースの出力が優先されますので、ポート6方向レジスタ (DDR6) のリソース兼用端子に対応したビットに "0" が設定されている場合でも、各リソースの出力が許可されていればリソースの出力を行うことができます。また、ポート6方向レジスタ (DDR6) のリソース兼用端子に対応したビットに "0" が設定されている場合に、リソースの出力を許可設定すると、リソースの出力値を読出すことができます。

リソース入力時の動作

リソース入力と兼用の入出力ポートをリソース入力として使用する場合は、ポート6方向レジスタ (DDR6) の対応するビットに "0" を設定してください。

リセット時の動作

- CPUがリセットされた場合は、ポート6方向レジスタ（DDR6）の値が"00H"に初期化され、ポート6端子はハイインピーダンスになります。
- ポート6データレジスタ（PDR6）は、リセットでは初期化されませんので、出力ポートとして使用する場合は、ポート6データレジスタ（PDR6）に出力値を設定した後に、ポート6方向レジスタ（DDR6）を出力に設定する必要があります。

ストップ、タイムベースタイマモード時の動作

ストップモード、タイムベースタイマモードに遷移した場合は、低消費電力モード制御レジスタ（LPMCR）の端子状態設定ビット（SPL）に"1"が設定されていると、端子はポート6方向レジスタ（DDR6）の値に関係なくハイインピーダンスになります。また、入力開放によるリークを防ぐため、入力バッファも強制的に遮断されます。

表 8.9-4 ポート6の端子状態

端子名	通常動作	スリープモード	ストップモード、 タイムベース タイマモード (SPL="0")	ストップモード、 タイムベース タイマモード (SPL="1")
P60～P63	入出力ポート	入出力ポート	入力遮断 /出力保持	入力遮断 /出力Hi-z
SIN1	UART1・シリアル データ入力	UART1・シリアル データ入力		
SOT1	UART1・シリアル データ出力	UART1・シリアル データ出力		
SCK1	UART1・シリアル クロック入出力	UART1・シリアル クロック入出力		
INT7	外部割込み要求7	外部割込み要求7	入力遮断 /出力保持 〔外部割込み許可 の場合は入力可〕	入力遮断 /出力Hi-z 〔外部割込み許可 の場合は入力可〕

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

Hi-z：ハイインピーダンス

(注) リソース機能を使用している場合は、SIN1, SOT1, SCK1, INT7として機能します。

8.10 入出力ポートのプログラム例

入出力ポートを使ったプログラム例を示します。

入出力ポートのプログラム例

処理仕様

- ポート0, 1で, 7セグメント (Dpを含めると8セグメント) LEDをすべて点灯させます。
- P00端子がLEDのアノードコモン端子に, P10 ~ P17端子がセグメント端子に対応します

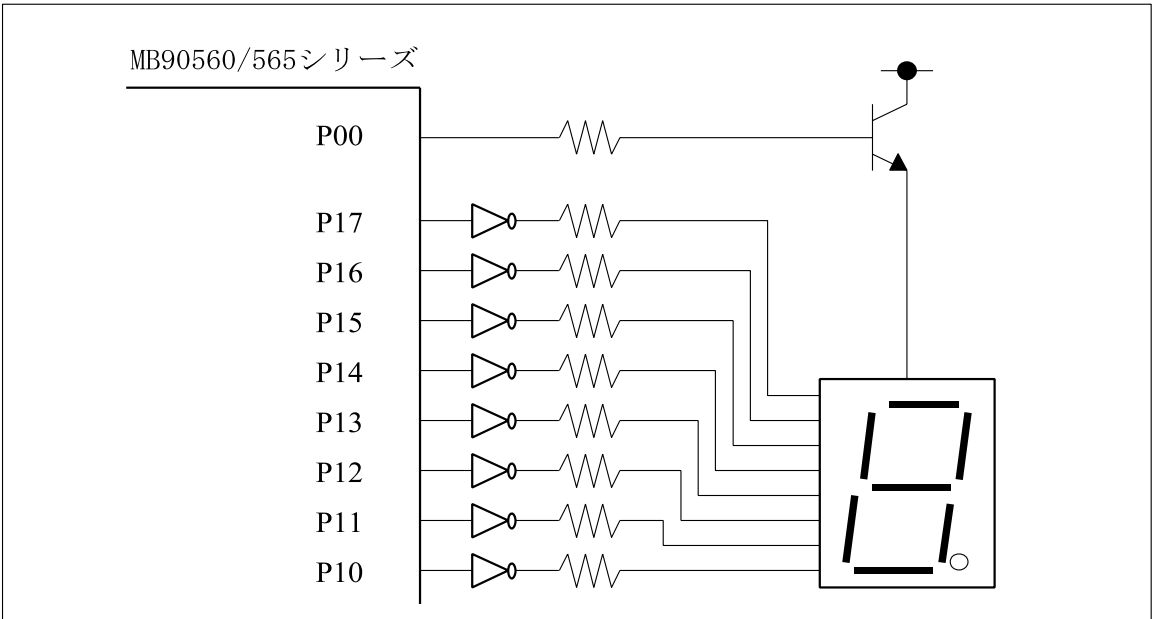


図 8.10-1 8セグメントLED接続例

コーディング例

```
PDR0 EQU 000000H
PDR1 EQU 000001H
DDR0 EQU 000010H
DDR1 EQU 000011H
;-----メインプログラム-----
CODE CSEG
START:
;初期設定済み
MOV I:PDR0, #00000000B ; P00 を"L" レベルに設定, #xxxxxxx0B
MOV I:DDR0, #11111111B ; ポート0を全ビット出力に設定
MOV I:PDR1, #11111111B ; ポート1は全ビット"1"とする
MOV I:DDR1, #11111111B ; ポート1を全ビット出力に設定
CODE ENDS
;-----
END START
```

第9章 タイムベースタイマ

この章では、MB90560/565シリーズのタイムベースタイマの機能と動作について説明します。

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

9.1 タイムベースタイマの概要

タイムベースタイマは、メインクロックに同期してカウントアップする18ビットのフリーランカウンタです。4種類のインターバル時間を設定できるインターバルタイマ機能と、発振安定待ち時間用のタイマやウォッチドッグタイマなどにクロックを供給する機能があります。

インターバルタイマ機能

インターバルタイマ機能は、一定時間間隔で割込み要求を出力する機能です。

- タイムベースタイマカウンタのインターバルタイマ用ビットがオーバーフローした場合に割込み要求を出力します。
- インターバルタイマのインターバル時間は、4種類の中から設定できます。

表 9.1-1 タイムベースタイマのインターバル時間

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

HCLK：発振クロック周波数

() 内の値は、発振クロック周波数4MHzの場合です。

クロック供給機能

クロック供給機能は、発振安定待ち時間用のタイマや、一部の周辺機能に対する動作クロックを供給する機能です。

表 9.1-2 タイムベースタイマから供給されるクロック周期

クロック供給先	クロック周期	備考
発振安定待ち時間	2^{13} /HCLK (約 2.0ms)	セラミック振動子用発振安定待ち時間
	2^{15} /HCLK (約 8.2ms)	水晶振動子用発振安定待ち時間
	2^{17} /HCLK (約 32.8ms)	
ウォッチドッグタイマ	2^{12} /HCLK (約 1.0ms)	ウォッチドッグタイマ のカウントアップクロック
	2^{14} /HCLK (約 4.1ms)	
	2^{16} /HCLK (約 16.4ms)	
	2^{19} /HCLK (約 131.1ms)	

HCLK：発振クロック周波数

() 内の値は、発振クロック周波数4MHzの場合です。

参考：

発振開始直後は発振周期が不安定なため、発振安定待ち時間は目安となります。

9.2 タイムベースタイマの構成

タイムベースタイマは、以下の4種類のブロックで構成されています。

- ・ タイムベースタイマカウンタ
- ・ カウンタクリア回路
- ・ インターバルタイマセクタ
- ・ タイムベースタイマ制御レジスタ (TBTC)

タイムベースタイマのブロックダイアグラム

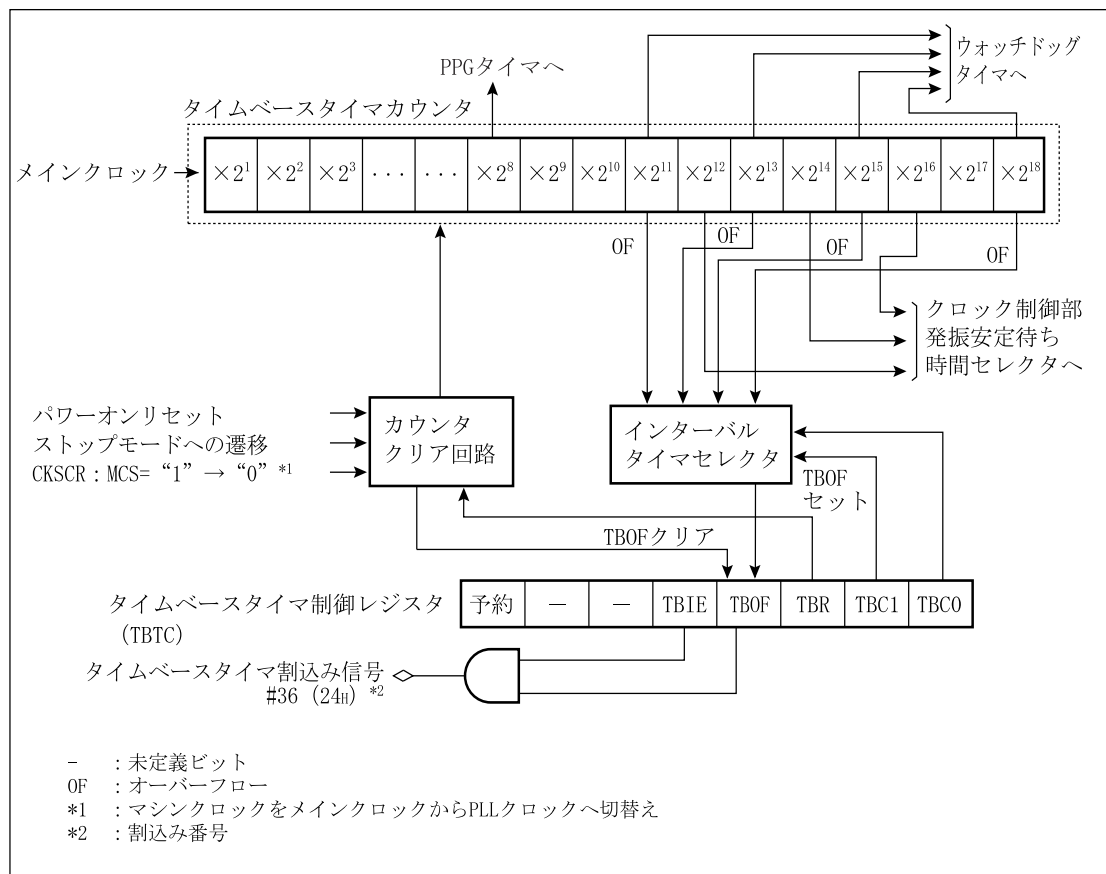


図 9.2-1 タイムベースタイマのブロックダイアグラム

タイムベースタイマカウンタ

メインクロックをカウントクロックとする、18ビットのアップカウンタです。

カウンタクリア回路

タイムベースタイマ制御レジスタ (TBTC) のタイムベースタイマ初期化ビット (TBR) への "0" 設定、パワーオンリセット、ストップモードへの遷移 (LPMCR : STP = "1")、およびマシンクロックをメインクロックからPLLクロックへ切替え (CKSCR : MCS = "1" → "0") でタイムベースタイマカウンタをクリアする回路です。

インターバルタイマセレクタ

タイムベースタイマカウンタの4つのインターバルタイマ用ビットの出力から一つを設定します。設定したインターバルタイマ用ビットのオーバーフローが割込み要因となります。

タイムベースタイマ制御レジスタ (TBTC)

インターバル時間の設定，タイムベースタイマカウンタのクリア，割込み要求の制御，および状態を確認します。

9.3 タイムベースタイマ制御レジスタ (TBTC)

タイムベースタイマ制御レジスタ (TBTC) は、インターバル時間の設定、タイムベースタイマカウンタのクリア、割り込み要求の制御、および状態の確認をするレジスタです。

タイムベースタイマ制御レジスタ (TBTC)

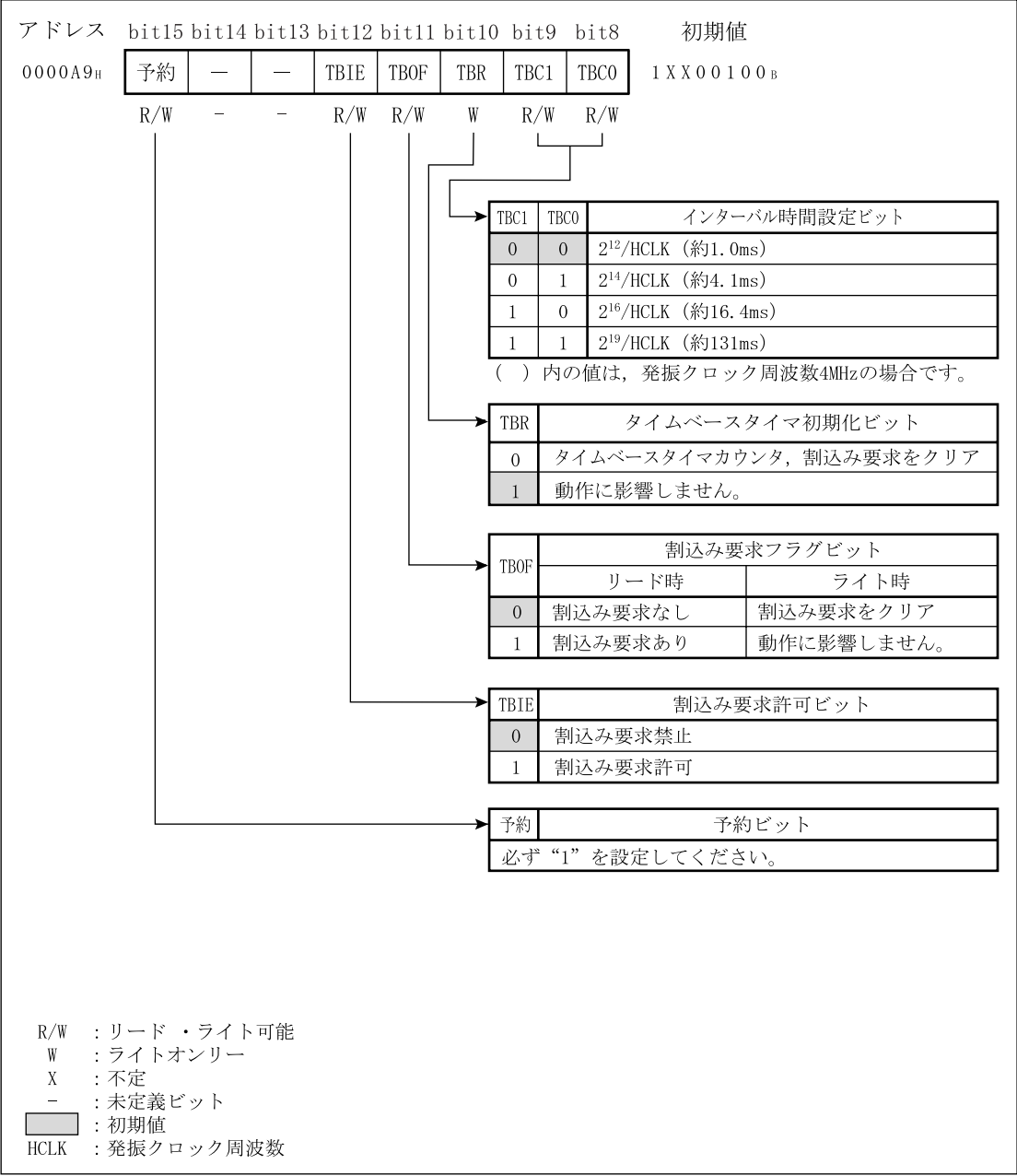


図 9.3-1 タイムベースタイマ制御レジスタ (TBTC)

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

ビット名		機 能
bit15	予約： 予約ビット	・必ず"1"を設定してください。
bit14 bit13	-： 未定義ビット	・読出しを行った場合は，不定値となります。 ・設定した値は，動作に影響しません。
bit12	TBIE： 割込み要求 許可ビット	・割込み要求を許可するビットです。 ・"1"を設定した場合は，割込み要求フラグビット (TBOF) に，"1"がセットされると，割込み要求を出力します。
bit11	TBOF： 割込み要求 フラグビット	・割込み要求のフラグビットです。 ・タイムベースタイマカウンタの設定したインターバルタイマ用ビットのオーバーフローで，"1"がセットされます。 ・割込み要求許可ビット (TBIE) に"1"を設定している場合に，"1"がセットされると，割込み要求を出力します。 ・"0"を設定した場合は，割込み要求がクリアされます。 ・"1"を設定した場合は，動作に影響しません。 <注記> ・"0"を設定する場合は，割込み要求許可ビット (TBIE)，またはプロセッサステータス (PS) のインタラプトレベルマスキレジスタ (ILM) を割込み禁止に設定してください。 ・ストップモードへの遷移，タイムベースタイマ初期化ビット (TBR) によるタイムベースタイマのクリア，およびリセットにより，"0"にクリアされます。
bit10	TBR： タイムベース タイマ初期化 ビット	・タイムベースタイマカウンタをクリアするビットです。 ・"0"を設定した場合は，タイムベースタイマカウンタがクリアされ，割込み要求フラグビット (TBOF) も"0"にクリアされます。 ・"1"を設定した場合は，動作に影響しません。 [参考] 読み出し値は，"1"です。
bit9 bit8	TBC1, TBC0： インターバル 時間設定 ビット	・インターバルタイマの周期を設定するビットです。 ・タイムベースタイマカウンタのインターバルタイマ用のビットを指定します。 ・4種類のインターバル時間から設定できます。

9.4 タイムベースタイマの割込み

設定されたインターバルタイマ用ビットが、オーバーフローすることにより、割込み要求を出力させることができます（インターバルタイマ機能）。

タイムベースタイマの割込み

タイムベースタイマカウンタがメインクロックでカウントアップし、設定されたインターバルタイマがオーバーフローした場合に、タイムベースタイマ制御レジスタ（TBTC）の割込み要求フラグビット（TBOF）に"1"がセットされます。割込み要求許可ビットを許可（TBTC：TBIE="1"）している状態で、割込み要求フラグビットに"1"がセットされると、割込み要求（割込み番号#36）を出力し、割込み処理ルーチンを実行します。割込み処理ルーチンでは、割込み要求フラグビット（TBOF）に"0"を設定し、割込み要求をクリアしてください。割込み要求フラグビット（TBOF）は、割込み要求許可ビット（TBIE）の値に関係なく、設定したインターバルタイマ用ビットがオーバーフローした場合に、"1"がセットされます。

< 注意事項 >

タイムベースタイマ制御レジスタ（TBTC）の割込み要求フラグビット（TBOF）を"0"にクリアする場合は、割込み要求許可ビット（TBIE）、またはプロセッサステータス（PS）のILMレジスタを割込み禁止に設定してください。

参考：

タイムベースタイマでは、拡張インテリジェントI/Oサービス（EI²OS）は使用できません。

タイムベースタイマの割込みとEI²OS

表 9.4-1 タイムベースタイマの割込みとEI²OS

割込み番号	割込みレベル設定レジスタ		ベクタテーブルのアドレス			EI ² OS
	レジスタ名	アドレス	下位	上位	バンク	
#36 (24 _H)	ICR12	0000BC _H	FFFF6C _H	FFFF6D _H	FFFF6E _H	x

x：使用不可

< 注意事項 >

ICR12はタイムベースタイマ割込みとインプットキャプチャ・チャンネル2/3割込みで共用しています。割込みは使用できますが、割込みレベルは同一になります。

9.5 タイムベースタイマの動作説明

インターバルタイマ機能の動作，発振安定待ち時間用のタイマ機能，およびクロック供給機能について示します。

インターバルタイマ機能（タイムベースタイマ）の動作

インターバルタイマ機能は，設定されたインターバル時間ごとに割り込み要求を出力します。インターバルタイマとして動作させるには，図 9.5-1に示すような設定が必要です。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
0000A9H	予約	—	—	TBIE	TBOF	TBR	TBC1	TBC0
	1	—	—	◎	0	0	◎	◎

◎ : 使用ビット
0 : “0” を設定
1 : “1” を設定
— : 未定義ビット

図 9.5-1 タイムベースタイマの設定

- タイムベースタイマカウンタは，クロックが発振していれば，カウントアップを継続します。
- タイムベースタイマカウンタがクリア（TBTC:TBR="0"）された場合は，"00000000000000000000_B"からカウントアップを行い，設定されたインターバルタイマ用ビットがオーバーフローした場合に，タイムベースタイマ制御レジスタ（TBTC）の割り込み要求フラグビット（TBOF）に"1"がセットされます。オーバーフローした場合に，割り込み要求許可ビットを許可（TBIE="1"）している場合は，カウンタがクリアされた時間を基準にして，設定されたインターバル時間ごとに割り込み要求を出力します。
- インターバル時間は，タイムベースタイマのクリア動作により，設定時間よりも長くなる場合があります。

発振安定待ち時間用タイマ機能

タイムベースタイマは，発振クロックの発振安定待ち時間，およびPLLクロックの発振安定待ち時間用のタイマとしても使用されます。発振安定待ち時間は，タイムベースタイマカウンタ値が"00000000000000000000_B"（カウンタクリア）からカウントアップし，発振安定待ち時間を検出するまでの時間となります。タイムベースタイマモードからPLLクロックモードへ復帰する場合は，タイムベースタイマカウンタがクリアされないため，発振安定待ち時間は，カウント途中からの時間となります。

表 9.5-1 タイムベースタイマカウンタのクリア動作と発振安定待ち時間

動 作	カウンタ クリア	割込み要求フラグ ビット (TBOF) クリ ア	発振安定待ち時間
TBTC : TBRに"0"を設定する			
パワーオンリセット			発振クロック発振安定待ち時間
ウォッチドッグリセット			
ストップモードの解除			発振クロック発振安定待ち時間 (メインクロックモード復帰時)
発振クロックモードから PLLクロックモードへの遷移 (MCS = "1" "0")			PLLクロック発振安定待ち時間
タイムベースタイマモードの解除	×	×	PLLクロック発振安定待ち時間 (PLLクロックモード復帰時)
スリープモードの解除	×	×	

: クリアを行う。

× : クリアを行わない。

クロック供給機能

タイムベースタイマは、ウォッチドックタイマにクロックを供給しています。タイムベースタイマカウンタをクリアした場合は、ウォッチドックタイマの動作に影響を与えます。詳細は、「9.6 タイムベースタイマ使用上の注意」を参照してください。

タイムベースタイマの動作状態

次の状態の動作を図 9.5-2に示します。

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

ストップモードへ遷移した場合、タイムベースタイマカウンタはクリアされ、タイムベースタイマカウンタは停止します。ストップモードからの復帰を行う場合は、タイムベースタイマカウンタで発振安定待ち時間をカウントします。

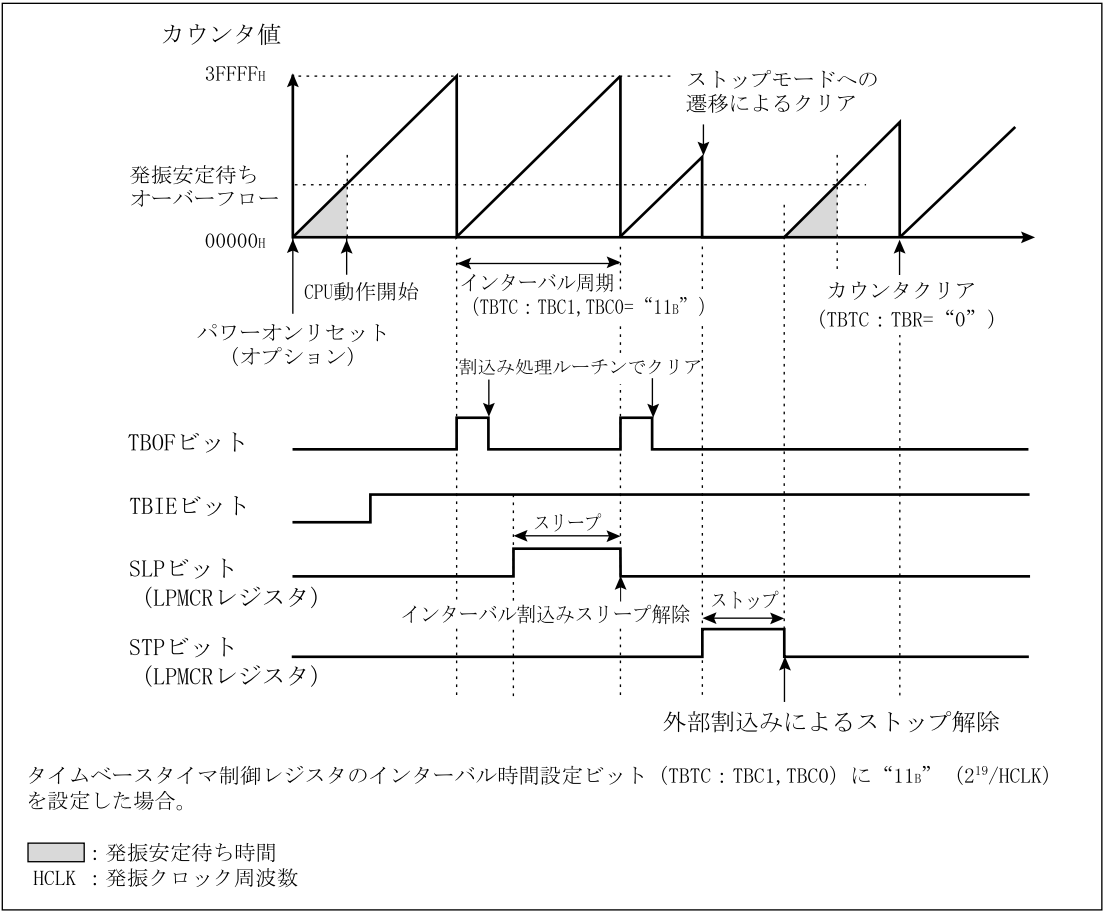


図 9.5-2 タイムベースタイマの動作状態

9.6 タイムベースタイマ使用上の注意

割込み要求のクリアおよびタイムベースタイマカウンタのクリアによる、機能への影響などの注意点を示します。

タイムベースタイマ使用上の注意

割込み要求のクリア

タイムベースタイマ制御レジスタ (TBTC) の割込み要求フラグビット (TB0F) を "0" にクリアする場合は、割込み要求許可ビット (TBIE), またはプロセッサステータス (PS) のインタラプトレベルマスクレジスタ (ILM) を割込み禁止に設定してください。

タイムベースタイマカウンタのクリアにより影響を受ける機能

- インターバルタイマ機能 (インターバル割込み)
- ウォッチドッグタイマ

タイムベースタイマカウンタより、クロックを供給される機能への注意

動作クロックが停止するストップモードでは、タイムベースタイマカウンタはクリアされ、タイムベースタイマカウンタは停止します。タイムベースタイマカウンタより供給されていたクロックは、タイムベースタイマカウンタがクリアされた場合に、初期状態からの供給となるため、"H" レベルが短く、あるいは "L" レベルが最大で 1/2 周期長くなる場合があります。ウォッチドッグタイマ用のクロックも同様に、初期状態からの供給となりますが、ウォッチドッグタイマのカウントが同時にクリアされるため、ウォッチドッグタイマは正常な周期で動作します。

発振安定待ち時間用タイマとして使用する場合

電源投入、およびストップモードの場合は、発振クロックが停止しているため、振動子が動作を開始したあと、タイムベースタイマカウンタが発振子から供給されるクロックを使用して動作クロックの発振安定待ち時間をとります。振動子の種類によって、適切な発振安定待ち時間を設定する必要があります。

詳細は、「4.5 発振安定待ち時間」を参照してください。

9.7 タイムベースタイマのプログラム例

タイムベースタイマのプログラム例を示します。

タイムベースタイマのプログラム例

処理仕様

$2^{12}/\text{HCLK}$ (HCLK: 発振クロック周波数) のインターバル割込みを繰り返し出力します。インターバル時間は、約1.0ms (発振クロック周波数4MHzの場合) となります。

コーディング例

```

ICR12 EQU      0000BCH          ; タイムベースタイマ用割込み制御レジスタ
TBTC   EQU      0000A9H          ; タイムベースタイマ制御レジスタ
TBOF   EQU      TBTC:3           ; 割込み要求フラグビット
;-----メインプログラム-----
CODE    CSEG
START:
;      :                          ; スタックポインタ (SP) などは初期化済み
;      :                          ; とする
;      AND      CCR, #0BFH        ; 割込みディセーブル
;      MOV      I:ICR12, #00H     ; 割込みレベル0 (最強)
;      MOV      I:TBTC, #10010000B ; 上位3ビットは固定
;      :                          ; 割込み許可, TBOF"0"にクリア
;      :                          ; カウンタクリア
;      :                          ; インターバル時間 $2^{12}/\text{HCLK}$ に設定
;      MOV      ILM, #07H        ; PS内 ILMをレベル7に設定
;      OR       CCR, #40H        ; 割込みイネーブル
LOOP:   MOV      A, #00H          ; 無限ループ
;      MOV      A, #01H
;      BRA      LOOP
;-----割込みプログラム-----
WARI:
;      CLRB     I:TBOF            ; 割込み要求フラグを"0"にクリア
;      :
;      :      ユーザ処理
;      :
;      RETI                      ; 割込みから復帰
CODE    ENDS
;-----ベクタ設定-----
VECT    CSEG      ABS=0FFH
;      ORG      OFF6CH            ; 割込み#36(24H)にベクタを設定
;      DSL      WARI
;      ORG      OFFDCH            ; リセットベクタを設定
;      DSL      START
;      DB       00H              ; シングルチップモードに設定
VECT    ENDS
END      START

```

第10章 ウォッチドッグタイマ

この章では、MB90560/565シリーズの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。

- 10.1 ウォッチドッグタイマの概要
- 10.2 ウォッチドッグタイマの構成
- 10.3 ウォッチドッグタイマ制御レジスタ (WDTC)
- 10.4 ウォッチドッグタイマの動作説明
- 10.5 ウォッチドッグタイマ使用上の注意
- 10.6 ウォッチドッグタイマのプログラム例

10.1 ウォッチドッグタイマの概要

ウォッチドッグタイマは、タイムベースタイマカウンタの出力をカウントクロックとする2ビットのタイマです。ウォッチドッグタイマを起動した場合は、設定されたインターバル時間内に、ウォッチドッグタイマをクリアしないと、CPUをリセットします。

ウォッチドッグタイマ機能

ウォッチドッグタイマは、プログラム暴走対策用のタイマです。ウォッチドッグタイマを起動した場合は、設定されたインターバル時間内にウォッチドッグタイマをクリアし続ける必要があります。プログラムが無限ループに陥るなどして、表 10.1-1に示す最小時間以上ウォッチドッグタイマがクリアされない場合は、CPUに対してウォッチドッグリセットを発生し、リセット状態へ遷移します。ウォッチドッグタイマのインターバル時間は、ウォッチドッグタイマ制御レジスタ(WDTC)のインターバル時間設定ビット(WT1,WT0)で設定します。

表 10.1-1 ウォッチドッグタイマのインターバル時間

WT1	WT0	インターバル時間		
		最小*	最大*	発振クロックサイクル数
0	0	約 3.58ms	約 4.61ms	$2^{14} \pm 2^{11}$ サイクル
0	1	約 14.33ms	約 18.3ms	$2^{16} \pm 2^{13}$ サイクル
1	0	約 57.23ms	約 73.73ms	$2^{18} \pm 2^{15}$ サイクル
1	1	約458.75ms	約589.82ms	$2^{21} \pm 2^{18}$ サイクル

*：発振クロック周波数4MHzの場合です。

ウォッチドッグタイマのインターバル時間の詳細は、「10.4 ウォッチドッグタイマの動作説明」を参照してください。

< 注意事項 >

ウォッチドッグタイマは、タイムベースタイマの桁上り信号をカウントする2ビットのタイマで構成されています。タイムベースタイマがクリアされた場合は、ウォッチドッグリセットのインターバル時間が、設定された時間より長くなることがあります。

参考：

ウォッチドッグタイマを起動後の停止は、パワーオンリセット、ウォッチドッグタイマによるリセットで行うことができます。外部リセット、内部リセット、ウォッチドッグタイマ制御レジスタ(WDTC)のウォッチドッグ制御ビット(WTE)の設定、スリープモード、およびストップモードへの遷移では、ウォッチドッグタイマはクリアできますが、ウォッチドッグ機能は設定されたままで、停止させることはできません。

10.2 ウォッチドッグタイマの構成

ウォッチドッグタイマは、以下の5種類のブロックで構成されています。

- ・ カウントクロックセクタ
- ・ ウォッチドッグタイマ (2ビットタイマ)
- ・ ウォッチドッグリセット発生回路
- ・ タイマクリア制御回路
- ・ ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグタイマのブロックダイアグラム

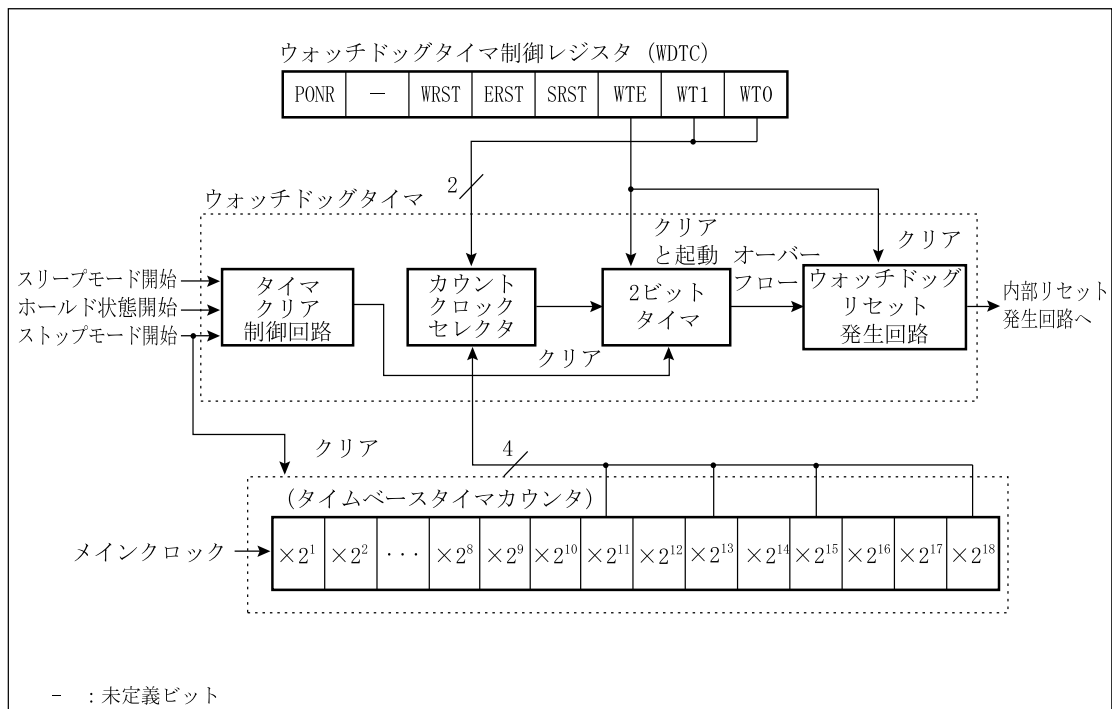


図 10.2-1 ウォッチドッグタイマのブロックダイアグラム

カウントクロックセクタ

ウォッチドッグタイマのカウントクロックを、4種類のタイムベースタイマ出力クロックから設定できます。ウォッチドッグタイマのカウントクロックが設定されることで、ウォッチドッグリセットのインターバル時間が設定できます。

ウォッチドッグタイマ (2ビットタイマ)

ウォッチドッグタイマは、カウントクロックセクタで設定されたクロックをカウントする2ビットのタイマです。

ウォッチドッグリセット発生回路

ウォッチドッグタイマのオーバーフローで、リセット信号を発生します。

タイマクリア制御回路

ウォッチドッグタイマのクリアと，ウォッチドッグタイマの動作 / 停止を制御します。

ウォッチドッグタイマ制御レジスタ (WDTC)

インターバル時間の設定，ウォッチドッグタイマの起動，クリアの設定，およびリセット発生要因を示すレジスタです。

10.3 ウォッチドッグタイマ制御レジスタ (WDTC)

ウォッチドッグタイマ制御レジスタ (WDTC) について示します。

ウォッチドッグタイマ制御レジスタ (WDTC)

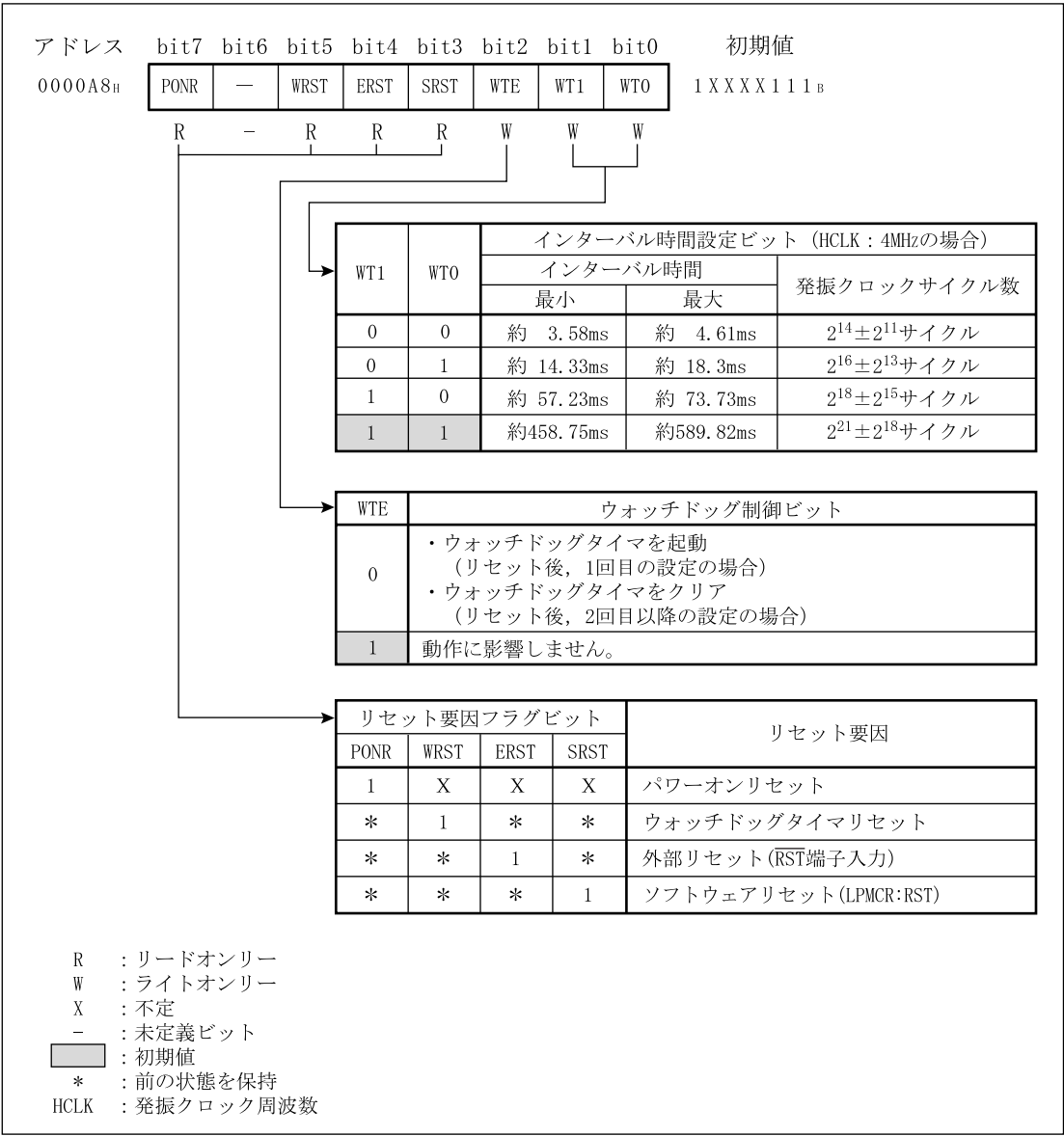


図 10.3-1 ウォッチドッグタイマ制御レジスタ (WDTC)

表 10.3-1 ウォッチドッグタイマ制御レジスタ (WDTC) の各ビットの機能説明

ビット名		機 能
bit7 bit5 bit4 bit3	PONR, WRST, ERST, SRST : リセット要因 フラグビット	<ul style="list-style-type: none"> ・リセット要因を示すフラグビットです。各リセット要因が発生した場合は, "1"がセットされます。 ・ウォッチドッグタイマ制御レジスタ (WDTC) を読出すことで, "0"にクリアされます。 ・パワーオンリセットの場合は, リセット要因フラグビット (PONR) に"1"がセットされ, リセット要因フラグビット (WRST, ERST, SRST) 不定値となります。 ・リセット要因フラグビット (PONR) に"1"がセットされている場合は, リセット要因フラグビット (WRST, ERST, SRST) の内容を無視してください。
bit6	- : 未定義ビット	<ul style="list-style-type: none"> ・読出しを行った場合は, 不定値となります。 ・設定した値は, 動作に影響しません。
bit2	WTE : ウォッチドッグ 制御ビット	<ul style="list-style-type: none"> ・"0"を設定した場合は, ウォッチドッグタイマを起動します。ただし, パワーオンリセット後, およびウォッチドッグタイマによるリセット後, 最初にレジスタアクセスを行った場合に有効です。 ・ウォッチドッグタイマ起動後に"0"を設定した場合は, ウォッチドッグタイマをクリアします。 ・"1"を設定した場合は, 動作に影響しません。
bit1 bit0	WT1, WT0 : インターバル時 間設定ビット	<ul style="list-style-type: none"> ・ウォッチドッグタイマのインターバル時間を設定するビットです。 ・ウォッチドッグタイマを起動する場合に, 設定されているデータが有効になります。ウォッチドッグ制御ビット (WTE) と同時に設定することができます。ウォッチドッグタイマを起動した後に設定したデータは, 無効です。 ・書込み専用ビットです。

10.4 ウォッチドッグタイマの動作説明

ウォッチドッグタイマは、ウォッチドッグタイマのオーバーフローで、ウォッチドッグタイマによるリセットが発生します。

ウォッチドッグタイマの動作

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

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0000A8H	PONR	—	WRST	ERST	SRST	WTE	WT1	WT0
						0	⊙	⊙

⊙ : 使用ビット
 0 : “0” を設定

図 10.4-1 ウォッチドッグタイマの設定

ウォッチドッグタイマの起動

- パワーオンリセット後、ウォッチドッグタイマによるリセット後、最初にウォッチドッグタイマ制御レジスタ(WDTC)のウォッチドッグ制御ビット(WTE)に"0"を設定することで、ウォッチドッグタイマを起動できます。ウォッチドッグタイマ制御レジスタ(WDTC)のインターバル時間設定ビット(WT1, WT0)でインターバル時間を設定します。
- ウォッチドッグタイマを起動した場合は、パワーオンリセット、ウォッチドッグタイマによるリセットで、ウォッチドッグタイマを停止できません。外部リセット、ソフトウェアリセット、ウォッチドッグタイマ制御レジスタ(WDTC)のウォッチドッグ制御ビット(WTE)への書込み、スリープモードへの遷移、およびタイムベースタイマモードへの遷移では、ウォッチドッグタイマを停止できません。

ウォッチドッグタイマのクリア

- ウォッチドッグタイマ起動後、ウォッチドッグタイマ制御レジスタ(WDTC)のウォッチドッグ制御ビット(WTE)に"0"を設定した場合、ウォッチドッグタイマをクリアできます。
- 外部リセットの入力設定、内部リセットの設定、およびスリープモードへの遷移を行った場合にも、ウォッチドッグタイマをクリアできます。
- タイムベースタイマモードへの遷移を行った場合は、ウォッチドッグタイマはクリアされ、一時停止します。

ウォッチドッグタイマのインターバル時間

図 10.4-2に、ウォッチドッグタイマをクリアするタイミングとインターバル時間の関係を示します。

リセット要因の確認

リセット後ウォッチドッグタイマ制御レジスタ(WDTC)のリセット要因フラグビット(PONR, WRST, ERST, SRST)を読み出した場合に、リセット要因が確認できます。

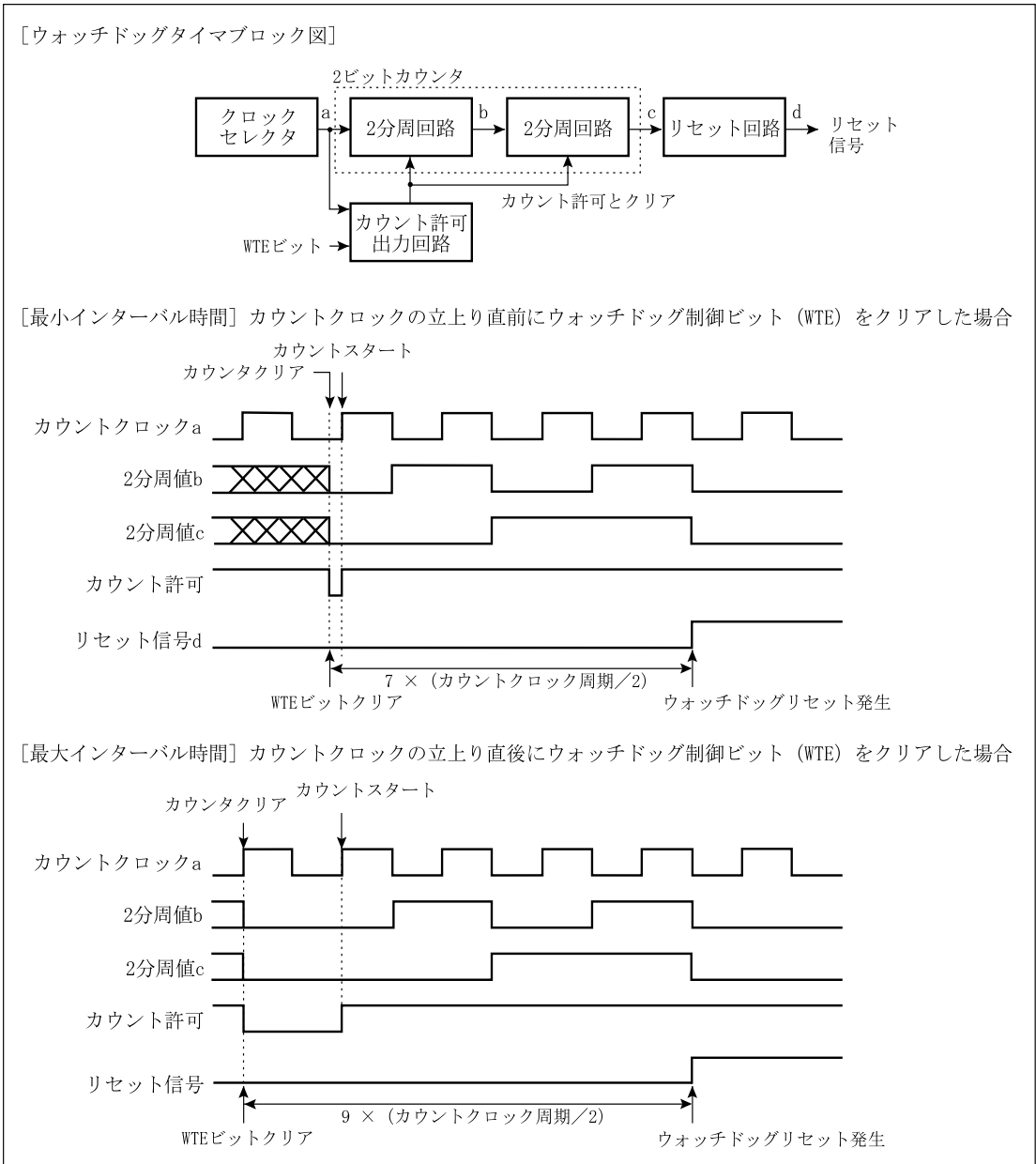


図 10.4-2 ウォッチドッグタイマのクリアタイミングとインターバル時間

10.5 ウォッチドッグタイマ使用上の注意

ウォッチドッグタイマ使用時の注意点を示します。

ウォッチドッグタイマ使用上の注意

ウォッチドッグタイマの停止

ウォッチドッグタイマを起動した場合は、パワーオンリセット、ウォッチドッグタイマによるリセットが発生するまで、ウォッチドッグタイマは停止しません。

インターバル時間の設定

ウォッチドッグタイマを起動した場合は、インターバル時間の設定が有効となります。ウォッチドッグタイマを起動しない場合は、インターバル時間の設定を無視します。

プログラム作成上の注意

プログラム中で繰り返しウォッチドッグタイマをクリアする場合、割込み処理を含めたプログラムの処理時間は、インターバル時間の最小時間以下となるようにしてください。

CPU停止中のウォッチドッグタイマ動作

CPU停止中は、ウォッチドッグタイマはクリアされ、ウォッチドッグタイマを一時停止します。タイムベースタイマモードから、メインクロックモード、またはPLLクロックモードへ復帰した場合は、再起動されます。

10.6 ウォッチドッグタイマのプログラム例

ウォッチドッグタイマのプログラム例を示します。

ウォッチドッグタイマのプログラム例

処理仕様

- メインプログラムのループの中で毎回ウォッチドッグタイマをクリアします。
- メインループは、ウォッチドッグタイマの最小インターバル時間内に1周する必要があります。

コーディング例

```

WDTC EQU 0000A8H ; ウォッチドッグタイマ制御レジスタ
WTE EQU WDTC:2 ; ウォッチドッグ制御ビット
;-----メインプログラム-----
CODE CSEG
START:
; ; ; スタックポインタ (SP) などは初期化済み
; ; ; とする

WDG_START:
MOV WDTC, #00000011B ; ウォッチドッグタイマの起動
; ; ; インターバル時間 $2^{21} \pm 2^{18}$ サイクルを設定
;-----メインループ-----
MAIN: CLRB I:WTE ; ウォッチドッグタイマのクリア
; ; ; 定期的に2ビットのクリア
; ; ; ユーザー処理
; ; ;
JMP MAIN ; ウォッチドッグタイマのインターバル時間
; ; ; より短い時間でループする

CODE ENDS
;-----ベクタ設定-----
VECT CSEG ABS=OFFH
ORG OFFDCH ; リセットベクタ設定
DSL START
DB 00H ; シングルチップモードに設定
VECT ENDS
END START

```

第11章 16ビットリロードタイマ

この章では、MB90560/565シリーズの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。

- 11.1 16ビットリロードタイマの概要
- 11.2 16ビットリロードタイマの構成
- 11.3 16ビットリロードタイマの端子
- 11.4 16ビットリロードタイマのレジスタ
- 11.5 16ビットリロードタイマの割込み
- 11.6 16ビットリロードタイマの動作説明
- 11.7 16ビットリロードタイマ使用上の注意
- 11.8 16ビットリロードタイマのプログラム例

11.1 16ビットリロードタイマの概要

MB90560/565シリーズは、16ビットリロードタイマを2チャンネル内蔵しており、以下の2種類のクロックモードと、2種類のカウンタ動作モードが設定できます。

クロックモード

- ・内部クロックモード：内部クロックに同期してカウントダウンを行うモード。
- ・イベントカウンタモード：外部入力パルスにてカウントダウンを行うモード。

カウンタ動作モード

- ・リロードモード：カウント設定値を再ロードしてカウントを繰り返すモード。
- ・ワンショットモード：アンダーフローでカウントを停止するモード。

16ビットリロードタイマの動作モード

表 11.1-1 16ビットリロードタイマの動作モード

クロックモード	カウンタ動作モード	動作モード
内部クロックモード	リロードモード	ソフトトリガ動作
	ワンショットモード	外部トリガ入力動作 外部ゲート入力動作
イベントカウンタモード (外部クロックモード)	リロードモード	ソフトトリガ動作
	ワンショットモード	

内部クロックモード

タイマコントロールステータスレジスタ (TMCSR) のカウントクロック設定ビット (CSL1, CSL0) に "00_B", "01_B", "10_B" を設定した場合は、内部クロックモードになります。

内部クロックモードは、以下の3種類の動作モードから動作を設定します。

ソフトトリガ動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に "1" を設定している場合に、ソフトウェアトリガビット (TRG) に "1" を設定すると、カウント動作を開始します。

外部トリガ入力動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に "1" を設定している場合に、動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) が TIN 端子に入力されると、カウント動作を開始します。

外部ゲート入力動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に "1" を設定している場合に、動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているゲート入力の有効レベル ("L", "H" から設定可能) が TIN 端子に入力されている間、カウント動作を行います。

イベントカウントモード（外部クロックモード）

タイマコントロールステータスレジスタ（TMCSR）のカウントクロック設定ビット（CSL1, CSL0）に"11_b"を設定した場合は、イベントカウントモード（外部クロック）になります。カウント許可ビット（CNTE）に"1"を設定している場合に、動作モード設定ビット（MOD2, MOD1, MOD0）で設定しているトリガ入力の有効エッジ（立上り、立下り、両エッジから設定可能）がTIN端子に入力されると、カウント動作を開始します。一定周期の外部クロックを入力する場合は、インターバルタイマとしても使用できます。

カウンタ動作

リロードモード

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、16ビットリロードレジスタ（TMRLR0/TMRHR0, TMRLR1/TMRHR1）の値を16ビットダウンカウンタにロードし、カウント動作を行います。また、アンダーフローで、割込み要求を出力しますので、インターバルタイマとして使用できます。アンダーフローごとに、反転するトグル波形をT0端子から出力できます。

表 11.1-2 16ビットリロードタイマのインターバル時間

カウントクロック	カウントクロック周期	インターバル時間
内部カウントクロック	2 ¹ / (0.125 μs)	0.125 μs ~ 8.192ms
	2 ³ / (0.5 μs)	0.5 μs ~ 32.768ms
	2 ⁵ / (2.0 μs)	2.0 μs ~ 131.1 ms
外部カウントクロック	2 ³ / (0.5 μs) 以上	0.5 μs 以上

：マシニングロック周波数
（ ）内はマシニングロック周波数16MHzの場合です。

ワンショットモード

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、カウント動作を停止します。

参考：

- ・16ビットリロードタイマ0は、UART0のボーレート作成に使用できます。
- ・16ビットリロードタイマ1は、UART1のボーレート作成、A/Dコンバータの起動トリガに使用できます。

16ビットリロードタイマの割込みとEI²OS

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で，割込み要求を出力します。

表 11.1-3 16ビットリロードタイマの割込みとEI²OS

チャンネル	割込み番号	割込み制御レジスタ		ベクタテーブルのアドレス			EI ² OS
		レジスタ名	アドレス	下位	上位	バンク	
16ビットリロードタイマ0 ^{*1}	#30(1E _H)	ICR09	0000B9 _H	FFFF84 _H	FFFF85 _H	FFFF86 _H	
16ビットリロードタイマ1 ^{*2}	#32(20 _H)	ICR10	0000BA _H	FFFF7C _H	FFFF7D _H	FFFF7E _H	

：使用可能

*1：16ビットリロードタイマ0は，8ビットタイマ0/1/2カウンタポローと同じ割込み制御レジスタ（ICR09）に割り当てられています。

*2：16ビットリロードタイマ1は，16ビットフリーランタイムオーバーフローと同じ割込み制御レジスタ（ICR10）に割り当てられています。

11.2 16ビットリロードタイマの構成

16ビットリロードタイマ0, 1は、それぞれ以下の7種類のブロックで構成されています。

- ・ カウントクロック生成回路
- ・ リロード制御回路
- ・ 出力制御回路
- ・ 動作制御回路
- ・ 16ビットタイマレジスタ (TMR0/TMR1)
- ・ 16ビットリロードレジスタ (TMRLR0/TMRLR1, TMRHR0/TMRHR1)
- ・ タイマコントロールステータスレジスタ (TMCSR0/TMCSR1:H, TMCSR0/TMCSR1:L)

16ビットリロードタイマのブロックダイアグラム

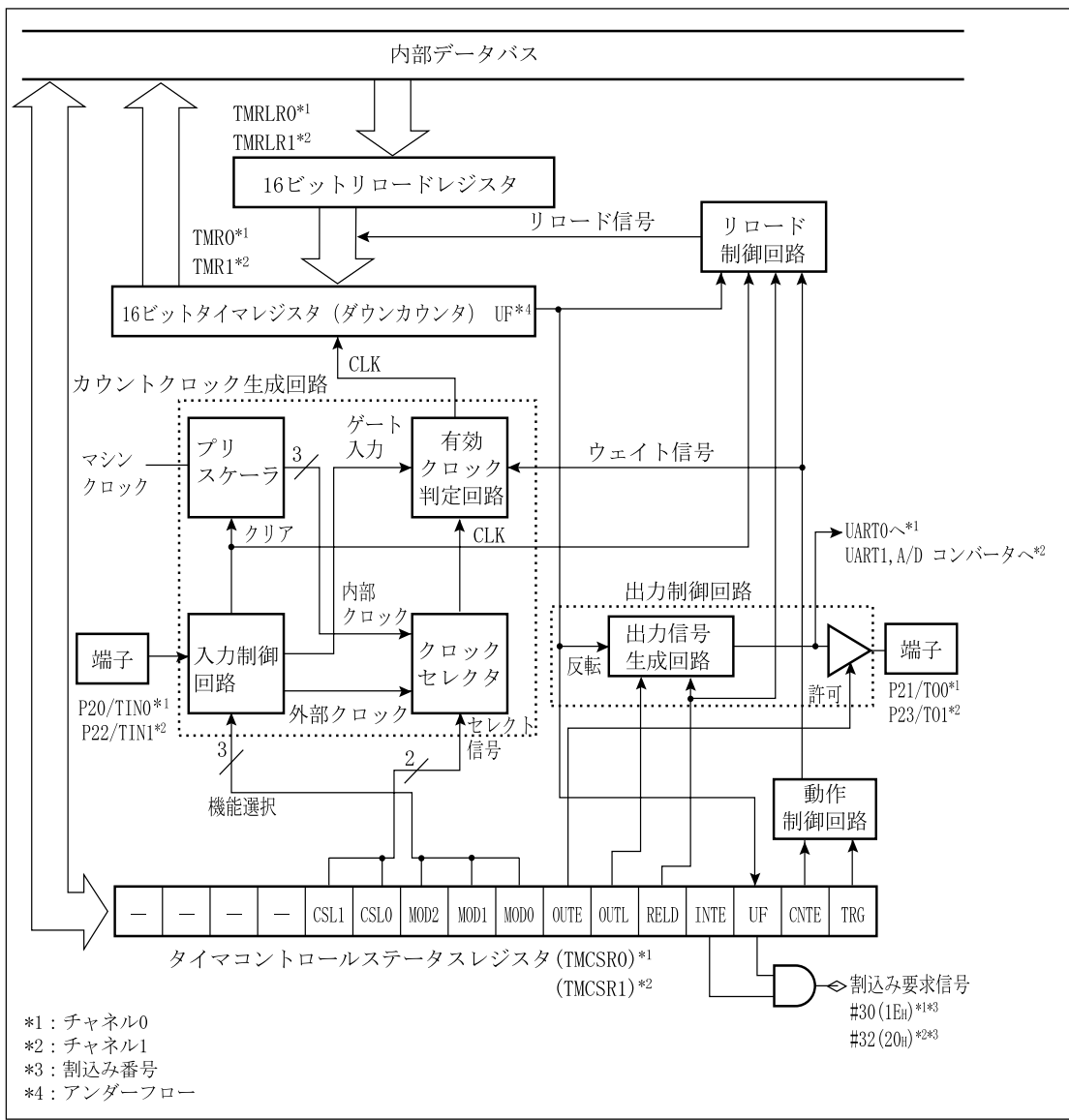


図 11.2-1 16ビットリロードタイマのブロックダイアグラム

カウントクロック生成回路

マシントクロック，または外部入力クロックを16ビットリロードタイマのカウントクロックとして使用します。

リロード制御回路

16ビットダウンカウンタの起動と，アンダーフロー（"0000_H" "FFFF_H"）で動作を制御します。

出力制御回路

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）によるT0端子出力の反転制御と，T0端子出力の許可／禁止を制御します。

動作制御回路

16ビットダウンカウンタの起動／停止を制御します。

16ビットタイマレジスタ（TMR0/TMR1）

16ビットダウンカウンタです。読み出しを行った場合は，カウンタの値が読み出せます。

16ビットリロードレジスタ（TMRLR0/TMRLR1, TMRHR0/TMRHR1）

16ビットダウンカウンタへのロード値を設定するレジスタです。16ビットリロードレジスタの設定値を16ビットダウンカウンタにロードし，ダウンカウントを行います。

タイマコントロールステータスレジスタ

（TMCSR0/TMCSR1:H, TMCSR0/TMCSR1:L）

16ビットリロードタイマの動作モードの設定，カウントクロックの設定，動作条件の設定，カウント動作の許可／禁止の設定，割込み制御，および割込み要求の状態を確認する機能があります

11.3 16ビットリロードタイマの端子

16ビットリロードタイマの端子，および端子部のブロックダイアグラムを示します。

16ビットリロードタイマの端子
16ビットリロードタイマの端子は，入出力ポートと兼用になっています。

表 11.3-1 16ビットリロードタイマの端子

端子名	端子機能	入出力形式	プルアップ 設定	スタンバイ 制御	端子の使用に 必要な設定
P20/TIN0	入出力ポート / タイマ0入力	CMOS出力 / CMOSヒステリ シス入力	なし	あり	入力ポートに設定 (DDR2 : bit0="0")
P21/T00	入出力ポート / タイマ0出力				タイマ0出力許可に設定 (TMCSR0 : OUTE="1")
P22/TIN1	入出力ポート / タイマ1入力				入力ポートに設定 (DDR2 : bit2="0")
P23/T01	入出力ポート / タイマ1出力				タイマ1出力許可に設定 (TMCSR1 : OUTE="1")

16ビットリロードタイマ端子部のブロックダイアグラム

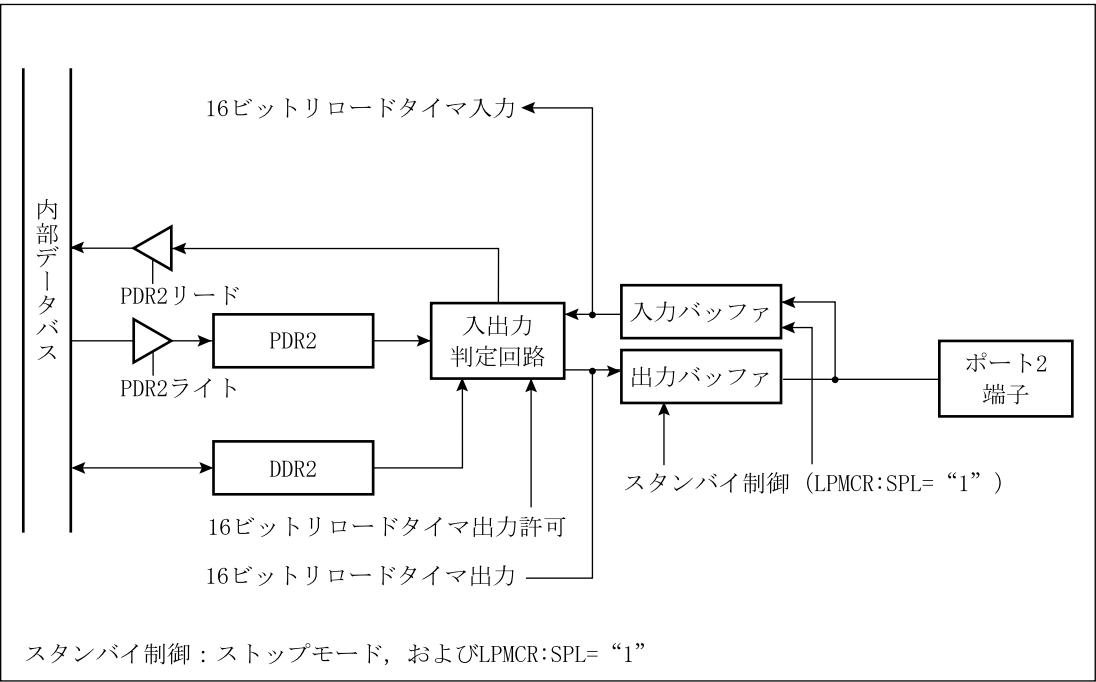


図 11.3-1 16ビットリロードタイマの端子部のブロックダイアグラム

11.4 16ビットリロードタイマのレジスタ

16ビットリロードタイマのレジスタ一覧を示します。

16ビットリロードタイマのレジスタ一覧

アドレス	bit15 bit8 bit7 bit0
ch0 : 000083 _H , 000082 _H	タイマコントロールステータスレジスタch0 (TMCSR0)
ch0 : 000085 _H , 000084 _H	16ビットタイマレジスタch0/16ビットリロードレジスタch0* (TMR0, TMRHR0/TMRLR0)
ch1:000087 _H , 000086 _H	タイマコントロールステータスレジスタch1 (TMCSR1)
ch1:000089 _H , 000088 _H	16ビットタイマレジスタch1/16ビットリロードレジスタch1* (TMR1, TMRHR1/TMRLR1)

* : リード時は16ビットタイマレジスタ (TMR) として機能し、ライト時は16ビットリロードレジスタ (TMRHR/TMRLR) として機能します。

図 11.4-1 16ビットリロードタイマのレジスタ一覧

11.4.1 タイマコントロールステータスレジスタ上位 (TMCSR0/TMCSR1 : H)

タイマコントロールステータスレジスタ (TMCSR0/TMCSR1) のbit11~bit7で、16ビットリロードタイマの動作モードの設定、カウントクロックの設定を行います。

タイマコントロールステータスレジスタ上位 (TMCSR0/TMCSR1 : H)

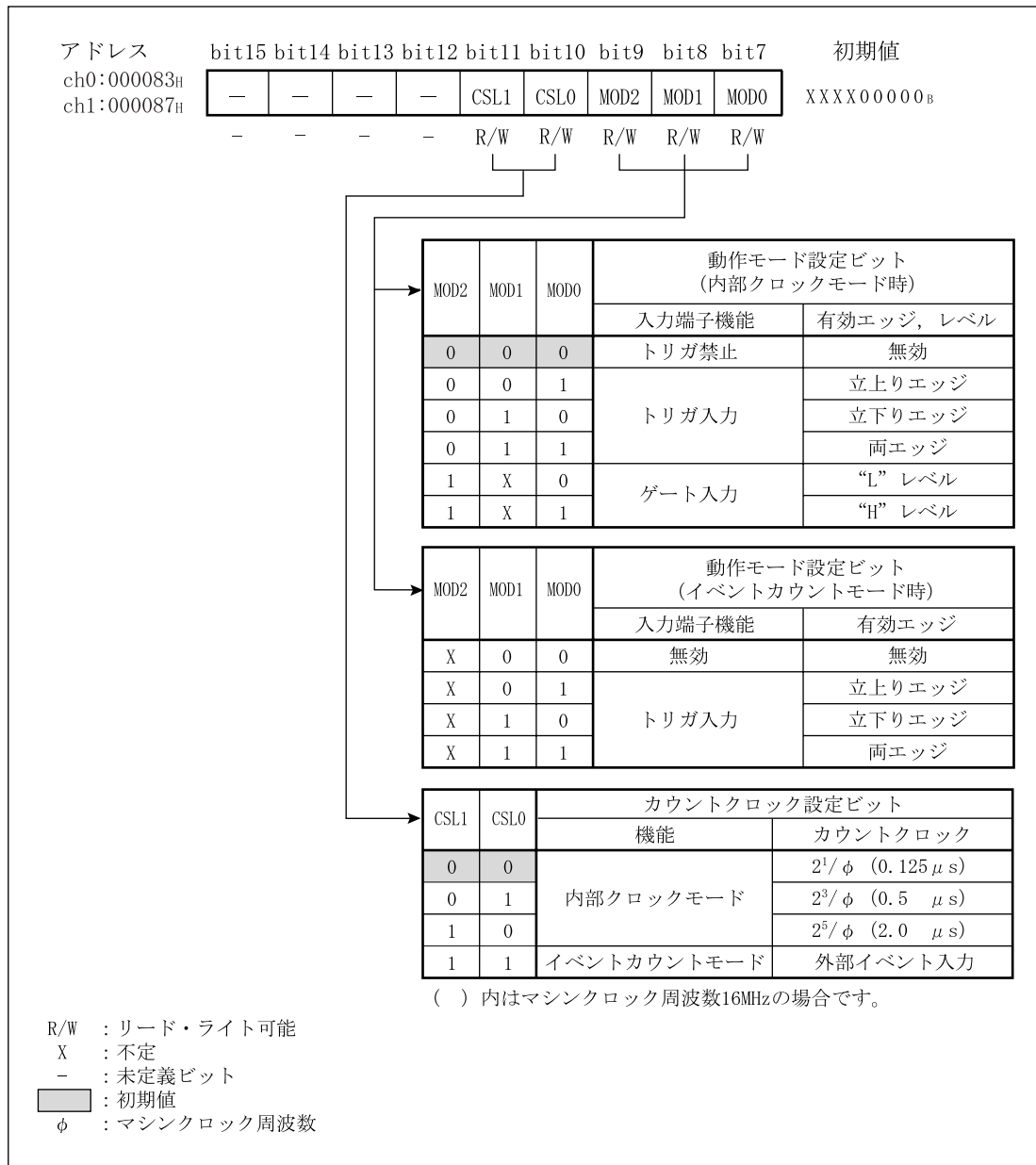


図 11.4-2 タイマコントロールステータスレジスタ上位 (TMCSR0/TMCSR1 : H)

表 11.4-1 タイマコントロールステータスレジスタ上位 (TMCSR0/TMCSR1 : H) の各ビットの機能説明

ビット名		機 能
bit15 bit14 bit13 bit12	- : 未定義ビット	<ul style="list-style-type: none"> ・ 読出しを行った場合は、不定値となります。 ・ 設定した値は、動作に影響しません。
bit11 bit10	CSL1, CSL0 : カウントクロック設定ビット	<ul style="list-style-type: none"> ・ 16ビットリロードタイマのカウントクロックを設定するビットです。 ・ "00_B", "01_B", "10_B"を設定した場合は、内部クロックモードとなります。 ・ "11_B"を設定した場合は、イベントカウントモードとなります。
bit9 bit8 bit7	MOD2, MOD1, MOD0 : 動作モード設定ビット	<ul style="list-style-type: none"> ・ 動作モードを設定するビットです。 <内部クロックモードを設定した場合> ・ MOD2ビットは、入力端子の機能を設定するビットです。 ・ MOD2ビットに"0"を設定した場合は、入力端子はトリガ入力端子となり、設定した有効エッジが入力されると、16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) の値を16ビットダウンカウンタにロードし、カウント動作を開始します。MOD1, MOD0ビットで、有効エッジを設定します。 ・ MOD2ビットに"1"を設定した場合は、入力端子はゲート入力となり、MOD0ビットで設定した有効レベルが入力されている間、カウント動作を行います。 ・ MOD1ビットに設定した値は、動作に影響しません。 <イベントカウントモードを設定した場合> ・ MOD2ビットに設定した値は、動作に影響しません。 ・ イベントカウントモードを設定した場合、入力端子はトリガ入力となり、MOD1, MOD0ビットで設定した有効エッジが入力されると、カウント動作を開始します。

11.4.2 タイマコントロールステータスレジスタ下位 (TMCSR0/TMCSR1 : L)

タイマコントロールステータスレジスタ (TMCSR0/TMCSR1) の下位7ビットで、16ビットリロードタイマの動作条件の設定、カウント動作の許可/禁止の設定、割り込み制御、および割り込み要求の状態を確認する機能があります。

タイマコントロールステータスレジスタ下位 (TMCSR0/TMCSR1 : L)

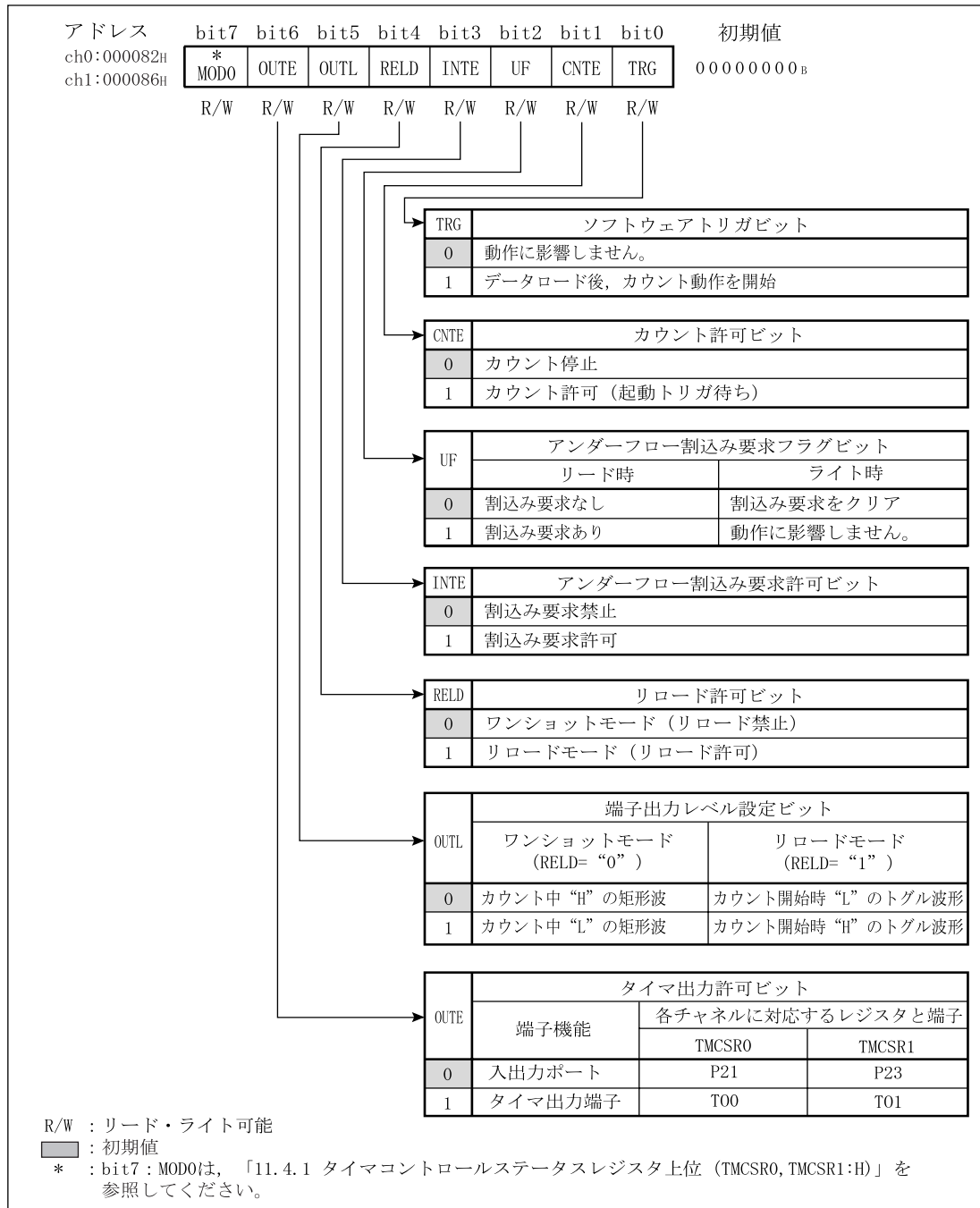


図 11.4-3 タイマコントロールステータスレジスタ下位 (TMCSR0/TMCSR1 : L)

表 11.4-2 タイマコントロールステータスレジスタ下位 (TMCSR0/TMCSR1:L) の各ビットの機能説明

ビット名		機 能
bit6	OUTE : タイマ出力許可 ビット	<ul style="list-style-type: none"> ・タイマ出力端子への出力を許可するビットです。 ・"0"を設定した場合は、端子は入出力ポートとなり、"1"を設定した場合は、端子はタイマ出力端子となります。
bit5	OUTL : 端子出力レベル 設定ビット	<ul style="list-style-type: none"> ・タイマ出力端子への出力レベルを設定するビットです。 ・タイマ出力端子の出力波形は、リロードモードを設定した場合は、トグル波形を出力し、ワンショットモードを設定した場合は、カウント動作中を示す矩形波を出力します。 ・"0"を設定した場合と"1"を設定した場合では、端子出力レベルが逆になります。
bit4	RELD : リロード許可 ビット	<ul style="list-style-type: none"> ・リロード動作を許可するビットです。 ・"1"を設定した場合は、リロードモードとなり、16ビットダウンカウンタのアンダーフローで、16ビットリロードレジスタに設定した値が、16ビットダウンカウンタにロードされ、カウント動作を継続します。 ・"0"を設定した場合は、ワンショットモードとなり、16ビットダウンカウンタのアンダーフローで、カウント動作を停止します。
bit3	INTE : アンダーフロー 割込み要求許可 ビット	<ul style="list-style-type: none"> ・割込み要求を許可するビットです。 ・"1"を設定した場合は、アンダーフロー割込み要求フラグビット (UF) に"1"がセットされると、割込み要求を出力します。
bit2	UF : アンダーフロー 割込み要求 フラグビット	<ul style="list-style-type: none"> ・割込み要求のフラグビットです。 ・16ビットダウンカウンタのアンダーフローで、"1"がセットされます。 ・アンダーフロー割込み要求許可ビット (INTE) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・"0"を設定した場合は、割込み要求がクリアされます。 ・"1"を設定した場合は、動作に影響しません。 ・EI²OSの起動で、"0"にクリアされます。
bit1	CNTE : カウント許可 ビット	<ul style="list-style-type: none"> ・カウント動作を許可するビットです。 ・"1"を設定した場合は、起動トリガ待ち状態となり、ソフトウェアトリガビット (TRG) に"1"を設定、または動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) がTIN端子に入力されると、カウント動作を開始します。 ・"0"を設定した場合は、カウント動作を停止します。
bit0	TRG : ソフトウェア トリガビット	<ul style="list-style-type: none"> ・インターバルタイマ機能、またはカウンタ機能をソフトウェアで起動するビットです。 ・カウント許可ビット (CNTE) に"1"を設定している場合に、"1"を設定すると、16ビットリロードレジスタに設定した値を16ビットダウンカウンタにロードされ、カウント動作を開始します。 ・"0"を設定した場合は、動作に影響しません。 ・読出し値は、"0"です。

11.4.3 16ビットタイマレジスタ (TMR0/TMR1)

16ビットタイマレジスタ (TMR0/TMR1) は、16ビットダウンカウンタのカウンタ値を読み出すことができます。

16ビットタイマレジスタ (TMR0/TMR1)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch0:000085H	D15	D14	D13	D12	D11	D10	D9	D8	X X X X X X X B
ch1:000089H	R	R	R	R	R	R	R	R	
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000084H	D7	D6	D5	D4	D3	D2	D1	D0	X X X X X X X B
ch1:000088H	R	R	R	R	R	R	R	R	
R : リードオンリー									
X : 不定									

図 11.4-4 16ビットタイマレジスタ (TMR0/TMR1)

16ビットダウンカウンタです。

タイマコントロールステータスレジスタ (TMCSR) のカウンタ許可ビット (CNTE) に"1"を設定している場合に、ソフトウェアトリガビット (TRG) に"1"を設定、または動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) がTIN端子に入力されると、16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) に設定した値が、16ビットダウンカウンタにロードされ、ダウンカウントを開始します。カウント停止状態 (TMCSR : CNTE="0") では、16ビットタイマレジスタ (TMR0/TMR1) の値を保持します。

< 注意事項 >

- ・ 16ビットタイマレジスタ (TMR0/TMR1) を読み出す場合は、必ずワード転送命令 (MOVW A, 003AH) を使用してください。
- ・ 16ビットタイマレジスタ (TMR0/TMR1) は、読出し専用のレジスタで、16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) は、書込み専用のレジスタですが、同一アドレスに配置されているため、16ビットタイマレジスタに対して値を設定した場合は、16ビットリロードレジスタに値が設定され、16ビットタイマレジスタは、影響を受けません。

11.4.4 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1)

16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) は、16ビットダウンカウンタへのロード値を設定するレジスタです。16ビットリロードレジスタに設定された値が、16ビットダウンカウンタにロードされ、ダウンカウンタ動作を行います。

16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1)

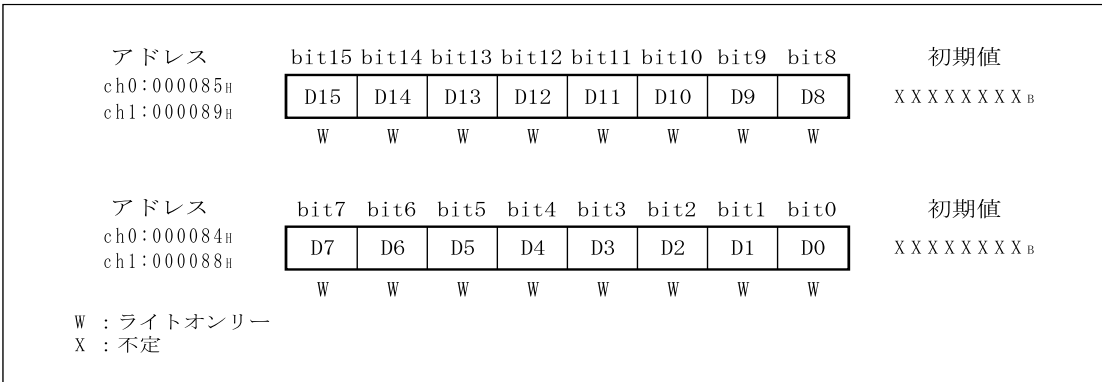


図 11.4-5 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1)

16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) に値を設定する場合は、16ビットリロードタイマの動作モードに関係なく、カウンタ動作を停止 (`TMCSR:CNT=0`) させてください。タイマコントロールステータスレジスタ (`TMCSR`) のカウンタ許可ビット (`CNT`) に"1"を設定している場合に、ソフトウェアトリガビット (`TRG`) に"1"を設定、または動作モード設定ビット (`MOD2, MOD1, MOD0`) で設定しているトリガ入力の有効エッジ (立上り、立下り、両エッジから設定可能) が `TIN` 端子に入力されると、16ビットリロードレジスタに設定した値が16ビットダウンカウンタにロードされ、ダウンカウンタを開始します。

16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) に設定した値は、リロードモードの場合は、16ビットダウンカウンタのアンダーフロー (`"0000H"` / `"FFFFH"`) で、16ビットリロードレジスタの値を16ビットダウンカウンタにロードし、ダウンカウンタを継続します。また、ワンショットモードの場合は、16ビットダウンカウンタのアンダーフローで、16ビットダウンカウンタの値は `"FFFFH"` で停止します。

< 注意事項 >

- 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) へ値を設定する場合は、カウンタ動作を停止 (`TMCSR:CNT=0`) させてください。
- 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) への設定は、ワード転送命令 (`MOVW 003AH, A`) を使用してください。
- 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) は、書込み専用のレジスタで、16ビットタイマレジスタ (`TMRO/TMR1`) は読み込み専用のレジスタですが、同一アドレスに配置されているため、書込み値と読み出し値は異なり、`INC/DEC` 命令などのリードモディファイライト (`RMW`) 動作をする命令は、使用できません。

11.5 16ビットリロードタイマの割込み

16ビットリロードタイマは、16ビットダウンカウンタのアンダーフローで割込み要求を出力します。拡張インテリジェントI/Oサービス (EI²OS) に対応しています。

16ビットリロードタイマの割込み

表 11.5-1 16ビットリロードタイマの割込み制御ビットと割込み要因

	16ビットリロードタイマ0	16ビットリロードタイマ1
アンダーフロー割込み 要求フラグビット	TMCSR0 : UF	TMCSR1 : UF
アンダーフロー割込み 要求許可ビット	TMCSR0 : INTE	TMCSR1 : INTE
割込み要因	16ビットダウンカウンタ (TMR0) のアンダーフロー	16ビットダウンカウンタ (TMR1) のアンダーフロー

16ビットリロードタイマでは、16ビットダウンカウンタのアンダーフロー ("0000_H" "FFFF_H") で、タイマコントロールステータスレジスタ (TMCSR) のアンダーフロー割込み要求フラグビット (UF) に"1"がセットされます。アンダーフロー割込み要求許可ビットが割込み要求を許可 (TMCSR : INTE="1") している場合に、割込み要求を出力します。

16ビットリロードタイマのEI²OS機能

16ビットリロードタイマは、16ビットダウンカウンタのアンダーフロー ("0000_H" "FFFF_H") で、拡張インテリジェントI/Oサービス (EI²OS) を使用できます。

11.6 16ビットリロードタイマの動作説明

16ビットリロードタイマの設定と，カウンタの動作状態について説明します。

16ビットリロードタイマの設定

内部クロックモードの設定

インターバルタイマとして動作させるには，図 11.6-1のように設定します。

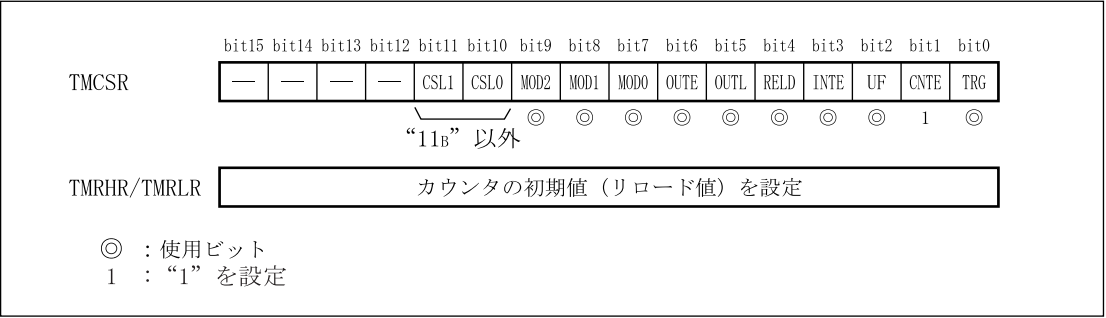


図 11.6-1 内部クロックモードの設定

イベントカウントモードの設定

イベントカウンタとして動作させるには，図 11.6-2のように設定します。

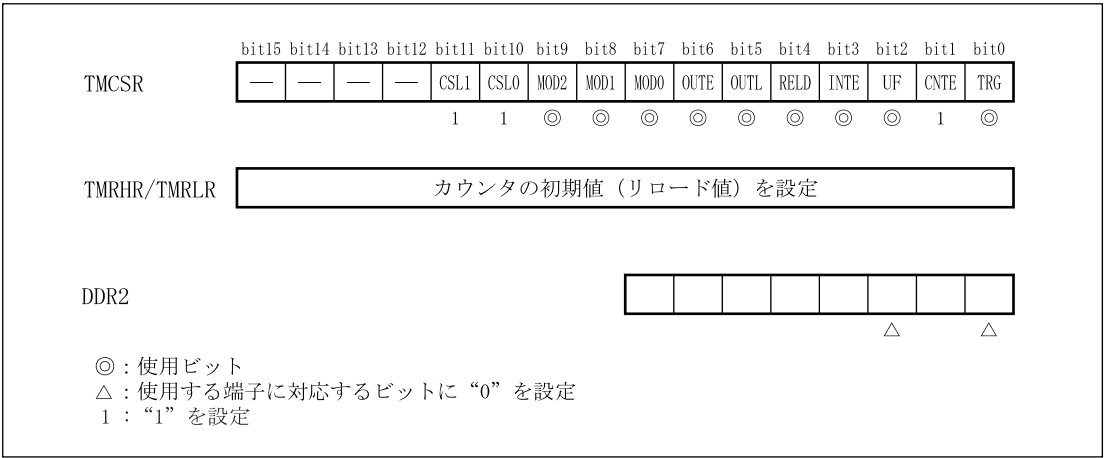


図 11.6-2 イベントカウンタモードの設定

カウンタの動作状態

16ビットダウンカウンタの状態は、タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) 値と内部信号の起動トリガ待ち信号値 (WAIT) で決まります。STOP状態 (停止状態), WAIT状態 (起動トリガ待ち状態), およびRUN状態 (動作状態) のカウント許可ビット (CNTE) 値と内部信号の起動トリガ待ち信号値 (WAIT) の関係を図 11.6-3に示します。

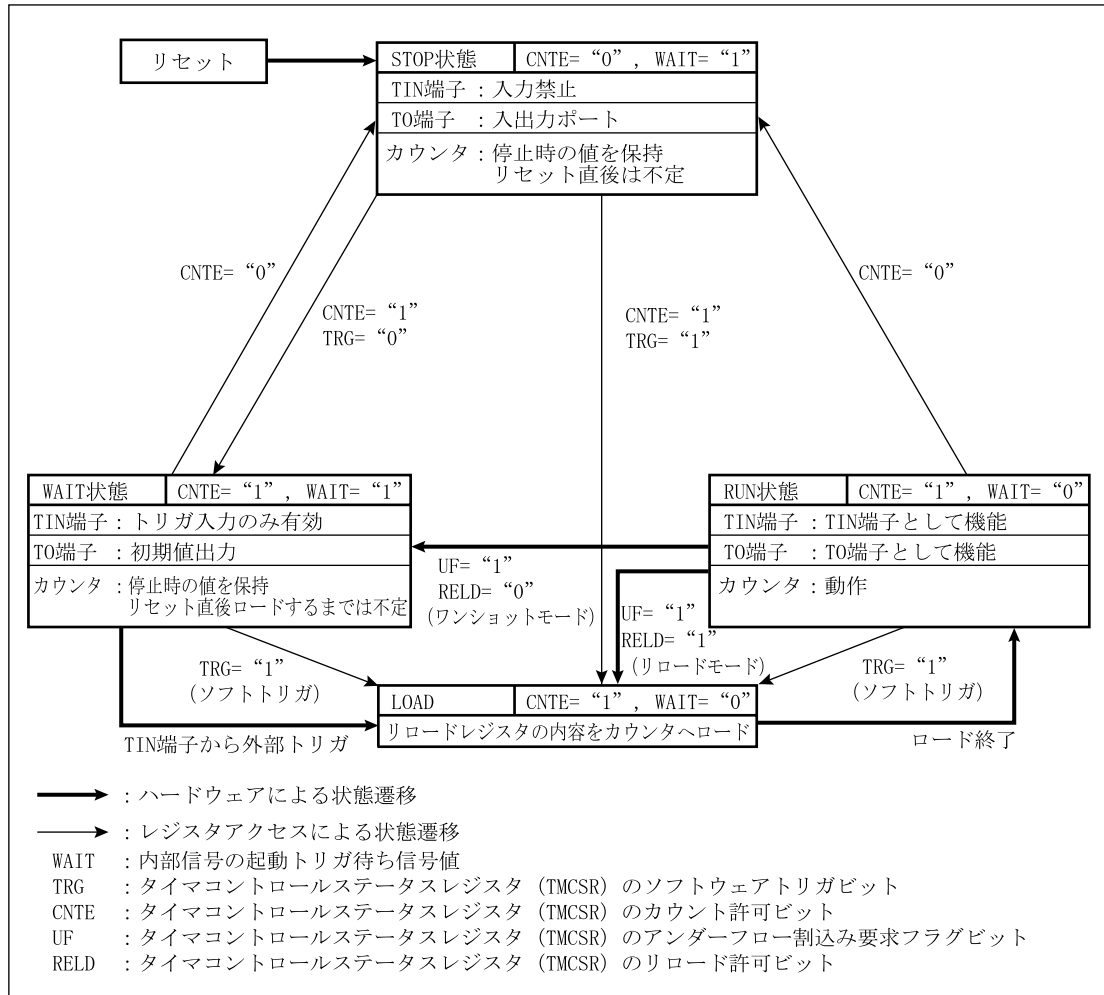


図 11.6-3 カウンタ状態遷移図

11.6.1 内部クロックモード（リロードモード）

内部カウントクロックに同期して、16ビットダウンカウンタをダウンカウントし、アンダーフロー（"0000_H" "FFFF_H"）で、割込み要求を出力します。また、タイマ出力端子からトグル波形を出力できます。

内部クロックモード（リロードモード）の動作

タイマコントロールステータスレジスタ（TMCSR）のカウンタ許可ビット（CNTE）に"1"を設定している場合に、ソフトウェアトリガビット（TRG）に"1"を設定、または動作モード設定ビット（MOD2, MOD1, MOD0）で設定しているトリガ入力の有効エッジ（立上り、立下り、両エッジから設定可能）がTIN端子に入力されると、16ビットリロードレジスタ（TMRLR0/TMRHR0, TMRLR1/TMRHR1）に設定した値が16ビットダウンカウンタにロードされ、ダウンカウントを開始します。カウンタ許可ビット（CNTE）とソフトウェアトリガビットに同時に"1"を設定した場合は、カウンタ動作を許可すると同時にダウンカウントを開始します。

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、16ビットリロードレジスタ（TMRLR0/TMRHR0, TMRLR1/TMRHR1）に設定した値が、16ビットダウンカウンタにロードされ、ダウンカウントを継続します。16ビットダウンカウンタのアンダーフローで、タイマコントロールステータスレジスタ（TMCSR）のアンダーフロー割込み要求フラグビット（UF）に"1"がセットされ、アンダーフロー割込み要求許可ビット（INTE）に"1"を設定している場合は、割込み要求を出力します。

T0端子からは、アンダーフローごとに反転するトグル波形が出力されます。

ソフトトリガ動作

タイマコントロールステータスレジスタ（TMCSR）のカウンタ許可ビット（CNTE）に"1"を設定している場合に、ソフトウェアトリガビット（TRG）に"1"を設定すると、カウンタ動作を開始します。

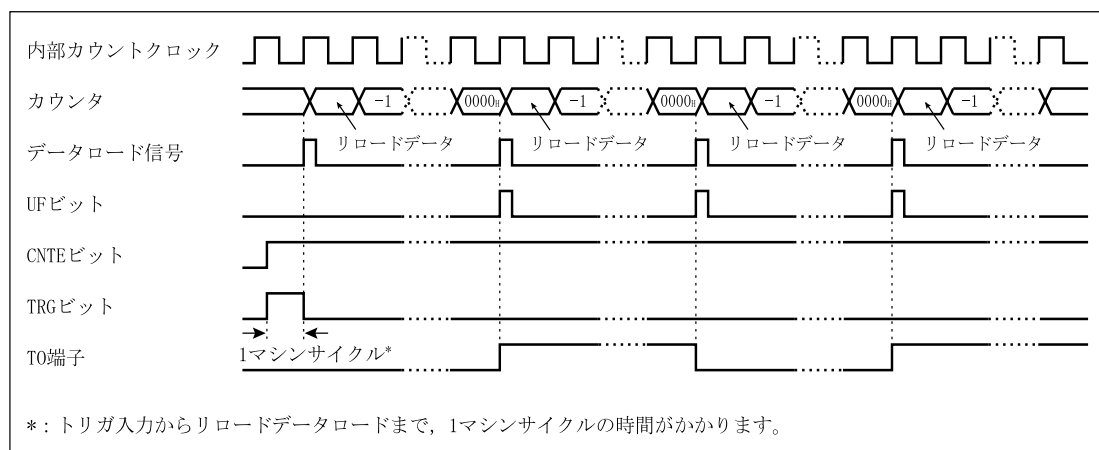


図 11.6-4 リロードモードのカウンタ動作（ソフトトリガ動作）

外部トリガ入力動作

タイマコントロールステータスレジスタ (TMC SR) のカウント許可ビット (CNTE) に "1" を設定している場合に、動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) が TIN 端子に入力されると、カウント動作を開始します。

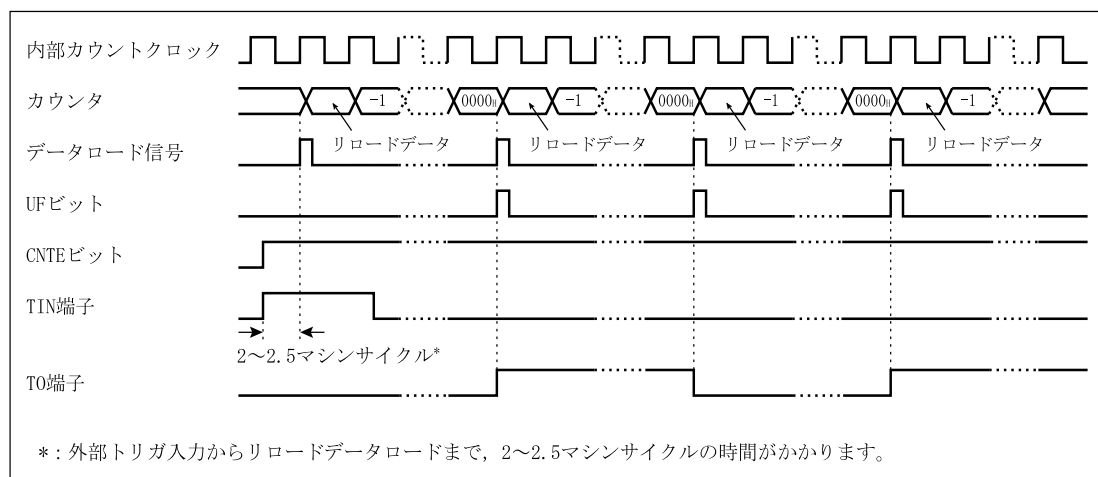


図 11.6-5 リロードモードのカウント動作（外部トリガ入力動作）

< 注意事項 >

TIN端子へ入力するトリガパルス幅は、 $2/$ （マシクロック周波数）以上としてください。

外部ゲート入力動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に"1"を設定している場合に、ソフトウェアトリガビット (TRG) に"1"を設定すると、カウント動作を開始します。

動作モード設定ビット (MOD2,MOD1,MOD0) で設定しているゲート入力の有効レベル ("L", "H" から設定可能) がTIN端子に入力されている間、カウント動作を行います。

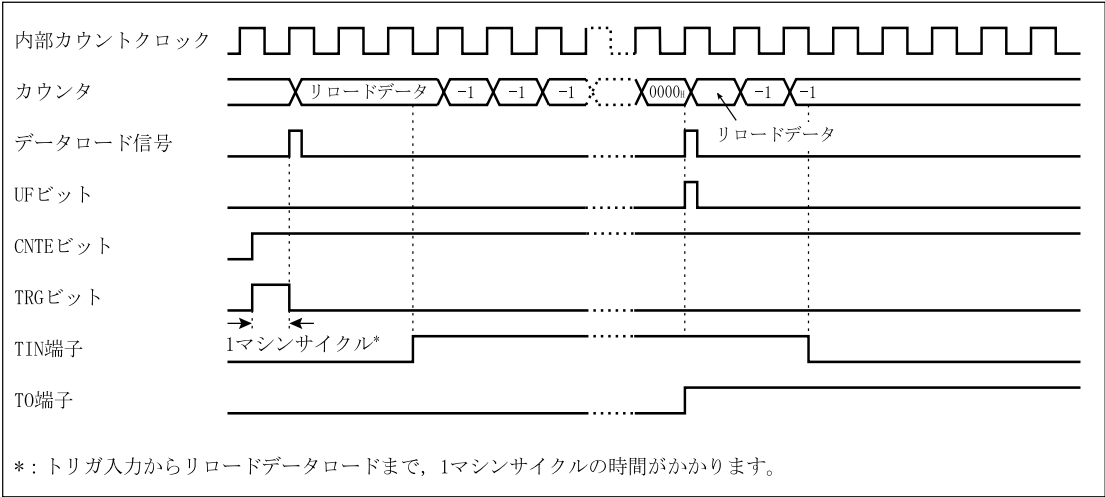


図 11.6-6 リロードモードのカウント動作 (外部ゲート入力動作)

< 注意事項 >

TIN端子へ入力するゲート入力のパルス幅は、 $2/f$ (f :マシクロック周波数) 以上としてください。

11.6.2 内部クロックモード (ワンショットモード)

内部カウントクロックに同期して、16ビットダウンカウンタをダウンカウントし、アンダーフロー ("0000_H" "FFFF_H") で、割込み要求を出力します。また、T0端子から、カウント中を示す矩形波を出力できます。

内部クロックモード (ワンショットモード)

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に"1"を設定している場合に、ソフトウェアトリガビット (TRG) に"1"を設定、または動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効エッジ (立上り, 立下り, 両エッジから設定可能) がTIN端子に入力されると、16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) に設定した値が16ビットダウンカウンタにロードされ、ダウンカウントを開始します。カウント許可ビット (CNTE) とソフトウェアトリガビット (TMCSR:TRG) に同時に"1"を設定した場合は、カウント動作を許可すると同時にダウンカウントを開始します。

16ビットダウンカウンタのアンダーフロー ("0000_H" "FFFF_H") で、16ビットダウンカウンタは"FFFF_H"の状態では、カウント動作を停止します。

16ビットダウンカウンタのアンダーフロー ("0000_H" "FFFF_H") で、タイマコントロールステータスレジスタ (TMCSR) のアンダーフロー割込み要求フラグビット (UF) に"1"がセットされ、アンダーフロー割込み要求許可ビット (INTE) に"1"を設定している場合は、割込み要求を出力します。

T0端子からは、カウント中を示す矩形波が出力できます。

ソフトトリガ動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に"1"を設定している場合に、ソフトウェアトリガビット (TRG) に"1"を設定すると、カウント動作を開始します。

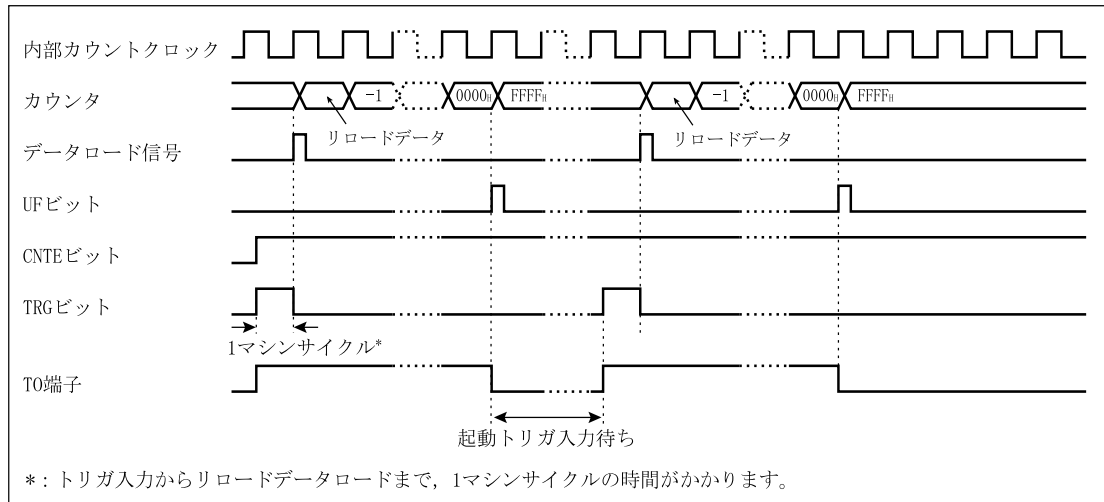


図 11.6-7 ワンショットモードのカウント動作 (ソフトトリガ動作)

外部トリガ入力動作

タイマコントロールステータスレジスタ (TMCSSR) のカウント許可ビット (CNTE) に "1" を設定している場合に , 動作モード設定ビット (MOD2 , MOD1 , MOD0) で設定しているトリガ入力の有効エッジ (立上り , 立下り , 両エッジから設定可能) が TIN 端子に入力されると , カウント動作を開始します。

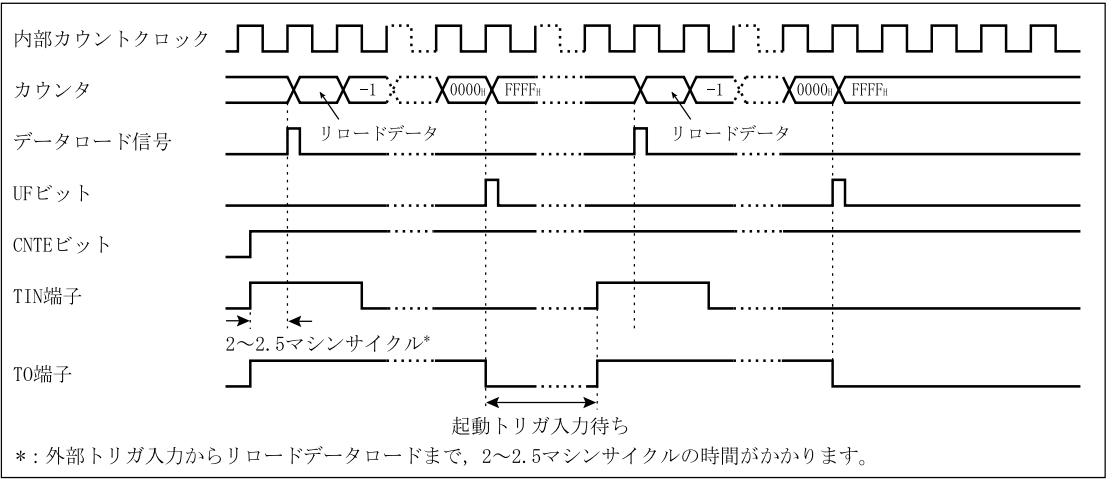


図 11.6-8 ワンショットモードのカウント動作 (外部トリガ入力動作)

< 注意事項 >

TIN端子へ入力するトリガパルス幅は , $2 / f_{clk}$ (f_{clk} : マシンクロック周波数) 以上としてください。

外部ゲート入力動作

タイマコントロールステータスレジスタ (TMCSR) のカウント許可ビット (CNTE) に"1"を設定している場合に、ソフトウェアトリガビット (TRG) に"1"を設定すると、カウント動作を開始します。

動作モード設定ビット (MOD2, MOD1, MOD0) で設定しているトリガ入力の有効レベル ("L", "H" から設定可能) がTIN端子に入力されている間、カウント動作を行います。

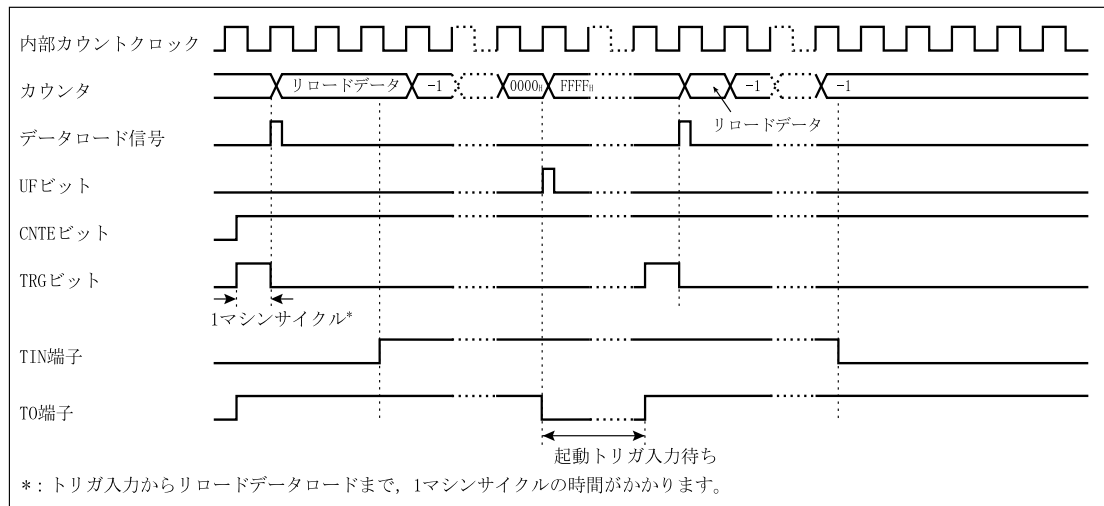


図 11.6-9 ワンショットモードのカウント動作 (外部ゲート入力動作)

< 注意事項 >

TIN端子へ入力するゲート入力のパルス幅は、 $2/$ (:マシクロック周波数) 以上としてください。

11.6.3 イベントカウントモード

TIN端子に入力されたパルスの有効エッジを検出することにより、16ビットダウンカウンタをダウンカウントし、アンダーフロー（"0000_H" "FFFF_H"）で、割込み要求を出力します。また、T0端子から、トグル波形、矩形波を出力できます。

イベントカウントモード

タイマコントロールステータスレジスタ（TMCSSR）のカウンタ許可ビット（CNTE）に"1"を設定している場合に、ソフトウェアトリガビット（TGR）に"1"を設定すると、16ビットリロードレジスタ（TMRLR0/TMRHR0, TMRLR1/TMRHR1）に設定した値が、16ビットダウンカウンタにロードされ、TIN端子に入力されたパルス（外部カウンタクロック）の有効エッジ（立上り、立下り、両エッジから設定可能）を検出することにより、ダウンカウントします。カウンタ許可ビット（CNTE）とソフトウェアトリガビット（TRG）に同時に"1"を設定した場合、カウンタ動作を許可すると同時に、ダウンカウントを開始します。

リロードモードの動作

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、16ビットリロードレジスタ（TMRLR0/TMRHR0, TMRLR1/TMRHR1）に設定した値が16ビットダウンカウンタにロードされ、ダウンカウント動作を継続します。

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、タイマコントロールステータスレジスタ（TMCSSR）のアンダーフロー割込み要求フラグビット（UF）に"1"がセットされ、アンダーフロー割込み要求許可ビット（INTE）に"1"を設定している場合は、割込み要求を出力します。

T0端子からは、アンダーフローごとに反転するトグル波形が出力できます。

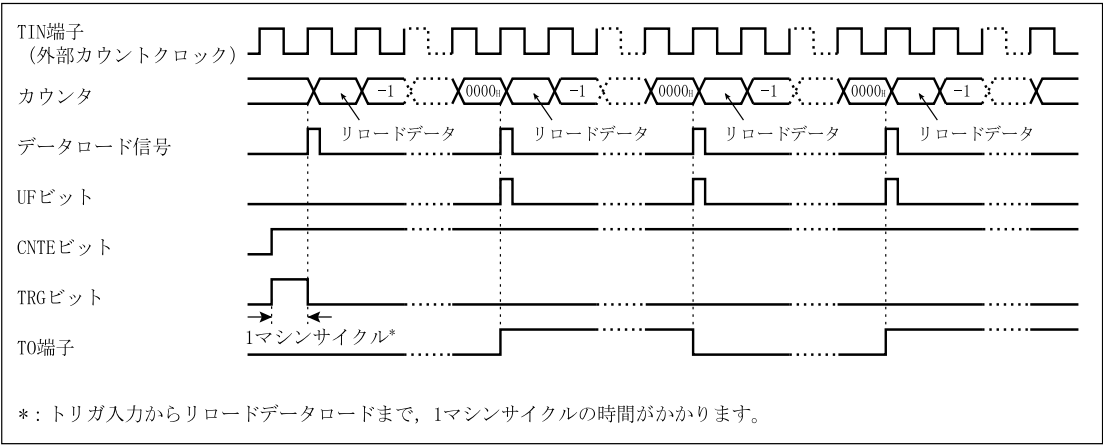


図 11.6-10 リロードモードのカウンタ動作（イベントカウントモード）

< 注意事項 >

TIN端子へ入力されるパルスの"H"幅、および"L"幅は、 $4/\text{（マシンのクロック周波数）}$ 以上としてください。

ワンショットモードの動作

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、16ビットダウンカウンタの値は、"FFFF_H"の状態にリセットされ、カウント動作を停止します。

16ビットダウンカウンタのアンダーフロー（"0000_H" "FFFF_H"）で、タイマコントロールステータスレジスタ(TMCSSR)のアンダーフロー要求フラグビット（UF）に"1"がセットされ、割り込み要求許可ビット（INTE）に"1"を設定している場合は、割り込み要求を出力します。

T0端子からは、カウント中を示す矩形波が出力できます。

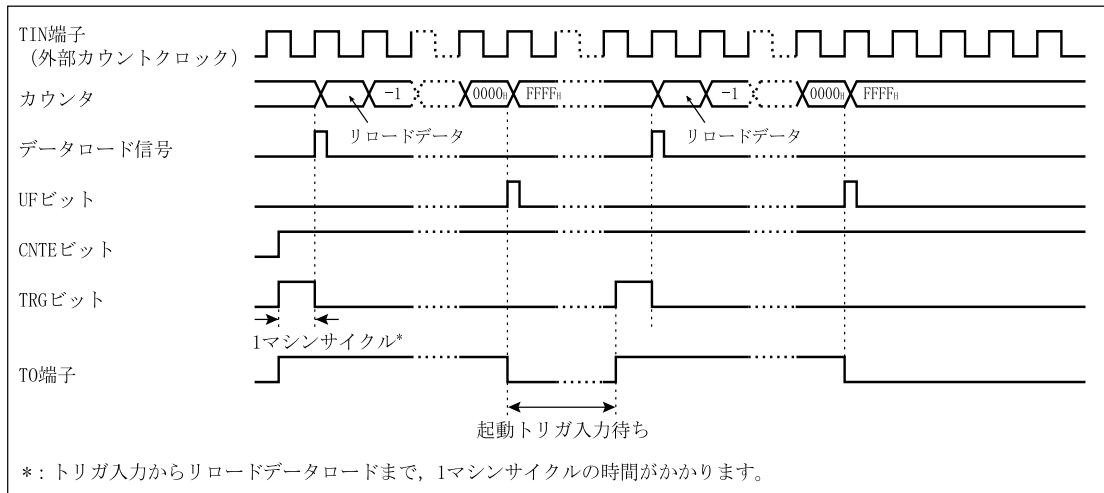


図 11.6-11 ワンショットモードのカウンタ動作（イベントカウントモード）

< 注意事項 >

TIN端子へ入力されるパルスの"H"幅、および"L"幅は、 $4/$ （ :マシクロック周波数）以上としてください。

11.7 16ビットリロードタイマ使用上の注意

16ビットリロードタイマを使用する場合の注意点を示します。

16ビットリロードタイマ使用上の注意

プログラムで設定する場合の注意

- 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) に値を設定する場合は、カウント動作を停止 (TMCSR:CNTE="0") させてください。16ビットタイマレジスタ (TMR0/TMR1) を読み出す場合は、必ずワード転送命令 (MOVW A,dir) を使用してください。
- タイマコントロールステータスレジスタ (TMCSR) のカウントクロック設定ビット (CSL1, CSL0) を変更する場合は、カウント動作を停止 (TMCSR:CNTE="0") させてください。

割込みに関連する注意

- タイマコントロールステータスレジスタ (TMCSR) のアンダーフロー割込み要求フラグビット (UF) に"1"がセットされ、アンダーフロー割込み要求許可ビット (INTE) に"1"を設定している場合は、割込み処理から復帰できません。アンダーフロー割込み要求フラグビット (UF) は、必ず"0"にクリアしてください。
- 16ビットリロードタイマ0は、8ビットタイマ0/1/2カウンタボローと、また、16ビットリロードタイマ1は16ビットフリーランタイムオーバーフローと割込み制御レジスタを共有しているため、同時に同じレベルの割込みを出力した場合は、割込みベクタ番号の小さい割込みを優先します。

11.8 16ビットリロードタイマのプログラム例

16ビットリロードタイマの内部クロックモードとイベントカウントモードのプログラム例を示します。

内部クロックモードのプログラム例

処理仕様

- 16ビットリロードタイマ0で、25msのインターバルタイマ割込みを出力します。
- リロードモードで使用し、繰り返し割込みを出力します。
- 外部トリガ入力を使用せず、ソフトウェアトリガで、16ビットリロードタイマを起動します。
- EI²OSは使用しません。
- マシンクロック周波数16MHz、カウントクロック2 μ sとします。

コーディング例

```

ICR09 EQU 0000B9H ; 16ビットリロードタイマ用割込み制御レジスタ
TMCSR EQU 000082H ; タイマコントロールステータスレジスタ
TMR EQU 000084H ; 16ビットタイマレジスタ
TMRLR EQU 000084H ; 16ビットリロードレジスタ
UF EQU TMCSR:2 ; アンダーフロー割込み要求フラグビット
CNTE EQU TMCSR:1 ; カウント許可ビット
TRG EQU TMCSR:0 ; ソフトウェアトリガビット
;-----メインプログラム-----
CODE CSEG
START:
; ; ; スタックポインタ (SP) などは初期化済み
; ; ; とする
AND CCR, #0BFH ; 割込みディセーブル
MOV I:ICR09, #00H ; 割込みレベル0 (最強)
CLRB I:CNTE ; カウンタを一時停止
MOVW I:TMRLR, #30D4H ; 25msタイマのデータを設定
MOVW I:TMCSR, #00001000000011011B ; インターバルタイマ動作, クロック2 $\mu$ s
; ; ; 外部トリガ禁止, 外部出力禁止,
; ; ; リロードモード設定, 割込み許可
; ; ; 割込みフラグクリア, カウンタスタート
MOV ILM, #07H ; PS内ILMレジスタをレベル7に設定
OR CCR, #40H ; 割込みイネーブル
LOOP: MOV A, #00H ; 無限ループ
MOV A, #01H ;
BRA LOOP ;
;-----割込みプログラム-----
WARI:
CLRB I:UF ; アンダーフロー割込み要求フラグをクリア
; ;
; ユーザ処理

```

```

;      :
;      RETI                      ; 割込みから復帰

CODE   ENDS
;-----ベクタ設定-----
VECT   CSEG      ABS=OFFH
      ORG        OFF88H          ; 割込み#29 (1DH) にベクタを設定
      DSL        WARI
      ORG        OFFDCH          ; リセットベクタ設定
      DSL        START
      DB         00H             ; シングルチップモードに設定
VECT   ENDS
      END          START

```

イベントカウントモードのプログラム例

処理仕様

- 16ビットリロードタイマ/カウンタ0で、外部イベント入力端子に入力される、パルスの立上りエッジを10,000回カウントすると割込みを出力します。
- ワンショットモードで動作させます。
- 外部トリガ入力は、立上りエッジを設定します。
- EI²OSは使用しません。

コーディング例

```

ICR09 EQU      0000B9H          ; 16ビットリロードタイマ用割込み制御レジ
;                               ; スタ
TMCSR EQU      000082H          ; タイマコントロールステータスレジスタ
TMR EQU      000084H           ; 16ビットタイマレジスタ
TMRLR EQU     000084H           ; 16ビットリロードレジスタ
DDR2 EQU      000012H          ; ポート方向レジスタ
UF EQU      TMCSR:2             ; アンダーフロー割込み要求フラグビット
CNTE EQU      TMCSR:1           ; カウント許可ビット
TRG EQU      TMCSR:0           ; ソフトウェアトリガビット
;-----メインプログラム-----
CODE   CSEG
START:
;      :                      ; スタックポインタ (SP) などは初期化済み
;      :                      ; とする
      AND        CCR, #0BFH     ; 割込みディセーブル
      MOV        I:ICR09, #00H  ; 割込みレベル0 (最強)
      MOV        I:DDR2, #00H   ; P20/TIN0 端子を入力に設定
      CLRB       I:CNTE         ; カウンタを一時停止
      MOVW       I:TMRLR, #2710H ; リロード値10000回の設定
      MOVW       I:TMCSR, #0000110010001011B
;                               ; カウンタ動作, 外部トリガ, 立上りエッジ,
;                               ; 外部出力禁止
;                               ; ワンショットモード設定, 割込み許可
;                               ; 割込みフラグクリア, カウンタスタート
      MOV        ILM, #07H      ; PS内 ILMレジスタをレベル7に設定
      OR         CCR, #40H      ; 割込みイネーブル
LOOP:   MOV      A, #00H        ; 無限ループ
      MOV      A, #01H        ;

```

```

        BRA      LOOP          ;
;-----割り込みプログラム-----
WARI:
        CLRB     I:UF          ; アンダーフロー割り込み要求フラグをクリア
        ;
        ; ユーザ処理
        ;
        RETI      ; 割り込みから復帰

CODE    ENDS
;-----ベクタ設定-----
VECT    CSEG      ABS=OFFH
        ORG       OFF84H      ; 割り込み #30 (1EH) にベクタを設定
        DSL       WARI
        ORG       OFFDCH      ; リセットベクタ設定
        DSL       START
        DB        00H         ; シングルチップモードに設定
        VECT      ENDS
        END        START

```


第12章 多機能タイマ

この章では，MB90560/565シリーズの多機能タイマの動作について説明します。

- 12.1 多機能タイマの概要
- 12.2 多機能タイマの構成
- 12.3 多機能タイマのレジスター一覧
- 12.4 多機能タイマの動作説明

12.1 多機能タイマの概要

多機能タイマは、16ビットフリーランタイマをベースに12チャンネルの独立した波形出力、入力パルス幅測定、および外部クロック周期測定ができます。

多機能タイマの機能

16ビットフリーランタイマ (1チャンネル)

16ビットフリーランタイマは16ビットアップカウンタ (タイマデータレジスタ (TCDT)), コンペアクリアレジスタ (CPCLR), タイマコントロールステータスレジスタ (TCCS), およびプリスケラで構成されています。

16ビットフリーランタイマのカウント出力値は、アウトプットコンペア、およびインプットキャプチャの基本時間 (ベースタイマ) として使用されます。

- カウントクロックは、次の8種類から設定できます。
 $1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128$
 : マシンクロック周波数
- 16ビットフリーランタイマのカウント値がオーバーフローした場合、または16ビットフリーランタイマのカウント値とコンペアクリアレジスタ (CPCLR) 値が一致 (TCCS:ICRE="1", MODE="1") し、16ビットフリーランタイマのカウント値が"0000H"にクリアされた場合に割込みを出力させることができます。
- 16ビットフリーランタイマのカウント値は、リセット、タイマコントロールステータスレジスタ (TCCS) のクリアビット (SCLR) に"1"を設定、16ビットフリーランタイマのカウント値とコンペアクリアレジスタ (CPCLR) 値の一致 (TCCS:MODE="1"), およびタイマデータレジスタ (TCDT) へ"0000H"を設定した場合、"0000H"にクリアできます。

アウトプットコンペア (6チャンネル)

アウトプットコンペアは、コンペアレジスタ (OCCP0 ~ OCCP5), コンペアコントロールレジスタ (OCS0 ~ OCS5), およびコンペア出力用ラッチで構成されています。

コンペアレジスタ (OCCP0 ~ OCCP5) 値と16ビットフリーランタイマのカウント値が一致した場合に出力レベルを反転し、割込みを出力させることができます。

- コンペアレジスタ (OCCP0 ~ OCCP5) は、6チャンネル独立して動作できます。各チャンネルのコンペアレジスタ (OCCP0 ~ OCCP5) は、対応した出力端子を、各チャンネルのコンペアコントロールレジスタ下位 (OCS0, OCS2, OCS4) は、割込み要求フラグを持っています。
- コンペアレジスタ (OCCP0 ~ OCCP5) を2チャンネル使用し、端子出力を反転させることができます。
- 16ビットフリーランタイマのカウント値とコンペアレジスタ (OCCP0 ~ OCCP5) 値の一致 (OCS0, OCS2, OCS4: IOP0="1", IOP1="1") により割込みを出力させることができます。(OCS0, OCS2, OCS4: IOE0="1", IOE1="1")
- 各チャンネルの端子出力の初期値を設定することができます。

インプットキャプチャ（4チャンネル）

インプットキャプチャは、外部入力端子（IN0～IN3）と対応したインプットキャプチャデータレジスタ（IPCP0～IPCP3）、およびインプットキャプチャコントロールレジスタ（ICS01, ICS23）で構成されています。

外部入力端子から入力された信号の有効エッジを検出することで、16ビットフリーランタイムのカウント値をインプットキャプチャデータレジスタ（IPCP0～IPCP3）に取り込み、割込みを出力させることができます。

- インプットキャプチャは、各チャンネル独立して動作できます。
- 外部信号の有効エッジ（立上りエッジ、立下りエッジ、両エッジ）が設定できます。
- 外部信号の有効エッジを検出することにより、割込みを出力させることができます。（ICS01, ICS23: ICE0="1", ICE1="1", ICE2="1", ICE3="1"）

8/16ビットPPGタイマ（8ビット：6チャンネル、16ビット：3チャンネル）

8/16ビットPPGタイマは、8ビットダウンカウンタ（PCNT）、PPGコントロールレジスタ（PPGC0～PPGC5）、PPGクロック制御レジスタ（PCS01, PCS23, PCS45）、PPGリロードレジスタ（PRL0～PRL5, PRLH0～PRLH5）で構成されています。

8/16ビットのリロードタイマとして使用する場合は、イベントタイマとして動作します。また、任意周波数、および任意デューティ比のパルスを出力させることができます。

- 8ビットPPGモード
 - 各チャンネル独立して8ビットPPGとして動作します。
- 8ビットプリスケラ + 8ビットPPGモード
 - ch0（ch2, ch4）を8ビットプリスケラとして動作させ、ch1（ch3, ch5）をch0（ch2, ch4）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として動作します。
- 16ビットPPGモード
 - ch0（ch2, ch4）とch1（ch3, ch5）を連結させ、16ビットPPGとして動作します。
- PPG動作
 - 任意周波数、および任意デューティ比（パルス波形の"H"レベル期間と"L"レベル期間の比）のパルスを出力します。外付け回路により、D/Aコンバータとしても使用できます。

波形生成部

波形生成部は、8ビットタイマ、8ビットタイマコントロールレジスタ（DTCR0～DTCR2）、8ビットリロードレジスタ（TMRRO～TMRR2）、波形制御レジスタ（SIGCR）で構成されています。

リアルタイムアウトプット（RT0～RT5）と8/16ビットPPGタイマを使用することにより、インバータ制御に用いるノンオーバーラップ三相波形出力、DCチョッパ出力ができます。

- 8ビットタイマをデッドタイムタイマとして使用することにより、PPGタイマのパルス出力にノンオーバーラップ時間のディレイを付けた、ノンオーバーラップ波形を生成することができます。（デッドタイムタイマ機能）
- 8ビットタイマをデッドタイムタイマとして使用することにより、リアルタイムアウトプット（RT1,RT3,RT5）に、ノンオーバーラップ時間のディレイを付けた、ノンオーバーラップ波形を生成することができます。（デッドタイムタイマ機能）
- 16ビットフリーランタイマのカウンタ値とアウトプットコンペアのコンペアレジスタ（OCCP0～OCCP5）値の一致（リアルタイムアウトプット（RT）の立上りエッジ）により、GATE信号を生成しPPGタイマの動作を制御することができます。（GATE機能）
- 16ビットフリーランタイマのカウンタ値とアウトプットコンペアのコンペアレジスタ（OCCP0～OCCP5）値の一致（リアルタイムアウトプット（RT）の立上りエッジ）で、8ビットタイマを起動することにより、GATE信号を生成しPPGタイマの動作を制御することができます。（GATE機能）
- DTTI端子入力により、RT00～RT05端子出力を制御することができます。DTTI端子入力のクロックレス化により、発振クロックが停止している場合でも、外部から端子制御ができます。（端子レベルは、各端子ごとにプログラムで設定できます）。ただし、事前に入出力ポート（P30～P35）を出力に設定し、ポート3データレジスタ（PDR3）に出力値を設定する必要があります。

12.2 多機能タイマの構成

多機能タイマユニットは、以下の5種類のブロックで構成されています。

- ・ 16ビットフリーランタイム (1チャンネル)
- ・ アウトプットコンペア (6チャンネル)
- ・ インプットキャプチャ (4チャンネル)
- ・ 8/16ビットPPGタイマ(8ビット：6チャンネル，16ビット：3チャンネル)
- ・ 波形生成部

リアルタイムI/O部のブロックダイアグラム

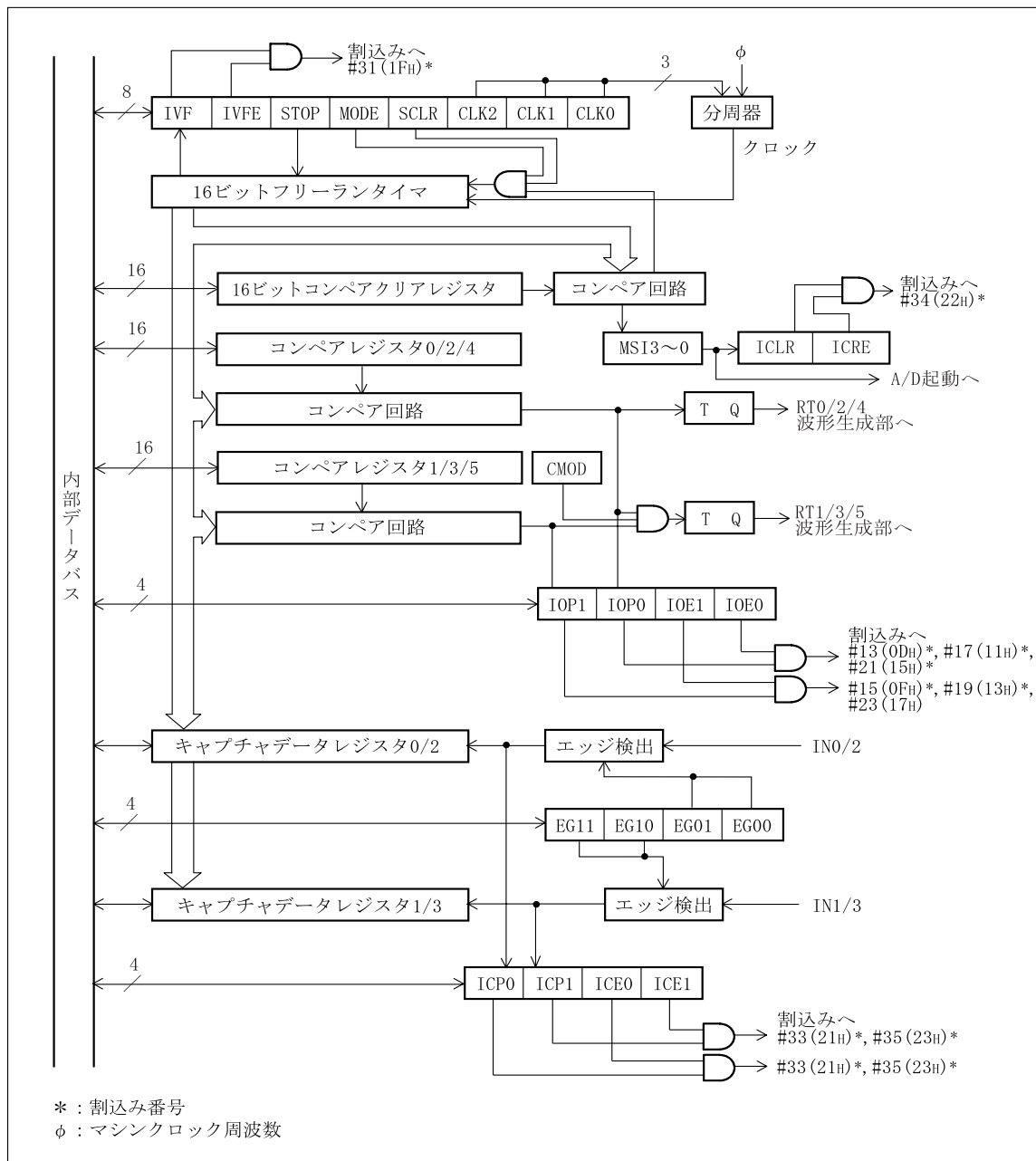


図 12.2-1 リアルタイムI/O部のブロックダイアグラム

8/16ビットPPGタイマのブロックダイヤグラム

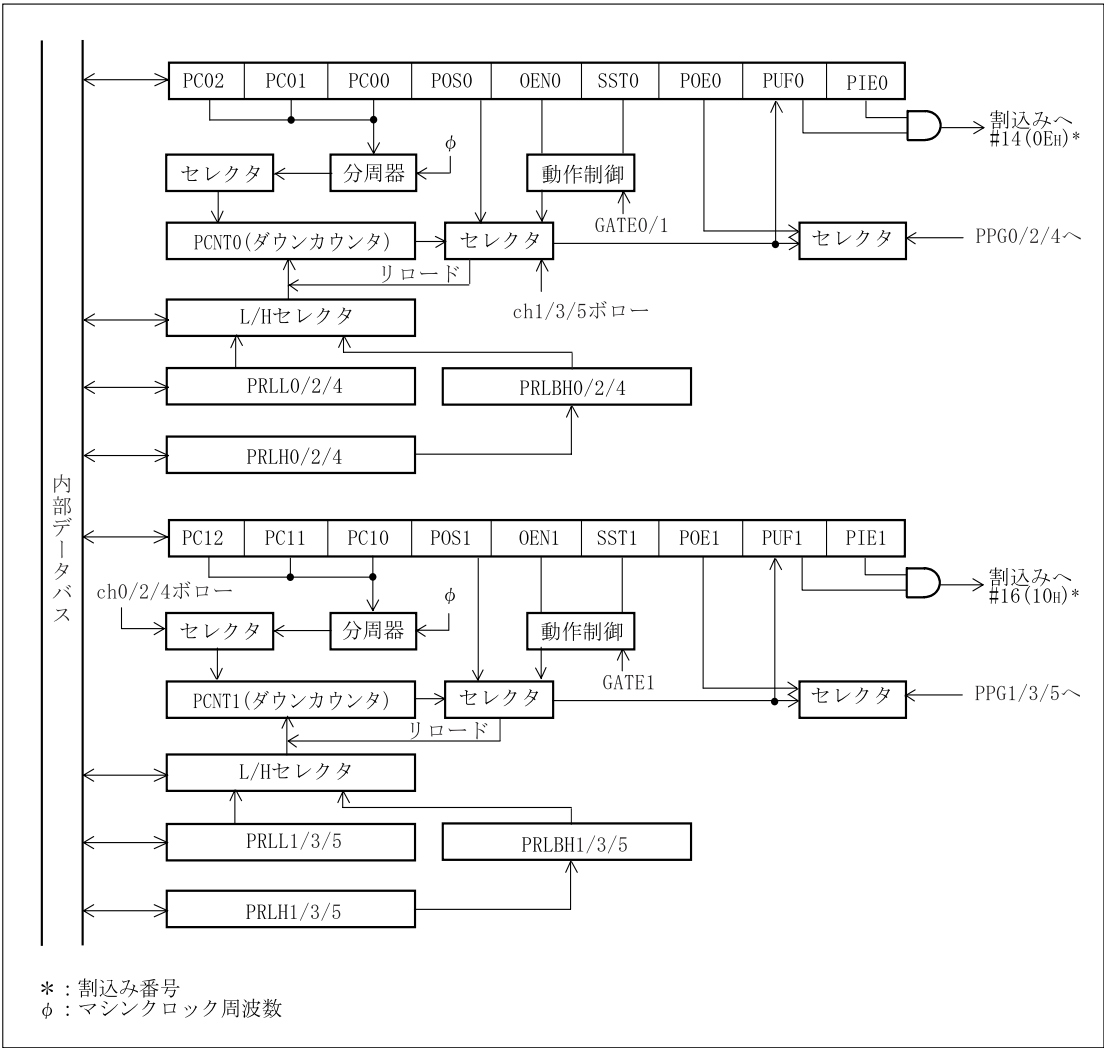


図 12.2-2 8/16ビットPPGタイマ部のブロックダイヤグラム

波形生成部のブロックダイアグラム

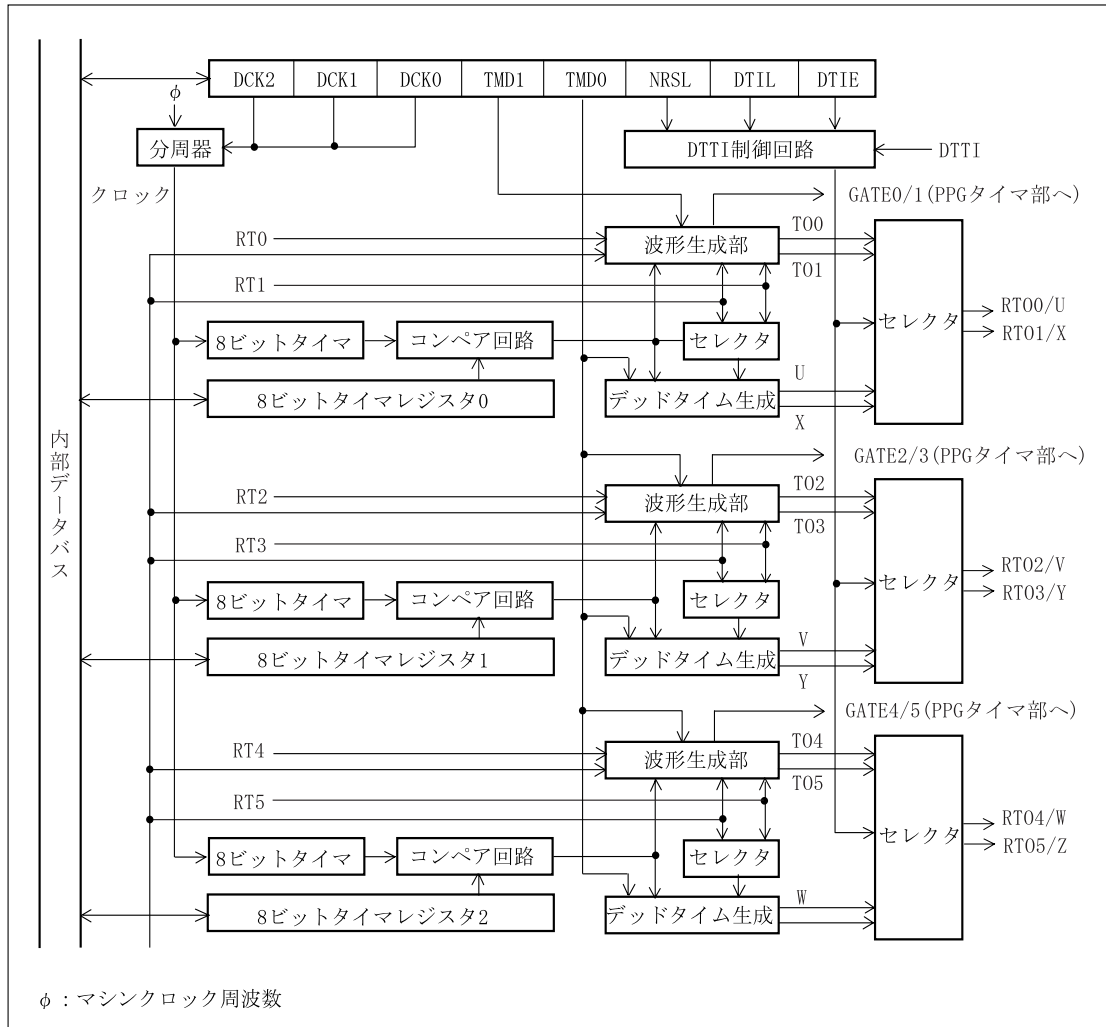


図 12.2-3 波形生成部のブロックダイアグラム

12.3 多機能タイマのレジスタ一覧

多機能タイマのレジスタ一覧を示します。

16ビットフリーランタイマのレジスタ一覧

アドレス	bit15	bit8	bit7	bit0
00005B _H , 00005A _H	タイマデータレジスタ (TCDT)			
000059 _H , 000058 _H	コンペアクリアレジスタ (CPCLR)			
00005D _H , 00005C _H	タイマコントロールステータスレジスタ (TCCS)			

図 12.3-1 16ビットフリーランタイマのレジスタ一覧

アウトプットコンペアのレジスタ一覧

アドレス	bit15	bit8	bit7	bit0
ch0:000071 _H , 000070 _H ch1:000073 _H , 000072 _H ch2:000075 _H , 000074 _H ch3:000077 _H , 000076 _H ch4:000079 _H , 000078 _H ch5:00007B _H , 00007A _H	コンペアレジスタ (OCCP0～OCCP5)			
ch1:00007D _H , ch0:00007C _H ch3:00007F _H , ch2:00007E _H ch5:000081 _H , ch4:000080 _H	コンペアコントロールレジスタ上位 (OCS1, OCS3, OCS5)		コンペアコントロールレジスタ下位 (OCS0, OCS2, OCS4)	

図 12.3-2 アウトプットコンペアのレジスタ一覧

インプットキャプチャのレジスタ一覧

アドレス	bit15	bit8	bit7	bit0
ch0:000061 _H , 000060 _H ch1:000063 _H , 000062 _H ch2:000065 _H , 000064 _H ch3:000067 _H , 000066 _H	インプットキャプチャデータレジスタ (IPCP0～IPCP3)			
000068 _H	使用禁止		インプットキャプチャコントロールレジスタ01 (ICS01)	
00006A _H			インプットキャプチャコントロールレジスタ23 (ICS23)	

図 12.3-3 インプットキャプチャのレジスタ一覧

8/16ビットPPGタイマのレジスター一覧

アドレス	bit15	bit8	bit7	bit0
ch1:00003D _H , ch0:00003C _H ch3:000045 _H , ch2:000044 _H ch5:00004D _H , ch4:00004C _H	PPGコントロールレジスタ上位 (PPGC1, PPGC3, PPGC5)		PPGコントロールレジスタ下位 (PPGC0, PPGC2, PPGC4)	
ch0, ch1:00003E _H ch2, ch3:000046 _H ch4, ch5:00004E _H	使用禁止		PPGクロック制御レジスタ (PCS01, PCS23, PCS45)	
ch0:000039 _H , 000038 _H ch1:00003B _H , 00003A _H ch2:000041 _H , 000040 _H ch3:000043 _H , 000042 _H ch4:000049 _H , 000048 _H ch5:00004B _H , 00004A _H	PPGリロードレジスタ上位 (PRLH0~PRLH5)		PPGリロードレジスタ下位 (PRL0~PRL5)	

図 12.3-4 8/16ビットPPGタイマのレジスター一覧

波形生成部のレジスター一覧

アドレス	bit15	bit8	bit7	bit0
ch0:000051 _H , 000050 _H ch1:000053 _H , 000052 _H ch3:000055 _H , 000054 _H	8ビットタイマコントロールレジスタ (DTCR0, DTCR1, DTCR2)		8ビットリロードレジスタ (TMRRO, TMRR1, TMRR2)	
000056 _H	使用禁止		波形制御レジスタ (SIGCR)	

図 12.3-5 波形生成部のレジスター一覧

12.3.1 16ビットフリーランタイマのレジスタ

- 16ビットフリーランタイマのレジスタには、以下の3種類のレジスタがあります。
- ・タイマデータレジスタ (TCDT)
 - ・コンペアクリアレジスタ (CPCLR)
 - ・タイマコントロールステータスレジスタ (TCCS)

タイマデータレジスタ (TCDT)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00005BH	T15	T14	T13	T12	T11	T10	T09	T08	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00005AH	T07	T06	T05	T04	T03	T02	T01	T00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W：リード・ライト可能

図 12.3-6 タイマデータレジスタ (TCDT)

16ビットフリーランタイマのカウンタです。カウンタ値は、リセットで"0000H"にクリアされます。タイマデータレジスタ (TCDT) にカウンタ値を設定する場合は、カウンタを停止 (TCCS:STOP="1")させてください。タイマデータレジスタ (TCDT) は、ワードアクセスしてください。

16ビットフリーランタイマのカウンタ値は、次の条件で"0000H"にクリアされます。

- リセット
- タイマコントロールステータスレジスタ (TCCS) のクリアビット (SCLR) に"1"を設定
- 16ビットフリーランタイマのカウンタ値のオーバーフロー
- 16ビットフリーランタイマのカウンタ値とコンペアクリアレジスタ (CPCLR) 値の一致 (TCCS:MODE="1")
- タイマデータレジスタ (TCDT) へ"0000H"を設定

コンペアクリアレジスタ (CPCLR)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000059 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	X X X X X X X B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000058 _H	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	X X X X X X X B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード・ライト可能									
X : 不定									

図 12.3-7 コンペアクリアレジスタ(CPCLR)

16ビットフリーランタイマのカウンタ値と比較する値を設定する，16ビット長のレジスタです。初期値は不定ですので，割込み動作を許可する場合，およびカウンタ値を"0000_H"にクリアする場合は，コンペアクリアレジスタ (CPCLR) 値を設定してください。コンペアクリアレジスタ (CPCLR) は，ワードアクセスしてください。

タイマコントロールステータスレジスタ上位 (TCCS)

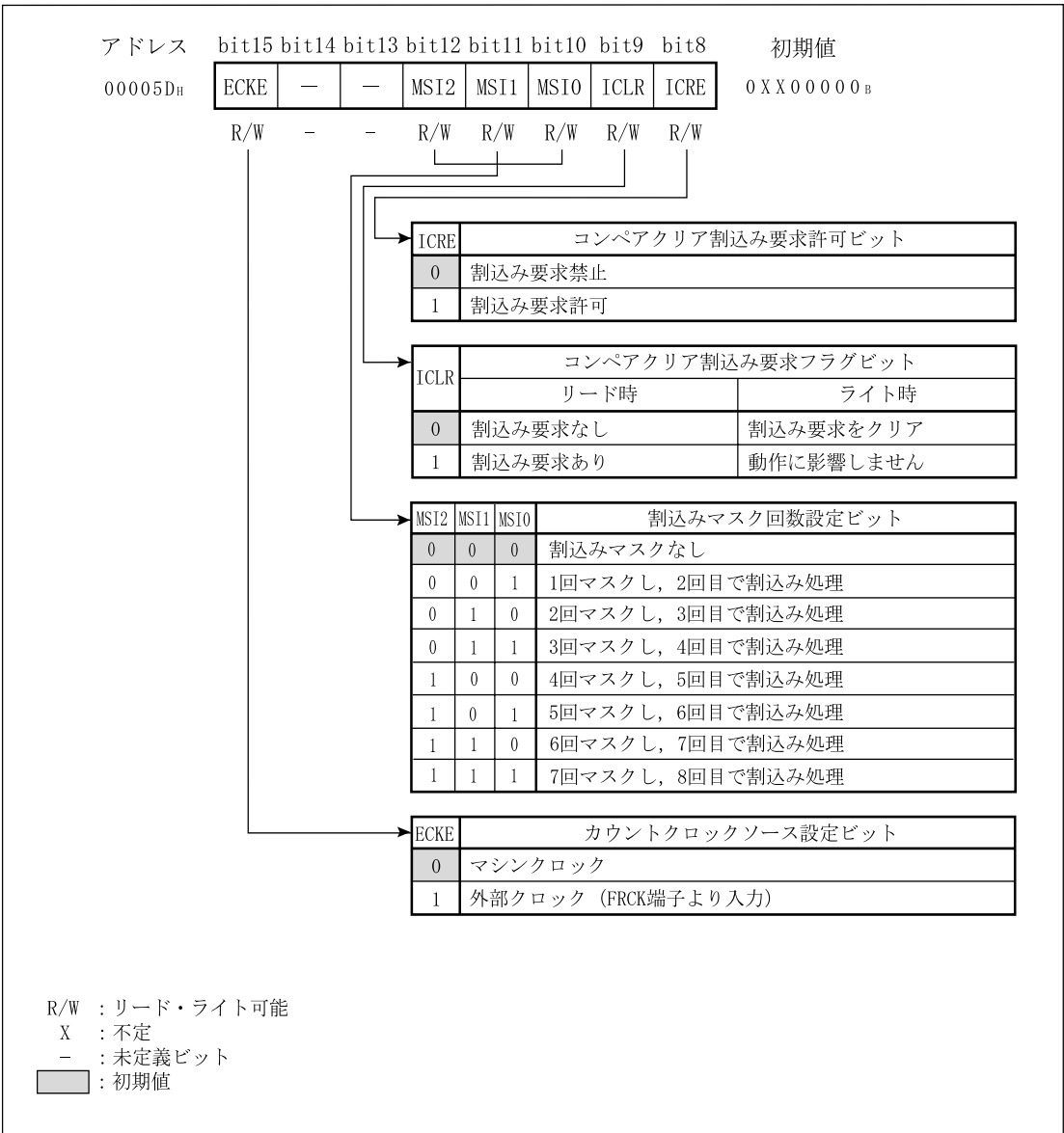


図 12.3-8 タイマコントロールステータスレジスタ上位 (TCCS)

表 12.3-1 タイマコントロールステータレジスタ上位 (TCCS) の各ビットの機能説明

ビット名		機 能
bit15	ECKE : カウントク ロックソース 設定ビット	16ビットフリーランタイムのカウントクロックとして、マシニングロック、または外部クロックのどちらを使用するかを設定するビットです。 カウントクロックソースは、設定後すぐに変更されますので、アウトプットコンペア、およびインプットキャプチャが停止状態で設定してください。 <注記> マシニングロックを設定する場合は、カウントクロック設定ビット (CLK2~CLK0) を設定してください。 外部クロックを設定する場合は、FRCK端子を入力ポートに設定 (DDR1: bit15="0") してください。
bit14 bit13	- : 未定義ビット	読出しを行った場合は、不定値となります。 設定した値は、動作に影響しません。
bit12 bit11 bit10	MSI2, MSI1, MSI0 : 割込みマスク 回数設定ビット	コンペアクリア割込みのマスク回数を設定するビットです。 設定値がそのままマスク回数となります。 <例> ・ "010 _b " を設定した場合は、2回マスクし、3回目で割込みを出力させます。 ・ "000 _b " を設定した場合は、割込み要因のマスクは行われません。
bit9	ICLR : コンペアクリア 割込み要求 フラグビット	割込み要求のフラグビットです。 16ビットフリーランタイムのカウンタ値とコンペアクリアレジスタ (CPCLR) 値が一致し、カウンタ値が "0000 _h " にクリア (bit4: MODE="1") された場合に、"1" がセットされます。 コンペアクリア割込み許可ビット (bit8: ICRE) に "1" を設定している場合に、"1" がセットされると、割込みを出力します。 "0" を設定した場合は、割込み要求がクリアされます。 "1" を設定した場合は、動作に影響しません。 リードモディファイライト系命令では、"1" が読み出されます。
bit8	ICRE : コンペアクリア 割込み要求許可 ビット	割込み要求を許可するビットです。 "1" を設定した場合は、コンペアクリア割込み要求フラグビット (bit9: ICLR) に、"1" がセットされると割込み要求を出力します。

タイマコントロールステータスレジスタ下位 (TCCS)

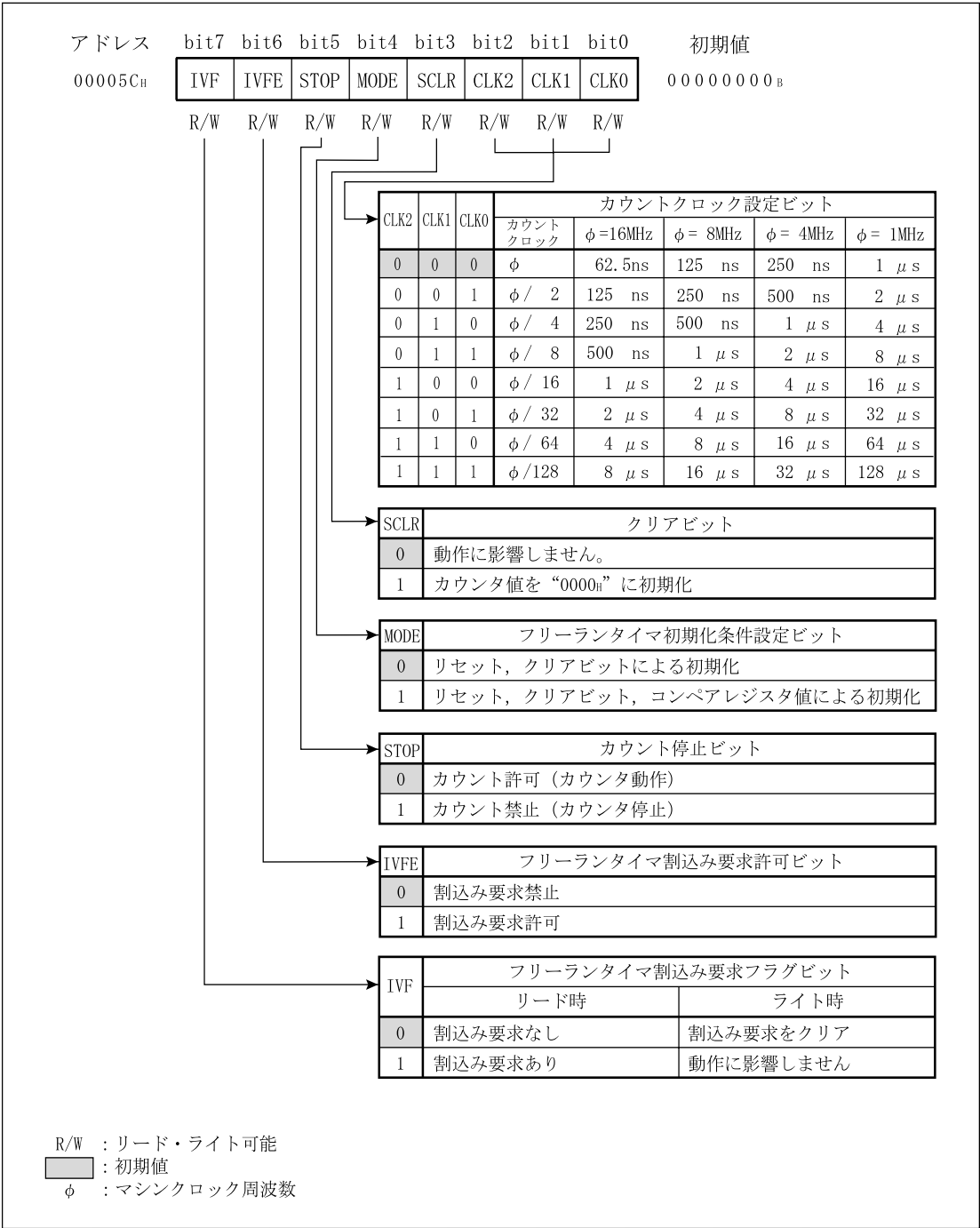


図 12.3-9 タイマコントロールステータスレジスタ下位 (TCCS)

表 12.3-2 タイマコントロールステータレジスタ下位 (TCCS) の各ビットの機能説明

ビット名		機 能
bit7	IVF : フリーランタイム 割込み要求フラグ ビット	割込み要求のフラグビットです。 16ビットフリーランタイムのカウント値のオーバーフローで, "1" がセットされます。 フリーランタイム割込み許可ビット (IVFE) に "1" を設定している場合に, "1" がセットされると, 割込みを出力します。 "0" を設定した場合は, 割込み要求がクリアされます。 "1" を設定した場合は, 動作に影響しません。 リードモディファイライト系命令では, "1" が読み出されます。
bit6	IVFE: フリーランタイム 割込み要求 許可ビット	割込み要求を許可するビットです。 "1" を設定した場合は, フリーランタイム割込み要求フラグビット (IVF) に, "1" がセットされると, 割込み要求を出力します。
bit5	STOP: カウント停止 ビット	16ビットフリーランタイムのカウントを停止するビットです。 "0" を設定した場合は, 16ビットフリーランタイムのカウントが動作します。 "1" を設定した場合は, 16ビットフリーランタイムのカウントが停止します。 <注記> 16ビットフリーランタイムのカウントが停止した場合, アウトプットコンペア動作も停止します。
bit4	MODE : フリーランタイム 初期化条件設定 ビット	16ビットフリーランタイムのカウント値の初期化条件を設定するビットです。 "0" を設定した場合は, リセット, クリアビット (SCLR="1") でカウント値を "0000H" にクリアします。 "1" を設定した場合は, リセット, クリアビット (SCLR="1"), 16ビットフリーランタイムのカウント値とコンペアクリアレジスタ (CPCLR) 値の一致でカウント値を "0000H" にクリアします。 <注記> カウント値のクリアは, MODE ビットに設定した初期化条件を検出した次のカウント値で行われます。
bit3	SCLR : クリアビット	16ビットフリーランタイムのカウント動作中に, カウント値を "0000H" にクリアするビットです。 "0" を設定した場合は, 動作に影響しません。 "1" を設定した場合は, カウント値を "0000H" にクリアします。 読出しを行った場合は, "0" が読出されます。 <注記> 16ビットフリーランタイムのカウント停止中 (STOP="1") に, カウント値を "0000H" にクリアする場合は, タイマデータレジスタ (TCDT) に "0000H" を設定してください。
bit2 bit1 bit0	CLK2, CLK1, CLK0 : カウントクロック 設定ビット	16ビットフリーランタイムのカウントクロックを設定するビットです。 カウントクロックは, 設定直後に変更されますので, アウトプットコンペア, およびインプットキャプチャが停止状態で設定してください。

12.3.2 アウトプットコンペアのレジスタ

- アウトプットコンペアのレジスタには，以下の2種類のレジスタがあります。
- ・コンペアレジスタ（OCCP0～OCCP5）
 - ・コンペアコントロールレジスタ（OCS0～OCS5）

コンペアレジスタ（OCCP0～OCCP5）

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch0:000071 _H	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	X X X X X X X B
ch1:000073 _H									
ch2:000075 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch3:000077 _H									
ch4:000079 _H									
ch5:00007B _H									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000070 _H	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	X X X X X X X B
ch1:000072 _H									
ch2:000074 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch3:000076 _H									
ch4:000078 _H									
ch5:00007A _H									
R/W：リード・ライト可能									
X：不定									

図 12.3-10 コンペアレジスタ（OCCP0～OCCP5）

16ビットフリーランタイマのカウンタ値と比較する値を設定する，16ビット長のレジスタです。初期値は不定ですので，割込み動作を許可する場合，およびアウトプットコンペアの端子出力を許可する場合は，コンペアレジスタ（OCCP0～OCCP5）値を設定してください。コンペアレジスタ（OCCP0～OCCP5）は，ワードアクセスしてください。

コンペアコントロールレジスタ上位 (OCS1,OCS3,OCS5)

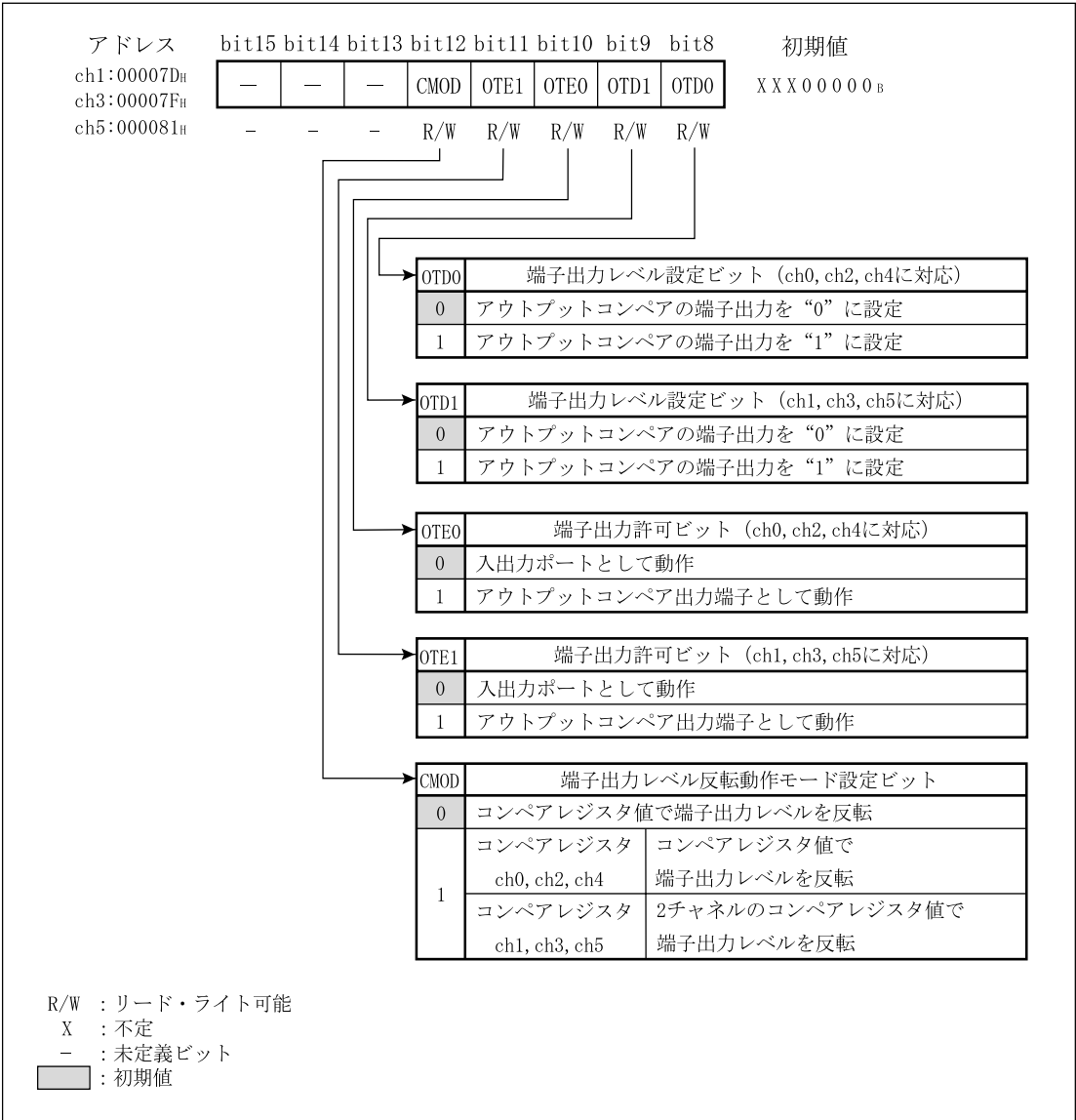


図 12.3-11 コンペアコントロールレジスタ上位 (OCS1,OCS3,OCS5)

表 12.3-3に、コンペアコントロールレジスタ上位 (OCS1, OCS3, OCS5) の各ビットの機能説明を示します。表 12.3-3では、コンペアコントロールレジスタ上位 (OCS1) について説明します。コンペアコントロールレジスタ上位 (OCS3) の場合は、ch0をch2, ch1をch3と、コンペアコントロールレジスタ上位 (OCS5) の場合は、ch0をch4, ch1を5chと読み替えてください。

表 12.3-3 コンペアコントロールレジスタ上位 (OCS1, OCS3, OCS5) の各ビットの機能説明

ビット名		機 能
bit15 bit14 bit13	- : 未定義ビット	読出しを行った場合は、不定値となります。 設定した値は、動作に影響しません。
bit12	CMOD : 端子出力レベル 反転動作モード 設定ビット	コンペアレジスタ (OCCP) ch0, ch1値と16ビットフリーランタイムのカウント値の一致により、端子出力レベルの反転動作モードを設定するビットです。 "0"を設定した場合、コンペアレジスタ (OCCP) ch0, およびコンペアレジスタ (OCCP) ch1に対応したRT0端子は、各チャンネルのコンペアレジスタ (OCCP) 値で出力レベルを反転します。 "1"を設定した場合、コンペアレジスタ (OCCP) ch0に対応したRT00端子は、ch0 のコンペアレジスタ (OCCP) 値で出力レベルを反転し、コンペアレジスタ (OCCP) ch1に対応したRT01端子は、ch0, ch1 のコンペアレジスタ (OCCP) 値で出力レベルを反転します。 コンペアレジスタ (OCCP) ch0値とコンペアレジスタ (OCCP) ch1値が同じ値の場合は、コンペアレジスタ (OCCP) 1チャンネルと同じ動作となります。
bit11 bit10	OTE1 (ch1に対応), OTE0 (ch0に対応) : 端子出力許可ビット	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。 "0"を設定した場合、各チャンネルに対応する端子は、入出力ポートとして動作します。 "1"を設定した場合、各チャンネルに対応する端子は、アウトプットコンペア出力端子として動作します。 <注記> 波形生成部でデッドタイムタイマを使用する場合は、8ビットタイマコントロールレジスタ (DTCR) の動作モード設定ビット (TMD2) に"1"を設定する必要があります。
bit9 bit8	OTD1 (ch1に対応), OTD0 (ch0に対応) : 端子出力レベル 設定ビット	アウトプットコンペアの端子出力レベルを設定するビットです。 "0"を設定した場合は、アウトプットコンペアの端子出力を"0"にします。 "1"を設定した場合は、アウトプットコンペアの端子出力を"1"にします。 設定する場合は、アウトプットコンペア動作を停止させてください。 読出しを行った場合は、アウトプットコンペアの端子出力値が読出されます。

コンペアコントロールレジスタ下位 (OCS0,OCS2,OCS4)

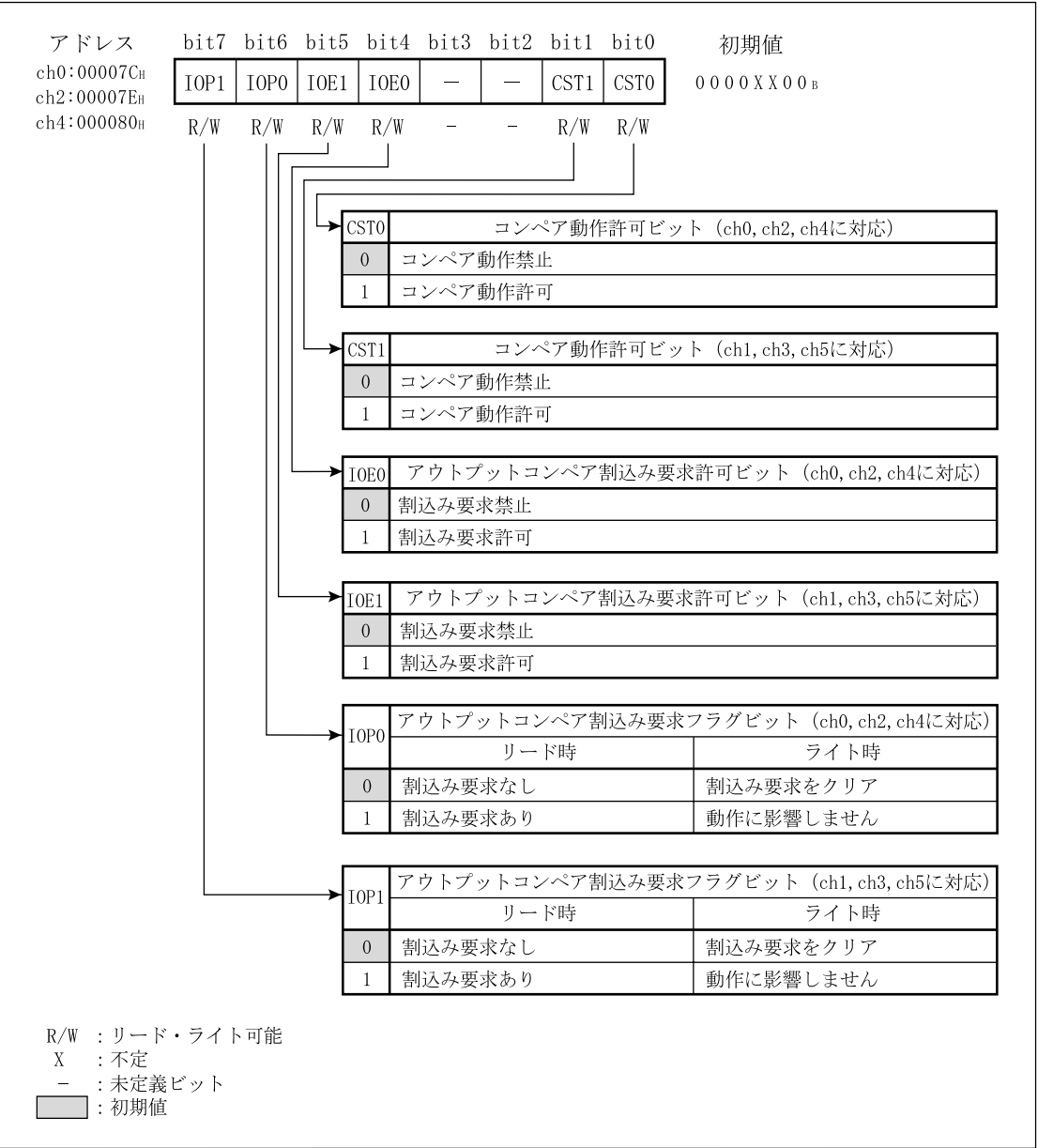


図 12.3-12 コンペアコントロールレジスタ下位 (OCS0,OCS2,OCS4)

表 12.3-4に、コンペアコントロールレジスタ下位(OCS0, OCS2, OCS4)の各ビットの機能説明を示します。表 12.3-4では、コンペアコントロールレジスタ下位(OCS0)について説明します。コンペアコントロールレジスタ下位(OCS2)の場合は、ch0をch2、ch1をch3と、コンペアコントロールレジスタ下位(OCS4)の場合は、ch0をch4、ch1を5chと読み替えてください。

表 12.3-4 コンペアコントロールレジスタ下位(OCS0, OCS2, OCS4)の各ビットの機能説明

ビット名		機 能
bit7 bit6	IOP1 (ch1に対応), IOP0 (ch0に対応): アウトプットコンペア 割込み要求フラグ ビット	割込み要求のフラグビットです。 コンペアレジスタ(OCCP)値と16ビットフリーランタイムのカウンタ値が一致した場合に,"1"がセットされます。 アウトプットコンペア割込み許可ビット(IOE1,IOE0)に"1"を設定している場合に,"1"がセットされると、割込み要求を出力します。 "0"を設定した場合は、割込み要求がクリアされます。 "1"を設定した場合は、動作に影響しません。 リードモディファイライト系命令では,"1"が読み出されます。
bit5 bit4	IOE1 (ch1に対応), IOE0 (ch0に対応): アウトプットコンペア 割込み要求許可ビット	割込み要求を許可するビットです。 "1"を設定した場合は、アウトプットコンペア割込み要求フラグビット(IOP1,IOP0)に,"1"がセットされると、割込み要求を出力します。
bit3 bit2	- : 未定義ビット	読出しを行った場合は、不定値となります。 設定した値は、動作に影響しません。
bit1 bit0	CST1 (ch1に対応), CST0 (ch0に対応): コンペア動作許可 ビット	コンペアレジスタ(OCCP)値と16ビットフリーランタイムのカウンタ値とのコンペア動作を許可するビットです。 コンペア動作を許可する前に、コンペアレジスタ(OCCP)値を設定してください。 "1"を設定した場合は、コンペア動作を行います。 <注記> アウトプットコンペアは、16ビットフリーランタイムと同期しているため、16ビットフリーランタイムのカウンタを停止(TCCS:STOP=1)させると、コンペア動作も停止します。

12.3.3 インพุットキャプチャのレジスタ

- インพุットキャプチャのレジスタには、以下の2種類のレジスタがあります。
- ・インพุットキャプチャデータレジスタ (IPCP0 ~ IPCP3)
 - ・インพุットキャプチャコントロールレジスタ (ICS01, ICS23)

インพุットキャプチャデータレジスタ (IPCP0 ~ IPCP3)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch0:000061 _H	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	X X X X X X X B
ch1:000063 _H									
ch2:000065 _H	R	R	R	R	R	R	R	R	
ch3:000067 _H									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000060 _H	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	X X X X X X X B
ch1:000062 _H									
ch2:000064 _H	R	R	R	R	R	R	R	R	
ch3:000066 _H									
R : リードオンリー									
X : 不定									

図 12.3-13 インพุットキャプチャデータレジスタ (IPCP0 ~ IPCP3)

外部入力端子 (IN0 ~ IN3) から入力波形の有効エッジを検出した場合に、16ビットフリーランタイムのカウンタ値を取り込むレジスタです。インพุットキャプチャデータレジスタ (IPCP0 ~ IPCP3) は、ワードアクセスしてください。レジスタへの設定はできません。

インプットキャプチャコントロールレジスタ01 (ICS01)

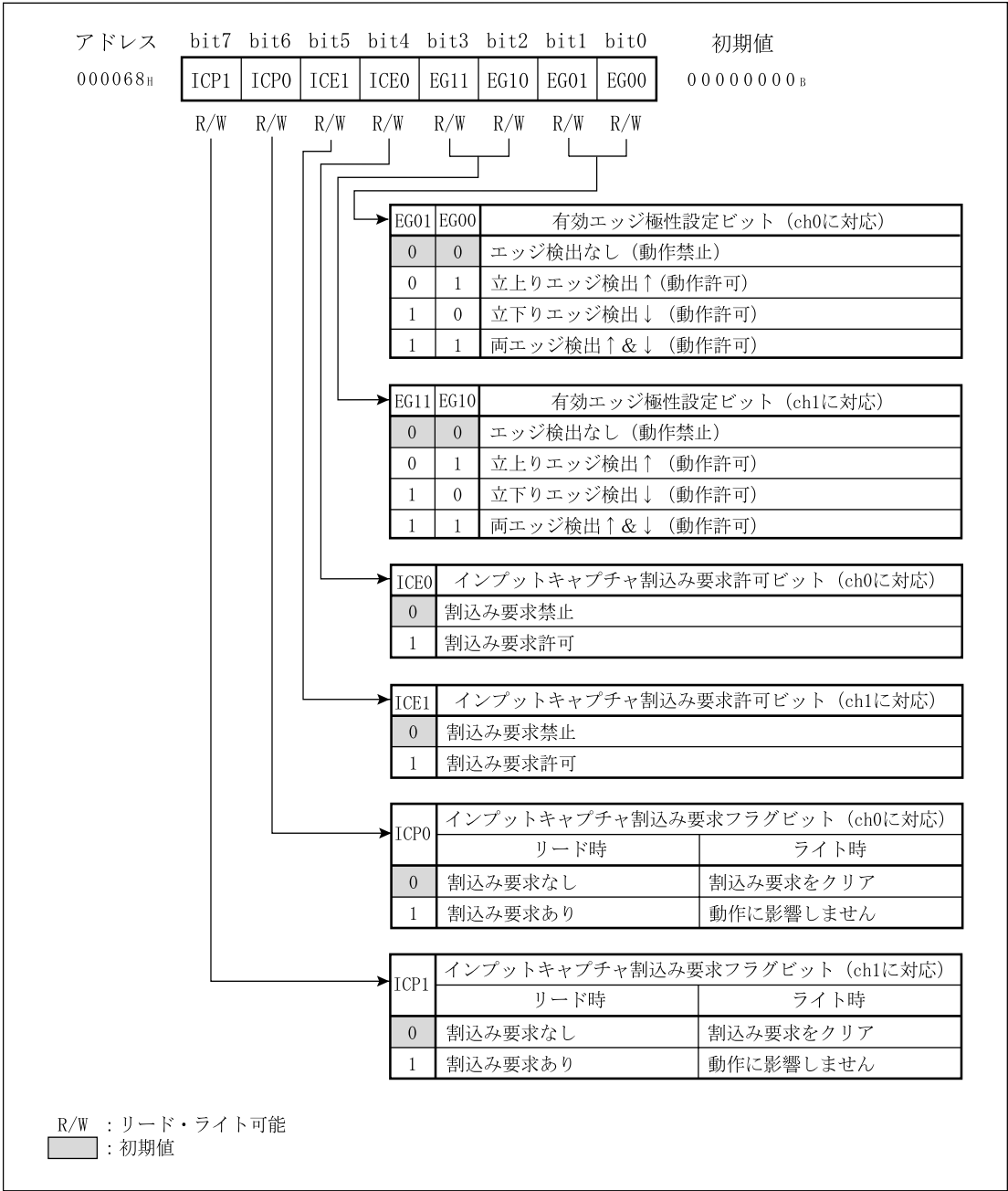


図 12.3-14 インプットキャプチャコントロールレジスタ01 (ICS01)

表 12.3-5 インพุットキャプチャコントロールレジスタ01 (ICS01) の各ビットの機能説明

ビット名		機 能
bit7 bit6	ICP1 : (ch1に対応) ICP0 : (ch0に対応) インพุット キャプチャ割込み 要求フラグビット	<p>割込み要求のフラグビットです。</p> <p>外部入力端子 (IN0 ~ IN3) の有効エッジ極性を検出した場合に "1" がセットされます。</p> <p>インพุットキャプチャ割込み許可ビット (ICE1 , ICE0) に "1" を設定している場合に, "1" がセットされると, 割込み要求を出力します。</p> <p>"0" を設定した場合は, 割込み要求がクリアされます。</p> <p>"1" を設定した場合は, 動作に影響しません。</p> <p>リードモディファイライト系命令では, "1" が読み出されます。</p>
bit5 bit4	ICE1 : (ch1に対応) ICE0 : (ch0に対応) インพุット キャプチャ割込み 要求許可ビット	<p>割込み要求を許可するビットです。</p> <p>"1" を設定した場合は, インพุットキャプチャ割込み要求フラグビット (ICP1 , ICP0) に, "1" がセットされると, 割込み要求を出力します。</p>
bit3 bit2 bit1 bit0	EG11, EG10 : (ch1に対応) EG01, EG00 : (ch0に対応) 有効エッジ極性 設定ビット	<p>入力波形の有効エッジ極性, およびインพุットキャプチャ動作許可を設定するビットです。</p> <p>"01_B" ~ "11_B" を設定した場合は, インพุットキャプチャ動作許可となり, 入力波形の有効エッジ極性は, 立上りエッジ検出, 立下りエッジ検出, および両エッジ検出から設定できます。</p> <p>"00_B" を設定した場合は, インพุットキャプチャ動作禁止, エッジ検出なしとなります。</p>

インプットキャプチャコントロールレジスタ23 (ICS23)

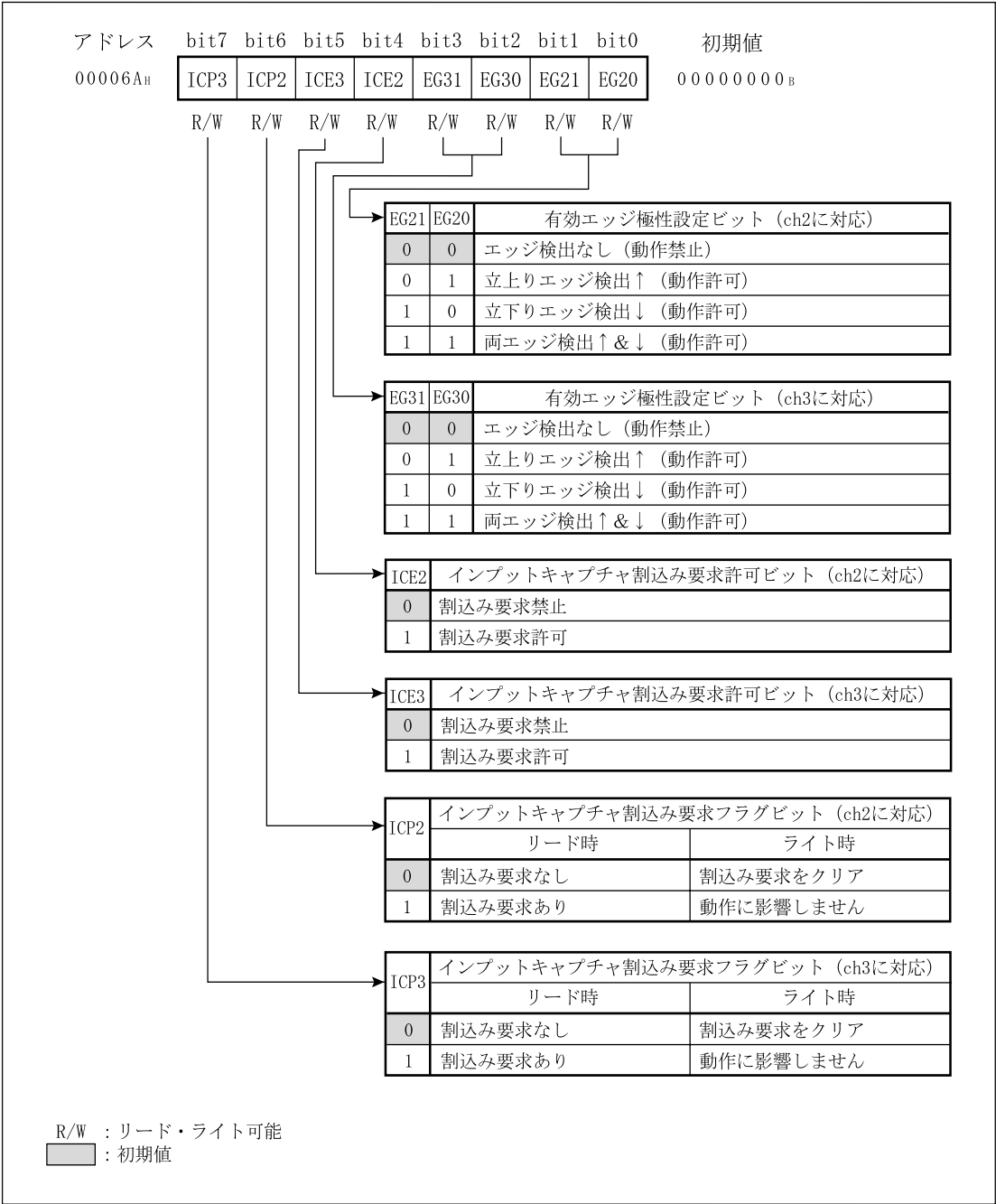


図 12.3-15 インプットキャプチャコントロールレジスタ23 (ICS23)

表 12.3-6 インพุットキャプチャコントロールレジスタ23 (ICS23) の各ビットの機能説明

ビット名		機 能
bit7 bit6	ICP3 : (ch3に対応) ICP2 : (ch2に対応) インพุット キャプチャ割込み 要求フラグビット	割込み要求のフラグビットです。 外部入力端子 (IN0 ~ IN3) の有効エッジ極性を検出した場合に "1" がセットされます。 インพุットキャプチャ割込み許可ビット (ICE3, ICE2) に "1" を設定している場合に, "1" がセットされると, 割込み要求を出力します。 "0" を設定した場合は, 割込み要求がクリアされます。 "1" を設定した場合は, 動作に影響しません。 リードモディファイライト系命令では, "1" が読み出されます。
bit5 bit4	ICE3: (ch3に対応) ICE2: (ch2に対応) インพุット キャプチャ割込み 要求許可ビット	割込み要求を許可するビットです。 "1" を設定した場合は, インพุットキャプチャ割込み要求フラグビット (ICP3, ICP2) に "1" がセットされると, 割込み要求を出力します。
bit3 bit2 bit1 bit0	EG31, EG30 : (ch3に対応) EG21, EG20 : (ch2に対応) 有効エッジ極性 設定ビット	入力波形の有効エッジ極性, およびインพุットキャプチャ動作許可を設定するビットです。 "01 _B " ~ "11 _B " を設定した場合は, インพุットキャプチャ動作許可となり, 入力波形の有効エッジ極性は, 立上りエッジ検出, 立下りエッジ検出, および両エッジ検出から設定できます。 "00 _B " を設定した場合は, インพุットキャプチャ動作禁止, エッジ検出なしとなります。

12.3.4 8/16ビットPPGタイマのレジスタ

8/16ビットPPGタイマのレジスタには、以下の3種類のレジスタがあります。

- ・ PPGコントロールレジスタ (PPGC0 ~ PPGC5)
- ・ PPGクロック制御レジスタ (PCS01, PCS23, PCS45)
- ・ PPGリロードレジスタ (PRL0 ~ PRL5, PRLH0 ~ PRLH5)

PPGコントロールレジスタ上位 (PPGC1, PPGC3, PPGC5)

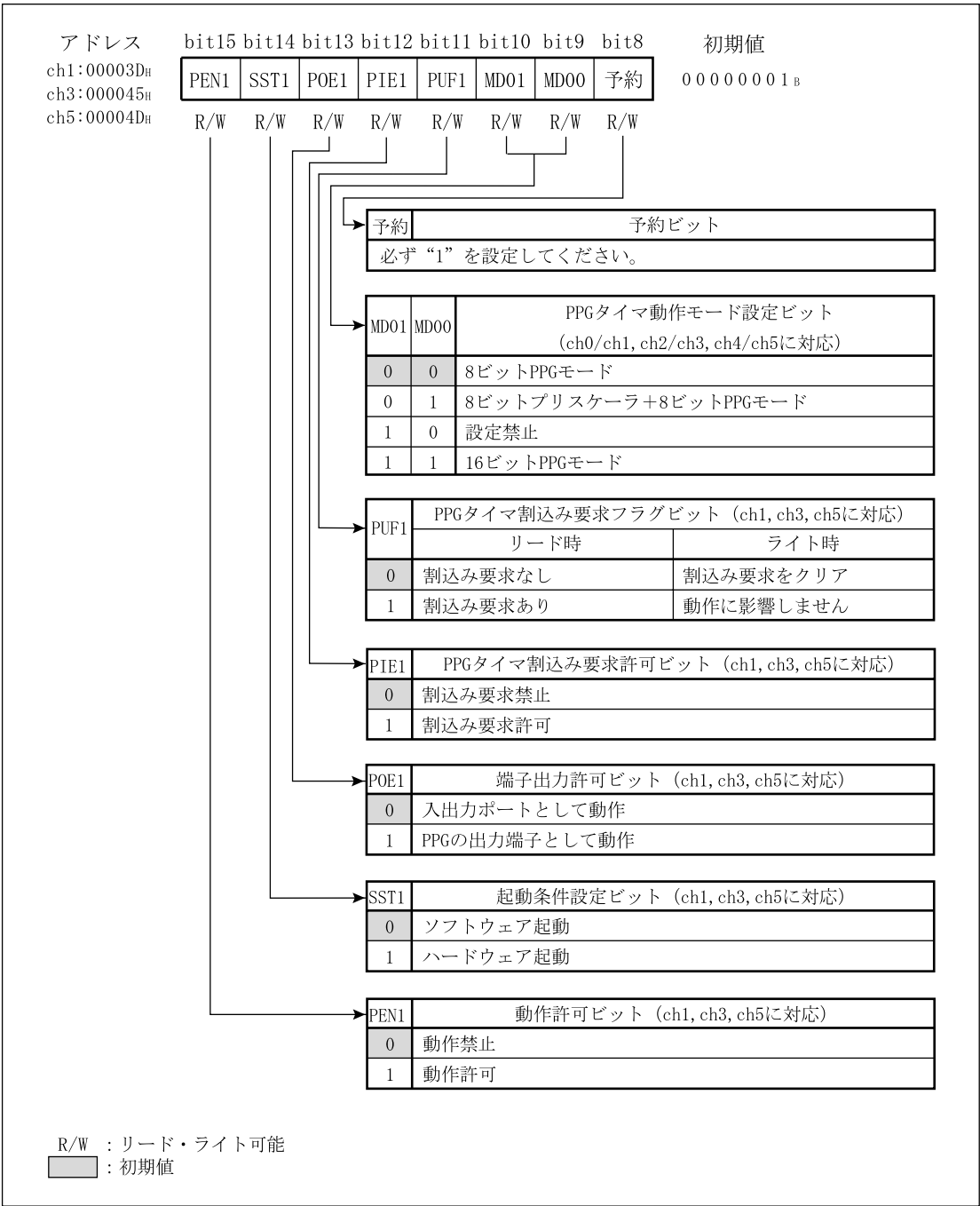


図 12.3-16 PPGコントロールレジスタ上位 (PPGC1, PPGC3, PPGC5)

表 12.3-7 PPGコントロールレジスタ上位 (PPGC1, PPGC3, PPGC5) の各ビットの機能説明
 [PPGコントロールレジスタ上位は, ch1, ch3, ch5に対応]

ビット名		機 能
bit15	PEN1 : 動作許可ビット	ソフトウェア起動を設定した場合の動作を許可するビットです。 "1"を設定した場合は, 起動条件設定ビット (SST1) に"0"を設定しているとPPGタイマはカウントを開始します。 起動条件設定ビット (SST1) に"1"が設定されている場合は, 動作に影響しません。
bit14	SST1 : 起動条件設定ビット	PPGタイマの起動条件を設定するビットです。 "0"を設定した場合は, 動作許可ビット (PEN1) に"1"が設定されるとPPGタイマはカウントを開始します。 "1"を設定した場合は, 動作許可ビット (PEN1) の設定は, 動作に影響しません。波形生成部より入力されるGATE信号の"H"期間, PPGは動作します。
bit13	POE1 : 端子出力 許可ビット	PPG端子にPPGタイマのパルス出力を許可するビットです。 "0"を設定した場合は, 各チャンネルに対応した端子は, 入出力ポートとして動作します。 "1"を設定した場合は, 各チャンネルに対応した端子は, PPGの出力端子として動作します。
bit12	PIE1 : PPGタイマ割込み要求 許可ビット	割込み要求を許可するビットです。 "1"を設定した場合は, PPGタイマ割込み要求フラグビット (PUF1) に"1"がセットされると, 割込み要求を出力します。
bit11	PUF1 : PPGタイマ割込み要求 フラグビット	割込み要求のフラグビットです。 PPGタイマのカウント値のアンダーフローで"1"がセットされます。 PPGタイマ割込み許可ビット (PIE1) に"1"を設定している場合に, "1"がセットされると, 割込み要求を出力します。 "0"を設定した場合は, 割込み要求がクリアされます。 "1"を設定した場合は, 動作に影響しません。 リードモディファイライト系命令では, "1"が読み出されます。 <注記> 8ビットPPGモード, および8ビットプリスケラ+8ビットPPGモードの場合は, カウント値のアンダーフロー ("00 _H " "FF _H ") により"1"がセットされます。 16ビットPPGモードの場合は, カウント値のアンダーフロー ("0000 _H " "FFFF _H ") により"1"がセットされます。
bit10 bit9	MD01, MD00 : PPGタイマ動作モード 設定ビット	PPGタイマの動作モードを設定するビットです。 <注記> ソフトウェア起動を設定 (SST1="0") する場合は, 次の点に注意してください。 ・ "01 _B "を設定する場合は, 動作許可ビット (PEN0) に"0", 動作許可ビット (PEN1) に"1"を設定しないでください。 ・ "11 _B "を設定する場合は, ワード転送で, 動作許可ビット (PEN0, PEN1) を同時に"1", または"0"の設定を行ってください。 ・ ハードウェア起動を設定 (SST1="1") する場合は, 次の点に注意してください。 ・ "01 _B "を設定する場合は, 動作許可ビット (PEN0) に"1"を設定後, 起動条件設定ビット (SST1) に"1"を設定してください。 ・ 動作許可ビット (PEN0) に"0"を設定する場合は, 起動条件設定ビット (SST1) に"0"を設定した後に行ってください。 ・ "11 _B "を設定する場合は, ワード転送で, 起動条件設定ビット (SST0, SST1) を同時に"1", または"0"の設定を行ってください。
bit8	予約: 予約ビット	必ず"1"を設定してください。

PPGコントロールレジスタ下位 (PPGC0,PPGC2,PPGC4)

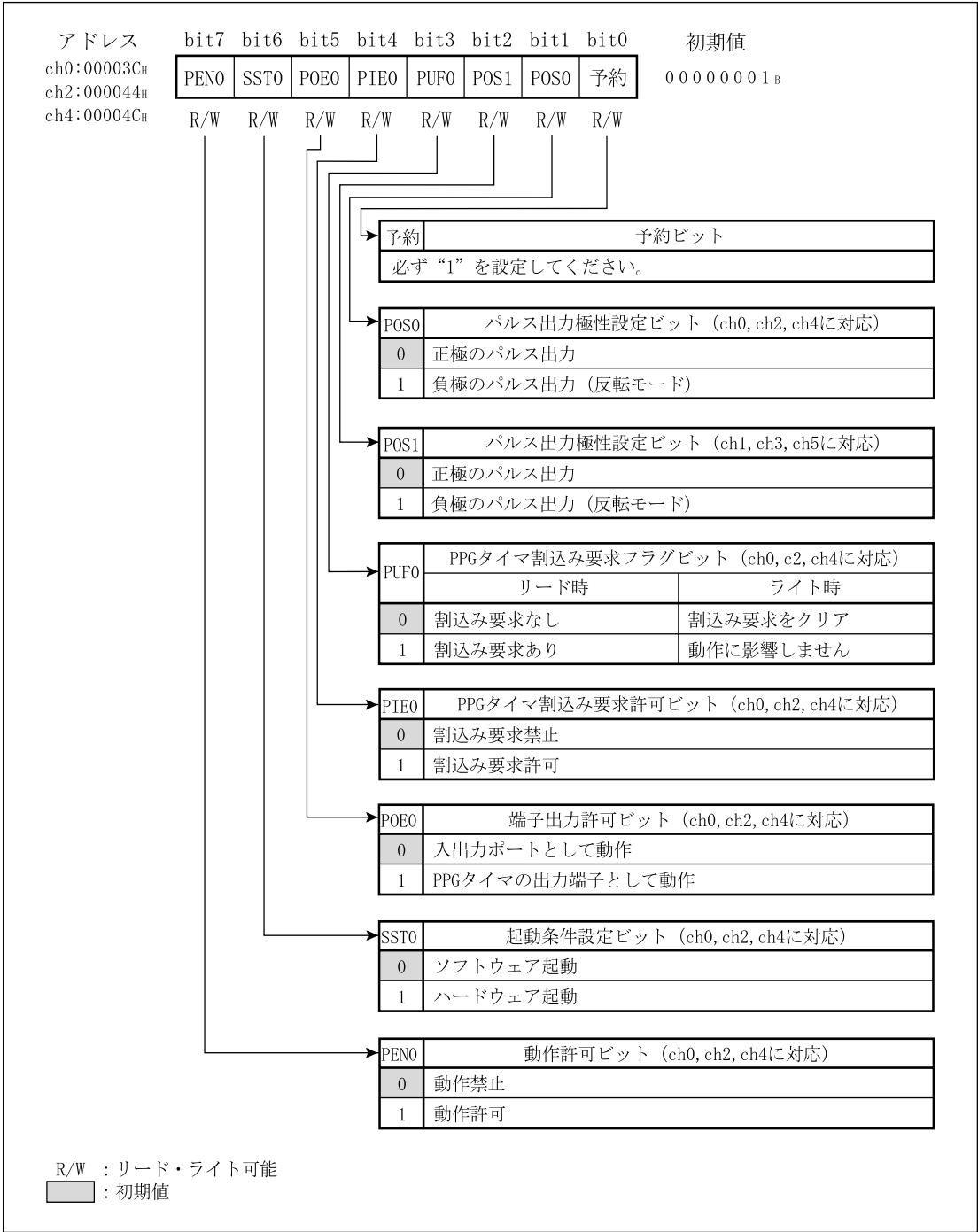


図 12.3-17 PPGコントロールレジスタ下位 (PPGC0,PPGC2,PPGC4)

表 12.3-8 PPGコントロールレジスタ下位 (PPGC0, PPGC2, PPGC4) の各ビットの機能説明
 [PPGコントロールレジスタ下位は, ch0, ch2, ch4に対応]

ビット名		機 能
bit7	PEN0 : 動作許可ビット	ソフトウェア起動を設定した場合の動作許可ビットです。 "1"を設定した場合は, 起動条件設定ビット (SST0) に"0"を設定しているとPPGタイマはカウントを開始します。 起動条件設定ビット (SST0) に"1"が設定されている場合は, 動作に影響しません。
bit6	SST0 : 起動条件設定ビット	PPGタイマの起動条件を設定するビットです。 "0"を設定した場合は, 動作許可ビット (PEN0) に"1"が設定されるとPPGタイマはカウントを開始します。 "1"を設定した場合は, 動作許可ビット (PEN0) の設定は, 動作に影響しません。波形生成部より入力されるGATE信号の"H"期間, PPGタイマは動作します。
bit5	POE0 : 端子出力許可ビット	外部端子にPPGタイマのパルス出力を許可するビットです。 "0"を設定した場合は, 各チャンネルに対応した端子は, 入出力ポートとして動作します。 "1"を設定した場合は, 各チャンネルに対応した端子は, PPGの出力端子として動作します。
bit4	PIE0 : PPGタイマ割込み要求許可ビット	割込み要求を許可するビットです。 "1"を設定した場合は, PPGタイマ割込み要求フラグビット (PUF0) に, "1"がセットされると, 割込み要求を出力します。
bit3	PUF0 : PPGタイマ割込み要求フラグビット	割込み要求のフラグビットです。 PPGタイマのカウント値のアンダーフローで"1"がセットされます。 PPGタイマ割込み許可ビット (PIE0) に"1"を設定している場合に, "1"がセットされると, 割込み要求を出力します。 "0"を設定した場合は, 割込み要求がクリアされます。 "1"を設定した場合は, 動作に影響しません。 リードモディファイライト系命令では, "1"が読み出されます。 <注記> 8ビットPPGモード, および8ビットプリスケラ+8ビットPPGモードの場合は, カウント値のアンダーフロー ("00H" "FFH") により"1"がセットされます。 16ビットPPGモードの場合は, カウント値のアンダーフロー ("0000H" "FFFFH") により"1"がセットされます。
bit2	POS1 : (ch1, ch3, ch5 に対応) パルス出力極性設定ビット	外部端子にPPGタイマのパルス出力極性を設定するビットです。
bit1	POS0 : (ch0, ch2, ch4 に対応) パルス出力極性設定ビット	外部端子にPPGタイマのパルス出力極性を設定するビットです。
bit0	予約: 予約ビット	必ず"1"を設定してください。

PPGクロック制御レジスタ (PCS01,PCS23,PCS45)

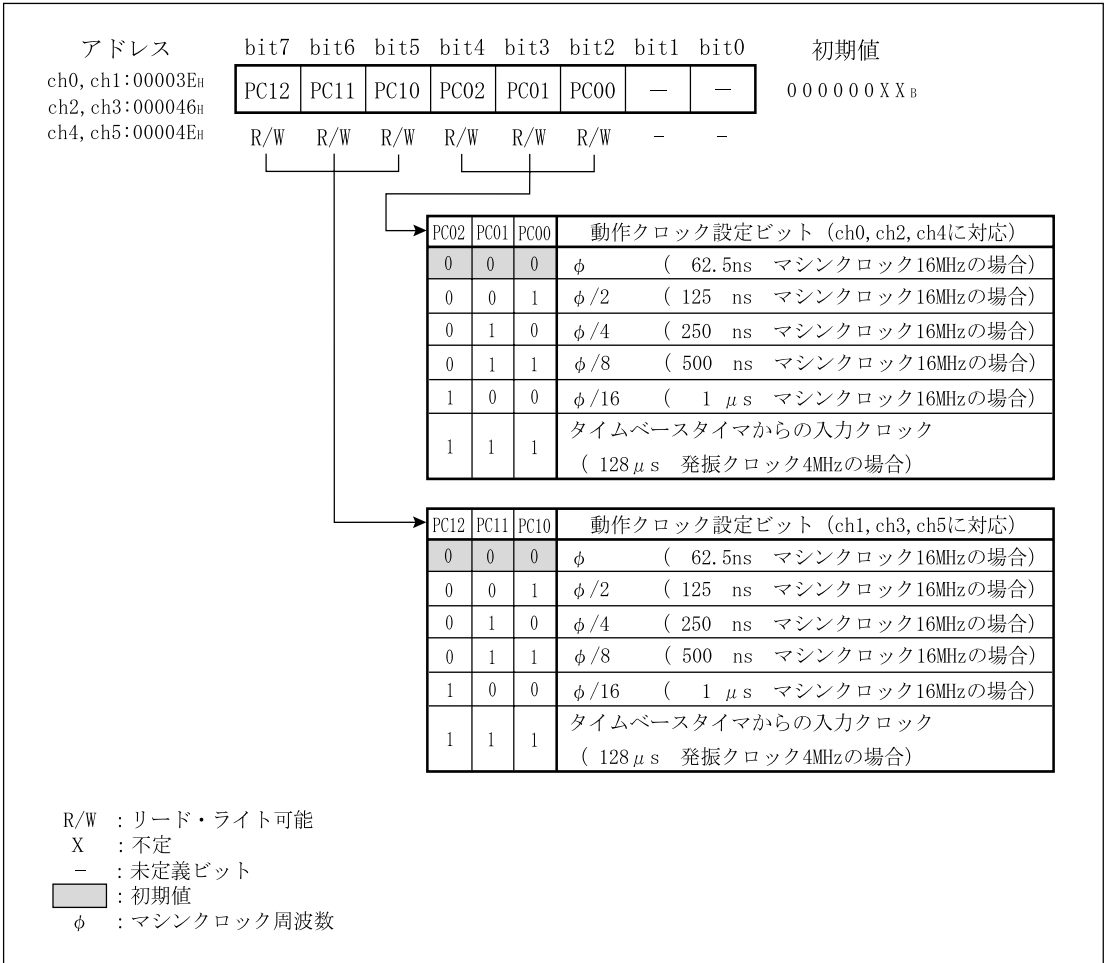


図 12.3-18 PPGクロック制御レジスタ (PCS01,PCS23,PCS45)

表 12.3-9 PPGクロック制御レジスタ (PCS01,PCS23,PCS45) の各ビットの機能説明

ビット名		機 能
bit7 bit6 bit5	PC12,PC11,PC10 : (ch1, ch3, ch5に対応) 動作クロック 設定ビット	8ビットダウンカウンタ (PCNT) の動作クロックを設定するビットです。 <注記> ・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 の場合は, PPG ch1(ch3,ch5)は, PPG ch0 (ch2, ch4) から動作クロッ クを受けて動作するため, PC12,PC11,PC10ビットの設定は, 動作に 影響しません。
bit4 bit3 bit2	PC02,PC01,PC00 : (ch0, ch2, ch4に対応) 動作クロック 設定ビット	8ビットダウンカウンタ (PCNT) の動作クロックを設定するビットです。
bit1 bit0	- : 未定義ビット	読出しを行った場合は, 不定値となります。 設定した値は, 動作に影響しません。

PPGリロードレジスタ上位 (PRLH0 ~ PRLH5)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch0:000039 _H									X X X X X X X _B
ch1:00003B _H									
ch2:000041 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch3:000043 _H									
ch4:000049 _H									
ch5:00004B _H									

R/W : リード・ライト可能
X : 不定

図 12.3-19 PPGリロードレジスタ上位 (PRLH0 ~ PRLH5)

8ビットダウンカウンタ (PCNT) の上位側リロード値を設定するレジスタです。

PPGリロードレジスタ下位 (PRLL0 ~ PRLL5)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000038 _H									X X X X X X X _B
ch1:00003A _H									
ch2:000040 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch3:000042 _H									
ch4:000048 _H									
ch5:00004A _H									

R/W : リード・ライト可能
X : 不定

図 12.3-20 PPGリロードレジスタ下位 (PRLL0 ~ PRLL5)

8ビットダウンカウンタ (PCNT) の下位側リロード値を設定するレジスタです。

< 注意事項 >

8ビットプリスケラ + 8ビットPPGモードの場合は、ch0 (ch2, ch4) のPPGリロードレジスタ上位 (PRLH) とPPGリロードレジスタ下位 (PRLL) に異なる値を設定すると、ch1 (ch3, ch5) のPPG波形がサイクルごとに異なりますので、ch0(ch2, ch4)のPPGリロードレジスタ上位 (PRLH) 値、およびPPGリロードレジスタ下位 (PRLL) 値は、同じ値を設定してください。

12.3.5 波形生成部のレジスタ

波形生成部には、以下の3種類のレジスタがあります。

- ・8ビットタイマコントロールレジスタ (DTCR0 ~ DTCR2)
- ・8ビットリロードレジスタ (TMRR0 ~ TMRR2)
- ・波形制御レジスタ (SIGCR)

8ビットタイマコントロールレジスタ (DTCR0 ~ DTCR2)

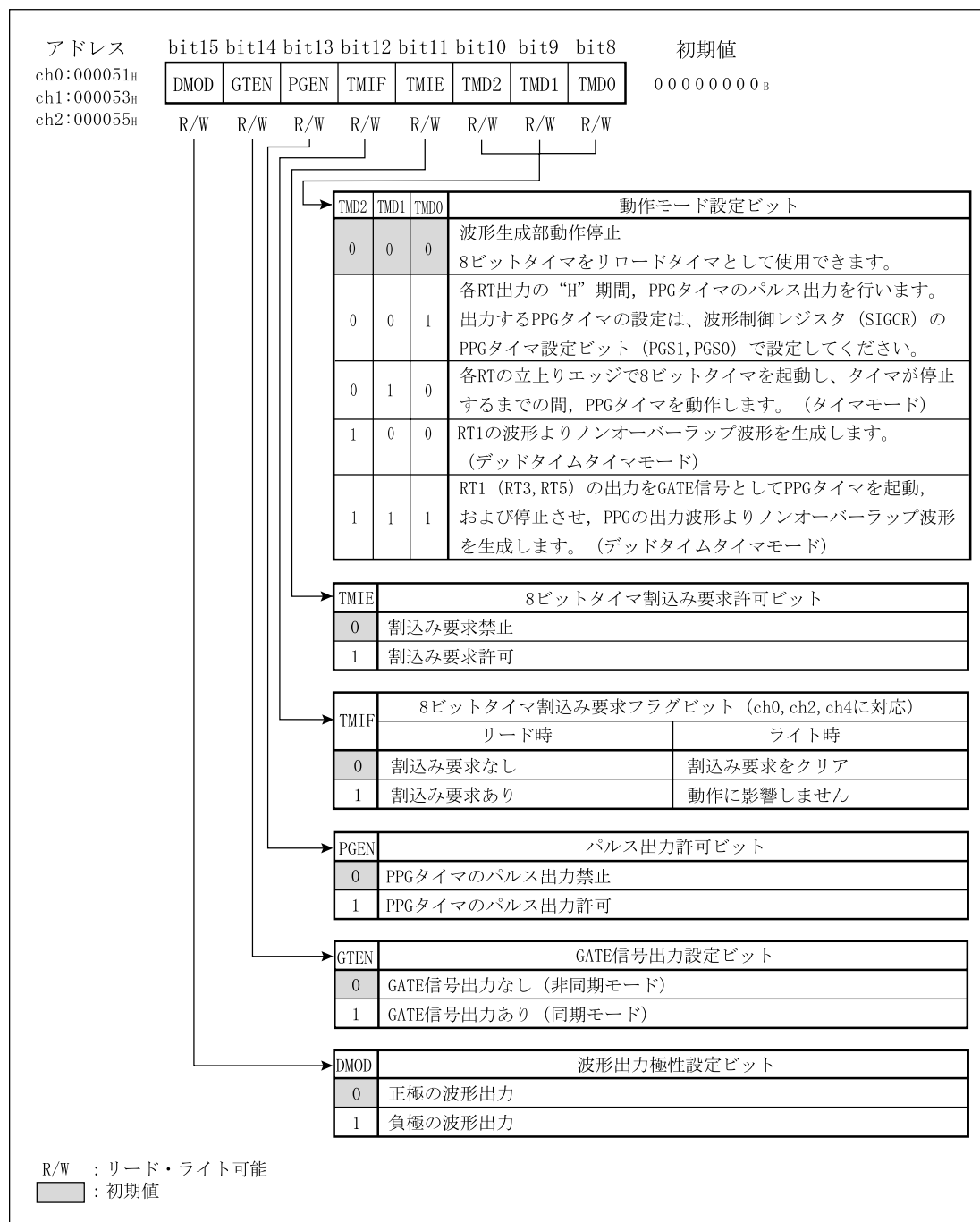


図 12.3-21 8ビットタイマコントロールレジスタ (DTCR0 ~ DTCR2)

表 12.3-10 8ビットタイマコントロールレジスタ (DTCR0 ~ DTCR2) の各ビットの機能説明

ビット名		機 能
bit15	DMOD : 波形出力極性設定 ビット	8ビットタイマをデッドタイムタイマとして動作させる場合の、波形出力極性を設定するビットです。 8ビットタイマをデッドタイムタイマとして動作させない場合 (TMD2="0") は、DMODビットの設定は動作に影響しません。
bit14	GTEN : GATE信号 出力設定ビット	PPGタイマ動作のGATE信号出力を設定するビットです。
bit13	PGEN : パルス出力 許可ビット	RTO端子にPPGタイマのパルス出力を許可するビットです。
bit12	TMIF : 8ビットタイマ割込み 要求フラグビット	割込み要求のフラグビットです。 8ビットタイマのカウント値のアンダーフローで"1"がセットされます。 8ビットタイマ割込み許可ビット (TMIE) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 "0"を設定した場合は、割込み要求がクリアされます。 "1"を設定した場合は、動作に影響しません。 リードモディファイライト系命令では、"0"が読み出されます。
bit11	TMIE : 8ビットタイマ割込み 許可ビット	8ビットタイマの起動、および割込み要求を許可するビットです。 "1"を設定した場合は、8ビットタイマが動作を開始し、8ビットタイマ割込み要求フラグビット (TMIF) に"1"がセットされると、割込み要求を出力します。 <注記> 8ビットタイマをリロードタイマとして使用できるのは、bit10 ~ bit8:TMD2 ~ TMD0に"000 _B "を設定されている場合です。
bit10 bit9 bit8	TMD2, TMD1, TMD0 : 動作モード設定 ビット	波形生成部の動作モードを設定するビットです。 <注記> 記載以外の設定は、誤動作を起こしますので、設定しないでください。 "111 _B "を設定する場合は、PPGタイマの動作モードを8ビットPPGモードに設定 (PPGC:MD01, MD00="00 _B ") してください。

8ビットリロードレジスタ (TMRRO ~ TMRR2)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000050 _H									X X X X X X X B
ch1:000052 _H									
ch2:000054 _H									
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード・ライト可能
X : 不定

図 12.3-22 8ビットリロードレジスタ (TMRRO ~ TMRR2)

8ビットタイマのコンペア値を設定するレジスタです。8ビットタイマが起動されるとリロードされます。8ビットタイマ動作中に8ビットリロードレジスタ (TMRR) 値を設定した場合は、次の8ビットタイマ起動から有効となります。デッドタイムタイマモードで使用する場合は、ノンオーバーラップ時間を設定するレジスタになります。

- ノンオーバーラップ時間は、次式から求められます。

ノンオーバーラップ時間 = (TMRRレジスタ設定値 + 1) × 動作クロック

TMRRレジスタ設定値には、"00H"は設定できません。タイマモードで使用する場合は、PPGタイマ動作のGATE時間を設定するレジスタとなります。

- GATE時間は、次式から求められます。

GATE時間 = (TMRRレジスタ設定値 + 1) × 動作クロック

TMRRレジスタ設定値には、"00H"は設定できません。

波形制御レジスタ (SIGCR)

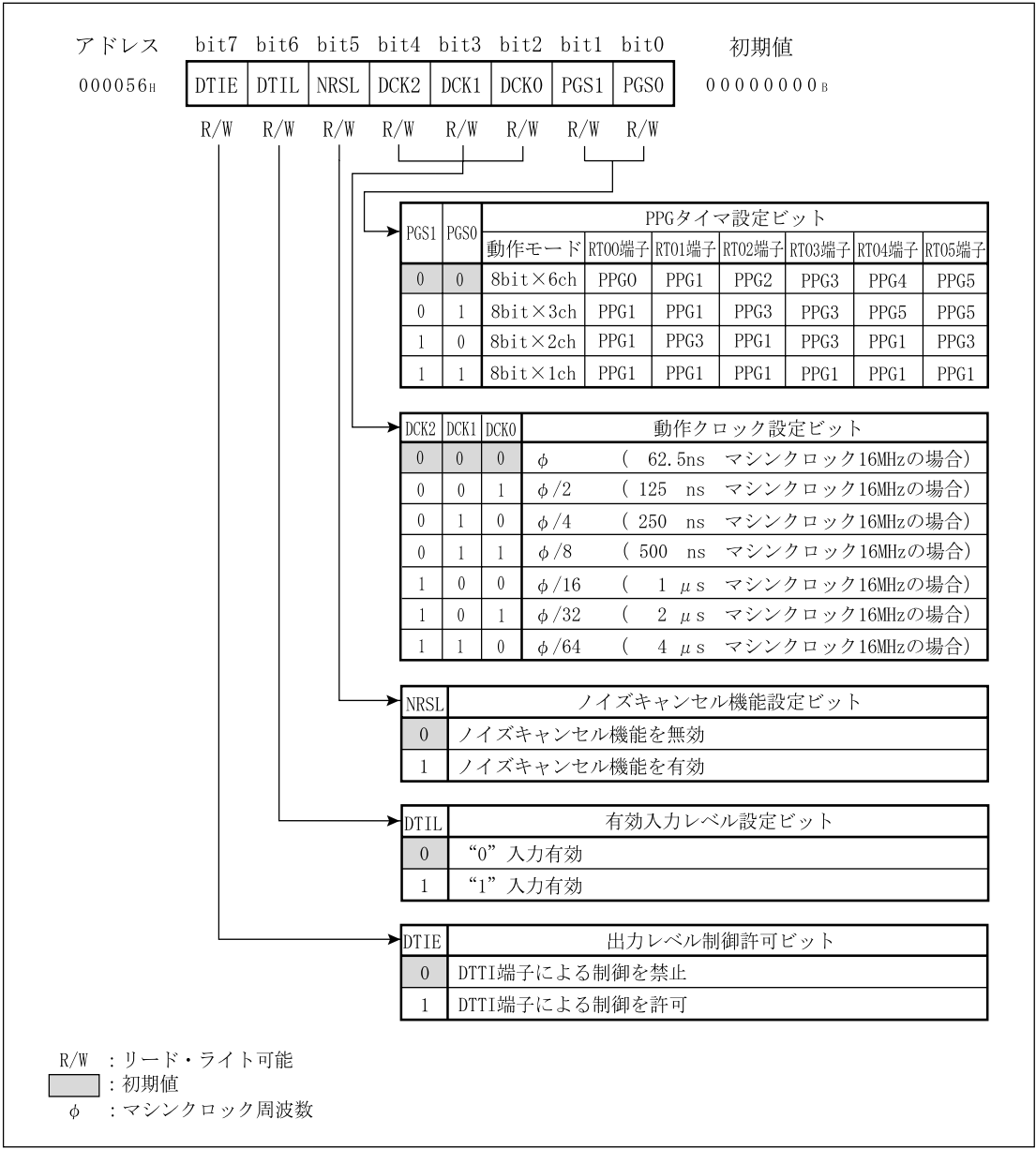


図 12.3-23 波形制御レジスタ (SIGCR)

表 12.3-11 波形制御レジスタ (SIGCR) の各ビットの機能説明

ビット名		機 能
bit7	DTIE : 出力レベル制御 許可ビット	DTTI端子入力により, RTO端子の出力レベルの制御を許可するビットです。
bit6	DTIL : 有効入力レベル 設定ビット	DTTI端子の有効入力レベルを設定するビットです。
bit5	NRSL : ノイズキャンセル 機能設定ビット	DTTI端子入力のノイズキャンセル機能を設定するビットです。 ノイズキャンセル回路は有効入力レベルにより, 内部 2ビットカウンタを動作させ, カウンタ値がオーバーフローするまで有効レベルが保持されていた場合に, DTTI端子入力を受け付けます。 キャンセルできるノイズのパルス幅は, 約4マシンスサイクルです。 <注記> ノイズキャンセル回路を有効にした場合は, ストップモードで発振クロックを停止させると, DTTI端子入力は無効となります。
bit4 bit3 bit2	DCK2, DCK1, DCK0 : 動作クロック 設定ビット	・8ビットタイマの動作クロックを設定するビットです。
bit1 bit0	PGS1, PGS0 : PPGタイマ設定ビット	RTO端子へ, PPGタイマのパルスを出力する場合に使用するPPGタイマを設定するビットです。 8ビットタイマコントロールレジスタ (DTCR) のゲート信号出力設定ビット (GTEN), パルス出力許可ビット (PGEN) を"1"に設定した場合は, 対応するPPGタイマ動作のGATE信号 (同期モード) が出力されます。 <注記> PPGタイマの設定で16ビットPPGモードを設定した場合でも, 対応するPPGタイマの出力, および起動/停止制御ができます。 PPGタイマ動作のGATE信号を使用しなくても (非同期モード), PPGタイマをソフトウェア起動 (PPGコントロールレジスタ (PPGC) で設定) させることにより, PPGタイマのパルスは出力します。

12.4 多機能タイマの動作説明

多機能タイマユニットの動作について説明します。

多機能タイマの動作

16ビットフリーランタイマ

16ビットフリーランタイマのカウント値は、リセット解除後 "0000H" からカウントを開始します。カウント出力値は、アウトプットコンペア、およびインプットキャプチャの基準時間（ベースタイマ）として使用されます。

アウトプットコンペア

アウトプットコンペアは、コンペアレジスタ（OCCP）値と16ビットフリーランタイマのカウント値が一致した場合に、出力レベルを反転し割込みを出力します。

インプットキャプチャ

インプットキャプチャは、外部入力端子（IN0～IN3）から入力された信号の有効エッジを検出することにより、16ビットフリーランタイマのカウント値をインプットキャプチャデータレジスタ（IPCP）に取り込み割込みを出力します。

8/16ビットPPGタイマ

8/16ビットPPGタイマには、8ビットPPGモード、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3種類の動作モードがあります。

波形生成部

波形生成部は、リアルタイムアウトプット（RT）、8/16ビットPPGタイマ、および8ビットタイマにより、様々なタイミング波形が生成できます。

12.4.1 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カウント値は，リセット解除後"0000H"からカウントを開始します。カウンタ出力値は，アウトプットコンペア，およびインプットキャプチャの基準時間（ベースタイム）として使用されます。

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カウント値は，次の条件で"0000H"にクリアされます。

- リセット
- タイマコントロールステータスレジスタ（TCCS）のクリアビット（SCLR）に"1"を設定
- 16ビットフリーランタイムのカウント値のオーバーフロー
- 16ビットフリーランタイムのカウント値とコンペアクリアレジスタ（CPCLR）値の一致（TCCS:MODE="1"）
- タイマデータレジスタ（TCDT）へ"0000H"を設定

16ビットフリーランタイムのカウント値がオーバーフローした場合，または16ビットフリーランタイムのカウント値とコンペアクリアレジスタ（CPCLR）値が一致（TCCS:ICRE="1"，MODE="1"）し，16ビットフリーランタイムのカウント値が"0000H"にクリアされた場合に割込みを出力します。

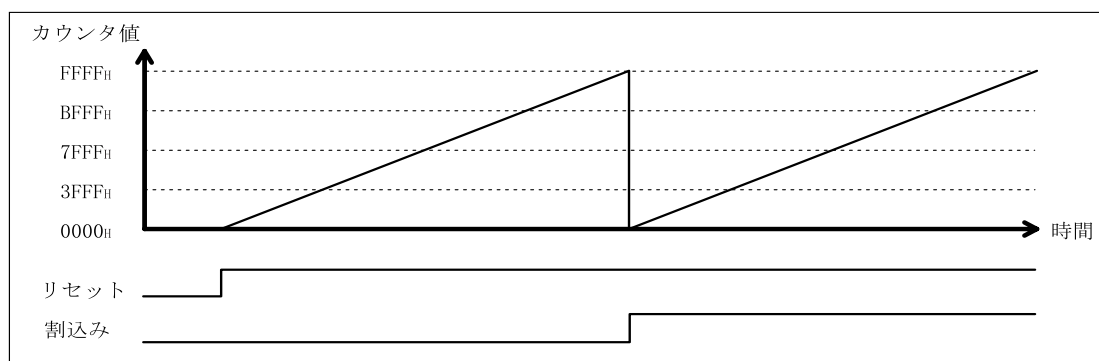


図 12.4-1 オーバーフローによる割込み

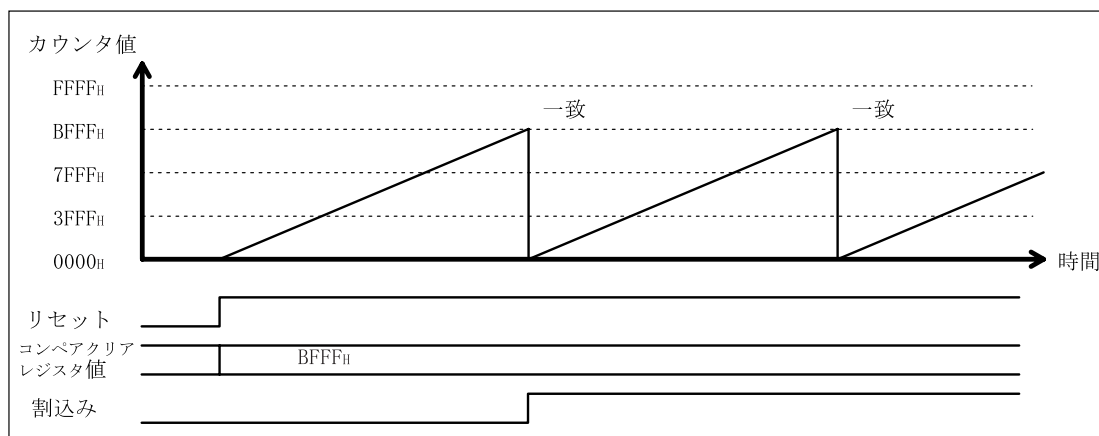


図 12.4-2 カウンタ値とコンペアクリアレジスタ値の一致による割込み

16ビットフリーランタイマのカウンタ値のクリアタイミング

16ビットフリーランタイマのカウンタ値は、リセット、タイマコントロールステータスレジスタ (TCCS) のクリアビット (SCLR) に"1"を設定、16ビットフリーランタイマのカウンタ値とコンペアクリアレジスタ (CPCLR) 値の一致 (TCCS:MODE="1"), およびタイマデータレジスタ (TCDT) へ"0000H"を設定した場合, "0000H"にクリアできます。

リセット, およびタイマコントロールステータスレジスタのクリアビットの設定によるカウンタ値のクリアは, クリア発生時点で非同期で行われますが, 16ビットフリーランタイマのカウンタ値とコンペアクリアレジスタ (CPCLR) 値の一致によるカウンタ値のクリアは, カウントタイミングに同期して行われます。

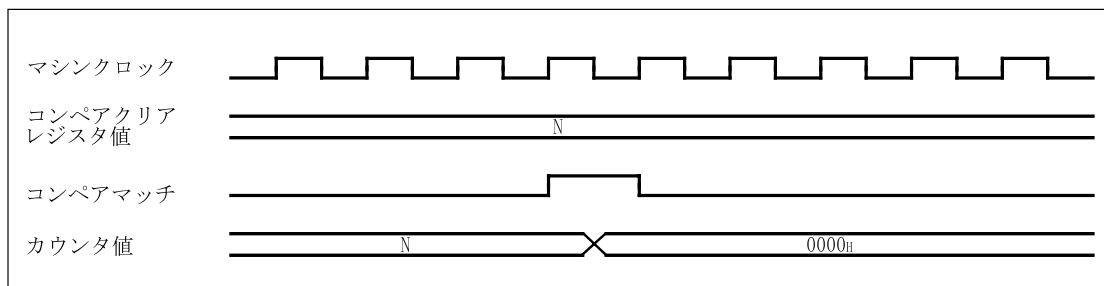


図 12.4-3 16ビットフリーランタイマのカウンタ値のクリアタイミング

16ビットフリーランタイマのカウンタ値のカウントタイミング

16ビットフリーランタイマのカウンタ値は、入力されたマシンクロック, または外部クロックによりカウントアップされます。外部クロックを設定した場合は, 立上りエッジでカウントされます。

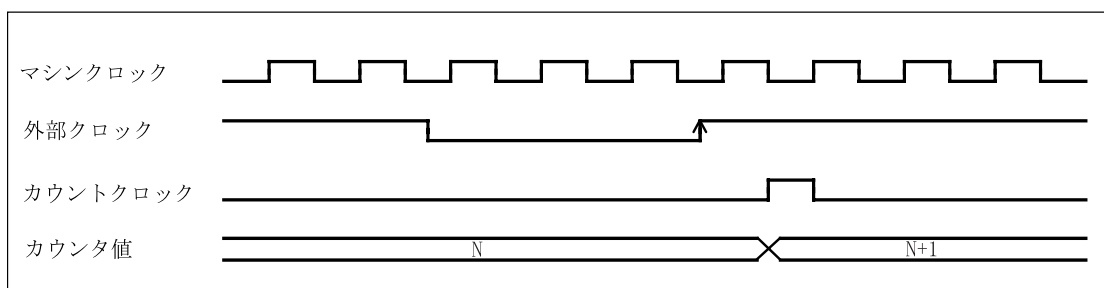


図 12.4-4 16ビットフリーランタイマのカウンタ値のカウントタイミング

12.4.2 アウトプットコンペア

アウトプットコンペアは，コンペアレジスタ（OCCP0～OCCP5）値と16ビットフリーランタイマのカウンタ値が一致した場合に，出力レベルを反転し割込みを出力できます。

アウトプットコンペアの動作

- コンペアコントロールレジスタ（OCS）のCMOD="0"の場合

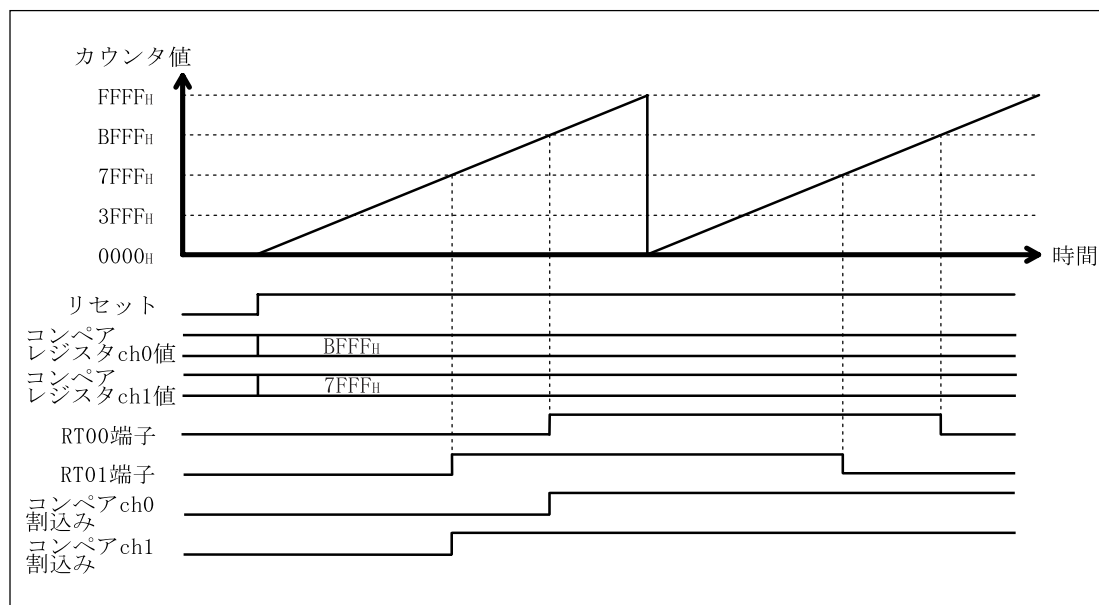


図 12.4-5 コンペアレジスタ（OCCP）ch0, ch1値を使用した場合の出力波形例

- コンペアコントロールレジスタ（OCS）のbit12:CMOD="1"の場合

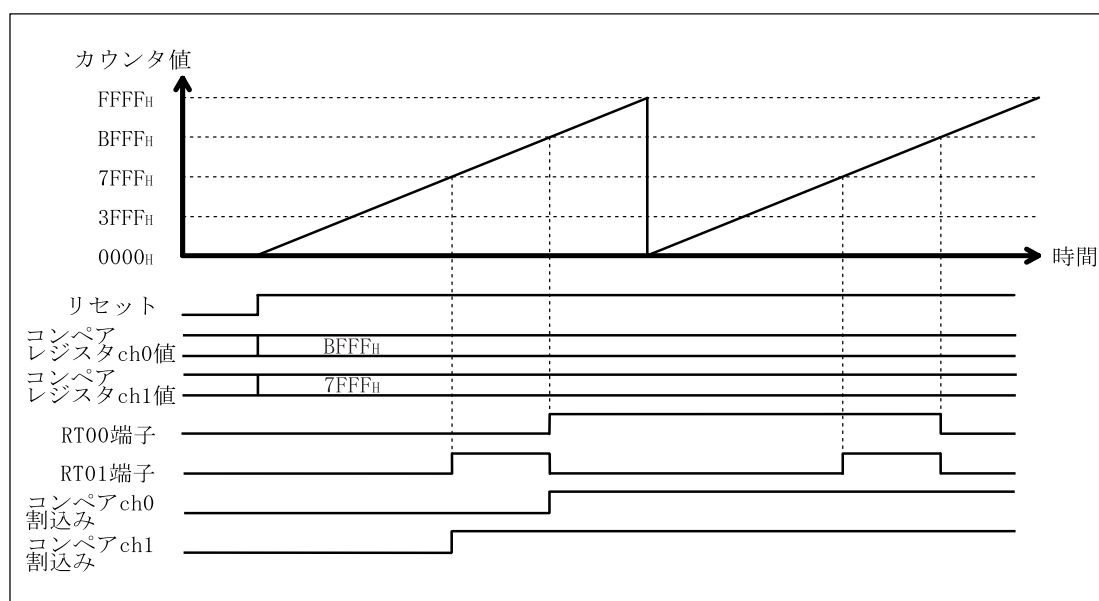


図 12.4-6 コンペアレジスタ（OCCP）ch0, ch1値を使用した場合の出力波形例

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、コンペアレジスタ（OCCP）値と16ビットフリーランタイマのカウンタ値が一致した場合に、コンペアマッチ信号を発生させ、出力レベルを反転させて割込みを出力できます。コンペアマッチした場合の出力反転タイミングは、16ビットフリーランタイマのカウンタ値のカウントタイミングに同期して行われます。コンペアレジスタ設定中は、カウンタ値とのコンペア動作を行いません。

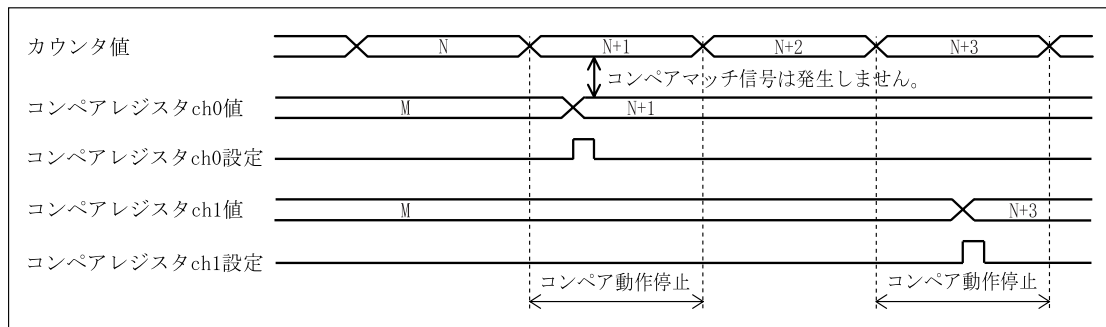


図 12.4-7 コンペアレジスタ設定中のコンペア動作

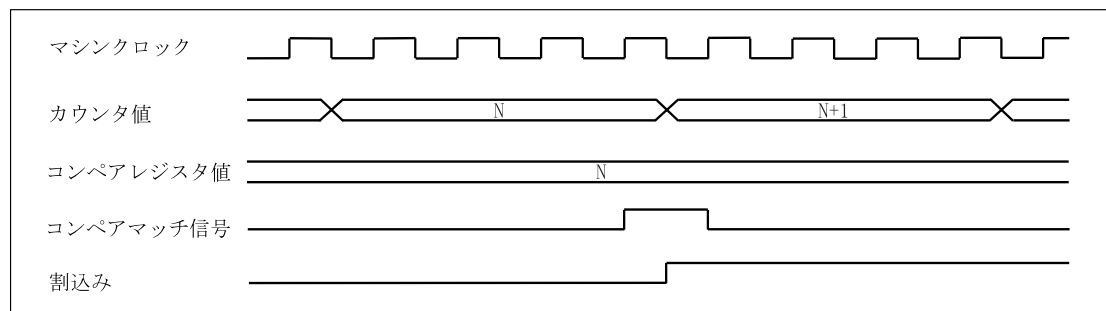


図 12.4-8 コンペア割込みタイミング

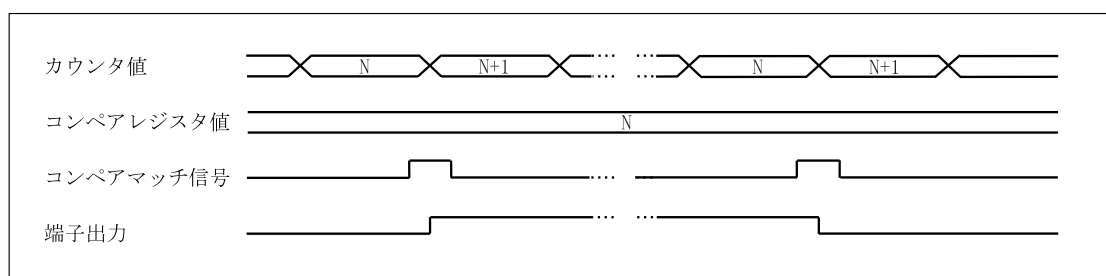


図 12.4-9 端子出力変化タイミング

12.4.3 インプットキャプチャ

インプットキャプチャは、外部入力端子（IN0～IN3）から入力された信号の有効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のカウンタ値をインプットキャプチャデータレジスタ（ICP）に取り込み、割込みを出力できます。

インプットキャプチャの動作

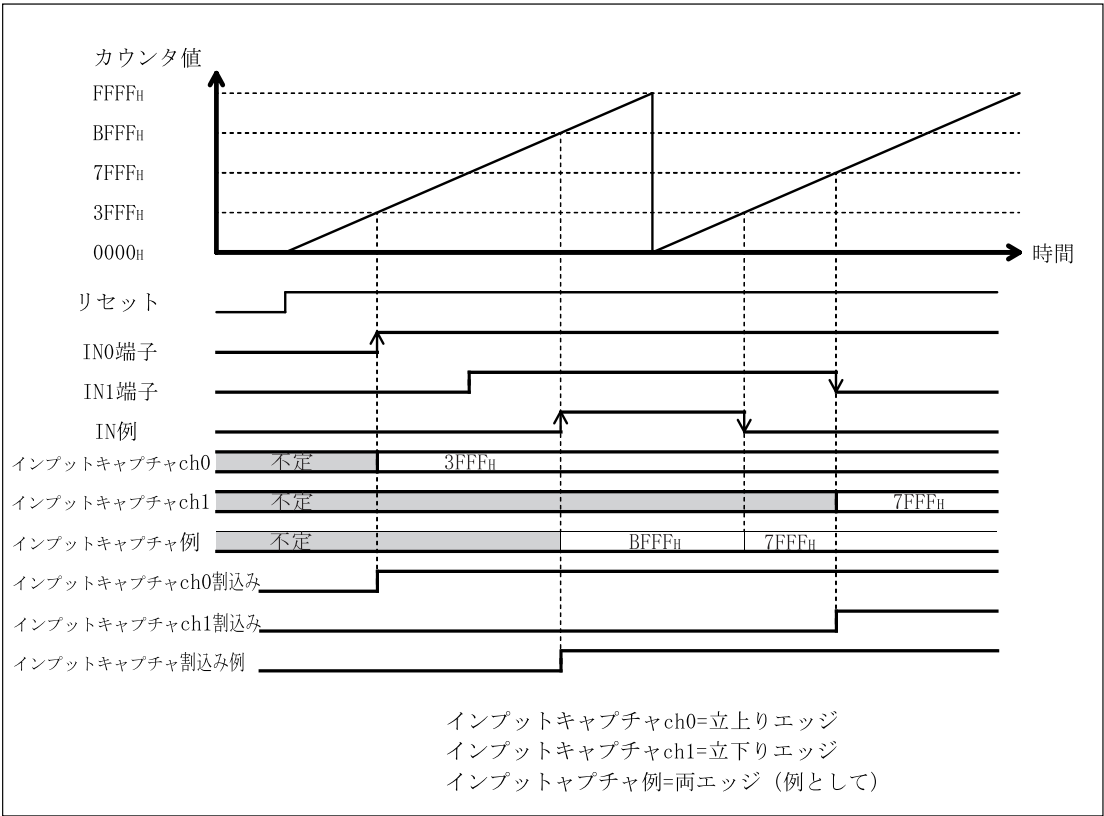


図 12.4-10 インプットキャプチャの取り込み例

インプットキャプチャの取込みタイミング

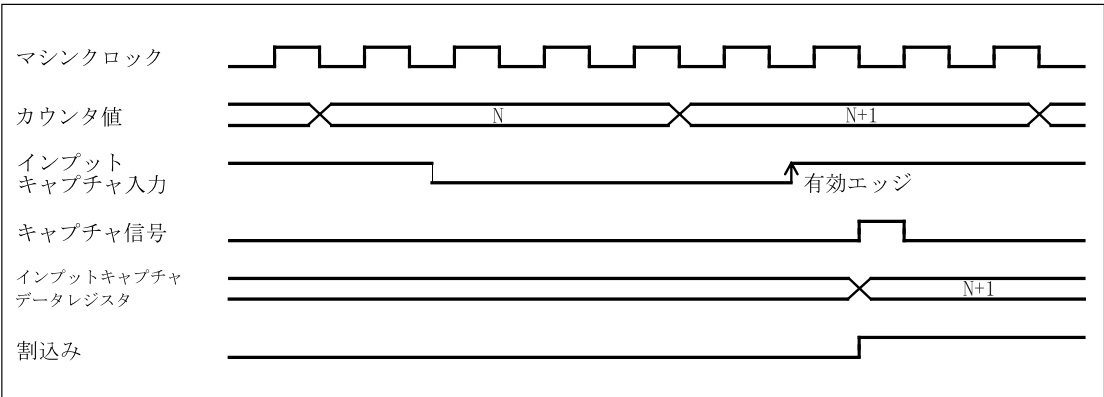


図 12.4-11 入力信号に対する取り込みタイミング

12.4.4 8/16ビットPPGタイマ

8/16ビットタイマには、8ビットPPGモード、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3種類の動作モードがあります。

ここでは、8/16ビットPPGタイマ ch0, ch1について説明します。8/16ビットPPGタイマ ch2, ch3の場合は、ch0をch2, ch1をch3と、8/16ビットPPGタイマ ch4, ch5の場合は、ch0をch4, ch1をch5と読み替えてください。





8/16ビットPPGタイマの動作

8ビット長のPPGユニットには、各チャンネルごとに8ビット長のPPGリロードレジスタ上位 (PRLH)、およびPPGリロードレジスタ下位 (PRL) があります。PPGリロードレジスタに設定された値は、8ビットダウンカウンタ(PCNT)に上位側/下位側交互にリロードされ、カウントクロックごとにダウンカウントします。カウンタ値のボロー発生でリロードした場合に、端子出力(PPG0～PPG5端子)の極性が反転します。端子出力(PPG端子)は、PPGリロードレジスタ (PRLH, PRL) 値に対応した"H"幅/"L"幅を持つパルスとなります。

また、出力極性をソフトウェアにより設定できますので、容易に出力が反転できます。動作開始は、PPGコントロールレジスタの設定、またはGATE信号の"H"入力となります。

- PPGC:SST1に"0", SST0に"0"を設定した場合に、PPGC1:PEN1に"1", PPGC0:PEN0に"1"が設定されると動作します。
- PPGC:SST1に"1", SST0に"1"を設定した場合に、GATE信号が"H"になると"H"期間動作します。

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	端子出力変化	
	正極性の場合	負極性の場合
PRLH PCNT	PPG ch0,ch1 [0 1]  Rise	PPG ch0,ch1 [1 0]  Fall
PRL PCNT	PPG ch0,ch1 [1 0]  Fall	PPG ch0,ch1 [0 1]  Rise

また、PPGC:PIE0に"1", PIE1に"1"を設定した場合、各チャンネルのカウント値のアンダーフロー ("00H" "FFH": 8ビットPPGモード, "0000H" "FFFFH": 16ビットPPGモード) によって割込みを出力します。

動作モード

8ビットPPGモード

8ビットPPGとして動作します。PPG0端子は、ch0のPPG出力となり、PPG1端子は、ch1のPPG出力となります。

8ビットプリスケラ+8ビットPPGモード

ch0を8ビットプリスケラとして動作させ、ch1をch0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として動作します。PPG0端子は、ch0のプリスケラ出力となり、PPG1端子は、ch1のPPG出力となります。

16ビットPPGモード

ch0とch1を連結させ、16ビットPPGとして動作させます。PPG0端子とPPG1端子は、16ビットPPG出力となります。

PPG動作

PPG ch0は、PPGC:SST0に"0"を設定した場合に、PPGC:PEN0に"1"が設定されると、PPGタイマはカウントを開始します。(ソフトウェア起動)、PPGC:SST0に"1"を設定した場合でも、GATE信号が"H"になると、PPGタイマはカウントを開始します。(ハードウェア起動)

PPG ch1は、PPGC:SST1に"0"を設定した場合に、PPGC:PEN1に"1"が設定されると、PPGタイマはカウントを開始します。(ソフトウェア起動)、PPGC:SST1に"1"を設定した場合でも、GATE信号が"H"になると、PPGタイマはカウントを開始します。(ハードウェア起動)

PPG ch0は、PPGC:PEN0に"0"、PPG ch1は、PPGC:PEN1に"0"を設定、またはGATE信号が"L"レベルになると、PPGタイマはカウントを停止します。停止中は、パルス出力を"L"レベルに保持します。(正極性の場合。負極性の場合は、"H"レベルに保持)

ソフトウェア起動で、8ビットプリスケラ + 8ビットPPGモードの場合は、ch0を動作禁止 (PPGC:PEN0="0")、ch1を動作許可 (PPGC:PEN1="1") に設定しないでください。16ビットPPGモードの場合は、PPGC:PEN0, PEN1は、同時に"0"、または"1"に設定してください。

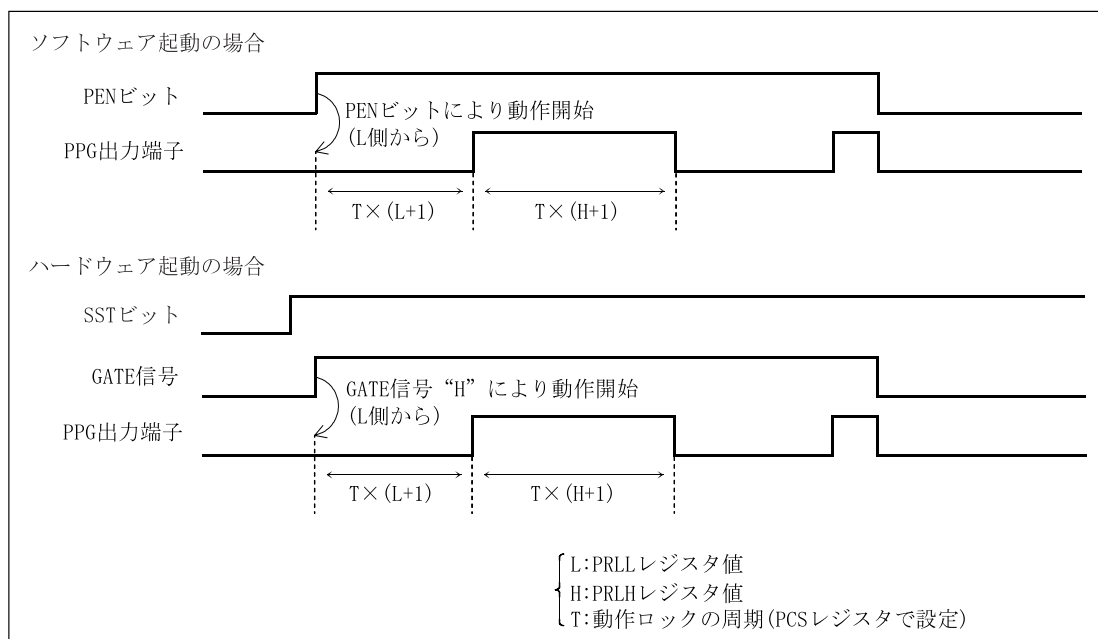


図 12.4-12 PPG動作出力パルス

リロード値とパルス幅の関係

PPGリロードレジスタ (PRLH, PRL) に設定した値 + 1 の値に、動作クロックの周期を掛けた値が、出力されるパルス幅となります。8ビットPPGモードのリロードレジスタ (PRLH, PRL) 値が"00H"の場合、および16ビットPPG動作モードのリロードレジスタ (PRLH, PRL) 値が"0000H"の場合は、動作クロック1周期分のパルス幅になります。また、8ビットPPG動作モードのリロードレジスタ (PRLH, PRL) 値が"FFH"の場合は、動作クロック256周期分のパルス幅になります。16ビットPPG動作モードのリロードレジスタ (PRLH, PRL) 値が"FFFFH"の場合は、動作クロック65,536周期分のパルス幅になります。

以下に、パルス幅の計算式を示します。

$P_H = T \times (H + 1)$ $P_L = T \times (L + 1)$

P_H : "H"パルス幅

P_L : "L"パルス幅

H : PRLHレジスタ設定値

L : PRLレジスタ設定値

T : 動作クロックの周期

< 注意事項 >

上記計算式は、出力極性を正極に設定した場合です。出力極性を負極に設定した場合は、 P_L 、 P_H の設定が逆になります。

動作クロックの設定

PPGタイマの8ビットダウンカウンタ (PCNT) に使用する動作クロックは、マシンクロックの1/16 ~ 1倍、およびタイムベースタイマからの入力クロックの6種類の動作クロックから設定できます。PPGクロック制御レジスタ (PCS) のbit4 ~ bit2 (PC02 ~ PC00) でPPG ch0の動作クロック、bit7 ~ bit5 (PC12 ~ PC10) でPPG ch1の動作クロックを設定します。

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モードの場合は、PPG ch1は、PPG ch0から動作クロックを受けて動作するため、PPGクロック制御レジスタ (PCS) のbit7 ~ bit5 (PC12 ~ PC10) の設定は、動作に影響しません。

< 注意事項 >

タイムベースタイマからの入力クロックを使用した場合、次のカウントで周期がずれます。

- ・トリガにより起動する最初のカウント
- ・ストップモード解除後の最初のカウント
- ・8ビットプリスケラ + 8ビットPPGモードのPPG ch0が動作状態、PPG ch1が停止状態で、PPG ch1の起動を行った場合の最初のカウント
- ・動作中にタイムベースタイマカウンタのクリア (TBTC:TBR="1") を行った場合のカウント

パルスの端子出力の制御

8/16ビットPPGタイマの動作によって生成されたパルス出力は、外部端子（PPG端子）に出力させることができます。外部端子への出力許可は、PPGコントロールレジスタ（PPGC）の設定で行います。PPGC0：POE0に"1"を設定した場合、PPG0端子にPPG ch0のパルス出力を、PPGC1：POE1に"1"を設定した場合、PPG1端子にPPG ch1のパルス出力を許可します。PPGC0：POE0に"0"、PPGC1：POE1に"0"を設定した場合は、外部端子（PPG0端子とPPG1端子）へのパルス出力は許可されず、PPG0端子とPPG1端子は、入出力ポートとして動作します。

8ビットプリスケラ+8ビットPPGモードでは、PPG0端子は、8ビットプリスケラのトグル波形を出力し、PPG1端子は、8ビットPPGのパルス出力を行います。16ビットPPGモードでは、PPG0、PPG1端子は同じ波形を出力しますので、どちらの外部端子を出力許可しても、同じ出力を得ることができます。

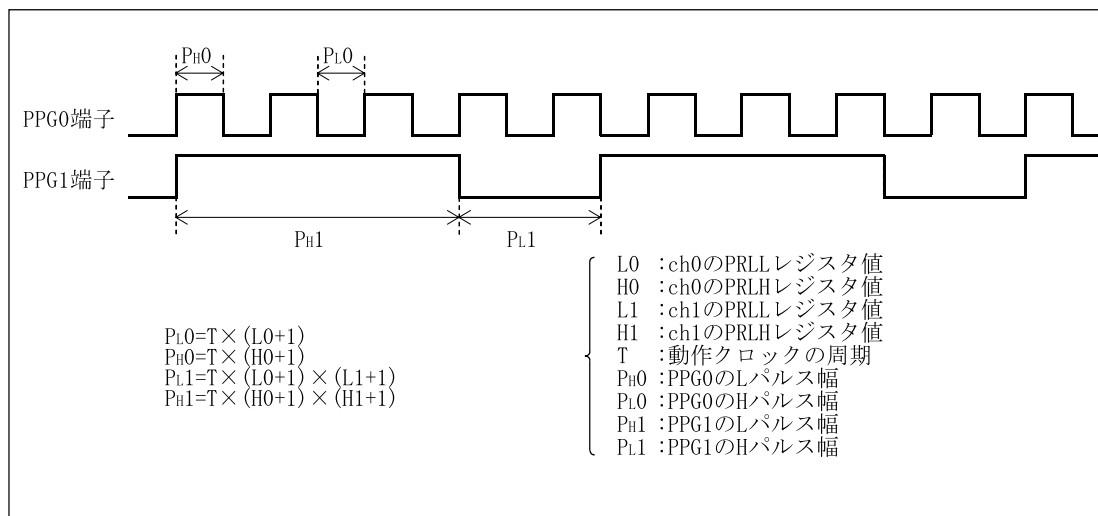


図 12.4-13 8ビットプリスケラ+8ビットPPGモード出力波形

< 注意事項 >

8ビットプリスケラ+8ビットPPGモードでは、ch0, ch2, ch4のPRLHレジスタ値とPRLレジスタ値は、同じ値を設定してください。

割込み

PPGタイマのカウント値のアンダーフローにより、8/16ビットPPGタイマの割込み要求を出力します。

8ビットPPGモード、および8ビットプリスケラ+8ビットPPGモードでは、8ビットカウンタ値のアンダーフロー（"00_H" "FF_H"）により、割込み要求を出力します。

16ビットPPGモードでは、16ビットカウンタ値のアンダーフロー（"0000_H" "FFFF_H"）へのアンダーフローにより、割込み要求を出力します。割込み要求を出力した場合は、PPGタイマ割込み要求フラグビット（PPGC1:PUF1, PPGC0:PUF0）に、同時に"1"がセットされます。割込み要求をクリアする場合は、同時に"0"を設定してください。

PPGリロードレジスタへのライトタイミング

8ビットPPGモード, および8ビットプリスケラ+8ビットPPGモードの場合, PPGリロードレジスタ (PRLH, PRLL) の設定は, ワード転送命令を使用してください。バイト転送命令2回で設定した場合は, タイミングによって, パルス幅が遅れます。

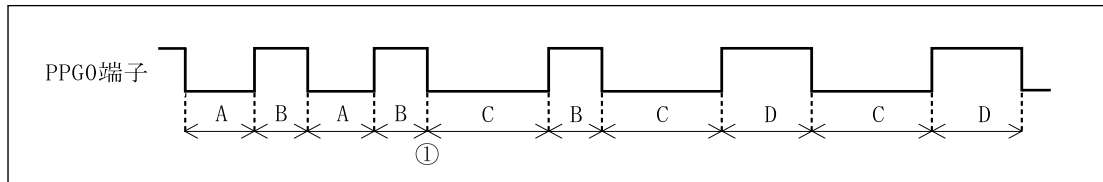


図 12.4-14 ライトタイミングタイムチャート

図 12.4-14に示すタイムチャートで, のタイミングの前で, PPGリロードレジスタ下位 (PRLL) 値をAからCに設定し, のタイミングの後で, PPGリロードレジスタ上位 (PRLH) 値をBからDに設定した場合は, のタイミングでのPPGリロードレジスタ (PRLH, PRLL) 値は, PRLLレジスタ値=C, PRLHレジスタ値=Bのため, 設定直後は, 下位側のカウンタ数C, 上位側のカウンタ数Bのパルスが発生し, パルス幅の変更が遅れます。

16ビットPPGモードを使用する場合は, PPGリロードレジスタ (PRLH, PRLL) ch0, ch1をロングワード転送で設定するか, またはch0 ch1の順にワード転送で設定してください。16ビットPPGモードでは, PPGリロードレジスタ (PRLH, PRLL) ch0の設定値を一時テンポラリラッチに保存し, PPGリロードレジスタ (PRLH, PRLL) ch1の設定に同期して, PPGリロードレジスタ (PRLH, PRLL) ch0がセットされます。

図 12.4-15に示すように, 8ビットPPGモード, および8ビットプリスケラ+8ビットPPGモードでは, PPGリロードレジスタ (PRLH, PRLL) ch0, ch1の設定をチャンネルごとに行うことができます。

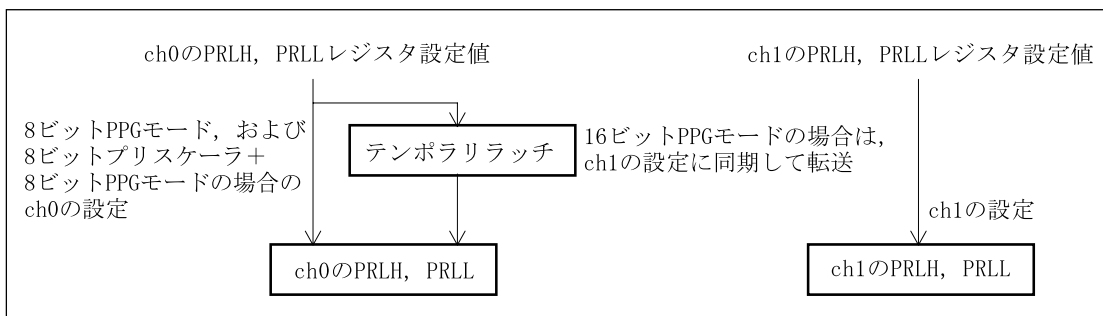


図 12.4-15 PPGリロードレジスタ (PRLH, PRLL) 設定のブロック図

12.4.5 波形生成部

波形生成部は，リアルタイムアウトプット(RT)，8/16ビットPPGタイマ，および8ビットタイマにより，様々なタイミング波形が生成できます。

波形生成部の動作

波形生成部では，次のようなタイミング波形を生成できます。

- 8ビットタイマをデッドタイムタイマとして使用することにより，リアルタイムアウトプット(RT1, RT3, RT5)，およびPPGタイマ(PPG1, PPG3, PPG5)のパルス出力に，ノンオーバーラップ時間のディレイを付けた，ノンオーバーラップ波形が生成できます。(デッドタイム機能)
- リアルタイムアウトプット(RT)の立上りエッジで，8ビットタイマを起動し，8ビットタイマのカウント値と8ビットリロードレジスタ(TMRR)値が一致するまで，設定(SIGCR:PGS1, PGS0)したPPGタイマを出力します。(GATE機能)

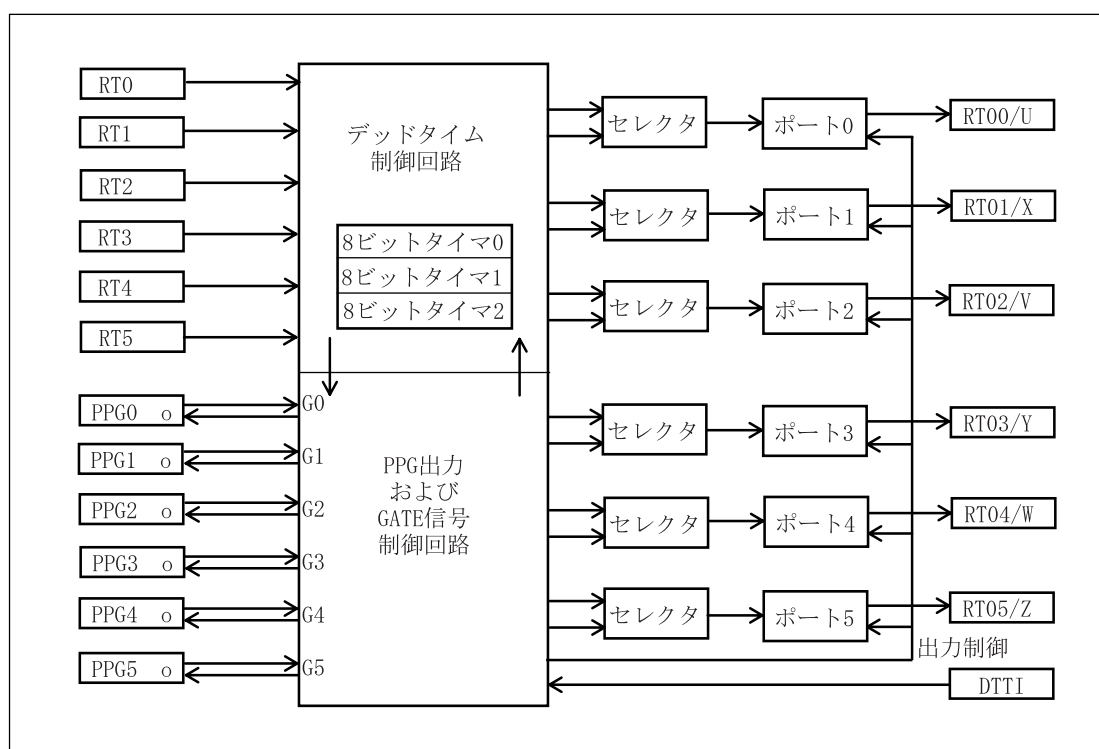


図 12.4-16 波形生成部の動作

12.4.6 デッドタイム制御回路部の動作

デッドタイム制御回路部は、リアルタイムアウトプット(RT1,RT3,RT5)、およびPPGタイマ(PPG1,PPG3,PPG5)のパルス出力に、ノンオーバーラップ時間のディレイを付けた、ノンオーバーラップ波形を生成します。

リアルタイムアウトプット(RT1,RT3,RT5)のノンオーバーラップ波形を生成する場合

- 波形出力極性を正極に設定(DTCR:DMOD="0")したノンオーバーラップ波形は、リアルタイムアウトプット(RT1,RT3,RT5)、およびリアルタイムアウトプット(RT1,RT3,RT5)反転パルスの立上りに、8ビットリロードレジスタ(TMRR)に設定したノンオーバーラップ時間のディレイを付けます。リアルタイムアウトプット(RT1,RT3,RT5)のパルス幅が、設定したノンオーバーラップ時間より小さい場合、8ビットタイマは、次のリアルタイムアウトプットのエッジで、"00H"からカウントを再開します。

【各レジスタの設定条件】

```

TCDT : 0000000000000000b
CPCLR : XXXXXXXXXXXXXXXXb (周期の設定)
TCCS : X--XXXXXXXXX0XXXXb
OCS : ---1XXXXXXXX--11b
OCCP : XXXXXXXXXXXXXXXXb (コンペア値の設定)
DTCR : 00000100b
TMRR : XXXXXXXXb (ノンオーバーラップ時間の設定)
SIGCR : XXXXXXXXb (DTTI入力、および8ビットタイマ動作クロックの設定)
X : 動作に合わせて設定してください。
- : 未定義ビット
  
```

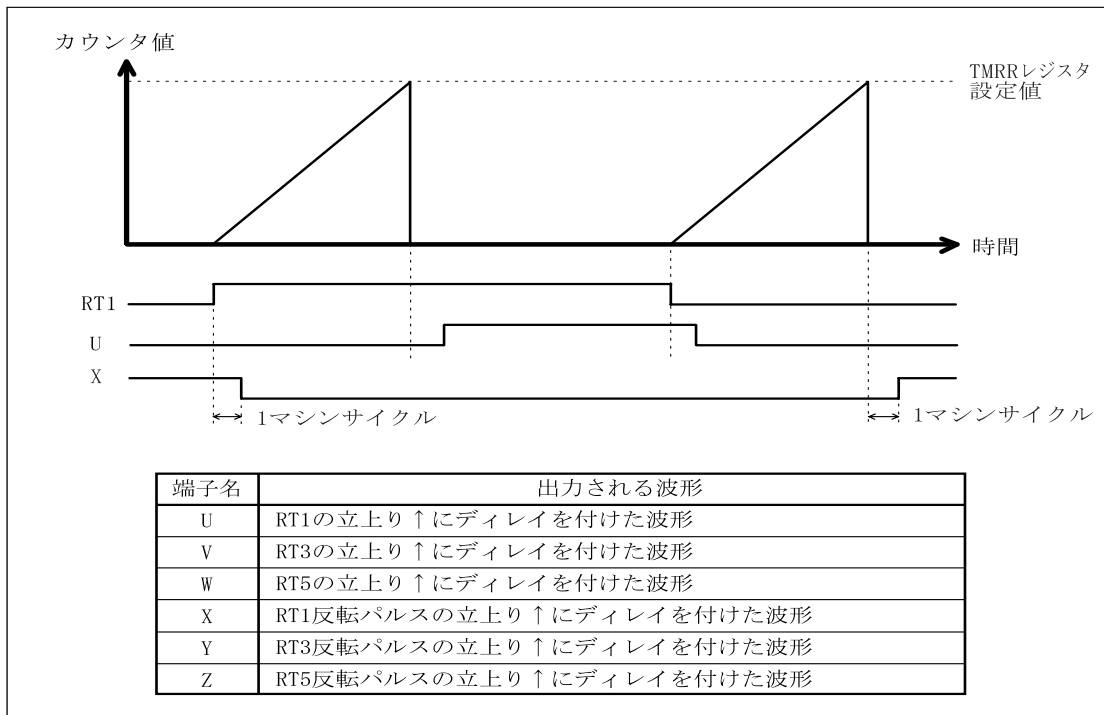


図 12.4-17 リアルタイムアウトプット正極性ノンオーバーラップ波形の生成

- 波形出力極性を負極に設定 (DTCR:DMOD="1") したノンオーバーラップ波形は、リアルタイムアウトプット (RT1,RT3,RT5) の反転パルス、およびリアルタイムアウトプット (RT1,RT3,RT5) の立下りに、8ビットリロードレジスタ (TMRR) に設定したノンオーバーラップ時間のディレイを付けます。リアルタイムアウトプット (RT1,RT3,RT5) のパルス幅が設定したノンオーバーラップ時間より小さい場合、8ビットタイマは、次のリアルタイムアウトプットのエッジで、"00H" からカウントを再開します。

【各レジスタの設定条件】

TCDT : 0000000000000000B
CPCLR : XXXXXXXXXXXXXXXXB (周期の設定)
TCCS : X--XXXXXXXXX0XXXXB
OCS : ----1XXXXXXXX--11B
OCCP : XXXXXXXXXXXXXXXXB (コンペア値の設定)
DTCR : 10000100B
TMRR : XXXXXXXXB (ノンオーバーラップ時間の設定)
SIGCR : XXXXXXXXB (DTTI入力, および8ビットタイマ動作クロックの設定)

X: 動作に合わせて設定してください。

- : 未定義ビット

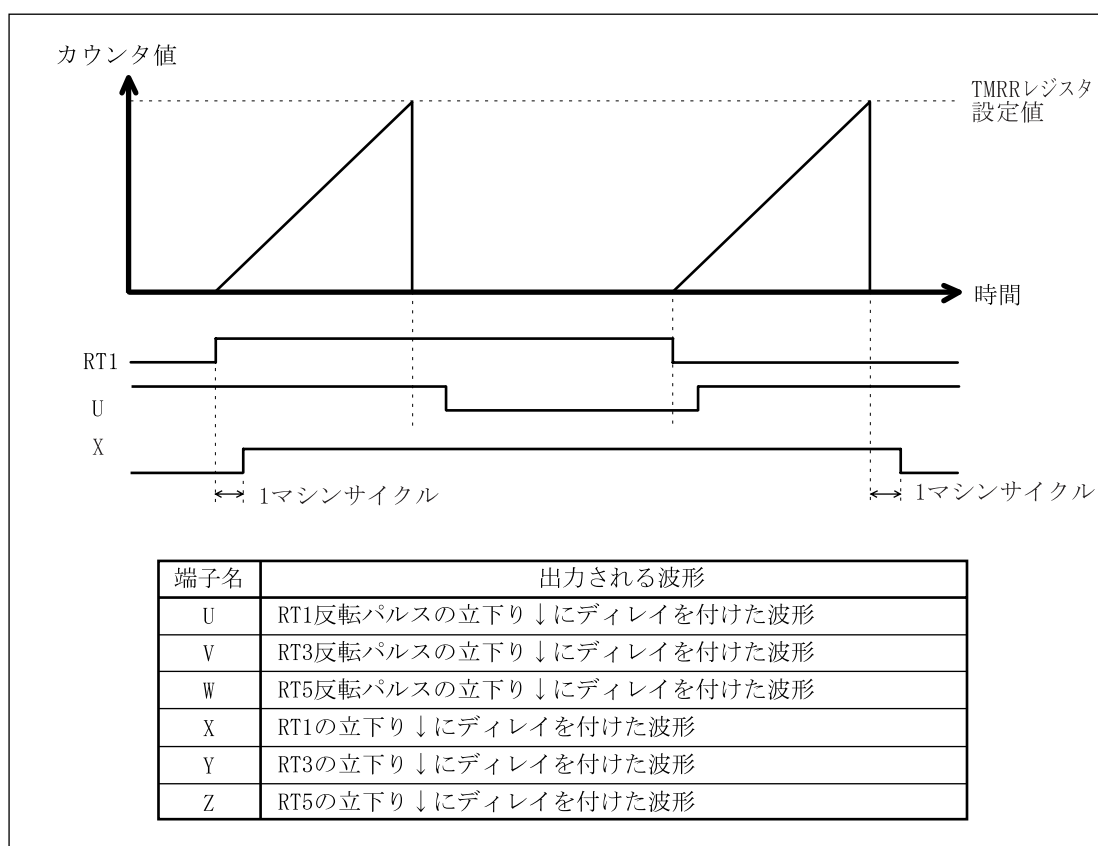


図 12.4-18 リアルタイムアウトプット負極性ノンオーバーラップ波形の生成

PPGタイマ（PPG1,PPG3,PPG5）のノンオーバーラップ波形を生成する場合

- 波形出力極性を正極に設定（DTCR:DMOD="0"）したノンオーバーラップ波形は，PPGタイマ（PPG1,PPG3,PPG5）のパルス，およびPPGタイマ（PPG1,PPG3,PPG5）反転パルスの立上りに，8ビットリロードレジスタ（TMRR）に設定したノンオーバーラップ時間のディレイを付けます。PPGタイマ（PPG1,PPG3,PPG5）のパルス幅が，設定したノンオーバーラップ時間より小さい場合，8ビットタイマは，次のPPGタイマのパルスエッジで"00H"からカウントを再開します。

【各レジスタの設定条件】

TCDT : 0000000000000000b
 CPCLR : XXXXXXXXXXXXXXXb (周期の設定)
 TCCS : X--XXXXXXXXX0XXXb
 OCS : ---1XXXXXXXX--11b
 OCCP : XXXXXXXXXXXXXXXb (コンペア値の設定)
 DTCR : 01000111b
 TMRR : XXXXXXXb (ノンオーバーラップ時間の設定)
 SIGCR : XXXXXXXb (DTTI入力，および8ビットタイマ動作クロックの設定)

X : 動作に合わせて設定してください。

- : 未定義ビット

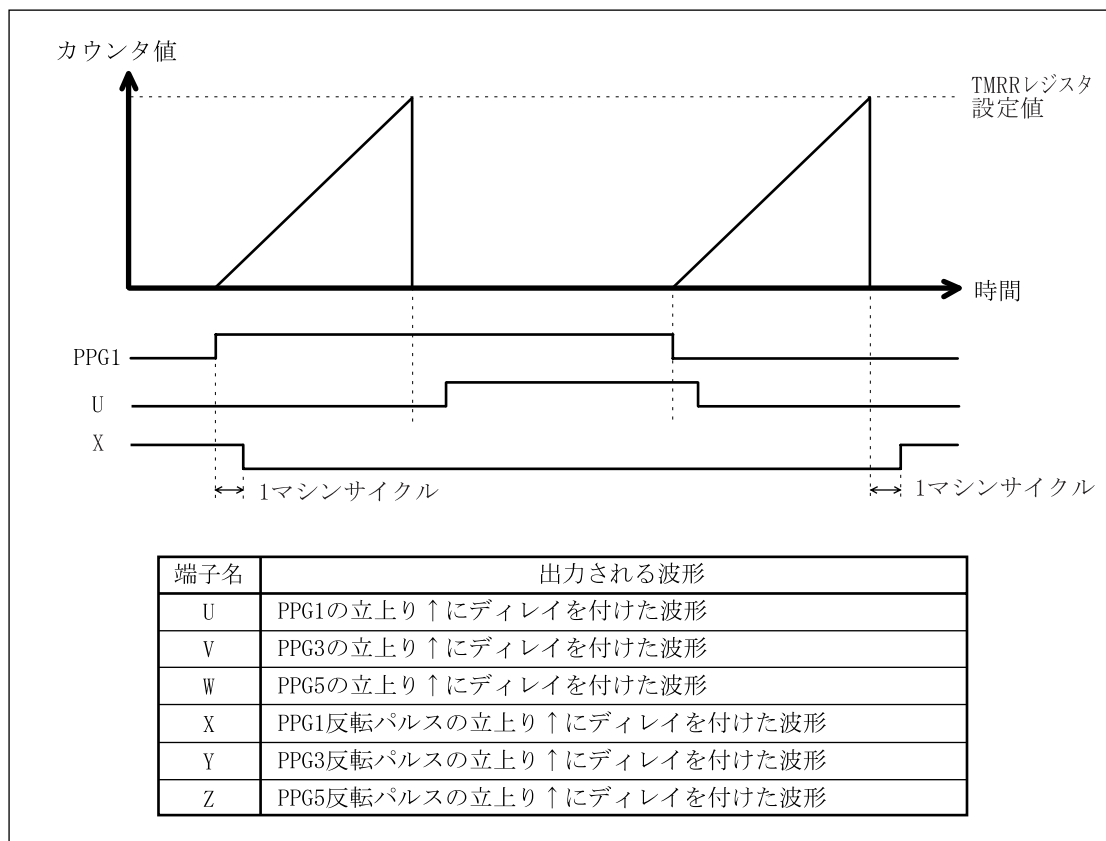


図 12.4-19 PPGタイマ正極性ノンオーバーラップ波形の生成

- 波形出力極性を負極に設定（DTCR:DMOD="1"）したノンオーバーラップ波形は，PPGタイマ（PPG1,PPG3,PPG5）反転パルス，およびPPGタイマ（PPG1,PPG3,PPG5）のパルスの立下りに，8ビットリロードレジスタ（TMRR）に設定したノンオーバーラップ時間のディレイを付けます。PPGタイマ（PPG1,PPG3,PPG5）のパルス幅が，設定したノンオーバーラップ時間より小さい場合，8ビットタイマは，次のPPGタイマのパルスエッジで"00H"からカウントを再開します。

【各レジスタの設定条件】

TCDT : 0000000000000000b
CPCLR : XXXXXXXXXXXXXXXXb (周期の設定)
TCCS : X--XXXXXXXXX0XXXXb
OCS : ---1XXXXXXXX--11b
OCCP : XXXXXXXXXXXXXXXXb (コンペア値の設定)
DTCR : 11000111b
TMRR : XXXXXXXXb (ノンオーバーラップ時間の設定)
SIGCR : XXXXXXXXb (DTTI入力，および8ビットタイマ動作クロックの設定)

X : 動作に合わせて設定してください。

- : 未定義ビット

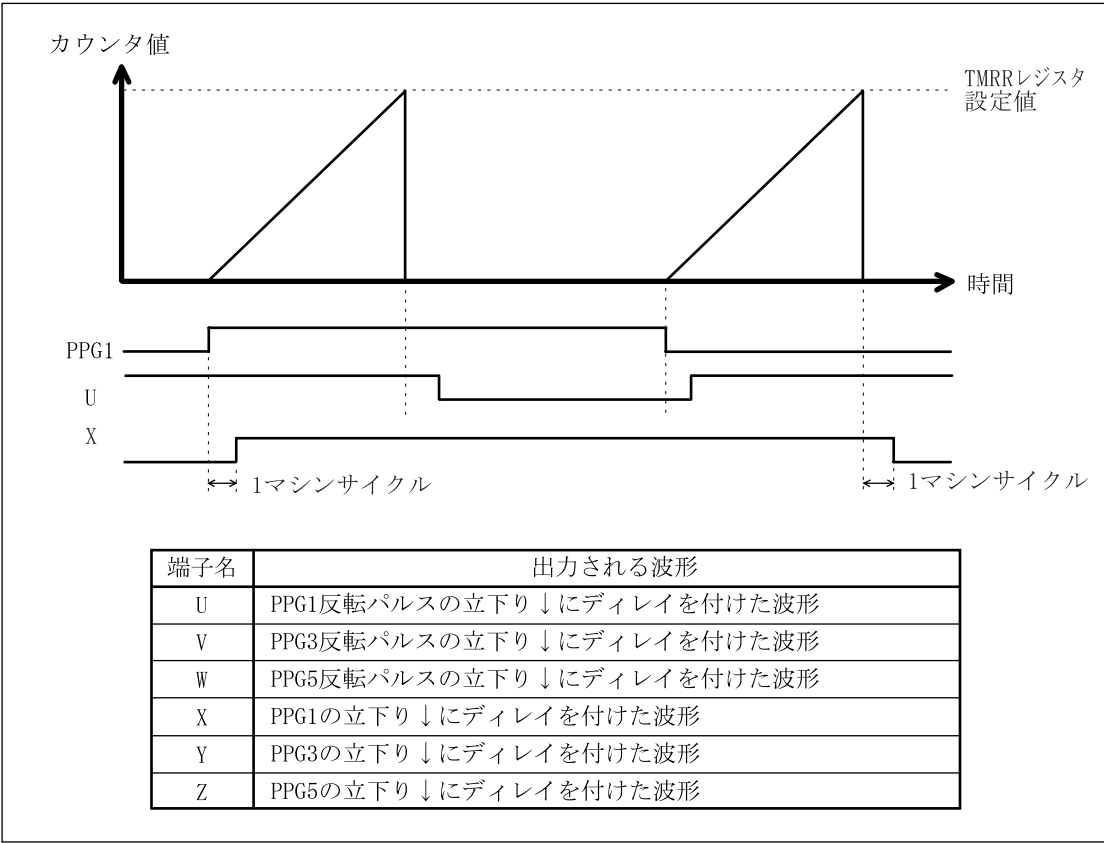


図 12.4-20 PPGタイマ負極性ノンオーバーラップ波形の生成

12.4.7 PPG出力およびGATE信号制御回路部の動作

PPG出力，およびGATE信号制御回路部は，8ビットタイマコントロールレジスタ（DTCR）のGATE信号出力設定ビット（GTEN），パルス出力許可ビット（PGEN）に"1"を設定した場合に，波形制御レジスタ（SIGCR）のPPGタイマ設定ビット（PGS1, PGS0）で設定したPPGタイマのパルスをRT0端子に出力します。また，PPGタイマを制御するGATE信号は，リアルタイムアウトプット（RT），および8ビットタイマにより生成します。

リアルタイムアウトプット（RT0～RT5）によりPPG出力およびGATE信号を生成する場合

【各レジスタの設定条件】

```

TCDT : 0000000000000000B
CPCLR : XXXXXXXXXXXXXXXXB (周期の設定)
TCCS : X--XXXXXXXXX0XXXB
OCS : ---0XXXXXXXX--11B
OCCP : XXXXXXXXXXXXXXXXB (コンペア値の設定)
DTCR : 01100001B
TMRR : XXXXXXXXB
SIGCR : XXXXXXXXB (DTTI入力，および8ビットタイマ動作クロックの設定)
X : 動作に合わせて設定してください。
- : 未定義ビット
  
```

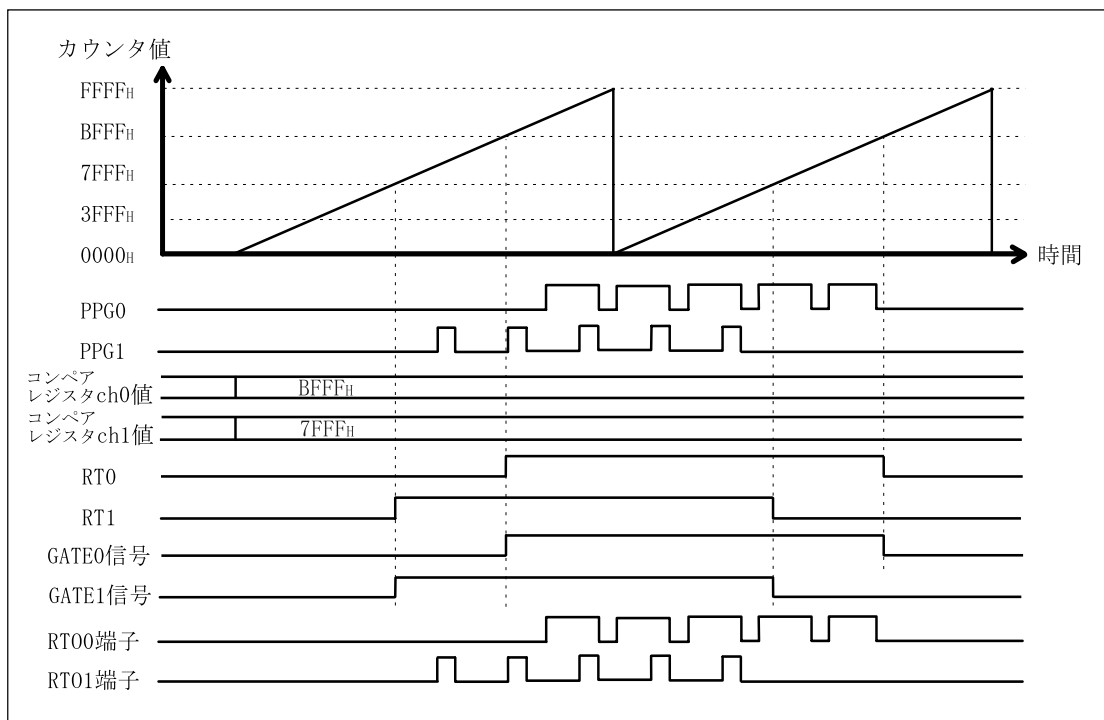


図 12.4-21 リアルタイムアウトプット（RT）によるPPG出力，およびGATE信号の生成

8ビットタイマによりPPG出力およびGATE信号を生成する場合

【各レジスタの設定条件】

TCDT : 0000000000000000b
CPCLR : XXXXXXXXXXXXXXXXb (周期の設定)
TCCS : X--XXXXXX0X0XXb
OCS : ---0XXXXXXXX--11b
OCCP : XXXXXXXXXXXXXXXXb (コンペア値の設定)
DTCR : 01100010b
TMRR : XXXXXXXb (GATE時間の設定)
SIGCR : XXXXXXXb (DTTI入力, および8ビットタイマ動作クロックの設定)

X : 動作に合わせて設定してください。

- : 未定義ビット

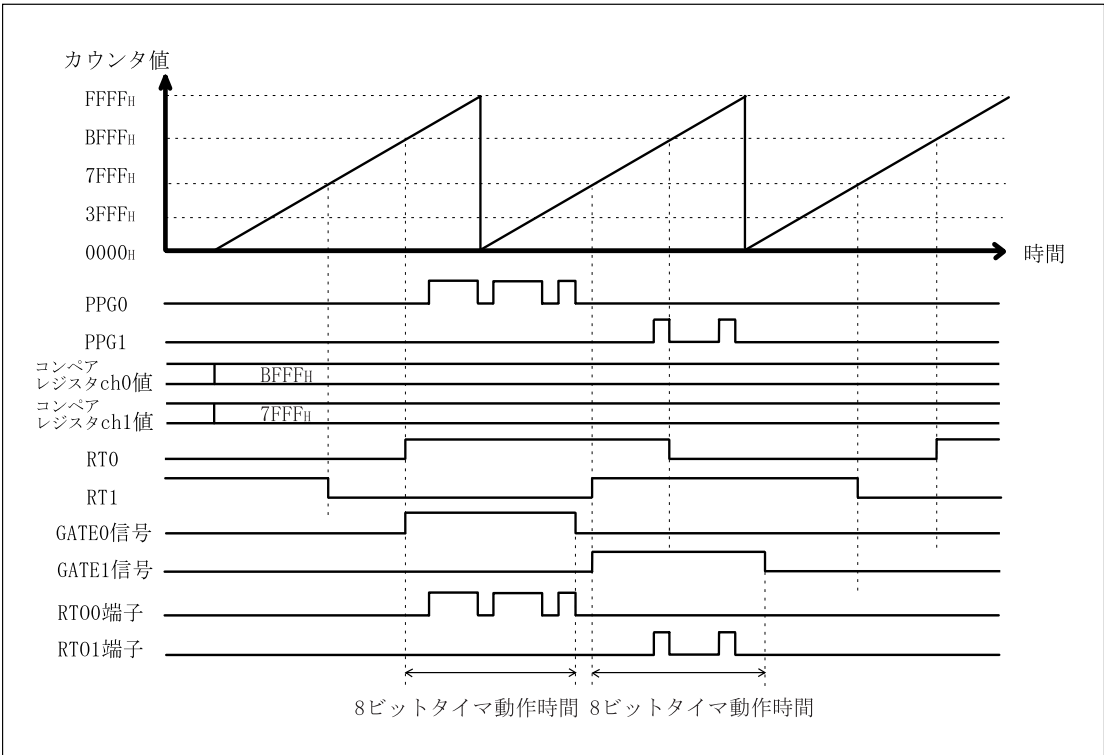


図 12.4-22 8ビットタイマによるPPG出力, およびGATE信号の生成

< 注意事項 >

RT0, RT1は8ビットタイマ0, RT2, RT3は8ビットタイマ1, RT4, RT5は8ビットタイマ2に対応します。8ビットタイマ動作中は,再起動させないでください。再起動した場合は,出力中のGATE信号が延長されます。

12.4.8 DTTI端子入力制御の動作

波形制御レジスタ (SIGCR) の出力レベル制御許可ビット (DTIE) を "1" に設定することにより, DTTI端子入力によるRT0端子の波形出力の制御を許可します。DTTI端子に, 波形制御レジスタ (SIGCR) の有効入力レベル設定ビット (DTIL) で設定した有効入力レベルを入力すると, RT0端子出力をポート3データレジスタ (PDR3) で設定したレベルに固定します。

DTTI端子入力制御の動作

DTTI端子入力により, ポート3データレジスタ (PDR3) で設定したレベルに固定されている間も, 8ビットタイマは停止せず, 波形生成動作を継続しますが, RT0端子には出力されません。

【各レジスタの設定条件】

```
TCDT : 0000000000000000b
CPCLR : XXXXXXXXXXXXXXXb (周期の設定)
TCCS : X--XXXXXXXXX0XXXb
OCS : ---0XXXXXXXX--11b
OCCP : XXXXXXXXXXXXXXXb (コンペア値の設定)
DTCR : 0000100b
TMRR : XXXXXXXb (ノンオーバーラップ時間の設定)
SIGCR : 110XXXXb (DTTI入力, および8ビットタイマ動作クロックの設定)
PDR3 : 00XXXX00b (インアクティブレベルの設定)
```

X : 動作に合わせて設定してください。

- : 未定義ビット

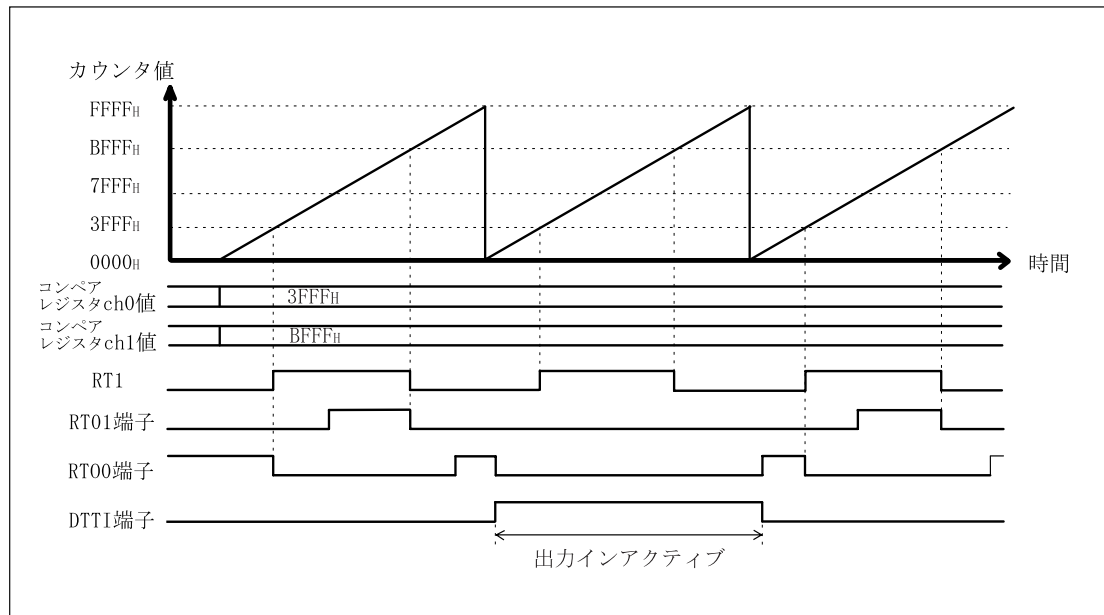


図 12.4-23 DTTI端子入力を許可した場合の動作

DTTI端子のノイズキャンセル機能

波形制御レジスタ (SIGCR) のノイズキャンセル機能設定ビット (NRSL) に "1" を設定することにより、DTTI端子入力のノイズキャンセル機能を使用することができます。ノイズキャンセル機能を有効にした場合は、ノイズキャンセル回路により、RT0端子の出力タイミングが、約4マシンサイクル遅れます。

ノイズキャンセル回路は、周辺クロックを使用し動作するため、ストップモードで発振クロックを停止させると、DTTI端子の入力は無効となります。

第13章 UART

この章では、MB90560/565シリーズのUARTの機能と動作について説明します。

- 13.1 UARTの概要
- 13.2 UARTの構成
- 13.3 UARTの端子
- 13.4 UARTのレジスタ
- 13.5 UARTの割込み
- 13.6 UARTのボーレート
- 13.7 UARTの動作説明
- 13.8 UART使用上の注意

13.1 UARTの概要

UARTは、外部装置と同期通信もしくは非同期通信（調歩同期）をするための、汎用のシリアルデータ通信インターフェースです。双方向通信機能（ノーマルモード）、マスタ/スレーブ型通信機能（マルチプロセッサモード：マスタ側だけサポート）があります。

UARTの機能

UARTの機能

UARTは、他のCPUや周辺装置とシリアルデータの送受信をする汎用シリアルデータ通信インターフェースで、表 13.1-1に示す機能をもっています。

表 13.1-1 UARTの機能

	機 能
データバッファ	全2重ダブルバッファ
転送モード	<ul style="list-style-type: none"> ・クロック同期（スタート/ストップビットなし） ・クロック非同期（調歩周期）
ボーレート	<ul style="list-style-type: none"> ・最大2MHz（マシクロック16MHz時） ・専用ボーレートジェネレータによるボーレート ・外部クロック（SCK0/SCK1端子入力のクロック）によるボーレート ・内部クロック（16ビットリロードタイマから供給されるクロック）によるボーレート ・ボーレートは全8種類から設定可能
データ長	<ul style="list-style-type: none"> ・7ビット（非同期ノーマルモード時のみ） ・8ビット
信号方式	NRZ（Non Return to Zero）方式
受信エラー検出	<ul style="list-style-type: none"> ・フレーミングエラー ・オーバーランエラー ・パリティエラー（マルチプロセッサモード時は検出不可）
割込み要求	<ul style="list-style-type: none"> ・受信割込み（受信完了，受信エラー検出） ・送信割込み（送信完了） ・送受信とも拡張インテリジェントI/Oサービス（EI²OS）の対応あり
マスタ/スレーブ型通信機能（マルチプロセッサモード）	1（マスタ）対n（スレーブ）間の通信が可能（マスタ側だけサポート）

< 注意事項 >

UARTは、クロック同期転送時にスタートビット/ストップビットは付加されず、データが転送されます。

図 13.1-1 UARTの動作モード

動作モード		データ長		同期方式	ストップビット長
		パリティなし	パリティあり		
0	ノーマルモード	7ビットまたは8ビット		非同期	1ビット または 2ビット ^{*2}
1	マルチプロセッサモード	8 + 1 ^{*1}		非同期	
2	ノーマルモード	8		同期	なし

：設定不可

*1： " + 1" は通信制御用に使われるアドレス/データ設定ビット（A/D）です。

*2： 受信時のストップビット長は1ビットのみ検出可能

UARTに関連する割り込みとEI²OS表 13.1-2 UARTに関連する割り込みとEI²OS

割り込み要因	割り込み番号	割り込み制御レジスタ		ベクタテーブルアドレス			EI ² OS
		レジスタ名	アドレス	下位	上位	バンク	
UART1 受信割り込み	#37 (25 _H)	ICR13	0000BD _H	FFFF68 _H	FFFF69 _H	FFFF6A _H	
UART1 送信割り込み	#38 (26 _H)	ICR13	0000BD _H	FFFF64 _H	FFFF65 _H	FFFF66 _H	
UART0 受信割り込み	#39 (27 _H)	ICR14	0000BE _H	FFFF60 _H	FFFF61 _H	FFFF62 _H	
UART0 送信割り込み	#40 (28 _H)	ICR14	0000BE _H	FFFF5C _H	FFFF5D _H	FFFF5E _H	

: UARTの受信エラー検出によるEI²OS停止機能付

: ICR13, ICR14を共有する割り込み要因を使用しない場合に使用可能

13.2 UARTの構成

UARTは、以下の11種類のブロックで構成されています。

- ・クロックセクタ
- ・受信制御回路
- ・送信制御回路
- ・受信状態判定回路
- ・受信用シフトレジスタ
- ・送信用シフトレジスタ
- ・モードレジスタ (SMR0/SMR1)
- ・制御レジスタ (SCR0/SCR1)
- ・ステータスレジスタ (SSR0/SSR1)
- ・インプットデータレジスタ (SIDR0/SIDR1)
- ・アウトプットデータレジスタ (SODR0/SODR1)

UARTのブロックダイアグラム

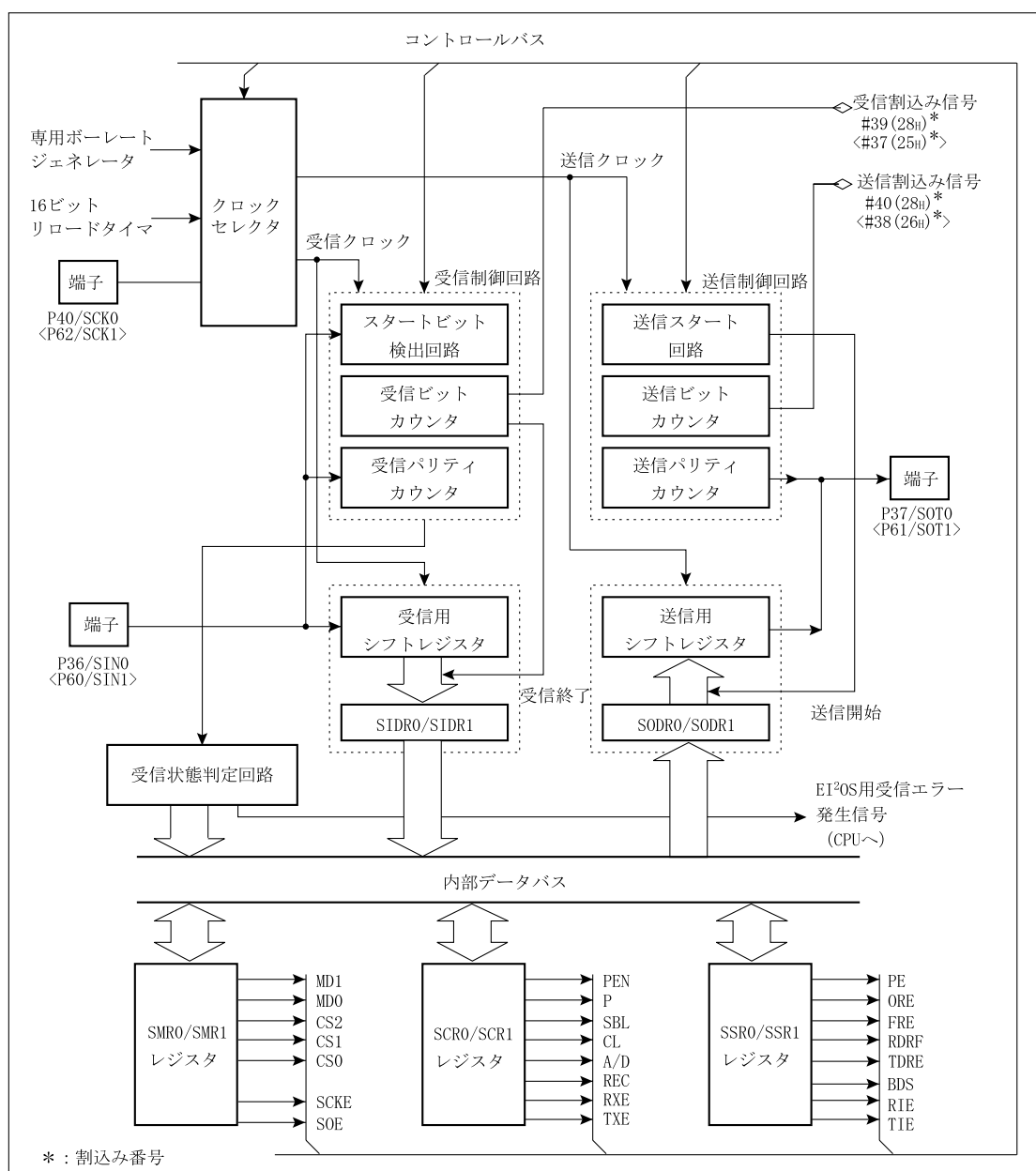


図 13.2-1 UARTのブロックダイアグラム

クロックセレクタ

専用ボーレートジェネレータ，外部入力クロック（SCK0/SCK1端子入力のクロック），内部クロック（16ビットリロードタイマから供給されるクロック）から送受信クロックを設定します。

受信制御回路

受信制御回路は，受信ビットカウンタ，スタートビット検出回路，および受信パリティカウンタで構成されています。受信ビットカウンタは受信データをカウントし，設定したデータ長に応じたデータの受信を完了すると，受信割込み要求を出力します。スタートビット検出回路は，シリアル入力信号からスタートビットを検出する回路で，スタートビットを検出した場合に，設定された転送速度に応じてシフトしながらインプットデータレジスタ（SIDR0/SIDR1）に受信データを格納します。受信パリティカウンタは，パリティありのデータを受信する場合に受信データのパリティを計算します。

送信制御回路

送信制御回路は，送信ビットカウンタ，送信スタート回路，および送信パリティカウンタで構成されています。送信ビットカウンタは送信データをカウントし，設定したデータ長に応じたデータの送信を完了すると，送信割込み要求を出力します。送信スタート回路は，アウトプットデータレジスタ（SODR0/SODR1）に送信データが格納されると，送信動作を開始します。送信パリティカウンタは，パリティありの場合のデータを送信する場合に送信するデータのパリティビットを生成します。

受信用シフトレジスタ

SIN0/SIN1端子から入力された受信データを，1ビットずつシフトしながら取り込み，受信が終了すると，インプットデータレジスタ（SIDR0/SIDR1）に受信データを転送します。

送信用シフトレジスタ

アウトプットデータレジスタ（SODR0/SODR1）にセットされた送信データを，送信用シフトレジスタに転送し，1ビットずつシフトしながらSOT0/SOT1端子に出力します。

モードレジスタ（SMR0/SMR1）

動作モードの設定，ボーレートクロックの設定，シリアルクロック入出力制御，およびシリアルデータの端子への出力許可を設定します。

制御レジスタ（SCR0/SCR1）

パリティ有無の設定，パリティの設定，ストップビット長やデータ長の設定，動作モード1でのフレームデータ形式の設定，受信エラーフラグビットのクリア，および送受信動作の許可／禁止の設定をします。

ステータスレジスタ（SSR0/SSR1）

送受信やエラーの状態の確認，シリアルデータの転送方向の設定，および送受信割込み要求の許可／禁止の設定をします。

インプットデータレジスタ（SIDR0/SIDR1）

受信したデータを格納するレジスタです。

アウトプットデータレジスタ（SODR0/SODR1）

送信するデータを設定するレジスタです。アウトプットデータレジスタにセットされたデータが，シリアル変換されて出力されます。

13.3 UARTの端子

UARTの端子および端子部のブロックダイヤグラムを示します。

UARTの端子

UARTの端子は，入出力ポートと兼用になっています。

表 13.3-1 UARTの端子

端子名	端子機能	入出力形式	プルアップ 設定	スタンバイ 制御	端子の使用に 必要な設定
P36/SIN0	入出力ポート / シリアルデータ入力	CMOS出力 / CMOSヒステリシス入力	なし	あり	入力ポートに設定 (DDR3 : bit0="0")
P37/S0T0	入出力ポート / シリアルデータ出力				シリアルデータ出力許可に設定 (SMR0 : SOE="1")
P40/SCK0	入出力ポート / シリアルクロック入出力				入力ポートに設定 (DDR4 : bit0="0")
					シリアルクロック出力許可に設定 (SMR0 : SCKE="1")
P60/SIN1	入出力ポート / シリアルデータ入力				入力ポートに設定 (DDR6 : bit0="0")
P61/S0T1	入出力ポート / シリアルデータ出力				シリアルデータ出力許可に設定 (SMR1 : SOE="1")
P62/SCK1	入出力ポート / シリアルクロック入出力				入力ポートに設定 (DDR6 : bit2="0")
					シリアルクロック出力許可に設定 (SMR1 : SCKE="1")

UARTの端子部のブロックダイヤグラム

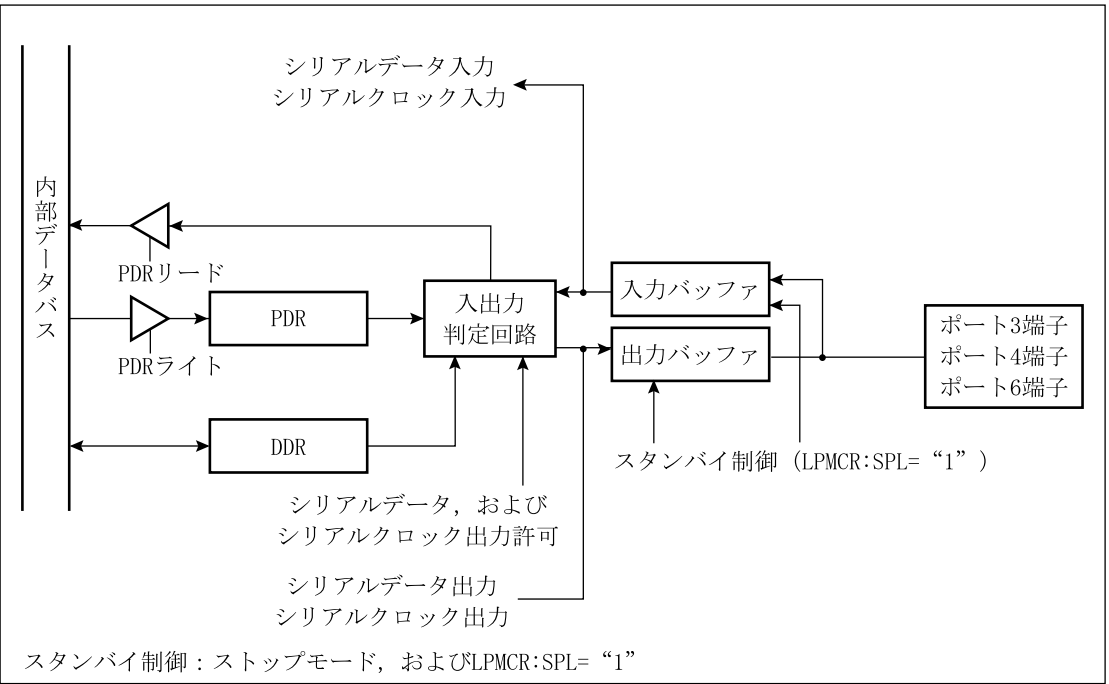


図 13.3-1 UARTの端子部のブロックダイヤグラム

13.4 UARTのレジスタ

UARTのレジスタ一覧を示します。

UARTのレジスタ一覧

アドレス	bit15.....	bit8	bit7	bit0
ch0:000021h, 20h ch1:000025h, 24h	制御レジスタ (SCR)		モードレジスタ (SMR)	
ch0:000023h, 22h ch1:000027h, 26h	ステータスレジスタ (SSR)		インプット/アウトプットデータレジスタ (SIDR/SODR)	
ch0:000029h ch1:00002Bh	通信プリスケアラ制御レジスタ (CDCR)			

図 13.4-1 UARTのレジスタ一覧

13.4.1 制御レジスタ (SCR0/SCR1)

制御レジスタ (SCR0/SCR1) は、パリティ有無の設定、パリティの設定、ストップビット長やデータ長の設定、動作モード1でのフレームデータ形式の設定、受信エラーフラグビットのクリアおよび送受信動作の許可 / 禁止を設定するレジスタです。

制御レジスタ (SCR0/SCR1)

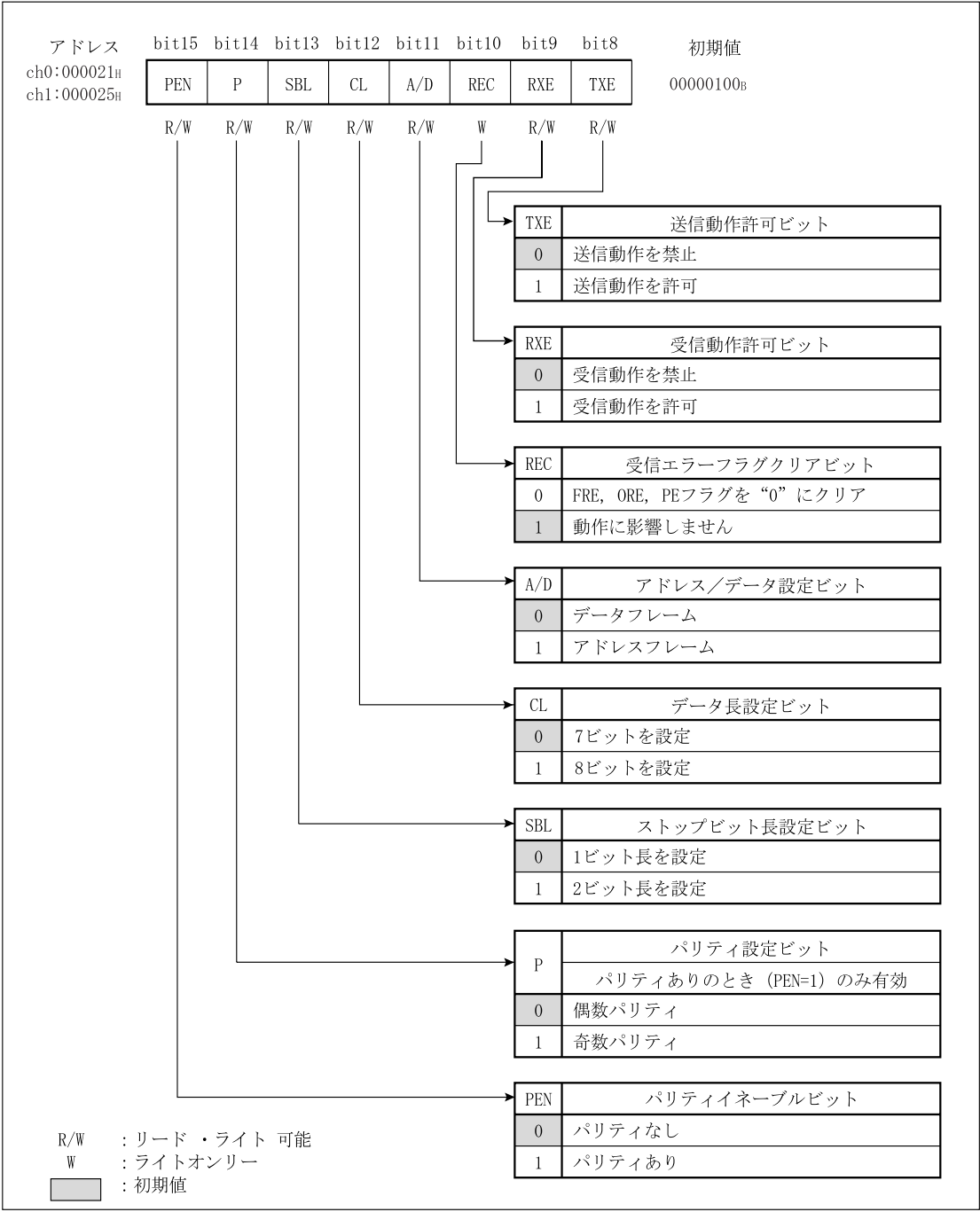


図 13.4-2 制御レジスタ (SCR0/SCR1)

表 13.4-1 制御レジスタ (SCR0/SCR1) の各ビットの機能説明

ビット名		機 能
bit15	PEN : パリティ イネーブル ビット	<ul style="list-style-type: none"> シリアルデータに対して、パリティビットの付加（送信時）および検出（受信時）をするかしないかを設定するビットです。 <p><注記> 動作モード1, 2を設定した場合は、パリティは使用できませんので、"0"を設定してください。</p>
bit14	P : パリティ設定 ビット	<ul style="list-style-type: none"> 奇数パリティ/偶数パリティを設定するビットです。 <p><注記> パリティあり (PEN="1") の場合のみ有効です。</p>
bit13	SBL : ストップビット 長設定ビット	<ul style="list-style-type: none"> 非同期転送モード時の送信データ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設定するビットです。 <p><注記> 受信時は、ストップビットの1ビット目を検出します。</p>
bit12	CL : データ長設定 ビット	<ul style="list-style-type: none"> 送受信データのデータ長を設定するビットです。 <p><注記> 7ビットを設定できるのは、動作モード0（非同期）の場合です。動作モード1（マルチプロセッサモード）、動作モード2（同期）の場合では、必ず8ビット (CL="1") を設定してください。</p>
bit11	A/D : アドレス /データ設定 ビット	<ul style="list-style-type: none"> マルチプロセッサモード（動作モード1）で、送受信するフレームのデータ形式を設定するビットです。 "0"を設定した場合は、通常データとなります。 "1"を設定した場合は、アドレスデータとなります。
bit10	REC : 受信エラー フラグクリア ビット	<ul style="list-style-type: none"> ステータスレジスタ (SSR0/SSR1) の受信エラーフラグビット (FRE, ORE, PE) を"0"にクリアするビットです。 "0"を設定した場合は、受信エラーフラグビット (FRE, ORE, PE) が"0"にクリアされます。 "1"を設定した場合は、動作に影響しません。 <p><注記> UART動作中の受信割込み許可状態で、"0"に設定する場合は、受信エラーフラグビット (FRE, ORE, PE) のいずれかに"1"が設定されている状態で行ってください。</p>
bit9	RXE : 受信動作許可 ビット	<ul style="list-style-type: none"> UARTの受信動作を制御するビットです。 "0"を設定している場合は、受信動作が禁止となります。 "1"を設定している場合は、受信動作が許可となります。 <p><注記> 受信中に受信動作を禁止した場合は、現在受信中のデータの受信を完了し、インプットデータレジスタ (SIDR0/SIDR1) に受信データを格納した時点で受信動作を停止します。</p>
bit8	TXE : 送信動作許可 ビット	<ul style="list-style-type: none"> UARTの送信動作を制御するビットです。 "0"を設定している場合は、送信動作が禁止となります。 "1"を設定している場合は、送信動作が許可となります。 <p><注記> 送信中に送信動作を禁止した場合は、アウトプットデータレジスタ (SODR0/SODR1) にデータがなくなった後で送信動作を停止します。 "0"を設定する場合は、アウトプットデータレジスタ (SODR0/SODR1) にデータを書込んだ後に、クロック非同期転送モードの場合であれば、ボーレートの1/16時間、クロック同期転送モードの場合であれば、ボーレートと同じ時間以上待ってから設定してください。</p>

13.4.2 モードレジスタ (SMR0/SMR1)

モードレジスタ (SMR0/SMR1) は、動作モードの設定、ボーレートクロックの設定、シリアルクロック入出力制御、およびシリアルデータの端子への出力許可を設定するレジスタです。

モードレジスタ (SMR0/SMR1)

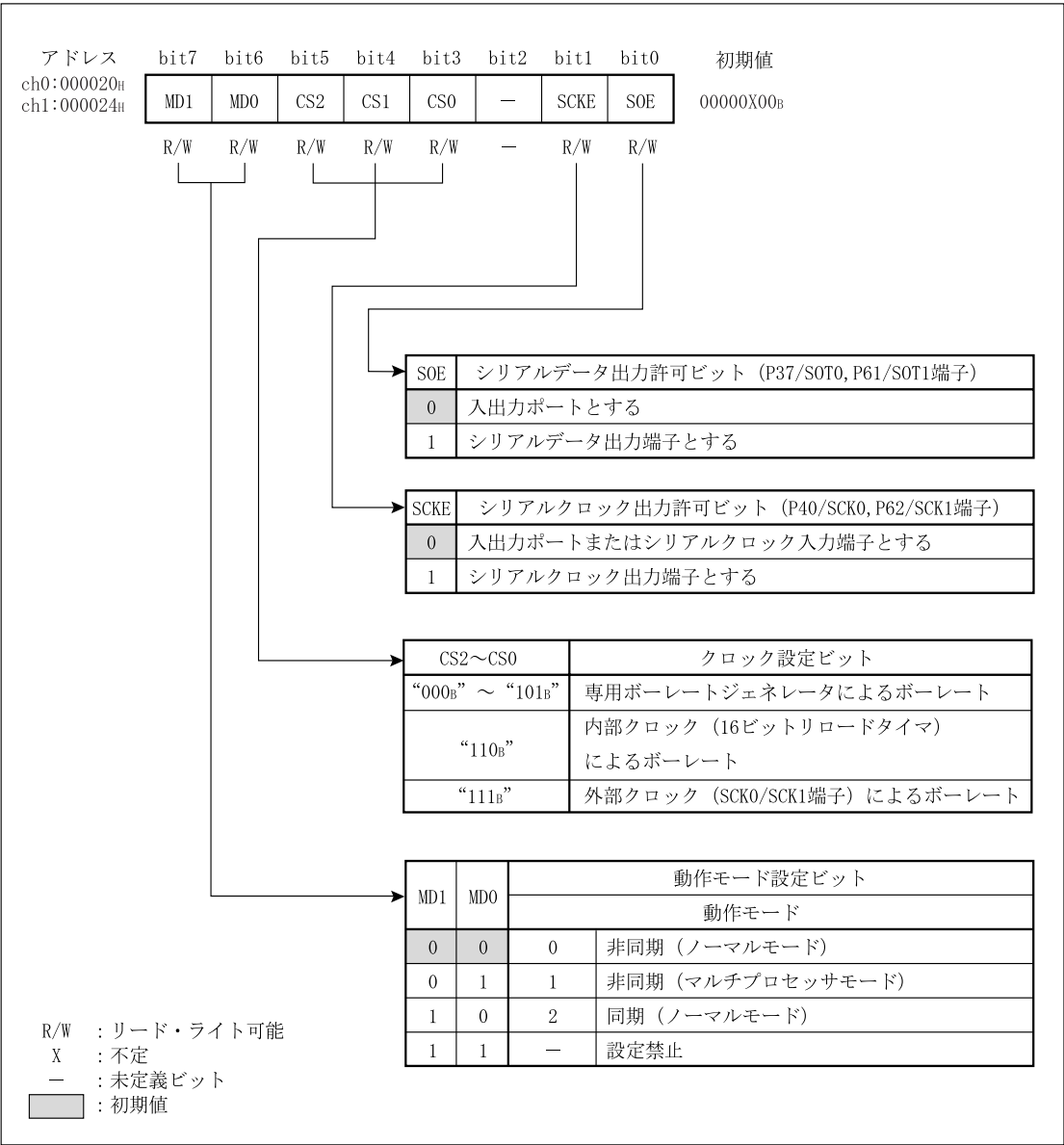


図 13.4-3 モードレジスタ (SMR0/SMR1)

表 13.4-2 モードレジスタ (SMR0/SMR1) の各ビットの機能説明

ビット名		機能
bit7 bit6	MD1, MD0 : 動作モード設定 ビット	<ul style="list-style-type: none"> 動作モードを設定するビットです。 <p><注記> 動作モード1 (マルチプロセッサモード) は、マスタ/スレーブ型通信のマスタとしてのみ使用できます。UARTは、受信時にアドレス/データ判別機能がないためスレーブとしては使用できません。</p>
bit5 bit4 bit3	CS2 ~ CS0 : クロック設定 ビット	<ul style="list-style-type: none"> ボーレートのクロックソースを設定するビットです。専用ボーレートジェネレータを設定した場合は、ボーレートも決定されます。 専用ボーレートジェネレータ6種類、内部クロックによるボーレート1種類、外部クロックによるボーレート1種類の計8種類のボーレートから設定できます。 クロック入力は、外部クロック (SCK0/SCK1端子入力)、内部クロック (16ビットリロードタイマ)、専用ボーレートジェネレータより設定できます。
bit2	- : 未定義ビット	<ul style="list-style-type: none"> 読出しを行った場合は、不定値となります。 設定した値は、動作に影響しません。
bit1	SCKE : シリアル クロック出力 許可ビット	<ul style="list-style-type: none"> シリアルクロックの入出力を制御するビットです。 "0"を設定した場合は、P40/SCK0, P62/SCK1端子は入出力ポート、またはシリアルクロック入力端子となります。 "1"を設定した場合は、シリアルクロック出力端子となります。 <p><注記> P40/SCK0, P62/SCK1端子をシリアルクロック入力 (SCKE="0") として使用する場合は、入力ポートに設定してください。クロック設定ビットによって外部クロックを設定 (SMR0/SMR1 : CS2 ~ CS0="111b") してください。</p> <ul style="list-style-type: none"> シリアルクロック出力 (SCKE="1") として使用する場合は、専用ボーレートジェネレータ (SMR0/SMR1 : CS2 ~ CS0="000b" ~ "101b")、または内部クロック (SMR0/SMR1 : CS2 ~ CS0="110b") を設定してください。 <p>[参考] シリアルクロック出力 (SCKE="1") の場合は、入出力ポートの状態にかかわらずシリアルクロック出力端子として機能します。</p>
bit0	SOE : シリアルデータ 出力許可ビット	<ul style="list-style-type: none"> シリアルデータの出力を許可するビットです。 "0"を設定した場合は、P37/SOT0, P61/SOT1端子は、入出力ポートとなります。 "1"を設定した場合は、シリアルデータ出力端子となります。 <p>[参考] シリアルデータ出力 (SOE="1") の場合は、入出力ポートの状態にかかわらずシリアルデータ出力端子として機能します。</p>

13.4.3 ステータスレジスタ (SSR0/SSR1)

ステータスレジスタ (SSR0/SSR1) は、送受信やエラーの状態の確認、シリアルデータの転送方向の設定、および割込みの許可 / 禁止を設定するレジスタです。

ステータスレジスタ (SSR0/SSR1)

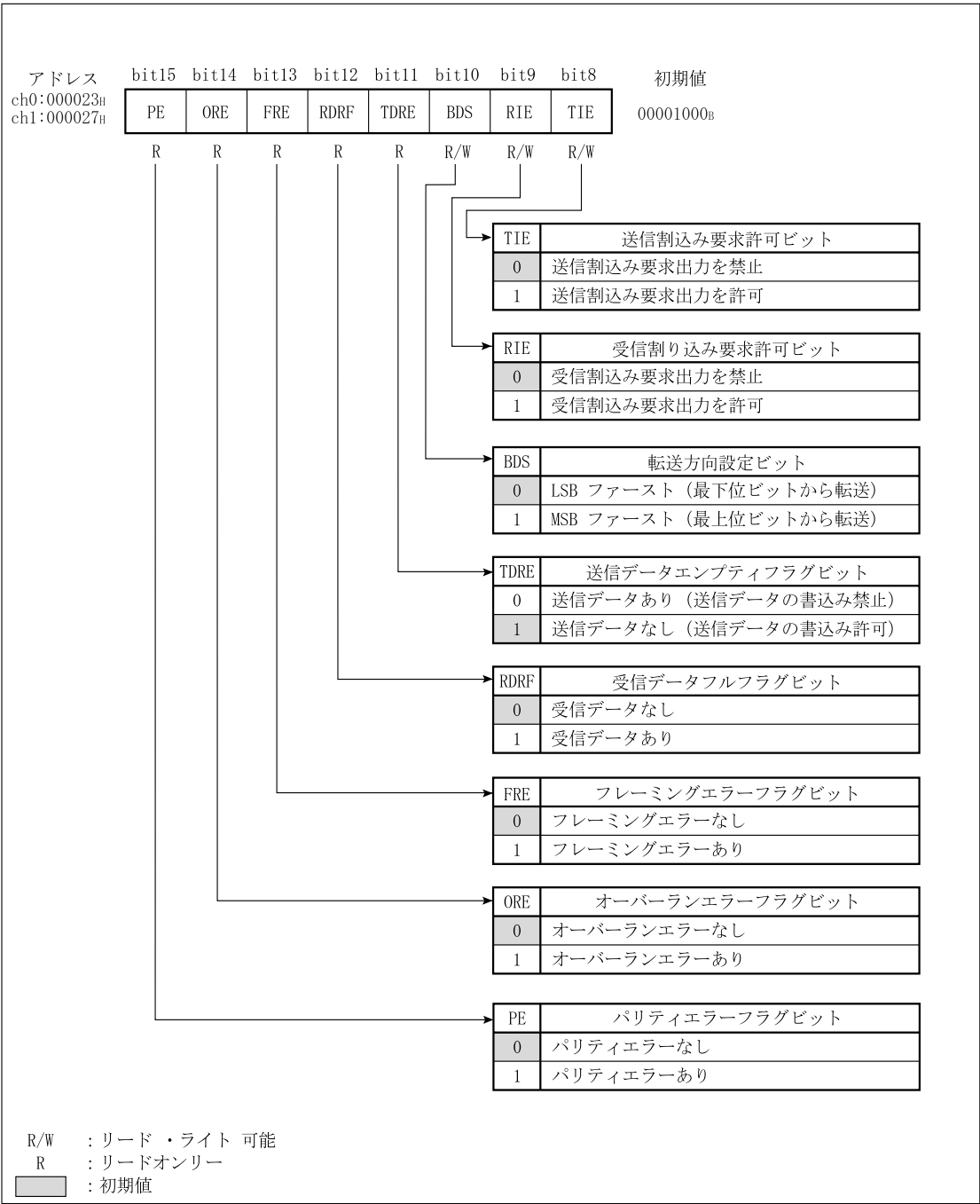


図 13.4-4 ステータスレジスタ (SSR0/SSR1)

表 13.4-3 ステータスレジスタ (SSR0/SSR1) の各ビットの機能説明

NO.	ビット名	機 能
bit15	PE : パリティエラー フラグビット	<ul style="list-style-type: none"> ・受信時にパリティエラーが発生した場合は, "1"がセットされます。 ・制御レジスタ (SCR0/SCR1) の受信エラーフラグクリアビット (REC) に "0"を設定した場合は, "0"にクリアされます。 ・"1"がセットされた場合に, 受信割込み要求許可ビット (RIE) に"1"が設定されていると, 受信割込み要求を出力します。 ・"1"がセットされた場合は, インプットデータレジスタ (SIDR0/SIDR1) のデータは無効になります。
bit14	ORE : オーバーラン エラーフラグ ビット	<ul style="list-style-type: none"> ・受信時にオーバーランエラーが発生した場合は, "1"がセットされます。 ・制御レジスタ (SCR0/SCR1) の受信エラーフラグクリアビット (REC) に "0"を設定した場合は, "0"にクリアされます。 ・"1"がセットされた場合に, 受信割込み要求許可ビット (RIE) に"1"が設定されていると, 受信割込み要求を出力します。 ・"1"がセットされた場合は, インプットデータレジスタ (SIDR0/SIDR1) のデータは無効になります。
bit13	FRE : フレーミング エラーフラグ ビット	<ul style="list-style-type: none"> ・受信時にフレーミングエラーが発生した場合は, "1"がセットされます。 ・制御レジスタ (SCR0/SCR1) の受信エラーフラグクリアビット (REC) に "0"を設定した場合は, "0"にクリアされます。 ・"1"がセットされた場合に, 受信割込み要求許可ビット (RIE) に"1"が設定されていると, 受信割込み要求を出力します。 ・"1"がセットされた場合は, インプットデータレジスタ (SIDR0/SIDR1) のデータは無効になります。
bit12	RDRF : 受信データフル フラグビット	<ul style="list-style-type: none"> ・インプットデータレジスタ (SIDR0/SIDR1) の状態を示すビットです。 ・インプットデータレジスタ (SIDR0/SIDR1) に受信データが格納された場合は, "1"がセットされます。 ・インプットデータレジスタ (SIDR0/SIDR1) を読み出した場合は, "0"にクリアされます。 ・"1"がセットされた場合は, 受信割込み要求許可ビット (RIE) に"1"が設定されると, 受信割込み要求を出力します。
bit11	TDRE : 送信データ エンプティ フラグビット	<ul style="list-style-type: none"> ・アウトプットデータレジスタ (SODR0/SODR1) の状態を示すビットです。 ・アウトプットデータレジスタ (SODR0/SODR1) に送信データを書き込んだ場合は, "0"にクリアされます。 ・データが送信用シフトレジスタに読み込まれ, 送信を開始した場合は, "1"がセットされます。 ・"1"がセットされた場合は, 送信割込み要求許可ビット (TIE) に"1"が設定されると, 送信割込み要求を出力します。 <p><注記> 初期状態では"1"がセットされています。</p>
bit10	BDS : 転送方向設定 ビット	<ul style="list-style-type: none"> ・シリアルデータの転送方向を設定します。 ・"0"を設定した場合は, 最下位ビット側から転送します (LSBファースト)。 ・"1"を設定した場合は, 最上位ビット側から転送します (MSBファースト)。 <p><注記> シリアルデータレジスタへの読み出し, 書き込み時にデータの上位側と下位側を入れ換えるため, アウトプットデータレジスタ (SODR0/SODR1) にデータを書き込んだ後に, 転送方向設定ビット (BDS) を書き換えた場合は, 書込まれたデータは無効になります。</p>
bit9	RIE : 受信割込み要求 許可ビット	<ul style="list-style-type: none"> ・受信割込み要求を許可するビットです。 ・"1"が設定されている場合に, 受信データフルフラグビット (RDRF) に"1"がセットされるか, もしくは受信エラーフラグビット (PE, ORE, FRE) のいずれかに"1"がセットされると, 受信割込み要求を出力します。
bit8	TIE : 送信割込み要求 許可ビット	<ul style="list-style-type: none"> ・送信割込み要求を許可するビットです。 ・"1"が設定されている場合に, 送信データエンプティフラグビット (TDRE) に"1"がセットされると, 送信割込み要求を出力します。

13.4.4 インプットデータレジスタ(SIDR0/SIDR1), アウトプットデータレジスタ(SODR0/SODR1)

インプットデータレジスタ (SIDR0/SIDR1) は, シリアルデータ受信用レジスタで, アウトプットデータレジスタ (SODR0/SODR1) は, シリアルデータ送信用レジスタです。

インプットデータレジスタ (SIDR0/SIDR1)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000022 _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
ch1:000026 _H	R	R	R	R	R	R	R	R	

R : リードオンリー
X : 不定

図 13.4-5 インプットデータレジスタ (SIDR0/SIDR1)

受信したデータを格納するレジスタです。SIN0/SIN1端子に送られてきたシリアルデータ信号をシフトレジスタで変換し, インプットデータレジスタ (SIDR0/SIDR1) に格納します。動作モード0でデータ長が7ビットに設定されている場合は, bit7 (D7) は無効データとなります。受信データが, インプットデータレジスタ (SIDR0/SIDR1) に格納されると, ステータスレジスタ (SSR0/SSR1) の受信データフルフラグビット (RDRF) に"1"がセットされ, 受信割込み要求出力を許可 (SSR0/SSR1:RIE="1") に設定されていると受信割込みを出力します。

インプットデータレジスタ (SIDR0/SIDR1) は, ステータスレジスタ (SSR0/SSR1) の受信データフルフラグビット (RDRF) に"1"がセットされている場合に読み出してください。受信データフルフラグビット (RDRF) はインプットデータレジスタ (SIDR0/SIDR1) を読み出すと, "0"にクリアされます。受信エラーが発生 (SSR0/SSR1:PE,ORE,FREのいずれかが"1") した場合は, インプットデータレジスタ (SIDR0/SIDR1) のデータは無効になります。

アウトプットデータレジスタ (SODR0/SODR1)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch0:000022 _H ch1:000026 _H	D7	D6	D5	D4	D3	D2	D1	D0	XXXXXXXX _B
	W	W	W	W	W	W	W	W	

W : ライトオンリー
 X : 不定

図 13.4-6 アウトプットデータレジスタ (SODR0/SODR1)

送信許可状態の場合、送信するデータをアウトプットデータレジスタ (SODR0/SODR1) に書き込むと、送信データを送信用シフトレジスタに転送し、シリアルデータに変換されて、シリアルデータ出力端子 (SOT0/SOT1端子) から送出されます。動作モード0でデータ長が7ビットに設定されている場合は、bit7 (D7) は無効データとなります。

送信データがアウトプットデータレジスタに書き込まれると、ステータスレジスタ (SSR0/SSR1) の送信データエンプティフラグビット (TDRE) が"0"にクリアされ、送信用シフトレジスタへの転送が終了すると、"1"がセットされます。送信データエンプティフラグビット (TDRE) に"1"が設定されている場合は、次の送信用データを書込むことができます。送信データエンプティフラグビット (TDRE) が"1"にセットされた場合に、送信割込み要求出力が許可 (SSR0/SSR1:TIE="1") されていると、送信割込みを出力します。次の送信データの書き込みは、送信割込みを出力した場合、または送信データエンプティフラグビット (TDRE) に"1"がセットされてから行ってください。

< 注意事項 >

アウトプットデータレジスタ (SODR0/SODR1) は書き込み専用のレジスタで、インプットデータレジスタ (SIDR0/SIDR1) は読み出し専用のレジスタですが、同一アドレスに配置されているため、書き込み値と読み出し値は異なり、INC/DEC命令などリードモディファイライト (RMW) 動作をする命令は使用できません。

13.4.5 通信プリスケアラ制御レジスタ (CDCR0/CDCR1)

通信プリスケアラ制御レジスタ (CDCR0/CDCR1) は、マシンのクロックの分周を制御するレジスタです。

通信プリスケアラ制御レジスタ (CDCR0/CDCR1)

UARTの動作クロックは、マシンのクロックを分周して生成します。通信プリスケアラ制御レジスタによって、種々のマシンサイクル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。通信プリスケアラ制御レジスタの出力は、拡張I/Oシリアルインターフェースの動作クロックにも使われています。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch0:000029 _H ch1:00002B _H	MD	—	—	—	予約	DIV2	DIV1	DIV0	0XXX0000 _B
	R/W	—	—	—	R/W	R/W	R/W	R/W	

R/W : リード・ライト可能
X : 不定
— : 未定義ビット

表 13.4-4 通信プリスケアラ制御レジスタ (CDCR0/CDCR1) の各ビットの機能説明

NO.	ビット名	機能
bit15	MD : 通信プリスケアラ 動作許可ビット	・ 通信プリスケアラの動作を許可するビットです。 ・ "1"を設定した場合は、通信プリスケアラは動作します。 ・ "0"を設定した場合は、通信プリスケアラは停止します。
bit14 bit13 bit12	- : 未定義ビット	・ 読出しを行った場合は、不定値となります。 ・ 設定した値は、動作に影響しません。
bit11	予約 : 予約ビット	・ 必ず"0"を設定してください。
bit10 bit9 bit8	DIV2 ~ DIV0 : 分周比設定ビット	・ マシンのクロックの分周比を設定するビットです。 ・ 設定値については表 13.4-5を参照してください。

表 13.4-5 マシンのクロック分周比

MD	DIV2	DIV1	DIV0	Div
0	-	-	-	停止
1	0	0	0	1
1	0	0	1	2
1	0	1	0	3
1	0	1	1	4
1	1	0	0	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	8

div: マシンのクロック分周比

< 注意事項 >

分周比を変えた場合は、クロックの安定時間として2周期分の時間を待ってから通信を行ってください。

13.5 UARTの割込み

UARTには、受信割込みと送信割込みがあり、次に示す要因で割込み要求を出力します。

- ・受信データがインプットデータレジスタ（SIDR0/SIDR1）にセットされた場合、または受信エラーが発生した場合に、受信割込みを出力します。
- ・送信データがアウトプットデータレジスタ（SODR0/SODR1）から送信用シフトレジスタに転送された場合、送信割込みを出力します。

受信割込み、送信割込みは拡張インテリジェントI/Oサービス（EI²OS）にも対応しています。

UARTの割込み

表 13.5-1 UARTの割込み制御ビットと割込み要因

送受信	割込み要求 フラグビット	動作モード			割込み要因	割込み要因 許可ビット	割込み要求フラグの クリア
		0	1	2			
受信	RDRF				受信データをインプットデータレジスタ（SIDR0/SIDR1）へ読み込み	SSR0/SSR1 : RIE	受信データの読み出し
	ORE				オーバーランエラー発生		受信エラーフラグ クリアビット （SCR0/SCR1 : REC） に"0"を設定
	FRE			×	フレーミングエラー発生		
	PE		×	×	パリティエラー発生		
送信	TDRE				アウトプットデータレジスタ（SODR0/SODR1）から送信データ転送終了	SSR0/SSR1 : TIE	送信データの書き込み

：使用ビット

×：未使用ビット

受信割込み

受信モード時は、データ受信完了、オーバーランエラー発生、フレーミングエラー発生、パリティエラー発生によってステータスレジスタ（SSR0/SSR1）中の受信データフルフラグビット（RDRF）、受信エラーフラグビット（ORE, FRE, PE）のいずれかに"1"がセットされ、受信割込み要求が許可（SSR0/SSR1 : RIE="1"）されている場合に受信割込み要求を出力します。

ステータスレジスタ（SSR0/SSR1）の受信データフルフラグビット（RDRF）は、インプットデータレジスタ（SIDR0/SIDR1）を読み出すと、"0"にクリアされます。ステータスレジスタ（SSR0/SSR1）の受信エラーフラグビット（PE, ORE, FRE）は、制御レジスタ（SCR0/SCR1）の受信エラーフラグクリアビット（REC）に"0"を設定した場合は、すべて"0"にクリアされます。

送信割込み

送信データがアウトプットデータレジスタ（SODR0/SODR1）から転送用シフトレジスタに転送された場合に、ステータスレジスタ（SSR0/SSR1）の送信データエンptyフラグビット（TDRE）に"1"がセットされます。送信割込みが許可（SSR0/SSR1 : TIE="1"）されている場合に、送信割込み要求を出力します。

UARTの割り込みとEI²OS表 13.5-2 UARTの割り込みとEI²OS

割り込み要因	割り込み番号	割り込み制御レジスタ		ベクタテーブルアドレス			EI ² OS
		レジスタ名	アドレス	下位	上位	バンク	
UART1 受信割り込み	#37(25 _H)	ICR13	0000BD _H	FFFF68 _H	FFFF69 _H	FFFF6A _H	
UART1 送信割り込み	#38(26 _H)	ICR13	0000BD _H	FFFF64 _H	FFFF65 _H	FFFF66 _H	
UART0 受信割り込み	#39(27 _H)	ICR14	0000BE _H	FFFF60 _H	FFFF61 _H	FFFF62 _H	
UART0 送信割り込み	#40(28 _H)	ICR14	0000BE _H	FFFF5C _H	FFFF5D _H	FFFF5E _H	

：UARTの受信エラー検出によるEI²OS停止機能付

：ICR13, ICR14を共有する割り込み要因を使用しない場合に使用可能

UARTのEI²OS機能

UARTはEI²OS対応の回路を有していますので受信 / 送信割り込みで別々にEI²OSを起動することができます。

受信時

他のリソースの状態に関係なく，EI²OSを使用することができます。

送信時

割り込み制御レジスタ（ICR13, ICR14）は，UARTの受信割り込みと共有していますので，UARTの受信で割り込みを使用しない場合に，EI²OSを起動できます。

13.5.1 受信割込み出力とフラグセットのタイミング

受信時の割込み要因として、受信完了（SSR0/SSR1：RDRF="1"）、および受信エラーの発生（SSR0/SSR1：PE,ORE,FREのいずれかが"1"）があります。

受信割込み出力とフラグセットのタイミング

動作モード0, 1の場合はストップビットの検出、動作モード2の場合はデータの最終ビット(D7)の検出で、受信データがインプットデータレジスタ(SIDR0/SIDR1)に格納され、ステータスレジスタ(SSR0/SSR1)の受信データフルフラグビット(RDRF)に"1"がセットされます。受信エラーが発生していれば、受信エラーフラグビット(PE,ORE,FRE)のいずれかに"1"がセットされます。各動作モードともいずれかの受信エラーフラグビットに"1"がセットされた場合は、インプットデータレジスタ(SIDR0/SIDR1)の値は無効データになります。

動作モード0（非同期、ノーマルモード）

ストップビット検出時にステータスレジスタ(SSR0/SSR1)の受信データフルフラグビット(RDRF)に"1"がセットされ、受信エラーがあれば、受信エラーフラグビット(PE,ORE,FRE)のいずれかに"1"がセットされます。

動作モード1（非同期、マルチプロセッサモード）

ストップビット検出時にステータスレジスタ(SSR0/SSR1)の受信データフルフラグビット(RDRF)に"1"がセットされ、受信エラーがあれば、受信エラーフラグビット(ORE,FRE)のいずれかに"1"がセットされます。パリティエラーの検出はできません。

動作モード2（同期、ノーマルモード）

受信データの最終ビット(D7)検出時にステータスレジスタ(SSR0/SSR1)の受信データフルフラグビット(RDRF)に"1"がセットされ、受信エラーがあれば、受信エラーフラグビット(ORE)に"1"がセットされます。パリティエラー、およびフレーミングエラーの検出はできません。

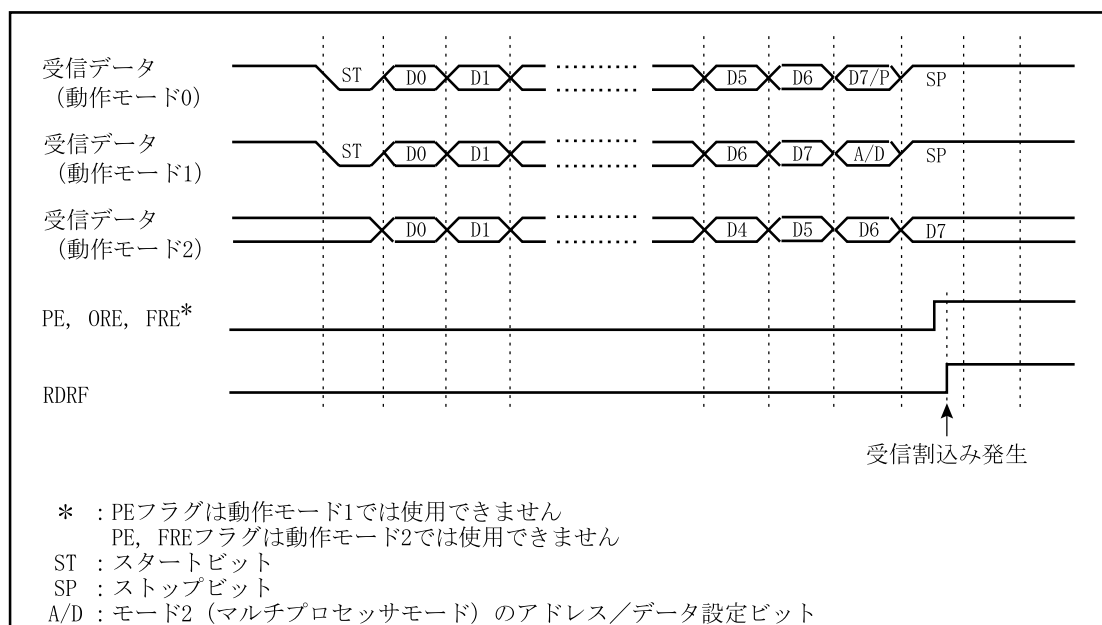


図 13.5-1 受信動作とフラグセットのタイミング

受信割込み出力のタイミング

受信割込みが許可 (SSR0/SSR1 : RIE="1") されている場合に、ステータスレジスタ (SSR0/SSR1) の受信データフルフラグビット (RDRF)、受信エラーフラグビット (PE, ORE, FRE) のいずれかに"1"がセットされると、受信割込み要求を出力します。

13.5.2 送信割込み出力とフラグセットのタイミング

送信時の割込みは、送信データがアウトプットデータレジスタ（SODR0/SODR1）から送信用シフトレジスタに転送され、次のデータの書き込みが可能になると出力します。

送信割込み出力とフラグセットのタイミング

ステータスレジスタ（SSR0/SSR1）の送信データエンプティフラグビット（TDRE）は、アウトプットデータレジスタ（SODR0/SODR1）に書き込まれたデータが、送信用シフトレジスタに転送され、次のデータ書き込みが可能になると"1"がセットされます。アウトプットデータレジスタ（SODR0/SODR1）に送信データを書き込んだ場合は、送信データエンプティフラグビット（TDRE）が"0"にクリアされます。

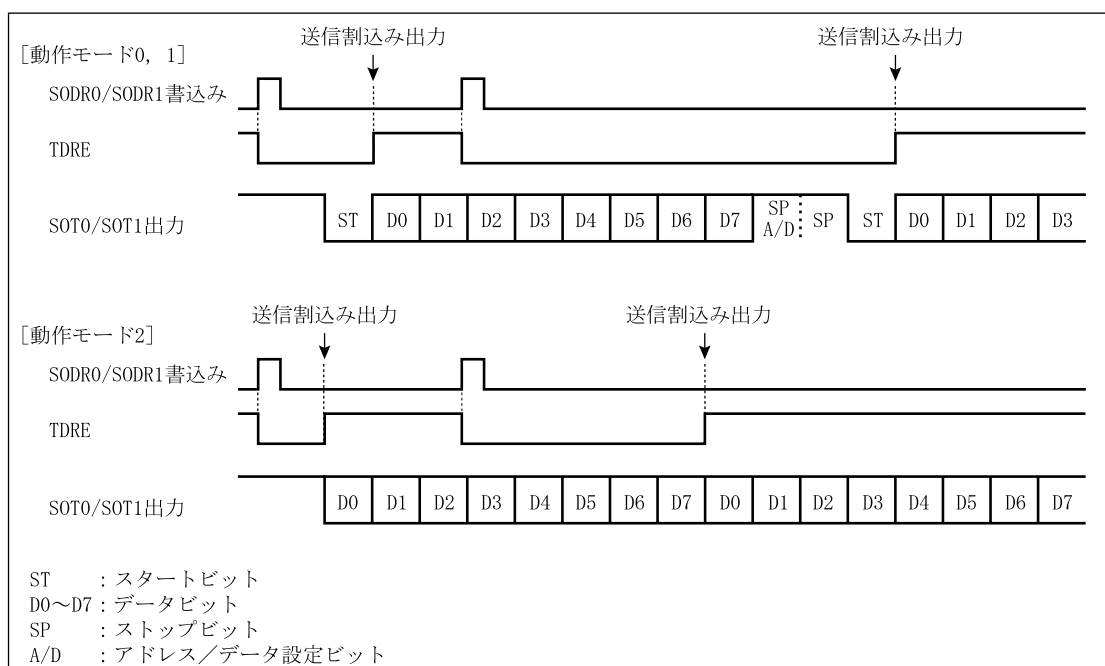


図 13.5-2 送信動作とフラグセットのタイミング

送信割込み要求出力のタイミング

送信割込みが許可（SSR0/SSR1：TIE="1"）されている場合に、ステータスレジスタ（SSR0/SSR1）の送信データエンプティフラグビット（TDRE）に"1"がセットされると、送信割込み要求を出力します。

< 注意事項 >

初期状態でステータスレジスタ（SSR0/SSR1）の送信データエンプティフラグビット（TDRE）は"1"がセットされているため、送信割込みを許可（SSR0/SSR1：TIE="1"）した場合は、送信割込みを出力します。送信データエンプティフラグビット（TDRE）は読み専用ですので、アウトプットデータレジスタ（SODR0/SODR1）に新規データを書き込む以外に"0"にクリアする方法がありません。送信割込み許可のタイミングに注意してください。

13.6 UARTのボーレート

UARTの送受信クロックは、次のいずれかを設定できます。

- ・専用ボーレートジェネレータ
 - ・内部クロック（16ビットリロードタイマ）
 - ・外部クロック（SCK0/SCK1端子入力のクロック）
-

UARTボーレート設定

ボーレートの選択回路は図 13.6-1のようになっており、ボーレートは次の3種類の中から1種類を設定できます。

専用ボーレートジェネレータによるボーレートの設定

UARTは専用ボーレートジェネレータを内蔵しており、モードレジスタ（SMR0/SMR1）で、6種類のボーレートから1種類を設定できます。マシニングロックとモードレジスタ（SMR0/SMR1）のクロック設定ビット（CS2～CS0）によって、非同期もしくは、同期ボーレートを設定します。

内部クロックによるボーレートの設定

16ビットリロードタイマから供給される内部クロックを、同期の場合はそのまま、非同期の場合は16分周して、ボーレートとして使用します。リロードタイマ値の設定によってボーレートを設定できます。

外部クロックによるボーレートの設定

UARTのクロック入力端子（P40/SCK0, P62/SCK1）から入力されたクロックを、同期の場合はそのまま、非同期の場合は16分周して、ボーレートとして使用します。外部クロックでボーレートを設定できます。

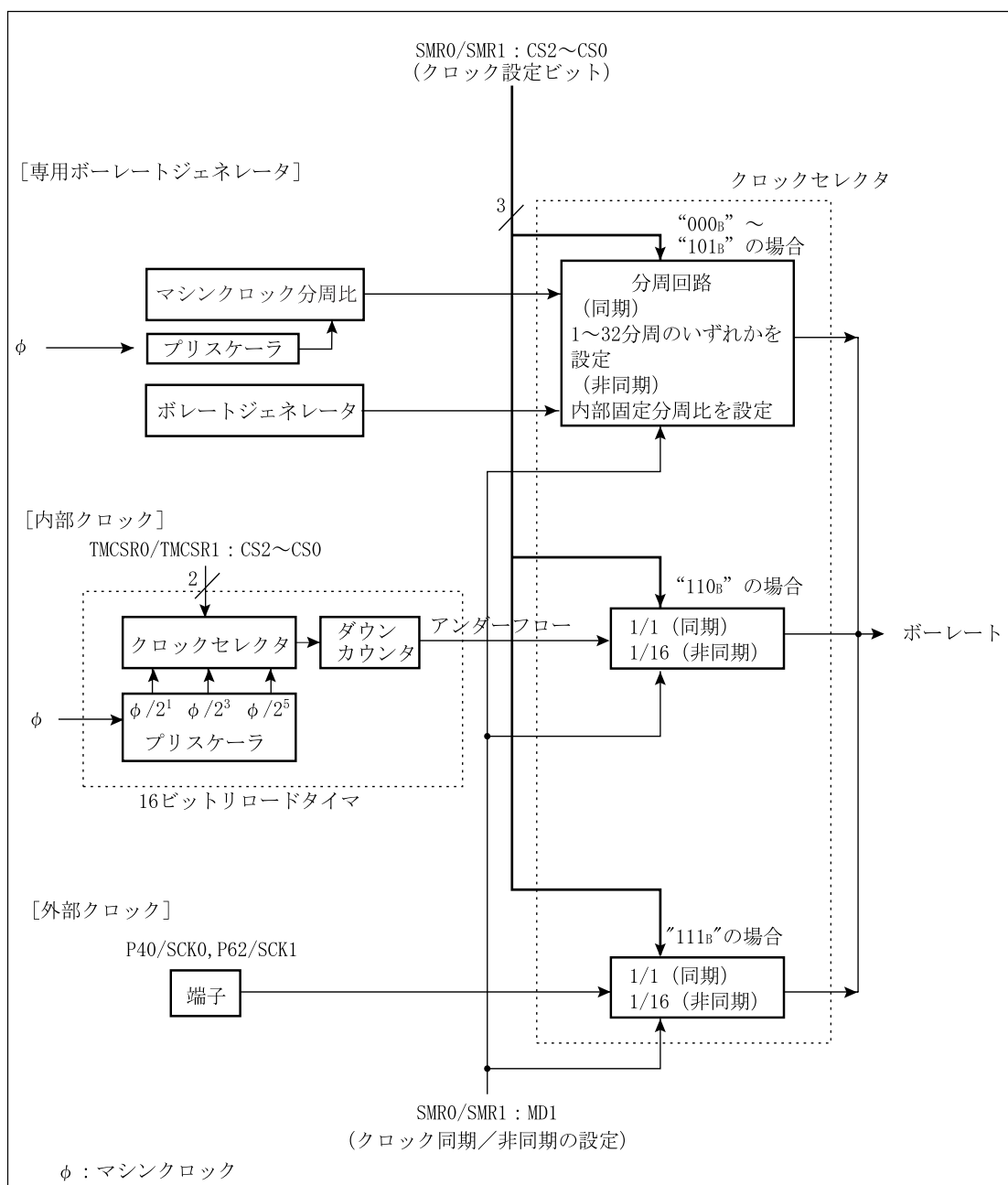


図 13.6-1 UARTボーレート選択回路

13.6.1 専用ボーレートジェネレータによるボーレート

UARTの転送クロックとして、専用ボーレートジェネレータの出力クロックを設定した場合の設定可能なボーレートを示します。

専用ボーレートジェネレータによるボーレートの設定

モードレジスタ (SMR0/SMR1) のクロック設定ビットに "000_B ~ 101_B" を設定した場合は、専用ボーレートジェネレータによるボーレートが設定されます。

専用ボーレートジェネレータで、転送クロックを生成する場合、マシニングロックは、マシニングロックプリスケアラで分周されたあと、クロックセクタで設定される転送クロック分周比で分周されます。マシニングロック分周比は、非同期 / 同期共通ですが、転送クロック分周比は、非同期と同期で、別々にモードレジスタ (SMR0/

SMR1) のクロック設定ビット (CS2 ~ CS0) で設定された値となります。

実際の転送レートは、次式で与えられます。

非同期ボーレート = $\div (\text{マシニングロック分周比}) / (\text{非同期転送クロック分周比})$

同期ボーレート = $\div (\text{マシニングロック分周比}) / (\text{同期転送クロック分周比})$

: マシニングロック周波数

プリスケアラによる分周比 (非同期 / 同期共通)

マシニングロックの分周比は、表 13.6-1 に示すように通信プリスケアラ制御レジスタ (CDCR0/CDCR1) の分周比設定ビット (DIV2 ~ DIV0) で設定します。

表 13.6-1 マシニングロックプリスケアラによる分周比の設定

MD	DIV2	DIV1	DIV0	Div
0	-	-	-	停止
1	0	0	0	1
1	0	0	1	2
1	0	1	0	3
1	0	1	1	4
1	1	0	0	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	8

div: マシニングロック分周比

同期転送クロック

同期ボーレートは、表 13.6-2に示すようにモードレジスタ (SMR0/SNR1) のクロック設定ビット (CS2 ~ CS0) で設定します。

表 13.6-2 同期ボーレートの設定

CS2	CS1	CS0	CLK同期時	算出式
0	0	0	2Mbps	$(\div \text{div})/1$
0	0	1	1Mbps	$(\div \text{div})/2$
0	1	0	500kbps	$(\div \text{div})/4$
0	1	1	250kbps	$(\div \text{div})/8$
1	0	0	125kbps	$(\div \text{div})/16$
1	0	1	62.5kbps	$(\div \text{div})/32$

ただし、

$(\text{マシクロック周波数}) = 16\text{MHz}$, $\text{div} (\text{マシクロック分周比}) = 8$

で算出しています。

非同期転送クロック

非同期ボーレートは、表 13.6-3に示すようにモードレジスタ (SMR0/SMR1) のクロック設定ビット (CS2 ~ CS0) で設定します。

表 13.6-3 非同期ボーレートの設定

CS2	CS1	CS0	非同期 (調歩同期)	算出式
0	0	0	76,923bps	$(\div \text{div})/(8 \times 13 \times 2)$
0	0	1	38,461bps	$(\div \text{div})/(8 \times 13 \times 4)$
0	1	0	19,230bps	$(\div \text{div})/(8 \times 13 \times 8)$
0	1	1	9,615bps	$(\div \text{div})/(8 \times 13 \times 16)$
1	0	0	500kbps	$(\div \text{div})/(8 \times 2 \times 2)$
1	0	1	250kbps	$(\div \text{div})/(8 \times 2 \times 4)$

ただし、

$(\text{マシクロック周波数}) = 16\text{MHz}$, $\text{div} (\text{マシクロック分周比}) = 1$

で算出しています。

13.6.2 内部クロック（16ビットリロードタイマ）によるボーレート

UARTの転送クロックとして、16ビットリロードタイマから供給される内部クロックを設定した場合の設定とボーレートの計算式を示します。

内部クロック（16ビットリロードタイマ）によるボーレートの設定

モードレジスタ（SMR0/SMR1）のクロック設定ビット（CS2～CS0）に“110_B”を設定した場合は、内部クロックによるボーレートが設定されます。ボーレートは、16ビットリロードタイマのプリスケアラ分周比とリロード値の設定により設定できます。

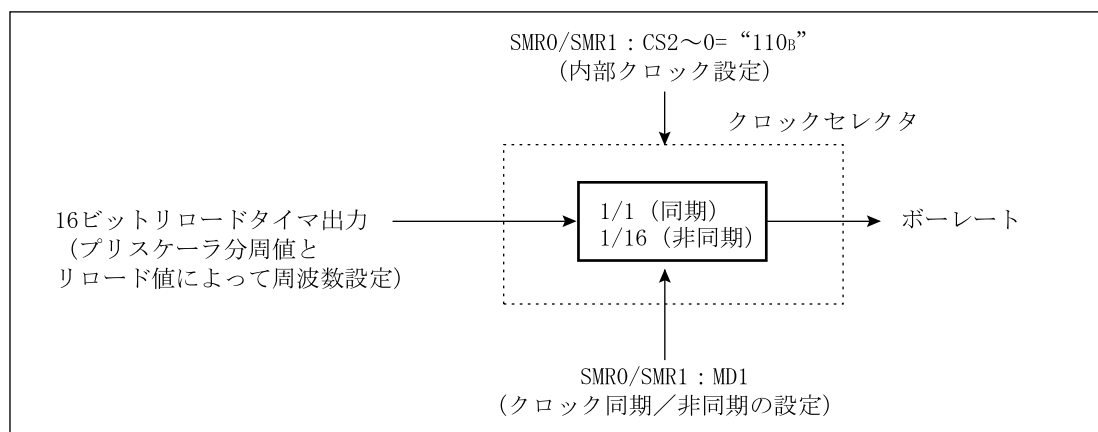


図 13.6-2 内部クロック（16ビットリロードタイマ）によるボーレート選択回路

ボーレート計算式

非同期ボーレート = $(\div N) / (16 \times 2 \times (n + 1))$ bps

同期ボーレート = $(\div N) / (2 \times (n + 1))$ bps

：マシクロック周波数

N : 16ビットリロードタイマのプリスケアラによる分周比 ($2^1, 2^3, 2^5$)

n : 16ビットリロードタイマのリロード値 (0 ~ 65,535)

リロード値の設定例（マシンクロック周波数：7.3728MHzの場合）

表 13.6-4 ボーレートとリロード値

ボーレート	リロード値 (n)			
	クロック非同期（調歩同期）		クロック同期	
	N=2 ¹ （マシンサイクル の2分周）	N=2 ³ （マシンサイクル の8分周）	N=2 ¹ （マシンサイクル の2分周）	N=2 ³ （マシンサイクル の8分周）
38400	2		47	11
19200	5		95	23
9600	11	2	191	47
4800	23	5	383	95
2400	47	11	767	191
1200	95	23	1535	383
600	191	47	3071	767
300	383	95	6143	1535

N：16ビットリロードタイマのプリスケアラによる分周比

-：設定禁止

< 注意事項 >

 動作モード2（CLK同期モード）での転送レートは，システムクロックの1/2としてください。

13.6.3 外部クロックによるボーレート

UARTの転送クロックとして、外部クロックを設定した場合の設定とボーレートの計算式を示します。

外部クロックによるボーレートの設定

外部クロックによるボーレートを設定するには、次に示す設定が必要です。

- モードレジスタ (SMR0/SMR1) のクロック設定ビット (CS2 ~ CS0) に "111_B" を設定してから、外部クロック入力によるボーレートを設定します。
- P40/SCK0, P62/SCK1端子を入力ポート (DDR4 : bit0="0", DDR6 : bit2="0") に設定します。
- モードレジスタ (SMR0/SMR1) のシリアルクロック出力許可ビット (SCKE) に "0" を設定して端子をシリアルクロック入力端子とします。

図 13.6-3に示すように、SCK0/SCK1端子から入力された外部クロックをもとに、ボーレートを設定します。内部の分周比は固定されていますので、ボーレートを変更するには外部の入力クロックの周期を変更する必要があります。

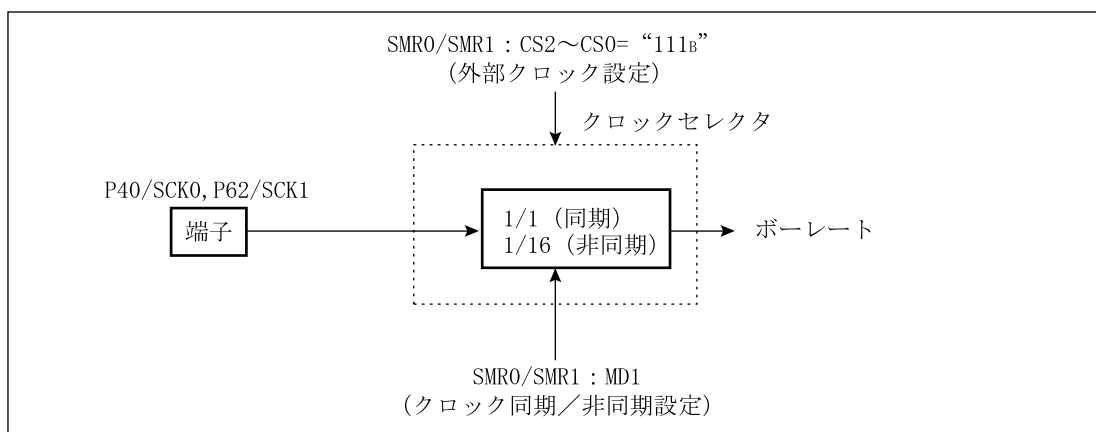


図 13.6-3 外部クロックによるボーレート選択回路

ボーレート計算式

$$\text{非同期ボーレート} = f / 16 \text{ bps}$$

$$\text{同期ボーレート} = f \text{ bps}$$

f : 外部クロック (fは最大で2MHz)

13.7 UARTの動作説明

UARTには、双方向シリアル通信機能（動作モード0, 2）、マスタ/スレーブ型通信機能（動作モード1）があります。

UARTの動作

動作モード

UARTの動作モードは動作モード0～2の3種類があり、表 13.7-1に示すようにCPU間の接続方式やデータ転送方式で設定できます。

表 13.7-1 UARTの動作モード

動作モード	データ長		同期方式	ストップビット長
	パリティなし	パリティあり		
0	7ビットまたは8ビット		非同期	1ビット
1	マルチプロセッサモード	8 + 1 ^{*1}	非同期	または 2ビット ^{*2}
2	ノーマルモード	8	同期	なし

：設定不可

*1："+1"は通信制御用に使用されるアドレス/データ設定ビット（A/D）です。

*2：受信時のストップビットは1ビットのみ検出可

< 注意事項 >

UARTの動作モード1は、マスタ/スレーブ型接続時のマスタ時に使用されます。

CPU間接続方式

1対1接続（ノーマルモード）かマスタ/スレーブ型接続（マルチプロセッサモード）を設定できます。ノーマルモード、マルチプロセッサモードともにデータ長、パリティの有無、同期方式などは、すべてのCPUで統一しておく必要があります。動作モードは次のように設定します。

- 1対1接続では、双方のCPUで動作モード0, 2のいずれか同じ方式を採用する必要があります。非同期方式では動作モード0を、同期方式では動作モード2を設定してください。
- マスタ/スレーブ型接続では、動作モード1を使用します。動作モード1を設定し、マスタとして使用してください。マスタ/スレーブ型接続ではパリティ無を設定してください。

同期方式

動作モードで、非同期方式（調歩同期）かクロック同期方式を設定できます。

信号方式

UARTは、NRZ（Non Return to Zero）形式のデータを扱えます。

動作許可

UARTは、制御レジスタ（SCR0/SCR1）に送信動作許可ビット（TXE）、受信動作許可ビット（RXE）があり、送受信動作を制御することができます。動作中に、動作禁止した場合は、次のようになります。

- 受信（受信シフトレジスタにデータが入力されている場合）に受信動作を禁止した場合は、現在受信しているデータの受信を終了し、インプットデータレジスタ（SIDR0/SIDR1）に受信データを格納して受信動作を停止します。
- 送信（送信シフトレジスタからデータが出力されている場合）に送信動作を禁止した場合は、アウトプットデータレジスタ（SODR0/SODR1）にデータがなくなった後に送信動作を停止します。
- UARTにおいて動作モード1の場合、受信データの9ビット目は無視されます。

13.7.1 非同期モード（動作モード0，1）時の動作

UARTを動作モード0（ノーマルモード），動作モード1（マルチプロセッサモード）で使用する場合，転送方式は非同期となります。

非同期モード時の動作

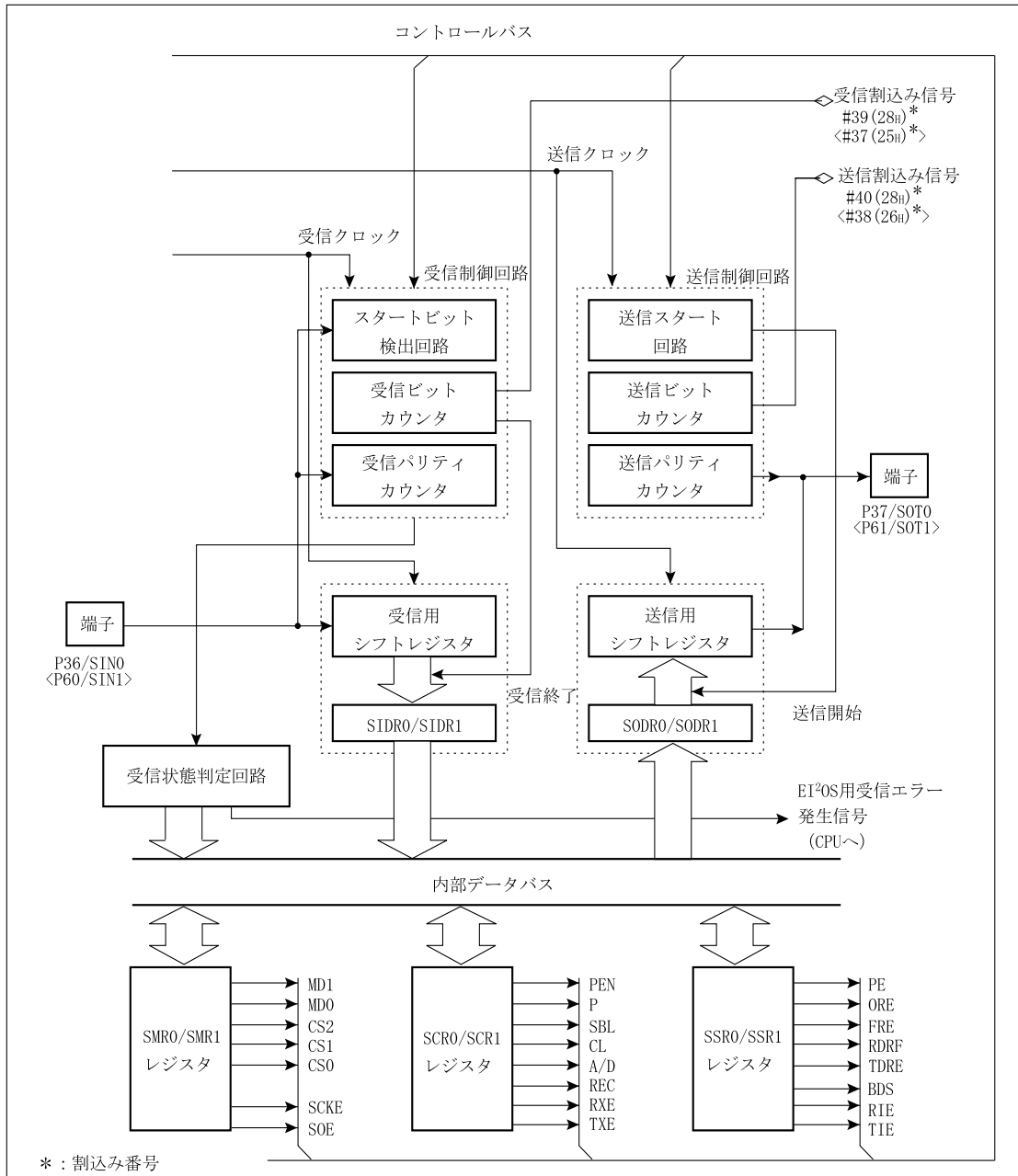


図 13.7-1 非同期モード（動作モード0，1）時のブロックダイアグラム

転送データフォーマット

転送データは必ずスタートビット ("L" レベル) から始まり, 指定されたデータビット長の転送がLSBファーストで行われ, ストップビット ("H" レベル) で終了します。

- 動作モード1では, データは8ビット長固定でパリティは付加されず, アドレス/データ設定ビット (A/D) が付加されます。

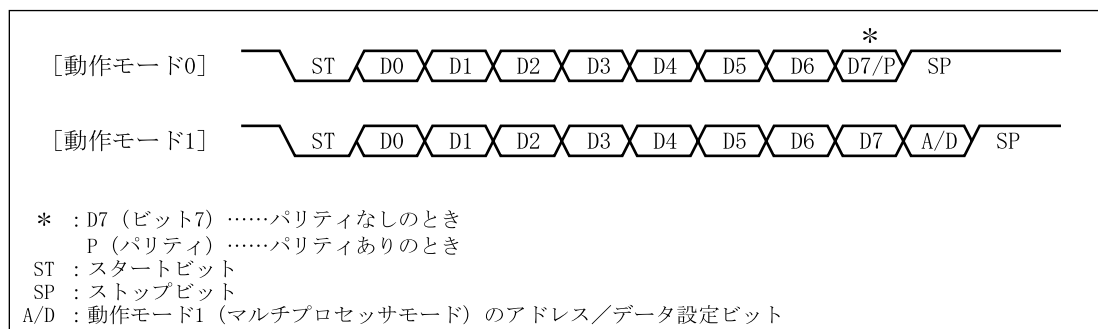


図 13.7-2 転送データフォーマット (動作モード0, 1)

送信動作

ステータスレジスタ (SSR0/SSR1) の送信データエンプティフラグビット (TDRE) に "1" が設定されている場合は, アウトプットデータレジスタ (SODR0/SODR1) に送信データを書き込みます。送信動作が許可 (SCR0/SCR1 : TXE="1") されていると, 送信が行われます。

送信データが送信用シフトレジスタに転送され, 送信が開始されるとステータスレジスタ (SSR0/SSR1) の送信データエンプティフラグビット (TDRE) に再び "1" がセットされ, 次の送信データをセットできるようになります。送信割込み要求出力が許可 (SSR0/SSR1 : TIE="1") されていると, 送信割込み要求を出力してアウトプットデータレジスタ (SODR0/SODR1) に送信データをセットするように要求します。送信データエンプティフラグビット (TDRE) は, アウトプットデータレジスタ (SODR0/SODR1) に送信データを書き込むと, "0" にクリアされます。

受信動作

受信動作が許可 (SCR0/SCR1 : RXE="1") されていると, 常に受信動作が行われます。スタートビットを検出した場合は, 制御レジスタ (SCR0/SCR1) で設定されたデータフォーマットに従って1フレームデータの受信が行われます。1フレームの受信が終わると, 受信エラーが発生した場合はステータスレジスタ (SSR0/SSR1) の受信エラーフラグビット (PE, ORE, FRE) のいずれかに "1" がセットされた後, 受信データフルフラグビット (RDRF) に "1" がセットされます。受信割込み要求出力が許可 (SSR0/SSR1 : RIE="1") されていると, 受信割込み要求を出力します。ステータスレジスタ (SSR0/SSR1) の各受信エラーフラグビット (PE, ORE, FRE) を調べ, 正常受信ならインプットデータレジスタ (SIDR0/SIDR1) を読み込み, エラーが発生していればエラー処理をするようにしてください。受信データフルフラグビット (RDRF) は, インプットデータレジスタ (SIDR0/SIDR1) から受信データを読み込むと, "0" にクリアされます。

ストップビット

送信時には1ビットまたは2ビットを設定できます。受信側では、常に最初の1ビットを判別します。

エラー検出

- 動作モード0では、パリティエラー、オーバーランエラー、フレーミングエラーが検出できます。
- 動作モード1では、オーバーランエラー、フレーミングエラーが検出でき、パリティエラーは検出できません。

パリティ

パリティは、動作モード0の場合に使用できます。制御レジスタ (SCR0/SCR1) のパリティイネーブルビット (PEN) でパリティの有無を、パリティ設定ビット (P) で偶数パリティ / 奇数パリティを設定できます。動作モード1ではパリティは使用できません。

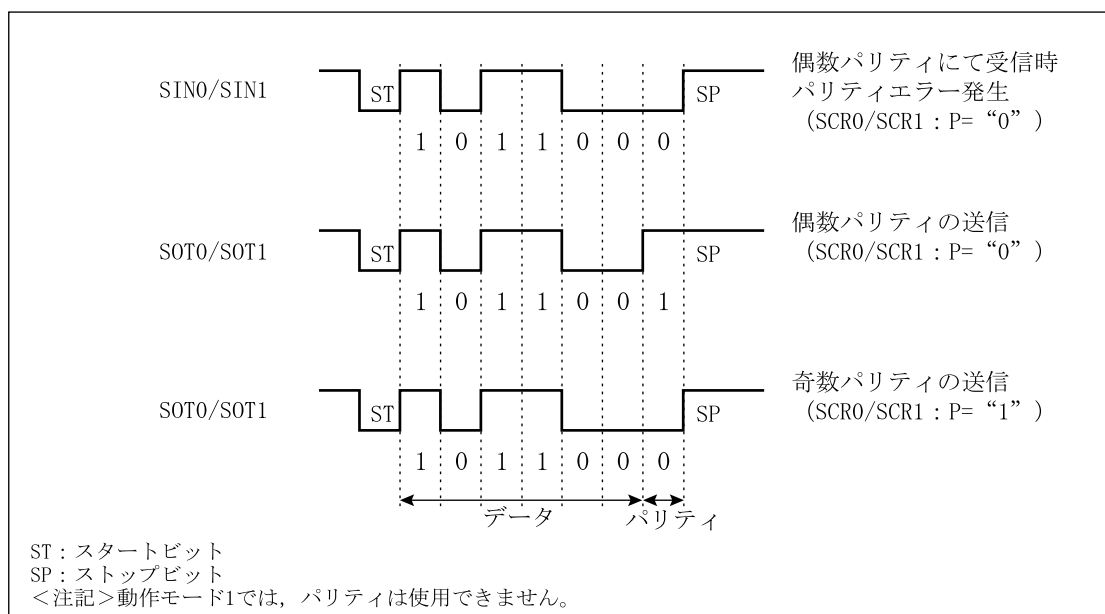


図 13.7-3 パリティ有効時の送信データ

13.7.2 同期モード（動作モード2）時の動作

UART動作モード2（ノーマルモード）で使用する場合，転送方式はクロック同期となります。

同期モード（動作モード2）時の動作

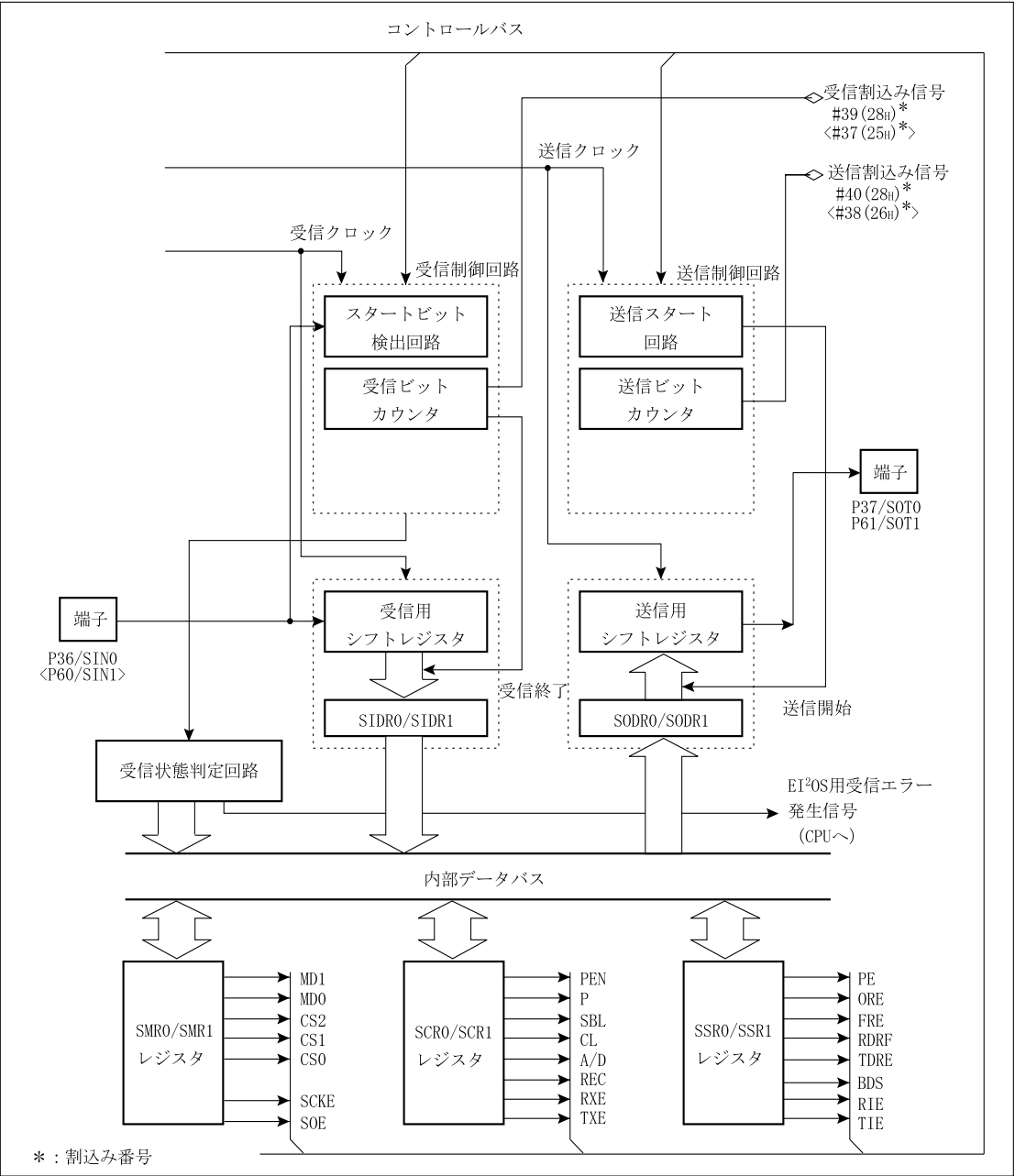


図 13.7-4 同期モード（動作モード2）時のブロックダイアグラム

転送データフォーマット

同期モードでは、8ビットデータをLSBファーストで転送します。

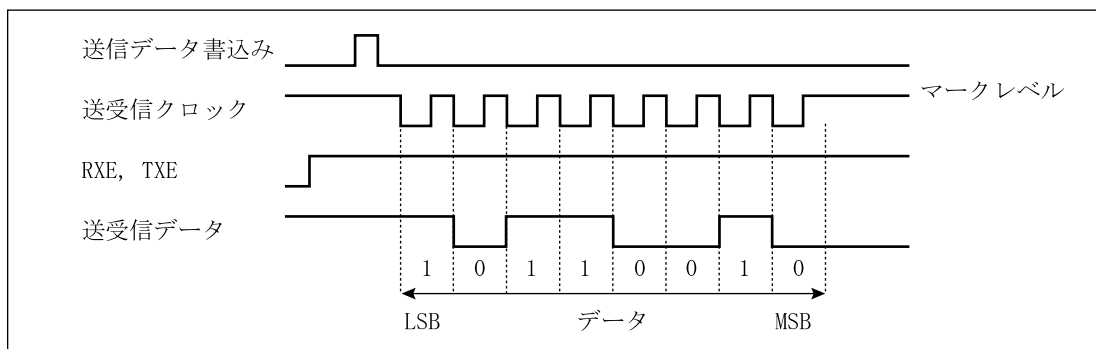


図 13.7-5 転送データフォーマット（動作モード2）

クロック供給

クロック同期（I/O拡張シリアル）方式では、送受信ビット数に等しい数のクロックの供給が必要になります。

- 内部クロックを設定している場合は、データを送信した場合にデータ受信用同期クロックが生成されます。
- 外部クロックを設定している場合は、送信側UARTのアウトプットデータレジスタ（SODR0/SODR1）にデータがあること（SSR0/SSR1：TDRE="0"）を確認した後、正確に1バイト分のクロックを外部から供給する必要があります。送信開始前と終了後は、必ずマークレベル"H"にする必要があります。

エラー検出

オーバーランエラーが検出可能で、パリティエラー、フレーミングエラーは検出できません。

初期化

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。

[モードレジスタ (SMR0/SMR1)]

MD1, MD0	: "10 _B "を設定
CS2, CS1, CS0	: クロックセレクタのクロック入力を指定
SCKE	: 専用ボーレートジェネレータまたは内部クロックの場合は"1"を設定 クロック出力、外部クロック (クロック入力) の場合は"0"を設定
SOE	: 送信する場合は, "1"を設定 受信する場合は, "0"を設定

[制御レジスタ (SCR0/SCR1)]

PEN	: "0"を設定
P, SBL, A/D	: 意味を持ちません
CL	: "1" (8ビットデータ) を設定
REC	: "0" (初期化するため, エラーフラグは全クリア) を設定
RXE, TXE	: どちらか一方に"1"を設定

[ステータスレジスタ (SSR0/SSR1)]

RIE	: 割込みを使用する場合は, "1"を設定 割込みを使用しない場合は, "0"を設定
TIE	: 割込みを使用する場合は, "1"を設定 割込みを使用しない場合は, "0"を設定

通信開始

アウトプットデータレジスタ (SODR0/SODR1) へのデータ書込みで通信を開始します。受信する場合でも通信を開始する場合は、必ず送信データをアウトプットデータレジスタ (SODR0/SODR1) に書き込む必要がありますので、注意してください。

通信終了

1フレームのデータ送受信が終了した場合は、ステータスレジスタ (SSR0/SSR1) の受信データフルフラグビット (RDRF) に"1"がセットされます。受信時は、オーバーランエラーフラグビット (ORE) をチェックし、通信が正常に行われたかどうかを判断してください。

13.7.3 双方向通信機能（ノーマルモード）

動作モード0, 2で, 1対1接続のシリアル双方向通信ができます。同期方式は, 動作モード0の場合は非同期, 動作モード2の場合は同期となります。

双方向通信機能

UARTをノーマルモード（動作モード0, 2）で動作させるためには, 図 13.7-6の設定が必要です。

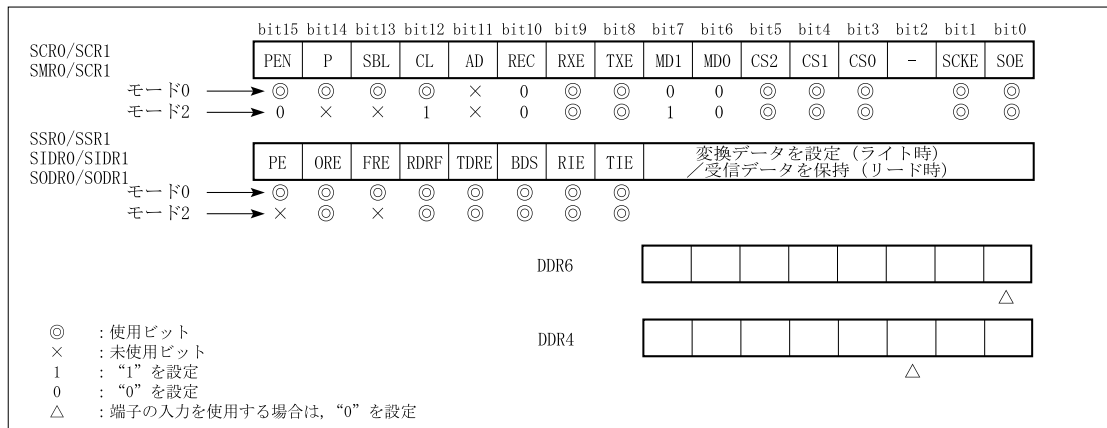


図 13.7-6 UARTの動作モード0の設定

CPU間接続

図 13.7-7に示すように, 2個のCPUを相互に接続します。

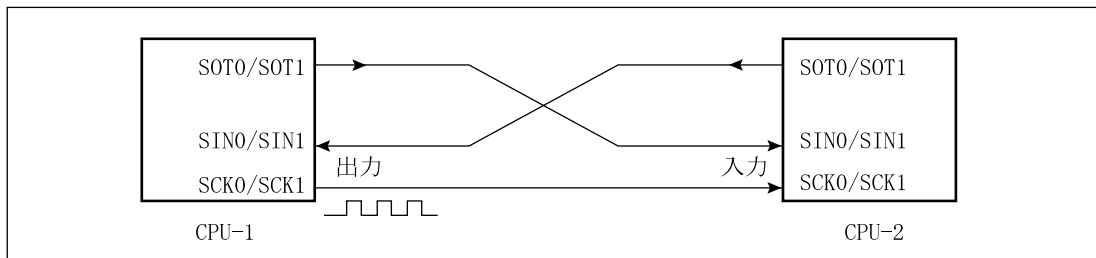


図 13.7-7 UARTの双方向通信の接続例

通信手順

通信は、送信データが準備できた場合に送信側から開始します。受信側で送信データを受けとると定期的に、ANS（本例では1バイトごと）を返します。

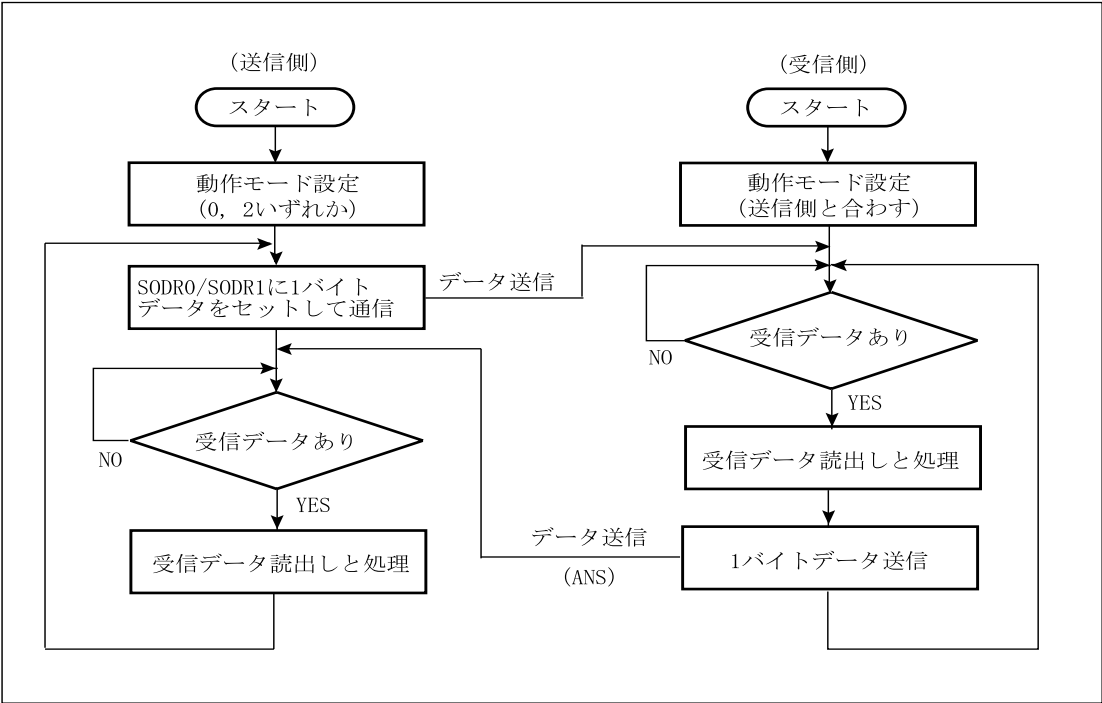


図 13.7-8 双方向通信フローの例

13.7.4 マスタ/スレーブ型通信機能 (マルチプロセッサモード)

UARTは、複数CPUのマスタ/スレーブ型接続による通信が可能であり、動作モード1を使用します。UARTは、マスタとして使用可能です。

マスタ/スレーブ型通信機能

UARTをマルチプロセッサモード(動作モード1)で動作させるためには、図 13.7-9 の設定が必要です。

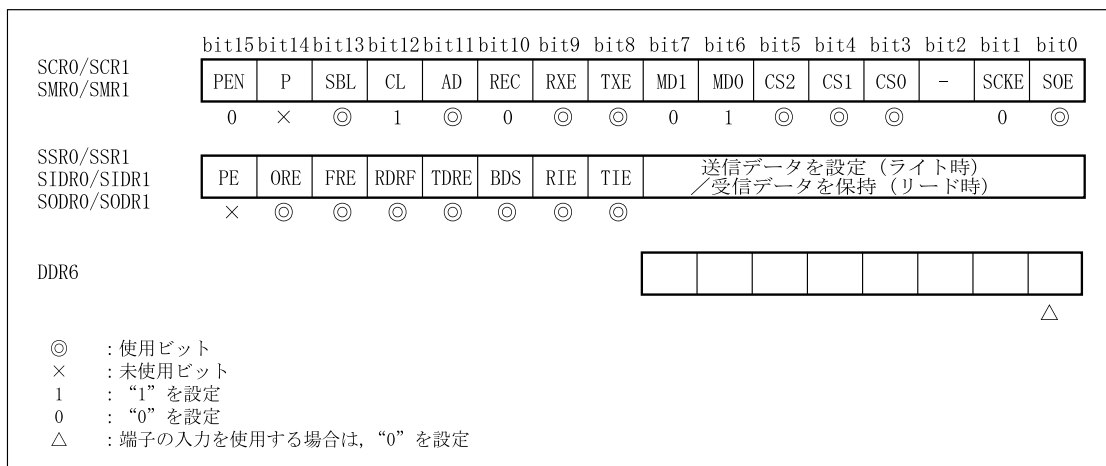


図 13.7-9 UARTの動作モード1の設定

CPU間接続

図 13.7-10に示すように、2本の共通通信ラインに1個のマスタCPUと、複数のスレーブCPUを接続して、通信システムを構成します。UARTは、マスタCPUとして使用可能です。

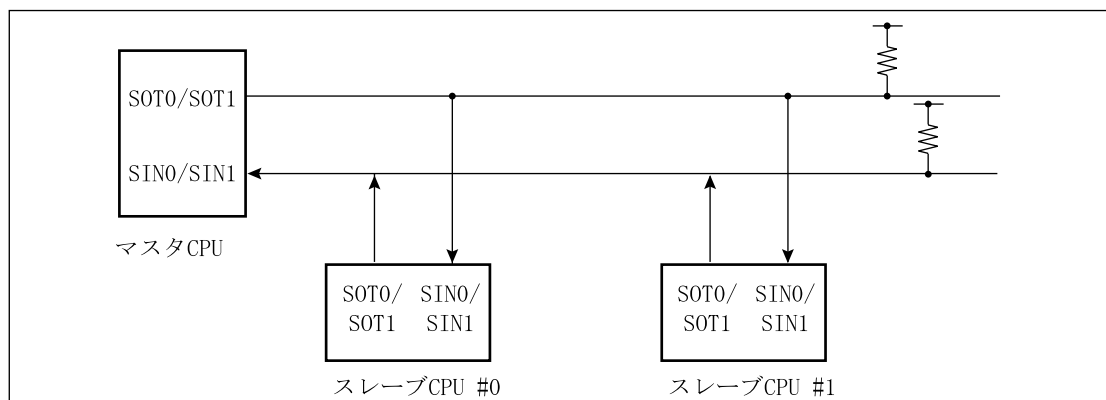


図 13.7-10 UARTのマスタ/スレーブ型通信の接続例

機能設定

マスタ/スレーブ型通信では、表 13.7-2のように、動作モードとデータ転送方式を設定してください。

表 13.7-2 マスタ/スレーブ型通信機能の設定

	動作モード		データ	パリティ	同期方式	ストップビット
	マスタCPU	スレーブCPU				
アドレス送受信	動作モード1		A/D="1" + 8ビットアドレス	無	非同期	1ビット または 2ビット
データ送受信			A/D="0" + 8ビットデータ			

通信手順

通信はマスタCPUがアドレスデータを送信することで始まります。アドレスデータとは、A/Dビットを"1"としたデータで、通信先となるスレーブCPUを設定します。各スレーブCPUはプログラムでアドレスデータを判断し、割り当てられたアドレスと一致した場合に、マスタCPUとの通信（通常データ）を開始します。

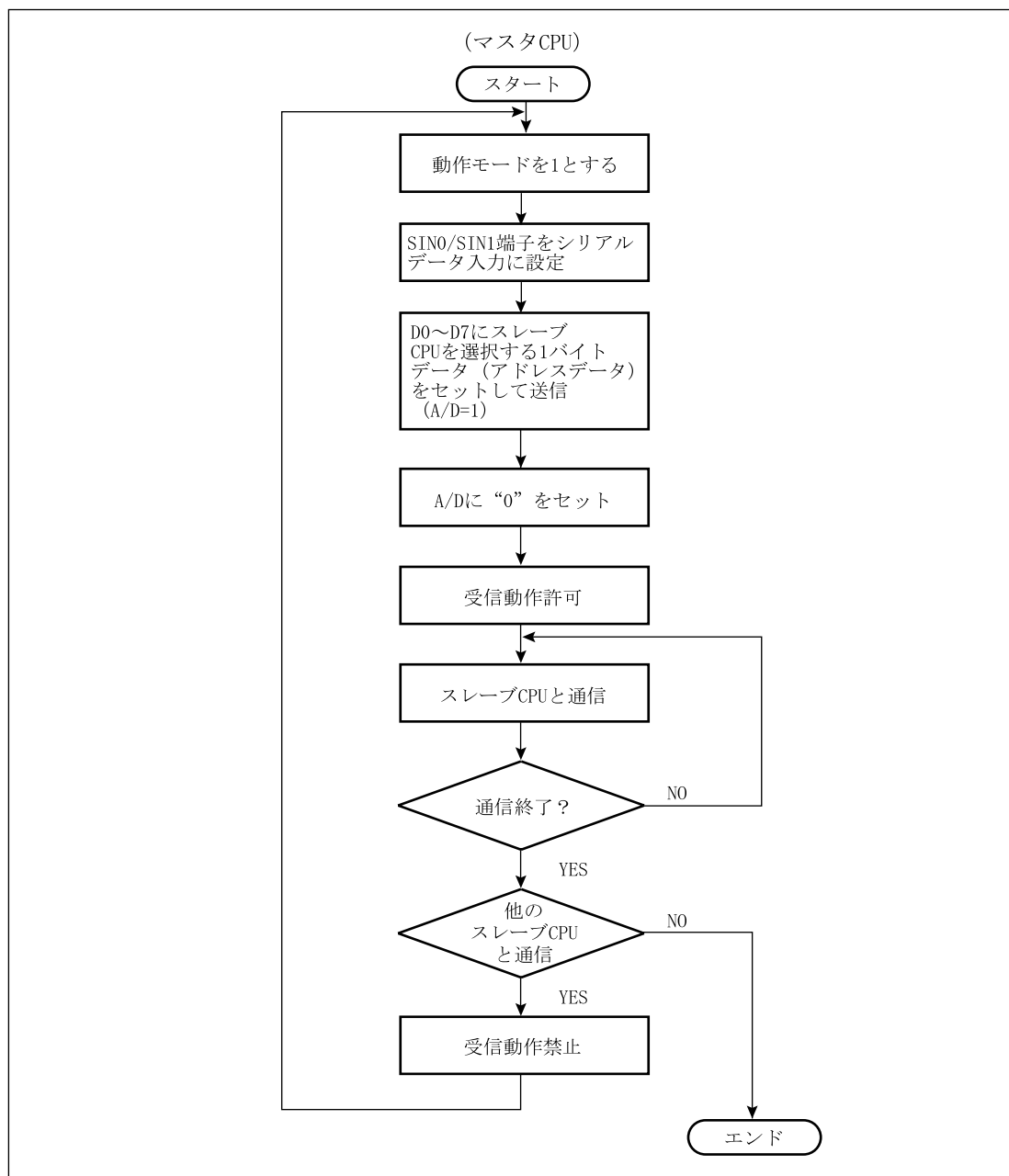


図 13.7-11 マスタ/スレーブ型通信フローチャート

13.8 UART使用上の注意

UARTを使用する場合の注意点を示します。

UART使用上の注意

動作許可

UARTには、制御レジスタ（SCR0/SCR1）に送信動作許可ビット（TXE）、受信動作許可ビット（RXE）があります。初期値では、送受信動作許可ビット（TXE,RXE）は、ともに"0"が設定されていて動作禁止であるため、転送前に"1"を設定して送受信動作を許可して下さい。必要に応じて送受信動作を禁止にして、転送を中止することもできます。

通信モードの設定

通信モードの設定は、停止中にしてください。送受信中に通信モードの設定をした場合、送受信されたデータは保証されません。

同期モード

UARTのクロック同期モード（動作モード2）は、クロック制御（I/O拡張シリアル）方式を採用しており、スタートビット、ストップビットは、データに付加されません。

送信割込み許可のタイミング

ステータスレジスタ（SSR0/SSR1）の送信データエンプティフラグビット（TDRE）は初期値が"1"（送信データなし、送信データ書込み許可）に設定されているため、送信割込み要求出力を許可（SSR0/SSR1：TIE="1"）すると、送信割込み要求を出力します。必ず、送信データを準備してから、送信割込み要求許可ビット（TIE）に"1"を設定してください。

動作モード1（マルチプロセッサモード）の受信

UARTの動作モード1（マルチプロセッサモード）においては、9ビット受信での受信動作はできません。

第14章 DTP / 外部割込み回路

この章では，MB90560/565シリーズのDTP / 外部割込み回路の機能と動作について説明します。

- 14.1 DTP / 外部割込み回路の概要
- 14.2 DTP / 外部割込み回路の構成
- 14.3 DTP / 外部割込み回路の端子
- 14.4 DTP / 外部割込み回路のレジスタ
- 14.5 DTP / 外部割込み回路の動作説明
- 14.6 DTP / 外部割込み回路使用上の注意
- 14.7 DTP / 外部割込み回路のプログラム例

14.1 DTP / 外部割込み回路の概要

DTP (Data Transfer Peripheral) / 外部割込み回路は，外部割込み入力端子 (INT7 ~ INT0) から入力された割込み要求を検出し，割込み要求を出力します。

DTP / 外部割込み回路の機能

DTP / 外部割込み回路の機能は，外部割込み入力端子 (INT7 ~ INT0) に入力されるエッジ，またはレベル信号を検出した場合に割込み要求を出力します。
 割込み要求が，CPUに受け付けられ，拡張インテリジェントI/Oサービス (EI²OS) が許可に設定されている場合は，EI²OSによる自動データ転送(DTP機能)を行ったあと，割込み処理ルーチンへ分岐します。また，EI²OSが禁止に設定されている場合は，EI²OSによる自動データ転送(DTP機能)を起動せず，割込み処理ルーチンへ分岐します。

表 14.1-1 DTP / 外部割込み回路の概要

	外部割込み機能	DTP機能
入力端子	8チャンネル (P10/INT0 ~ P16/INT6 , P63/INT7)	
割込み要因	要求レベル設定レジスタ (ELVR) にて検出レベル，またはエッジの種類を端子ごとに設定可能	
	"L"レベル / "H"レベル / 立上りエッジ / 立下りエッジの入力	
割込み番号	#25 (19 _H) ~ #28 (1C _H)	
割込み制御	DTP / 割込み許可レジスタ (ENIR) による，割込み要求出力の許可と禁止	
割込みフラグ	DTP / 割込み要因レジスタ (EIRR) による，割込み要因の保持	
処理設定	EI ² OSを禁止に設定 (ICR : ISE="0")	EI ² OSを許可に設定 (ICR : ISE="1")
処理	割込み処理ルーチンへ分岐	EI ² OSによる自動データ転送後，割込み処理ルーチンへ分岐

ICR : 割込み制御レジスタ

DTP / 外部割込み回路の割込みとEI²OS

表 14.1-2 DTP / 外部割込み回路の割込みとEI²OS

チャンネル	割込み番号	割込み制御レジスタ		ベクタテーブルアドレス			EI ² OS
		レジスタ名	アドレス	下位	上位	バンク	
INT0/INT1	#25 (19 _H)	ICR07	0000B7 _H	FFFF98 _H	FFFF99 _H	FFFF9A _H	
INT2/INT3	#26 (1A _H)			FFFF94 _H	FFFF95 _H	FFFF96 _H	
INT4/INT5	#27 (1B _H)	ICR08	0000B8 _H	FFFF90 _H	FFFF91 _H	FFFF92 _H	
INT6/INT7	#28 (1C _H)			FFFF8C _H	FFFF8D _H	FFFF8E _H	

: ICR07, ICR08を共有する割込み要因を使用しない場合に使用可能

14.2 DTP / 外部割込み回路の構成

DTP / 外部割込み回路は以下の4種類のブロックで構成されています。

- DTP / 割込み入力検出回路
- DTP / 割込み要因レジスタ (EIRR)
- DTP / 割込み許可レジスタ (ENIR)
- 要求レベル設定レジスタ (ELVR)

DTP / 外部割込み回路のブロックダイアグラム

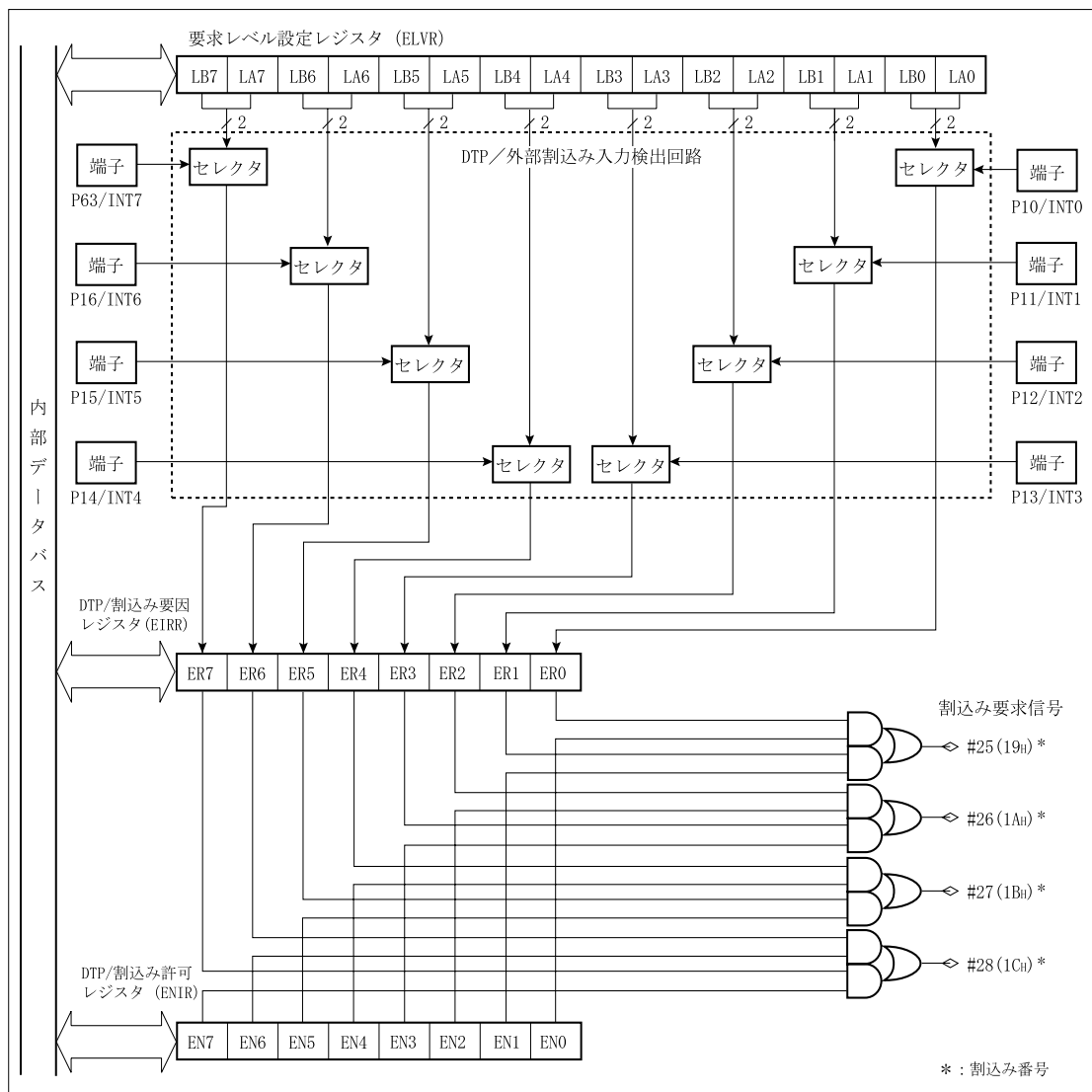


図 14.2-1 DTP / 外部割込み回路のブロックダイアグラム

DTP / 外部割込み入力検出回路

外部割込み入力端子 (INT7 ~ INT0) に入力された信号と、要求レベル設定レジスタ (ELVR) で設定したレベル、またはエッジが一致すると、外部割込み入力端子 (INT7 ~ INT0) に対応するDTP/外部割込み要因フラグビット (EIRR:ER7 ~ ER0) に"1"がセットされます。

要求レベル設定レジスタ (ELVR)

外部割込み入力端子 (INT7 ~ INT0) ごとに、割込み要求の検出条件 (レベル、またはエッジ) を設定します。

DTP / 割込み要因レジスタ (EIRR)

割込み要因の保持とクリアを行います。

DTP / 割込み許可レジスタ (ENIR)

外部割込み入力端子 (INT7 ~ INT0) ごとに、割込み要求の許可 / 禁止を設定します。

14.3 DTP / 外部割込み回路の端子

DTP / 外部割込み回路の端子，および端子部のブロックダイアグラムを示します。

DTP / 外部割込み回路の端子

DTP / 外部割込み回路の端子は，入出力ポートと兼用になっています。

表 14.3-1 DTP / 外部割込み回路の端子

端子名	端子機能	入出力形式	プルアップ抵抗	スタンバイ制御	端子の使用に必要な設定
P10/INT0	ポート1入出力 / 外部割込み 入力	CMOS出力 / CMOSヒステリ シス入力	なし	なし	入力ポートに設定 (DDR1 : bit8="0")
P11/INT1					入力ポートに設定 (DDR1 : bit9="0")
P12/INT2					入力ポートに設定 (DDR1 : bit10="0")
P13/INT3					入力ポートに設定 (DDR1 : bit11="0")
P14/INT4					入力ポートに設定 (DDR1 : bit12="0")
P15/INT5					入力ポートに設定 (DDR1 : bit13="0")
P16/INT6					入力ポートに設定 (DDR1 : bit14="0")
P63/INT7	ポート6入出力 / 外部割込み入力				入力ポートに設定 (DDR6 : bit3="0")

DTP / 外部割込み回路の端子部のブロックダイアグラム

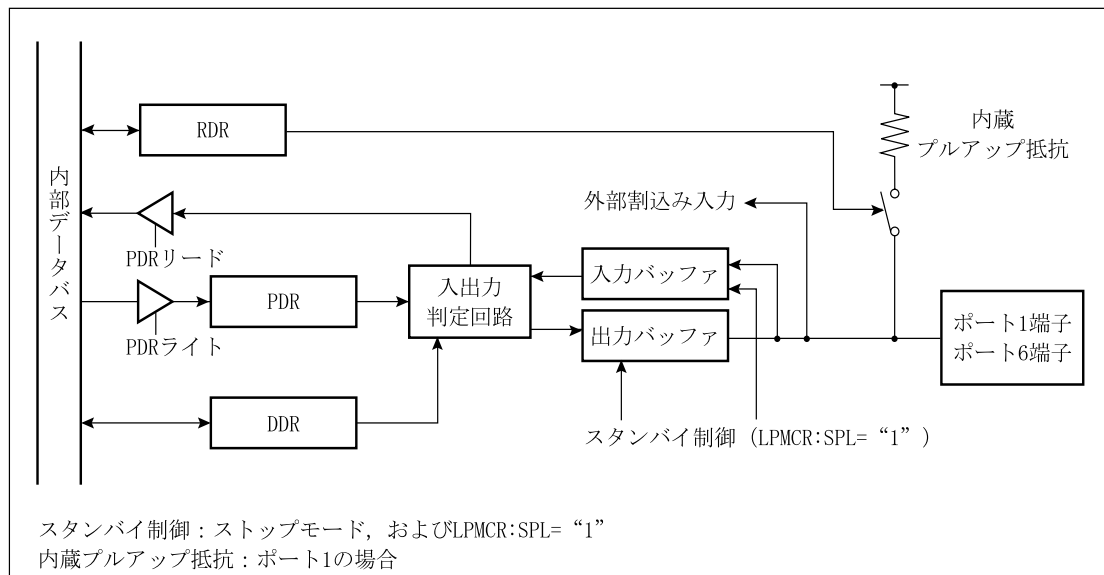


図 14.3-1 DTP / 外部割込み回路の端子部のブロックダイアグラム

14.4 DTP / 外部割込み回路のレジスタ

DTP / 外部割込み回路のレジスタ一覧を示します。

DTP / 外部割込み回路のレジスタ一覧

アドレス	bit15 bit8	bit7 bit0
000031 _H , 000030 _H	DTP／割込み要因レジスタ (EIRR)	DTP／割込み許可レジスタ (ENIR)
000033 _H , 000032 _H	要求レベル設定レジスタ (ELVR)	

図 14.4-1 DTP / 外部割込み回路のレジスタ一覧

14.4.1 DTP / 割込み要因レジスタ (EIRR)

DTP / 割込み要因レジスタ (EIRR) は、割込み要因の保持とクリアを行います。

DTP / 割込み要因レジスタ (EIRR)

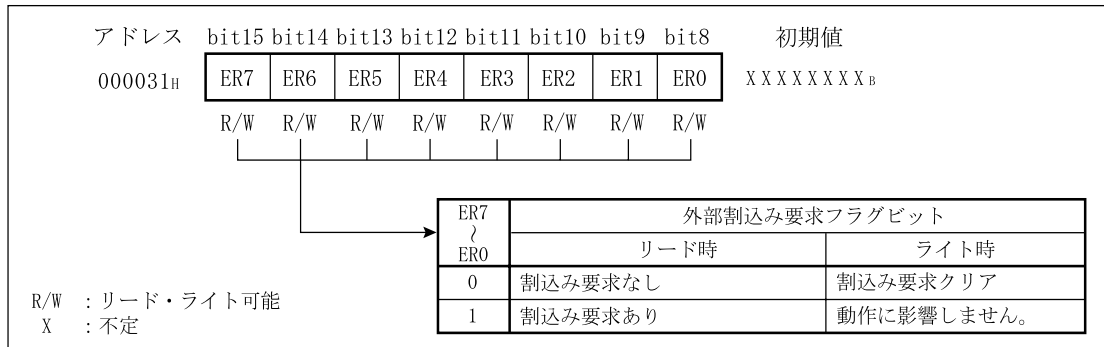


図 14.4-2 DTP / 割込み要因レジスタ (EIRR)

表 14.4-1 DTP / 割込み要因レジスタ (EIRR) の各ビットの機能説明

ビット名			機 能
bit15	ER7:	外部割込み要求 フラグビット	<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT7) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB7, LA7) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN7) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit14	ER6:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT6) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB6, LA6) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN6) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit13	ER5:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT5) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB5, LA5) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN5) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit12	ER4:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT4) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB4, LA4) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN4) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit11	ER3:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT3) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB3, LA3) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN3) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。

表 14.4-2 DTP / 割込み要因レジスタ (EIRR) の各ビットの機能説明 (続き)

ビット名			機 能
bit10	ER2:	外部割込み要求 フラグビット	<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT2) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB2, LA2) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN2) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit9	ER1:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT1) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB1, LA1) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN1) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。
bit8	ER0:		<ul style="list-style-type: none"> ・ 割込み要求のフラグビットです。 ・ 外部割込み入力端子 (INT0) に、要求レベル設定レジスタ (ELVR) の外部割込み要求検出条件設定ビット (LB0, LA0) で設定したレベル、またはエッジ信号を検出した場合に、"1"がセットされます。 ・ DTP/外部割込み許可レジスタ (ENIR) の外部割込み要求許可ビット (EN0) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・ "0"を設定した場合は、割込み要求がクリアされます。 ・ "1"を設定した場合は、動作に影響しません。

参考：

DTP機能として拡張インテリジェントI/Oサービス (EI²OS) を起動した場合は、1回のデータ転送が終了した時点で、対応する外部割込み要求フラグビット (ER7 ~ ER0) は"0"にクリアされます。

< 注意事項 >

外部割込み要求フラグビット (ER7 ~ ER0) に"0"を設定する場合は、必ずソフトウェアで"1"をリードしたビットに"0"を設定してください。"0"を設定する場合に、ハードウェアにて、外部割込み要求フラグビット (ER7 ~ ER0) に"1"がセットされた場合は、外部割込み要求フラグビット (ER7 ~ ER0) を"0"にクリアしてしまいます。

14.4.2 DTP / 割込み許可レジスタ (ENIR)

DTP / 割込み許可レジスタ (ENIR) は , 外部割込み端子 (INT7 ~ INT0) ごとに外部割込み要求の許可 / 禁止を設定します。

DTP / 割込み許可レジスタ (ENIR)

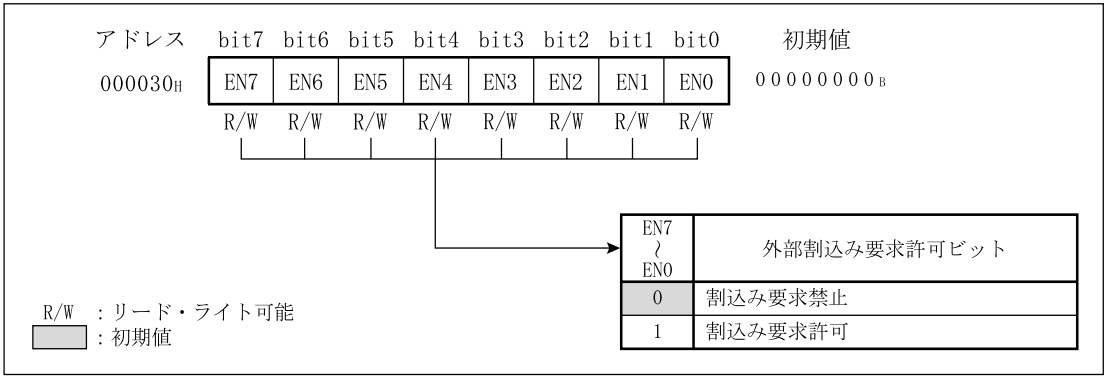


図 14.4-3 DTP / 割込み許可レジスタ (ENIR)

表 14.4-3 DTP / 割込み許可レジスタ (ENIR) の各ビットの機能説明

ビット名		機能
bit7	EN7:	外部割込み 要求許可 ビット
bit6	EN6:	
bit5	EN5:	
bit4	EN4:	
bit3	EN3:	
bit2	EN2:	
bit1	EN1:	
bit0	EN0:	

参考：

入出力ポートと兼用する，外部割込み入力端子 (INT7 ~ INT0) を使用する場合は，ポート方向レジスタ (DDR) の対応する入出力ポートと兼用するビットに "0" を設定し，端子を入力ポートとしてください。

外部割込み要求許可ビット (ENIR:EN7 ~ EN0) の状態にかかわらず，外部割込み入力端子 (INT7 ~ INT0) の状態は，ポートデータレジスタ (PDR) で直接読み出すことができます。

DTP / 割込み要因レジスタ (EIRR) の外部割込み要求フラグビット (ER7 ~ ER0) は，外部割込み要求許可ビット (ENIR:EN7 ~ EN0) の値に関係なく，DTP / 外部割込み要求の信号を検出すると，"1" がセットされます。

表 14.4-4 DTP / 割込み制御レジスタ (EIRR, ENIR) と各チャネルの対応

外部割込み入力端子	割込み番号	外部割込み要求フラグ ビット (EIRR)	外部割込み要求許可ビット (ENIR)
P63/INT7	#28 (1C _H)	ER7	EN7
P16/INT6		ER6	EN6
P15/INT5	#27 (1B _H)	ER5	EN5
P14/INT4		ER4	EN4
P13/INT3	#26 (1A _H)	ER3	EN3
P12/INT2		ER2	EN2
P11/INT1	#25 (19 _H)	ER1	EN1
P10/INT0		ER0	EN0

14.4.3 要求レベル設定レジスタ (ELVR)

要求レベル設定レジスタ (ELVR) は、外部割込み入力端子 (INT7 ~ INT0) ごとに、割込み要求の検出条件 (レベル, またはエッジ) を設定します。

要求レベル設定レジスタ (ELVR)

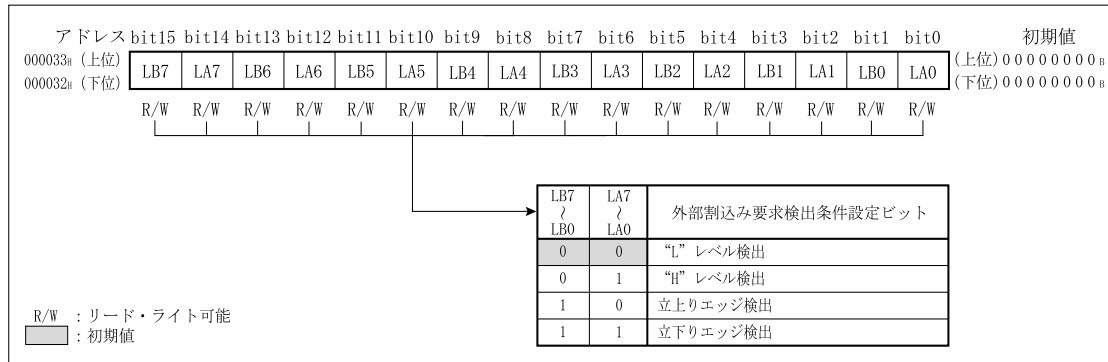


図 14.4-4 要求レベル設定レジスタ (ELVR)

表 14.4-5 要求レベル設定レジスタ (ELVR) の各ビットの機能説明

ビット名		機 能	
bit15	LB7:	外部割込み 要求検出条件 設定ビット	・ 外部割込み入力端子（INT7）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit14	LA7:		
bit13	LB6:		・ 外部割込み入力端子（INT6）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit12	LA6:		
bit11	LB5:		・ 外部割込み入力端子（INT5）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit10	LA5:		
bit9	LB4:		・ 外部割込み入力端子（INT4）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit8	LA4:		
bit7	LB3:		・ 外部割込み入力端子（INT3）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit6	LA3:		
bit5	LB2:		・ 外部割込み入力端子（INT2）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit4	LA2:		
bit3	LB1:		・ 外部割込み入力端子（INT1）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit2	LA1:		
bit1	LB0:		・ 外部割込み入力端子（INT0）に入力される信号から，割込み要求の検出条件（レベル，またはエッジ）を設定するビットです。
bit0	LA0:		

参考:

外部割込み入力端子 (INT7 ~ INT0) に、要求レベル設定レジスタ (ELVR) で設定した検出信号が入力されると、DTP / 割込み許可レジスタ (ENIR) の設定に関係なく、対応する端子の外部割込み要求フラグビット (EIRR:ER7 ~ ER0) に"1"がセットされます。

表 14.4-6 要求レベル設定レジスタ (ELVR) と各チャネルの対応

外部割込み入力端子	割込み番号	ビット名
P63/INT7	#28(10 _H)	LB7, LA7
P16/INT6		LB6, LA6
P15/INT5	#27(1B _H)	LB5, LA5
P14/INT4		LB4, LA4
P13/INT3	#26(1A _H)	LB3, LA3
P12/INT2		LB2, LA2
P11/INT1	#25(19 _H)	LB1, LA1
P10/INT0		LB0, LA0

14.5 DTP / 外部割込み回路の動作説明

DTP / 外部割込み回路は、外部割込み機能とDTP機能があります。各機能の設定と動作について説明します。

DTP / 外部割込み回路の設定

DTP / 外部割込み回路を動作させるには、図 14.5-1の設定が必要です。

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
ICR07 (上位) /ICR08 (下位)	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	
	—	—	—	—	0	◎	◎	◎	—	—	—	—	0	◎	◎	◎	外部割込み時
	◎	◎	◎	◎	1	◎	◎	◎	◎	◎	◎	◎	1	◎	◎	◎	DTP時
EIRR /ENIR	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	
	◎	◎	◎	◎	◎	◎	◎	◎	○	○	○	○	○	○	○	○	
ELVR	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	
	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	◎	
DDR1/DDR6		P16	P15	P14	P13	P12	P11	P10					P63				
		△	△	△	△	△	△	△					△				

◎：使用ビット
○：使用する端子に対応するビットに“1”を設定
△：使用する端子に対応するビットに“0”を設定
0：“0”を設定
1：“1”を設定
—：未使用ビット

図 14.5-1 DTP / 外部割込み回路の設定

DTP / 外部割込み回路のレジスタは、レジスタを設定する場合に、誤って割込み要求が発生することを避けるため、以下に示す手順で設定してください。

- 1) DTP / 割込み許可レジスタ (ENIR) に "00H" を設定し、割込み要求禁止にします。
- 2) 要求レベル設定レジスタ (ELVR) の外部割込み入力端子 (INT7 ~ INT0) に対応する外部割込み要求検出条件設定ビット (LB7 ~ LB0, LA7 ~ LA0) に、割込み検出条件を設定します。
- 3) DTP / 割込み要因レジスタ (EIRR) の外部割込み入力端子 (INT7 ~ INT0) に対応する外部割込み要求フラグビット (ER7 ~ ER0) に "0" を設定し、割込み要求クリアをします。
- 4) DTP / 割込み許可レジスタ (ENIR) の外部割込み入力端子 (INT7 ~ INT0) に対応する外部割込み要求許可ビット (EN7 ~ EN0) に "1" を設定し、割込み要求を許可します。

外部割込み機能とDTP機能の切替え

外部割込み機能とDTP機能の切替えは、使用する割込み要因に対応する割込み制御レジスタ (ICR) のEI²OS許可ビット (ISE) で設定します。EI²OS許可ビット (ISE) に "1" を設定した場合は、拡張インテリジェントI/Oサービス (EI²OS) が許可され、DTP機能として動作します。また、EI²OS許可ビット (ISE) に "0" を設定した場合は、EI²OSが禁止され、外部割込み機能として動作します。

DTP / 外部割込み動作

表 14.5-1 DTP / 外部割込み回路の制御ビットと割込み要因

	DTP / 外部割込み回路
外部割込み要求フラグビット	EIRR : ER7 ~ ER0
外部割込み要求許可ビット	ENIR : EN7 ~ EN0
割込み要因	INT7 ~ INT0端子への有効レベル, またはエッジの入力

DTP / 外部割込み回路は、要求レベル設定レジスタ (ELVR), DTP / 割込み要因レジスタ (EIRR), DTP / 割込み許可レジスタ (ENIR) への動作設定を行ったあと、対応する外部割込み入力端子 (INT7 ~ INT0) に、要求レベル設定レジスタ (ELVR) で設定した検出条件が入力されると、割込みコントローラに対して、割込み要求を出力します。割込み制御レジスタのEI²OS許可ビット (ICR : ISE) に"0"を設定した場合は、割込み処理を実行し、"1"を設定した場合は、拡張インテリジェント I/Oサービス処理 (DTP処理) を実行したあとに、割込み処理を実行します。

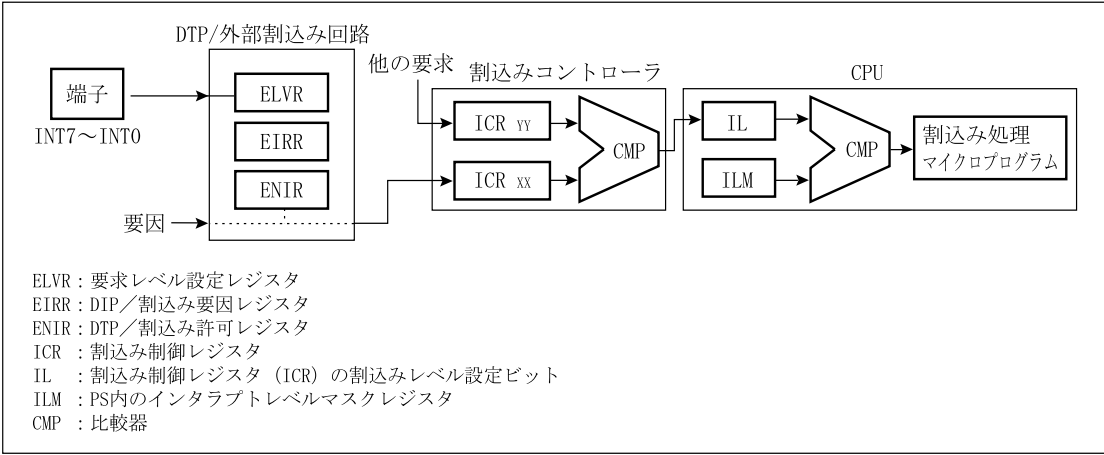


図 14.5-2 DTP / 外部割込み回路

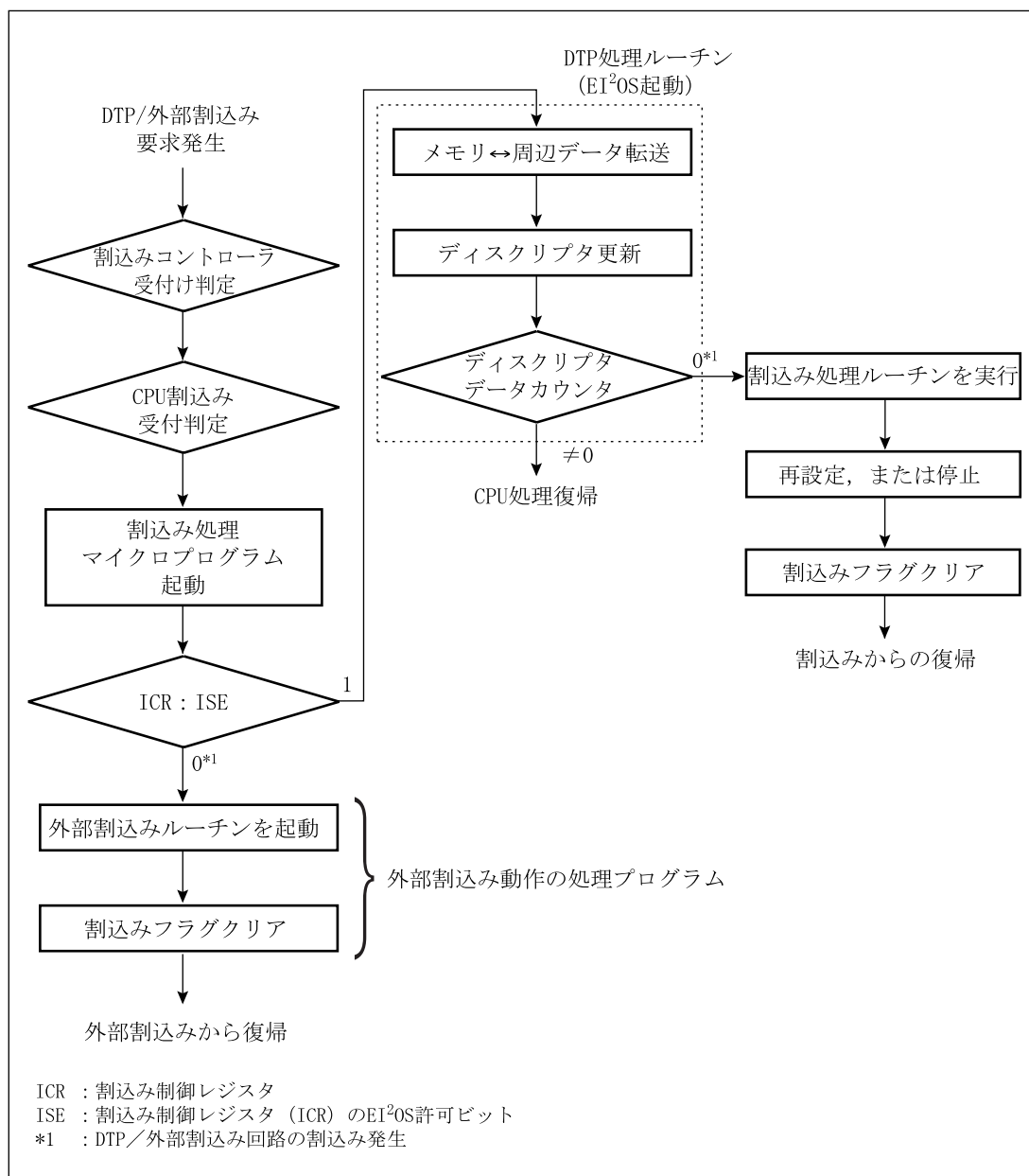


図 14.5-3 DTP / 外部割込み回路のフローチャート

14.5.1 外部割込み機能

DTP / 外部割込み回路には、外部割込み入力端子 (INT7 ~ INT0) への入力信号で、割込み要求を出力させる外部割込み機能があります。

外部割込み機能

外部割込み入力端子 (INT7 ~ INT0) に、要求レベル設定レジスタ (ELVR) で設定した検出条件 (レベル, またはエッジ) が入力されると, DTP / 割込み要因レジスタ (EIRR) の端子に対応する外部割込み要求フラグビット (ER7 ~ ER0) に"1"がセットされ, DTP / 割込み許可レジスタ (ENIR) の外部割込み入力端子 (INT7 ~ INT0) に対応する外部割込み要求許可ビット (EN7 ~ EN0) に"1"を設定していると, 割込みコントローラに対して, 割込み要求を出力します。割込みコントローラは, 周辺機能 (リソース) からの割込み要求の割込みレベル (ICR: IL2 ~ IL0), 割込みの同時出力による優先順位を判定し, CPUは, インタラプトレベルマスクレジスタ (PS: ILM), および割込み許可フラグ (PS: CCR: I) により, 割込み要求を受け付けるか判定します。CPUが割込み要求を受け付けると, 割込み処理を実行し, 割込み処理ルーチンに分岐します。割込み処理プログラムでは, 対応する外部割込み要求フラグビット (ER7 ~ ER0) に"0"を設定し, 割込み要求をクリアしたあと, 割込み復帰命令で, 割込みから復帰させてください。

< 注意事項 >

割込み処理プログラムが起動された場合は, 必ず起動要因となった外部割込み要求フラグビット (EIRR: EN7 ~ EN0) に"0"を設定してください。外部割込み要求フラグビット (EIRR: EN7 ~ EN0) に"1"がセットされた状態では, 割込みから復帰できません。

14.5.2 DTP機能

DTP / 外部割込み回路には、外部接続の周辺装置から、外部割込み入力端子 (INT7 ~ INT0) へ入力されるデータ転送要求信号を検出し、拡張インテリジェントI/Oサービスを起動するDTP(Data Transfer Peripheral)機能があります。

DTP機能の動作説明

DTP機能は、外部接続の周辺装置から、外部割込み入力端子 (INT7 ~ INT0) へ入力されるデータ転送要求信号を検出し、メモリと周辺装置の間で、データの自動転送を行う機能です。

外部割込み機能で、拡張インテリジェントI/Oサービス (EI²OS) を起動します。CPUに、割込み要求が受け付けられるまでは、外部割込み機能と同様の動作をしますが、EI²OSの動作が許可 (ICR: ISE="1") されていれば、割込み要求が受け付けられると、EI²OSを起動し、データ転送を開始します。データの転送が終了するとディスクリプタの更新などが行われ、外部割込み要求フラグビット (EIRR: ER7 ~ ER0) を"0"にクリアし、再び外部割込み機能として動作します。EI²OSによる転送がすべて終了すると、外部割込みのベクタアドレスが指し示す割込み処理ルーチンに分岐します。外部接続の周辺装置は、最初の転送が開始されてから3マシンサイクル以内にデータ転送要求信号 (DTP要因) の要因入力を取り下げてください。

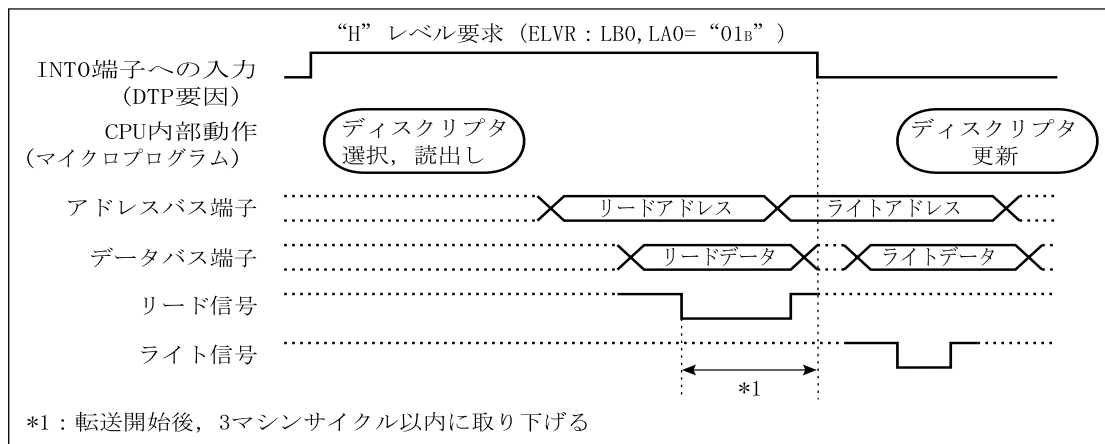


図 14.5-4 外部周辺装置とのインターフェース例 (タイミング)

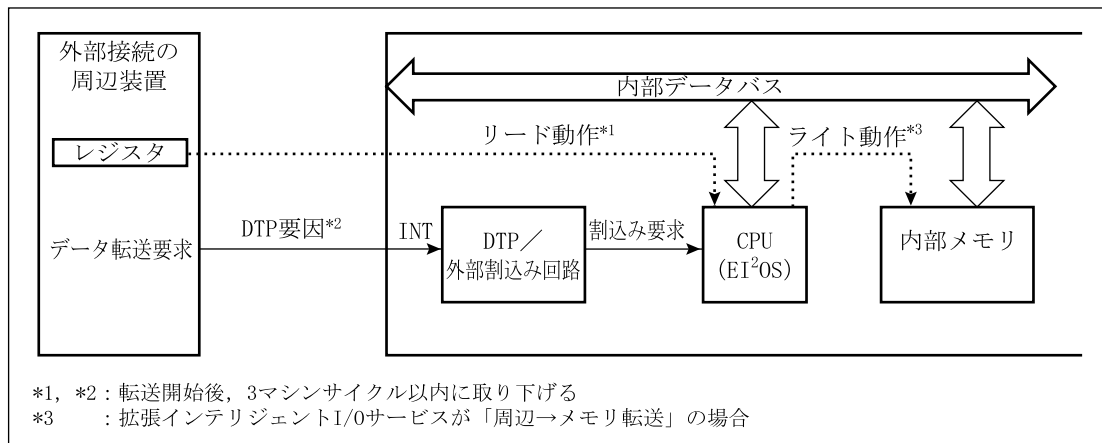


図 14.5-5 外部周辺装置とのインターフェース例 (ブロックダイアグラム)

14.6 DTP / 外部割込み回路使用上の注意

DTP / 外部割込み回路の入力信号，スタンバイモードの解除，および割込みに関連する注意を示します。

DTP / 外部割込み回路使用上の注意

DTP機能を用いた場合の外部に接続する周辺装置の条件について

DTP機能がサポートできる外部接続の周辺装置は，転送が行われたことで自動的にデータ転送要求をクリアできるものでなければなりません。外部接続の周辺装置は，CPUが転送動作を開始してから，3マシンサイクル以上転送要求を出力し続けた場合，DTP / 外部割込み回路は，次の転送要求が発生したものととして，再度データ転送動作を行います。

外部割込み入力極性について

- 要求レベル設定レジスタ（ELVR）の設定がエッジ検出の場合は，割込み要求となるエッジが入力されたことを検出するために，入力レベルの変化ポイントから，最小3マシンサイクル以上のパルス幅が必要です。
- 要求レベル設定レジスタ（ELVR）の設定がレベル検出の場合は，割込み要求となるレベルが入力されると，図 14.6-1に示すように，DTP / 割込み要因レジスタ（EIRR）内部の要因フリップフロップに"1"がセットされ，要因を保持し続けますので，割込み要因を取り下げ後も，割込みコントローラへの要求は出力されたままとなります。割込みコントローラへの要求を解除するには，図 14.6-2に示すように，外部割込み要求フラグビット（EIRR:ER7～ER0）を"0"に設定し，要因フリップフロップを"0"にクリアしてください。

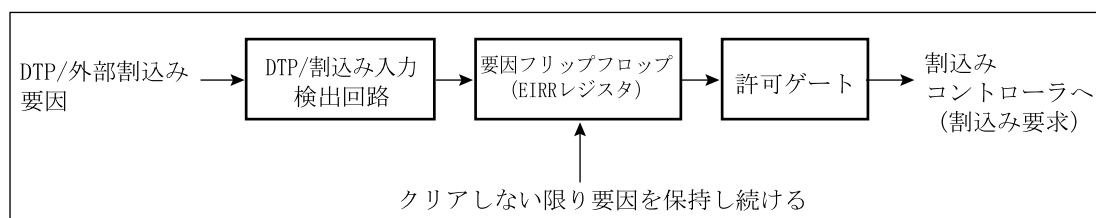


図 14.6-1 レベル設定時の要因保持回路のクリア

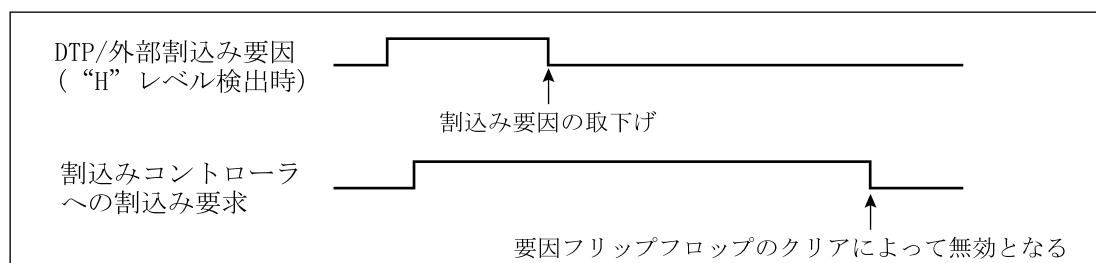


図 14.6-2 割込み要求出力許可時のDTP / 外部割込み要因と割込み要求

割込みに関する注意

外部割込み機能により、割込み処理ルーチンに分岐した場合、外部割込み要求フラグビット (EIRR:ER7 ~ ER0) が"1", 外部割込み要求許可ビット (ENIR:EN7 ~ EN0) が"1"の状態では、割込み処理プログラムから復帰できません。必ず割込み処理プログラム内で、外部割込み要求フラグビット (EIRR:ER7 ~ ER0) を"0"にクリアしてください。(DTP機能を使用した場合は、EI²OSにより、外部割込み要求フラグビット (EIRR:ER7 ~ ER0) は"0"にクリアされます。)

レベル検出設定を行っている場合は、外部割込み入力端子 (INT7 ~ INT0) に割込み要求のレベル信号が入力されていると、外部割込み要求フラグビット (EIRR:ER7 ~ ER0) に"0"を設定しても、再度、外部割込み要求フラグビット (EIRR:ER7 ~ ER0) に"1"がセットされ、割込み処理プログラムから復帰できませんので、割込み要求を禁止にするか、割込み要求のレベル信号を取り下げてください。

14.7 DTP / 外部割込み回路のプログラム例

外部割込み機能とDTP機能のプログラム例を示します。

外部割込み機能のプログラム例

処理仕様

- INT0端子に入力される，パルスの立上りエッジを検出して外部割込みを出力します。

コーディング例

```

ICR07 EQU      0000B7H          ; DTP / 外部割込み回路用割込み制御レジスタ
DDR1  EQU      000011H          ; ポート1方向レジスタ
ENIR  EQU      000030H          ; DTP / 割込み許可レジスタ
EIRR  EQU      000031H          ; DTP / 割込み要因レジスタ
ELVRL EQU      000032H          ; 要求レベル設定レジスタ
ELVRH EQU      000033H          ; 要求レベル設定レジスタ
ERO   EQU      EIRR:0           ; INT0 割込みフラグビット
ENO   EQU      ENIR:0           ; INT0 割込み許可ビット
;-----メインプログラム-----
CODE  CSEG
START:
;      ;                          ; スタックポインタ (SP) などは初期化済み
;      ;                          ; とする
MOV    I:DDR1, #00000000B       ; DDR1を入力に設定
AND    CCR, #0BFH               ; 割込みディセーブル
MOV    I:ICR07, #00H            ; 割込みレベル0 (最強), EI2O2Siはディセーブ
;      ;                          ; ル
CLRB   ENO                      ; ENIRでINT0を禁止
MOV    I:ELVR, #00000010B       ; INT0は立上りエッジ設定
CLRB   I:ERO                    ; EIRRでINT0の要因クリア
SETB   I:ENO                    ; ENIRでINT0を許可
MOV    ILM, #07H                ; PS内ILMをレベル7に設定
OR     CCR, #40H                ; 割込み許可イネーブル
LOOP:  MOV    A, #00H            ; 無限ループ
MOV    A, #01H
BRA    LOOP
;-----割込みプログラム-----
WARI:  CLRB   ERO                ; 割込み要求フラグをクリア
;      ;
;      ; ユーザ処理
;      ;
RETI   ; 割込みから復帰
CODE  ENDS
;-----ベクタ設定-----
VECT  CSEG      ABS=OFFH
ORG    OFF98H   ; 割込み#25(19H)にベクタを設定
DSL    WARI
ORG    OFFDCH   ; リセットベクタ設定
DSL    START
DB      00H     ; シングルチップモードに設定
VECT  ENDS
END      START

```

DTP機能のプログラム例

処理仕様

- INT0端子に入力される，信号の"H"レベルを検出して拡張インテリジェントI/Oサービス (EI²OS) のチャンネル0を起動します。
- DTP処理 (EI²OS) でRAM上のデータをポート0に出力します。

コーディング例

```

ICR07 EQU      0000B7H      ; DTP / 外部割込み回路用割込み制御レジスタ
DDR0 EQU      000010H      ; ポート0方向レジスタ
DDR1 EQU      000011H      ; ポート1方向レジスタ
ENIR EQU      000030H      ; DTP / 割込み許可レジスタ
EIRR EQU      000031H      ; DTP / 割込み要因レジスタ
ELVRL EQU     000032H      ; 要求レベル設定レジスタ
ELVRH EQU     000033H      ; 要求レベル設定レジスタ
ERO EQU      EIRR:0        ; INT0 割込みフラグビット
ENO EQU      ENIR:0        ; INT0 割込み許可ビット
BAPL EQU      000100H      ; バッファアドレスポインタ下位
BAPM EQU      000101H      ; バッファアドレスポインタ中位
BAPH EQU      000102H      ; バッファアドレスポインタ上位
ISCS EQU      000103H      ; EI2OSステータスレジスタ
IOAL EQU      000104H      ; I/Oアドレスレジスタ下位
IOAH EQU      000105H      ; I/Oアドレスレジスタ上位
DCTL EQU      000106H      ; データカウンタ下位
DCTH EQU      000107H      ; データカウンタ上位
;-----メインプログラム-----
CODE CSEG
START:
;      :      ; スタックポインタ (SP) などは初期化済み
;      :      ; とする
MOV     I:DDR0, #11111111B ; DDR0を出力に設定
MOV     I:DDR5, #00000000B ; DDR5を入力に設定
AND     CCR, #0BFH        ; 割込みディセーブル
MOV     I:ICR07, #08H      ; 割込みレベル0 (最強)
;      :      ; EI2OSイネーブル, チャンネル0
;      :      ; 出力データのアドレスを設定
MOV     BAPL, #00H
MOV     BAPM, #06H
MOV     BAPH, #00H
MOV     ISCS, #12H
;      :      ; バイト転送, I/Oアドレス固定,
;      :      ; バッファアドレス+1, メモリ I/Oへ転送
;      :      ; 転送先アドレスポインタとして
MOV     IOAL, #00H
MOV     IOAH, #00H
MOV     DCTL, #0AH
MOV     DCTH, #00H
;      :      ;
CLR     I:ENO              ; ENIRでINT0を禁止
MOV     I:ELVR, #00000001B ; INT0は"H"レベル設定
CLR     I:ERO              ; EIRRでINT0の要因クリア
SET     I:ENO              ; ENIRでINT0を許可
MOV     ILM, #07H          ; PS内ILMをレベル7に設定
OR      CCR, #40H          ; 割込みイネーブル
LOOP:   MOV     A, #00H     ; 無限ループ
        MOV     A, #01H
        BRA     LOOP
;-----割込みプログラム-----
WARI:   CLR     I:ERO      ; 割込み要求フラグをクリア
;      :      ; 必要に応じてチャンネル切替え, 転送
;      :      ; アドレスの変更
;      :      ; ユーザ処理
;      :      ; EI2OSの終了などの再設定を行う, 終了させ
;      :      ; る場合は割込みも禁止すること
;      :      ; 割込みから復帰
        RETI
CODE    ENDS

```

```

;-----ベクタ設定-----
VECT  CSEG      ABS=OFFH
      ORG      OFF98H          ; 割込み#25(19H)にベクタを設定
      DSL      WARI
      ORG      OFFDCH          ; リセットベクタ設定
      DSL      START
      DB       00H              ; シングルチップモードに設定
VECT  ENDS
      END      START

```

第15章 遅延割込み発生モジュール

この章では MB90560/565シリーズの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。

- 15.1 遅延割込み発生モジュールの概要
- 15.2 遅延割込み要因発生/解除レジスタ (DIRR)
- 15.3 遅延割込み発生モジュールの動作説明
- 15.4 遅延割込み発生モジュール使用上の注意

15.1 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要求を出力します。遅延割込み発生モジュールを使用すると、ソフトウェアでMB90560/565シリーズ CPUに対し、タスク切替えのための、割込み要求の出力および解除を行うことができます。

遅延割込み発生モジュールのブロックダイヤグラム

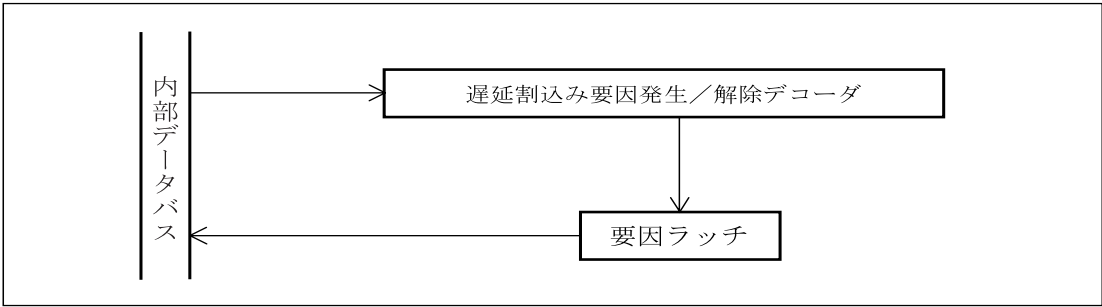


図 15.1-1 遅延割込み発生モジュールのブロックダイヤグラム

15.2 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 (DIRR) について説明します。

遅延割込み要因発生/解除レジスタ (DIRR)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00009F _H	—	—	—	—	—	—	—	R0	XXXXXXX0 _B
	—	—	—	—	—	—	—	R/W	

R/W : リード・ライト可能
 X : 不定
 — : 未定義ビット

図 15.2-1 遅延割込み要因発生/解除レジスタ (DIRR)

表 15.2-1 遅延割込み要因/解除レジスタ(DIRR)の各ビットの機能説明

ビット名		機 能
bit15 bit9	- : 未定義ビット	<ul style="list-style-type: none"> ・ 読出しを行った場合は、不定値となります。 ・ 書込みを行った場合は、動作に影響しません。
bit8	R0 : 遅延割込み 要求 出力ビット	<ul style="list-style-type: none"> ・ 遅延割込み要求の発生/解除を設定します。 ・ "1"を設定した場合は、遅延割込みの要求を出力します。 ・ "0"を設定した場合は、遅延割込み要求をクリアします。 ・ リセットを設定した場合は、割込み要因解除状態 ("0"にクリア) になります。

15.3 遅延割込み発生モジュールの動作説明

ソフトウェアにて、遅延割込み要因/解除レジスタ（DIRR）の遅延割込み要求出力ビット（R0）に"1"を設定した場合は、割込みコントローラに、遅延割込み要求を出力します。

遅延割込み発生モジュールの動作

ソフトウェアにて、遅延割込み要因/解除レジスタ（DIRR）の遅延割込み要求出力ビット（R0）に"1"を設定した場合は、割込みコントローラに割込み要求を出力します。遅延割込み以外の割込み要求が、遅延割込みより優先順位が低い、あるいは遅延割込み以外の割込み要求がない場合に、割込みコントローラは、CPUに対して割込み要求を出力します。CPUはプロセッサステータスレジスタ（PS）内のインタラプトレベルマスクレジスタ（ILM）と割込み要求レベルを比較し、割込み要求レベルが、インタラプトレベルマスクレジスタ（ILM）より強い場合には、現在実行中の命令が終了すると、ハードウェア割込み処理マイクロプログラムを起動し、遅延割込み処理ルーチンを実行します。割込み処理ルーチン内で遅延割込み要因/解除レジスタ（DIRR）の遅延割込み要求出力ビット（R0）に"0"を設定した場合、遅延割込み要因をクリアし、タスクの切換えを行います。

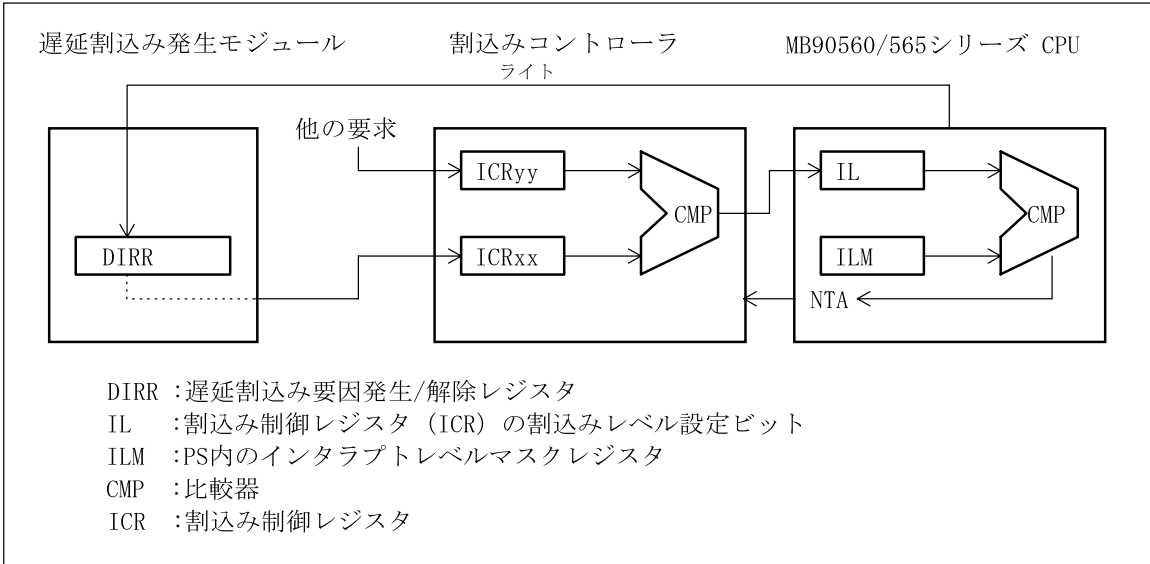


図 15.3-1 遅延割込み発生モジュールの動作

15.4 遅延割込み発生モジュール使用上の注意

遅延割込み発生モジュールを使用する場合の注意点を示します。

遅延割込み発生モジュール使用上の注意

遅延割込み要求

割込み処理ルーチンで割込み処理を終了した後、または、割込み処理ルーチンを実行している間に、遅延割込み発生要因/解除レジスタ（DIRR）の遅延割込み要求出力ビット（R0）に"0"を設定していない場合は、割込み処理から復帰できません。

第16章 8/10ビットA/Dコンバータ

この章では、MB90560/565シリーズの8/10ビットA/Dコンバータの機能と動作について説明します。

- 16.1 8/10ビットA/Dコンバータの概要
- 16.2 8/10ビットA/Dコンバータの構成
- 16.3 8/10ビットA/Dコンバータの端子
- 16.4 8/10ビットA/Dコンバータのレジスタ
- 16.5 8/10ビットA/Dコンバータの割込み
- 16.6 8/10ビットA/Dコンバータの動作説明
- 16.7 8/10ビットA/Dコンバータ使用上の注意
- 16.8 8/10ビットA/Dコンバータのプログラム例-1
(単発変換モードのEI²OS起動例)
- 16.9 8/10ビットA/Dコンバータのプログラム例-2
(連続変換モードのEI²OS起動例)
- 16.10 8/10ビットA/Dコンバータのプログラム例-3
(停止変換モードのEI²OS起動例)

16.1 8/10ビットA/Dコンバータの概要

8/10ビットA/Dコンバータには，RC逐次比較変換方式でアナログ入力電圧を10ビットもしくは8ビットの値に変換する機能があります。

8/10ビットA/Dコンバータの機能

以下に8/10ビットA/Dコンバータの機能について示します。

- 変換時間は，最小6.13 μ s(マシクロック周波数16MHzの場合，サンプリング時間を含む)です。
- サンプリング時間は，最小2.0 μ s(マシクロック周波数16MHzの場合)です。
- 変換方式は，サンプルホールド回路付RC逐次変換比較方式です。
- 10ビット，または8ビットの分解能が設定できます。
- 入力信号は，8チャンネルのアナログ入力端子からプログラムで設定可能です。
- A/D変換終了時に割込み要求を出力し，EI²OSを起動できます。
- 割込み許可状態でA/D変換を実行した場合，変換データ保護機能が働きます。
- 変換の起動要因は，ソフトウェア，16ビットリロードタイマ1出力(立上りエッジ)，16ビットフリーランタイマのゼロ検出エッジから設定できます。

変換モードは，表 16.1-1に示すように4種類あります。

表 16.1-1 8/10ビットA/Dコンバータの変換モード

変換モード	シングル変換動作	スキャン変換動作
単発変換モード1 単発変換モード2	設定したチャンネル(1チャンネル)を1回変換後，終了	連続した複数のチャンネル(最大8チャンネルまで設定可能)を1回変換後，終了
連続変換モード	設定したチャンネル(1チャンネル)を繰り返し変換	連続した複数のチャンネル(最大8チャンネルまで設定可能)を繰り返し変換
停止変換モード	設定したチャンネル(1チャンネル)を1回変換実行後，一時停止し，次の起動がかかるまで待機	連続した複数のチャンネル(最大8チャンネルまで設定可能)を1回変換実行後，一時停止し，次の起動がかかるまで待機

8/10ビットA/Dコンバータの割込みとEI²OS

表 16.1-2 8/10ビットA/Dコンバータの割込みとEI²OS

割込み番号	割込み制御レジスタ		ベクタテーブルアドレス			EI ² OS
	レジスタ名	アドレス	下位	上位	バンク	
#11 (0B _H)	ICR00	0000B0 _H	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	

:使用可能

16.2 8/10ビットA/Dコンバータの構成

8/10ビットA/Dコンバータは、以下の9種類のブロックで構成されています。

- ・ A/D制御ステータスレジスタ (ADCS0/ADCS1)
- ・ A/Dデータレジスタ (ADCR0/ADCR1)
- ・ クロックセクタ (A/D変換起動用入力クロックセクタ)
- ・ デコーダ
- ・ アナログチャンネルセクタ
- ・ サンプルホールド回路
- ・ D/Aコンバータ
- ・ コンパレータ
- ・ コントロール回路

8/10ビットA/Dコンバータのブロックダイアグラム

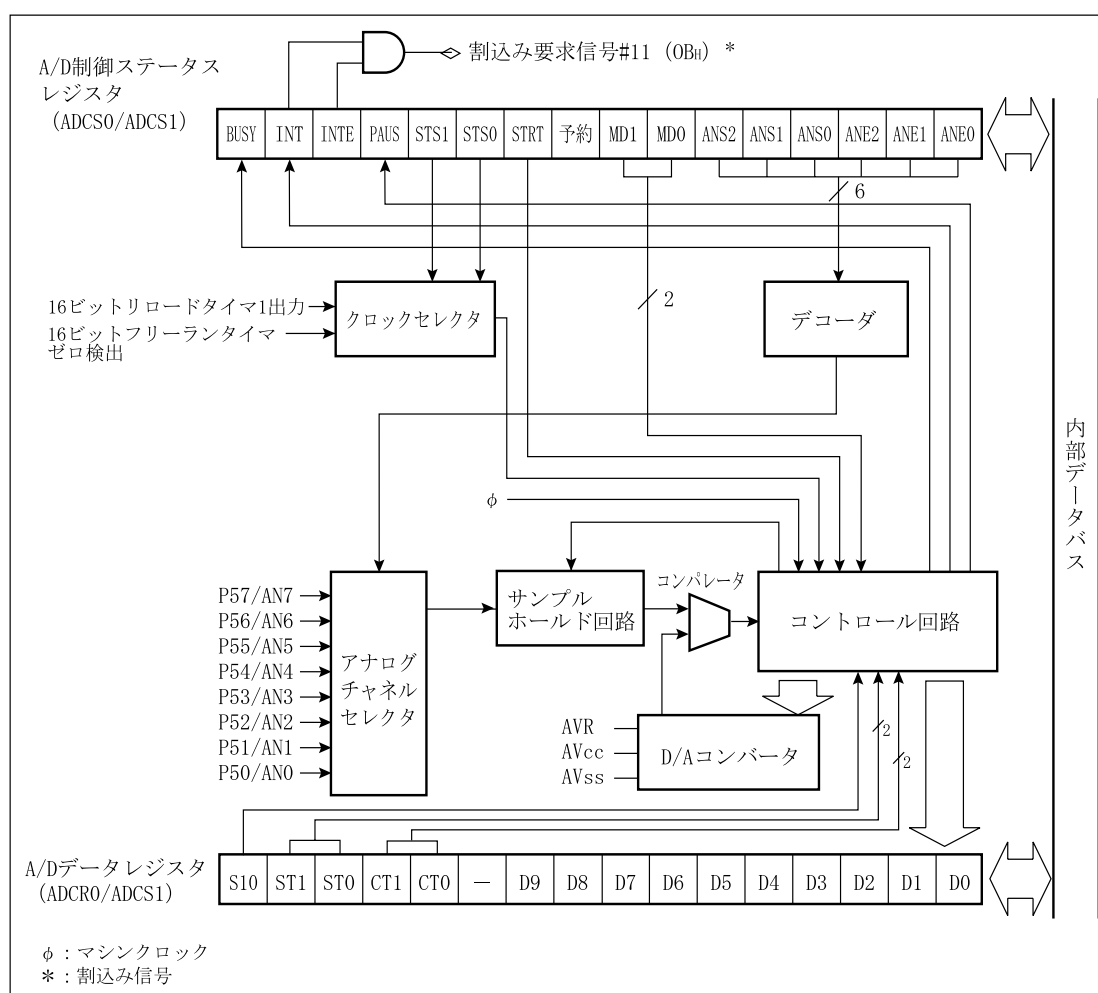


図 16.2-1 8/10ビットA/Dコンバータのブロックダイアグラム

A/D制御ステータスレジスタ (ADCS0/ADCS1)

A/D制御ステータスレジスタ (ADCS0) は、A/D変換モードの設定とA/D変換開始 / 終了チャンネルの設定を行う機能があります。

A/D制御ステータスレジスタ (ADCS1) は、A/D変換起動要因の設定、割込み要求の許可 / 禁止の設定、割込み要求の状態確認、A/D変換の停止中 / 動作中の確認を行う機能があります。

A/Dデータレジスタ (ADCR0/ADCR1)

A/D変換結果を格納するレジスタですが、A/D変換の分解能の設定、A/D変換時のサンプリング時間の設定、およびA/D変換時のコンペア時間の設定を行う機能もあります。

クロックセレクト

A/D変換起動クロックを設定するセレクトです。起動クロックには、16ビットリロードタイマ1出力または、16ビットフリーランタイマのゼロ検出が設定できます。

デコーダ

A/D制御ステータスレジスタ (ADCS0) のA/D変換終了チャンネル設定ビット (ANE0 ~ ANE2)、A/D変換開始チャンネル設定ビット (ANS0 ~ ANS2) の設定値から使用するアナログ入力端子を設定する回路です。

アナログチャンネルセレクト

8チャンネルのアナログ入力端子の中から使用する端子を設定する回路です。

サンプルホールド回路

アナログチャンネルセレクトで設定された端子からの入力電圧を保持する回路です。A/D変換を起動した直後の入力電圧を保持することで、A/D変換中 (比較中) は、A/D変換における入力電圧変動の影響を受けません。

D/Aコンバータ

保持された入力電圧と比較するための、基準電圧を発生します。

コンパレータ

保持された入力電圧と、D/Aコンバータの出力電圧を比較し、大小を判定します。

コントロール回路

コンパレータからの大小信号で、A/D変換値を決定します。A/D変換が終了した場合、変換結果をA/Dデータレジスタ (ADCR0/ADCR1) に格納し、割込み要求を出力します。

16.3 8/10ビットA/Dコンバータの端子

8/10ビットA/Dコンバータの端子，および端子部のブロックダイアグラムを示します。

8/10ビットA/Dコンバータの端子

A/Dコンバータの端子は，入出力ポートと兼用になっています。

表 16.3-1 8/10ビットA/Dコンバータの端子

機 能	端子名	端子機能	入出力形式	プルアップ 設定	スタンバイ 制御	端子の使用に 必要な設定
チャンネル0	P50/AN0	ポート5 入出力/ アナログ 入力	CMOS出力/ CMOSヒステリ シス入力 または アナログ入力	なし	なし	入力ポートに設定 (DDR5:bit15 ~ bit8="0") アナログ入力に設定 (ADER:bit15 ~ bit8="1")
チャンネル1	P51/AN1					
チャンネル2	P52/AN2					
チャンネル3	P53/AN3					
チャンネル4	P54/AN4					
チャンネル5	P55/AN5					
チャンネル6	P56/AN6					
チャンネル7	P57/AN7					

8/10ビットA/Dコンバータの端子部のブロックダイヤグラム

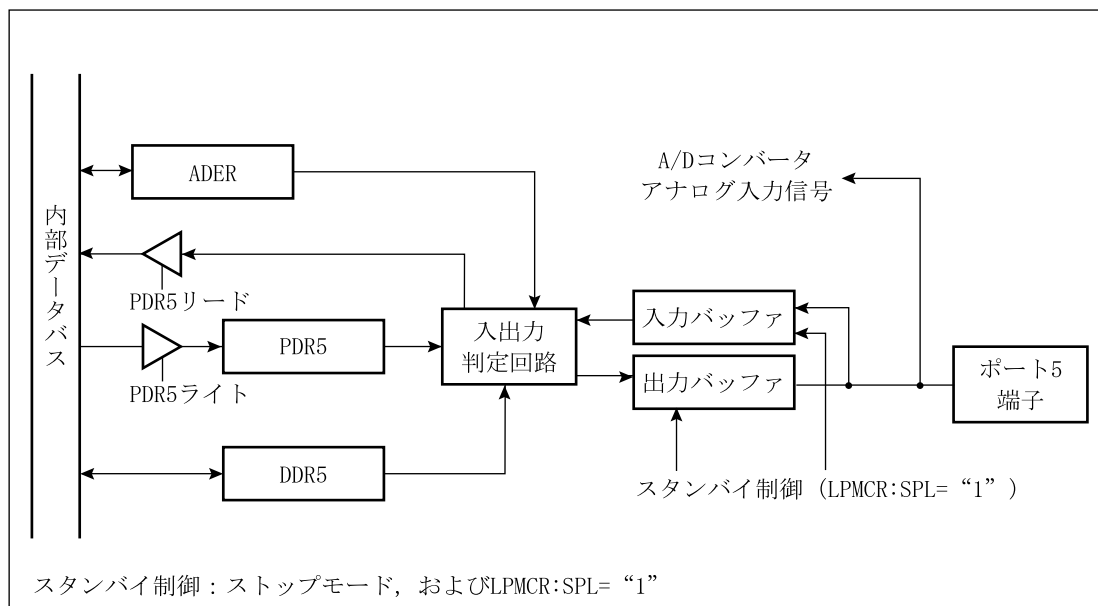


図 16.3-1 P50/AN0 ~ P57/AN7端子部のブロックダイヤグラム

< 注意事項 >

- ・ 入力ポートとして使用する端子は，ポート方向レジスタ（DDR5）の対応するビット（bit15～bit8）に"0"を設定し，アナログ入力許可レジスタ（ADER）の対応するビット（bit15～bit8）に"0"を設定してください。
- ・ アナログ入力端子として使用する端子は，アナログ入力許可レジスタ（ADER）の対応するビット（bit15～bit8）に"1"を設定してください。ポートデータレジスタ（PDR5）の読出し値は"00H"になります。

16.4 8/10ビットA/Dコンバータのレジスタ

8/10ビットA/Dコンバータのレジスタ一覧を示します。

8/10ビットA/Dコンバータのレジスタ一覧

アドレス	bit15..... bit8	bit7..... bit0
000017 _H	アナログ入力許可レジスタ (ADER)	
000035 _H , 000034 _H	A/D制御ステータスレジスタ (ADCS1)	A/D制御ステータスレジスタ (ADCS0)
000037 _H , 000036 _H	A/Dデータレジスタ (ADCR1)	A/Dデータレジスタ (ADCR0)

図 16.4-1 8/10ビットA/Dコンバータのレジスタ一覧

16.4.1 A/D制御ステータスレジスタ (ADCS1)

A/D制御ステータスレジスタ (ADCS1) は、A/D変換起動要因の設定、割込み要求の許可 / 禁止の設定、割込み要求の状態確認、A/D変換の停止中 / 動作中の確認を行う機能があります。

A/D制御ステータスレジスタ (ADCS1)

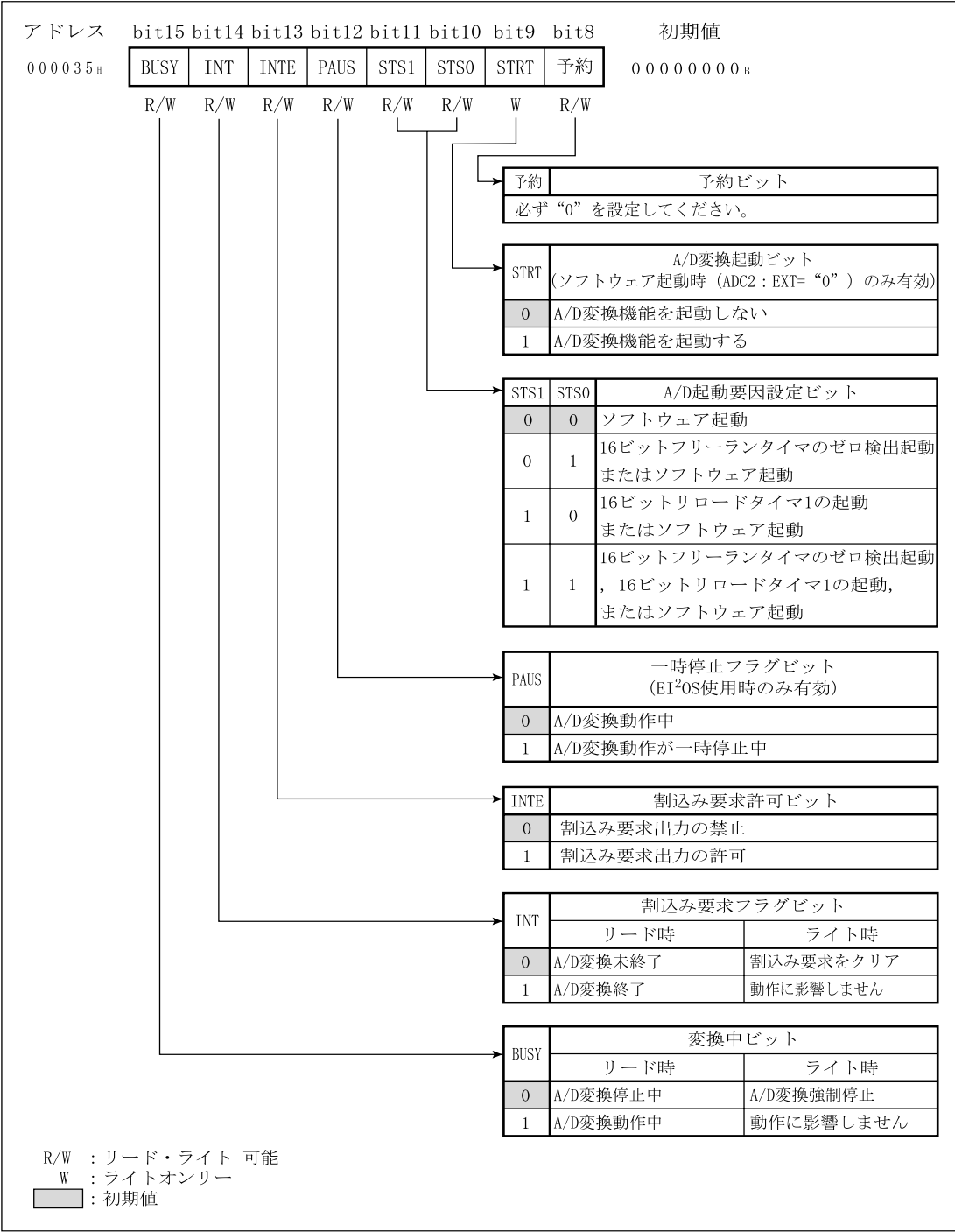


図 16.4-2 A/D制御ステータスレジスタ (ADCS1)

表 16.4-1 A/D制御ステータスレジスタ1 (ADCS1) の各ビットの機能説明

ビット名		機 能
bit15	BUSY : 変換中ビット	<ul style="list-style-type: none"> ・A/Dコンバータの動作表示ビットです。 ・"0"がセットされている場合は、A/D変換停止中であることを示します。 ・"1"がセットされている場合は、A/D変換動作中であることを示します。 ・"0"を設定した場合は、A/D変換動作を強制停止します。 ・"1"を設定した場合は、動作に影響しません。 <p><注記> A/D変換動作の強制停止と起動 (BUSY="0", STRT="1") は、同時に設定しないでください。</p>
bit14	INT : 割込み要求フラグビット	<ul style="list-style-type: none"> ・割込み要求のフラグビットです。 ・A/D変換結果がA/Dデータレジスタ (ADCR0/ADCR1) に格納されると、"1"がセットされます。 ・割込み要求許可ビット (INTE) に"1"を設定している場合に、"1"がセットされると、割込み要求を出力します。 ・"0"を設定した場合は、割込み要求がクリアされます。 ・"1"を設定した場合は、動作に影響しません。 ・EI²OSを使用すると、"0"にクリアされます。 <p><注記> 割込み要求をクリアする場合は、A/D変換を停止してください。</p>
bit13	INTE : 割込み要求許可ビット	<ul style="list-style-type: none"> ・割込み要求を許可するビットです。 ・"1"を設定した場合は、割込み要求フラグビット (INT) に"1"がセットされると、割込み要求を出力します。 ・EI²OSを使用する場合は、"1"を設定してください。
bit12	PAUS : 一時停止フラグビット	<ul style="list-style-type: none"> ・A/D変換動作が一時停止した場合に、"1"がセットされます。 ・連続変換モードでEI²OSを使用した場合、A/D変換が終了しても、前データのメモリへの転送が完了していないと、"1"がセットされ、A/D変換動作を一時停止し、変換データをA/Dデータレジスタ (ADCR0/ADCR1) に格納しません。 ・前データのメモリへのデータ転送が完了した場合、"0"にクリアされ、A/D変換動作を再開します。 <p><注記> EI²OSを使用した場合に有効です。</p>
bit11 bit10	STS1, STS0 : A/D起動要因 設定ビット	<ul style="list-style-type: none"> ・A/D変換の起動要因の設定を行うビットです。 ・起動要因が兼用になっている場合は、最初に発生した起動要因で起動します。 <p><注記> ・起動要因は、設定と同時に変更されますので、A/D変換動作中に設定する場合は、目的とする起動要因がない状態で切り替えてください。</p>
bit9	STRT : A/D変換起動 ビット	<ul style="list-style-type: none"> ・A/D変換動作を起動するビットです。 ・"1"を設定した場合は、A/D変換が起動します。 ・停止変換モード時は、STRTビットによる再起動はかかりません。 ・読出し値は"1"です。 <p><注記> A/D変換動作の強制停止と起動 (BUSY="0", STRT="1") は同時に行わないでください。</p>
bit8	予約 : 予約ビット	必ず"0"を設定してください。

16.4.2 A/D制御ステータスレジスタ (ADCS0)

A/D制御ステータスレジスタ (ADCS0) は、A/D変換モードの設定とA/D変換開始 / 終了チャネルの設定を行う機能があります。

A/D制御ステータスレジスタ (ADCS0)

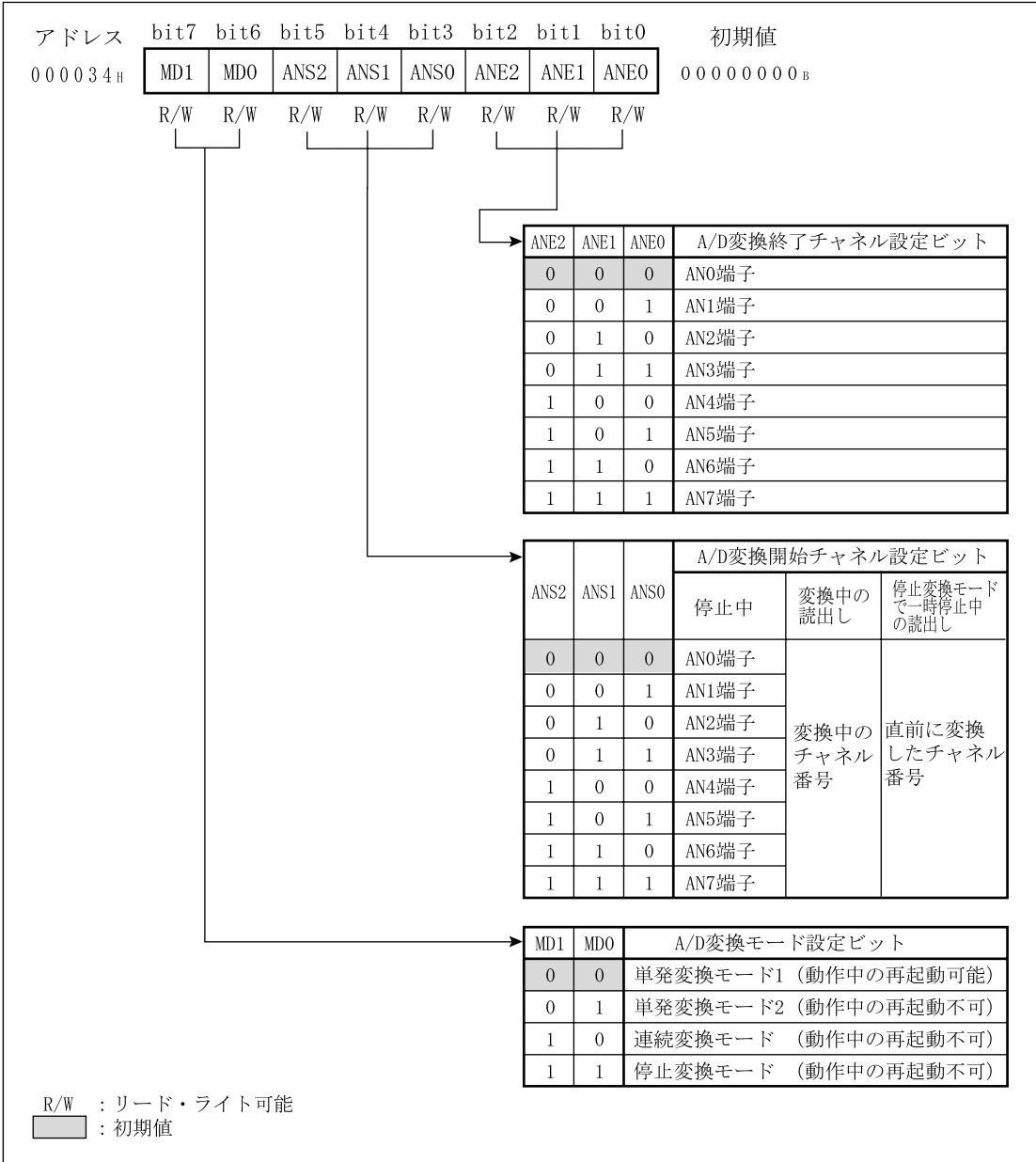


図 16.4-3 A/D制御ステータスレジスタ (ADCS0)

表 16.4-2 A/D制御ステータスレジスタ (ADCS0) の各ビットの機能説明

ビット名		機 能
bit7 bit6	MD1, MD0 : A/D変換モード 設定ビット	<ul style="list-style-type: none"> ・ A/D変換モードを設定するビットです。 ・ 単発変換モード1, 単発変換モード2, 連続変換モード, 停止変換モードが設定できます。 ・ 単発変換モード1 : A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定したチャンネルからA/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) で設定したチャンネルまでA/D変換後, 終了します。動作中の再起動ができます。 ・ 単発変換モード2 : A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定したチャンネルからA/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) で設定したチャンネルまでA/D変換後, 終了します。動作中の再起動はできません。 ・ 連続変換モード : A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定したチャンネルからA/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) で設定したチャンネルまでのA/D変換を, 変換中ビット (BUSY) で強制停止するまで繰り返します。動作中の再起動はできません。 ・ 停止変換モード : A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定したチャンネルからA/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) で設定したチャンネルまでのA/D変換を1チャンネルごとに一時停止しながら, 変換中ビット (BUSY) で強制停止するまで繰り返します。動作中の再起動はできません。一時停止中の再起動は, A/D起動要因設定ビット (STS1, STS0) で設定した起動要因の発生によって異なります。 <p><注記> 単発, 連続, 停止の各変換モードの再起動不可は16ビットフリーランタイム0検出, 16ビットリロードタイマ1, ソフトウェア全ての起動に適用されます。</p>
bit5 bit4 bit3	ANS2, ANS1, ANS0 : A/D変換開始 チャンネル設定 ビット	<ul style="list-style-type: none"> ・ A/D変換の開始チャンネルの設定, および変換中チャンネル番号の確認を行うビットです。 ・ A/D変換を起動した場合は, A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定されたチャンネルからA/D変換を開始します。 ・ A/D変換中は, 変換中のチャンネル番号が読み出されます。停止変換モードでの一時停止中は, 直前に変換したチャンネルの番号が読み出されます。
bit2 bit1 bit0	ANE2, ANE1, ANE0 : A/D変換終了 チャンネル設定 ビット	<ul style="list-style-type: none"> ・ A/D変換の終了チャンネルの設定を行うビットです。 ・ A/D変換を起動した場合は, A/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) ビットで設定されたチャンネルまでA/D変換を行います。 ・ A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定したチャンネルと同じチャンネルを設定した場合は, 設定したチャンネルがA/D変換されます。 ・ 連続変換モード, または停止変換モードを設定した場合は, A/D変換終了チャンネル設定ビット (ANE2 ~ ANE0) で設定したチャンネルまでのA/D変換が終了すると, A/D変換開始チャンネル設定ビット (ANS2 ~ ANS0) で設定した開始チャンネルに戻り, A/D変換を繰り返します。開始チャンネルの設定値が, 終了チャンネルの設定値より大きい値に設定されている場合は, 開始チャンネルからAN7までA/D変換し, さらにAN0から終了チャンネルまでA/D変換し, 1度目の変換動作を終了します。

16.4.3 A/Dデータレジスタ (ADCR0/ADCR1)

A/Dデータレジスタ (ADCR0/ADCR1) は、A/D変換結果を格納するレジスタですが、A/D変換の分解能を設定、A/D変換時のサンプリング時間の設定、およびA/D変換時のコンペア時間設定を行う機能もあります。

A/Dデータレジスタ (ADCR0/ADCR1)

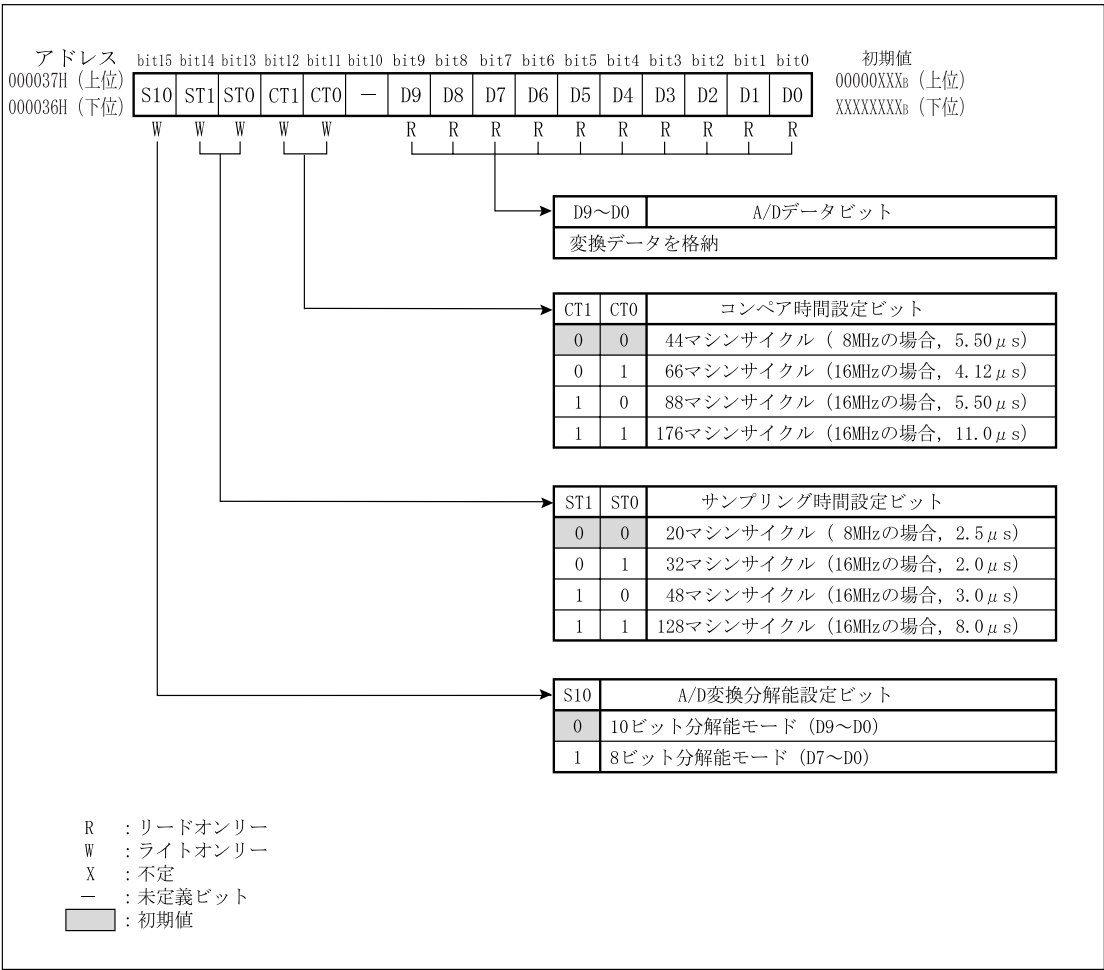


図 16.4-4 A/Dデータレジスタ (ADCR0/ADCR1)

表 16.4-3 A/Dデータレジスタ (ADCR0/ADCR1) の各ビットの機能説明

ビット名		機 能
bit15	S10 : A/D変換分解能 設定ビット	<ul style="list-style-type: none"> ・ A/D変換の分解能を設定するビットです。 ・ "0"を設定した場合は、10ビット分解能が設定されます。 ・ "1"を設定した場合は、8ビット分解能が設定されます。 <p><注記> 分解能によって、使用されるA/Dデータビットが異なります。 10ビット分解能モードでは、D9～D0ビットが使用されます。 8ビット分解能モードでは、D7～D0ビットが使用されます。</p>
bit14 bit13	ST1,ST0 : サンプリング時間 設定ビット	<ul style="list-style-type: none"> ・ A/D変換のサンプリング時間を設定するビットです。 ・ A/D変換が起動されると、サンプリング時間設定ビット (ST1,ST0) で設定した時間、アナログ入力に取り込まれます。 <p><注記> ・ "00_b"を設定する場合は、マシクロック周波数を8MHz以下としてください。 ・ マシクロック周波数16MHzの場合に、"00_b"を設定すると、正常なアナログ変換値が得られない場合があります。</p>
bit12 bit11	CT1,CT0 : コンペア時間設定 ビット	<ul style="list-style-type: none"> ・ A/D変換時のコンペア時間を設定するビットです。 ・ アナログ入力に取り込まれた (サンプリング時間経過) 後、コンペア時間設定ビット (CT1,CT0) で設定した時間後に、A/D変換結果が確定し、10ビット分解能モードでは、A/Dデータビット (D9～D0) に格納され、8ビット分解能モードでは、A/Dデータビット (D7～D0) に格納されます。 <p><注記> ・ "00_b"を設定する場合は、マシクロック周波数を8MHz以下としてください。 ・ マシクロック周波数16MHzの場合に、"00_b"を設定すると、正常なアナログ変換値が得られない場合があります。</p>
bit10	- : 未定義ビット	<ul style="list-style-type: none"> ・ 読出しを行った場合は、不定値となります。 ・ 設定した値は、動作に影響しません。
bit9 ~ bit0	D9～D0: A/Dデータビット	<ul style="list-style-type: none"> ・ A/D変換結果が格納されるビットです。1回のA/D変換終了ごとに書き換えられます。 ・ 通常は、最終変換値が格納されます。 ・ 初期値は不定です。 <p><注記> A/D変換データ保護機能があります (詳細は、「16.6 8/10ビットA/Dコンバータの動作説明」を参照してください)。 A/D変換中に、A/Dデータビットにデータを書込まないでください。</p>

< 注意事項 >

- ・ A/D変換分解能設定ビット (S10) を書き換える場合は、必ずA/D変換動作前にA/D変換動作を停止させてから書き換えてください。A/D変換動作開始後に書き換えた場合は、A/Dデータレジスタ (ADCR0/ADCR1) の内容は不定となります。
- ・ A/Dデータレジスタ (ADCR0/ADCR1) を読み出す場合、10ビット分解能モードを設定した場合は、必ずワード転送命令 (MOVW A, 002EHなど) を使用してください。

16.5 8/10ビットA/Dコンバータの割込み

8/10ビットA/Dコンバータは、A/D変換でA/Dデータレジスタにデータがセットされると、割込み要求を出力します。拡張インテリジェントI/Oサービス (EI²OS) にも対応しています。

8/10ビットA/Dコンバータの割込み

表 16.5-1 8/10ビットA/Dコンバータの割込み制御ビットと割込み要因

	8/10ビットA/Dコンバータ
割込み要求フラグビット	ADCS1 : INT="1"
割込み要求許可ビット	ADCS1 : INTE="1"
割込み要因	A/D変換結果のA/Dデータレジスタへ格納

A/D変換動作が起動され、A/D変換結果がA/Dデータレジスタ (ADCRO/ADCR1) に格納されると、A/D制御ステータスレジスタ (ADCS1) の割込み要求フラグビット (INT) に"1"がセットされ、割込み要求許可ビット (INTE) に"1"が設定されていると、割込み要求を出力します。

8/10ビットA/Dコンバータの割込みとEI²OS

表 16.5-2 8/10ビットA/Dコンバータの割込みとEI²OS

割込み番号	割込み制御レジスタ		ベクタテーブルのアドレス			EI ² OS
	レジスタ名	アドレス	下位	上位	バンク	
#11 (0B _H)	ICR00	0000B0 _H	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	

: 使用可能

8/10ビットA/DコンバータのEI²OS機能

10ビットA/Dコンバータは、EI²OS機能を使用してA/D変換結果をメモリに転送することができます。EI²OS機能を使用する場合は、変換データ保護機能が働き、A/D変換データがメモリに転送され、A/D制御ステータスレジスタ (ADCS1) の割込み要求フラグビット (INT) が"0"にクリアされるまで、A/D変換は一時停止されるので、データの欠落を防止することができます。

16.6 8/10ビットA/Dコンバータの動作説明

8/10ビットA/Dコンバータには、単発変換モード1、単発変換モード2、連続変換モード、停止変換モードの4種類のモードがあります。

各モードの動作説明をします。

単発変換モード1、単発変換モード2の動作

単発変換モード1、単発変換モード2は、A/D制御ステータスレジスタ（ADCS0）のA/D変換開始チャンネル設定ビット（ANS2～ANS0）で設定した開始チャンネルからA/D変換終了チャンネル設定ビット（ANE2～ANE0）で設定した終了チャンネルまでのアナログ入力をA/D変換後、終了します。

開始チャンネルと終了チャンネルが同じ場合は、A/D変換開始チャンネル設定ビット（ANS2～ANS0）で設定した1チャンネルがA/D変換されます。

単発変換モード1は、動作中の再起動ができますが、単発変換モード2は動作中の再起動ができません。単発変換モード1、単発変換モード2で動作させるには、図 16.6-1に示す設定が必要です。

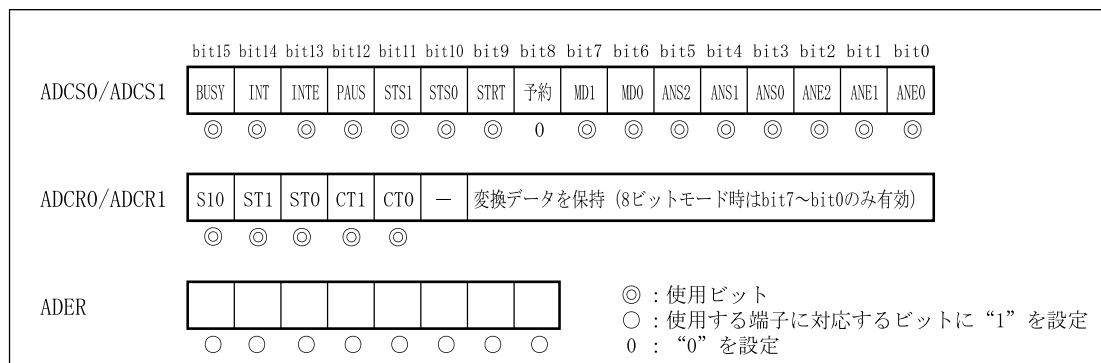


図 16.6-1 単発変換モードでの設定

参考：

単発変換モードでの変換順序の例を以下に示します。

ANS="000_B", ANE="011_B"の場合：AN0 AN1 AN2 AN3 終了

ANS="110_B", ANE="010_B"の場合：AN6 AN7 AN0 AN1 AN2 終了

ANS="011_B", ANE="011_B"の場合：AN3 終了

連続変換モードの動作

連続変換モードは、A/D制御ステータスレジスタ（ADCS0）のA/D変換開始チャネル設定ビット（ANS2～ANS0）で設定した開始チャネルからA/D変換終了チャネル設定ビット（ANE2～ANE0）で設定した終了チャネルまでのアナログ入力をA/D変換後、A/D変換開始チャネル設定ビット（ANS2～ANS0）で設定したアナログ入力に戻り、A/D変換を繰り返します。

開始チャネルと終了チャネルが同じ場合は、A/D変換開始チャネル設定ビット（ANS2～ANS0）で設定したチャネルのA/D変換を繰り返します。

A/D制御ステータスレジスタ（ADCS1）の変換中ビット（BUSY）に"0"を設定するまで、A/D変換は停止しません。動作中の再起動はできません。連続変換モードで動作させるには、図 16.6-2に示す設定が必要です。

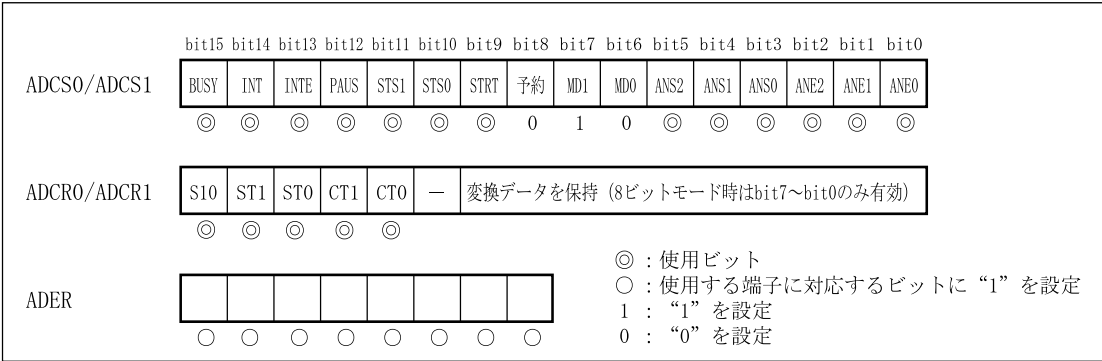


図 16.6-2 連続変換モードでの設定

参考：

連続変換モードでの変換順序の例を以下に示します。

ANS="000_B", ANE="011_B"の場合：AN0 AN1 AN2 AN3 AN0 繰り返し
ANS="110_B", ANE="010_B"の場合：AN6 AN7 AN0 AN1 AN2 AN6 繰り返し
ANS="011_B", ANE="011_B"の場合：AN3 AN3 繰り返し

停止変換モードの動作

停止変換モードは、A/D制御ステータスレジスタ（ADCS0）のA/D変換開始チャネル設定ビット（ANS2～ANS0）で設定した開始チャネルからA/D変換終了チャネル設定ビット（ANE2～ANE0）で設定した終了チャネルまでのアナログ入力を1チャネルごとに一時停止しながらA/D変換後、A/D変換開始チャネル設定ビット（ANS2～ANS0）で設定したアナログ入力に戻り、A/D変換と一時停止の動作を繰り返します。

開始チャネルと終了チャネルが同じ場合は、A/D変換開始チャネル設定ビット（ANS2～ANS0）で設定したチャネルのA/D変換を繰り返します。

一時停止時のA/D変換の再起動は、A/D制御ステータスレジスタ（ADCS1）のA/D起動要因設定ビット（STS1,STS0）で設定した起動要因によって異なります。

A/D制御ステータスレジスタ（ADCS1）の変換中ビット（BUSY）に“0”を設定するまで、A/D変換は停止しません。動作中の再起動はできません。停止変換モードで動作させるには、図 16.6-3に示す設定が必要です。

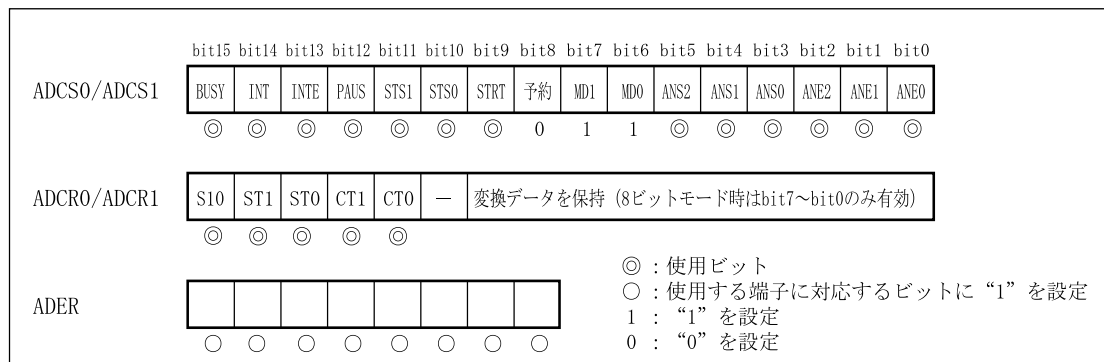


図 16.6-3 停止変換モードでの設定

参考：

停止変換モードでの変換順序の例を以下に示します。

ANS="000_B", ANE="011_B"の場合：AN0 一時停止 AN1 一時停止 AN2 一時停止 AN3 一時停止 AN0 繰り返し
ANS="110_B", ANE="001_B"の場合：AN6 一時停止 AN7 一時停止 AN0 一時停止 AN1 一時停止 AN6 繰り返し
ANS="011_B", ANE="011_B"の場合：AN3 一時停止 AN3 一時停止 繰り返し

16.6.1 EI²OSを使用した変換動作

8/10ビットA/Dコンバータは，EI²OSを使用して，A/D変換結果をメモリに転送することができます。

EI²OSを使用した変換動作

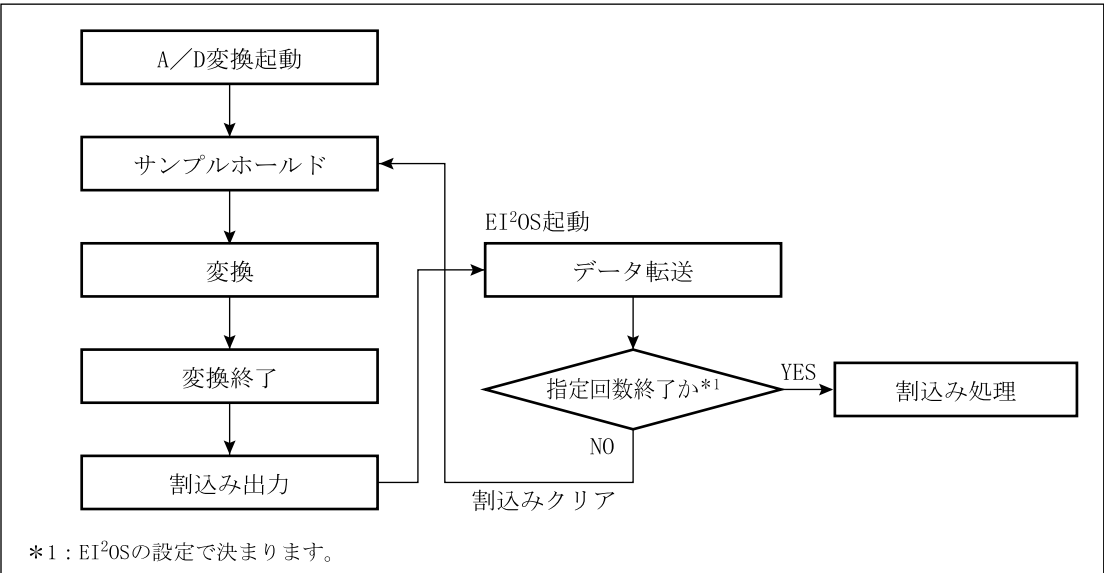


図 16.6-4 EI²OS使用時の動作フローチャート例

EI²OSを使用した場合，変換データ保護機能が働き，連続変換時にもデータ欠落を起こすことなく複数のデータを確実にメモリに転送することができます。

16.6.2 A/D変換データ保護機能

割込み許可状態でA/D変換を実行した場合、変換データ保護機能が働きます。

A/D変換データ保護機能

8/10ビットA/Dコンバータでは、変換データ格納用のデータレジスタが1個ですので、A/D変換を行った場合には、変換終了時にデータレジスタ内の格納データが書換えられます。連続変換モードで、変換データのメモリへの転送が間に合わなかった場合には、格納されているデータが一部欠落します。

データ欠落の対策として、割込み要求許可（ADCS1: INTE="1"）の場合は、以下のようにデータ保護機能が働きます。

EI²OSを使用しない場合のデータ保護機能

変換データがA/Dデータレジスタ（ADCRO/ADCR1）に格納されるとA/D制御ステータスレジスタ1（ADCS1）の割込み要求フラグビット（INT）に"1"がセットされ、A/D変換を一時停止します。割込みルーチン内で、A/Dデータレジスタ（ADCRO/ADCR1）をメモリなどに転送した後、割込み要求フラグビット（INT）を"0"にクリアした場合、A/D変換が再開します。

EI²OSを使用する場合のデータ保護機能

連続変換モードで、EI²OSを使用した場合、A/D変換が終了しても、前データのメモリへの転送が完了していないと、A/D制御ステータスレジスタ1（ADCS1）の一時停止フラグビット（PAUS）に"1"がセットされ、A/D変換動作を一時停止し、変換データをA/Dデータレジスタ（ADCRO/ADCR1）に格納しません。前データのメモリへのデータ転送が完了した場合、一時停止フラグビット（PAUS）は"0"にクリアされ、A/D変換動作を再開します。

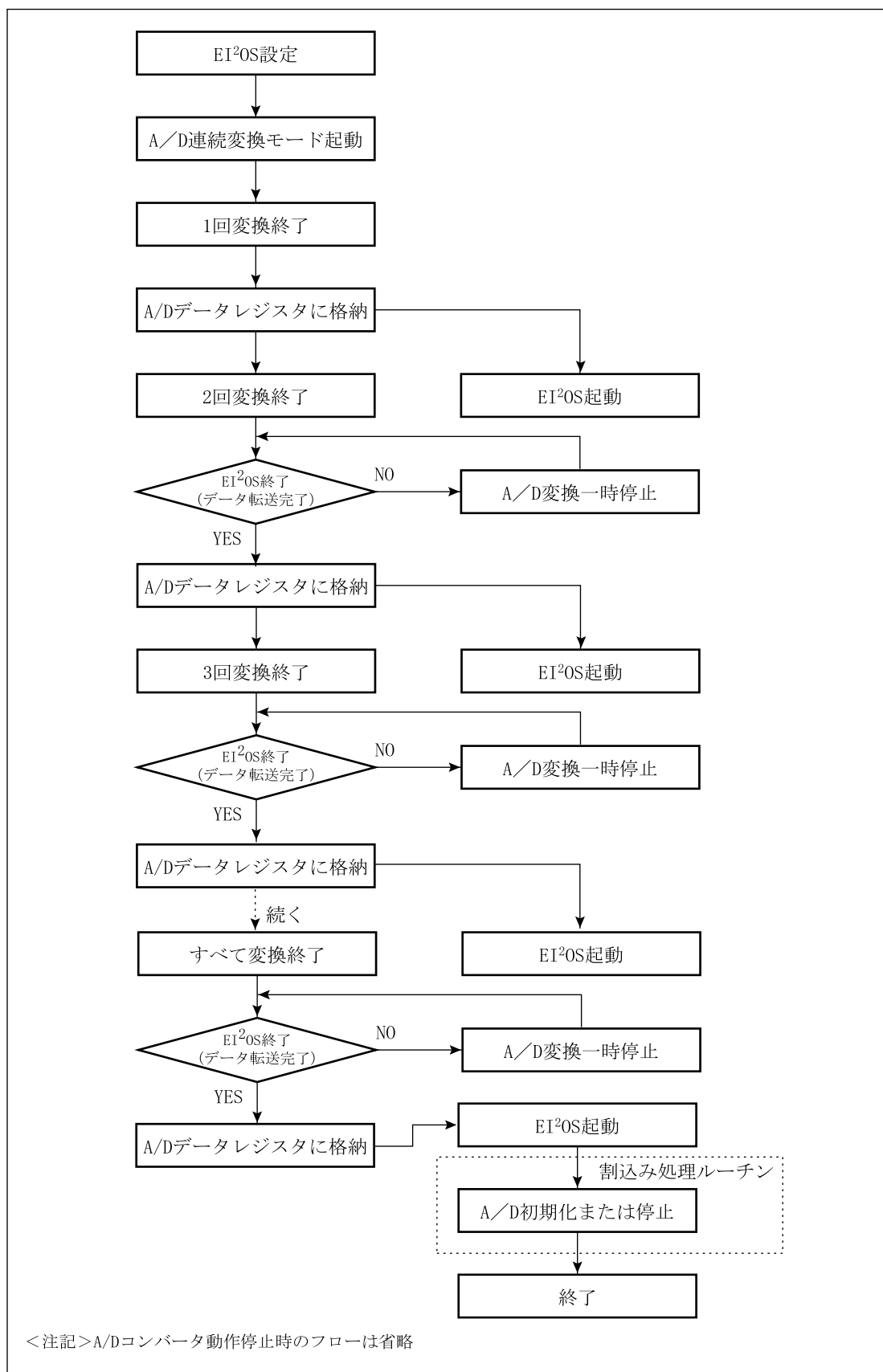


図 16.6-5 EI²OS使用時のデータ保護機能フロー

< 注意事項 >

- ・変換データ保護機能は、割込みが許可されている場合（ADCS1：INTE="1"）に動作します。
- ・EI²OS使用中で、A/D変換が一時停止している場合に、割込みを禁止すると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場合があります。
- ・一時停止中に再起動をかけると、待機データは破壊されます。

16.7 8/10ビットA/Dコンバータ使用上の注意

8/10ビットA/Dコンバータを使用する場合の注意点を示します。

8/10ビットA/Dコンバータ使用上の注意

アナログ入力端子

A/Dコンバータのアナログ入力端子はポート5の入出力端子と兼用になっており、ポート5方向レジスタ (DDR5) とアナログ入力許可レジスタ (ADER) を切替えて使用します。A/Dコンバータのアナログ入力端子として使用する場合は、ポート5方向レジスタ (DDR5) の対応するビット (bit15～bit8) に"0"を設定 (入力ポートに設定) 後、アナログ入力許可レジスタ (ADER) レジスタの対応するビット (bit7～bit0) に"1"を設定 (アナログ入力モードに設定) し、ポート側の入力ゲートを固定してください。ポート入出力モード (ADER: bit7～bit0="0") では中間レベルの信号が入力されると、ゲートに入力リーク電流が流れます。

ポート5方向レジスタ (DDR5)、アナログ入力許可レジスタ (ADER) の詳細は、「8.8.1 ポート5のレジスタ (PDR5, DDR5, ADER)」を参照してください。

内部タイマで使用する場合の注意

A/Dコンバータを内部タイマで起動する場合、A/D制御ステータスレジスタ (ADCS1) のA/D起動要因設定ビット (STS1, STS0) で設定しますが、内部タイマの入力値は、インアクティブ側 (内部タイマの場合は"L") にしてください。アクティブ側にしておくと、A/D制御ステータスレジスタ (ADCS0/ADCS1) へデータ書き込みを行った場合、A/Dコンバータが動作しはじめる場合があります。

A/Dコンバータの電源の投入および切断順序

A/Dコンバータの電源端子 (AVcc, AVR, AVss) とアナログ入力端子 (AN0～AN7) への電圧印加は、必ずデジタル電源 (Vcc) の投入後に行ってください。デバイスの電源切断は、A/Dコンバータの電源、およびアナログ入力電源を切断した後、デジタル電源 (Vcc) を切断してください。AVRは、AVccを超えないように電圧の印加と切断を行ってください。

A/Dコンバータの電源電圧について

ラッチアップ防止のため、A/Dコンバータの電源 (Avcc) は、デジタル電源 (Vcc) の電圧を超えないようにしてください。

16.8 8/10ビットA/Dコンバータのプログラム例-1 (単発変換モードのEI²OS起動例)

単発変換モードのEI²OS起動によるA/D変換処理プログラム例を示します。

単発変換モードのEI²OS起動プログラム例

処理仕様

- アナログ入力AN1～AN3まで変換して終了します。
- 変換データは、200_H～205_H番地に順に転送します。
- 分解能は、10ビットとします。
- 起動は、ソフトウェア起動で行います。

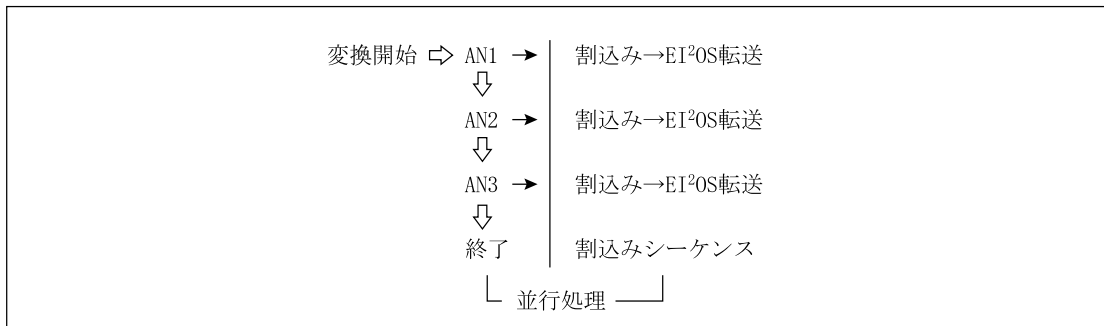


図 16.8-1 単発変換モードのEI²OS起動プログラム例のフロー

コーディング例

```

BAPL EQU 000100H ; バッファアドレスポインタ下位
BAPM EQU 000101H ; バッファアドレスポインタ中位
BAPH EQU 000102H ; バッファアドレスポインタ上位
ISCS EQU 000103H ; EI²OSステータスレジスタ
IOAL EQU 000104H ; I/Oアドレスレジスタ下位
IOAH EQU 000105H ; I/Oアドレスレジスタ上位
DCTL EQU 000106H ; データカウンタ下位
DCTH EQU 000107H ; データカウンタ上位
DDR5 EQU 000015H ; ポート5方向レジスタ
ADER EQU 000017H ; アナログ入力許可レジスタ
ICR00 EQU 0000B0H ; A/D用割込み制御レジスタ
ADCS0 EQU 000034H ; A/D制御ステータスレジスタ
ADCS1 EQU 000035H ;
ADCRO EQU 000036H ; A/Dデータレジスタ
ADCR1 EQU 000037H ;
;-----メインプログラム-----
CODE CSEG
START :
        AND CCR, #0BFH ; スタックポインタ (SP) などは初期化済み
        MOV ICR00, #00H ; とする
        MOV BAPL, #00H ; 割込みディセーブル
        MOV BAPM, #02H ; 割込みレベル0 (最強)
        ; 変換データの格納先アドレスの設定
        ; (200H～205Hを使用)

```

```

MOV      BAPH, #00H      ;
MOV      ISCS, #18H      ; ワードデータ転送, 転送後アドレス + 1,
                          ; I/O メモリに転送
MOV      IOAL, #36H      ; 転送元アドレスポインタとして
MOV      IOAH, #00H      ; アナログデータレジスタのアドレスを
                          ; セット
MOV      DCTL, #03H      ; EI2OS転送を3回, 変換回数と同じにする
MOV      DDR5, #11110001B ; P51 ~ P53を入力に設定
MOV      ADER, #00001110B ; P51/AN1 ~ P53/AN3をアナログ入力に設定
MOV      DCTH, #00H      ;
MOV      ADCS0, #0BH      ; 単発起動, AN1 ~ AN3チャンネルを変換
MOV      ADCS1, #0A2H     ; ソフト起動, A/D変換開始, 割込み許可
MOV      ILM, #07H        ; PS内ILMをレベル7に設定
OR       CCR, #40H        ; 割込みイネーブル
LOOP:    MOV      A, #00H   ; 無限ループ
MOV      A, #01H
BRA     LOOP
; -----割込みプログラム-----
ED_INT1:
MOV      I:ADCS1, #00H     ; A/D停止, 割込みのフラグクリアと禁止
RETI                                           ; 割込みから復帰
CODE     ENDS
; -----ベクタ設定-----
VECT     CSEG      ABS = 0FFH
ORG      OFFD0H    ; 割込み#11 (0BH) にベクタを設定
DSL      ED_INT1
ORG      OFFDCH    ; リセットベクタ設定
DSL      START
DB        00H      ; シングルチップモードに設定
VECT     ENDS
END      START

```

16.9 8/10ビットA/Dコンバータのプログラム例-2 (連続変換モードのEI²OS起動例)

連続変換モードのEI²OS起動によるA/D変換処理プログラム例を示します。

連続変換モードのEI²OS起動プログラム例

処理仕様

- アナログ入力AN3～AN5の変換を2度行い，各チャネルの変換データを2個取得します。
- 変換データは，600_H～60B_H番地に順に転送します。
- 分解能は，10ビットとします。
- 起動は，16ビットリロードタイマ1で行います。

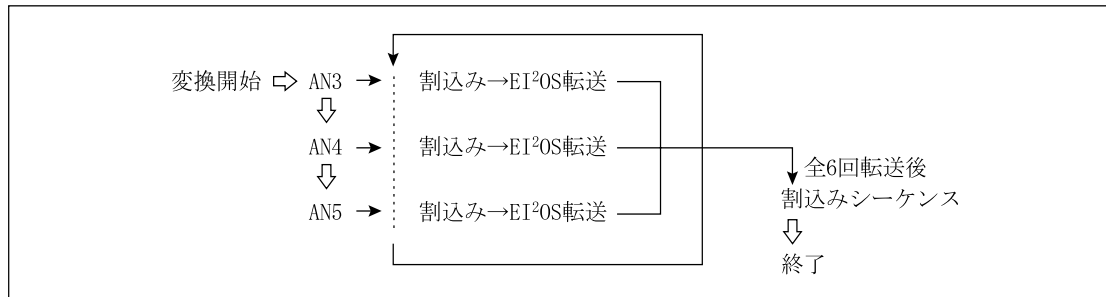


図 16.9-1 連続変換モードのEI²OS起動プログラム例のフロー

コーディング例

BAPL	EQU	000100H	; バッファアドレスポインタ下位
BAPM	EQU	000101H	; バッファアドレスポインタ中位
BAPH	EQU	000102H	; バッファアドレスポインタ上位
ISCS	EQU	000103H	; EI ² OSステータスレジスタ
IOAL	EQU	000104H	; I/Oアドレスレジスタ下位
IOAH	EQU	000105H	; I/Oアドレスレジスタ上位
DCTL	EQU	000106H	; データカウンタカウンタ下位
DCTH	EQU	000107H	; データカウンタ上位
DDR5	EQU	000015H	; ポート5方向レジスタ
ADER	EQU	000017H	; アナログ入力許可レジスタ
ICR00	EQU	0000B0H	; A/DC用割込み制御レジスタ
ADCS0	EQU	000034H	; A/D制御ステータスレジスタ
ADCS1	EQU	000035H	; A/Dデータレジスタ
ADCRO	EQU	000036H	; A/Dデータレジスタ
ADCR1	EQU	000037H	; A/Dデータレジスタ
TMCSR1:L	EQU	000086H	; コントロールステータスレジスタ1
TMCSR1:H	EQU	000087H	; コントロールステータスレジスタ1
TMRLR1	EQU	000088H	; 16ビットリロードレジスタ
TMRHR1	EQU	000089H	; 16ビットリロードレジスタ
; -----メインプログラム-----			
CODE	CSEG		
START :			; スタックポインタ (SP) などは初期化済み
			; と
			; する
AND	CCR, #0BFH		; 割込みディセーブル

```

MOV      ICR00, #08H      ; 割込みレベル0 (最強), 割込み許可
MOV      BAPL, #00H      ; 変換データの格納先アドレスの設定
MOV      BAPM, #06H      ; ( 600H ~ 60BHを使用 )
MOV      BAPH, #00H      ;
MOV      ISCS, #18H      ;
                                ; ワードデータ転送, 転送後アドレス + 1,
                                ; I/O メモリに転送
MOV      IOAL, #36H      ; 転送元アドレスポインタとして
MOV      IOAH, #00H      ; アナログデータレジスタのアドレスを
                                ; セット
MOV      DCTL, #06H      ; EI2OS転送を6回, 3ch x 2回分の転送
MOV      DDR5, #00000000B ; P50 ~ P57を入力に設定
MOV      ADER, #00111000B ; P53/AN3 ~ P55/AN5をアナログ入力に設定
MOV      DCTH, #00H      ;
MOV      ADCS0, #9DH      ; 連続モード, AN3 ~ AN5チャンネルを変換
MOV      ADCS1, #0A8H      ; 16ビットタイマ起動, A/D変換開始, 割込み
                                ; 許可
MOVW     TMRLR1, #0320H    ; タイマ値を設定 800(320H) 100 μs
MOV      TMCRH1, #00H      ; クロックソースを125nsに設定, 外部トリガ
                                ; 禁止
MOV      TMCRL1, #12H      ; タイマ出力禁止, 割込み禁止, リロード許可
MOV      TMCRL1, #13H      ; 16ビットタイマ起動
MOV      ILM, #07H         ; PS内ILMをレベル7に設定
OR        CCR, #40H        ; 割込みイネーブル
LOOP:    MOV      A, #00H    ; 無限ループ
        MOV      A, #01H
        BRA      LOOP
;-----割込みプログラム-----
ED_INT1:
        MOV      I:ADCS1, #80H ; A/Dは停止しない, 割込みのフラグクリアと
                                ; 禁止
        RETI      ; 割込みから復帰
CODE    ENDS
;-----ベクタ設定-----
VECT    CSEG      ABS=OFFH
        ORG      OFFD0H      ; 割込み#11 ( 0BH ) にベクタを設定
        DSL      ED_INT1
        ORG      OFFDCH      ; リセットベクタ設定
        DSL      START
        DB        00H        ; シングルチップモードに設定
VECT    ENDS
        END      START

```

16.10 8/10ビットA/Dコンバータのプログラム例-3 (停止変換モードのEI²OS起動例)

停止変換モードのEI²OS起動によるA/D変換処理プログラム例を示します。

停止変換モードのEI²OS起動プログラム例

処理仕様

- アナログ入力AN3を一定期間で12回変換します。
- 変換データは、600_H～617_H番地に順に転送します。
- 分解能は、10ビットとします。
- 起動は、16ビットリロードタイマ1で行います。

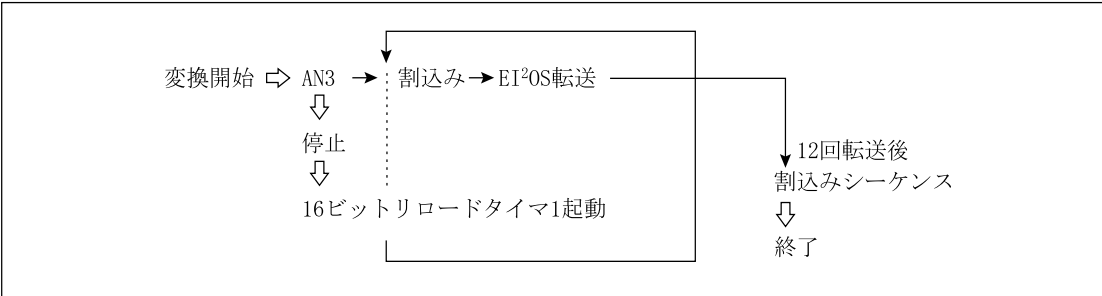


図 16.10-1 停止変換モードのEI²OS起動プログラム例のフロー

コーディング例

```
BAPL EQU 000100H ; バッファアドレスポインタ下位
BAPM EQU 000101H ; バッファアドレスポインタ中位
BAPH EQU 000102H ; バッファアドレスポインタ上位
ISCS EQU 000103H ; EI2OSステータスレジスタ
IOAL EQU 000104H ; I/Oアドレスレジスタ下位
IOAH EQU 000105H ; I/Oアドレスレジスタ上位
DCTL EQU 000106H ; データカウンタ下位
DCTH EQU 000107H ; データカウンタ上位
DDR5 EQU 000015H ; ポート5方向レジスタ
ADER EQU 000017H ; アナログ入力許可レジスタ
ICR00 EQU 0000B0H ; A/DC用割込み制御レジスタ
ADCS0 EQU 000034H ; A/D制御ステータスレジスタ
ADCS1 EQU 000035H ;
ADCRO EQU 000036H ; A/Dデータレジスタ
ADCR1 EQU 000037H ;
TMCSR1:L EQU 000086H ; コントロールステータスレジスタ1
TMCSR1:H EQU 000087H ;
TMRLR1 EQU 000088H ; 16ビットリロードレジスタ
TMRHR1 EQU 000089H ;
;-----メインプログラム-----
CODE CSEG
START : ; スタックポインタ (SP) などは初期化済み
        ; とする
        AND CCR, #0BFH ; 割込みディセーブル
        MOV ICR00, #08H ; 割込みレベル0 (最強)
```



```

MOV      BAPL, #00H      ; 変換データの格納先アドレスの設定
MOV      BAPM, #06H      ; ( 600H ~ 617Hを使用 )
MOV      BAPH, #00H      ;
MOV      ISCS, #19H      ;
                        ; ワードデータ転送, 転送後アドレス + 1,
                        ; I/O メモリに転送, リソースの要求で終了
MOV      IOAL, #36H      ; 転送元アドレスポインタとして
MOV      IOAH, #00H      ; アナログデータレジスタのアドレスを
                        ; セット
MOV      DCTL, #0CH      ; EI2OS転送を12回, 3ch
MOV      DCTH, #00H      ;
MOV      DDR5, #00000000B ; P50 ~ P57を入力に設定
MOV      ADER, #00001000B ; P53/AN3をアナログ入力に設定
MOV      ADCS0, #0DBH    ; 停止モード, AN3チャンネルを変換
MOV      ADCS1, #0A8H    ; 16ビットタイマ起動, A/D変換開始, 割り込み
                        ; 許可
MOVW     TMRLR1, #0320H   ; タイマ値を設定 800(320H) 100 μs
MOV      TMCRH1, #00H    ; クロックソースを125nsに設定, 外部トリガ
                        ; 禁止
MOV      TMCRL1, #12H    ; タイマ出力禁止, 割り込み禁止, リロード許可
MOV      TMCRL1, #13H    ; 16ビットタイマ起動
MOV      ILM, #07H       ; PS内ILMをレベル7に設定
OR       CCR, #40H       ; 割り込みイネーブル
LOOP:    MOV      A, #00H ; 無限ループ
MOV      A, #01H
BRA     LOOP
; -----割り込みプログラム-----
ED_INT1:
MOV      I:ADCS1, #80H   ; A/Dは停止しない, 割り込みのフラグクリアと
                        ; 禁止
RETI                                           ; 割り込みから復帰
CODE     ENDS
; -----ベクタ設定-----
VECT     CSEG      ABS = 0FFH
ORG      OFFDOH    ; 割り込み#11 ( 0BH ) にベクタを設定
DSL      ED_INT1
ORG      OFFDCH    ; リセットベクタ設定
DSL      START
DB       00H       ; シングルチップモードに設定
VECT     ENDS
END      START

```

第17章 アドレス一致検出機能

この章では MB90560/565シリーズのアドレス一致検出機能と動作について説明します。

- 17.1 アドレス一致検出機能の概要
- 17.2 アドレス一致検出機能のレジスタ
- 17.3 アドレス一致検出機能の動作説明
- 17.4 アドレス一致検出機能の使用例

17.1 アドレス一致検出機能の概要

プログラムアドレスが、アドレス一致検出レジスタに設定された値と一致した場合に、CPU に読み込まれる命令コードを INT9命令のコードに置き換えます。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ブロックダイアグラム

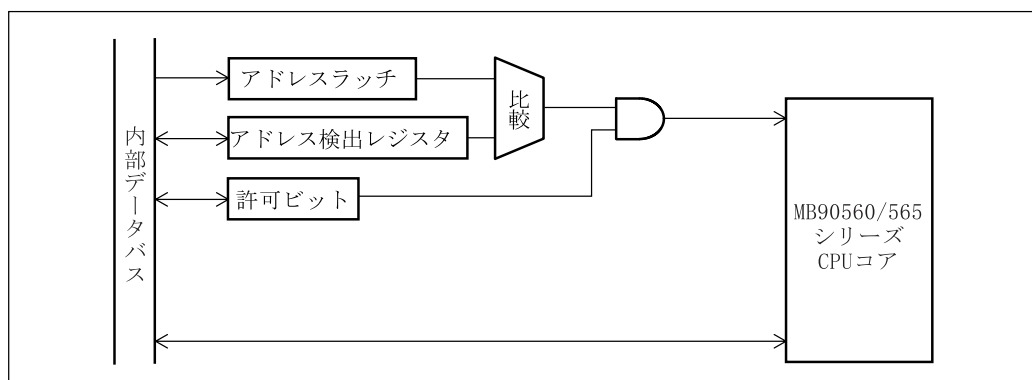


図 17.1-1 アドレス一致検出機能のブロックダイアグラム

17.2 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のレジスタ一覧を示します。

アドレス一致検出機能のレジスタ一覧

アドレス	bit23	bit8	bit7	bit0
ch0:001FF2 _H ch0:001FF1 _H ch0:001FF0 _H	PADR0 (プログラムアドレス検出レジスタ 上位/中位/下位)			
ch1:001FF5 _H ch1:001FF4 _H ch1:001FF3 _H	PADR1 (プログラムアドレス検出レジスタ 上位/中位/下位)			
00009E _H	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)			

図 17.2-1 アドレス一致検出機能のレジスタ一覧

17.2.1 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 (PADR0/PADR1) は、比較対象となるアドレスを設定するレジスタです。

プログラムアドレス検出レジスタ (PADR0/PADR1)

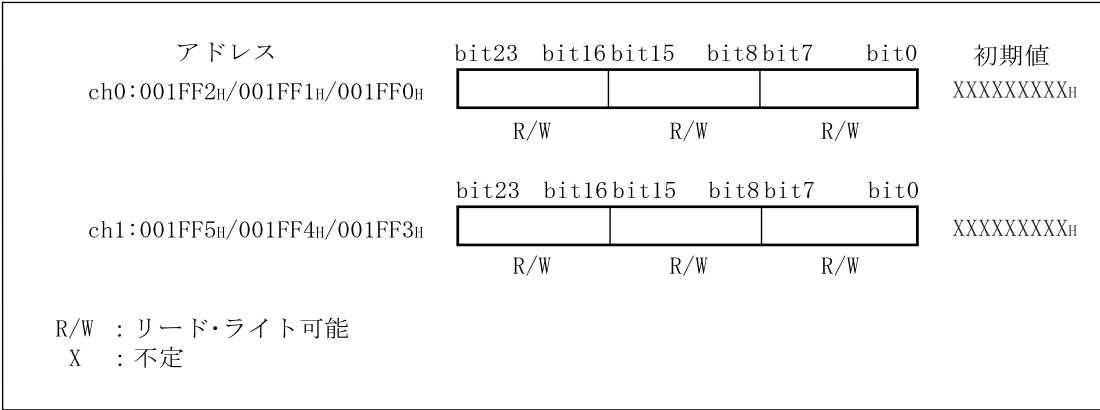


図 17.2-2 プログラムアドレス検出レジスタ (PADR0/PADR1)

プログラムアドレス検出コントロールステータスレジスタ (PACSR) の対応する割込み許可ビットに"1"が設定されている場合は、プログラムアドレス検出レジスタ (PADR0/PADR1) に設定された値とプログラムアドレスを比較します。プログラムアドレス検出レジスタ (PADR0/PADR1) 値とプログラムアドレスが一致した場合は、INT9命令を出力します。割込み許可ビットに"0"が設定されている場合は、INT9命令を出力しません。

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) との対応は以下の通りとなります。

アドレス検出レジスタ	割込み許可ビット
PADR0	AD0E
PADR1	AD1E

17.2.2 **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（PACSR）は、アドレス一致検出機能の割込み制御を行うレジスタです。

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）

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00009Eh	予約	予約	予約	予約	AD1E	予約	AD0E	予約	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード・ライト可能
X : 不定

図 17.2-3 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）

表 17.2-1 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)の各ビットの機能説明

ビット名		機 能
bit7 bit6 bit5 bit4	予約: 予約ビット	・必ず"0"を設定してください。
bit3	AD1E: PADR1の割込み 要求許可ビット	・ PADR1の割込みを許可するビットです。 ・ "1"を設定した場合は、プログラムアドレス検出レジスタ（PADR1）とプログラムアドレスの比較を行い、プログラムアドレス検出レジスタ（PADR1）とプログラムアドレスが一致すると、INT9命令を出力します。
bit2	予約: 予約ビット	・必ず"0"を設定してください。
bit1	AD0E: PADR0の割込み 要求許可ビット	・ PADR0の割込みを許可するビットです。 ・ "1"を設定した場合は、プログラムアドレス検出レジスタ（PADR0）とプログラムアドレスの比較を行い、プログラムアドレス検出レジスタ（PADR0）とプログラムアドレスが一致すると、PADR0の割込みフラグビット（AD0D）に"1"がセットされ、INT9命令を出力します。
bit0	予約: 予約ビット	・必ず"0"を設定してください。

17.3 アドレス一致検出機能の動作説明

アドレス一致検出機能の動作について説明します。

アドレス一致検出機能の動作

プログラムアドレスが、アドレス検出レジスタに設定された値と一致した場合、CPUに読み込まれる命令コードをINT9命令のコード(01_H)に置き換えますので、CPUが設定されたプログラムアドレスの命令を実行すると、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。

プログラムアドレス検出レジスタ(PADRO/PADR1)は2本あり、各レジスタごとに割込み許可ビット(AD1E,AD0E)があります。割込み許可ビット(AD1E,AD0E)に"1"が設定されている場合は、アドレス検出レジスタに設定された値と、プログラムアドレスを比較し、一致した場合は、CPUに読み込まれる命令コードをINT9命令のコードに置き換えます。

< 注意事項 >

アドレス検出レジスタに、命令の1バイト目以降のプログラムアドレスを設定した場合は、アドレス一致検出機能が正しく動作しません。また、アドレス検出レジスタの変更は、割込み許可ビットに"0"を設定してから行ってください。割込み許可ビットに"1"が設定されている場合にアドレス検出レジスタの設定を行うと、設定中に誤ってアドレス検出を行ってしまうことがあります。

17.4 アドレス一致検出機能の使用例

アドレス一致検出機能の使用例を示します。

システム構成図

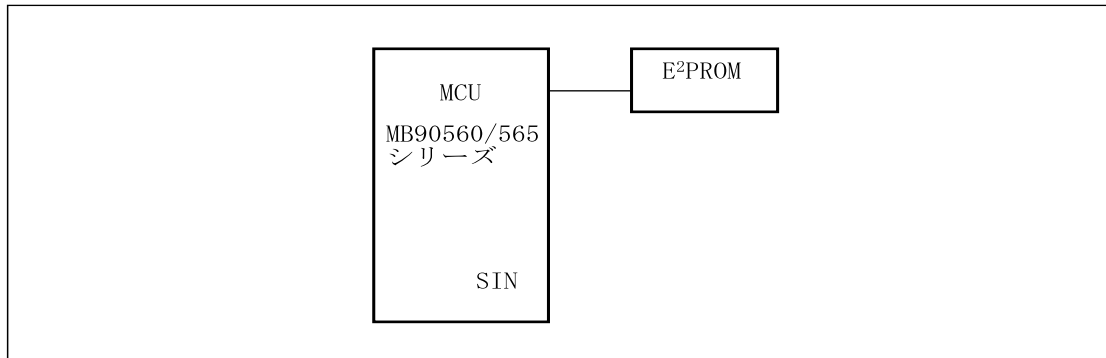


図 17.4-1 システム構成例

E²PROMメモリマップ

表 17.4-1 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の場合プログラムミスなし)
0001 _H	プログラムアドレスNo.0 bit7～bit0
0002 _H	プログラムアドレスNo.0 bit15～bit8
0003 _H	プログラムアドレスNo.0 bit24～bit16
0004 _H	パッチプログラムNo.1バイト数 (0の場合プログラムミスなし)
0005 _H	プログラムアドレスNo.1 bit7～bit0
0006 _H	プログラムアドレスNo.1 bit15～bit8
0007 _H	プログラムアドレスNo.1 bit24～bit16
～0010 _{HH} 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の全ビットは"0"です。

INT9割込み

割込みルーチンでは、どのアドレス検出要因で割込み要求出力があったかを、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。

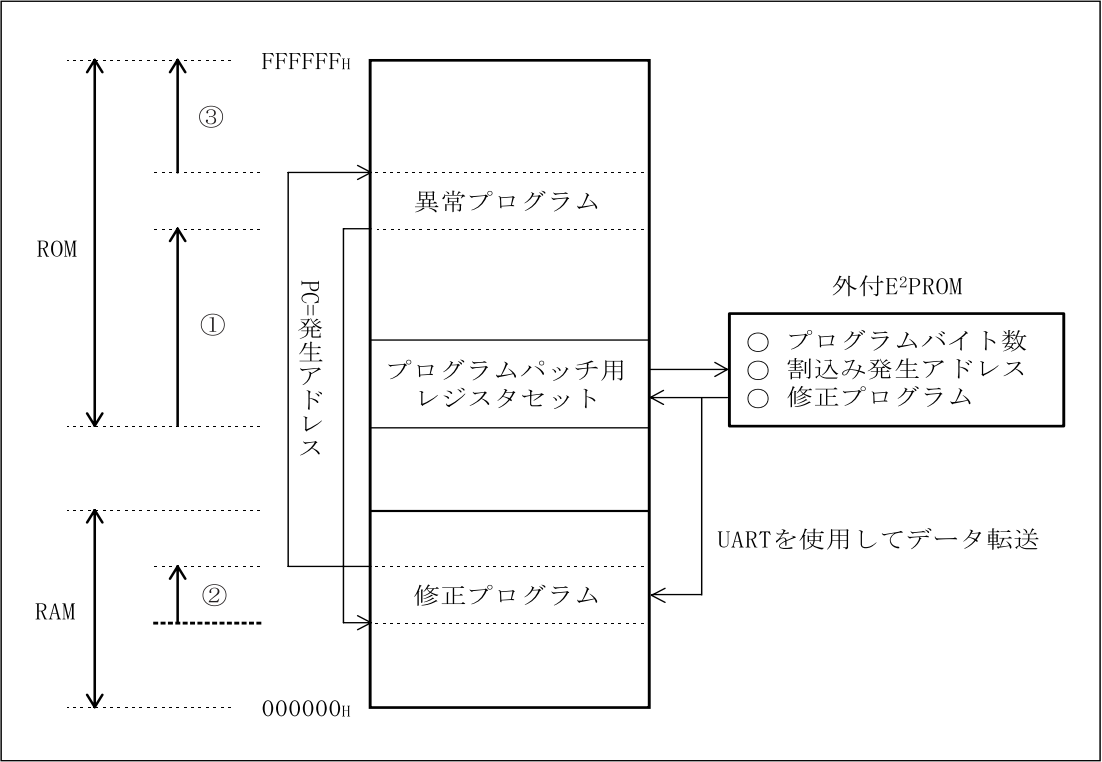


図 17.4-2 プログラムパッチ処理例

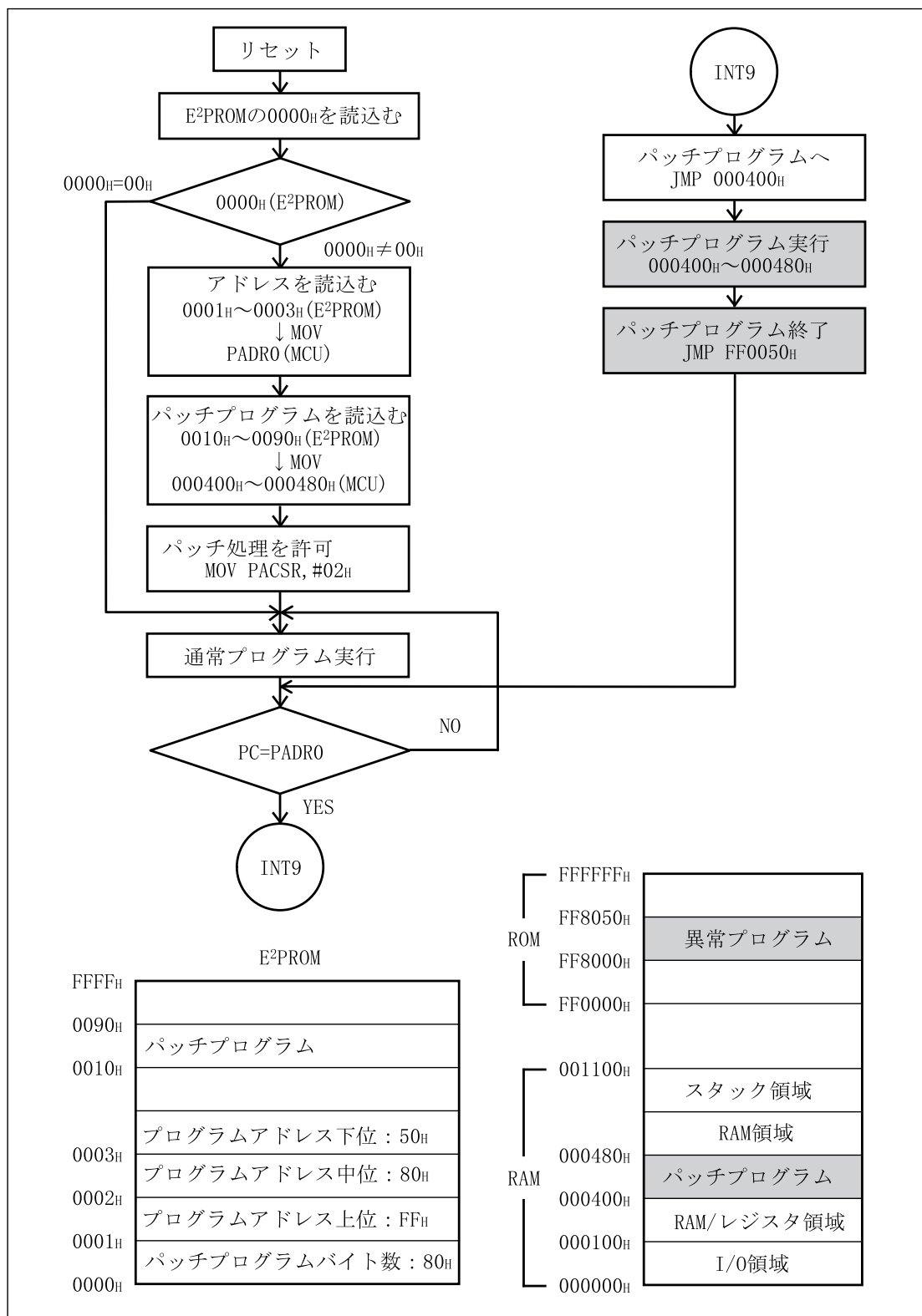


図 17.4-3 プログラムパッチ処理フロー

第18章 ROMミラー機能選択モジュール

この章では MB90560/565シリーズの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。

18.1 ROMミラー機能選択モジュールの概要

18.2 ROMミラー機能選択レジスタ (ROMM)

18.1 ROMミラー機能選択モジュールの概要

ROMミラー機能選択モジュールは、FFバンクのROMデータをROM機能選択モジュールのレジスタを設定することにより00バンクから参照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

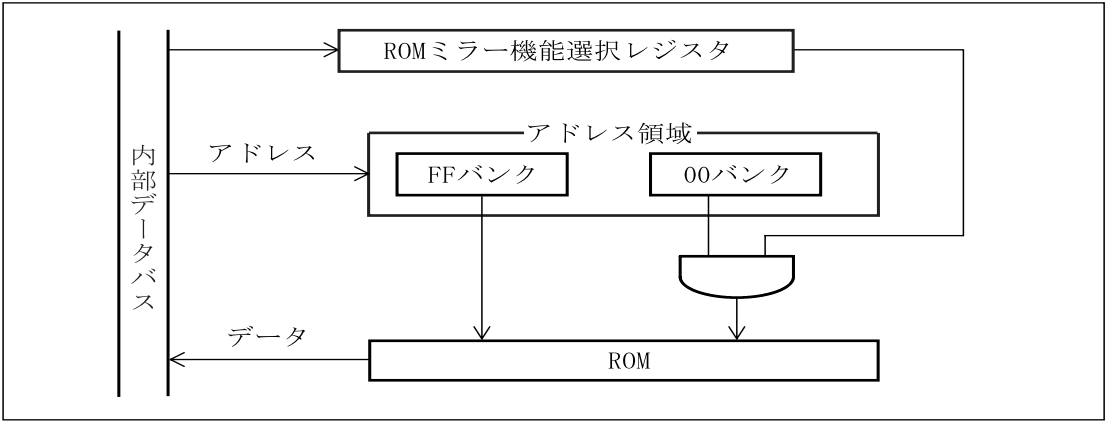


図 18.1-1 ROMミラー機能選択モジュールのブロックダイアグラム

18.2 ROMミラー機能選択レジスタ (ROMM)

ROMミラー機能選択モジュールのレジスタについて説明します。

ROMミラー機能選択レジスタ (ROMM)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00006F _H	-	-	-	-	-	-	-	MI	XXXXXX1 _B
	-	-	-	-	-	-	-	W	

W : ライトオンリー
X : 不定
- : 未定義ビット

18.2-1 ROMミラー機能選択レジスタ (ROMM)

< 注意事項 >

ROMミラー機能選択レジスタはアドレス "004000_H ~ 00FFFF_H" 番地をアクセスしている場合に，設定しないでください。

表 18.2-1 ROMミラー機能選択レジスタ (ROMM) の機能説明

ビット名		機 能
bit15 bit9	- : 未定義ビット	・ 読み出しを行った場合は，不定値となります。
bit8	MI : ROMミラー 機能設定 ビット	ROMミラー機能を設定するビットです。 "1"を設定した場合は，FFバンクのROMのデータを00バンクから読出すことができます。 "0"を設定した場合は，FFバンクのROMのデータを00バンクから読出すことはできません。

< 注意事項 >

00バンクは"004000H ~ 00FFFFH"番地から，"FF4000H ~ FFFFFH"番地を参照しますので，"FF0000H ~ FF3FFFH"番地はROMのミラー機能を設定しても参照できません。

	MB90561	MB90562	MB90F562	MB90567	MB90568	MB90F568	MB90V560	MB90V565
アドレス1	FF8000H	FF0000H	FF0000H	FE8000H	FE0000H	FE0000H	FF0000H	FF0000H
アドレス2	000500H	000900H	000900H	001100H	001100H	001100H	001100H	001100H

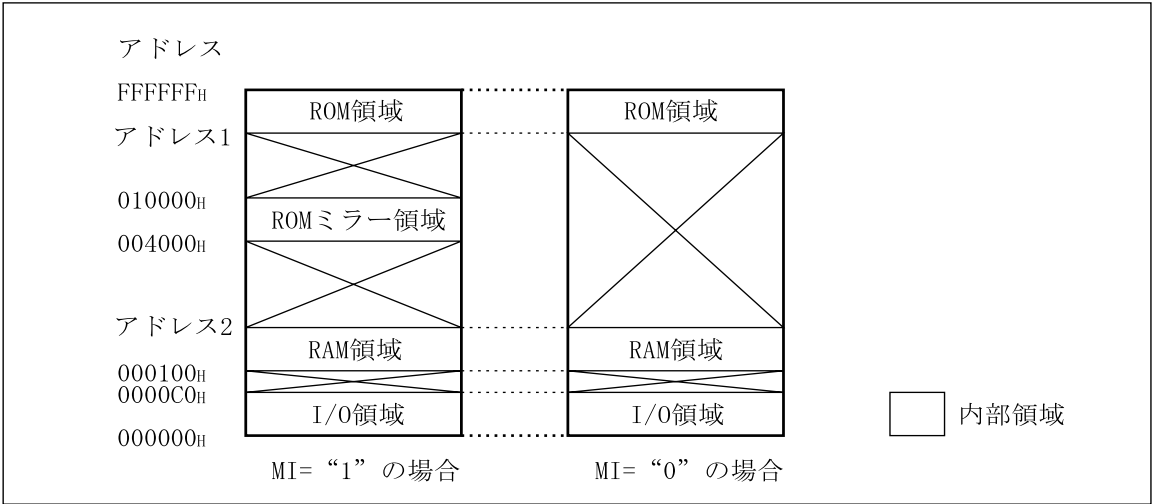


図 18.2-2 メモリ空間

第19章 512Kビット(64Kバイト)フラッシュメモリ

この章では、MB90560/565シリーズの512Kビット(64Kバイト)フラッシュメモリの機能と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、および消去の方法には、下記の3通りの方法があります。

- ・ パラレルライタ
- ・ シリアル専用ライタ
- ・ プログラム実行による書込み/消去

この章では、"プログラム実行による書込み、および消去"について解説します。

- 19.1 512Kビット(64Kバイト)フラッシュメモリの概要
- 19.2 512Kビット(64Kバイト)フラッシュメモリセクタ構成
- 19.3 フラッシュメモリコントロールステータスレジスタ (FMCS)
- 19.4 フラッシュメモリ自動アルゴリズム起動方法
- 19.5 自動アルゴリズム実行状態の確認
- 19.6 フラッシュメモリ書込み/消去の詳細説明
- 19.7 512Kビット(64Kバイト)フラッシュメモリのコマンドプログラム例

19.1 512Kビット(64Kバイト)フラッシュメモリの概要

512Kビット(64Kバイト)フラッシュメモリは、CPUメモリマップの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からの命令を使用して行えますので、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、およびデータの変更が効率よく行えます。ただし、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。

512Kビット(64Kバイト)フラッシュメモリの特長

- 64kワード×8ビット/32kワード×16ビット(16k+8k+8k+32k)セクタ構成
- 自動プログラムアルゴリズム (Embedded Alogrithm:MBM29F400TA互換)
- 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 1万回保証

Embedded AlgorithmはAdvanced Micro Device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、書込みと読出しを同時に行うことはできません。フラッシュメモリにデータ書込み/消去動作を行う場合は、フラッシュメモリ上にあるプログラムを、いったんRAMにコピーし、RAMにコピーされたプログラムを実行することにより、フラッシュメモリへの、書込み動作を行うことが可能となります。詳細は、「19.6.2 データ書込みを行う」を参照してください。

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)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000AE _H	INTE	RDYINT	WE	RDY	予約	LPM1	予約	LPM0	00000000 _B
	R/W	R/W	R/W	R	W	W	W	R/W	

R/W : リード・ライト可能
R : リードオンリー
W : ライトオンリー

19.2 512Kビット(64Kバイト)フラッシュメモリセクタ構成

図 19.2-1に、512Kビット(64Kバイト)フラッシュメモリのセクタ構成を示します。
図中のアドレスには、各セクタの上位アドレスと下位アドレスを示します。

セクタ構成

CPUからアクセスする場合、FFバンクレジスタにSA0～SA3が配置されています。

フラッシュメモリ		CPUアドレス	ライターアドレス*
SA3 (16Kバイト)	上位	FFFFFF _H	7FFFF _H
	下位	FFC000 _H	7C000 _H
SA2 (8Kバイト)	上位	FFBFFF _H	7BFFF _H
	下位	FFA000 _H	7A000 _H
SA1 (8Kバイト)	上位	FF9FFF _H	79FFF _H
	下位	FF8000 _H	78000 _H
SA0 (32Kバイト)	上位	FF7FFF _H	77FFF _H
	下位	FF0000 _H	70000 _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き込みを行う際、CPUアドレスに相当するアドレスです。
汎用ライターを使用し書き込み/消去を行う際は、ライターアドレスで書き込み/消去を実行します。

図 19.2-1 512Kビット(64Kバイト)フラッシュメモリのセクタ構成

19.3 フラッシュメモリコントロールステータスレジスタ (FMCS)

フラッシュメモリコントロールステータスレジスタ (FMCS) では、フラッシュメモリの書込み/消去の制御を行います。

フラッシュメモリコントロールステータスレジスタ (FMCS)

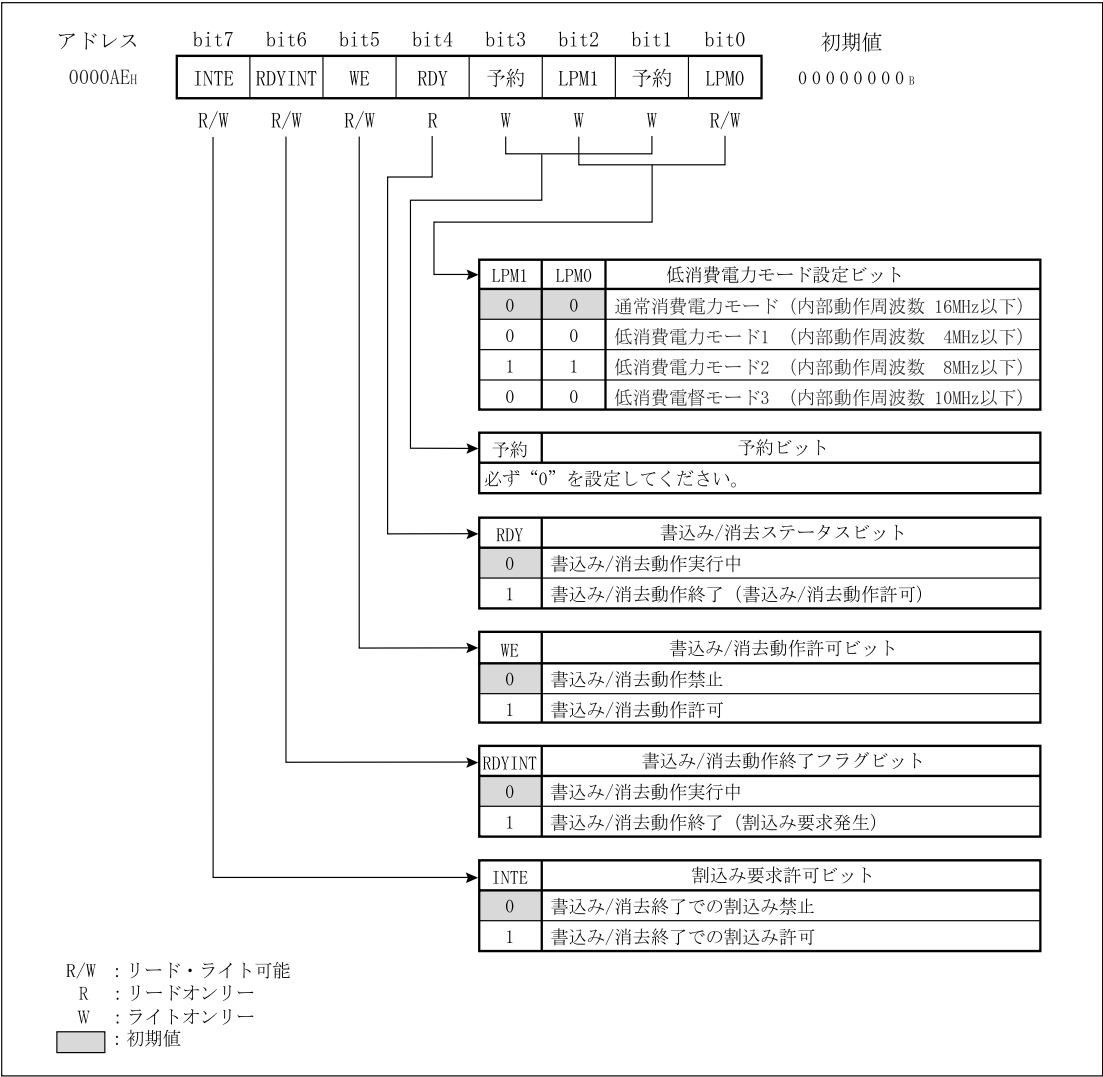


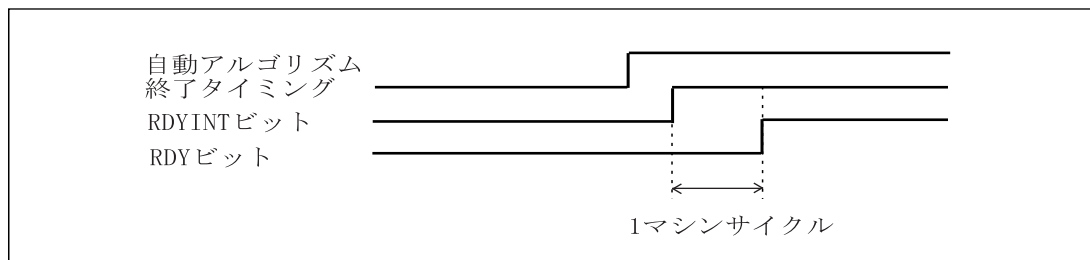
図 19.3-1 フラッシュメモリコントロールステータスレジスタ (FMCS)

表 19.3-1 フラッシュメモリコントロールステータスレジスタ (FMCS) の各ビットの機能説明

ビット名		機 能
bit7	INTE : 割込み要求許可 ビット	<ul style="list-style-type: none"> ・フラッシュメモリの書込み/消去動作の終了により、CPUへの割込み要求出力を許可するビットです。 ・"1"を設定した場合に、RDYINTビットに"1"がセットされると、割込み要求が出力されます。 ・"0"を設定した場合に、RDYINTビットに"1"がセットされても、割込み要求は出力されません。
bit6	RDYINT : 書込み/消去動作 終了フラグ ビット	<ul style="list-style-type: none"> ・フラッシュメモリの書込み/消去動作が終了すると、"1"がセットされ、フラッシュメモリへの書込み/消去動作が可能になります。 ・"0"を設定した場合は、"0"にクリアされ、フラッシュメモリへの書込み/消去動作はできません。 ・"1"を設定した場合は、動作に影響しません。 ・フラッシュメモリ自動アルゴリズム（詳細は、「19.4 フラッシュメモリ自動アルゴリズム起動方法」を参照してください。）終了タイミングでも、"1"がセットされます。 ・リードモディファイライト（RMW）命令使用時は、必ず"1"が読出されます。
bit5	WE : 書込み/消去動作 許可ビット	<ul style="list-style-type: none"> ・"1"を設定した場合は、FFバンクへの書込み/消去コマンドシーケンス（詳細は、「19.4 フラッシュメモリ自動アルゴリズム起動方法」を参照してください。）を実行したあと、フラッシュメモリへの書込み/消去ができます。 ・"0"を設定した場合は、FFバンクへの書込み/消去コマンドシーケンスを実行しても、書込み/消去の信号は発生されません。 ・初期値は"0"が設定され、動作が禁止されていますので、フラッシュメモリの書込み/消去のコマンドを起動する前に必ず"1"を設定して、動作を許可してください。 <p><注記></p> <ul style="list-style-type: none"> ・書込み/消去動作を行わない場合は、"0"を設定してください。
bit4	RDY : 書込み/消去 ステータス ビット	<ul style="list-style-type: none"> ・"0"にクリアされている間は、フラッシュメモリへの書込み/消去動作はできません。 ・"0"にクリアされている場合でも、読出し/リセットコマンド、セクタ消去一時停止などのサスペンドコマンドは受け付けられます。書込み/消去動作を終了すると、"1"がセットされます。
bit3,1	予約 : 予約ビット	<ul style="list-style-type: none"> ・必ず"1"を設定してください。
bit2,0	LPM1,LPM0 : 低消費電力 モード設定 ビット	<ul style="list-style-type: none"> ・CPUからフラッシュメモリへのアクセスタイムを設定します。 ・内部動作周波数により設定できる値が変わります。 ・内部動作周波数が低いほどフラッシュメモリ本体の消費電力を少なくすることができます。

< 注意事項 >

動作終了フラグビット (RDYINT) と書込み/消去ステータスビット (RDY) は同時に変化しませんので、動作終了フラグビット (RDYINT) か書込み/消去ステータスビット (RDY) のいずれかで書込み/消去終了判定をするようプログラムを作成してください。



19.4 フラッシュメモリ自動アルゴリズム起動方法

フラッシュメモリの自動アルゴリズムを起動するコマンドには、読出し/リセット、書込み、チップ消去、およびセクタ消去の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

コマンドレジスタに書込むデータは、すべてバイト長ですが、ワードアクセスで設定してください。

表 19.4-1 コマンドシーケンス表

コマンド シーケンス	バス ライト アクセス	1st/バスライト サイクル		2nd/バスライト サイクル		3rd/バスライト サイクル		4th/バスライト サイクル		5th/バスライト サイクル		6th/バスライト サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し/ リセット	1	FFXXXH	XF0H	-	-	-	-	-	-	-	-	-	-
読出し/ リセット	4	FFAAAH	XXAH	FF554H	XX5H	FFAAAH	XF0H	RA	RD	-	-	-	-
書込みプ ログラム	4	FFAAAH	XXAH	FF554H	XX5H	FFAAAH	XX4H	PA (even)	PD (word)	-	-	-	-
チップ 消去	6	FFAAAH	XXAH	FF554H	XX5H	FFAAAH	XX8H	FFAAAH	XXAH	FF554H	XX5H	FFAAAH	XX10H
セクタ 消去	6	FFAAAH	XXAH	FF554H	XX5H	FFAAAH	XX8H	FFAAAH	XXAH	FF554H	XX5H	SA (even)	XX30H
セクタ消去一時停止		アドレス "FFXXXH" データ(XX8H)の入力で、セクタ消去中の消去一時停止											
セクタ消去再開		アドレス "FFXXXH" データ(XX30H)の入力で、セクタ消去中の消去一時停止後、消去開始											

* : 2種類の読出し/リセットコマンドは、フラッシュメモリを読出しモードにリセットすることができます。

<注記> 表中のアドレスは、CPUメモリマップ上の値です。"X"は任意の値です。

RA: 読出しアドレス

PA: 書込みアドレス、偶数アドレスが指定可能です。

SA: セクタアドレス、詳細は「19.2 512Kビット(64Kバイト)フラッシュメモリセクタ構成」を参照してください。

偶数アドレスが指定可能です。

RD: 読出しデータ

PD: 書込みデータ、ワードデータが指定可能です。

19.5 自動アルゴリズム実行状態の確認

フラッシュメモリは、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が確認できる、ハードウェアシーケンスを内蔵しています。

ハードウェアシーケンスフラグ

ハードウェアシーケンスフラグは、DQ7、DQ6、DQ5、DQ3の4ビットの出力で構成されます。DQ7はデータポーリングフラグ、DQ6はトグルビットフラグ、DQ5はタイミングリミット超過フラグ、DQ3はセクタ消去タイマフラグです。ハードウェアシーケンスにより、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無効かなど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(詳細は、「表 19.4-1 コマンドシーケンス表」を参照)設定後に、フラッシュメモリ内部のハードウェアシーケンスフラグを参照したいセクタのアドレスをリードすることで参照できます。

表 19.5-1 ハードウェアシーケンスフラグのビット割当て

ビットNo.	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
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、終了しているかを判断するには、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)の書込み/消去ステータスビット(RDY)を確認してください。書込み/消去終了後は、読出し/リセット状態に戻ります。書込み/消去プログラムを作成する際には、DQ7、DQ6、DQ5、及びDQ3のハードウェアシーケンスフラグで書込み/消去終了を確認後に、データの読み出しなどの、処理を行ってください。2回目以降のセクタ消去コードライトが有効か無効かについても、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。

表 19.5-2 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

19.5.1 データポーリングフラグ (DQ7)

データポーリングフラグ (DQ7) は、自動アルゴリズム実行が進行中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ことができます。データポーリングフラグの状態遷移を、表 19.5-3と表 19.5-4に示します。

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、データポーリングフラグ (DQ7) は最後に書込まれたデータを反転させた値を出力します。自動書込みアルゴリズムが終了したあとに、リードアクセスを行うと、データポーリングフラグ (DQ7) は、リードアクセスを行ったアドレスの値を出力します。

チップ消去/セクタ消去動作時

セクタ消去アルゴリズムを実行中に、現在消去しているセクタをリードアクセスすると、データポーリングフラグ (DQ7) は"0"を出力します。セクタ消去が終了すると、データポーリングフラグ (DQ7) は"1"を出力します。チップ消去アルゴリズムを実行中にリードアクセスすると、データポーリングフラグ (DQ7) は"0"を出力します。チップ消去が終了すると、データポーリングフラグ (DQ7) 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中に、リードアクセスすると、データポーリングフラグ (DQ7) は、リードアクセスしたアドレス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、読出し値 (DATA:7) を出力します。トグルビットフラグ (DQ6) と同時に参照することで、セクタ消去一時停止状態であるか、また、どこ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) の終了を受けて他のビットの出力が可能となります。自動アルゴリズム終了後のデータの読出しは、データポーリング終了を確認したリードアクセスのあとに行ってください。

表 19.5-3 データポーリングフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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 19.5-4 データポーリングフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ 消去動作
DQ7	$\overline{\text{DQ7}}$	0

表 19.5-3 と 表 19.5-4 は「表 19.5-2 ハードウェアシーケンスフラグ機能一覧」の抜粋です。

19.5.2 トグルビットフラグ (DQ6)

トグルビットフラグ (DQ6) は、自動アルゴリズム実行が進行中、もしくは終了状態であることをトグルビットフラグによって認識することができます。トグルビットフラグの状態遷移を、表 19.5-5と表 19.5-6に示します。

書込み時とチップ消去/セクタ消去時

自動書込みアルゴリズムおよび、チップ消去/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、トグルビットフラグ (DQ6) は、リードを行うごとに、"1"と"0"を交互にトグル出力を行います。自動書込みアルゴリズム、およびチップ消去/セクタ消去アルゴリズム終了したあとに、連続したリードアクセスを行うとトグルビットフラグ (DQ6) は、リードを行うごとに、リードアドレスの読出し値を出力します。

セクタ消去一時停止時

セクタ消去一時停止状態で、リードアクセスを行うと、トグルビットフラグ (DQ6) は、リードアドレスが消去中のセクタならば、"1"を出力します。消去中のセクタでなければ、リードアドレスの読出し値を出力します。

参考：

書込みを行うセクタが、書換え保護(セクタプロテクト)されている場合、約 2 μ sの間、トグルビットフラグ(DQ6)からトグル出力を行ったあと、データを書換えずに、トグル出力を終了します。また、消去を行う全てのセクタが書換え保護(セクタプロテクト)されている場合、トグルビットフラグ(DQ6)は約100 μ sの期間、トグルビットフラグ(DQ6)からトグル出力を行ったあと、データを書換えずに読出し/リセット状態に戻ります。

表 19.5-5 トグルビットフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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 19.5-6 トグルビットフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ 消去動作
DQ6	Toggle	Toggle

表 19.5-5 と 表 19.5-6 は、「表 19.5-2 ハードウェアシーケンスフラグ機能一覧」の抜粋です。

19.5.3 タイミングリミット超過フラグ (DQ5)

タイミングリミット超過フラグ (DQ5) は、自動アルゴリズムの実行がフラッシュメモリ内部に規定された時間 (内部パルス回数) を超えてしまったことを認識することができます。タイミングリミット超過フラグの状態遷移を、表 19.5-7と表 19.5-8に示します。

書込み時とチップ消去/セクタ消去時

書込み、またはチップ消去/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"をタイミングリミット超過フラグ (DQ5) から出力します。タイミングリミット超過フラグ (DQ5) は、自動アルゴリズムが実行中、終了状態にかかわらず、書込み/消去が成功したか失敗したかの判定ができます。タイミングリミット超過フラグが"1"を出力した場合、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実行中であれば、書込みが失敗していると判断できます。

例えば、"0"が書込まれているフラッシュメモリビットに"1"を書込もうとすると、フラッシュメモリはロックされ、自動アルゴリズムは終了せず、データポーリングフラグ (DQ7) は有効なデータを出力しません。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。タイミングリミット超過フラグが"1"を出力した場合は、フラッシュメモリが不良ではなく、正しく使用されなかったということを示していますので、リセットコマンドを実行してください。

表 19.5-7 タイミングリミット超過フラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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 19.5-8 タイミングリミット超過フラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ 消去動作
DQ5	1	1

表 19.5-7 と 表 19.5-8 は「表 19.5-2 ハードウェアシーケンスフラグ機能一覧」の抜粋です。

19.5.4 セクタ消去タイマフラグ (DQ3)

セクタ消去タイマフラグ (DQ3) は、セクタ消去コマンドを起動した後に、セクタ消去ウェイト期間中であるか否かを認識することができます。セクタ消去タイマフラグの状態遷移を、表 19.5-9と表 19.5-10に示します。

セクタ消去動作時

セクタ消去コマンドを起動した後に、リードアクセスすると、セクタ消去タイマフラグ (DQ3) はセクタ消去ウェイト期間中であれば"0"を出力し、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(DQ7="0", DQ6=トグル出力)は、セクタ消去タイマフラグ(DQ3)が"1"を出力していれば、セクタ消去を行っています。続けて、消去一時停止以外のコマンドを設定した場合は、消去が終了されるまで無視されます。

セクタ消去タイマフラグ(DQ3)が"0"を出力している場合、フラッシュメモリは、セクタ消去コードの書込みを受け付けます。セクタ消去コードの書込みは、必ず、セクタ消去タイマフラグ(DQ3)が"0"を出力していることを確認してから書込みを行ってください。セクタ消去タイマフラグ(DQ3)が"1"を出力している場合は、セクタの消去コマンドは受け付けられません。

セクタ消去一時停止時

セクタ消去一時停止中に、リードアクセスすると、セクタ消去タイマフラグ (DQ3) は、リードアドレスが、消去中のセクタに属する場合は"1"を出力します。リードアドレスが、消去中のセクタに属しない場合は、読出し値を出力します。

表 19.5-9 セクタ消去タイマフラグの状態遷移 (正常動作時の状態変化)

動作状態	書込み動作 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 19.5-10 セクタ消去タイマフラグの状態遷移 (異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ 消去動作
DQ3	0	1

表 19.5-9 と 表 19.5-10 は「表 19.5-2 ハードウェアシーケンスフラグ機能一覧」の抜粋です。

19.6 フラッシュメモリ書込み/消去の詳細説明

自動アルゴリズムを起動するコマンドを設定し、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、セクタ消去再開の動作を行う手順を説明します。

フラッシュメモリ書込み/消去の詳細説明

自動アルゴリズムは、読出し/リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、消去再開の動作がコマンドシーケンス(詳細は、「19.4 フラッシュメモリ自動アルゴリズム起動方法」を参照)をCPUからフラッシュメモリへ書込むことにより起動することができます。CPUからフラッシュメモリへの書込みは、必ず続けて行ってください。自動アルゴリズムはデータポーリング機能などで終了しているか、実行中であるかを確認することができます。正常終了したあとは、読出し/リセット状態に戻ります。

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、セクタ消去再開の動作の詳細は、「19.6.1 読出し/リセット状態にする」～「19.6.6 セクタ消去を再開する」に示します。

19.6.1 読出し/リセット状態にする

読出し/リセットコマンドを実行し、フラッシュメモリを読出し/リセット状態にする手順について説明します。

読出し/リセット状態にする

フラッシュメモリを読出し/リセット状態にするには、コマンドシーケンス表（「表 19.4-1 コマンドシーケンス表」を参照）の読出し/リセットコマンドを、CPUからフラッシュメモリへ続けて送ります。

読出し/リセットコマンドには、2通りのコマンドシーケンスがありますが、違いはありません。

読出し/リセット状態は、フラッシュメモリの初期状態ですので、電源投入直後、コマンドが正常終了した場合は、常に読出し/リセット状態になります。読出し/リセット状態は、コマンドの入力待ち状態となります。

読出し/リセット状態では、フラッシュメモリへリードアクセスを行うことにより、データを読み出すことができます。マスクROMと同様に、CPUからのプログラムアクセスが可能です。フラッシュメモリへリードアクセスを行う場合は、読出し/リセットコマンドは必要ありません

コマンドが正常に終了しなかった場合は、自動アルゴリズムを初期化するために、読出し/リセットコマンドを使用してください。

19.6.2 データ書き込みを行う

書き込みコマンドを実行し、フラッシュメモリにデータ書き込みを行う手順について説明します。図 19.6-1にフラッシュメモリ書き込み手順の例を示します。

データ書き込みを行う

フラッシュメモリのデータ書き込み自動アルゴリズムを起動するには、コマンドシーケンス表（「表 19.4-1 コマンドシーケンス表」を参照）の書き込みコマンドを、CPUからフラッシュメモリへ続けて送ります。4サイクル後に目的のアドレスへのデータライトが終了すると、自動アルゴリズムが起動され自動書き込みを開始します。

アドレス指定方法

書き込みデータサイクルで指定する書き込みアドレスは、偶数アドレスが設定可能です。奇数アドレスを設定した場合は、正しく書き込みができません。偶数アドレスへワードデータ単位での書き込みを行ってください。

書き込みは、アドレスの順番に関係なく、また、セクタの境界を越えても可能ですが、1回の書き込みコマンドにて、書込まれるデータは1ワードです。

データ書き込み上の注意

書き込みによって、ビットデータを"0"から"1"に戻すことはできません。ビットデータ"0"にビットデータ"1"を書込んだ場合は、データポーリングアルゴリズム、トグル動作が終了せず、フラッシュメモリ素子が不良と判定され、書き込み規定時間を超えてタイミングリミット超過フラグ（DQ5）に"1"が出力されるか、見かけ上ビットデータに"1"が書込まれたように見えるかのどちらかとなります。読出し/リセット状態でデータを読出すとビットデータは"0"です。ビットデータを"0"から"1"に戻すには消去動作を行います。

自動書き込み実行中は、全てのコマンドが無視されます。書き込み中にハードウェアリセットが起動されると、書き込みアドレスのデータは、保証されません。

フラッシュメモリ書き込み手順

ハードウェアシーケンスフラグ（詳細は、「19.5 自動アルゴリズム実行状態の確認」を参照）を用いることでフラッシュメモリ内部の自動アルゴリズムの状態を判定することが可能です。フラッシュメモリ書き込みでは、書き込み終了の確認にデータポーリングフラグ（DQ7）を用いてください。

フラグチェックのために、読込みデータは最後に書き込みを行ったアドレスからとなります。

データポーリングフラグ（DQ7）は、タイミングリミット超過フラグ（DQ5）と同時に変化するので、タイミングリミット超過フラグ（DQ5）が"1"であってもデータポーリングフラグ（DQ7）を再チェックしてください。

トグルビットフラグ（DQ6）も、タイミングリミット超過フラグ（DQ5）が"1"に変化するのと同時にトグル動作を止めますので、トグルビットフラグ（DQ6）を再チェックしてください。

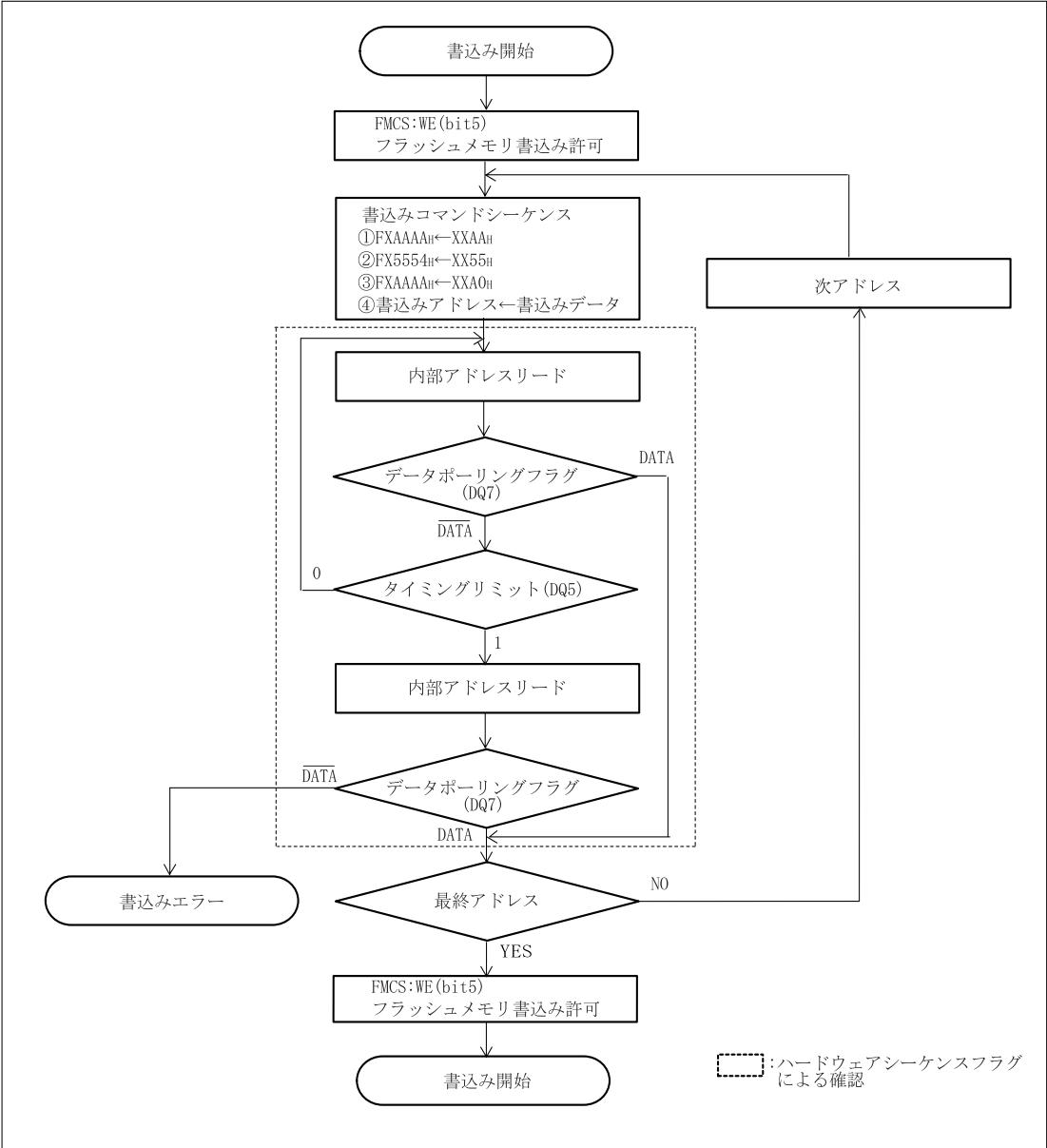


図 19.6-1 フラッシュメモリ書き込み手順の例

19.6.3 データ消去を行う（チップ消去）

チップ消去コマンドを実行し、フラッシュメモリの全データ消去を行う手順について説明します。

データ消去を行う（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（表 19.4-1 コマンドシーケンス表）を参照）のチップ消去コマンドを、CPUからフラッシュメモリへ送ります。

チップ消去コマンドは、6サイクル目の書き込みが完了した時点でチップ消去動作を開始します。チップ消去では、消去を行う前に、フラッシュメモリへ書き込みを行う必要はありません。自動消去機能処理中に、フラッシュメモリは全てのビットデータを消去する前に、“0” を書込んで検証を行います。

19.6.4 データ消去を行う（セクタ消去）

セクタ消去コマンドを実行し、フラッシュメモリの任意のセクタ消去を行う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。

データ消去を行う（セクタ消去）

フラッシュメモリの任意のセクタを消去するためには、コマンドシーケンス表（「表 19.4-1 コマンドシーケンス表」を参照）のセクタ消去コマンドを、CPUからフラッシュメモリへ送ります。

セクタ指定方法

セクタ消去コマンドは、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アドレスへ、セクタ消去コード（30_H）を書込むことにより、50 μ sのセクタ消去ウェイトを開始します。複数のセクタ消去を行う場合は、引き続き消去する目的のセクタアドレスに消去コード（30_H）を書込みます。

複数のセクタを指定する場合の注意

最後のセクタ消去コードの書込みから50 μ sのセクタ消去ウェイト期間終了後に、消去を開始します。複数のセクタを同時に消去する場合は、消去セクタのアドレスと消去コード（コマンドシーケンス6サイクル目）を、50 μ s以内に入力してください。50 μ s以上では受け付けられません。連続したセクタ消去コードの書込みが有効か有効でないかは、セクタ消去タイマフラグ（DQ3）によって調べることができます。セクタ消去タイマフラグ（DQ3）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設定してください。

セクタ消去手順

ハードウェアシーケンスフラグ（詳細は、「19.5 自動アルゴリズム実行状態の確認」を参照）を用いることでフラッシュメモリ内部の自動アルゴリズムの状態を判定することができます。セクタ消去では、消去終了の確認にトグルビットフラグ（DQ6）を用いてください。フラグチェックのために読込むデータは、消去しようとしているセクタからの読込みとなります。

トグルビットフラグ（DQ6）は、タイミングリミット超過フラグ（DQ5）が"1"に変化すると同時にトグル出力を止めるので、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）は再チェックしてください。

データポーリングフラグ（DQ7）も、タイミングリミット超過フラグ（DQ5）と同時に変化するので、データポーリングフラグ（DQ7）を再チェックしてください。

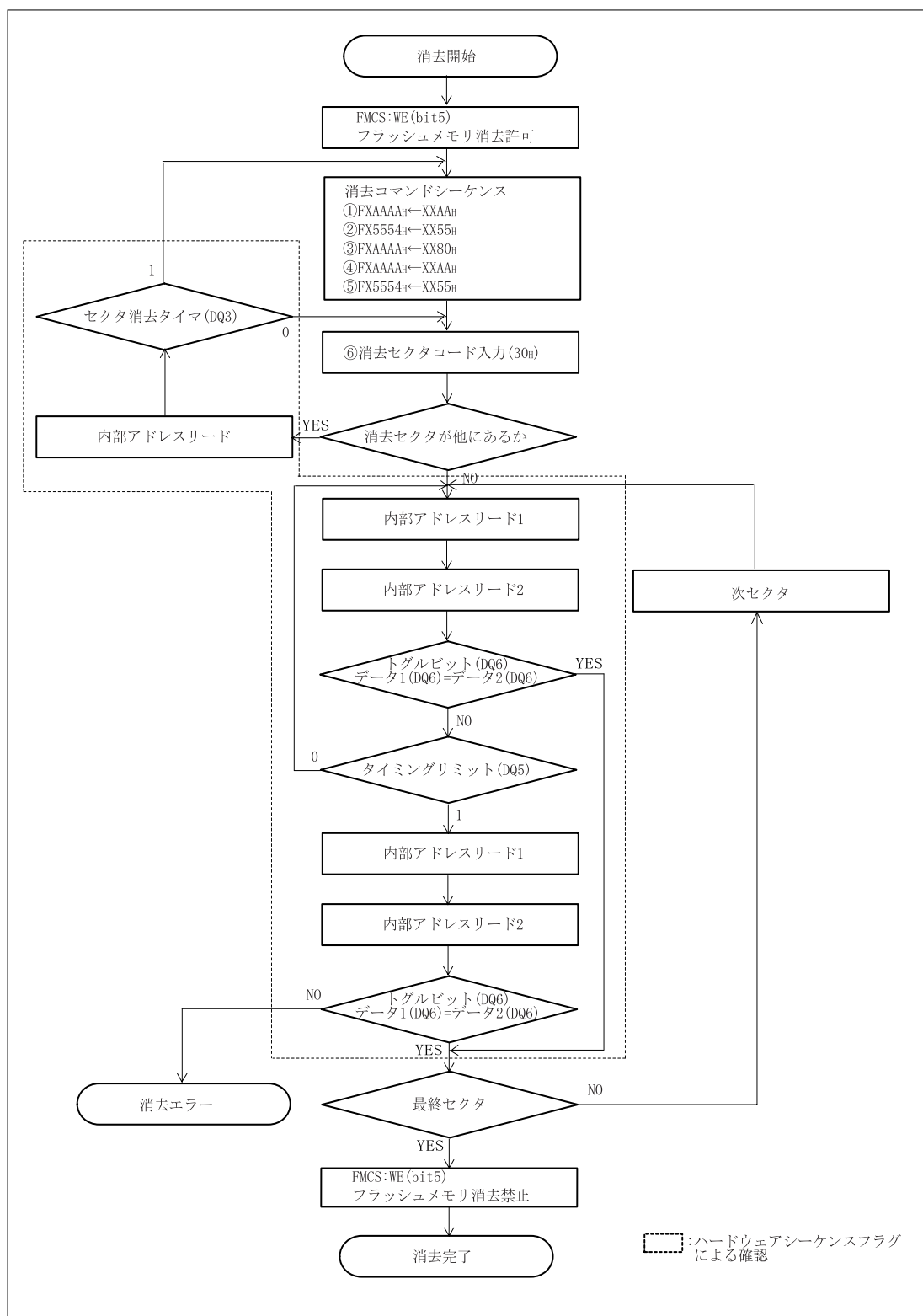


図 19.6-2 セクタ消去手順の例

19.6.5 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設定し、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、データを読出すことが可能です。

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(「表 19.4-1 コマンドシーケンス表」を参照)のセクタ消去一時停止コマンドを、CPUからフラッシュメモリに送ります。

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、データを読出すことができます。セクタ消去一時停止状態では読出しが可能で書込みはできません。セクタ消去一時停止コマンドは、消去ウェイト時間を含むセクタ消去中のみ有効で、チップ消去中や書込み動作中は無視されます。

セクタ消去一時停止コマンドは、消去一時停止コード(B0_H)を書込むことで実行されます。アドレスはフラッシュメモリ内の任意のアドレスを設定します。消去一時停止中は消去一時停止コマンドを実行しても無視されます。

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、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15μsの時間の後、消去一時停止状態に入ります。

19.6.6 セクタ消去を再開する

セクタ消去再開コマンドを設定し、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。

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(「表 19.4-1 コマンドシーケンス表」を参照)のセクタ消去再開コマンドを、CPUからフラッシュメモリへ送ります。

セクタ消去再開コマンドは、セクタ消去一時停止コマンドによる、セクタ消去一時停止状態から、セクタ消去を再開するためのコマンドです。セクタ消去再開コマンドは、消去再開コード(30_H)の書き込みを行うことで実施されますが、アドレスはフラッシュメモリ内の任意のアドレスを設定します。

セクタ消去中のセクタ消去再開コマンドの実行は無視されます。

19.7 512Kビット(64Kバイト)フラッシュメモリのコマンドプログラム例

512Kビット(64Kバイト)フラッシュメモリのプログラム例を示します。

512Kビット(64Kバイト)フラッシュメモリのプログラム例

```

NAME                               FLASHWE
TITLE                              FLASHWE
;-----
;512K-FLASH サンプルプログラム
;
;1: FLASHにあるプログラム(アドレス FFBC00H セクタ SA2)をRAM(アドレス 000700H)に
   転送する。
;2: RAM上でプログラムを実行する。
;3: PDR1の値をFLASH(アドレス FF0000H セクタ SA0)に書込む。
;4: 書込んだ値(アドレス FF0000H セクタ SA0)を読み出しPDR2に出力する。
;5: 書込んだセクタ(SA0)を消去する。
;6: 消去データ確認の出力
;   条件
;       ・ RAM転送バイト数: 100H(256バイト)
;       ・ 書込み, 消去の終了判定
;           DQ5(タイミングリミット超過フラグ)での判定
;           DQ6(トグルビットフラグ)での判定
;           RDY(FMCS)での判定
;       ・ エラー時の処理
;           P00 ~ P07に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
```

19.7 512Kビット(64Kバイト)フラッシュメモリのコマンドプログラム例

```

COMADR2 RW      1
        ORG      0AAAAH
COMADR1 RW      1
DATA    ENDS
;////////////////////////////////////
;メインプログラム(SA1)
;////////////////////////////////////
CODE    CSEG
START:
;////////////////////////////////////
;初期化
;////////////////////////////////////
MOV      CKSCR,#0BAH      ; 3週に設定
MOV      RP,#0
MOV      A,#!STA_T
MOV      SSB,A
MOVW     A,#STA_T
MOVW     SP,A
MOV      ROMM,#00H      ; ミラーオフ
MOV      PDRO,#00H      ; エラー確認用
MOV      DDRO,#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(1500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(SA2)
;////////////////////////////////////
RAMPRG  CSEG      ABS=0FFH
        ORG      0BC00H
;
; 初期化
;
MOVW     RW0,#0500H      ; RW0: 入力データ確保用RAM空間 00:0500 ~
MOVW     RW2,#0000H      ; RW2: フラッシュメモリ書き込みアドレス
                        ; FF:0000H ~
MOV      A,#00H      ; DTB変更
MOV      DTB,A      ; @RW0用ハック指定

```

```

MOV      A, #0FFH          ; ADB変更1
MOV      ADB, A            ; 書込みモード指定アドレス用バンク指定
MOV      PDR3, #00H        ; スイッチ初期化
MOV      DDR3, #00H
;
WAIT1    BBC      PDR3:0, WAIT1      ; PDR3:0 Hiで書込みスタート
;
;////////////////////////////////////
;書込み(SA0)
;////////////////////////////////////
MOV      A, PDR1
MOVW     @RW0+00, A          ; RAMにPDR1データを確保
MOV      FMCS, #20H         ; 書込みモード設定
MOVW     ADB:COMADR1, #00AAH   ; フラッシュ書込みコマンド 1
MOVW     ADB:COMADR2, #0055H   ; フラッシュ書込みコマンド 2
MOVW     ADB:COMADR1, #00A0H   ; フラッシュ書込みコマンド 3
;
MOVW     A, @RW0+00          ; 入力データ(RW0)をフラッシュメモリ(RW2)
                             ; に書込む
MOVW     @RW2+00, A
WRITE    ;待ち時間チェック
;////////////////////////////////////
;
; タイムリミット超過チェック - フラグが立ちトグル動作中である場合ERROR
;////////////////////////////////////
MOVW     A, @RW2+00
AND      A, #20H             ; DQ5タイムリミットチェック
BZ       NTOW                 ; タイムリミットオーバー
MOVW     A, @RW2+00           ; AH
MOVW     A, @RW2+00           ; AL
XORW     A                   ; AH ALのXOR(値が違えば1)
AND      A, #40H              ; DQ6トグルビットは違っているか
BNZ      ERROR                ; 違えばERRORへ
;
;////////////////////////////////////
;
; 書込み終了チェック(FMCS-RDY)
;////////////////////////////////////
NTOW     MOVW     A, FMCS
AND      A, #10H              ; FMCS RDYビット(4bit)抽出
BZ       WRITE                ; 書込み終了か?
MOV      FMCS, #00H           ; 書込みモード解除
;////////////////////////////////////
;書込みデータ出力
;////////////////////////////////////
MOVW     RW2, #0000H          ; 書込みデータ出力
MOVW     A, @RW2+00
MOV      PDR2, A
;
WAIT2    BBC      PDR3:1, WAIT2      ; PDR3:1 Hiでセクタ消去スタート
;
;////////////////////////////////////
;セクタ消去(SA0)
;////////////////////////////////////
MOVW     @RW2+00, #0000H      ; アドレス初期化
MOV      FMCS, #20H           ; 消去モード設定
MOVW     ADB:COMADR1, #00AAH   ; フラッシュ消去コマンド 1
MOVW     ADB:COMADR2, #0055H   ; フラッシュ消去コマンド 2
MOVW     ADB:COMADR1, #0080H   ; フラッシュ消去コマンド 3
MOVW     ADB:COMADR1, #00AAH   ; フラッシュ消去コマンド 4

```

```

MOVW      ADB:COMADR2,#0055H    ; フラッシュ消去コマンド 5
MOVW      @RW2+00,#0030H        ; 消すセクタに消去コマンド発行 6
ELS       ;待ち時間チェック
;      ;////////////////////////
;      ; タイムリミット超過チェック - フラグが立ちトグル動作中である場合ERROR
;      ;////////////////////////
MOVW      A,@RW2+00
AND        A,#20H                ; DQ5タイムリミットチェック
BZ         NTOE                  ; タイムリミットオーバー
MOVW      A,@RW2+00              ; AH 書き込み動作中は, DQ6から
MOVW      A,@RW2+00              ; AL リードごと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書き込み動作
                                ; 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        ADB:COMADR1,#0F0H    ; リセットコマンド(読出しが可能になる)
MOV        FMCS,#00H             ; FLASHモード解除
MOV        PDRO,#0FFH           ; エラー処理の確認
RETP       ; メインプログラムに戻る
RAMPRG     ENDS
;////////////////////////
VECT       CSEG      ABS=0FFH
ORG        OFFDCH
DSL        START
DB         00H
VECT       ENDS
;
END        START

```


第20章 F²MC-16LX MB90F562/F568シリアル書込み接続例

この章では，株式会社製ワイ・ディ・シー製フラッシュマイコンプログラマ（AF220/AF210/AF120/AF110）を用いた場合の，シリアル書込み接続例について説明します。

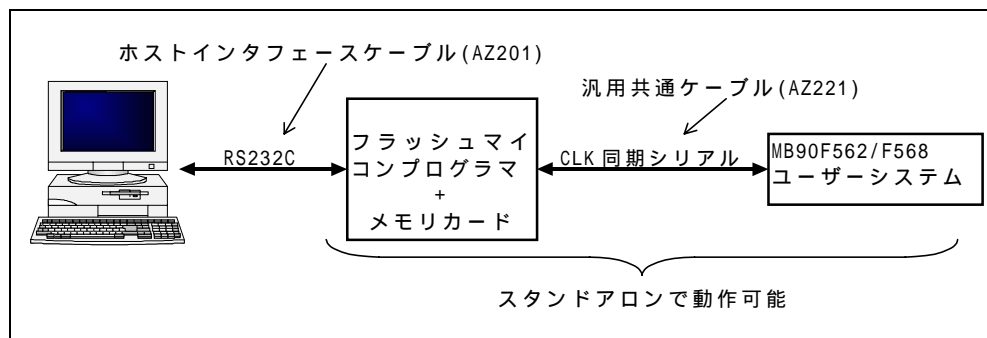
- 20.1 富士通標準シリアルオンボード書込みの標準構成
- 20.2 シリアル書込み接続例(ユーザー電源使用时)
- 20.3 シリアル書込み接続例(ライター電源から供給時)
- 20.4 フラッシュマイコンプログラマとの最小限の接続例(ユーザー電源使用时)
- 20.5 フラッシュマイコンプログラマとの最小限の接続例(ライター電源から供給時)

20.1 富士通標準シリアルオンボード書込みの標準構成

MB90F562/F568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。MB90F562/F568仕様について以下に解説します。

富士通標準シリアルオンボード書込みの標準構成

富士通標準シリアルオンボード書込みには、株式会社ワイ・ディ・シー製フラッシュマイコンプログラマ (AF220/AF210/AF120/AF110) を使用します。



< 注意事項 >

フラッシュマイコンプログラマ (AF220/AF210/AF120/AF110) の機能・操作方法および、接続用汎用共通ケーブル (AZ221)、コネクタにつきましては、株式会社ワイ・ディ・シー殿へお問い合わせください。

表 20.1-1 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	機能説明
MD2, MD1, MD0	モード端子	MD2=1, MD=1, MD=0に設定する事で、シリアル書込みモードになります。
X0, X1	発振用端子	シリアル書込みモード時にCPU内部動作クロックのは、PLLクロック 1通倍になりますので、内部動作クロック周波数は、発振クロック周波数と同様になります。従いまして、シリアル書込みを行う場合、高速発振入力端子に入力可能な周波数は1MHz ~ 16MHzとなります。
P00, P01	書込みプログラム起動端子	P00に"L"レベル, P01に"H"レベルを入力してください。
RST	リセット端子	-
SIN1	シリアルデータ入力端子	UART0, UART1をCLK同期モードとして使用します。
SOT1	シリアルデータ出力端子	書込みモードにおいて、UART0のCLK同期モードで使用する端子はSIN1, SOT1, SCK0となります。
SCK0	シリアルクロック入力端子	
C	C端子	電源安定化の容量端子です。外部に0.1 μ F程度のセラミックコンデンサを接続してください。
VCC	電源電圧供給端子	書込み電圧 (MB90F562: 5V \pm 10%, MB90F568: 3V \pm 10%) をユーザーシステムから供給する場合は、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。
VSS	GND端子	フラッシュマイコンプログラマのGNDと共通にしてください。

P00, P01, SIN1, SOT1, SCK0 端子をユーザーシステムで使用される場合は、図 20.1-1に示す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、シリアルオンボード書込み中は、ユーザー回路を切り離して下さい。

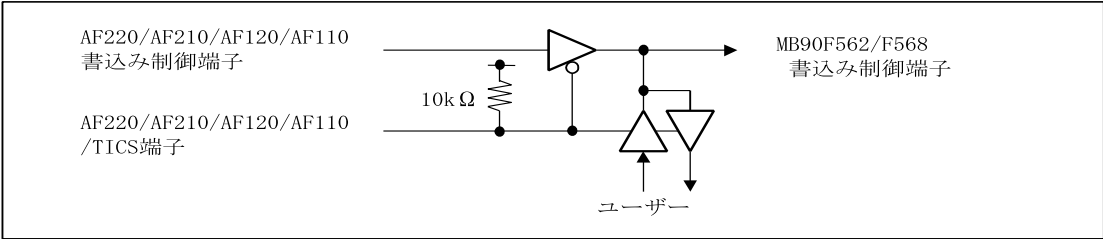


図 20.1-1 制御回路

20.2項～20.5項に、以下の4種類のシリアル書込み接続例を示していますので、参照してください。

- シリアル書込み接続例(ユーザー電源使用)
- シリアル書込み接続例(フラッシュマイコンプログラマ電源使用)
- フラッシュマイコンプログラマとの最小限の接続例(ユーザー電源使用)
- フラッシュマイコンプログラマとの最小限の接続例(フラッシュマイコンプログラマ電源使用)

MB90F562/F568の入力可能なシリアルクロック周波数は以下の計算式により求められます。従いまして、御使用の発振クロック周波数によって、シリアルクロック入力周波数をフラッシュマイコンプログラマの設定にて変更してください。

入力可能なシリアルクロック周波数=0.125×発振クロック周波数

表 20.1-2 最大シリアルクロック周波数

発振クロック 周波数	マイコンの入力可能な最大 シリアルクロック周波数	AF220/AF210/AF120/AF110の 設定可能な最大シリアル クロック周波数	AF200の 設定可能な最大シリアル クロック周波数
4MHz時	500kHz	500kHz	500kHz
8MHz時	1MHz	850kHz	500kHz
16MHz時	2MHz	1.25MHz	500kHz

表 20.1-2 フラッシュマイコンプログラマ (AF220/AF210/AF120/AF110) のシステム構成
(株式会社ワイ・ディ・シー製)

型 格		機 能
本 体	AF200/AC4P	イーサネットインターフェイス内蔵モデル /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ーフェイス内蔵モデル /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
AZ264		電源レギュレータ (MB90F568 : 3V製品にフラッシュマイコンプログラマから電源供給する場合に必要です。)
/P2		2MB PC Card (オプション) FLASHメモリ容量 ~128KB対応
/P4		4MB PC Card (オプション) FLASHメモリ容量 ~512KB対応

問い合わせ先：株式会社ワイ・ディ・シー 電話：042-333-6224

< 注意事項 >

AF200フラッシュマイコンプログラマにつきましては終息製品ですが，コントロールモジュールFF201を用いることで対応可能です。シリアル書込み接続例に関しましても，事項に示します接続例にて対応可能です。

20.2 シリアル書込み接続例(ユーザー電源使用時)

図 20.2-1に、マイコンの電源電圧をユーザー電源より供給する場合の、シリアル書込み接続例を示します。モード端子MD2, MD0にはAF220/AF210/AF120/AF110のTAUX3, TMODEよりMD2=1, MD0=0が入力されます。

・シリアル書換えモード：MD2，MD1，MD0="110B"

シリアル書込み接続例（ユーザー電源使用時）

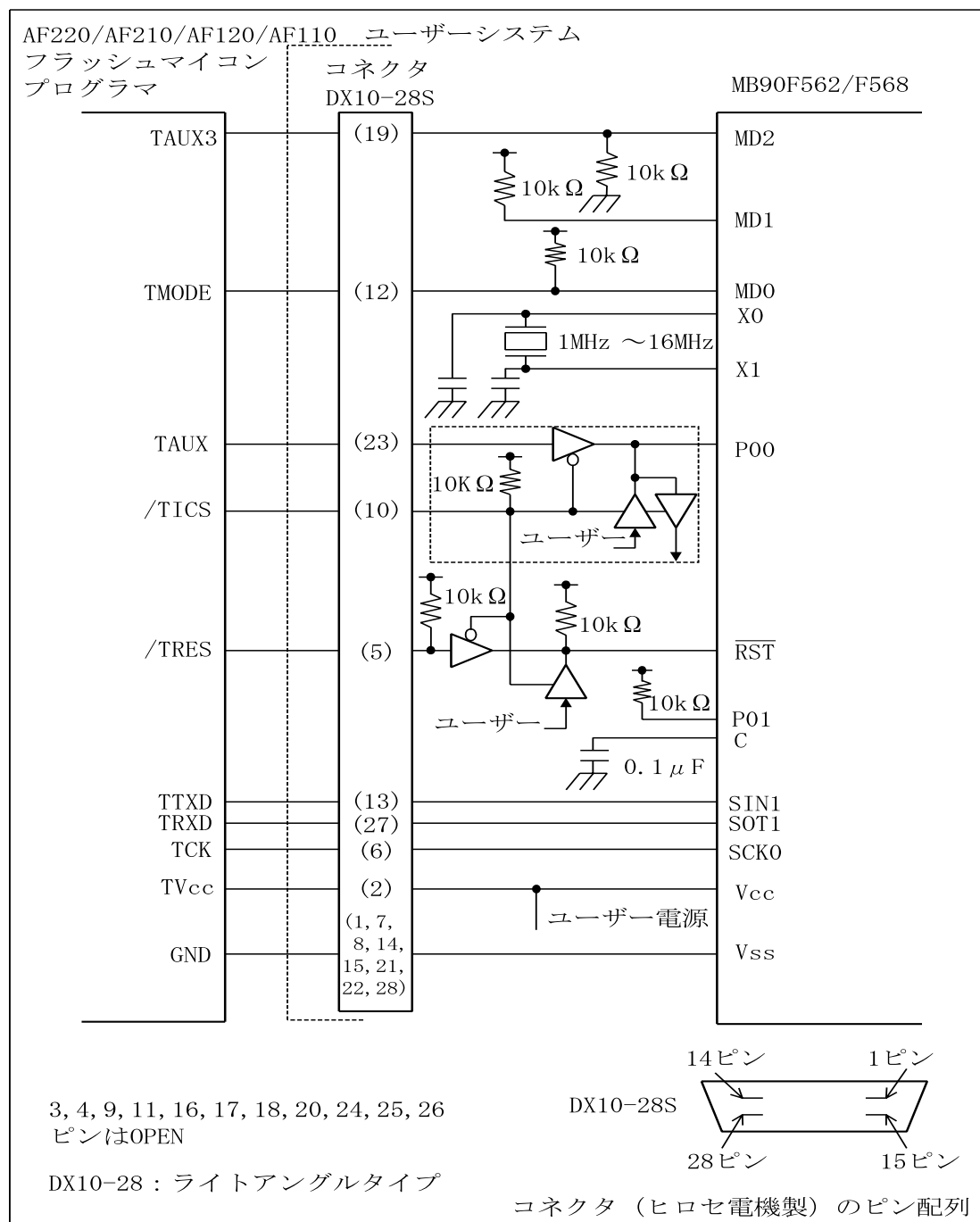
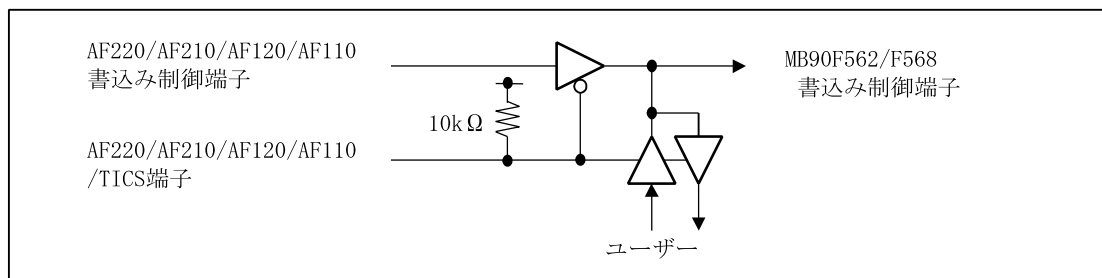


図 20.2-1 MB90F562/F568 内部ベクタモードでのシリアル書込み接続例（ユーザー電源使用時）

- SIN1, SOT1, SCK0端子をユーザーシステムでも使用する場合には, P00と同様に, 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, シリアル書き込み中はユーザー回路を切り離してください。



- AF220/AF210/AF120/AF110との接続は, ユーザー電源をOffしてください。

20.3 シリアル書き込み接続例(ライタ電源から供給時)

図 20.3-1に、マイコンの電源電圧をライタ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。モード端子MD2, MD0にはAF220/AF210/AF120/AF110のTAUX3, TMODEよりMD2=1, MD0=0が入力されます。

・シリアル書換えモード：MD2, MD1, MD0="110B"

シリアル書き込み接続例（ライタ電源から供給時）

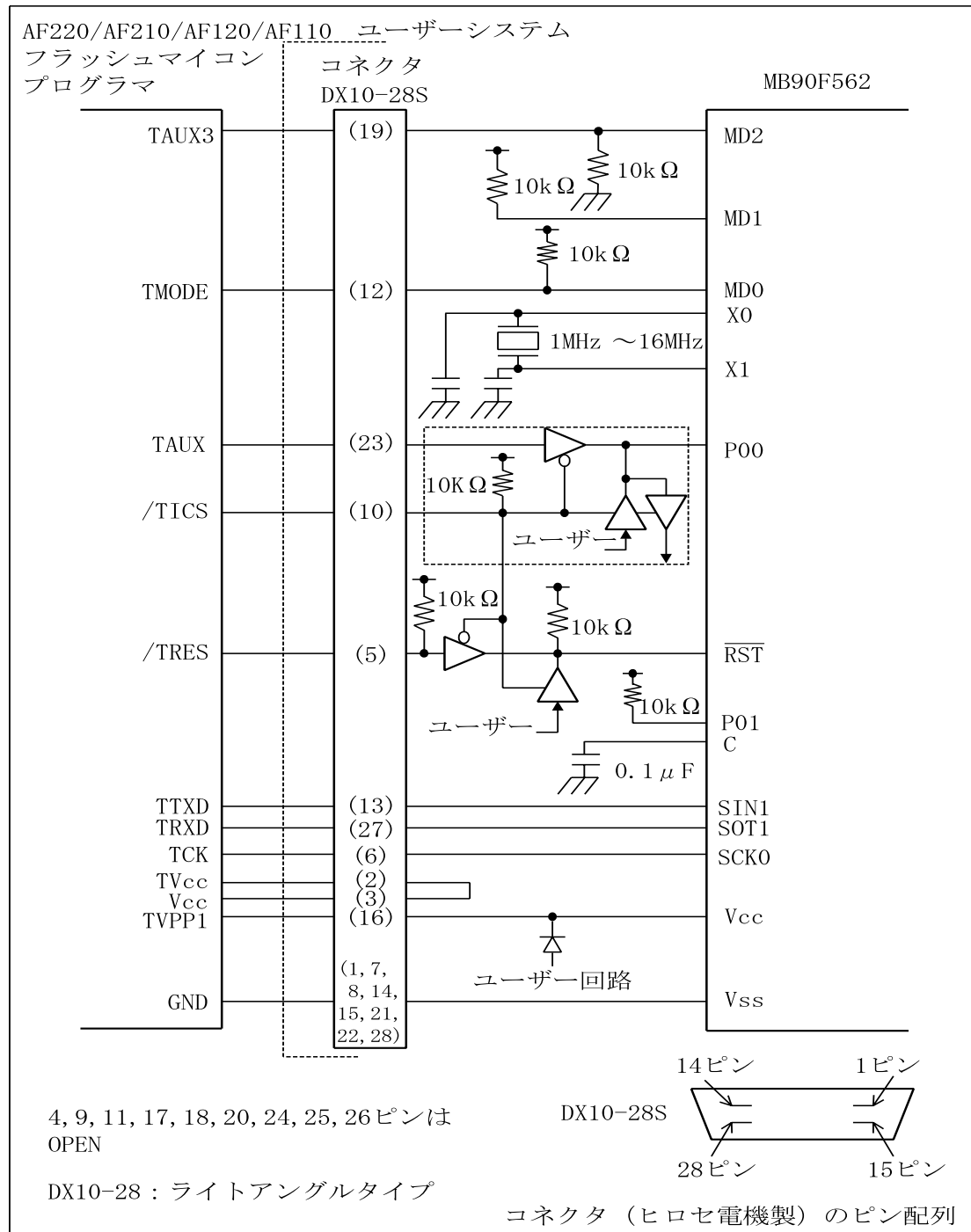


図 20.3-1 MB90F562 内部ベクタモードでのシリアル書き込み接続例（ライタ電源から供給時）

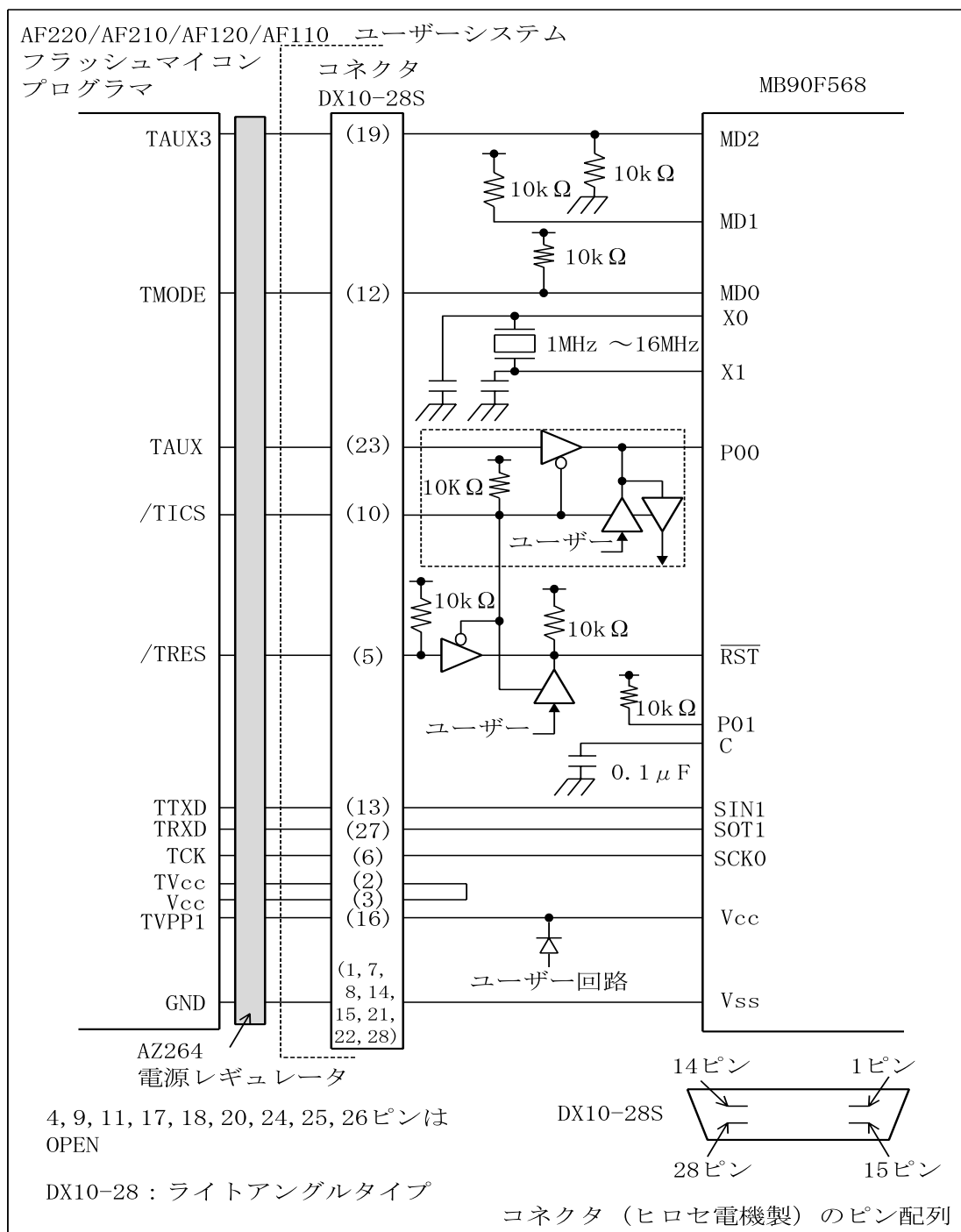
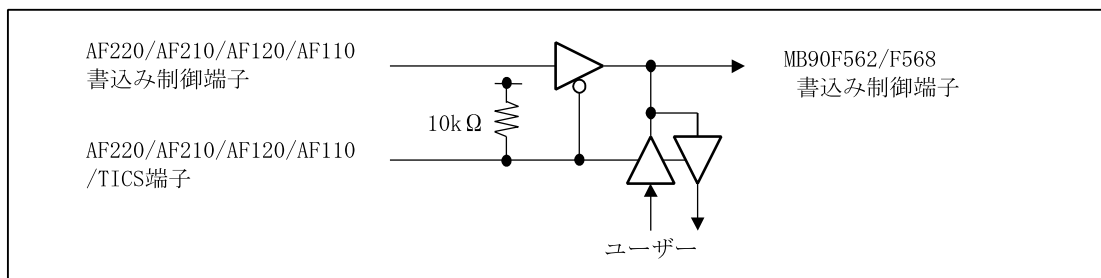


図 20.3-2 MB90F568 内部ベクタモードでのシリアル書き込み接続例 (ライタ電源から供給時)

- SIN1, SOT1, SCK0端子をユーザーシステムでも使用する場合には, P00と同様に, 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, シリアル書き込み中はユーザー回路を切り離してください。



- AF220/AF210/AF120/AF110との接続は, ユーザー電源をOffしてください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給する場合は, ユーザー電源と短絡しないでください。

20.4 フラッシュマイコンプログラムの最小限の接続例 (ユーザー電源使用時)

図 20.4-1に、マイコンの電源電圧をユーザ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。

・シリアル書換えモード：MD2, MD1, MD0="110_B"

フラッシュマイコンプログラムの最小限の接続例（ユーザー電源使用時）

フラッシュメモリへ書き込みでは、MB90F562/F568の端子を図 20.4-1に示すように設定した場合、MD2, MD1, MD0, P00とフラッシュマイコンプログラムの接続の必要はありません。

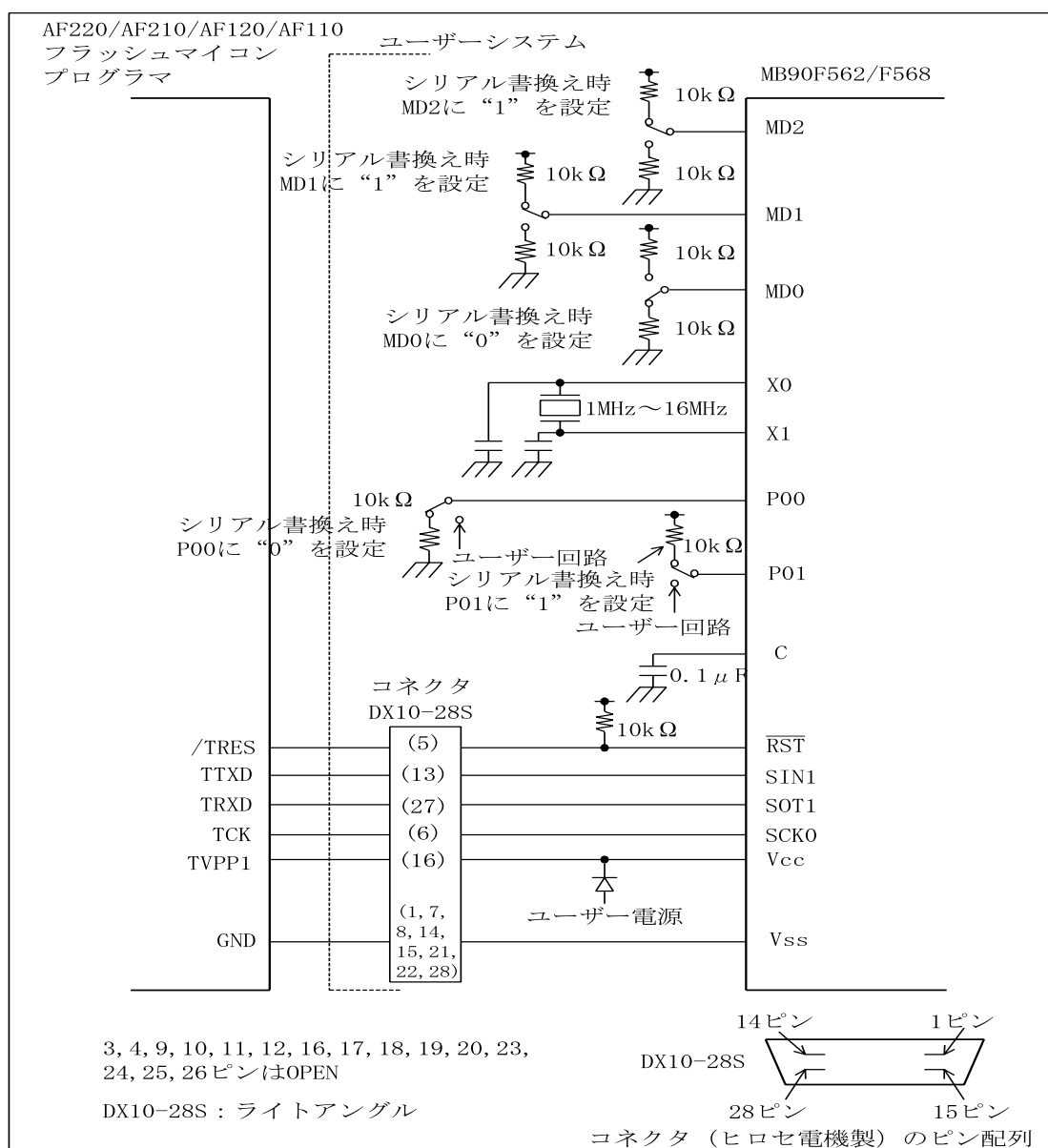


図 20.4-1 フラッシュマイコンプログラムの最小限の接続例（ユーザー電源使用時）

20.4 フラッシュマイコンプログラマとの最小限の接続例(ユーザー電源使用時)

- AF220/AF210/AF120/AF110との接続は、ユーザー電源をOffしてください。

20.5 フラッシュマイコンプログラマとの最小限の接続例 (ライタ電源から供給時)

図 20.5-1に、マイコンの電源電圧をライタ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。

・シリアル書換えモード：MD2，MD1，MD0="110_B"

フラッシュマイコンプログラマとの最小限の接続例（ライタ電源から供給時）

フラッシュメモリへ書込みでは、MB90F562/F568の端子を図 20.5-1に示すように設定した場合、MD2,MD1,MD0,P00とフラッシュマイコンプログラマとの接続の必要はありません。

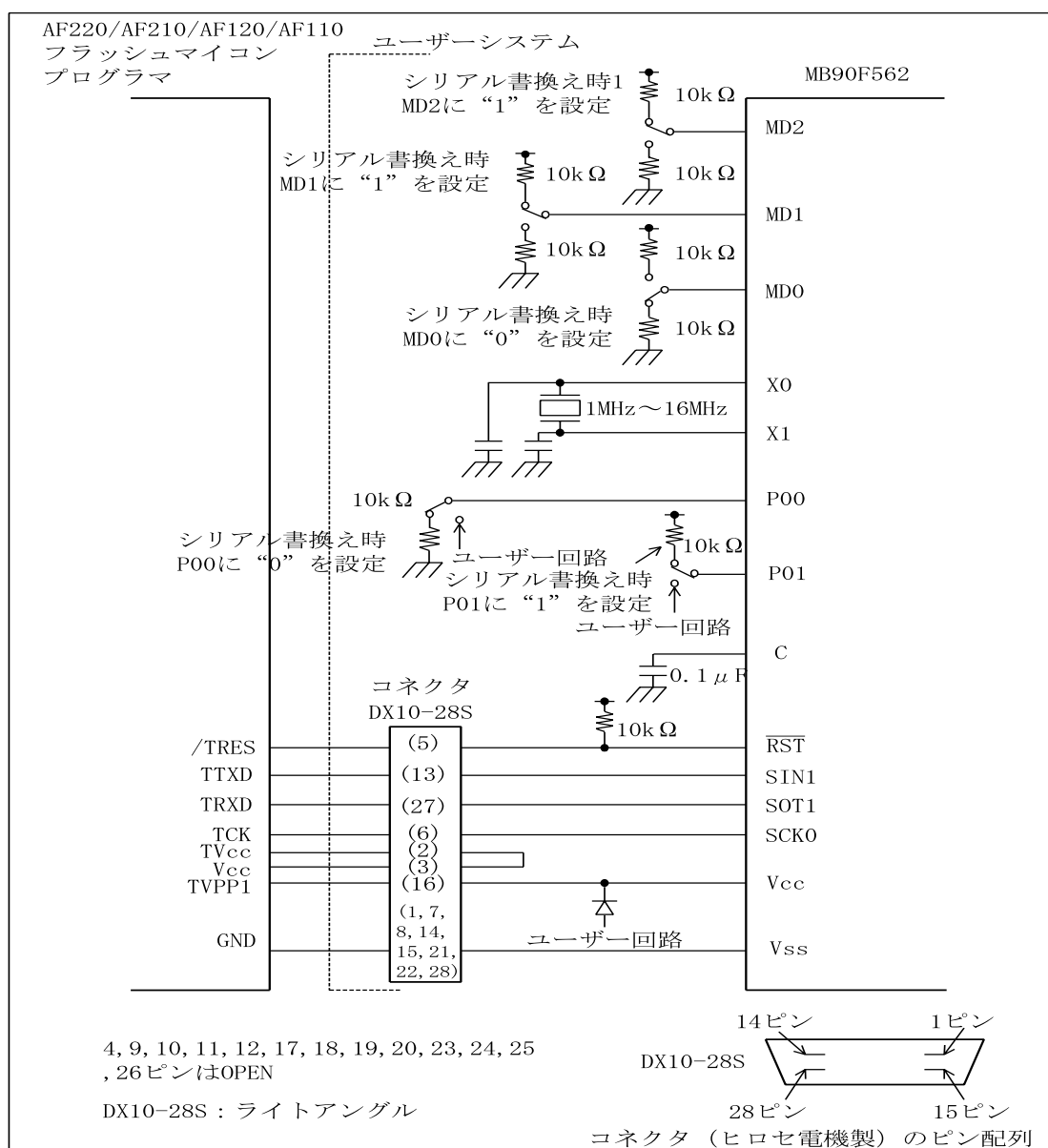


図 20.5-1 フラッシュマイコンプログラマとの最小限の接続例（ライタ電源から供給時）

20.5 フラッシュマイコンプログラマとの最小限の接続例(ライタ電源から供給時)

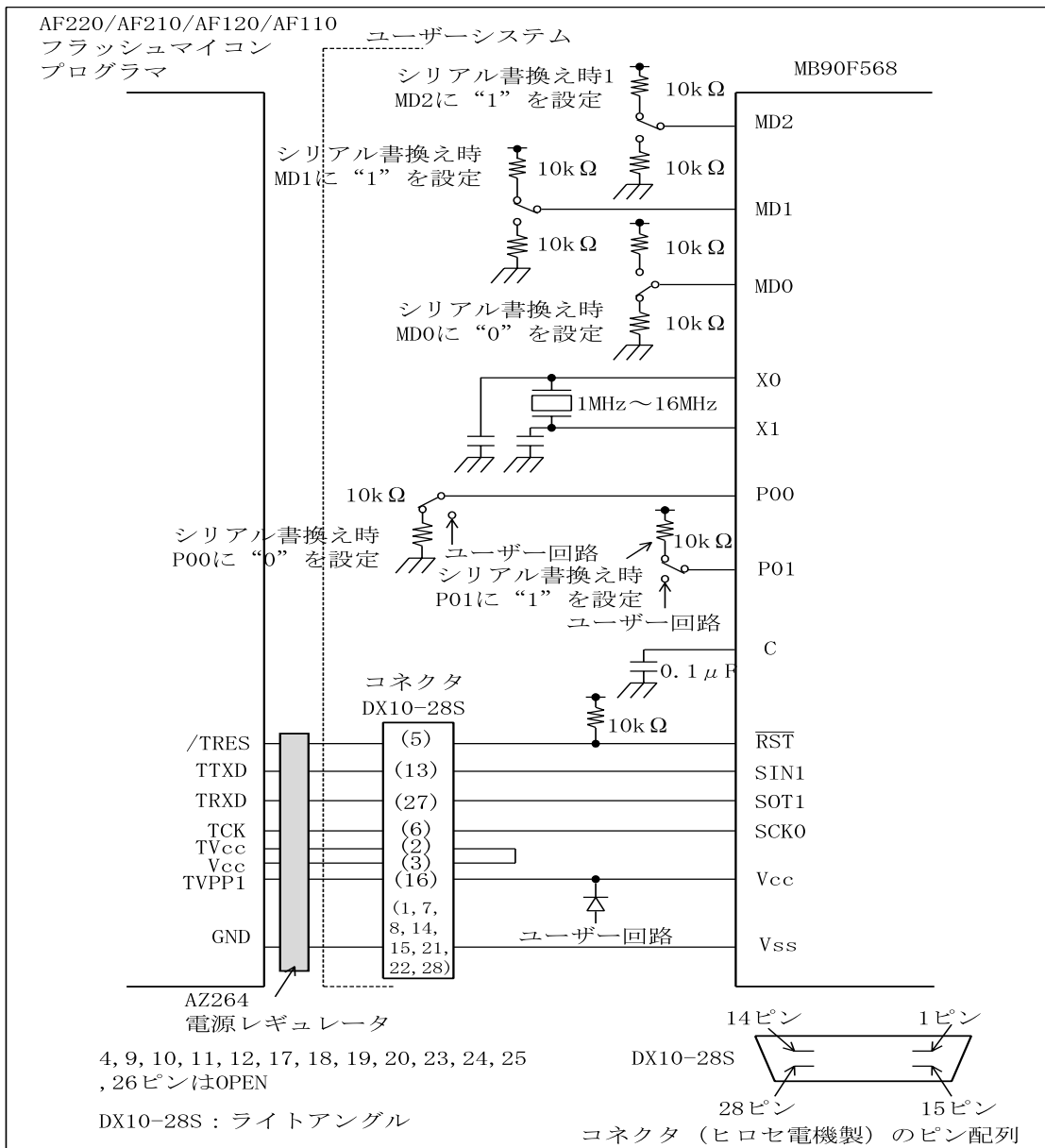


図 20.5-2 フラッシュマイコンプログラマとの最小限の接続例（ライタ電源から供給時）

- AF220/AF210/AF120/AF110との接続は、ユーザー電源をOffしてください。
- 書込み電源をAF220/AF210/AF120/AF110から供給する場合は、ユーザー電源と短絡しないでください。

付録

ここでは、I/Oマップ、命令一覧表などを掲載します。

- A I/Oマップ
- B 命令概要

A I/Oマップ

MB90560/565シリーズに内蔵された周辺機能の各レジスタは、付表 A-1に示すようなアドレスが割り当てられています。

I/Oマップ

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ略称	レジスタ名称	書込み / 読み込み	リソース名	初期値
000000 _H	PDR0	ポート0データレジスタ	R / W	ポート0	XXXXXXXX _B
000001 _H	PDR1	ポート1データレジスタ	R / W	ポート1	XXXXXXXX _B
000002 _H	PDR2	ポート2データレジスタ	R / W	ポート2	XXXXXXXX _B
000003 _H	PDR3	ポート3データレジスタ	R / W	ポート3	XXXXXXXX _B
000004 _H	PDR4	ポート4データレジスタ	R / W	ポート4	XXXXXXXX _B
000005 _H	PDR5	ポート5データレジスタ	R / W	ポート5	XXXXXXXX _B
000006 _H	PDR6	ポート6データレジスタ	R / W	ポート6	XXXXXXXX _B
000007 _H 00000F _H	使用禁止				
000010 _H	DDR0	ポート0方向レジスタ	R / W	ポート0	00000000 _B
000011 _H	DDR1	ポート1方向レジスタ	R / W	ポート1	00000000 _B
000012 _H	DDR2	ポート2方向レジスタ	R / W	ポート2	00000000 _B
000013 _H	DDR3	ポート3方向レジスタ	R / W	ポート3	00000000 _B
000014 _H	DDR4	ポート4方向レジスタ	R / W	ポート4	X0000000 _B
000015 _H	DDR5	ポート5方向レジスタ	R / W	ポート5	00000000 _B
000016 _H	DDR6	ポート6方向レジスタ	R / W	ポート6	XXXX0000 _B
000017 _H	ADER	アナログ入力許可レジスタ	R / W	ポート5, A/D コンバータ	11111111 _B
000018 _H 00001F _H	使用禁止				
000020 _H	SMR0	モードレジスタch0	R / W	UART0	0000X00 _B
000021 _H	SCR0	制御レジスタch0	W, R / W		00000100 _B
000022 _H	SIDR0	インプットデータレジスタch0	R		XXXXXXXX _B
	SODR0	アウトプットデータレジスタch0	W		
000023 _H	SSR0	ステータスレジスタch0	R, R / W		00001000 _B
000024 _H	SMR1	モードレジスタch1	R / W	UART1	0000X00 _B
000025 _H	SCR1	制御レジスタch1	W, R / W		00000100 _B
000026 _H	SIDR1	インプットデータレジスタch1	R		XXXXXXXX _B
	SODR1	アウトプットデータレジスタch1	W		
000027 _H	SSR1	ステータスレジスタch1	R, R / W		00001000 _B
000028 _H	使用禁止				
000029 _H	CDCR0	通信プリスケラ制御レジスタch0	R / W	通信 プリスケラ	0XX0000 _B
00002A _H	使用禁止				
00002B _H	CDCR1	通信プリスケラ制御レジスタch1	R / W	通信 プリスケラ	0XX0000 _B
00002C _H 00002F _H	使用禁止				
000030 _H	ENIR	DTP / 割込み許可レジスタ	R / W	DTP /	00000000 _B
000031 _H	EIRR	DTP / 割込み要因レジスタ	R / W	外部割込み	XXXXXXXX _B

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ略称	レジスタ名称	書込み / 読み込み	リソース名	初期値
000032 _H	ELVR	要求レベル設定レジスタ 下位	R / W	DTP /	00000000 _B
000033 _H		要求レベル設定レジスタ 上位	R / W	外部割込み	00000000 _B
000034 _H	ADCS0	A/D制御ステータスレジスタ 下位	R / W	8/10ビット A/Dコンバータ	00000000 _B
000035 _H	ADCS1	A/D制御ステータスレジスタ 上位	W, R / W		00000000 _B
000036 _H	ADCRO	A/Dデータレジスタ 下位	R		XXXXXXXX _B
000037 _H	ADCR1	A/Dデータレジスタ 上位	R, W		00000XXX _B
000038 _H	PRLLO	PPGリロードレジスタch0 下位	R / W	8/16ビット PPGタイマ	XXXXXXXX _B
000039 _H	PRLH0	PPGリロードレジスタch0 上位	R / W		XXXXXXXX _B
00003A _H	PRLLO	PPGリロードレジスタch1 下位	R / W		XXXXXXXX _B
00003B _H	PRLH1	PPGリロードレジスタch1 上位	R / W		XXXXXXXX _B
00003C _H	PPGC0	PPGコントロールレジスタch0 下位	R / W		00000001 _B
00003D _H	PPGC1	PPGコントロールレジスタch1 上位	R / W		00000001 _B
00003E _H	PCS01	PPGクロック制御レジスタch0, 1	R / W		000000XX _B
00003F _H	使用禁止				
000040 _H	PRLLO	PPGリロードレジスタch2 下位	R / W	8/16ビット PPGタイマ	XXXXXXXX _B
000041 _H	PRLH2	PPGリロードレジスタch2 上位	R / W		XXXXXXXX _B
000042 _H	PRLLO	PPGリロードレジスタch3 下位	R / W		XXXXXXXX _B
000043 _H	PRLH3	PPGリロードレジスタch3 上位	R / W		XXXXXXXX _B
000044 _H	PPGC2	PPGコントロールレジスタch2 下位	R / W		00000001 _B
000045 _H	PPGC3	PPGコントロールレジスタch3 上位	R / W		00000001 _B
000046 _H	PCS23	PPGクロック制御レジスタch2, 3	R / W		000000XX _B
000047 _H	使用禁止				
000048 _H	PRLLO	PPGリロードレジスタch4 下位	R / W	8/16ビット PPGタイマ	XXXXXXXX _B
000049 _H	PRLH4	PPGリロードレジスタch4 上位	R / W		XXXXXXXX _B
00004A _H	PRLLO	PPGリロードレジスタch5 下位	R / W		XXXXXXXX _B
00004B _H	PRLH5	PPGリロードレジスタch5 上位	R / W		XXXXXXXX _B
00004C _H	PPGC4	PPGコントロールレジスタch4 下位	R / W		00000001 _B
00004D _H	PPGC5	PPGコントロールレジスタch5 上位	R / W		00000001 _B
00004E _H	PCS45	PPGクロック制御レジスタch4, 5	R / W		000000XX _B
00004F _H	使用禁止				
000050 _H	TMRR0	8ビットリロードレジスタch0	R / W	波形生成部	XXXXXXXX _B
000051 _H	DTCR0	8ビットタイマコントロール レジスタch0	R / W		00000000 _B
000052 _H	TMRR1	8ビットリロードレジスタch1	R / W		XXXXXXXX _B
000053 _H	DTCR1	8ビットタイマコントロール レジスタch1	R / W		00000000 _B
000054 _H	TMRR2	8ビットリロードレジスタch2	R / W		XXXXXXXX _B
000055 _H	DTCR2	8ビットタイマコントロール レジスタch2	R / W		00000000 _B
000056 _H	SIGCR	波形制御レジスタ	R / W		00000000 _B
000057 _H	使用禁止				
000058 _H	CPCLR	コンペアクリアレジスタ 下位	R / W	16ビット フリーラン タイマ	XXXXXXXX _B
000059 _H		コンペアクリアレジスタ 上位	R / W		XXXXXXXX _B
00005A _H	TCDDT	タイマデータレジスタ 下位	R / W		00000000 _B
00005B _H		タイマデータレジスタ 上位	R / W		00000000 _B
00005C _H	TCCS	タイマコントロールステータス レジスタ 下位	R / W		00000000 _B
00005D _H		タイマコントロールステータス レジスタ 上位	R / W		0XX00000 _B

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ略称	レジスタ名称	書き込み / 読み込み	リソース名	初期値
00005E _H	使用禁止				
00005F _H					
000060 _H	IPCP0	インプットキャプチャデータレジスタ ch0 下位	R	インプット キャプチャ	XXXXXXXX _B
000061 _H		インプットキャプチャデータレジスタ ch0 上位	R		XXXXXXXX _B
000062 _H	IPCP1	インプットキャプチャデータレジスタ ch1 下位	R		XXXXXXXX _B
000063 _H		インプットキャプチャデータレジスタ ch1 上位	R		XXXXXXXX _B
000064 _H	IPCP2	インプットキャプチャデータレジスタ ch2 下位	R		XXXXXXXX _B
000065 _H		インプットキャプチャデータレジスタ ch2 上位	R		XXXXXXXX _B
000066 _H	IPCP3	インプットキャプチャデータレジスタ ch3 下位	R		XXXXXXXX _B
000067 _H		インプットキャプチャデータレジスタ ch3 上位	R		XXXXXXXX _B
000068 _H	ICS01	インプットキャプチャコントロールレ ジスタ01	R / W		00000000 _B
000069 _H	使用禁止				
00006A _H	ICS23	インプットキャプチャコントロール レジスタ23	R / W	インプット キャプチャ	00000000 _B
00006B _H 00006E _H	使用禁止				
00006F _H	ROMM	ROMミラー機能選択レジスタ	W	ROMミラー 機能選択 モジュール	XXXXXXXX1 _B
000070 _H	OCCP0	コンペアレジスタch0 下位	R / W	アウトプット コンペア	XXXXXXXX _B
000071 _H		コンペアレジスタch0 上位	R / W		XXXXXXXX _B
000072 _H	OCCP1	コンペアレジスタch1 下位	R / W		XXXXXXXX _B
000073 _H		コンペアレジスタch1 上位	R / W		XXXXXXXX _B
000074 _H	OCCP2	コンペアレジスタch2 下位	R / W		XXXXXXXX _B
000075 _H		コンペアレジスタch2 上位	R / W		XXXXXXXX _B
000076 _H	OCCP3	コンペアレジスタch3 下位	R / W		XXXXXXXX _B
000077 _H		コンペアレジスタch3 上位	R / W		XXXXXXXX _B
000078 _H	OCCP4	コンペアレジスタch4 下位	R / W		XXXXXXXX _B
000079 _H		コンペアレジスタch4 上位	R / W		XXXXXXXX _B
00007A _H	OCCP5	コンペアレジスタch5 下位	R / W		XXXXXXXX _B
00007B _H		コンペアレジスタch5 上位	R / W		XXXXXXXX _B
00007C _H	OCS0	コンペアコントロールレジスタ ch0 下位	R / W		0000XX00 _B
00007D _H	OCS1	コンペアコントロールレジスタ ch1 上位	R / W		XXX00000 _B
00007E _H	OCS2	コンペアコントロールレジスタ ch2 下位	R / W		0000XX00 _B
00007F _H	OCS3	コンペアコントロールレジスタ ch3 上位	R / W		XXX00000 _B

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ略称	レジスタ名称	書込み / 読み込み	リソース名	初期値
000080 _H	OCS4	コンペアコントロールレジスタ ch4 下位	R / W	アウトプット コンペア	0000XX00 _B
000081 _H	OCS5	コンペアコントロールレジスタ ch5 上位	R / W		XXX00000 _B
000082 _H	TMCSR0 : L	タイマコントロールステータス レジスタch0 下位	R / W	16ビット リロード タイマ	00000000 _B
000083 _H	TMCSR0 : H	タイマコントロールステータス レジスタch0 上位	R / W		XXXX0000 _B
000084 _H	TMR0	16ビットタイマレジスタch0 下位	R		XXXXXXXX _B
	TMRLR0	16ビットリロードレジスタch0 下位	W		XXXXXXXX _B
000085 _H	TMR0	16ビットタイマレジスタch0 上位	R		XXXXXXXX _B
	TMRHR0	16ビットリロードレジスタch0 上位	W		XXXXXXXX _B
000086 _H	TMCSR1 : L	タイマコントロールステータス レジスタch1 下位	R / W		00000000 _B
000087 _H	TMCSR1 : H	タイマコントロールステータス レジスタch1 上位	R / W		XXXX0000 _B
000088 _H	TMR1	16ビットタイマレジスタch1 下位	R		XXXXXXXX _B
	TMRLR1	16ビットリロードレジスタch1 下位	W		XXXXXXXX _B
000089 _H	TMR1	16ビットタイマレジスタch1 上位	R		XXXXXXXX _B
	TMRHR1	16ビットリロードレジスタch1 上位	W		XXXXXXXX _B
00008A _H 00008B _H	使用禁止				
00008C _H	RDR0	ポート0プルアップ抵抗設定レジスタ	R / W	ポート0	00000000 _B
00008D _H	RDR1	ポート1プルアップ抵抗設定レジスタ	R / W	ポート1	00000000 _B
00008E _H 00009D _H	使用禁止				
00009E _H	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	R / W	アドレス 一致検出	00000000 _B
00009F _H	DIRR	遅延割込み要因 / 解除レジスタ	R / W	遅延割込み	XXXXXXXX _B
0000A0 _H	LPMCR	低消費電力モードレジスタ	W, R / W	低消費電力 制御回路	00011000 _B
0000A1 _H	CKSCR	クロック選択レジスタ	R, R / W	クロック	11111100 _B
0000A2 _H 0000A7 _H	使用禁止				
0000A8 _H	WDTC	ウォッチドッグタイマ制御レジスタ	R, W	ウォッチ ドッグタイマ	1XXXX111 _B
0000A9 _H	TBTC	タイムベースタイマ制御レジスタ	W, R / W	タイムベ ースタイマ	1XX00100 _B
0000AA _H 0000AD _H	使用禁止				
0000AE _H	FMCS	フラッシュメモリコントロールス テータスレジスタ	R, W, R / W	フラッシュ メモリ	00000000 _B

付表 A-1 I/Oマップ (続き)

アドレス	レジスタ略称	レジスタ名称	書込み / 読み込み	リソース名	初期値	
0000AF _H	使用禁止					
0000B0 _H	ICR00	割込み制御レジスタ00（書込む場合）	W, R / W	割込み	XXXX0111 _B	
		割込み制御レジスタ00（読む場合）	R, R / W		XX000111 _B	
0000B1 _H	ICR01	割込み制御レジスタ01（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ01（読む場合）	R, R / W		XX000111 _B	
0000B2 _H	ICR02	割込み制御レジスタ02（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ02（読む場合）	R, R / W		XX000111 _B	
0000B3 _H	ICR03	割込み制御レジスタ03（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ03（読む場合）	R, R / W		XX000111 _B	
0000B4 _H	ICR04	割込み制御レジスタ04（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ04（読む場合）	R, R / W		XX000111 _B	
0000B5 _H	ICR05	割込み制御レジスタ05（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ05（読む場合）	R, R / W		XX000111 _B	
0000B6 _H	ICR06	割込み制御レジスタ06（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ06（読む場合）	R, R / W		XX000111 _B	
0000B7 _H	ICR07	割込み制御レジスタ07（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ07（読む場合）	R, R / W		XX000111 _B	
0000B8 _H	ICR08	割込み制御レジスタ08（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ08（読む場合）	R, R / W		XX000111 _B	
0000B9 _H	ICR09	割込み制御レジスタ09（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ09（読む場合）	R, R / W		XX000111 _B	
0000BA _H	ICR10	割込み制御レジスタ10（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ10（読む場合）	R, R / W		XX000111 _B	
0000BB _H	ICR11	割込み制御レジスタ11（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ11（読む場合）	R, R / W		XX000111 _B	
0000BC _H	ICR12	割込み制御レジスタ12（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ12（読む場合）	R, R / W		XX000111 _B	
0000BD _H	ICR13	割込み制御レジスタ13（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ13（読む場合）	R, R / W		XX000111 _B	
0000BE _H	ICR14	割込み制御レジスタ14（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ14（読む場合）	R, R / W		XX000111 _B	
0000BF _H	ICR15	割込み制御レジスタ15（書込む場合）	W, R / W		XXXX0111 _B	
		割込み制御レジスタ15（読む場合）	R, R / W		XX000111 _B	
0000C0 _H 0000FF _H	未使用領域					
000100 _H # _H	RAM領域					
# _H 001FE _H	予約領域					

付表 A-1 I/Oマップ (続き)

アドレス	レジスタ略称	レジスタ名称	書込み / 読み込み	リソース名	初期値
001FF0 _H	PADR0	プログラムアドレス検出レジスタch0 下位	R / W	アドレス 一致検出	XXXXXXXX _B
001FF1 _H		プログラムアドレス検出レジスタch0 中位	R / W		XXXXXXXX _B
001FF2 _H		プログラムアドレス検出レジスタch0 上位	R / W		XXXXXXXX _B
001FF3 _H	PADR1	プログラムアドレス検出レジスタch1 下位	R / W		XXXXXXXX _B
001FF4 _H		プログラムアドレス検出レジスタch1 中位	R / W		XXXXXXXX _B
001FF5 _H		プログラムアドレス検出レジスタch1 上位	R / W		XXXXXXXX _B
001FF6 _H 001FFF _H	未使用領域				

書込み / 読み込みについての説明

R/W : リード・ライト可能

R : リードオンリー

W : ライトオンリー

初期値についての説明

0 : 初期値は"0"です。

1 : 初期値は"1"です。

X : 初期値は不定です。

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。

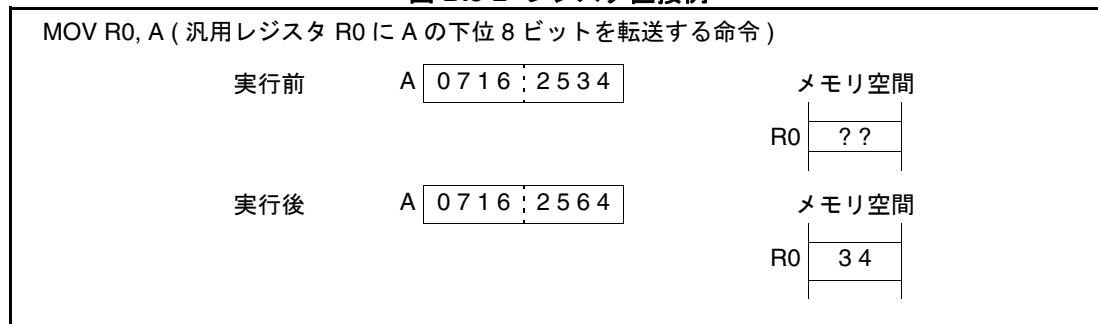
表 B.3-1 レジスタ直接

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。

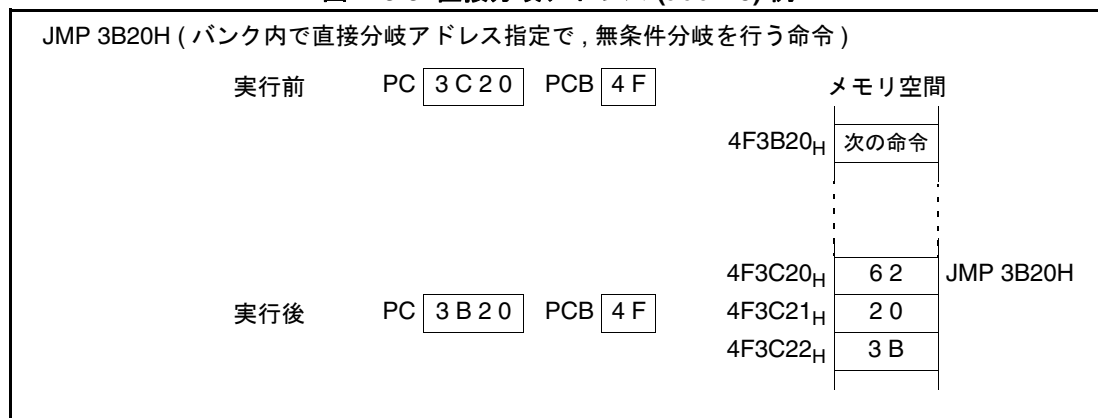
図 B.3-2 レジスタ直接例



● 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。

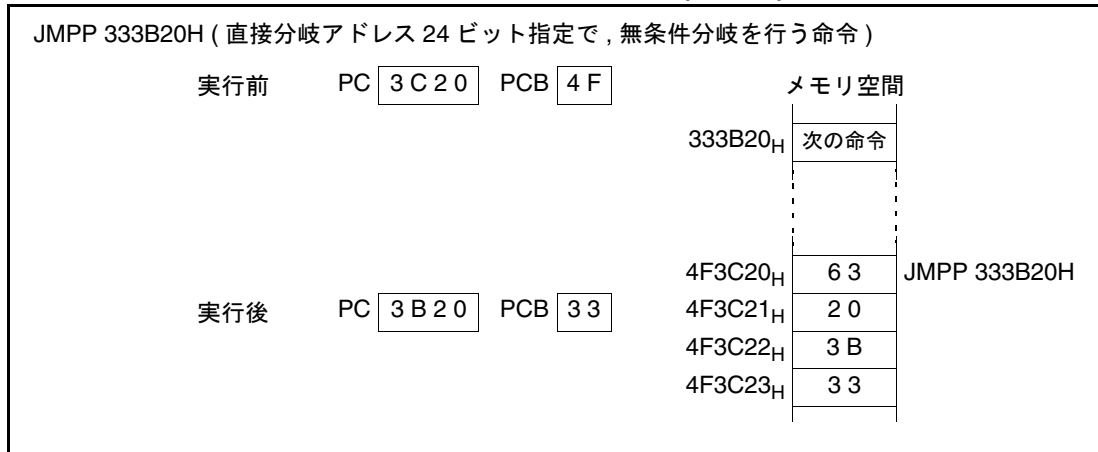
図 B.3-3 直接分岐アドレス (addr16) 例



● 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。

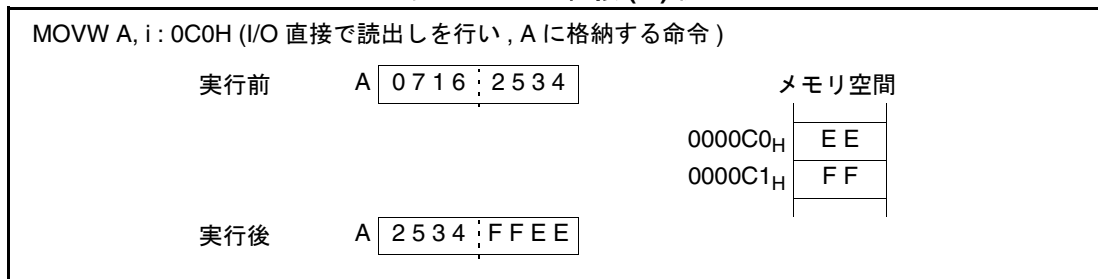
図 B.3-4 物理直接分岐アドレス (addr24) 例



● 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。

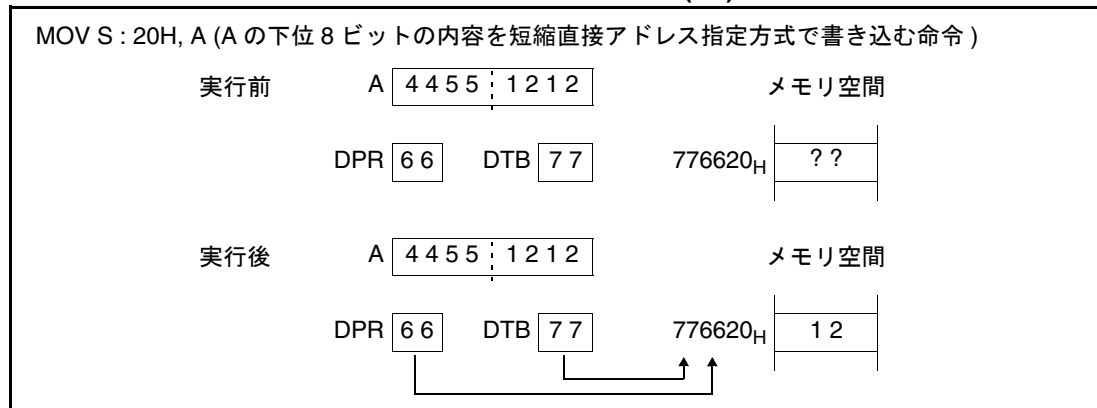
図 B.3-5 I/O 直接 (io) 例



● 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。

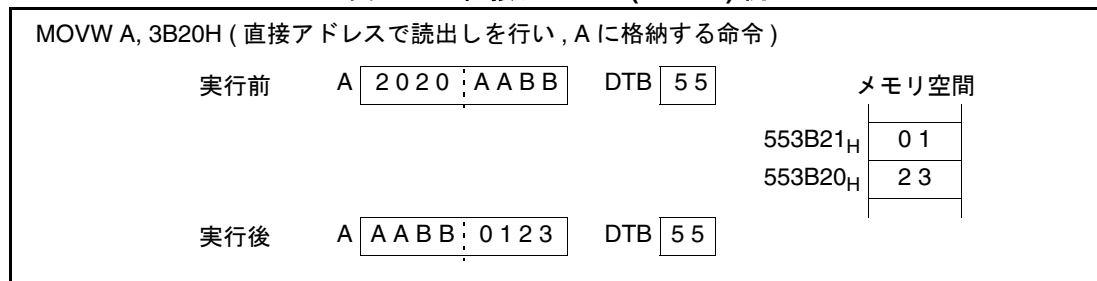
図 B.3-6 短縮直接アドレス (dir) 例



● 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。

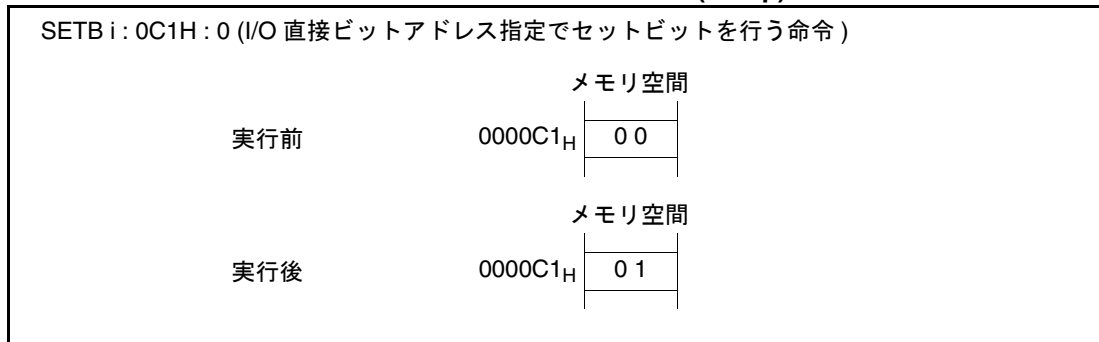
図 B.3-7 直接アドレス (addr16) 例



● 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。

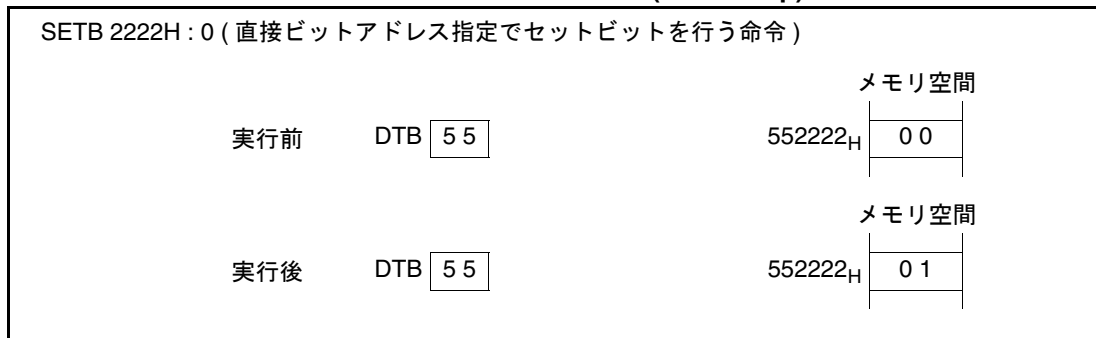
図 B.3-9 短縮直接ビットアドレス (dir: bp) 例



● 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

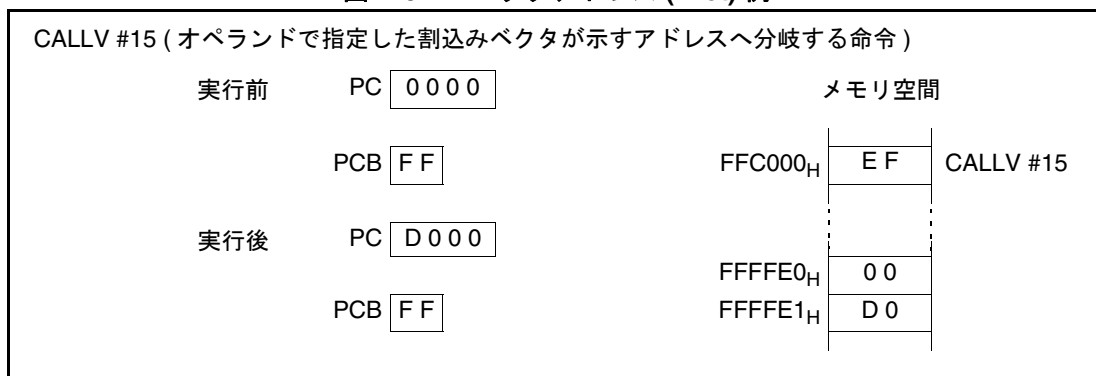


表 B.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE _H	XXXXFF _H
CALLV #1	XXXXFFC _H	XXXXFFD _H
CALLV #2	XXXXFFA _H	XXXXFFB _H
CALLV #3	XXXXFF8 _H	XXXXFF9 _H
CALLV #4	XXXXFF6 _H	XXXXFF7 _H
CALLV #5	XXXXFF4 _H	XXXXFF5 _H
CALLV #6	XXXXFF2 _H	XXXXFF3 _H
CALLV #7	XXXXFF0 _H	XXXXFF1 _H
CALLV #8	XXXXFEE _H	XXXXFEF _H
CALLV #9	XXXXFEC _H	XXXXFED _H
CALLV #10	XXXXFEA _H	XXXXFEB _H
CALLV #11	XXXXFE8 _H	XXXXFE9 _H
CALLV #12	XXXXFE6 _H	XXXXFE7 _H
CALLV #13	XXXXFE4 _H	XXXXFE5 _H
CALLV #14	XXXXFE2 _H	XXXXFE3 _H
CALLV #15	XXXXFE0 _H	XXXXFE1 _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

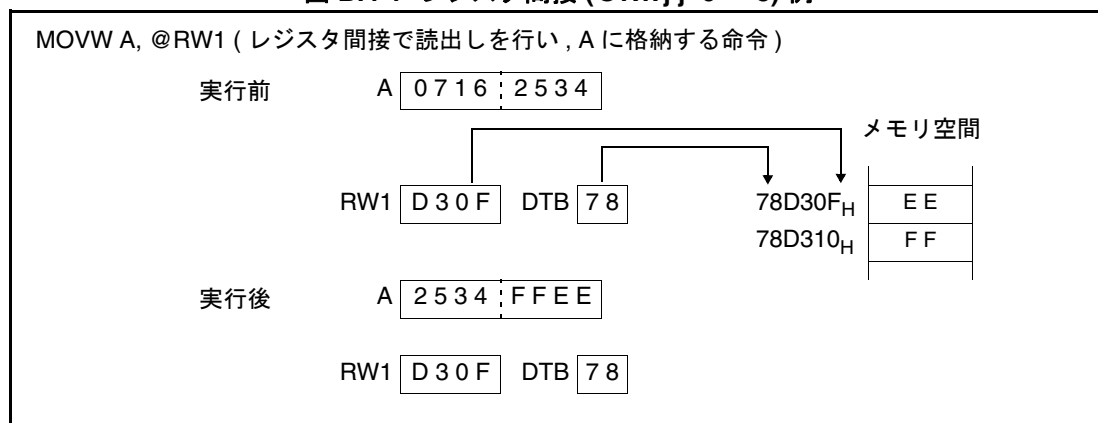
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

■ 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



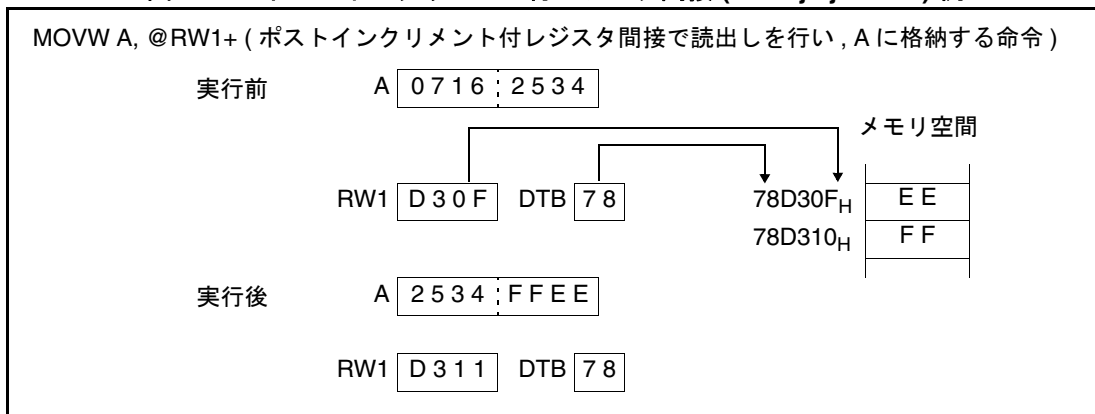
● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

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

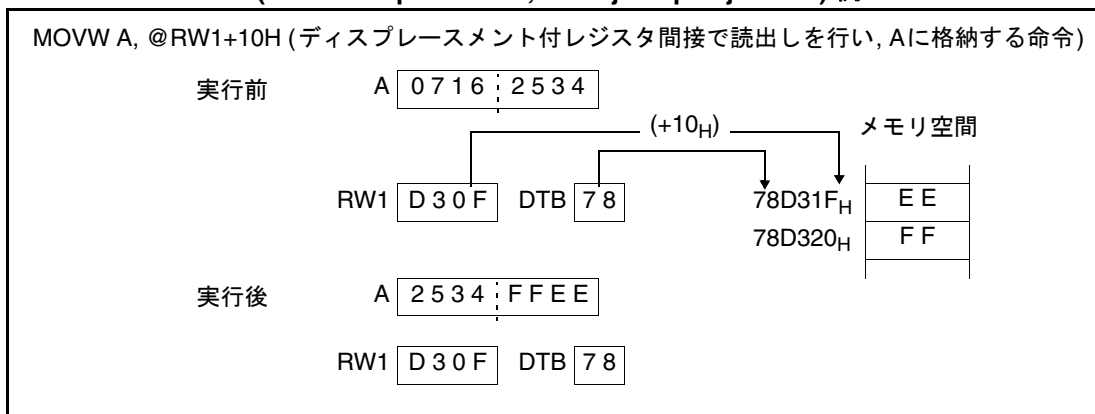


● ディスプレースメント付レジスタ間接

(@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。

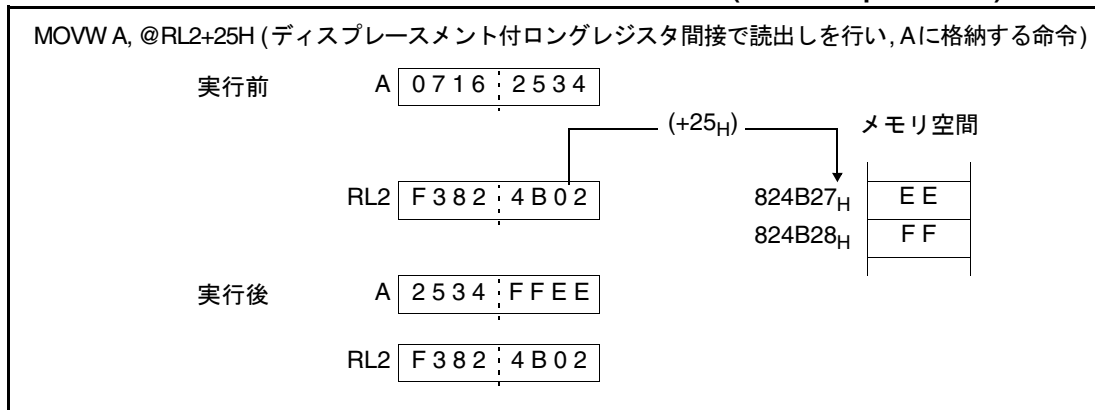
図 B.4-3 ディスプレースメント付レジスタ間接
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● 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



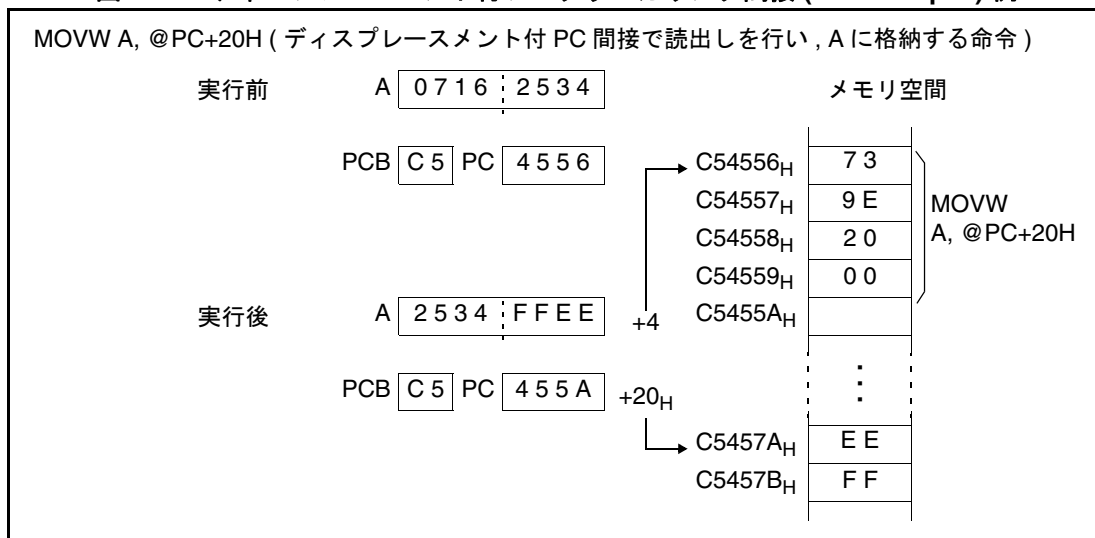
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(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。

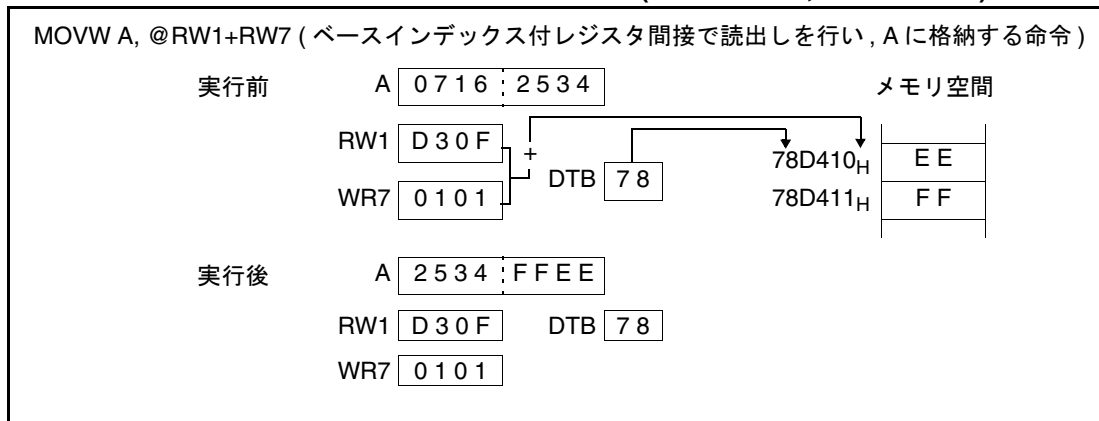
図 B.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例



● 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

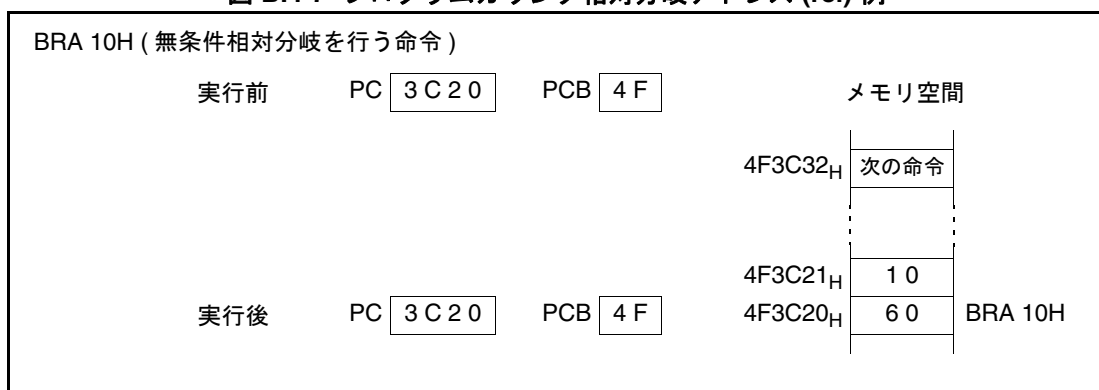


● プログラムカウンタ相対分岐アドレス (rel)

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

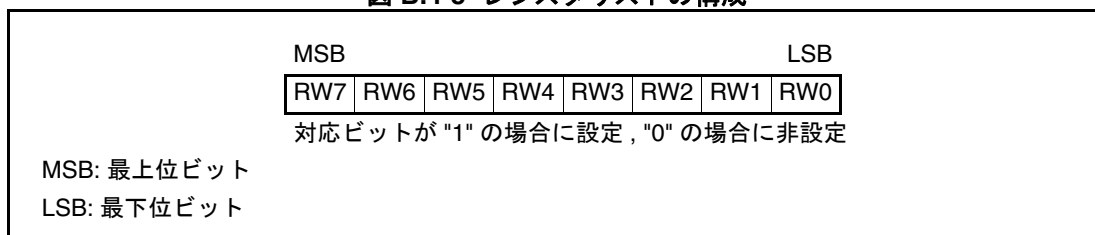
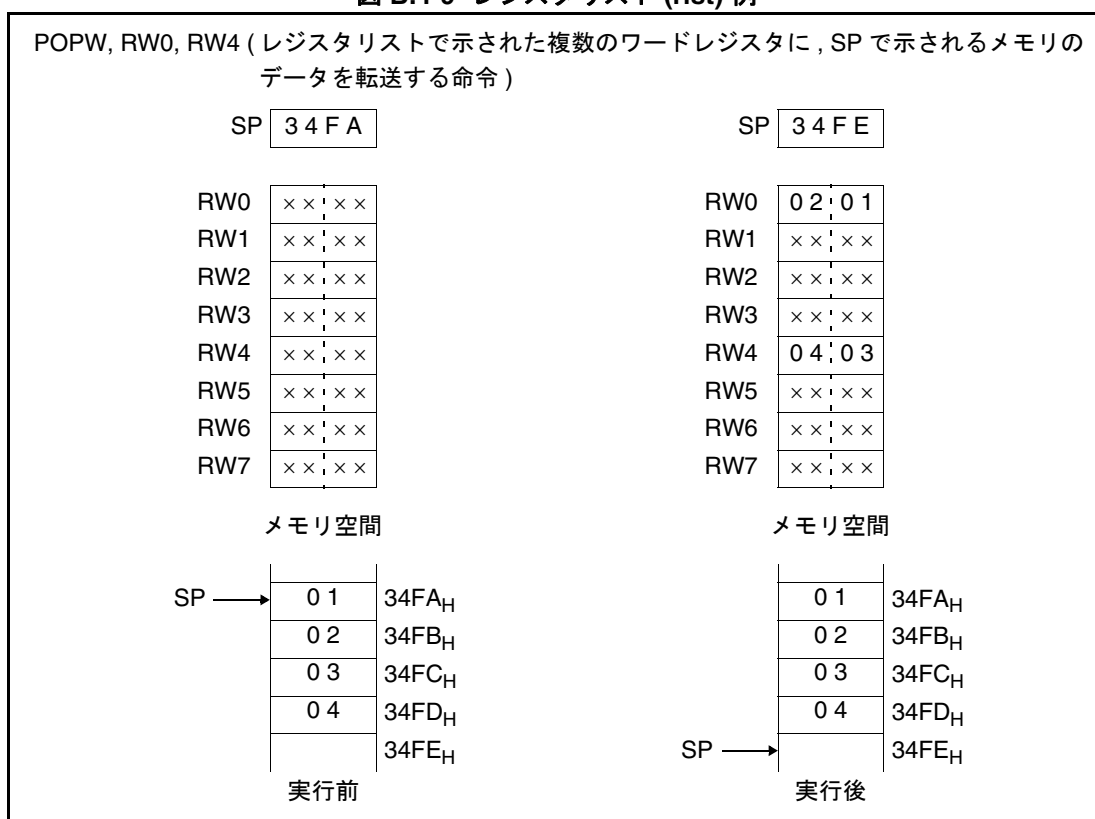


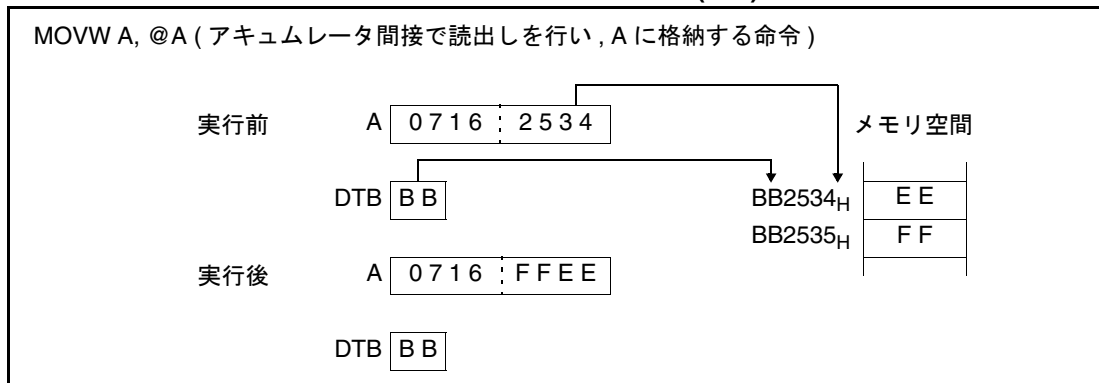
図 B.4-9 レジスタリスト (rlst) 例



● 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。

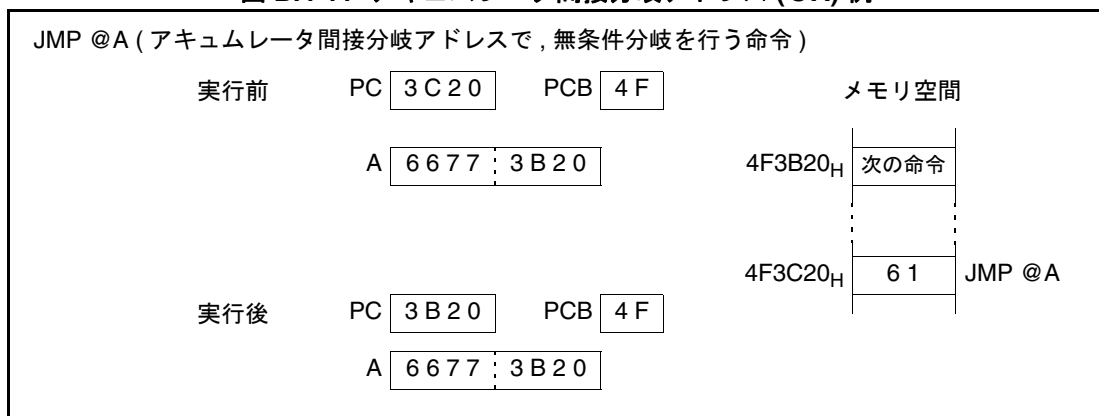
図 B.4-10 アキュムレータ間接 (@A) 例



● 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。

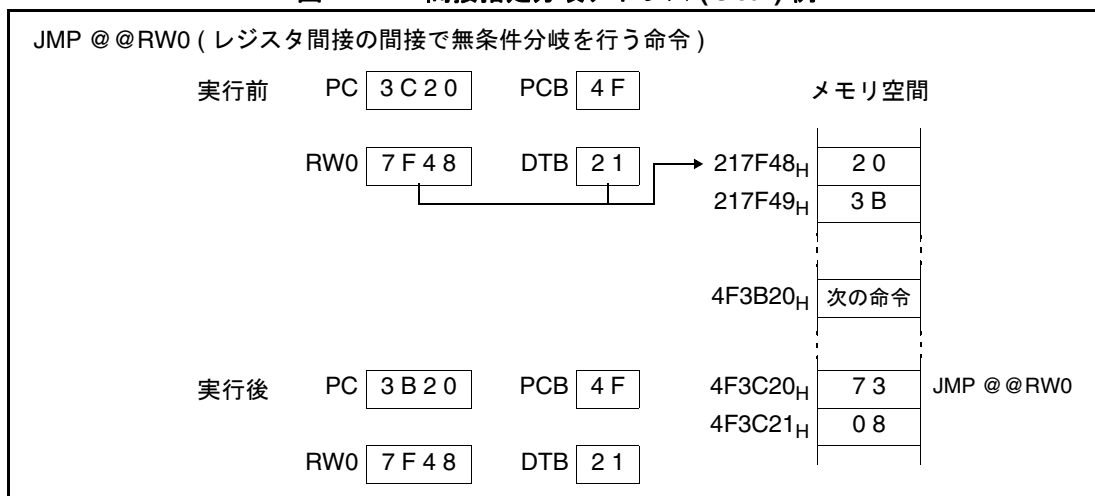
図 B.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 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。

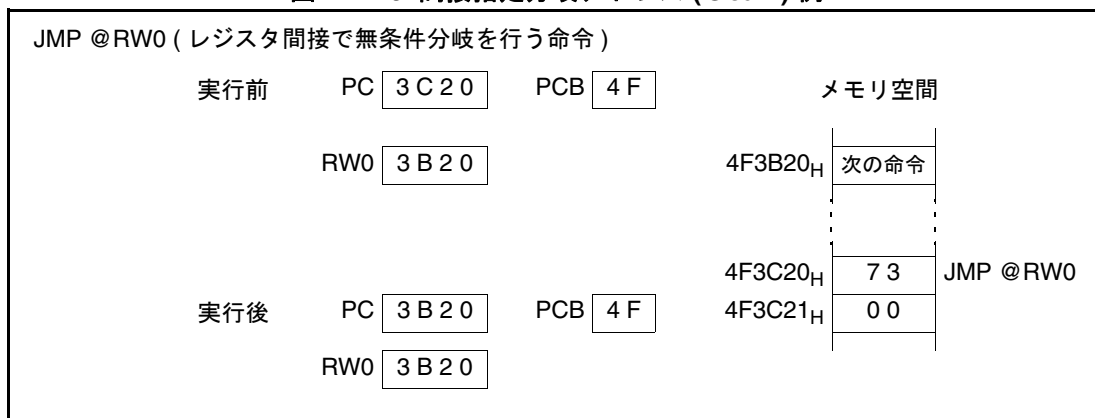
図 B.4-12 間接指定分岐アドレス (@ear) 例



● 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	
RMW	リードモディファイライト命令 (I 命令でメモリなどからデータを読み出し、メモリへ書き込む) であるかどうか示します。 *: リードモディファイライト命令である。 －: リードモディファイライト命令ではない。 (注意事項) 読み書きで意味の異なるアドレスには使用できません。

表 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) \leftarrow (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD A,dir	2	5	0	(b)	byte (A) \leftarrow (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD A,ear	2	3	1	0	byte (A) \leftarrow (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD ear,A	2	3	2	0	byte (ear) \leftarrow (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) \leftarrow (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC A	1	2	0	0	byte (A) \leftarrow (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,ear	2	3	1	0	byte (A) \leftarrow (A) + (ear) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) + (eam) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC A	1	3	0	0	byte (A) \leftarrow (AH) + (AL) + (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
SUB A,#imm8	2	2	0	0	byte (A) \leftarrow (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB A,dir	2	5	0	(b)	byte (A) \leftarrow (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB A,ear	2	3	1	0	byte (A) \leftarrow (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB ear,A	2	3	2	0	byte (ear) \leftarrow (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB eam,A	2+	5 + (a)	0	2 × (b)	byte (eam) \leftarrow (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC A	1	2	0	0	byte (A) \leftarrow (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,ear	2	3	1	0	byte (A) \leftarrow (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC A,eam	2+	4 + (a)	0	(b)	byte (A) \leftarrow (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC A	1	3	0	0	byte (A) \leftarrow (AH) - (AL) - (C) (10 進数)	Z	-	-	-	-	*	*	*	*	-
ADDW A	1	2	0	0	word (A) \leftarrow (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW A,ear	2	3	1	0	word (A) \leftarrow (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW A,#imm16	3	2	0	0	word (A) \leftarrow (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW ear,A	2	3	2	0	word (ear) \leftarrow (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW A,ear	2	3	1	0	word (A) \leftarrow (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW A	1	2	0	0	word (A) \leftarrow (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW A,ear	2	3	1	0	word (A) \leftarrow (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW A,#imm16	3	2	0	0	word (A) \leftarrow (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW ear,A	2	3	2	0	word (ear) \leftarrow (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW eam,A	2+	5+(a)	0	2 × (c)	word (eam) \leftarrow (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW A,ear	2	3	1	0	word (A) \leftarrow (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL A,ear	2	6	2	0	long (A) \leftarrow (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL A,#imm32	5	4	0	0	long (A) \leftarrow (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL A,ear	2	6	2	0	long (A) \leftarrow (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL A,eam	2+	7+(a)	0	(d)	long (A) \leftarrow (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL A,#imm32	5	4	0	0	long (A) \leftarrow (A) - imm32	-	-	-	-	-	*	*	*	*	-

(注意事項) 表中の (a) ～ (d) は、表 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) \leftarrow (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND A,ear	2	3	1	0	byte (A) \leftarrow (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND A,eam	2+	4+(a)	0	(b)	byte (A) \leftarrow (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND ear,A	2	3	2	0	byte (ear) \leftarrow (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND eam,A	2+	5+(a)	0	2 \times (b)	byte (eam) \leftarrow (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR A,#imm8	2	2	0	0	byte (A) \leftarrow (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR A,ear	2	3	1	0	byte (A) \leftarrow (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR A,eam	2+	4+(a)	0	(b)	byte (A) \leftarrow (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR ear,A	2	3	2	0	byte (ear) \leftarrow (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR eam,A	2+	5+(a)	0	2 \times (b)	byte (eam) \leftarrow (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR A,#imm8	2	2	0	0	byte (A) \leftarrow (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR A,ear	2	3	1	0	byte (A) \leftarrow (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR A,eam	2+	4+(a)	0	(b)	byte (A) \leftarrow (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR ear,A	2	3	2	0	byte (ear) \leftarrow (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR eam,A	2+	5+(a)	0	2 \times (b)	byte (eam) \leftarrow (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT A	1	2	0	0	byte (A) \leftarrow not (A)	-	-	-	-	-	*	*	R	-	-
NOT ear	2	3	2	0	byte (ear) \leftarrow not (ear)	-	-	-	-	-	*	*	R	-	-
NOT eam	2+	5+(a)	0	2 \times (b)	byte (eam) \leftarrow not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW A	1	2	0	0	word (A) \leftarrow (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW A,#imm16	3	2	0	0	word (A) \leftarrow (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW A,ear	2	3	1	0	word (A) \leftarrow (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW ear,A	2	3	2	0	word (ear) \leftarrow (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW eam,A	2+	5+(a)	0	2 \times (c)	word (eam) \leftarrow (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW A	1	2	0	0	word (A) \leftarrow (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW A,#imm16	3	2	0	0	word (A) \leftarrow (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW A,ear	2	3	1	0	word (A) \leftarrow (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW ear,A	2	3	2	0	word (ear) \leftarrow (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW eam,A	2+	5+(a)	0	2 \times (c)	word (eam) \leftarrow (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW A	1	2	0	0	word (A) \leftarrow (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW A,#imm16	3	2	0	0	word (A) \leftarrow (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW A,ear	2	3	1	0	word (A) \leftarrow (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW A,eam	2+	4+(a)	0	(c)	word (A) \leftarrow (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW ear,A	2	3	2	0	word (ear) \leftarrow (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW eam,A	2+	5+(a)	0	2 \times (c)	word (eam) \leftarrow (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW A	1	2	0	0	word (A) \leftarrow not (A)	-	-	-	-	-	*	*	R	-	-
NOTW ear	2	3	2	0	word (ear) \leftarrow not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW eam	2+	5+(a)	0	2 \times (c)	word (eam) \leftarrow 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 : $3 \times (c) + (b)$

注 1) 分岐先アドレスの読み込み (word)

注 2) W : スタックへの退避 (word), R : 分岐先アドレスの読み込み (word)

注 3) スタックへの退避 (word)

注 4) W : スタックへの退避 (long), R : 分岐先アドレスの読み込み (long)

注 5) スタックへの退避 (long)

(注意事項) 表中の (a) ～ (d) は , 表 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) \leftarrow (dir:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,addr16:bp	4	5	0	(b)	byte (A) \leftarrow (addr16:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB A,io:bp	3	4	0	(b)	byte (A) \leftarrow (io:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB dir:bp,A	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
MOVB addr16:bp,A	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
MOVB io:bp,A	3	6	0	2 × (b)	bit (io:bp)b \leftarrow (A)	-	-	-	-	-	*	*	-	-	*
SETB dir:bp	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
SETB addr16:bp	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
SETB io:bp	3	7	0	2 × (b)	bit (io:bp)b \leftarrow 1	-	-	-	-	-	-	-	-	-	*
CLRB dir:bp	3	7	0	2 × (b)	bit (dir:bp)b \leftarrow 0	-	-	-	-	-	-	-	-	-	*
CLRB addr16:bp	4	7	0	2 × (b)	bit (addr16:bp)b \leftarrow 0	-	-	-	-	-	-	-	-	-	*
CLRB io:bp	3	7	0	2 × (b)	bit (io:bp)b \leftarrow 0	-	-	-	-	-	-	-	-	-	*
BBC dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBC io:bp,rel	4	*2	0	(b)	(io:bp) b = 0 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS dir:bp,rel	4	*1	0	(b)	(dir:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS addr16:bp,rel	5	*1	0	(b)	(addr16:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
BBS io:bp,rel	4	*2	0	(b)	(io:bp) b = 1 の場合分岐	-	-	-	-	-	-	*	-	-	-
SBBS addr16:bp,rel	5	*3	0	2 × (b)	(addr16:bp) b = 1 で分岐, bit (addr16:bp) b \leftarrow 1	-	-	-	-	-	-	*	-	-	*
WBTS io:bp	3	*4	0	*5	(io:bp) b = 1 になるまで待つ	-	-	-	-	-	-	-	-	-	-
WBTC io:bp	3	*4	0	*5	(io:bp) b = 0 になるまで待つ	-	-	-	-	-	-	-	-	-	-

*1 : 分岐が行われる場合 , 8 分岐が行われない場合 , 7

*2 : 分岐が行われる場合 , 7 分岐が行われない場合 , 6

*3 : 条件成立の場合 , 10 未成立の場合 , 9

*4 : 不定回数

*5 : 条件が成立するまで

(注意事項) 表中の (b) は , 表 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 \longleftrightarrow (A)8-15	-	-	-	-	-	-	-	-	-	-
SWAPW	1	2	0	0	word (AH) \longleftrightarrow (AL)	-	*	-	-	-	-	-	-	-	-
EXT	1	1	0	0	byte 符号拡張	X	-	-	-	-	*	*	-	-	-
EXTW	1	2	0	0	word 符号拡張	-	X	-	-	-	*	*	-	-	-
ZEXT	1	1	0	0	byte ゼロ拡張	Z	-	-	-	-	R	*	-	-	-
ZEXTW	1	1	0	0	word ゼロ拡張	-	Z	-	-	-	R	*	-	-	-

表 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。

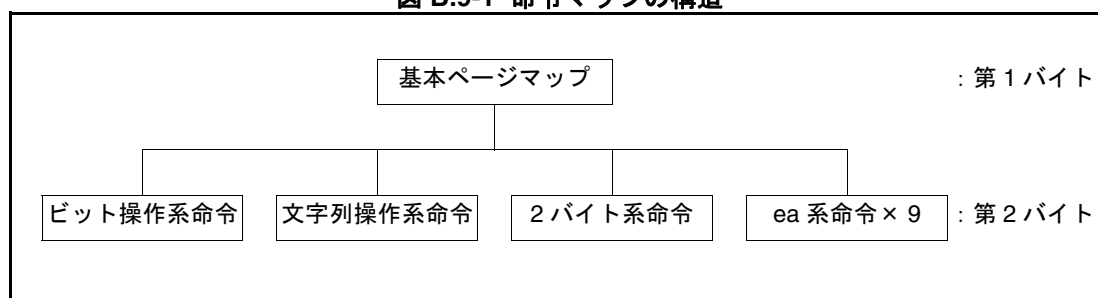
B.9 命令マップ

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

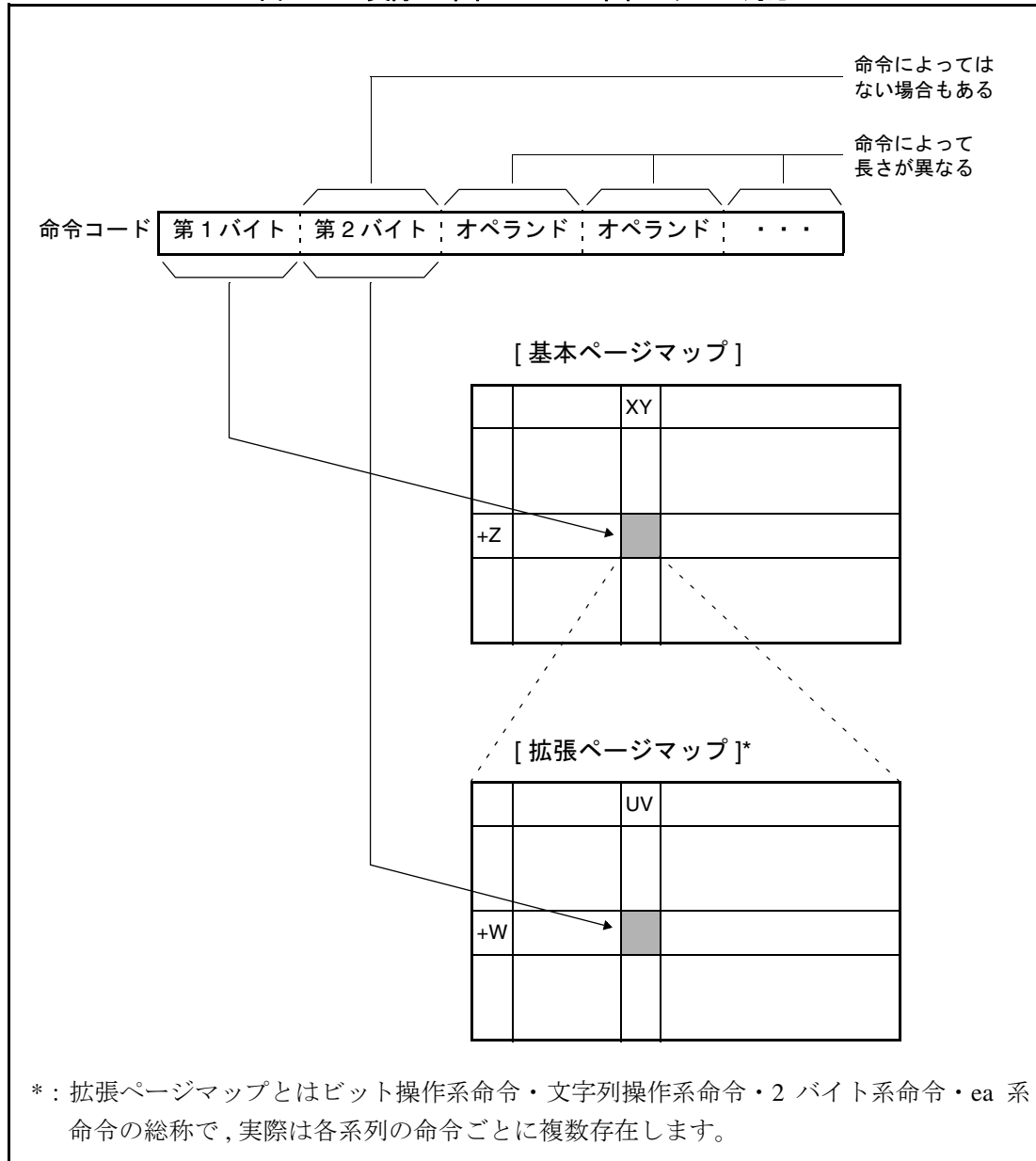
図 B.9-1 命令マップの構造



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, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	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 Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	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	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	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 Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	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	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	MOV A, #4	CALL #4	BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, Ri	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	MOV A, #4	CALL #4	BGT rel
+E	ASRW A	SWAPW	XORW A	XORW A, #16	PUSHW PS	POPW PS	文字列操 作系命令	XCH Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	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 Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	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															
+2	PCB, ADB															
+3	PCB, 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 A, @RL0+d8											
+1	MOV A, ADB	MOV ADB, A														
+2	MOV A, SSB	MOV SSB, A	MOVX A, @RL1+d8	MOV @RL1+d8, A	MOV A, @RL1+d8											
+3	MOV A, USB	MOV USB, A														
+4	MOV A, DPR	MOV DPR, A	MOVX A, @RL2+d8	MOV @RL2+d8, A	MOV A, @RL2+d8											
+5	MOV A, @A	MOV @AL, AH														
+6	MOV A, PCB	MOV A, @A	MOVX A, @RL3+d8	MOV @RL3+d8, A	MOV A, @RL3+d8											
+7	ROL A	ROL A														
+8				MOVW @RL0+d8, A	MOVW A, @RL0+d8			MUL A								
+9								MULW A								
+A				MOVW @RL1+d8, A	MOVW A, @RL1+d8			DIVU A								
+B																
+C	LSLW A, R0	LSLL A, R0	LSL A, R0	MOVW @RL2+d8, A	MOVW A, @RL2+d8											
+D	MOVW A, @A	MOVW @AL, AH	NRML A, R0													
+E	ASRW A, R0	ASRL A, R0	ASR A, R0	MOVW @RL3+d8, A	MOVW A, @RL3+d8											
+F	LSRW A, R0	LSRL A, R0	LSR A, R0													

表 B.9-6 ea 系命令 その 1 (第 1 バイト = 70_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	@RW0+R7 #8, rel	@RW0+R7 #8, rel
+D	ADDL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	SUBL A, A, @RW1+R7	@RW1+R7 #16, rel	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	CMPL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ANDL A, A, @RW1+R7	ORL A, A, @RW1+R7	ORL A, A, @RW1+R7	XORL A, A, @RW1+R7	XORL A, A, @RW1+R7	@RW1+R7 #8, rel	@RW1+R7 #8, rel
+E	ADDL A, A, @RW2+PC+d16	SUBL A, A, @RW2+PC+d16	SUBL A, A, @RW2+PC+d16	SUBL A, A, @RW2+PC+d16	@RW2+PC+d16 #16, rel	CMPL A, A, @RW2+PC+d16	CMPL A, A, @RW2+PC+d16	CMPL A, A, @RW2+PC+d16	ANDL A, A, @RW2+PC+d16	ANDL A, A, @RW2+PC+d16	ORL A, A, @RW2+PC+d16	ORL A, A, @RW2+PC+d16	XORL A, A, @RW2+PC+d16	XORL A, A, @RW2+PC+d16	@PC+d16 #8, rel	@PC+d16 #8, rel
+F	ADDL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	addr16 #16, rel	CMPL A, A, @RW3+addr16	CMPL A, A, @RW3+addr16	CMPL A, A, @RW3+addr16	ANDL A, A, @RW3+addr16	ANDL A, A, @RW3+addr16	ORL A, A, @RW3+addr16	ORL A, A, @RW3+addr16	XORL A, A, @RW3+addr16	XORL A, A, @RW3+addr16	addr16 #8, rel	addr16 #8, rel

表 B.9-7 ea 系命令 その 2 (第 1 バイト = 71_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	MOV R1, #8, @ @RW0+d8	MOVEA A, @ @RW0+d8	MOVEA A, @ @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	MOV R2, #8, @ @RW1+d8	MOVEA A, @ @RW1+d8	MOVEA A, @ @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	MOV R3, #8, @ @RW2+d8	MOVEA A, @ @RW2+d8	MOVEA A, @ @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	MOV R4, #8, @ @RW3+d8	MOVEA A, @ @RW3+d8	MOVEA A, @ @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	MOV R5, #8, @ @RW4+d8	MOVEA A, @ @RW4+d8	MOVEA A, @ @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	MOV R6, #8, @ @RW5+d8	MOVEA A, @ @RW5+d8	MOVEA A, @ @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	MOV R7, #8, @ @RW6+d8	MOVEA A, @ @RW6+d8	MOVEA A, @ @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	MOV R8, #8, @ @RW7+d8	MOVEA A, @ @RW7+d8	MOVEA A, @ @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	MOV @ @RW1, #8, @ @RW0+d16	MOVEA A, @ @RW0+d16	MOVEA A, @ @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	MOV @ @RW2, #8, @ @RW1+d16	MOVEA A, @ @RW1+d16	MOVEA A, @ @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	MOV @ @RW3, #8, @ @RW2+d16	MOVEA A, @ @RW2+d16	MOVEA A, @ @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	MOV @ @RW4, #8, @ @RW3+d16	MOVEA A, @ @RW3+d16	MOVEA A, @ @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	MOV @ @RW1+, #8, @ @RW0+RW7	MOVEA A, @ @RW0+RW7	MOVEA A, @ @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	MOV @ @RW2+, #8, @ @RW1+RW7	MOVEA A, @ @RW1+RW7	MOVEA A, @ @RW1+RW7
+E	JMPP @ @ @PC+d16	JMPP @ @ @PC+d16	CALLP @ @ @PC+d16	CALLP @ @ @PC+d16	INCL @ @PC+d16	INCL @ @PC+d16	DECL @ @PC+d16	DECL @ @PC+d16	MOVL A, @ @PC+d16	MOVL A, @ @PC+d16	MOVL @ @PC+d16, A	MOVL @ @PC+d16, A	MOV @ @PC+d16, #8	MOV @ @PC+d16, #8	MOVEA A, @ @PC+d16	MOVEA A, @ @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	MOVL @ @RW3+, A, @ @addr16	MOV @ @RW3+, #8, @ @addr16	MOV @ @RW3+, #8, @ @addr16	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	ROLc	R0i @RW0+d8	RORc	R0c @RW0+d8	INC	R0i @RW0+d8	DEC	R0i @RW0+d8	MOV	A, R0i @RW0+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+1	ROLc	R1i @RW1+d8	RORc	R1c @RW1+d8	INC	R1i @RW1+d8	DEC	R1i @RW1+d8	MOV	A, R1i @RW1+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+2	ROLc	R2i @RW2+d8	RORc	R2c @RW2+d8	INC	R2i @RW2+d8	DEC	R2i @RW2+d8	MOV	A, R2i @RW2+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+3	ROLc	R3i @RW3+d8	RORc	R3c @RW3+d8	INC	R3i @RW3+d8	DEC	R3i @RW3+d8	MOV	A, R3i @RW3+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+4	ROLc	R4i @RW4+d8	RORc	R4c @RW4+d8	INC	R4i @RW4+d8	DEC	R4i @RW4+d8	MOV	A, R4i @RW4+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+5	ROLc	R5i @RW5+d8	RORc	R5c @RW5+d8	INC	R5i @RW5+d8	DEC	R5i @RW5+d8	MOV	A, R5i @RW5+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+6	ROLc	R6i @RW6+d8	RORc	R6c @RW6+d8	INC	R6i @RW6+d8	DEC	R6i @RW6+d8	MOV	A, R6i @RW6+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+7	ROLc	R7i @RW7+d8	RORc	R7c @RW7+d8	INC	R7i @RW7+d8	DEC	R7i @RW7+d8	MOV	A, R7i @RW7+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+8	ROLc	R0i @RW0+d16	RORc	R0c @RW0+d16	INC	R0i @RW0+d16	DEC	R0i @RW0+d16	MOV	A, R0i @RW0+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+9	ROLc	R1i @RW1+d16	RORc	R1c @RW1+d16	INC	R1i @RW1+d16	DEC	R1i @RW1+d16	MOV	A, R1i @RW1+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+A	ROLc	R2i @RW2+d16	RORc	R2c @RW2+d16	INC	R2i @RW2+d16	DEC	R2i @RW2+d16	MOV	A, R2i @RW2+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+B	ROLc	R3i @RW3+d16	RORc	R3c @RW3+d16	INC	R3i @RW3+d16	DEC	R3i @RW3+d16	MOV	A, R3i @RW3+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+C	ROLc	R0i @RW0+RW7	RORc	R0c @RW0+RW7	INC	R0i @RW0+RW7	DEC	R0i @RW0+RW7	MOV	A, R0i @RW0+RW7	MOV	MOV	MOVX	MOVX	XCH	XCH
+D	ROLc	R1i @RW1+RW7	RORc	R1c @RW1+RW7	INC	R1i @RW1+RW7	DEC	R1i @RW1+RW7	MOV	A, R1i @RW1+RW7	MOV	MOV	MOVX	MOVX	XCH	XCH
+E	ROLc	R2i @RW2+PC+d16	RORc	R2c @RW2+PC+d16	INC	R2i @RW2+PC+d16	DEC	R2i @RW2+PC+d16	MOV	A, R2i @RW2+PC+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+F	ROLc	R3i @RW3+addr16	RORc	R3c @RW3+addr16	INC	R3i @RW3+addr16	DEC	R3i @RW3+addr16	MOV	A, R3i @RW3+addr16	MOV	MOV	MOVX	MOVX	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, A', @RW0+d8,A	MOVW RW0, A', @RW0+d8,A	MOVW RW0, #16', @RW0+d8,#16	MOVW RW0, #16', @RW0+d8,#16	XCHW A, RW0', @RW0+d8	XCHW A, RW0', @RW0+d8
+1	JMP @RW1', @@RW1+d8	JMP @RW1', @@RW1+d8	CALL RW1', @@RW1+d8	CALL RW1', @@RW1+d8	INCW RW1', @@RW1+d8	INCW RW1', @@RW1+d8	DECW RW1', @@RW1+d8	DECW RW1', @@RW1+d8	MOVW A, RW1', @@RW1+d8	MOVW A, RW1', @@RW1+d8	MOVW RW1, A', @RW1+d8,A	MOVW RW1, A', @RW1+d8,A	MOVW RW1, #16', @RW1+d8,#16	MOVW RW1, #16', @RW1+d8,#16	XCHW A, RW1', @RW1+d8	XCHW A, RW1', @RW1+d8
+2	JMP @RW2', @@RW2+d8	JMP @RW2', @@RW2+d8	CALL RW2', @@RW2+d8	CALL RW2', @@RW2+d8	INCW RW2', @@RW2+d8	INCW RW2', @@RW2+d8	DECW RW2', @@RW2+d8	DECW RW2', @@RW2+d8	MOVW A, RW2', @@RW2+d8	MOVW A, RW2', @@RW2+d8	MOVW RW2, A', @RW2+d8,A	MOVW RW2, A', @RW2+d8,A	MOVW RW2, #16', @RW2+d8,#16	MOVW RW2, #16', @RW2+d8,#16	XCHW A, RW2', @RW2+d8	XCHW A, RW2', @RW2+d8
+3	JMP @RW3', @@RW3+d8	JMP @RW3', @@RW3+d8	CALL RW3', @@RW3+d8	CALL RW3', @@RW3+d8	INCW RW3', @@RW3+d8	INCW RW3', @@RW3+d8	DECW RW3', @@RW3+d8	DECW RW3', @@RW3+d8	MOVW A, RW3', @@RW3+d8	MOVW A, RW3', @@RW3+d8	MOVW RW3, A', @RW3+d8,A	MOVW RW3, A', @RW3+d8,A	MOVW RW3, #16', @RW3+d8,#16	MOVW RW3, #16', @RW3+d8,#16	XCHW A, RW3', @RW3+d8	XCHW A, RW3', @RW3+d8
+4	JMP @RW4', @@RW4+d8	JMP @RW4', @@RW4+d8	CALL RW4', @@RW4+d8	CALL RW4', @@RW4+d8	INCW RW4', @@RW4+d8	INCW RW4', @@RW4+d8	DECW RW4', @@RW4+d8	DECW RW4', @@RW4+d8	MOVW A, RW4', @@RW4+d8	MOVW A, RW4', @@RW4+d8	MOVW RW4, A', @RW4+d8,A	MOVW RW4, A', @RW4+d8,A	MOVW RW4, #16', @RW4+d8,#16	MOVW RW4, #16', @RW4+d8,#16	XCHW A, RW4', @RW4+d8	XCHW A, RW4', @RW4+d8
+5	JMP @RW5', @@RW5+d8	JMP @RW5', @@RW5+d8	CALL RW5', @@RW5+d8	CALL RW5', @@RW5+d8	INCW RW5', @@RW5+d8	INCW RW5', @@RW5+d8	DECW RW5', @@RW5+d8	DECW RW5', @@RW5+d8	MOVW A, RW5', @@RW5+d8	MOVW A, RW5', @@RW5+d8	MOVW RW5, A', @RW5+d8,A	MOVW RW5, A', @RW5+d8,A	MOVW RW5, #16', @RW5+d8,#16	MOVW RW5, #16', @RW5+d8,#16	XCHW A, RW5', @RW5+d8	XCHW A, RW5', @RW5+d8
+6	JMP @RW6', @@RW6+d8	JMP @RW6', @@RW6+d8	CALL RW6', @@RW6+d8	CALL RW6', @@RW6+d8	INCW RW6', @@RW6+d8	INCW RW6', @@RW6+d8	DECW RW6', @@RW6+d8	DECW RW6', @@RW6+d8	MOVW A, RW6', @@RW6+d8	MOVW A, RW6', @@RW6+d8	MOVW RW6, A', @RW6+d8,A	MOVW RW6, A', @RW6+d8,A	MOVW RW6, #16', @RW6+d8,#16	MOVW RW6, #16', @RW6+d8,#16	XCHW A, RW6', @RW6+d8	XCHW A, RW6', @RW6+d8
+7	JMP @RW7', @@RW7+d8	JMP @RW7', @@RW7+d8	CALL RW7', @@RW7+d8	CALL RW7', @@RW7+d8	INCW RW7', @@RW7+d8	INCW RW7', @@RW7+d8	DECW RW7', @@RW7+d8	DECW RW7', @@RW7+d8	MOVW A, RW7', @@RW7+d8	MOVW A, RW7', @@RW7+d8	MOVW RW7, A', @RW7+d8,A	MOVW RW7, A', @RW7+d8,A	MOVW RW7, #16', @RW7+d8,#16	MOVW RW7, #16', @RW7+d8,#16	XCHW A, RW7', @RW7+d8	XCHW A, RW7', @RW7+d8
+8	JMP @RW0', @RW0+d16	JMP @RW0', @RW0+d16	CALL @RW0', @RW0+d16	CALL @RW0', @RW0+d16	INCW @RW0', @RW0+d16	INCW @RW0', @RW0+d16	DECW @RW0', @RW0+d16	DECW @RW0', @RW0+d16	MOVW A, @RW0', @RW0+d16	MOVW A, @RW0', @RW0+d16	MOVW @RW0,A', @RW0+d16,A	MOVW @RW0,A', @RW0+d16,A	MOVW @RW0, #16', @RW0+d16,#16	MOVW @RW0, #16', @RW0+d16,#16	XCHW A, @RW0', @RW0+d16	XCHW A, @RW0', @RW0+d16
+9	JMP @RW1', @RW1+d16	JMP @RW1', @RW1+d16	CALL @RW1', @RW1+d16	CALL @RW1', @RW1+d16	INCW @RW1', @RW1+d16	INCW @RW1', @RW1+d16	DECW @RW1', @RW1+d16	DECW @RW1', @RW1+d16	MOVW A, @RW1', @RW1+d16	MOVW A, @RW1', @RW1+d16	MOVW @RW1, A', @RW1+d16,A	MOVW @RW1, A', @RW1+d16,A	MOVW @RW1, #16', @RW1+d16,#16	MOVW @RW1, #16', @RW1+d16,#16	XCHW A, @RW1', @RW1+d16	XCHW A, @RW1', @RW1+d16
+A	JMP @RW2', @RW2+d16	JMP @RW2', @RW2+d16	CALL @RW2', @RW2+d16	CALL @RW2', @RW2+d16	INCW @RW2', @RW2+d16	INCW @RW2', @RW2+d16	DECW @RW2', @RW2+d16	DECW @RW2', @RW2+d16	MOVW A, @RW2', @RW2+d16	MOVW A, @RW2', @RW2+d16	MOVW @RW2, A', @RW2+d16,A	MOVW @RW2, A', @RW2+d16,A	MOVW @RW2, #16', @RW2+d16,#16	MOVW @RW2, #16', @RW2+d16,#16	XCHW A, @RW2', @RW2+d16	XCHW A, @RW2', @RW2+d16
+B	JMP @RW3', @RW3+d16	JMP @RW3', @RW3+d16	CALL @RW3', @RW3+d16	CALL @RW3', @RW3+d16	INCW @RW3', @RW3+d16	INCW @RW3', @RW3+d16	DECW @RW3', @RW3+d16	DECW @RW3', @RW3+d16	MOVW A, @RW3', @RW3+d16	MOVW A, @RW3', @RW3+d16	MOVW @RW3, A', @RW3+d16,A	MOVW @RW3, A', @RW3+d16,A	MOVW @RW3, #16', @RW3+d16,#16	MOVW @RW3, #16', @RW3+d16,#16	XCHW A, @RW3', @RW3+d16	XCHW A, @RW3', @RW3+d16
+C	JMP @RW0+, @RW0+RW7	JMP @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	CALL @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	INCW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	DECW @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW A, @RW0+, @RW0+RW7	MOVW @RW0+, A', @RW0+RW7	MOVW @RW0+, A', @RW0+RW7	MOVW @RW0+, #16', @RW0+RW7	MOVW @RW0+, #16', @RW0+RW7	XCHW A, @RW0+, @RW0+RW7	XCHW A, @RW0+, @RW0+RW7
+D	JMP @RW1+, @RW1+RW7	JMP @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	CALL @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	INCW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	DECW @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW A, @RW1+, @RW1+RW7	MOVW @RW1+, A', @RW1+RW7	MOVW @RW1+, A', @RW1+RW7	MOVW @RW1+, #16', @RW1+RW7	MOVW @RW1+, #16', @RW1+RW7	XCHW A, @RW1+, @RW1+RW7	XCHW A, @RW1+, @RW1+RW7
+E	JMP @RW2+, @PC+d16	JMP @RW2+, @PC+d16	CALL @RW2+, @PC+d16	CALL @RW2+, @PC+d16	INCW @RW2+, @PC+d16	INCW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	DECW @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW A, @RW2+, @PC+d16	MOVW @RW2+, A', @PC+d16,A	MOVW @RW2+, A', @PC+d16,A	MOVW @RW2+, #16', @PC+d16,#16	MOVW @RW2+, #16', @PC+d16,#16	XCHW A, @RW2+, @PC+d16	XCHW A, @RW2+, @PC+d16
+F	JMP @RW3+, @addr16	JMP @RW3+, @addr16	CALL @RW3+, @addr16	CALL @RW3+, @addr16	INCW @RW3+, @addr16	INCW @RW3+, @addr16	DECW @RW3+, @addr16	DECW @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW A, @RW3+, @addr16	MOVW @RW3+, A', @addr16,A	MOVW @RW3+, A', @addr16,A	MOVW @RW3+, #16', @addr16,#16	MOVW @RW3+, #16', @addr16,#16	XCHW A, @RW3+, @addr16	XCHW A, @RW3+, @addr16

表 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	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	CMP A, R0, @RW0+d8	CMP A, R0, @RW0+d8	AND A, R0, @RW0+d8	AND A, R0, @RW0+d8	OR A, R0, @RW0+d8	OR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	DBNZ R0, @RW0+d8, r	DBNZ @RW0+d8, r
+1	ADD A, R1, @RW1+d8	SUB A, R1, @RW1+d8	SUB A, R1, @RW1+d8	SUB A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	CMP A, R1, @RW1+d8	CMP A, R1, @RW1+d8	AND A, R1, @RW1+d8	AND A, R1, @RW1+d8	OR A, R1, @RW1+d8	OR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	DBNZ R1, @RW1+d8, r	DBNZ @RW1+d8, r
+2	ADD A, R2, @RW2+d8	SUB A, R2, @RW2+d8	SUB A, R2, @RW2+d8	SUB A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	CMP A, R2, @RW2+d8	CMP A, R2, @RW2+d8	AND A, R2, @RW2+d8	AND A, R2, @RW2+d8	OR A, R2, @RW2+d8	OR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	DBNZ R2, @RW2+d8, r	DBNZ @RW2+d8, r
+3	ADD A, R3, @RW3+d8	SUB A, R3, @RW3+d8	SUB A, R3, @RW3+d8	SUB A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	CMP A, R3, @RW3+d8	CMP A, R3, @RW3+d8	AND A, R3, @RW3+d8	AND A, R3, @RW3+d8	OR A, R3, @RW3+d8	OR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	DBNZ R3, @RW3+d8, r	DBNZ @RW3+d8, r
+4	ADD A, R4, @RW4+d8	SUB A, R4, @RW4+d8	SUB A, R4, @RW4+d8	SUB A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	CMP A, R4, @RW4+d8	CMP A, R4, @RW4+d8	AND A, R4, @RW4+d8	AND A, R4, @RW4+d8	OR A, R4, @RW4+d8	OR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	DBNZ R4, @RW4+d8, r	DBNZ @RW4+d8, r
+5	ADD A, R5, @RW5+d8	SUB A, R5, @RW5+d8	SUB A, R5, @RW5+d8	SUB A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	CMP A, R5, @RW5+d8	CMP A, R5, @RW5+d8	AND A, R5, @RW5+d8	AND A, R5, @RW5+d8	OR A, R5, @RW5+d8	OR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	DBNZ R5, @RW5+d8, r	DBNZ @RW5+d8, r
+6	ADD A, R6, @RW6+d8	SUB A, R6, @RW6+d8	SUB A, R6, @RW6+d8	SUB A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	CMP A, R6, @RW6+d8	CMP A, R6, @RW6+d8	AND A, R6, @RW6+d8	AND A, R6, @RW6+d8	OR A, R6, @RW6+d8	OR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	DBNZ R6, @RW6+d8, r	DBNZ @RW6+d8, r
+7	ADD A, R7, @RW7+d8	SUB A, R7, @RW7+d8	SUB A, R7, @RW7+d8	SUB A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	CMP A, R7, @RW7+d8	CMP A, R7, @RW7+d8	AND A, R7, @RW7+d8	AND A, R7, @RW7+d8	OR A, R7, @RW7+d8	OR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	DBNZ R7, @RW7+d8, r	DBNZ @RW7+d8, r
+8	ADD A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	DBNZ @RW0, @RW0+d16, r	DBNZ @RW0+d16, r
+9	ADD A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	DBNZ @RW1, @RW1+d16, r	DBNZ @RW1+d16, r
+A	ADD A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	DBNZ @RW2, @RW2+d16, r	DBNZ @RW2+d16, r
+B	ADD A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	DBNZ @RW3, @RW3+d16, r	DBNZ @RW3+d16, r
+C	ADD A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	DBNZ @RW0+, @RW0+RW7, r	DBNZ @RW0+RW7, r
+D	ADD A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	DBNZ @RW1+, @RW1+RW7, r	DBNZ @RW1+RW7, r
+E	ADD A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	DBNZ @RW2+, @PC+d16, r	DBNZ @RW2+PC+d16, r
+F	ADD A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	DBNZ @RW3+, A, addr16, r	DBNZ @RW3+, A, 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, R01 @RW0+d8	MULU A, R01 @RW0+d8	MULUW A, RW01 @RW0+d8	MULUW A, RW01 @RW0+d8	MUL A, R01 @RW0+d8	MUL A, R01 @RW0+d8	MULW A, RW01 @RW0+d8	MULW A, RW01 @RW0+d8	DIVU A, R01 @RW0+d8	DIVU A, R01 @RW0+d8	DIVUW A, RW01 @RW0+d8	DIVUW A, RW01 @RW0+d8	DIV A, R01 @RW0+d8	DIV A, R01 @RW0+d8	DIVW A, RW01 @RW0+d8	DIVW A, RW01 @RW0+d8
+1	MULU A, R11 @RW1+d8	MULU A, R11 @RW1+d8	MULUW A, RW11 @RW1+d8	MULUW A, RW11 @RW1+d8	MUL A, R11 @RW1+d8	MUL A, R11 @RW1+d8	MULW A, RW11 @RW1+d8	MULW A, RW11 @RW1+d8	DIVU A, R11 @RW1+d8	DIVU A, R11 @RW1+d8	DIVUW A, RW11 @RW1+d8	DIVUW A, RW11 @RW1+d8	DIV A, R11 @RW1+d8	DIV A, R11 @RW1+d8	DIVW A, RW11 @RW1+d8	DIVW A, RW11 @RW1+d8
+2	MULU A, R21 @RW2+d8	MULU A, R21 @RW2+d8	MULUW A, RW21 @RW2+d8	MULUW A, RW21 @RW2+d8	MUL A, R21 @RW2+d8	MUL A, R21 @RW2+d8	MULW A, RW21 @RW2+d8	MULW A, RW21 @RW2+d8	DIVU A, R21 @RW2+d8	DIVU A, R21 @RW2+d8	DIVUW A, RW21 @RW2+d8	DIVUW A, RW21 @RW2+d8	DIV A, R21 @RW2+d8	DIV A, R21 @RW2+d8	DIVW A, RW21 @RW2+d8	DIVW A, RW21 @RW2+d8
+3	MULU A, R31 @RW3+d8	MULU A, R31 @RW3+d8	MULUW A, RW31 @RW3+d8	MULUW A, RW31 @RW3+d8	MUL A, R31 @RW3+d8	MUL A, R31 @RW3+d8	MULW A, RW31 @RW3+d8	MULW A, RW31 @RW3+d8	DIVU A, R31 @RW3+d8	DIVU A, R31 @RW3+d8	DIVUW A, RW31 @RW3+d8	DIVUW A, RW31 @RW3+d8	DIV A, R31 @RW3+d8	DIV A, R31 @RW3+d8	DIVW A, RW31 @RW3+d8	DIVW A, RW31 @RW3+d8
+4	MULU A, R41 @RW4+d8	MULU A, R41 @RW4+d8	MULUW A, RW41 @RW4+d8	MULUW A, RW41 @RW4+d8	MUL A, R41 @RW4+d8	MUL A, R41 @RW4+d8	MULW A, RW41 @RW4+d8	MULW A, RW41 @RW4+d8	DIVU A, R41 @RW4+d8	DIVU A, R41 @RW4+d8	DIVUW A, RW41 @RW4+d8	DIVUW A, RW41 @RW4+d8	DIV A, R41 @RW4+d8	DIV A, R41 @RW4+d8	DIVW A, RW41 @RW4+d8	DIVW A, RW41 @RW4+d8
+5	MULU A, R51 @RW5+d8	MULU A, R51 @RW5+d8	MULUW A, RW51 @RW5+d8	MULUW A, RW51 @RW5+d8	MUL A, R51 @RW5+d8	MUL A, R51 @RW5+d8	MULW A, RW51 @RW5+d8	MULW A, RW51 @RW5+d8	DIVU A, R51 @RW5+d8	DIVU A, R51 @RW5+d8	DIVUW A, RW51 @RW5+d8	DIVUW A, RW51 @RW5+d8	DIV A, R51 @RW5+d8	DIV A, R51 @RW5+d8	DIVW A, RW51 @RW5+d8	DIVW A, RW51 @RW5+d8
+6	MULU A, R61 @RW6+d8	MULU A, R61 @RW6+d8	MULUW A, RW61 @RW6+d8	MULUW A, RW61 @RW6+d8	MUL A, R61 @RW6+d8	MUL A, R61 @RW6+d8	MULW A, RW61 @RW6+d8	MULW A, RW61 @RW6+d8	DIVU A, R61 @RW6+d8	DIVU A, R61 @RW6+d8	DIVUW A, RW61 @RW6+d8	DIVUW A, RW61 @RW6+d8	DIV A, R61 @RW6+d8	DIV A, R61 @RW6+d8	DIVW A, RW61 @RW6+d8	DIVW A, RW61 @RW6+d8
+7	MULU A, R71 @RW7+d8	MULU A, R71 @RW7+d8	MULUW A, RW71 @RW7+d8	MULUW A, RW71 @RW7+d8	MUL A, R71 @RW7+d8	MUL A, R71 @RW7+d8	MULW A, RW71 @RW7+d8	MULW A, RW71 @RW7+d8	DIVU A, R71 @RW7+d8	DIVU A, R71 @RW7+d8	DIVUW A, RW71 @RW7+d8	DIVUW A, RW71 @RW7+d8	DIV A, R71 @RW7+d8	DIV A, R71 @RW7+d8	DIVW A, RW71 @RW7+d8	DIVW A, RW71 @RW7+d8
+8	MULU A, @RW0 @RW0+d16	MULU A, @RW0 @RW0+d16	MULUW A, @RW01 @RW0+d16	MULUW A, @RW01 @RW0+d16	MUL A, @RW01 @RW0+d16	MUL A, @RW01 @RW0+d16	MULW A, @RW01 @RW0+d16	MULW A, @RW01 @RW0+d16	DIVU A, @RW01 @RW0+d16	DIVU A, @RW01 @RW0+d16	DIVUW A, @RW01 @RW0+d16	DIVUW A, @RW01 @RW0+d16	DIV A, @RW01 @RW0+d16	DIV A, @RW01 @RW0+d16	DIVW A, @RW01 @RW0+d16	DIVW A, @RW01 @RW0+d16
+9	MULU A, @RW1 @RW1+d16	MULU A, @RW1 @RW1+d16	MULUW A, @RW11 @RW1+d16	MULUW A, @RW11 @RW1+d16	MUL A, @RW11 @RW1+d16	MUL A, @RW11 @RW1+d16	MULW A, @RW11 @RW1+d16	MULW A, @RW11 @RW1+d16	DIVU A, @RW11 @RW1+d16	DIVU A, @RW11 @RW1+d16	DIVUW A, @RW11 @RW1+d16	DIVUW A, @RW11 @RW1+d16	DIV A, @RW11 @RW1+d16	DIV A, @RW11 @RW1+d16	DIVW A, @RW11 @RW1+d16	DIVW A, @RW11 @RW1+d16
+A	MULU A, @RW2 @RW2+d16	MULU A, @RW2 @RW2+d16	MULUW A, @RW21 @RW2+d16	MULUW A, @RW21 @RW2+d16	MUL A, @RW21 @RW2+d16	MUL A, @RW21 @RW2+d16	MULW A, @RW21 @RW2+d16	MULW A, @RW21 @RW2+d16	DIVU A, @RW21 @RW2+d16	DIVU A, @RW21 @RW2+d16	DIVUW A, @RW21 @RW2+d16	DIVUW A, @RW21 @RW2+d16	DIV A, @RW21 @RW2+d16	DIV A, @RW21 @RW2+d16	DIVW A, @RW21 @RW2+d16	DIVW A, @RW21 @RW2+d16
+B	MULU A, @RW3 @RW3+d16	MULU A, @RW3 @RW3+d16	MULUW A, @RW31 @RW3+d16	MULUW A, @RW31 @RW3+d16	MUL A, @RW31 @RW3+d16	MUL A, @RW31 @RW3+d16	MULW A, @RW31 @RW3+d16	MULW A, @RW31 @RW3+d16	DIVU A, @RW31 @RW3+d16	DIVU A, @RW31 @RW3+d16	DIVUW A, @RW31 @RW3+d16	DIVUW A, @RW31 @RW3+d16	DIV A, @RW31 @RW3+d16	DIV A, @RW31 @RW3+d16	DIVW A, @RW31 @RW3+d16	DIVW A, @RW31 @RW3+d16
+C	MULU A, @RW0+ @RW0+RW7	MULU A, @RW0+ @RW0+RW7	MULUW A, @RW0+1 @RW0+RW7	MULUW A, @RW0+1 @RW0+RW7	MUL A, @RW0+1 @RW0+RW7	MUL A, @RW0+1 @RW0+RW7	MULW A, @RW0+1 @RW0+RW7	MULW A, @RW0+1 @RW0+RW7	DIVU A, @RW0+1 @RW0+RW7	DIVU A, @RW0+1 @RW0+RW7	DIVUW A, @RW0+1 @RW0+RW7	DIVUW A, @RW0+1 @RW0+RW7	DIV A, @RW0+1 @RW0+RW7	DIV A, @RW0+1 @RW0+RW7	DIVW A, @RW0+1 @RW0+RW7	DIVW A, @RW0+1 @RW0+RW7
+D	MULU A, @RW1+ @RW1+RW7	MULU A, @RW1+ @RW1+RW7	MULUW A, @RW1+1 @RW1+RW7	MULUW A, @RW1+1 @RW1+RW7	MUL A, @RW1+1 @RW1+RW7	MUL A, @RW1+1 @RW1+RW7	MULW A, @RW1+1 @RW1+RW7	MULW A, @RW1+1 @RW1+RW7	DIVU A, @RW1+1 @RW1+RW7	DIVU A, @RW1+1 @RW1+RW7	DIVUW A, @RW1+1 @RW1+RW7	DIVUW A, @RW1+1 @RW1+RW7	DIV A, @RW1+1 @RW1+RW7	DIV A, @RW1+1 @RW1+RW7	DIVW A, @RW1+1 @RW1+RW7	DIVW A, @RW1+1 @RW1+RW7
+E	MULU A, @RW2+ @PC+d16	MULU A, @RW2+ @PC+d16	MULUW A, @RW2+1 @PC+d16	MULUW A, @RW2+1 @PC+d16	MUL A, @RW2+1 @PC+d16	MUL A, @RW2+1 @PC+d16	MULW A, @RW2+1 @PC+d16	MULW A, @RW2+1 @PC+d16	DIVU A, @RW2+1 @PC+d16	DIVU A, @RW2+1 @PC+d16	DIVUW A, @RW2+1 @PC+d16	DIVUW A, @RW2+1 @PC+d16	DIV A, @RW2+1 @PC+d16	DIV A, @RW2+1 @PC+d16	DIVW A, @RW2+1 @PC+d16	DIVW A, @RW2+1 @PC+d16
+F	MULU A, @RW3+ addr16	MULU A, @RW3+ addr16	MULUW A, @RW3+1 addr16	MULUW A, @RW3+1 addr16	MUL A, @RW3+1 addr16	MUL A, @RW3+1 addr16	MULW A, @RW3+1 addr16	MULW A, @RW3+1 addr16	DIVU A, @RW3+1 addr16	DIVU A, @RW3+1 addr16	DIVUW A, @RW3+1 addr16	DIVUW A, @RW3+1 addr16	DIV A, @RW3+1 addr16	DIV A, @RW3+1 addr16	DIVW A, @RW3+1 addr16	DIVW A, @RW3+1 addr16

表 B.9-16 MOV Ri, ea 命令 (第1バイト = 7A_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0 @RW0+d8	MOV R0, R0 @RW0+d8	MOV R1, R0 @RW0+d8	MOV R1, R0 @RW0+d8	MOV R2, R0 @RW0+d8	MOV R3, R0 @RW0+d8	MOV R3, R0 @RW0+d8	MOV R3, R0 @RW0+d8	MOV R4, R0 @RW0+d8	MOV R4, R0 @RW0+d8	MOV R5, R0 @RW0+d8	MOV R5, R0 @RW0+d8	MOV R6, R0 @RW0+d8	MOV R6, R0 @RW0+d8	MOV R7, R0 @RW0+d8	MOV R7, R0 @RW0+d8
+1	MOV R0, R1 @RW1+d8	MOV R0, R1 @RW1+d8	MOV R1, R1 @RW1+d8	MOV R1, R1 @RW1+d8	MOV R2, R1 @RW1+d8	MOV R3, R1 @RW1+d8	MOV R3, R1 @RW1+d8	MOV R3, R1 @RW1+d8	MOV R4, R1 @RW1+d8	MOV R4, R1 @RW1+d8	MOV R5, R1 @RW1+d8	MOV R5, R1 @RW1+d8	MOV R6, R1 @RW1+d8	MOV R6, R1 @RW1+d8	MOV R7, R1 @RW1+d8	MOV R7, R1 @RW1+d8
+2	MOV R0, R2 @RW2+d8	MOV R0, R2 @RW2+d8	MOV R1, R2 @RW2+d8	MOV R1, R2 @RW2+d8	MOV R2, R2 @RW2+d8	MOV R3, R2 @RW2+d8	MOV R3, R2 @RW2+d8	MOV R3, R2 @RW2+d8	MOV R4, R2 @RW2+d8	MOV R4, R2 @RW2+d8	MOV R5, R2 @RW2+d8	MOV R5, R2 @RW2+d8	MOV R6, R2 @RW2+d8	MOV R6, R2 @RW2+d8	MOV R7, R2 @RW2+d8	MOV R7, R2 @RW2+d8
+3	MOV R0, R3 @RW3+d8	MOV R0, R3 @RW3+d8	MOV R1, R3 @RW3+d8	MOV R1, R3 @RW3+d8	MOV R2, R3 @RW3+d8	MOV R3, R3 @RW3+d8	MOV R3, R3 @RW3+d8	MOV R3, R3 @RW3+d8	MOV R4, R3 @RW3+d8	MOV R4, R3 @RW3+d8	MOV R5, R3 @RW3+d8	MOV R5, R3 @RW3+d8	MOV R6, R3 @RW3+d8	MOV R6, R3 @RW3+d8	MOV R7, R3 @RW3+d8	MOV R7, R3 @RW3+d8
+4	MOV R0, R4 @RW4+d8	MOV R0, R4 @RW4+d8	MOV R1, R4 @RW4+d8	MOV R1, R4 @RW4+d8	MOV R2, R4 @RW4+d8	MOV R3, R4 @RW4+d8	MOV R3, R4 @RW4+d8	MOV R3, R4 @RW4+d8	MOV R4, R4 @RW4+d8	MOV R4, R4 @RW4+d8	MOV R5, R4 @RW4+d8	MOV R5, R4 @RW4+d8	MOV R6, R4 @RW4+d8	MOV R6, R4 @RW4+d8	MOV R7, R4 @RW4+d8	MOV R7, R4 @RW4+d8
+5	MOV R0, R5 @RW5+d8	MOV R0, R5 @RW5+d8	MOV R1, R5 @RW5+d8	MOV R1, R5 @RW5+d8	MOV R2, R5 @RW5+d8	MOV R3, R5 @RW5+d8	MOV R3, R5 @RW5+d8	MOV R3, R5 @RW5+d8	MOV R4, R5 @RW5+d8	MOV R4, R5 @RW5+d8	MOV R5, R5 @RW5+d8	MOV R5, R5 @RW5+d8	MOV R6, R5 @RW5+d8	MOV R6, R5 @RW5+d8	MOV R7, R5 @RW5+d8	MOV R7, R5 @RW5+d8
+6	MOV R0, R6 @RW6+d8	MOV R0, R6 @RW6+d8	MOV R1, R6 @RW6+d8	MOV R1, R6 @RW6+d8	MOV R2, R6 @RW6+d8	MOV R3, R6 @RW6+d8	MOV R3, R6 @RW6+d8	MOV R3, R6 @RW6+d8	MOV R4, R6 @RW6+d8	MOV R4, R6 @RW6+d8	MOV R5, R6 @RW6+d8	MOV R5, R6 @RW6+d8	MOV R6, R6 @RW6+d8	MOV R6, R6 @RW6+d8	MOV R7, R6 @RW6+d8	MOV R7, R6 @RW6+d8
+7	MOV R0, R7 @RW7+d8	MOV R0, R7 @RW7+d8	MOV R1, R7 @RW7+d8	MOV R1, R7 @RW7+d8	MOV R2, R7 @RW7+d8	MOV R3, R7 @RW7+d8	MOV R3, R7 @RW7+d8	MOV R3, R7 @RW7+d8	MOV R4, R7 @RW7+d8	MOV R4, R7 @RW7+d8	MOV R5, R7 @RW7+d8	MOV R5, R7 @RW7+d8	MOV R6, R7 @RW7+d8	MOV R6, R7 @RW7+d8	MOV R7, R7 @RW7+d8	MOV R7, R7 @RW7+d8
+8	MOV R0, @RW0	MOV R0, @RW0+d16	MOV R1, @RW0	MOV R1, @RW0+d16	MOV R2, @RW0	MOV R3, @RW0+d16	MOV R3, @RW0	MOV R3, @RW0+d16	MOV R4, @RW0	MOV R4, @RW0+d16	MOV R5, @RW0	MOV R5, @RW0+d16	MOV R6, @RW0	MOV R6, @RW0+d16	MOV R7, @RW0	MOV R7, @RW0+d16
+9	MOV R0, @RW1	MOV R0, @RW1+d16	MOV R1, @RW1	MOV R1, @RW1+d16	MOV R2, @RW1	MOV R3, @RW1+d16	MOV R3, @RW1	MOV R3, @RW1+d16	MOV R4, @RW1	MOV R4, @RW1+d16	MOV R5, @RW1	MOV R5, @RW1+d16	MOV R6, @RW1	MOV R6, @RW1+d16	MOV R7, @RW1	MOV R7, @RW1+d16
+A	MOV R0, @RW2	MOV R0, @RW2+d16	MOV R1, @RW2	MOV R1, @RW2+d16	MOV R2, @RW2	MOV R3, @RW2+d16	MOV R3, @RW2	MOV R3, @RW2+d16	MOV R4, @RW2	MOV R4, @RW2+d16	MOV R5, @RW2	MOV R5, @RW2+d16	MOV R6, @RW2	MOV R6, @RW2+d16	MOV R7, @RW2	MOV R7, @RW2+d16
+B	MOV R0, @RW3	MOV R0, @RW3+d16	MOV R1, @RW3	MOV R1, @RW3+d16	MOV R2, @RW3	MOV R3, @RW3+d16	MOV R3, @RW3	MOV R3, @RW3+d16	MOV R4, @RW3	MOV R4, @RW3+d16	MOV R5, @RW3	MOV R5, @RW3+d16	MOV R6, @RW3	MOV R6, @RW3+d16	MOV R7, @RW3	MOV R7, @RW3+d16
+C	MOV R0, @RW0+	MOV R0, @RW0+RW7	MOV R1, @RW0+	MOV R1, @RW0+RW7	MOV R2, @RW0+	MOV R3, @RW0+RW7	MOV R3, @RW0+	MOV R3, @RW0+RW7	MOV R4, @RW0+	MOV R4, @RW0+RW7	MOV R5, @RW0+	MOV R5, @RW0+RW7	MOV R6, @RW0+	MOV R6, @RW0+RW7	MOV R7, @RW0+	MOV R7, @RW0+RW7
+D	MOV R0, @RW1+	MOV R0, @RW1+RW7	MOV R1, @RW1+	MOV R1, @RW1+RW7	MOV R2, @RW1+	MOV R3, @RW1+RW7	MOV R3, @RW1+	MOV R3, @RW1+RW7	MOV R4, @RW1+	MOV R4, @RW1+RW7	MOV R5, @RW1+	MOV R5, @RW1+RW7	MOV R6, @RW1+	MOV R6, @RW1+RW7	MOV R7, @RW1+	MOV R7, @RW1+RW7
+E	MOV R0, @RW2+	MOV R0, @PC+d16	MOV R1, @RW2+	MOV R1, @PC+d16	MOV R2, @RW2+	MOV R3, @PC+d16	MOV R3, @RW2+	MOV R3, @PC+d16	MOV R4, @RW2+	MOV R4, @PC+d16	MOV R5, @RW2+	MOV R5, @PC+d16	MOV R6, @RW2+	MOV R6, @PC+d16	MOV R7, @RW2+	MOV R7, @PC+d16
+F	MOV R0, @RW3+	addr16	MOV R1, @RW3+	addr16	MOV R2, @RW3+	MOV R3, addr16	MOV R3, @RW3+	MOV R3, addr16	MOV R4, @RW3+	MOV R4, addr16	MOV R5, @RW3+	MOV R5, addr16	MOV R6, @RW3+	MOV R6, addr16	MOV R7, @RW3+	addr16

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, 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	MOV R0, R15, @RW0+d8, R15
+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)

[illegible]

表 B.9-20 XCH Ri, ea 命令 (第1バイト = 7EH)

[illegible]

索引

<数字>

1

- 16ビットタイマレジスタ
 - 16ビットタイマレジスタ (TMR0/TMR1) ... 229
- 16ビットフリーランタイマ
 - 16ビットフリーランタイマのカウント値のカウントタイミング..... 285
 - 16ビットフリーランタイマのカウント値のクリアタイミング..... 285
 - 16ビットフリーランタイマの動作..... 284
- 16ビットフリーランタイマ部
 - 16ビットフリーランタイマ部のレジスタ一覧..... 254
- 16ビットリロードタイマ
 - 16ビットリロードタイマ使用上の注意.... 242
 - 16ビットリロードタイマの端子..... 223
 - 16ビットリロードタイマの端子のブロックダイアグラム..... 223
 - 16ビットリロードタイマの動作モード.... 218
 - 16ビットリロードタイマのブロックダイアグラム..... 221
 - 16ビットリロードタイマのレジスタ一覧.. 224
 - 16ビットリロードタイマの割込み..... 231
- 16ビットリロードタイマの設定
 - 16ビットリロードタイマの設定..... 232
- 16ビットリロードタイマのEI²OS機能
 - 16ビットリロードタイマのEI²OS機能..... 231
- 16ビットリロードタイマ
 - 16ビットリロードタイマの割込みとEI²OS . 220
- 16ビットリロードレジスタ
 - 16ビットリロードレジスタ (TMRLR0/TMRHR0, TMRLR1/TMRHR1) .. 230

5

- 512kビットフラッシュメモリ
 - 512kビットフラッシュメモリの特長..... 418

8

- 8/10ビットA/Dコンバータ
 - 8/10ビットA/Dコンバータの機能..... 376
- 8/10ビットA/Dコンバータ
 - 8/10ビットA/Dコンバータ使用上の注意..... 396
 - 8/10ビットA/DコンバータのEI²OS機能.. 388
 - 8/10ビットA/Dコンバータの端子..... 379
 - 8/10ビットA/Dコンバータの端子部のブロックダイアグラム..... 380

- 8/10ビットA/Dコンバータのレジスタ一覧..... 381
- 8/10ビットA/Dコンバータの割込み..... 388
- 8/10ビットA/Dコンバータの割込みとEI²OS..... 376, 388
- 8/10ビットA/Dコンバータ
 - 8/10ビットA/Dコンバータのブロックダイアグラム..... 377
- 8/16ビットPPGタイマ
 - 8/16ビットPPGタイマのレジスタ一覧..... 255
- 8/16ビットPPGタイマ部
 - 8/16ビットPPGタイマ部のブロックダイアグラム..... 252
- 8/16ビットPPGタイマ
 - 8/16ビットPPGタイマの動作..... 289
- 8ビットタイマコントロールレジスタ
 - 8ビットタイマコントロールレジスタ (DTCR0~DTCR2) 278
- 8ビットリロードレジスタ
 - 8ビットリロードレジスタ (TMRRO~TMRR2) 280

<アルファベット>

A**A/D制御ステータスレジスタ**

A/D制御ステータスレジスタ (ADCS1) 382

A/D制御ステータスレジスタ

A/D制御ステータスレジスタ (ADCS0) . . . 384

A/Dデータレジスタ

A/Dデータレジスタ (ADCR0/ADCR1) 386

A/D変換データ保護機能

A/D変換データ保護機能. 393

ADCR

A/Dデータレジスタ (ADCR0/ADCR1) 386

ADCS0

A/D制御ステータスレジスタ (ADCS0) . . . 384

ADCS1

A/D制御ステータスレジスタ (ADCS1) . . . 382

C**CKSCR**

クロック選択レジスタ (CKSCR) 80

CPU

CPUの概要. 24

CPU間欠動作モード

CPU間欠動作モード. 89, 95

CPU動作モード

CPU動作モードと消費電流. 88

D**DIP-64P-M01**

端子配列図. 9

DIP-64P-M01パッケージ外形寸法図. 12

DTP

DTP／外部割込み回路使用上の注意. 364

DTP／外部割込み回路の設定. 359

DTP／外部割込み回路の端子. 349

DTP／外部割込み回路の端子部のブロックダイヤグラム. 349

DTP／外部割込み回路のブロックダイヤグラム

. 347

DTP／外部割込み回路のレジスタ一覧. . . . 350

DTP／外部割込み回路の割込みとEI²OS . . . 346

DTP／外部割込み動作. 360

DTP／外部割込み回路

DTP／外部割込み回路の機能. 346

DTP／割込み許可レジスタ

DTP／割込み許可レジスタ (ENIR) 354

DTP／割込み要因レジスタ

DTP／割込み要因レジスタ (EIRR) 351

DTP機能

DTP機能の動作説明. 363

DTP機能のプログラム例. 367

DTTI端子入力制御

DTTI端子入力制御の動作. 301

E**EI²OSを使用した変換動作**EI²OSを使用した変換動作. 392**EIRR**

DTP／割込み要因レジスタ (EIRR) 351

ELVR

要求レベル設定レジスタ (ELVR) 357

ENIR

DTP／割込み許可レジスタ (ENIR) 354

F**F²MC-16LX**F²MC-16LX命令一覧表 489**FPT-64P-M06**

FPT-64P-M06端子配列図. 7

パッケージ外形寸法図. 10

FPT-64P-M09

FPT-64P-M09端子配列図. 8

FPT-64P-M09パッケージ外形寸法図. 11

G**GATE信号**8ビットタイマによりPPG出力およびGATE信号
を生成する場合. 300リアルタイムアウトプット (RT0～RT5) により
PPG出力およびGATE信号を生成する場合

. 299

I**I/Oマップ**

I/Oマップ. 458

I/O領域

I/O領域. 26

I/Oレジスタアドレスポインタレジスタ

I/Oレジスタアドレスポインタレジスタ (IOA)

. 135

P**PLL**

PLLクロック通倍率の選択. 82

メインクロックモードとPLLクロックモード 82

PPG

PPG動作. 290

PPGクロック制御レジスタ

PPGクロック制御レジスタ

(PCS01, PCS23, PCS45) 276

PPGコントロールレジスタ下位	
PPGコントロールレジスタ下位	
(PPGC0, PPGC2, PPGC4)	274
PPGコントロールレジスタ上位	
PPGコントロールレジスタ上位	
(PPGC1, PPGC3, PPGC5)	272
PPG出力	
8ビットタイマによりPPG出力およびGATE信号	
を生成する場合.....	300
リアルタイムアウトプット (RT0~RT5) により	
PPG出力およびGATE信号を生成する場合	
.....	299
PPGリロードレジスタ	
PPGリロードレジスタへのライトタイミング	
.....	293
PPGリロードレジスタ下位	
PPGリロードレジスタ下位 (PRL0~PRL5)	
.....	277
PPGリロードレジスタ上位	
PPGリロードレジスタ上位 (PRLH0~PRLH5)	
.....	277
R	
RAM領域	
RAM領域.....	26
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.....	414
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)	415
ROM領域	
ROM領域.....	26
RUNモード	
RUNモード.....	150
U	
UART	
UART使用上の注意.....	344
UARTの端子のブロックダイアグラム.....	308
UARTに関連する割込みとEI ² OS.....	305
UARTの端子.....	308
UARTの動作.....	331
UARTのブロックダイアグラム.....	306
UARTのレジスタ一覧.....	309
UARTの割込み.....	319
UARTの割込みとEI ² OS.....	320
UARTボーレート設定.....	324
UARTのEI ² OS機能	
UARTのEI ² OS機能.....	320
UARTの機能	
UARTの機能.....	304

<五十音>

あ

アウトプットコンペア	
アウトプットコンペアのタイミング.....	287
アウトプットコンペアの動作.....	286
アウトプットコンペア	
アウトプットコンペアのレジスタ一覧.....	254
アウトプットデータレジスタ	
アウトプットデータレジスタ (SODR0/SODR1)	
.....	317
アキュムレータ	
アキュムレータ (A)	38
アクセス空間	
バンクレジスタとアクセス空間.....	31
アドレス	
アドレス指定方法.....	431
アドレス一致検出機能	
アドレス一致検出機能の使用例.....	409
アドレス一致検出機能の動作.....	408
アドレス一致検出機能のブロックダイアグラ	
ム.....	404
アドレス一致検出機能のレジスタ一覧....	405
アドレッシング	
32ビットレジスタ間接指定によるアドレッシ	
ング.....	30
アドレッシング.....	466
間接アドレッシング	475
直接アドレッシング	468

い

イベントカウントモード	
イベントカウントモード.....	240
イベントカウントモードのプログラム例..	244
イベントカウントモード	
イベントカウントモード (外部クロックモ	
ード)	219
インターバルタイマ機能	
インターバルタイマ機能.....	196
インターバルタイマ機能 (タイムベースタイ	
マ) の動作.....	202
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)	
.....	47
インプットキャプチャ	
インプットキャプチャの動作.....	288
16ビットインプットキャプチャの取込みタイ	
ミング.....	287
インプットキャプチャコントロールレジスタ01	
インプットキャプチャコントロールレジスタ	
01 (ICS01)	268
インプットキャプチャコントロールレジスタ23	
インプットキャプチャコントロールレジスタ	
23 (ICS23)	270

インプットキャプチャデータレジスタ

インプットキャプチャデータレジスタ
(IPCP0~IPCP3) 267

インプットデータレジスタ

インプットデータレジスタ (SIDR0/SIDR1)
..... 316

インプットキャプチャ

インプットキャプチャのレジスタ一覧... 254

う**ウォッチドッグタイマ**

ウォッチドッグタイマ機能..... 208
ウォッチドッグタイマ使用上の注意..... 215
ウォッチドッグタイマの動作..... 213
ウォッチドッグタイマのプログラム例... 216
ウォッチドッグタイマのブロックダイアグラ
ム..... 209

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)
..... 211

お**オペランド指定**

24ビットオペランド指定によるリニアアド
レッシング..... 30

か**外部クロック**

外部クロックによるボーレートの設定... 330
振動子と外部クロックの接続..... 85

外部クロックモード

イベントカウントモード (外部クロックモー
ド) 219

外部リセット端子

外部リセット端子のブロックダイアグラム. 67

外部割込み

DTD/外部割込み回路のレジスタ一覧..... 350
DTP/外部割込み回路使用上の注意..... 364
DTP/外部割込み回路の設定..... 359
DTP/外部割込み回路の端子..... 349
DTP/外部割込み回路の端子部のブロックダイ
ヤグラム..... 349
DTP/外部割込み回路のブロックダイアグラム
..... 347
DTP/外部割込み回路の割込みとEI²OS... 346
DTP/外部割込み動作..... 360
外部割込み機能..... 362
外部割込み機能のプログラム例..... 366
スタンバイモードの外部割込みによる解除
..... 105

カウンタ動作

カウンタ動作..... 219

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 (EI²OS)
..... 132
拡張インテリジェントI/Oサービス (EI²OS) 処
理時間 (1回の転送時間) 140
拡張インテリジェントI/Oサービス (EI²OS) ス
テータスレジスタ (ISCS) 136
拡張インテリジェントI/Oサービス (EI²OS) ディ
スクリプタ (ISD) の構成..... 134
拡張インテリジェントI/Oサービス (EI²OS) の処
理手順..... 138
拡張インテリジェントI/Oサービス (EI²OS) の
設定手順..... 139
拡張インテリジェントI/Oサービス (EI²OS) の
動作..... 133
拡張インテリジェントI/Oサービス (EI²OS) のプ
ログラム例..... 146

間接アドレッシング

間接アドレッシング..... 475

く**クロック**

PLLクロック通倍率の選択..... 82
クロック供給マップ..... 77
クロック選択レジスタ (CKSCR) 80
クロックの概要..... 76
クロック発生部のブロックダイアグラム... 78
クロックモードの移行..... 82
振動子と外部クロックの接続..... 85
マシニングクロック..... 82
メインクロックモードとPLLクロックモード
..... 82

クロック供給機能

クロック供給機能..... 196, 203

クロックモード

クロックモード..... 89

こ**コマンドシーケンス表**

コマンドシーケンス表..... 423

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)
..... 56

コンディションコードレジスタ

コンディションコードレジスタ (CCR) の構成
..... 44

コンペアクリアレジスタ

コンペアクリアレジスタ (CPCLR) 257

コンペアコントロールレジスタ下位

コンペアコントロールレジスタ下位
(OCS, OCS2, OCS4) 265

コンペアコントロールレジスタ上位

コンペアコントロールレジスタ上位
(OCS, OCS3, OCS5) 263

コンペアレジスタ	
コンペアレジスタ (OCCP0～OCCP5)	262

し

システムスタックポインタ	
システムスタックポインタ (SSP)	42
実効アドレス	
実効アドレスフィールド	484
実効アドレスフィールド	
実効アドレスフィールド	467
実行サイクル数	
実行サイクル数	482
実行サイクル数計算方法	482
周辺機能	
内蔵周辺機能 (リソース)	3
使用上の注意	
「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意	60
ウォッチドッグタイマ使用上の注意	215
タイムベースタイマ使用上の注意	205
遅延割込み発生モジュールの使用上の注意	373
状態遷移	
状態遷移図	103
消費電流	
CPU動作モードと消費電流	88
シリアルオンボード書込み	
富士通標準シリアルオンボード書込み	444
シリアル書込み	
シリアル書込み接続例 (ユーザ電源使用時)	447
シリアル書込み接続例 (ライタ電源から供給時)	449
シングルチップモード	
シングルチップモードの端子状態	104
振動子	
振動子と外部クロックの接続	85

す

スタック	
スタック上の多バイトデータの構成	34
スタックの設定	41
スタック動作	
割込み処理開始時のスタック動作	143
割込み処理復帰時のスタック動作	143
スタック領域	
スタック領域	144
スタンバイモード	
スタンバイモード	89
スタンバイモードでの動作状態	96
スタンバイモードの外部割込みによる解除	105
スタンバイモードの優先順位	94
スタンバイモードの割込みによる解除	105

スタンバイモードへの遷移と割込み	105
------------------------	-----

ステータスレジスタ	
ステータスレジスタ (SSR0/SSR1)	314
ストップモード	
ストップモードの解除	102
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	97

せ

制御レジスタ	
制御レジスタ (SCR0/SCR1)	310
セクタ	
セクタ指定方法	434
セクタ消去一時停止時	428
セクタ消去動作時	428
複数のセクタを指定する時の注意	434
セクタ構成	
セクタ構成	419
セクタ消去	
書込み時とチップ消去/セクタ消去時	426, 427
セクタ消去一時停止時	425, 426
セクタ消去手順	434
セクタ消去を一時停止する	436
セクタ消去を再開する	437
チップ消去/セクタ消去動作時	425
データ消去を行う (セクタ消去)	434
セクタ消去タイマフラグ	
セクタ消去タイマフラグ (DQ3)	428
専用ボーレートジェネレータ	
専用ボーレートジェネレータによるボーレートの設定	326
専用レジスタ	
専用レジスタと汎用レジスタ	35
専用レジスタの構成	36

そ

双方向通信機能	
双方向通信機能	339
ソフトウェアプルアップ抵抗	
ソフトウェアプルアップ抵抗	104
ソフトウェア割込み	
ソフトウェア割込みからの復帰	130
ソフトウェア割込みの起動	130
ソフトウェア割込みの動作	131

た

タイマコントロールステータスレジスタ下位	
タイマコントロールステータスレジスタ下位 (TCCS)	260

タイマコントロールステータスレジスタ下位	
タイマコントロールステータスレジスタ下位	
(TMCSR0/TMCSR1 : L)	227
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上位	
(TCCS)	258
タイマコントロールステータスレジスタ上位	
(TMCSR0/TMCSR1 : H)	225
タイマデータレジスタ	
タイマデータレジスタ (TCDT)	256
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) ...	427
タイムベースタイマ	
インターバルタイマ機能 (タイムベースタイマ) の動作.....	202
タイムベースタイマ使用上の注意.....	205
タイムベースタイマの動作.....	204
タイムベースタイマのプログラム例.....	206
タイムベースタイマのブロックダイアグラム	
.....	197
タイムベースタイマの割込み.....	201
タイムベースタイマの割込みとEI ² OS	201
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)	
.....	199
タイムベースタイマモード	
タイムベースタイマモードの解除.....	100
タイムベースタイマモードへの遷移.....	99
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)	49
多機能タイマ	
多機能タイマの機能.....	248
多機能タイマの動作説明.....	283
多重割込み	
多重割込み.....	126
多バイト長オペランド	
多バイト長オペランドの場合の構成.....	33
多バイト長データ	
多バイト長データのアクセス.....	34
多バイトデータ	
スタック上の多バイトデータの構成.....	34
メモリ上の多バイトデータの格納状態.....	33
端子機能説明	
端子機能説明.....	13
端子配列図	
端子配列図.....	7, 8, 9
単発変換モード	
単発変換モード1, 単発変換モード2の動作	
.....	389
単発モード	
単発モードのEI ² OS起動プログラム例	397

ち

遅延割込み発生モジュール	
遅延割込み発生モジュールの使用上の注意	
.....	373
遅延割込み発生モジュールの動作.....	372
遅延割込み発生モジュールのブロックダイアグラム.....	370
遅延割込み要因発生/解除レジスタ	
遅延割込み要因発生/解除レジスタ (DIRR)	
.....	371
チップ消去	
書込み時とチップ消去/セクタ消去時	
.....	426, 427
チップ消去/セクタ消去動作時.....	425
データ消去を行う (チップ消去)	433
直接アドレッシング	
直接アドレッシング.....	468

つ

通信プリスケラ制御レジスタ (CDCR0/CDCR1)	
通信プリスケラ制御レジスタ	
(CDCR0/CDCR1)	318

て

停止変換モード	
停止変換モードの動作.....	391
停止モード	
停止モードのEI ² OS起動プログラム例	401
低消費電力制御回路	
低消費電力制御回路のブロックダイアグラム	
.....	90
低消費電力モード	
低消費電力モード制御レジスタ (LPMCR) ..	92
低消費電力モード制御レジスタへのアクセス	
.....	94
低消費電力モード	
低消費電力モード使用上の注意.....	105
データカウントレジスタ	
データカウントレジスタ (DCT)	135
データ書込み	
データ書込み上の注意.....	431
データポーリングフラグ	
データポーリングフラグ (DQ7)	425
デバイスの取扱い	
デバイスの取扱いに関する注意事項.....	18
デフォルト空間	
バンクアドレッシングとデフォルト空間... 32	

と

同期モード	
同期モード (動作モード2) 時の動作.....	336

動作クロック	
動作クロックの設定.....	291
動作モード	
動作モード.....	150, 289
特徴	
MB90560/565シリーズの特徴.....	2
トグルビットフラグ	
トグルビットフラグ (DQ6)	426

な

内部クロックモード	
内部クロックモード.....	218
内部クロックモード (リロードモード) の動作.....	234
内部クロックモード (ワンショットモード)	237
内部クロックモードのプログラム例.....	243
内部クロック (16ビットリロードタイマ)	
内部クロック (16ビットリロードタイマ) によるボーレートの設定.....	328

に

入出力回路形式	
入出力回路形式.....	16
入出力ポート	
入出力ポート機能.....	156
入出力ポートのプログラム例.....	194
入出力ポート	
入出力ポートのレジスタ一覧.....	158

の

ノイズキャンセル機能	
DTTI端子のノイズキャンセル機能.....	302
ノンオーバーラップ波形	
PPGタイマ (PPG1, PPG3, PPG5) のノンオーバーラップ波形を生成する場合.....	297
リアルタイムアウトプット (RT1, RT3, RT5) のノンオーバーラップ波形を生成する場合.....	295

は

ハードウェアシーケンスフラグ	
ハードウェアシーケンスフラグ.....	424
ハードウェア割込み	
ハードウェア割込み.....	120
ハードウェア割込みからの復帰.....	122
ハードウェア割込み禁止.....	121
ハードウェア割込み使用手順.....	125
ハードウェア割込み処理時間.....	128
ハードウェア割込みの起動.....	122
ハードウェア割込みのこうぞう.....	120

ハードウェア割込みの動作.....	123
波形制御レジスタ	
波形制御レジスタ (SIGCG)	281
波形生成部	
波形生成部の動作.....	294
波形生成部のブロックダイアグラム.....	253
波形生成部のレジスタ一覧.....	255
バスモード	
バスモード.....	150
バスモード設定ビット	
バスモード設定ビット.....	152
パッケージ外形寸法図	
パッケージ外形寸法図.....	10, 11, 12
発振安定待ち時間	
発振安定待ち時間.....	84, 106
発振安定待ち時間用タイマ機能.....	202
リセット要因と発振安定待ち時間.....	66
バッファアドレスポインタレジスタ	
バッファアドレスポインタレジスタ (BAP)	137
パルス	
パルスの端子出力の制御.....	292
パルス幅	
リロード値とパルス幅の関係.....	291
バンクアドレッシング	
バンクアドレッシングとデフォルト空間... 32	
リニアアドレッシングとバンクアドレッシング.....	29
バンクセレクトプリフィックス	
バンクセレクトプリフィックス (PCB, DTB, ADB, SPB)	54
バンクレジスタ	
バンクレジスタ (PCB, DTB, USB, SSB, ADB)	50
バンクレジスタとアクセス空間.....	31
汎用レジスタ	
専用レジスタと汎用レジスタ.....	35
汎用レジスタの構成.....	51
汎用レジスタ領域とレジスタバンクポインタ (RP).....	46

ひ

非同期モード	
非同期モード時の動作.....	333
品種構成	
品種構成.....	4

ふ

フラグ	
受信割込み出力とフラグセットのタイミング.....	321
送信割込み出力とフラグセットのタイミング.....	323
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)	57

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）..... 452
- フラッシュマイコンプログラマとの最小限の
接続例（ライター電源から供給時）... 454

フラッシュメモリ

- フラッシュメモリ書込み/消去の詳細説明
..... 429
- フラッシュメモリ書込み手順..... 431

フラッシュメモリコントロールステータスレジスタ

- フラッシュメモリコントロールステータスレ
ジスタ（FMCS）..... 420

プリフィックスコード

- プリフィックスコード..... 53
- プリフィックスコードと割込み／ホールド抑
止命令..... 58
- プリフィックスコードの連続..... 59

プログラム

- 512Kビット（64Kバイト）フラッシュメモリのプ
ログラム例..... 438

**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）..... 407

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
（PADRO/PADR1）..... 406

プログラムカウンタ

- プログラムカウンタ（PC）..... 48

プロセッサステータス

- プロセッサステータス（PS）の構成..... 43

ブロックダイアグラム

- 波形生成部のブロックダイアグラム..... 253
- ブロックダイアグラム..... 6

ブロックダイアグラム

- 16ビットリロードタイマの端子のブロックダ
イヤグラム..... 223
- 16ビットリロードタイマのブロックダイアグ
ラム..... 221
- 8/10ビットA/Dコンバータの端子部のブロッ
クダイアグラム..... 380
- 8/10ビットA/Dコンバータのブロックダイア
グラム..... 377
- 8/16ビットPPGタイマ部のブロックダイアグラ
ム..... 252
- DTP／外部割込み回路の端子部のブロックダイ
アグラム..... 349
- DTP／外部割込み回路のブロックダイアグラム
..... 347
- ROMミラー機能選択モジュールのブロックダイ
アグラム..... 414
- UARTの端子のブロックダイアグラム..... 308
- UARTのブロックダイアグラム..... 306
- アドレス一致検出機能のブロックダイアグラ
ム..... 404
- ウォッチドッグタイマのブロックダイアグラ
ム..... 209

外部リセット端子のブロックダイアグラム

- 67
- クロック発生部のブロックダイアグラム... 78
- タイムベースタイマのブロックダイアグラム
..... 197
- 遅延割込み発生モジュールのブロックダイア
グラム..... 370
- 低消費電力制御回路のブロックダイアグラム
..... 90
- ポート0端子のブロックダイアグラム..... 160
- ポート1端子のブロックダイアグラム..... 165
- ポート2端子のブロックダイアグラム..... 170
- ポート3端子のブロックダイアグラム..... 175
- ポート4端子のブロックダイアグラム..... 180
- ポート5端子のブロックダイアグラム..... 185
- ポート6端子のブロックダイアグラム..... 190
- リアルタイムI/O部のブロックダイアグラム
..... 251

ほ**ポート0**

- ポート0端子のブロックダイアグラム..... 160
- ポート0の構成..... 159
- ポート0の端子..... 159
- ポート0の動作..... 162
- ポート0のレジスタ..... 160
- ポート0のレジスタの機能..... 161

ポート1

- ポート1端子のブロックダイアグラム..... 165
- ポート1の構成..... 164
- ポート1の端子..... 164
- ポート1の動作..... 167
- ポート1のレジスタ..... 165
- ポート1のレジスタの機能..... 166

ポート2

- ポート2端子のブロックダイアグラム..... 170
- ポート2の構成..... 169
- ポート2の端子..... 169
- ポート2の動作..... 172
- ポート2のレジスタ..... 170
- ポート2のレジスタの機能..... 171

ポート3

- ポート3端子のブロックダイアグラム..... 175
- ポート3の構成..... 174
- ポート3の端子..... 174
- ポート3の動作..... 177
- ポート3のレジスタ..... 175
- ポート3のレジスタの機能..... 176

ポート4

- ポート4端子のブロックダイアグラム..... 180
- ポート4の構成..... 179
- ポート4の端子..... 179
- ポート4の動作..... 182
- ポート4のレジスタ..... 180
- ポート4のレジスタの機能..... 181

ポート5

ポート5端子のブロックダイアグラム.....	185
ポート5の構成.....	184
ポート5の端子.....	184
ポート5の動作.....	187
ポート5のレジスタ.....	185
ポート5のレジスタの機能.....	186

ポート6

ポート6端子のブロックダイアグラム.....	190
ポート6の構成.....	189
ポート6の端子.....	189
ポート6の動作.....	192
ポート6のレジスタ.....	190
ポート6のレジスタの機能.....	191

ま**マシクロック**

マシクロック.....	82
-------------	----

マスタ／スレーブ型通信機能

マスタ／スレーブ型通信機能.....	341
--------------------	-----

め**命令**

命令の種類.....	465
命令マップの構造.....	503

命令一覧表

F ² MC-16LX命令一覧表.....	489
----------------------------------	-----

メインクロックモード

メインクロックモードとPLLクロックモード.....	82
----------------------------	----

メモリ空間

メモリ空間.....	25
------------	----

メモリマップ

メモリマップ.....	27
-------------	----

も**モード設定**

モード設定.....	150
------------	-----

モード端子

モード端子.....	68
モード端子 (MD2～MD0).....	151
モード端子とモードデータの関係.....	153

モードデータ

モード端子とモードデータの関係.....	153
モードデータ取込み.....	68
モードデータ読出し後の端子状態.....	73

モードデータレジスタ

モードデータレジスタ.....	152
-----------------	-----

モードレジスタ

モードレジスタ (SMR0/SMR1).....	312
--------------------------	-----

ゆ**ユーザスタックポインタ**

ユーザスタックポインタ (USP).....	42
------------------------	----

ユーザ電源

シリアル書込み接続例 (ユーザ電源使用時).....	447
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).....	452

優先順位

スタンバイモードの優先順位.....	94
--------------------	----

よ**要求レベル設定レジスタ**

要求レベル設定レジスタ (ELVR).....	357
-------------------------	-----

ら**ライタ電源**

シリアル書込み接続例 (ライタ電源から供給時).....	449
フラッシュマイコンプログラムの最小限の接続例 (ライタ電源から供給時)...	454

り**リアルタイムI/O部**

リアルタイムI/O部のブロックダイアグラム.....	251
----------------------------	-----

リアルタイムアウトプット

リアルタイムアウトプット (RT1, RT3, RT5) のノンオーバーラップ波形を生成する場合.....	295
---	-----

リセット

リセットによる各端子の状態.....	73
--------------------	----

リセット動作

リセット動作の概要.....	68
----------------	----

リセット要因

リセット要因.....	64
リセット要因と発振安定待ち時間.....	66

リセット要因フラグビット

リセット要因フラグビット.....	70
リセット要因フラグビットとリセット要因の対応.....	71
リセット要因フラグビットの注意事項.....	71

リニアアドレッシング

24ビットオペランド指定によるリニアアドレッシング.....	30
リニアアドレッシングとバンクアドレッシング.....	29

リロード値

リロード値とパルス幅の関係.....	290
--------------------	-----

れ

例外処理

例外処理..... 142

レジスタ

16ビットタイマレジスタ (TMR0/TMR1) ... 229

16ビットフリーランタイマ部のレジスタ一覧
..... 25416ビットリロードレジスタ
(TMRLR0/TMRHR0, TMRLR1/TMRHR1)
..... 2308/10ビットA/Dコンバータのレジスタ一覧
..... 381

8/16ビットPPGタイマのレジスタ一覧..... 255

8ビットタイマコントロールレジスタ (DTCR0~
DTCR2) 2788ビットリロードレジスタ (TMRR0~TMRR2)
..... 280

A/D制御ステータスレジスタ (ADCS0) ... 384

A/D制御ステータスレジスタ (ADCS1) ... 382

A/Dデータレジスタ (ADCR0/ADCR1) 386

DTP/外部割込み回路のレジスタ一覧..... 350

DTP/わりこみきょかれじすた (ENIR)..... 354

DTP/割込み要因レジスタ (EIRR) 351

I/Oレジスタアドレスポインタレジスタ (IOA)
..... 135PPGクロック制御レジスタ
(PCS01, PCS23, PCS45) 276PPGコントロールレジスタ下位
(PPGC0, PPGC2, PPGC4) 274PPGコントロールレジスタ上位
(PPGC1, PPGC3, PPGC5) 272PPGリロードレジスタ上位 (PRLH0~PRLH5)
..... 277

UARTのレジスタ一覧..... 309

アウプットコンペアのレジスタ一覧..... 254

アウプットデータレジスタ (SODR0/SODR1)
..... 317

アドレス一致検出機能のレジスタ一覧..... 405

インタラプトレベルマスクレジスタ (ILM)
..... 47インプットキャプチャコントロールレジスタ
01 (ICS01) 268インプットキャプチャコントロールレジスタ
23 (ICS23) 270インプットキャプチャデータレジスタ (IPCP0
~IPCP3) 267インプットデータレジスタ (SIDR0/SIDR1)
..... 316

インプットキャプチャのレジスタ一覧..... 254

ウォッチドッグタイマ制御レジスタ (WDTC)
..... 211拡張インテリジェントI/Oサービス (EI²OS) ス
テータスレジスタ (ISCS) 136

クロック選択レジスタ (CKSCR) 80

コモンレジスタバンクプリフィックス (CMR)
..... 56コンディションコードレジスタ (CCR) の構成
..... 44

コンペアクリアレジスタ (CPCLR) 257

コンペアコントロールレジスタ下位
(OCS0, OCS2, OCS4) 265コンペアコントロールレジスタ上位
(OCS1, OCS3, OCS5) 263

コンペアレジスタ (OCCP0~OCCP5) 262

ステータスレジスタ (SSR0/SSR1) 314

制御レジスタ (SCR0/SCR1) 310

専用レジスタと汎用レジスタ..... 35

専用レジスタの構成..... 36

タイマコントロールステータスレジスタ下位
(TCCS) 260タイマコントロールステータスレジスタ下位
(TMCSR0/TMCSR1 : L) 227タイマコントロールステータスレジスタ上位
(TCCS) 258タイマコントロールステータスレジスタ上位
(TMCSR0/TMCSR1 : H) 225

タイマデータレジスタ (TCDT) 256

タイムベースタイマ制御レジスタ (TBTC)
..... 199

ダイレクトページレジスタ (DPR) 49

遅延割込み要因発生/解除レジスタ (DIRR)
..... 371通信プリスケラ制御レジスタ
(CDCR0/CDCR1) 318

低消費電力モード制御レジスタ (LPMCR) .. 92

低消費電力モード制御レジスタへのアクセス
..... 94

データカウントレジスタ (DCT) 135

入出力ポートのレジスタ一覧..... 158

波形制御レジスタ (SIGCG) 281

波形生成部のレジスタ一覧..... 255

バッファアドレスポインタレジスタ (BAP)
..... 137

バンクレジスタ (PCB, DTB, USB, SSB, ADB).... 50

汎用レジスタの構成..... 51

汎用レジスタ領域とレジスタバンクポインタ
(RP)..... 46フラッシュメモリコントロールステータスレ
ジスタ (FMCS)..... 420

フラッシュメモリのレジスタ..... 418

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) 407プログラムアドレス検出レジスタ
(PADR0/PADR1) 406

ポート0のレジスタ..... 160

ポート0のレジスタの機能..... 161

ポート1のレジスタ..... 165

ポート1のレジスタの機能..... 166

ポート2のレジスタ..... 170

ポート2のレジスタの機能..... 171

ポート3のレジスタ..... 175

ポート3のレジスタの機能..... 176

ポート4のレジスタ..... 180

- ポート4のレジスタの機能..... 181
- ポート5のレジスタ..... 185
- ポート5のレジスタの機能..... 186
- ポート6のレジスタ..... 190
- ポート6のレジスタの機能..... 191
- モードレジスタ (SMR0/SMR1) 312
- 要求レベル設定レジスタ (ELVR)..... 357
- レジスタバンク..... 52
- レジスタバンクポインタ (RP)..... 46
- 割込み制御レジスタ (ICR) と割込み要因
..... 112
- 割込み制御レジスタ (ICR) の構成..... 117
- 割込み制御レジスタ (ICR00~ICR15) 115
- 割込み制御レジスタ一覧..... 113
- 割込み制御レジスタの機能..... 114, 118
- 割込み要因と割込みベクタ・割込み制御レジ
スタ..... 111
- レジスタ間接指定**
32ビットレジスタ間接指定によるアドレッ
シング..... 30
- レジスタバンク**
レジスタバンク..... 52
- レジスタバンクポインタ**
汎用レジスタ領域とレジスタバンクポインタ
(RP)..... 46
- レジスタバンクポインタ (RP)..... 46
- 連続変換モード**
連続変換モードの動作..... 390
- 連続モード**
連続モードのEI²OS起動プログラム例..... 399
- わ**
- 割込み**
連する割込みとEI²OS 305
- UARTの割込み..... 319
- UARTの割込みとEI²OS 320
- スタンバイモードの割込みによる解除.... 105
- スタンバイモードへの遷移と割込み..... 105
- プリフィックスコードと割込み／ホールド抑
止命令..... 58
- 割込み..... 292
- 割込み動作時の処理..... 124
- 割込みの種類と機能..... 108
- 割込み処理**
割込み処理開始時のスタック動作..... 143
- 割込み処理のプログラム例..... 145
- 割込み処理復帰時のスタック動作..... 143
- 割込み制御レジスタ**
割込み制御レジスタ (ICR) と割込み要因
..... 112
- 割込み制御レジスタ (ICR) の構成..... 117
- 割込み制御レジスタ (ICR00~ICR15) 115
- 割込み制御レジスタ一覧..... 113
- 割込み制御レジスタの機能..... 114, 118
- 割込み要因と割込みベクタ・割込み制御レジ
スタ..... 111
- 割込み動作**
割込み動作..... 109
- 割込みベクタ**
割込みベクタ..... 110
- 割込み要因と割込みベクタ・割込み制御レジ
スタ..... 111
- 割込み要因**
割込み制御レジスタ (ICR) と割込み要因
..... 112
- 割込み要因と割込みベクタ・割込み制御レジ
スタ..... 111
- ワンショットモード**
内部クロックモード (ワンショットモード)
..... 237

CM44-10107-5

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

MB90560/565series

ハードウェアマニュアル

2008年7月 第5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

FUJITSU



* C M 4 4 - 1 0 1 0 7 - 4 *