



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

MB90570series

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

MB90570series

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。

MB90570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、実際にこの半導体を使って製品を設計する技術者の方を対象に、MB90570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。

そのほか、本書で登場するシステム名、製品名はそれぞれの会社もしくは組織の商標です。

なお、本文中では™マークや®マークを必ずしも明記していません。

■ 本書の全体構成

本書は、以下に示す27の章、および付録から構成されています。

第1章 概要

この章では、MB90570シリーズの特長や基本的な仕様について示します。

第2章 CPU

この章では、CPUの機能と動作について説明します。

第3章 割込み

この章では、割込みの機能と動作について説明します。

第4章 クロックとリセット

この章では、クロックとリセットの機能と動作について説明します。

第5章 低消費電力制御回路

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

第6章 低消費電力モード

この章では、低消費電力モードの機能と動作について説明します。

第7章 メモリアクセスモード

この章では、メモリアクセスモードの機能と動作について説明します。

第8章 I/Oポート

この章では、I/O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、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、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 時計タイマ

この章では、時計タイマの機能と動作について説明します。

第12章 16ビット入出力タイマ

この章では、16ビット入出力タイマの機能と動作について説明します。

第13章 8/16ビットPPG

この章では、8/16ビットPPGの機能と動作について説明します。

第14章 8/16ビットアップダウンカウンタ/タイマ

この章では、8/16ビットアップダウンカウンタ/タイマの機能と動作について説明します。

第15章 DTP/外部割込み

この章には、DTP/外部割込みの機能と動作について示します。

第16章 遅延割込み発生モジュール

この章では、遅延割込み発生モジュールの機能と動作について説明します。

第17章 A/Dコンバータ

この章では、A/Dコンバータの機能と動作について説明します。

第18章 D/Aコンバータ

この章では、D/Aコンバータの機能と動作について説明します。

第19章 UART

この章では、UARTの機能と動作について説明します。

第20章 I/O拡張シリアルインタフェース

この章では、I/O拡張シリアルインタフェースの機能と動作について説明します。

第21章 I²Cインタフェース

この章では、I²Cインタフェースの機能と概要について説明します。

第22章 チップセレクト機能

この章では、チップセレクトの機能と動作について説明しています。

第23章 クロックモニタ機能

この章では、クロックモニタの機能と動作について説明します。

第24章 アドレス一致検出機能

この章は、アドレス一致検出の機能と動作について説明します。

第25章 ROMミラー機能選択モジュール

この章では、ROMミラー機能選択モジュールの機能と動作について説明します。

第26章 2Mビットフラッシュメモリ

この章では、2M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27章 MB90F574/Aシリアル書込み接続例

この章では、横河デジタルコンピュータ株式会社製 AF220/210/110フラッシュマイコンプログラマを用いた場合の、シリアル書込みの接続例について説明します。

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	特長	2
1.2	品種構成	4
1.3	ブロックダイヤグラム	5
1.4	端子配列図	6
1.5	外形寸法図	7
1.6	端子機能説明	10
1.7	入出力回路形式	14
1.8	デバイスの取り扱いに関する注意事項	17
第2章	CPU	21
2.1	メモリ空間	22
2.2	アドレス指定	23
2.3	メモリ空間における多バイト長データの配置	25
2.4	専用レジスタ	26
2.4.1	アキュムレータ(A)	28
2.4.2	ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)	30
2.4.3	プロセッサステータス(PS)	31
2.4.4	プログラムカウンタ(PC)	34
2.4.5	ダイレクトページレジスタ(DPR)	35
2.4.6	バンクレジスタ	36
2.5	汎用レジスタ	37
2.6	プリフィックスコード	39
2.6.1	プリフィックスコードに関する制約	41
2.6.2	「DIV A,Ri」,「DIVW A,Rwi」命令の使用上の注意	43
第3章	割込み	45
3.1	割込みの概要	46
3.2	割込み要因	47
3.3	割込みベクタ	48
3.4	ハードウェア割込み	49
3.4.1	ハードウェア割込みの動作	52

3.4.2	ハードウェア割込みの動作フロー	54
3.4.3	ハードウェア割込み使用手順	55
3.5	ソフトウェア割込み	56
3.6	拡張インテリジェントI/Oサービス (EI ² OS)	58
3.6.1	割込み制御レジスタ(ICR)	60
3.6.2	拡張インテリジェントI/Oサービスディスクリプタ(ISD)	63
3.6.3	拡張インテリジェントI/Oサービスディスクリプタ(ISD)の各レジスタ	64
3.6.4	拡張インテリジェントI/Oサービス(EI ² OS)の動作	67
3.6.5	拡張インテリジェントI/Oサービス(EI ² OS)の使用手順	68
3.6.6	拡張インテリジェントI/Oサービス(EI ² OS) 処理時間	69
3.7	例外処理割込み	71
第4章	クロックとリセット	73
4.1	クロック発生部	74
4.2	クロック供給マップ	75
4.3	リセット要因	76
4.4	リセット解除後の動作	78
第5章	低消費電力制御回路	79
5.1	低消費電力制御回路の概要	80
5.2	低消費電力制御回路のブロックダイアグラム	82
5.3	低消費電力制御回路のレジスタ	83
5.3.1	低消費電力モード制御レジスタ(LPMCR)	84
5.3.2	クロック選択レジスタ(CKSCR)	86
5.4	クロック選択の状態遷移	88
第6章	低消費電力モード	91
6.1	低消費電力モード	92
6.1.1	スリープモード	94
6.1.2	擬似時計モード	95
6.1.3	時計モード	96
6.1.4	ストップモード	97
6.1.5	ハードウェアスタンバイモード	99
6.2	低消費電力モードの遷移条件	100
6.3	低消費電力モード状態遷移図	103

第7章	メモリアクセスモード	107
7.1	メモリアクセスモード	108
7.1.1	モード端子	109
7.1.2	モードデータ	110
7.1.3	バスモード別メモリ空間	111
7.2	外部メモリアクセス（外部バス端子制御回路）の概要	113
7.2.1	自動レディ機能選択レジスタ (ARSR)	115
7.2.2	外部アドレス出力制御レジスタ (HACR)	117
7.2.3	バス制御信号選択レジスタ (ECSR)	118
7.3	外部メモリアクセス制御信号の動作	121
7.3.1	レディ機能	123
7.3.2	ホールド機能	125
第8章	I/Oポート	127
8.1	I/Oポートの概要	128
8.2	I/Oポートのレジスタ一覧	129
8.2.1	ポートデータレジスタ (PDR)	130
8.2.2	ポート方向レジスタ (DDR)	132
8.2.3	出力端子レジスタ (ODR)	133
8.2.4	入力抵抗レジスタ (RDR)	134
8.2.5	アナログ入力許可レジスタ (ADER)	135
第9章	タイムベースタイマ	137
9.1	タイムベースタイマの概要	138
9.2	タイムベースタイマ制御レジスタ (TBTC)	139
9.3	タイムベースタイマの動作	141
第10章	ウォッチドッグタイマ	143
10.1	ウォッチドッグタイマの概要	144
10.2	ウォッチドッグタイマ制御レジスタ (WDTC)	145
10.3	ウォッチドッグタイマの動作	147
第11章	時計タイマ	149
11.1	時計タイマの概要	150
11.2	時計タイマ制御レジスタ (WTC)	151

11.3	時計タイマの動作	153
第12章	16ビット入出力タイマ	155
12.1	16ビット入出力タイマの概要	156
12.2	16ビット入出力タイマのブロックダイアグラム.....	158
12.3	16ビット入出力タイマのレジスタ	159
12.4	16ビットフリーランタイマ	161
12.4.1	タイマカウンタデータレジスタ(TCDT)	162
12.4.2	タイマカウンタコントロールステータスレジスタ(TCCS)	163
12.5	アウトプットコンペア	165
12.5.1	アウトプットコンペアレジスタ(OCCP0 ~ 3)	167
12.5.2	アウトプットコンペアコントロールステータスレジスタ(OCS0 ~ 3)	168
12.6	インプットキャプチャ	171
12.6.1	インプットキャプチャデータレジスタ(IPCP0,1)	172
12.6.2	インプットキャプチャコントロールステータスレジスタ(ICS01)	173
12.7	16ビットフリーランタイマの動作	175
12.8	16ビットアウトプットコンペアの動作	177
12.9	16ビットインプットキャプチャの動作	179
第13章	8/16ビットPPG	181
13.1	8/16ビットPPGの概要	182
13.2	8/16ビットPPGのブロックダイアグラム.....	183
13.3	8/16ビットPPGのレジスタ	185
13.3.1	PPG0動作モード制御レジスタ(PPGC0)	186
13.3.2	PPG1動作モード制御レジスタ(PPGC1)	188
13.3.3	PPG0,1出力端子制御レジスタ(PPGOE)	190
13.3.4	リロードレジスタ(PRL/PRLL)	192
13.4	8/16ビットPPGの動作	193
13.4.1	8/16ビットPPGの動作モード	194
13.4.2	8/16ビットPPGの出力動作	195
13.4.3	8/16ビットPPGのカウントクロックの選択	196
13.4.4	8/16ビットPPGのパルスの端子出力の制御	197
13.4.5	8/16ビットPPGのリロードレジスタへのライトタイミング	198
13.4.6	8/16ビットPPGの割込み	199

13.4.7	8 / 16ビットPPGの各ハードウェアの初期値	200
第14章	8/16ビットアップダウンカウンタ / タイマ	201
14.1	8/16ビットアップダウンカウンタ / タイマの概要	202
14.2	8/16ビットアップダウンカウンタ / タイマのブロックダイアグラム	204
14.3	8/16ビットアップダウンカウンタ / タイマのレジスタ一覧	206
14.3.1	アップダウンカウントレジスタ0,1(UDCR0,1)	207
14.3.2	リロード / コンペアレジスタ0,1(RCR0,1)	208
14.3.3	カウンタステータスレジスタ0,1(CSR0,1)	209
14.3.4	カウンタコントロールレジスタ0上位(CCRH0)	211
14.3.5	カウンタコントロールレジスタ1上位(CCRH1)	213
14.3.6	カウンタコントロールレジスタ0,1下位 (CCRL0,1)	215
14.4	8/16ビットアップダウンカウンタ / タイマのカウントモード選択	217
14.5	8/16ビットアップダウンカウンタ / タイマのリロード機能とコンペア機能	220
14.6	8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能の同時起動	222
14.7	8/16ビットアップダウンカウンタ / タイマのUDCRレジスタへのデータの書込み	224
第15章	DTP/外部割込み	227
15.1	DTP/外部割込みの概要	228
15.2	DTP/外部割込みのレジスタ	229
15.2.1	DTP/割込み許可レジスタ(ENIR)	230
15.2.2	DTP/割込み要因レジスタ(EIRR)	231
15.2.3	要求レベル設定レジスタ(ELVR)	232
15.3	DTP/外部割込みの動作	233
15.4	DTP/外部割込みの使用上の注意	236
第16章	遅延割込み発生モジュール	239
16.1	遅延割込み発生モジュールの概要	240
16.2	遅延割込み発生モジュールの動作	241
第17章	A/Dコンバータ	243
17.1	A/Dコンバータの概要	244
17.2	A/Dコンバータのレジスタ	246

17.2.1	コントロールステータスレジスタ(ADCS1,2)	247
17.2.2	データレジスタ(ADCR1,2)	251
17.3	A/Dコンバータの動作	253
17.3.1	EI ² OSを使用した変換動作	255
17.3.2	単発モード時のEI ² OSの起動例	256
17.3.3	連続モード時のEI ² OSの起動例	257
17.3.4	停止モード時のEI ² OS起動例	258
17.4	A/Dコンバータ使用上の注意	259
17.5	変換データ保護機能	260
第18章	D/Aコンバータ	261
18.1	D/Aコンバータの概要	262
18.2	D/Aコンバータのレジスタ	264
18.3	D/Aコンバータの動作	266
第19章	UART	267
19.1	UARTの概要	268
19.2	UARTのブロックダイアグラム	269
19.3	UARTのレジスタ	270
19.3.1	シリアルモードレジスタ(SMR)	271
19.3.2	シリアルコントロールレジスタ(SCR)	273
19.3.3	シリアルインプットデータレジスタ(SIDR)/シリアルアウトプットデータレジスタ(SODR)	276
19.3.4	シリアルステータスレジスタ(SSR)	277
19.3.5	通信プリスケアラコントロールレジスタ(CDCR)	280
19.4	UARTのボーレート	282
19.5	UARTの動作	284
19.5.1	非同期(調歩同期)モード	285
19.5.2	CLK同期モード	286
19.6	UARTのフラグと割込み発生要因	288
19.6.1	UARTの割込みとフラグのセットタイミング	289
19.7	UARTの応用例と使用上の注意	292

第20章	I/O拡張シリアルインタフェース	295
20.1	I/O拡張シリアルインタフェースの概要	296
20.2	I/O拡張シリアルインタフェースのレジスタ	297
20.2.1	シリアルモードコントロールステータスレジスタ (SMCS)	298
20.2.2	シリアルシフトデータレジスタ (SDR)	302
20.3	I/O拡張シリアルインタフェースの動作	303
20.3.1	シフトクロック	304
20.3.2	I/O拡張シリアルインタフェースの動作状態	305
20.3.3	シフト動作のスタート/ストップタイミング	307
20.3.4	シリアルデータの入出力のタイミング	309
20.3.5	I/O拡張シリアルインタフェースの割込み機能	310
第21章	I ² Cインタフェース	311
21.1	I ² Cインタフェースの概要	312
21.2	I ² Cインタフェースのブロックダイアグラム	313
21.3	I ² Cインタフェースのレジスタ	314
21.3.1	バスステータスレジスタ (IBSR)	315
21.3.2	バスコントロールレジスタ (IBCR)	318
21.3.3	クロックコントロールレジスタ (ICCR)	321
21.3.4	アドレスレジスタ (IADR)	323
21.3.5	データレジスタ (IDAR)	324
21.4	I ² Cインタフェースの動作	325
21.4.1	I ² Cインタフェースの動作フロー	327
第22章	チップセレクト機能	329
22.1	チップセレクト機能の概要	330
22.2	チップセレクト機能のレジスタ	331
22.3	チップセレクト機能の動作	332
22.4	チップセレクト機能のデコードアドレス空間	333
第23章	クロックモニタ機能	335
23.1	クロックモニタ機能の概要	336
23.2	クロック出力許可レジスタ (CLKR)	337

第24章	アドレス一致検出機能	339
24.1	アドレス一致検出機能の概要	340
24.2	アドレス一致検出機能のレジスタ	341
24.2.1	プログラムアドレス検出レジスタ(PADRO,1)	342
24.2.2	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)	343
24.3	アドレス一致検出機能の動作	344
24.4	アドレス一致検出機能の使用例	345
24.5	プログラムパッチ処理例とフロー	347
第25章	ROMミラー機能選択モジュール	349
25.1	ROMミラー機能選択モジュールの概要	350
25.2	ROMミラー機能選択レジスタ(ROMM)	351
第26章	2Mビットフラッシュメモリ	353
26.1	2Mビットフラッシュメモリの概要	354
26.2	2Mビットフラッシュメモリのセクタ構成.....	355
26.3	フラッシュメモリコントロールステータスレジスタ (FMCS)	356
26.4	フラッシュメモリ自動アルゴリズムの起動方法.....	358
26.5	自動アルゴリズム実行状態の確認	359
26.5.1	データポーリングフラグ(DQ7)	360
26.5.2	トグルビットフラグ(DQ6)	361
26.5.3	タイミングリミット超過フラグ(DQ5)	362
26.5.4	セクタ消去タイマフラグ(DQ3)	363
26.6	フラッシュメモリ書込み / 消去の詳細説明.....	364
26.6.1	フラッシュメモリの読出し / リセット状態	365
26.6.2	フラッシュメモリへのデータ書込み	366
26.6.3	フラッシュメモリの全データ消去(チップ消去)	368
26.6.4	フラッシュメモリの任意データ消去(セクタ消去)	369
26.6.5	フラッシュメモリのセクタ消去の一時停止	371
26.6.6	フラッシュメモリのセクタ消去の再開	372
26.7	フラッシュメモリのプログラム例	373

第27章	MB90F574/Aシリアル書込み接続例	377
27.1	MB90F574/Aシリアル書込み接続の基本構成	378
27.2	シリアル書込み接続例（ユーザ電源使用時）	381
27.3	シリアル書込み接続例（ライターから電源供給時）	383
27.4	フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用時）	385
27.5	フラッシュマイコンプログラマとの最小限の接続例（ライターから電源供給時）	387
付録		389
A	I/Oマップ	390
B	命令	394
B.1	命令の種類	395
B.2	アドレッシング	396
B.3	直接アドレッシング	398
B.4	間接アドレッシング	405
B.5	実行サイクル数	412
B.6	実効アドレスフィールド	414
B.7	命令一覧表の読み方	416
B.8	F ² MC-16LX命令一覧表	419
B.9	命令マップ	433
索引		455

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。）
394 ~ 454	「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、MB90570シリーズの概要について説明します。

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

1.1 特長

MB90570シリーズは、民生機器などの高速リアルタイム処理が要求されるプロセス制御用途向けに設計された、汎用の富士通16ビットマイクロコントローラです。

特長

MB90570シリーズの命令体系は、FMCシリーズのATアーキテクチャを継承するとともに、高級言語対応命令の追加やアドレッシングモードの拡張、乗除算命令の強化、ビット処理の充実化を図っています。さらに、32ビットアキュムレータの搭載によりロングワードデータの処理も可能となっています。本シリーズはI²CBUSインタフェースを内蔵しており、機器間の通信が簡易に実現できます。カーオーディオやVTR システムなどに適した製品です。

最小命令実行時間：62.5ns / 4MHz原発振4逓倍(PLLクロック逓倍方式)

最大メモリ空間：16Mbyte

コントローラ用途に最適化された命令体系

- 取り扱い可能なデータタイプ：ビット/バイト/ワード/ロングワード
- 標準アドレッシングモード：23種類
- 32ビットアキュムレータ採用による高精度演算の強化
- 符号付き乗除算命令，RET1命令機能強化

高級言語（C言語）/ マルチタスクに対応をとった命令体系

- システムスタックポインタの採用
- 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アドレスポインタ分あり）

実行速度の向上：4byteキュー

強力な割込み機能

- プライオリティレベルがプログラマブルに8レベル設定可能
- 外部割込み入力：8ch

データ転送機能

- インテリジェントI/O サービス
- 最大16チャンネル
- DTP要求入力：8ch(うち2ch両方向エッジ設定可：レベル検出設定不可)

内蔵ROM容量とROMタイプ

- MASK-ROM：128KB/256KB
- FLASH-ROM：256KB
- 内蔵RAM容量
 - FLASH：10KB
 - EVA：最大10KB
 - MASK-ROM：6KB/10KB

汎用ポート

- 最大97本
- 入力プルアップ抵抗設定可能：24本（ポート0,1,6）
- オープンドレイン設定可能：8本（ダイオードクランプ付き：ポート4）

I²CBUSインタフェース：1ch

チップセレクト出力：8本（アクティブレベル設定可能）

A/Dコンバータ(RC逐次比較方式)

- 分解能：8/10ビット：8ch(マルチプレクス)
- 変換時間：26.3 μs

D/Aコンバータ(R-2R方式)

- 分解能：8ビット：2ch(独立)
- セットアップ時間：12.5 μs

UART(全2ダブルバッファ方式)：2ch

I²O拡張シリアルインタフェース：3ch

8/16ビットアップダウンカウンタ：1ch(8ビット×2ch)

8/16ビットPPGタイマ：1ch(8ビット×2ch)

入出力タイマ：1ch

- 16ビットフリーランタイマ
- 16ビットインプットキャプチャ×2ch
- 16ビットアウトプットコンペア×4ch

クロック出力機能

時計タイマ：1ch

18ビットタイムベース/ウォッチドッグタイマ(18ビット)

低消費電力モード

スリープ/ストップ/ハードウェアスタンバイ/CPU間欠動作モード機能

パッケージ

- LQFP-120
 - FPT-120P-M21(リードピッチ0.5mm)：(MB90573, MB90574は無し)
 - FPT-120P-M05(リードピッチ0.4mm)：(MB90574Cは無し)
- QFP-120
 - FPT-120P-M13(リードピッチ0.5mm)

CMOSテクノロジー

1.2 品種構成

表 1.2-1に、MB90570シリーズの品種構成を示します。A,C付の品種と、A,C無しの品種の違いは、A,C付の品種が、DTP/外部割込みでch0～ch1のエッジ要求による割込みにてスタンバイモードから復帰ができる点です。

品種構成

表 1.2-1 MB90570シリーズの品種構成

	MB90V570	MB90V570A	MB90F574/A	MB90573	MB90574	MB90574C
ROM容量	-	-	256Kbyte	128Kbyte	256Kbyte	256Kbyte
RAM容量	10Kbyte	10Kbyte	10Kbyte	6Kbyte	10Kbyte	10Kbyte
備考	EVA品	EVA品	FLASH品	MASK品	MASK品	MASK品
エミュレータ専用電源*	-	無	-	-	無	-

*：エバリュエーションボッド MB2145-507をご使用いただ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(エミュレータ専用電源端子)』を参照ください。

1.3 ブロックダイアグラム

図 1.3-1に、ブロックダイアグラムを示します。

ブロックダイアグラム

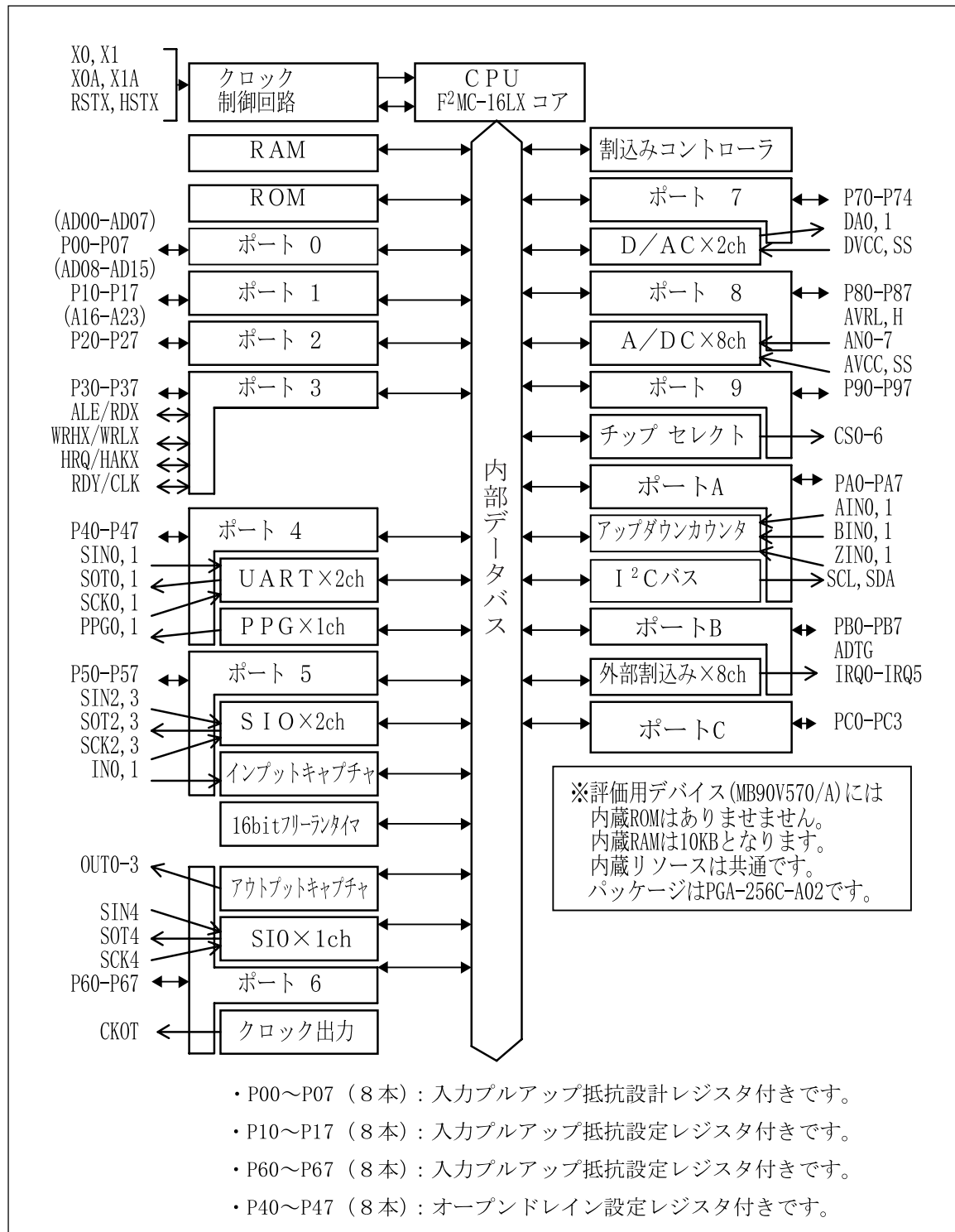


図 1.3-1 ブロックダイアグラム

1.4 端子配列図

図 1.4-1に，端子配列図を示します。

端子配列図

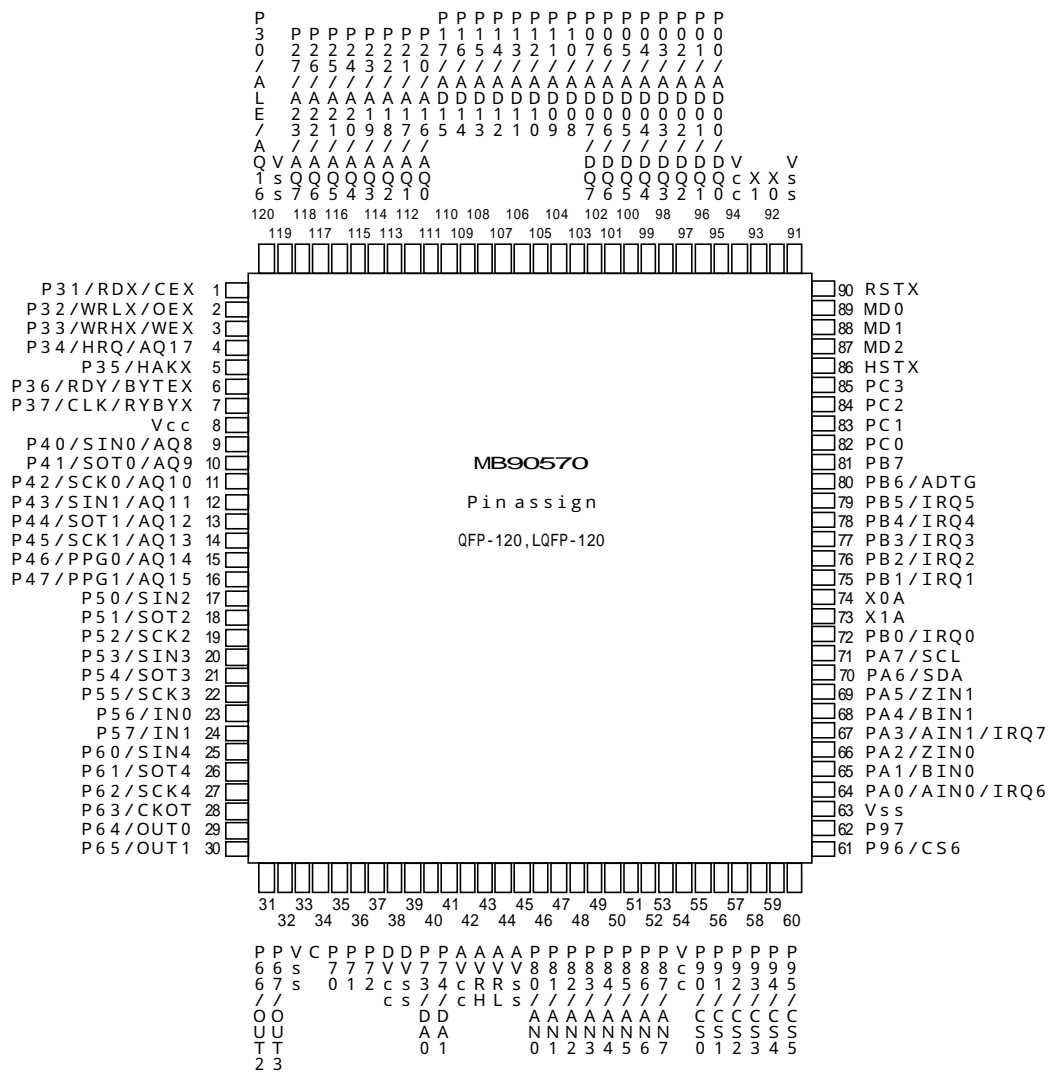


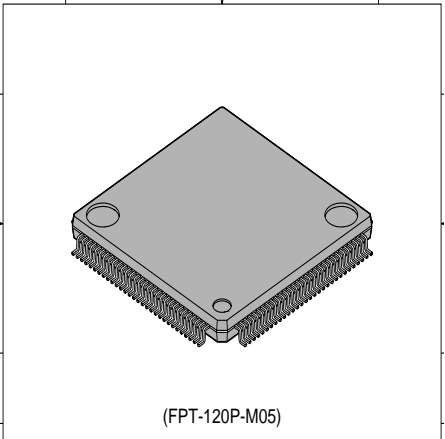
図 1.4-1 端子配列図

1.5 外形寸法図

MB90570シリーズには、3種類のパッケージが用意されています。

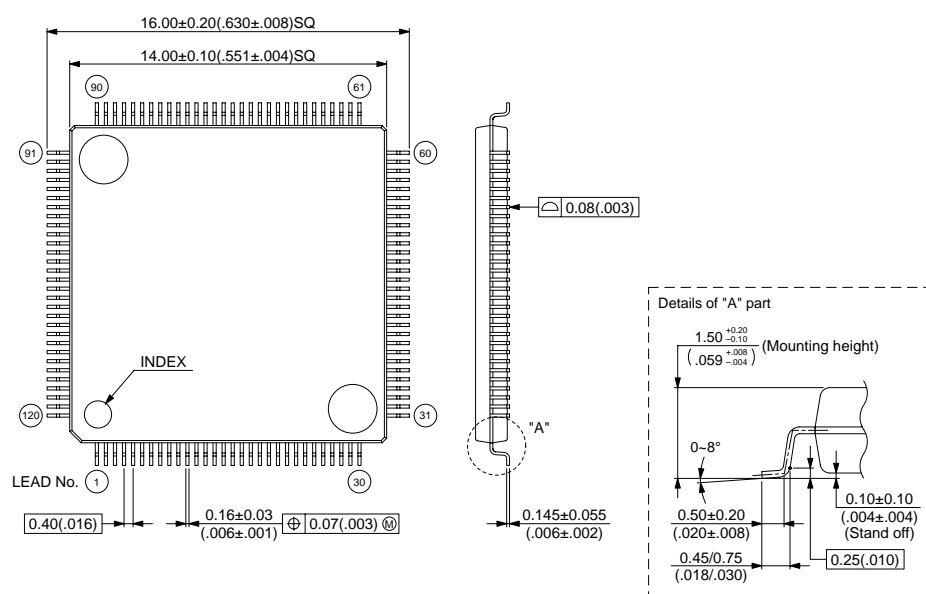
FPT-120P-M05の外形寸法図

EIAJコード：*P-LQFP-0120-1414-0.40

<p>プラスチック・LQFP、120ピン</p>  <p>(FPT-120P-M05)</p>	リードピッチ	0.40mm
	パッケージ幅× パッケージ長さ	14.0×14.0mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	1.70mm MAX
	重 さ	0.62g

プラスチック・LQFP、120ピン
(FPT-120P-M05)

* 端子幅および端子厚さはメッキ厚を含む。



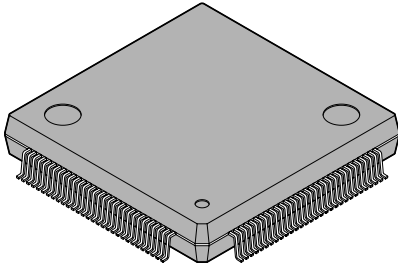
© 1998 FUJITSU LIMITED F120006S-3C-4

単位：mm(inches)

図 1.5-1 FPT-120P-M05の外形寸法図

FPT-120P-M13の外形寸法図

EIAJコード：*QFP120-P-2020-1

<p>プラスチック・QFP , 120ピン</p>  <p>(FPT-120P-M13)</p>	リードピッチ	0.50mm
	パッケージ幅× パッケージ長さ	20×20mm
	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

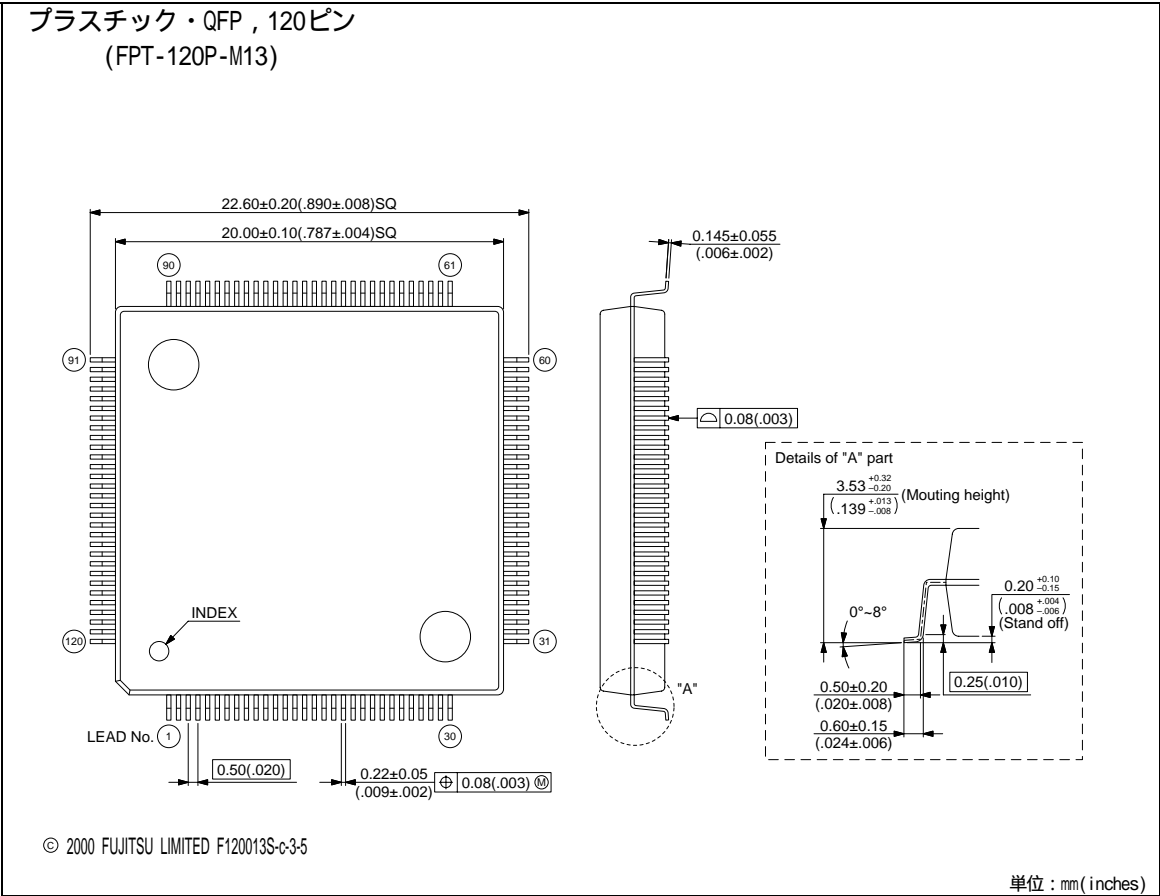
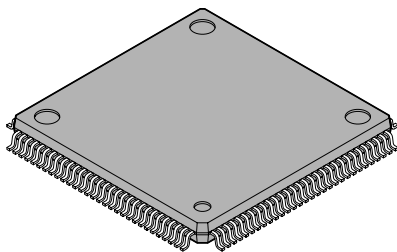
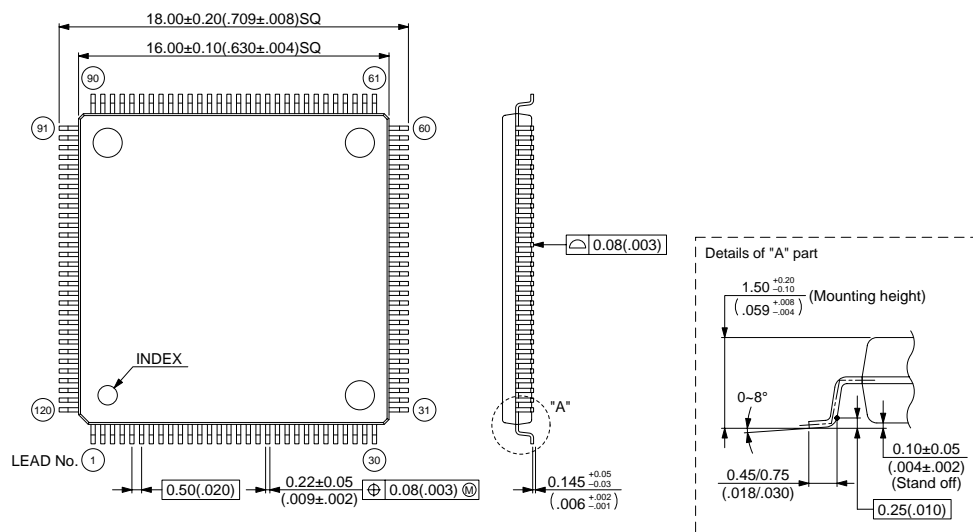


図 1.5-2 FPT-120P-M13の外形寸法図

FPT-120P-M21の外形寸法図

EIAJコード：*P-LQFP-0120-1616-0.50

<p>プラスチック・LQFP，120ピン</p>  <p>(FPT-120P-M21)</p>	リードピッチ	0.50mm
	パッケージ幅× パッケージ長さ	16.0×16.0mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	1.70mm MAX
	重さ	0.88g

プラスチック・LQFP，120ピン
(FPT-120P-M21)

© 1998 FUJITSU LIMITED F120033S-2C-2

単位：mm (inches)

図 1.5-3 FPT-120P-M21の外形寸法図

1.6 端子機能説明

表 1.6-1に，端子機能を説明します。

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	機 能 説 明
92/93	X0/X1	A	高速発振入力端子です。
74/73	X0A/X1A	B	低速発振入力端子です。
90	RSTX	C	リセット入力端子です。
86	HSTX	C	ハードウェアスタンバイ入力端子です。
95～102	P00～07	D	汎用入出力ポートです。また，入力設定時には，プルアップ抵抗設定レジスタ(RDR0)により設定できます。ただし，出力設定時は無効となります。
	AD00～07		外バスモード時はアドレス下位出力／データ下位入出力端子として機能します。
103～110	P10～17	D	汎用入出力ポートです。また，入力設定時には，プルアップ抵抗設定レジスタ(RDR1)により設定できます。ただし，出力設定時は無効となります。
	AD08～15		外バスモード時は，アドレス中位出力／データ上位入出力端子として機能します。
111～118	P20～27	E	汎用入出力ポートです。
	A16～23		外バスモード時はアドレス上位出力端子として機能します。
120	P30	E	汎用入出力ポートです。
	ALE		外バスモード時はアドレスラッチイネーブル信号出力端子となります。
1	P31	E	汎用入出力ポートです。
	RDX		外バスモード時はリードストローブ信号出力端子となります。
2	P32	E	汎用入出力ポートです。
	WRLX		外バスモード時はデータバス下位8ビットライトストローブ信号出力端子となります。
3	P33	E	汎用入出力ポートです。
	WRHX		外バスモード時はデータバス上位8ビットライトストローブ信号出力端子となります。
4	P34	E	汎用入出力ポートです。
	HRQ		外バスモード時はホールドリクエスト信号入力端子となります。
5	P35	E	汎用入出力ポートです。
	HAKX		外バスモード時はホールドアクノリッジ信号出力端子となります。
6	P36	E	汎用入出力ポートです。
	RDY		外バスモード時はレディ信号入力端子となります。
7	P37	E	汎用入出力ポートです。
	CLK		外バスモード時はクロック(CLK) 信号出力端子となります。
9	P40	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SIN0		UART Ch.0のシリアルデータ入力端子です。UART Ch.0が入力動作中には，この入力を随時使用していますので，意図的に行う以外は出力機能を使用しないでください。また，他の機能出力と兼用させる場合はSIN動作中に出力禁止としてください。

表 1.6-1 端子機能説明 (続く)

端子番号	端子名	回路形式	機 能 説 明
10	P41	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SOT0		UART Ch.0 のシリアルデータ出力端子です。UART Ch.0 がデータ出力 指定を許可しているときに有効となります。
11	P42	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SCK0		UART Ch.0 のシリアルクロック入出力端子です。UART Ch.0 がクロック出力指定を許可しているときに有効となります。
12	P43	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SIN1		UART Ch.1 のシリアルデータ入力端子です。UART Ch.1 が入力動作中には、この入力を随時使用していますので、意図的に行う以外は出力機能を使用しないでください。また他の機能出力と兼用させる場合はSIN動作中に出力禁止としてください。
13	P44	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SOT1		UART Ch.1 のシリアルデータ出力端子です。UART Ch.1 がデータ出力指定を許可しているときに有効となります。
14	P45	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	SCK1		UART Ch.1 のシリアルクロック入出力端子です。UART Ch.1 がクロック出力指定を許可しているときに有効となります。
15 ~ 16	P46 ~ 47	F	汎用入出力ポートです。ODR4レジスタによりオープンドレインに設定することができます。
	PPG0 ~ 1		PPG0,1出力端子です。この機能は、PPG0,1の出力指定が許可時に有効です。
17	P50	E	汎用入出力ポートです。
	SIN2		I/O シリアルCh.0のデータ入力端子です。シリアルデータ入力中は、この端子を随時使用しますので、意図的に行う場合以外は出力させないでください。
18	P51	E	汎用入出力ポートです。
	SOT2		I/O シリアルCh.0のデータ出力端子です。シリアルCh.0のシリアルデータ出力が許可のとき有効となります。
19	P52	E	汎用入出力ポートです。
	SCK2		I/OシリアルCh.0のシリアルクロック入出力端子です。シリアルCh.0のシリアルクロック出力が許可のとき有効となります。
20	P53	E	汎用入出力ポートです。
	SIN3		I/O シリアルCh.1のデータ入力端子です。シリアルデータ入力中は、この端子を随時使用しますので、意図的に行う場合以外は出力させないでください。
21	P54	E	汎用入出力ポートです。
	SOT3		I/O シリアルCh.1のデータ出力端子です。シリアルCh.1のシリアルデータ出力が許可のとき有効となります。
22	P55	E	汎用入出力ポートです。
	SCK3		I/O シリアルCh.1のシリアルクロック入出力端子です。シリアルCh.1のシリアルクロック出力が許可のとき有効となります。
23 ~ 24	P56 ~ 57	E	汎用入出力ポートです。
	INO ~ 1		インプットキャプチャCh.0/1のトリガ入力端子です。この機能はインプットキャプチャCh.0/1が入力動作を行っている間は随時使用していますので意図的に行う以外は出力許可しないでください。

表 1.6-1 端子機能説明 (続く)

端子番号	端子名	回路形式	機 能 説 明
25	P60	D	汎用入出力ポートです。また入力設定時にはプルアップ抵抗設定レジスタ(RDR6)により設定できます。ただし出力設定時は無効となります。
	SIN4		I/OシリアルCh.2のデータ入力端子です。シリアルデータ入力中は、この端子を随時使用しますので、意図的に行う場合以外は出力させないでください。
26	P61	D	汎用入出力ポートです。また入力設定時にはプルアップ抵抗設定レジスタ(RDR6)により設定できます。ただし出力設定時は無効となります。
	SOT4		I/OシリアルCh.2のデータ出力端子です。シリアルCh.2のシリアルデータ出力が許可のとき有効となります。
27	P62	D	汎用入出力ポートです。また入力設定時にはプルアップ抵抗設定レジスタ(RDR6)により設定できます。ただし出力設定時は無効となります。
	SCK4		I/OシリアルCh.2のシリアルクロック入出力端子です。シリアルCh.2のシリアルクロック出力が許可のとき有効となります。
28	P63	D	汎用入出力ポートです。また入力設定時にはプルアップ抵抗設定レジスタ(RDR6)により設定できます。ただし出力設定時は無効となります。
	CKOT		クロックモニタ機能出力端子です。クロックモニタ出力許可により有効になります。
29 ~ 32	P64 ~ 67	D	汎用入出力ポートです。また入力設定時にはプルアップ抵抗設定レジスタ(RDR6)により設定できます。ただし出力設定時は無効となります。
	OUT0 ~ 3		アウトプットコンペアCh.0 ~ 3イベント出力端子です。各チャンネルの出力許可状態になっているときに有効です。
34	C	G	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。ただし、FLASH 品(MB90F574/A)および、MB90574Cには不要です。
35 ~ 37	P70 ~ 72	E	汎用入出力ポートです。
38	DVcc	H	D/A コンバータVref 入力端子です。Vccを超えないようにしてください。
39	DVss	H	D/A コンバータGND 電源端子です。Vssと同電位にしてください。
40 ~ 41	P73 ~ 74	I	汎用入出力ポートです。
	DA0 ~ 1		D/A コンバータCh.0,1のアナログ信号出力端子です。
42	AVcc	H	アナログマクロ(D/A,A/D他) のVcc電源入力端子です。
43	AVRH	J	A/D コンバータVref+入力端子です。Vccを超えないようにしてください。
44	AVRL	H	A/DコンバータVref-入力端子です。Vss以下にならないようにしてください。
45	AVss	H	アナログマクロ(D/A,A/D他) のVss電源入力端子です。
46 ~ 53	P80 ~ 87	K	汎用入出力ポートです。
	AN0 ~ 7		A/Dコンバータアナログ入力端子です。この機能はアナログ入力指定が許可のときに有効になります。
55 ~ 62	P90 ~ 97	E	汎用入出力ポートです。
	CS0 ~ 7		チップセレクト出力端子です。この機能はチップセレクト出力許可にて有効になります。

表 1.6-1 端子機能説明 (続き)

端子番号	端子名	回路形式	機 能 説 明
64	PA0	E	汎用入出力ポートです。
	AIN0		8/16ビットアップダウンカウンタCh.0のカウントクロックA入力として使用できます。
	IRQ6		割込み要求入力Ch.6として使用できます。
65	PA1	E	汎用入出力ポートです。
	BIN0		8/16ビットアップダウンカウンタCh.0のカウントクロックB入力として使用できます。
66	PA2	E	汎用入出力ポートです。
	ZIN0		8/16ビットアップダウンカウンタCh.0の制御クロックZ入力として使用できます。
67	PA3	E	汎用入出力ポートです。
	AIN1		8/16ビットアップダウンカウンタCh.1のカウントクロックA入力として使用できます。
	IRQ7		割込み要求入力Ch.7として使用できます。
68	PA4	E	汎用入出力ポートです。
	BIN1		8/16ビットアップダウンカウンタCh.1のカウントクロックB入力として使用できます。
69	PA5	E	汎用入出力ポートです。
	ZIN1		8/16ビットアップダウンカウンタCh.1の制御クロックZ入力として使用できます。
70	PA6	L	汎用入出力ポートです。
	SDA		I ² Cインタフェースのデータ入出力端子です。この機能はI ² Cインタフェースが、動作許可のときに有効となります。I ² Cインタフェースが動作している間はポートを入力設定(DDRA:bit6=0)としてください。
71	PA7	L	汎用入出力ポートです。
	SCL		I ² Cインタフェースのクロック入出力端子です。この機能はI ² Cインタフェースが、動作許可のときに有効となります。I ² Cインタフェースが動作している間はポートを入力設定(DDRA:bit7=0)としてください。
72, 75~79	PB0~5	E	汎用入出力ポートです。
	IRQ0~5		外部割込み入力端子です。IRQ0,1は両エッジ検出ができますが、レベルによる割込み要因検出はできませんので、MB90V570やMB90F574, MB90573, MB90574 ではSTOPからの復帰には使用できません。ただし、MB90V570A, MB90F574A, MB90574Cについては、IRQ0,1のエッジ検出によるSTOPからの復帰が可能です。
80	PB6	E	汎用入出力ポートです。
	ADTG		A/D コンバータ外部トリガ入力端子です。A/D コンバータが入力動作を行っている場合はこの入力を随時使用していますので意図的に行う以外は出力停止としてください。
81	PB7	E	汎用入出力ポートです。
82~85	PC0~3	E	汎用入出力ポートです。
88,89	MD1,MD0	C	動作モード指定用入力端子です。VccまたはVssに直接(直結)してください。
87	MD2	C	動作モード指定用入力端子です。VccまたはVssに直接(直結)してください。
8,54,94	Vcc	-	電源(5V)入力端子です。
33,63,91 119	Vss	-	電源(0V)入力端子です。

1.7 入出力回路形式

表 1.7-1に，入出力回路形式を説明します。

入出力回路形式

表 1.7-1 入出力回路形式（続く）

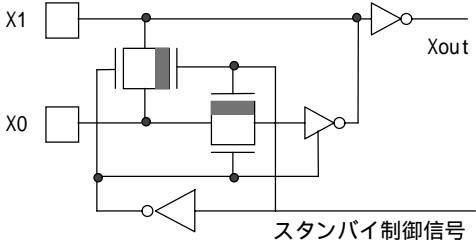
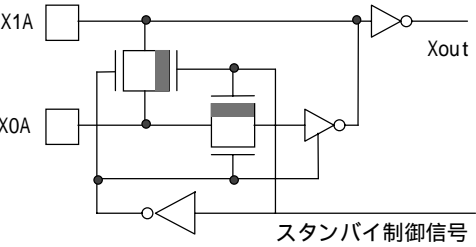
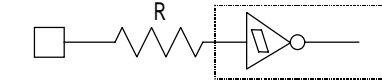
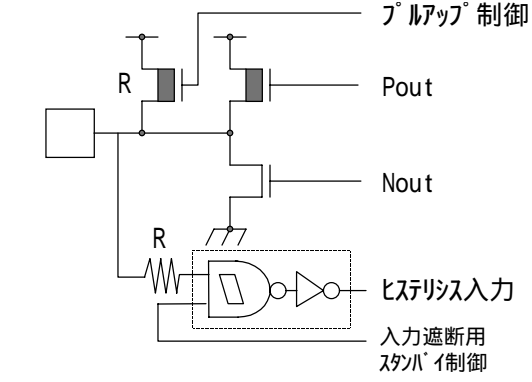
分類	回 路	備 考
A		発振回路 ・高速用発振帰還抵抗 = 約1M
B		発振回路 ・低速用発振帰還抵抗 = 約1M
C		ヒステリシス入力端子 ・抵抗値 = 約50K (TYP)
D		入力プルアップ制御付きCMOSヒステリシス入力端子 ・CMOSレベル出力 ・CMOSヒステリシス入力 (スタンバイ時入力遮断機能付き) ・プルアップ抵抗値 約50K (TYP) $I_{OL}=4\text{Ma}$

表 1.7-1 入出力回路形式（続く）

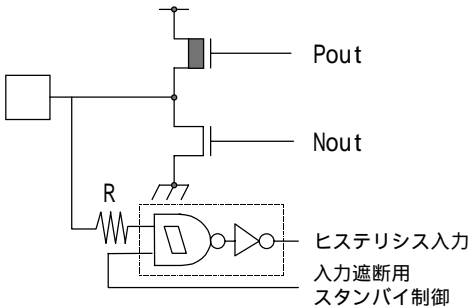
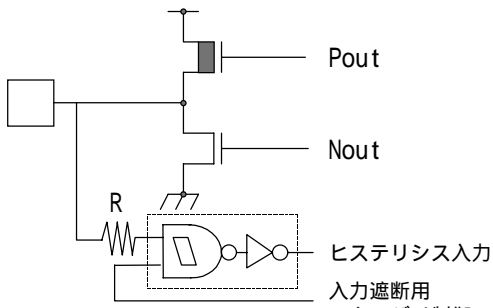
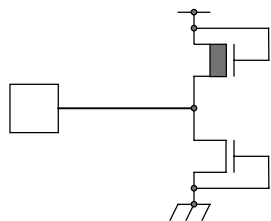
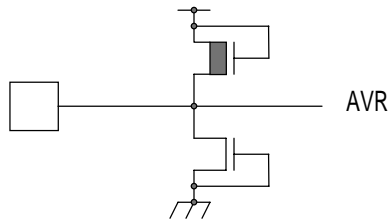
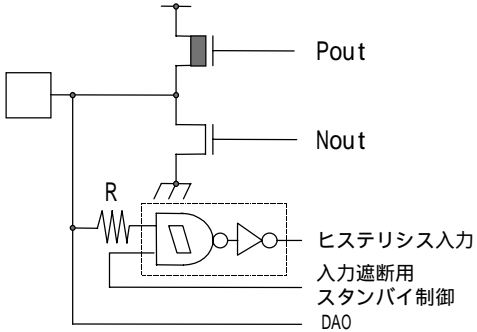
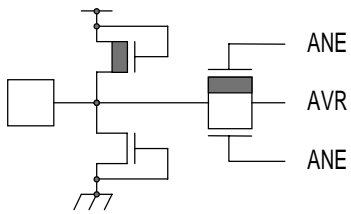
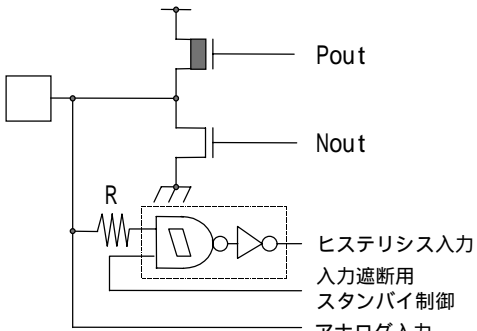
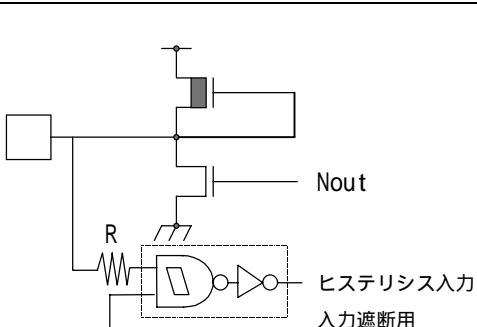
分類	回 路	備 考
E		CMOSヒステリシス入出力端子 ・CMOSレベル出力 ・CMOSヒステリシス入力 （スタンバイ時入力遮断機能付き） $I_{OL}=4mA$
F		CMOSヒステリシス入出力端子 ・CMOSレベル出力 ・CMOSヒステリシス入力 （スタンバイ時入力遮断機能付き） $I_{OL}=10mA$ （大電流ポート）
G		C端子出力 （コンデンサ接続用端子） MB90F574/AではN.C.端子
H		アナログ電源入力保護回路

表 1.7-1 入出力回路形式 (続き)

分類	回 路	備 考
I		<ul style="list-style-type: none"> ・CMOSヒステリシス入出力 ・アナログ出力/CMOS出力兼用端子 (アナログ出力時にはCMOS出力されません) (アナログ出力優先: DAE = 1) ・入力遮断用スタンバイ制御機能付き $I_{OL}=4mA$
J		A/Dコンバータref+(AVRH)電源入力端子 電源保護回路付き
K		<ul style="list-style-type: none"> ・CMOSヒステリシス入力/アナログ入力兼用端子 ・CMOS出力 ・入力遮断スタンバイ時入力遮断機能付き
L		ヒステリシス入力 N-chオープンドレイン出力 入力遮断用スタンバイ制御機能付き $I_{OL}=4mA$

1.8 デバイスの取り扱いに関する注意事項

デバイスを取り扱う際には、次の12の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
- ・ サブクロックモードを使用しない場合
- ・ ポート0, 1からの出力が不定になる場合
- ・ REALOSを使用する場合

デバイスの取り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）

CMOS ICでは、中・高耐圧以外の入力端子や出力端子にVccより高い電圧やVssより低い電圧が印加された場合、またはVcc～Vss間に定格を超える電圧が印加された場合、ラッチアップ現象が発生することがあります。ラッチアップが起きると電源電流が激増し、素子が熱破壊する恐れがあります。使用に際して、最大定格を超えることのないよう十分注意してください。

また、アナログ系の電源投入時、および切断時においてもアナログ電源（AVcc, AVRH, DVcc）とアナログ入力は、デジタル電源（Vcc）を超えないように注意してください。

供給電圧の安定化

Vcc電源電圧の動作保証範囲内においても、電源電圧の急激な変化があると誤動作を生じることがあります。安定化の基準として、商用周波数（50～60Hz）でのVccリップル変動（P-P値）は、標準Vcc値の10%以下に、また電源の切替え時などの瞬時変化においては、過渡変動率が0.1V/ms以下になるよう電圧変動を抑えることを推奨します。

電源投入時

内部に内蔵している降圧回路の誤動作を防ぐために、電源投入時における電圧の立ち上がり時間は50μs(0.2V～2.7Vの間)以上を確保してください。

未使用端子の処理

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

A/Dコンバータの電源端子処理

A/Dコンバータを使用しない場合においても、AVcc=Vcc, AVss=AVRH=AVRL=Vssとなるよう接続してください。

外部クロック使用時

外部クロック使用時においても、パワーオンリセット、サブクロックモードおよびストップモードからの解除には、発振安定待ち時間がとられます。また、図 1.8-1に示すように、外部クロックを使用する際には、X0端子のみを駆動し、X1端子はオープンとしてください。

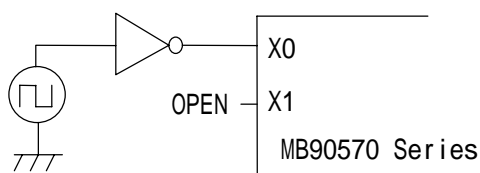


図 1.8-1 外部クロック使用例

電源端子

Vcc,Vssが複数ある場合、デバイス設計上はラッチアップなどの誤動作を防止するために、同電位にすべき端子はデバイス内部で接続してありますが、不要輻射の低減、グラウンドレベルの上昇によるストロブ信号の誤動作防止、総出力電流規格を守るなどのために、必ずそれら全てを外部で電源、およびグラウンドへ接続してください。

また、電流供給源から出来る限り低いインピーダンスでこのデバイスのVcc,Vssに接続する様配慮ください。このデバイスのVcc,Vss間に0.1 μ F程度のバイパスコンデンサを端子近くで接続する事を推奨します。

A/Dコンバータの電源，アナログ入力の投入順序

A/Dコンバータの電源 (AVcc,AVRH,AVRL) およびアナログ入力 (AN0 ~ AN7) の印加は、必ずデジタル電源 (Vcc) の投入後に行ってください。また、電源切断時はA/Dコンバータの電源およびアナログ入力切断の後で、デジタル電源 (Vcc) の切断を行ってください。その際、AVRHはAVccを超えないように投入、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おいても、入力電圧は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を同時に投入・切断をする事は問題ありません)。

「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」命令は、対応するバンクレジスタ (DTB, ADB, USB, SSB) の値を "00H" にし、使用してください。

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

詳細は、「2.6.2 「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意」を参照してください。

サブクロックモードを使用しない場合

サブクロックモードを使用しない場合においても、X0A,X1A端子に発振器を接続してください。

ポート0, 1からの出力が不定になる場合

電源を投入後、降圧回路の発振安定待ち時間(パワーオンリセット中)にポート0, 1から不定を出力します。タイミングは、図 1.8-2のようになりますので注意してください(対象品種：MB90V570, MB90V570A, MB90573, MB90574)。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：MB90F574, MB90F574A, MB90574C)。

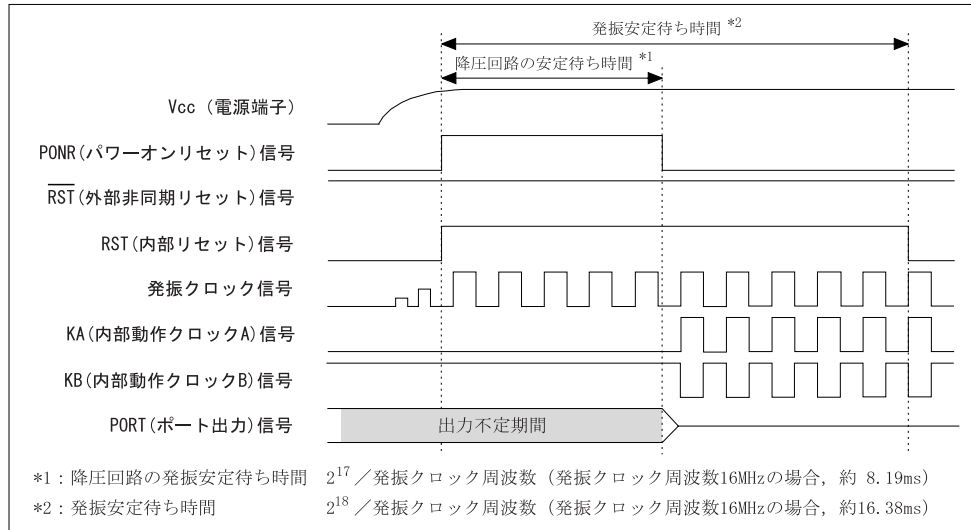


図 1.8-2 ポート0, 1が不定出力になるタイミングチャート

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，CPUの機能と動作について説明します。

- 2.1 メモリ空間
- 2.2 アドレス指定
- 2.3 メモリ空間における多バイト長データの配置
- 2.4 専用レジスタ
- 2.5 汎用レジスタ
- 2.6 プリフィックスコード

2.1 メモリ空間

F²MC-16LX CPUコアは、民生用・車載用機器などの高速リアルタイム処理が要求される用途向けに設計された16bitCPUです。F²MC-16LXの命令セットはコントローラ用途向けに設計されており、各種制御の高速・高効率処理が可能です。F²MC-16LXは16ビットデータ処理は勿論、内部に32ビットアキュムレータを搭載しているため32ビットデータ処理も可能です。メモリ空間は最大16Mbyte（拡張可能）、リニア方式およびバンク方式のいずれかにてアクセス可能です。また、命令体系はF²MC-8LのA-Tアーキテクチャをベースに、高級言語対応命令の追加・アドレッシングモードの拡張・乗除算命令の強化・ビット処理の充実化により命令が強化されています。

メモリ空間

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

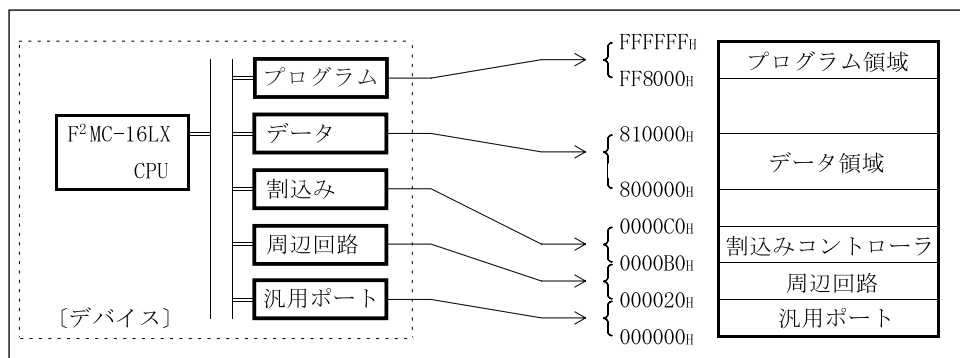


図 2.1-1 F²MC-16LX システムとメモリマップの関係例

2.2 アドレス指定

F²MC-16LXのアドレス指定には、以下に示す2種類の方式があります。

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

リニア方式によるアドレス指定

リニア方式には以下に示す2種類の方式があります。

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

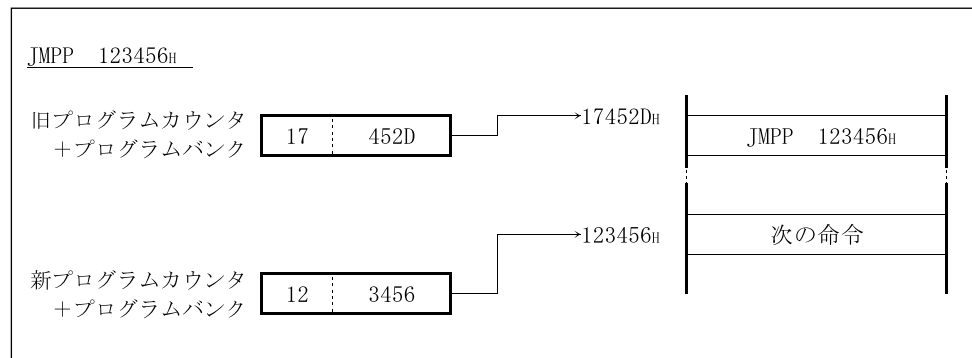


図 2.2-1 リニア方式の24ビットオペランド指定の例

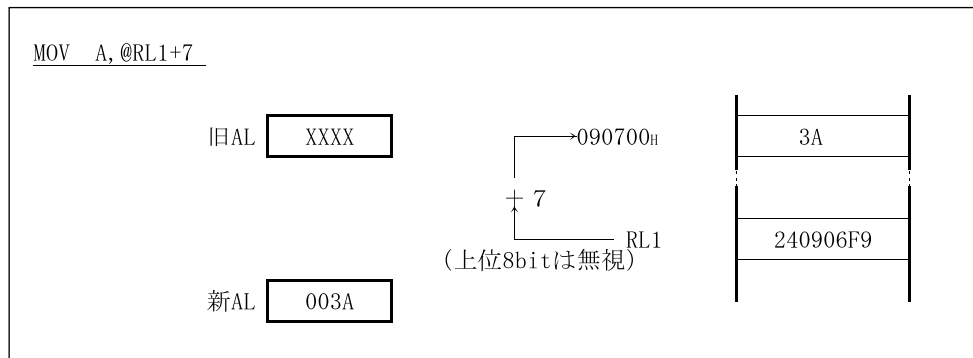


図 2.2-2 リニア方式の32ビットレジスタ間接指定の例

バンク方式によるアドレス指定

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

プログラムバンクレジスタ(PCB) [リセット時初期値 "FF_H"]

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

データバンクレジスタ(DTB) [リセット時初期値 "00H"]

DTBによって指定される64Kbyteのバンクをデータ(DT)空間と呼び、主として読み書き可能なデータや内外リソースの制御/データレジスタなどが存在します。

ユーザスタックバンクレジスタ(USB) [リセット時初期値 "00H"]

システムスタックバンクレジスタ(SSB) [リセット時初期値 "00H"]

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

アディショナルバンクレジスタ(ADB) [リセット時初期値 "00H"]

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

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

DTB, USB, SSB, ADBはリセットにより00Hに初期化され、PCBはリセットベクタにより指定された値に初期化されます。リセット後、DT, SP, ADの各空間はバンク00H(000000H ~ 00FFFFH)に配置され、PC空間はリセットベクタにより指定されたバンクに配置されます。

表 2.2-1 デフォルト空間

デフォルト空間	アドレッシング
プログラム空間	PC間接, プログラムアクセス, 分岐系
データ空間	@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

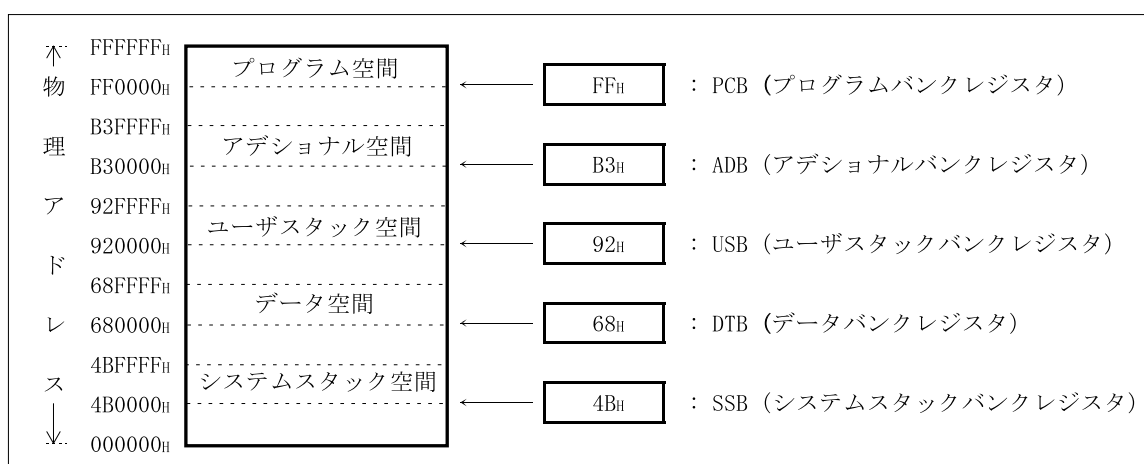


図 2.2-3 各空間の物理アドレスの例

2.3 メモリ空間における多バイト長データの配置

メモリ空間への多バイト長データの書き込みはアドレスの低い方から順に行われます。したがって32ビット長データであれば下位16bitが先に転送され、続いて上位16bitが転送されます。

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

メモリ空間における多バイト長データの配置

図 2.3-1に示すように、メモリ空間における多バイト長データは、下位8bitがn番地に、以下n+1番地、n+2番地、n+3番地・・・の順に配置されます。

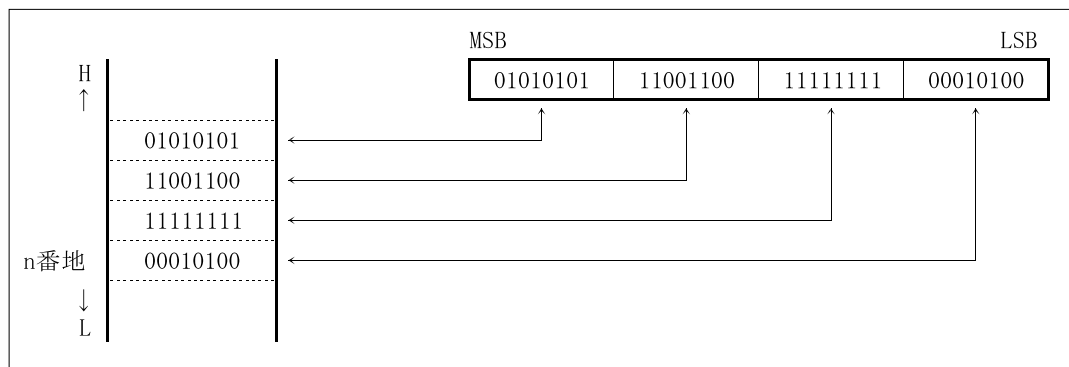


図 2.3-1 メモリ空間における多バイト長データの配置例

多バイト長データのアクセス

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

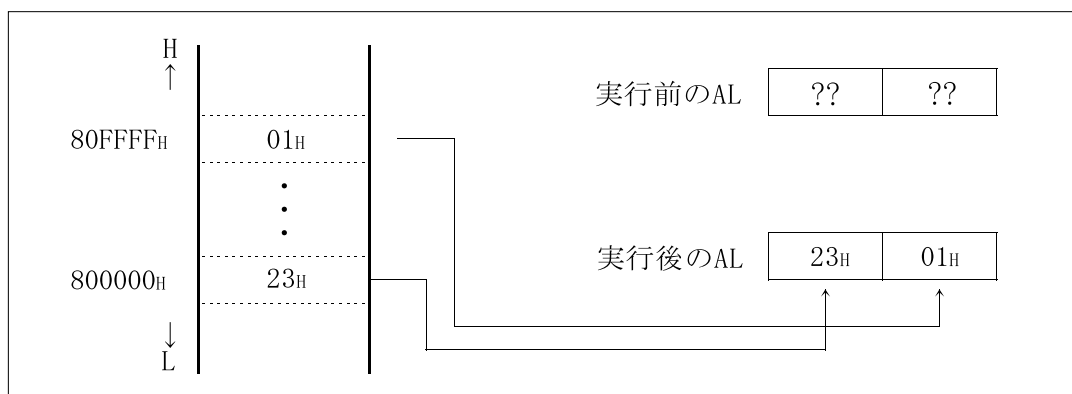


図 2.3-2 多バイト長データのアクセス例 (MOVWA, 080FFFFHの実行)

2.4 専用レジスタ

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

専用レジスタ

F²MC-16LXの専用レジスタは、以下に示す11種類があります。

アキュムレータ (A=AH:AL)

16bit × 2本のアキュムレータです (合計32bitのアキュムレータとしても使用可能)。

ユーザスタックポインタ (USP)

ユーザスタック領域を示す16bitのポインタです。

システムスタックポインタ (SSP)

システムスタック領域を示す16bitのポインタです。

プロセッサステータス (PS)

システムの状態を示す16bitのレジスタです。

プログラムカウンタ (PC)

プログラムが格納されているアドレスを持つ16bitのレジスタです。

プログラムバンクレジスタ (PCB)

PC空間を示す8bitのレジスタです。

データバンクレジスタ (DTB)

DT空間を示す8bitのレジスタです。

ユーザスタックバンクレジスタ (USB)

ユーザスタック空間を示す8bitのレジスタです。

システムスタックバンクレジスタ (SSB)

システムスタック空間を示す8bitのレジスタです。

アディショナルバンクレジスタ (ADB)

AD空間を示す8bitのレジスタです。

ダイレクトページレジスタ (DPR)

ダイレクトページを示す8bitのレジスタです。

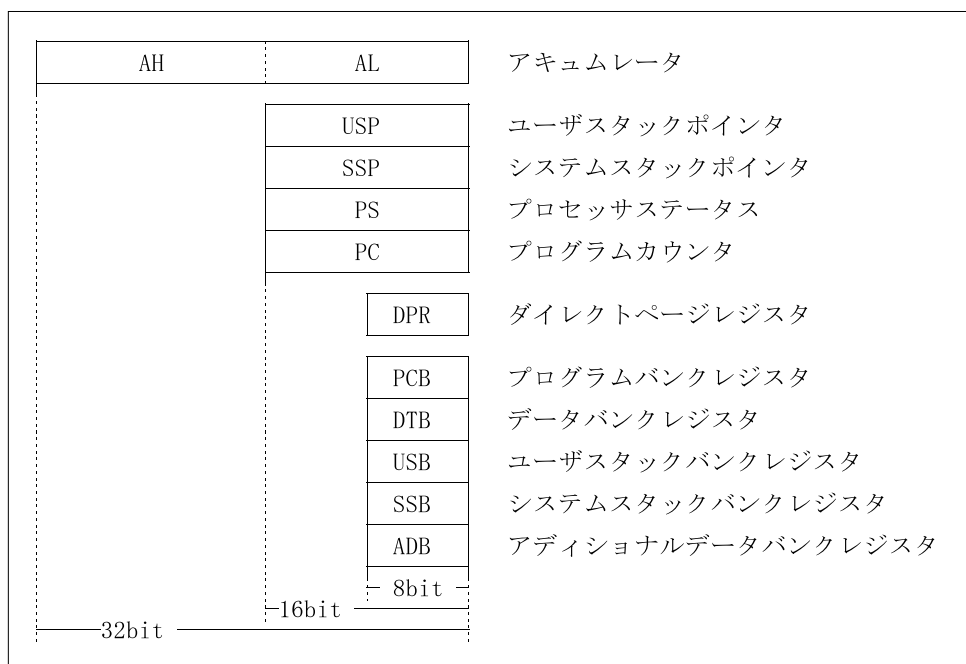


図 2.4-1 専用レジスタ

2.4.1 アキュムレータ(A)

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

アキュムレータ(A)

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

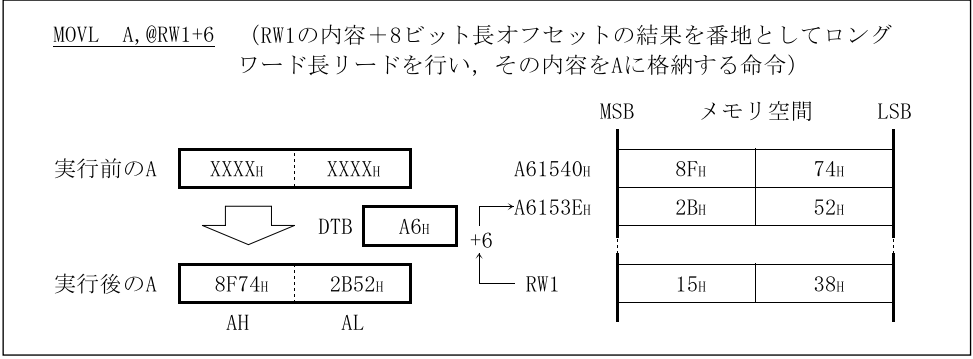


図 2.4-2 32bitデータ転送例

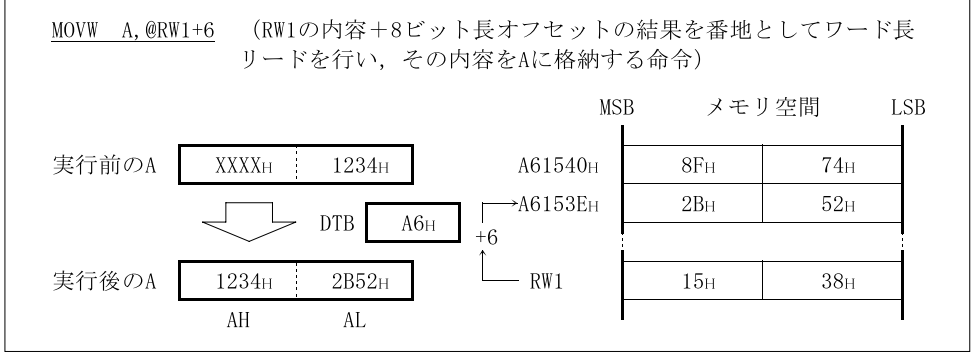


図 2.4-3 AL - AH転送例

図 2.4-4, 図 2.4-5に示すように, ALへバイト長以下のデータを転送する場合, データはゼロ拡張または符号拡張され16bit長となってALへ格納されます。AL中のデータは, ワード長としてもバイト長としても扱えます。ALにバイト処理の算術演算命令を実行すると, 演算前のALの上位8bitは無視され演算結果の上位8bitはすべて"0"になります。

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

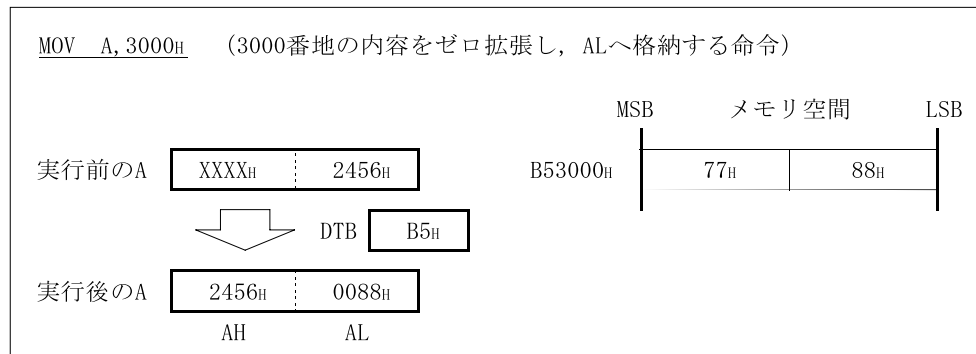


図 2.4-4 ゼロ拡張実行例

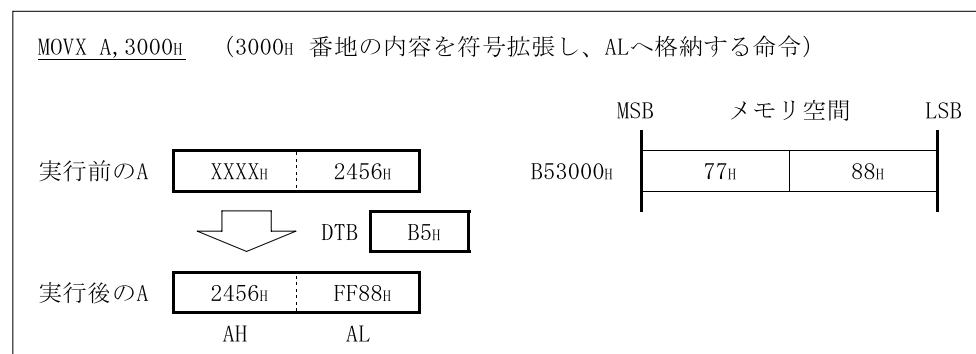


図 2.4-5 符号拡張実行例

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

ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)は、16bitのレジスタであり、プッシュ/ポップ命令およびサブルーチン実行時のデータ退避/復帰のメモリアドレスを示します。

ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)

ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)は、スタック系の命令により使用されますが、プロセッサステータス中のSフラグが"0"の場合は、USPレジスタが有効になり、Sフラグが"1"の場合はSSPレジスタが有効になります(図 2.4-6, 図 2.4-7を参照)。

また、割込みが受付られるとSフラグがセットされるため、割込み時のレジスタ退避は必ずSSPの示すメモリ中に行われます。割込みルーチンでのスタック処理はSSPが用いられ、割込みルーチン以外のスタック処理はUSPが用いられます。スタック空間を分ける必要のない場合はSSPだけを使ってください。スタック時のアドレスの上位8bitは、SSP SSB, USP USBにより示されます。また、USPおよびSSPは、リセットでは初期化されず、不定値になります。

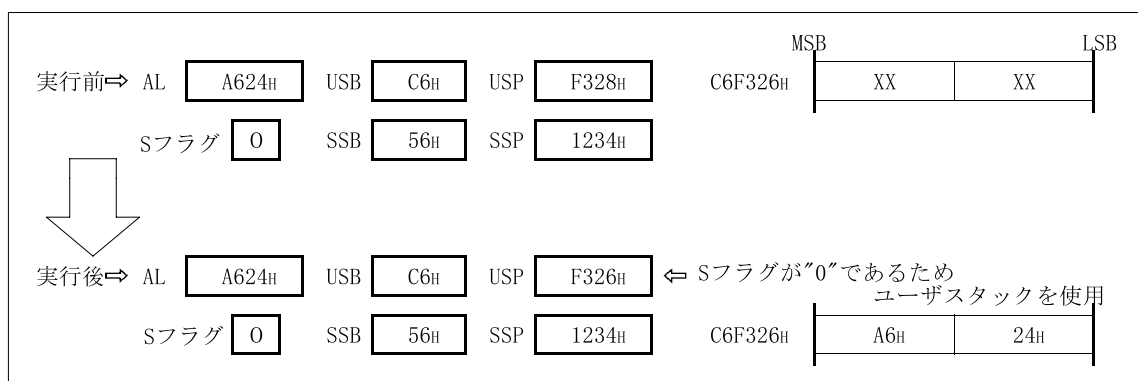


図 2.4-6 スタック操作命令とスタックポインタ(Sフラグが"0"の場合の PUSHW A例)

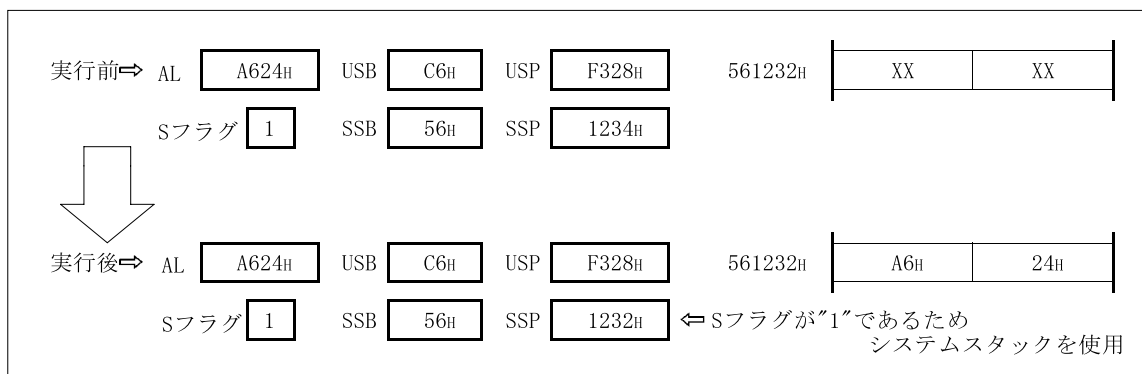


図 2.4-7 スタック操作命令とスタックポインタ(Sフラグが"1"の場合の PUSHW A例)

< 注意事項 >

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

2.4.3 プロセッサステータス(PS)

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

プロセッサステータス(PS)

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

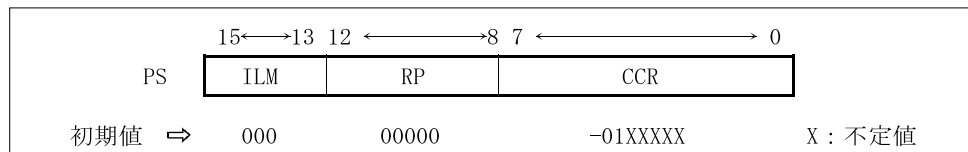


図 2.4-8 プロセッサステータス(PS)の構成

コンディションコードレジスタ(CCR)

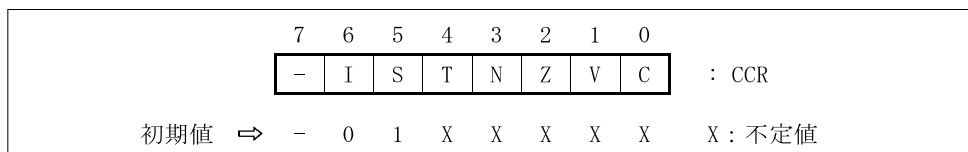


図 2.4-9 コンディションコードレジスタ(CCR)の構成

割込み許可フラグ(I)

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

スタックフラグ(S)

Sが"0"の場合は、スタック操作作用ポインタとしてUSPが有効になり、"1"の場合はSSPが有効になります。また、割込み受付時およびリセット時にはセットされます。

スティッキビットフラグ(T)

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

ネガティブフラグ(N)

演算結果のMSBが"1"の場合はセットされ、"0"の場合はクリアされます。

ゼロフラグ(Z)

演算結果がすべて"0"の場合はセットされ、それ以外の場合はクリアされず。

オーバーフローフラグ(V)

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

キャリーフラグ(C)

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

レジスタバンクポインタ(RP)

レジスタバンクポインタ(RP)は、F²MC-16LXの持つ汎用レジスタとそれが存在する内部RAMのアドレスとの関係を示すレジスタで、現在使用中のレジスタバンクの先頭のメモリアドレスを $[000180_{\text{H}} + (\text{RP}) * 10_{\text{H}}$ という変換式で示します。RPは5bitで構成され、00_H ~ 1F_Hまでの値をとることができ、また、000180_H ~ 00037F_Hのメモリ中にレジスタバンクを配置できます。

ただし、この範囲内であっても内部RAMでなければ汎用レジスタとして使用することはできません。RPはリセットによりすべて"0"に初期化されます。命令上ではRPに8bitの即値が転送できますが、実際に使用されるのはそのデータの下位5bitのみです。

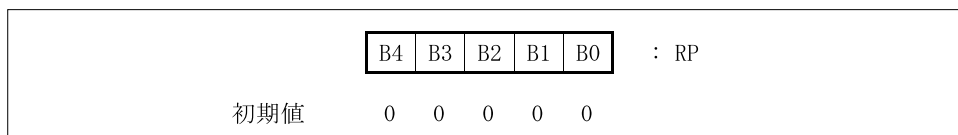


図 2.4-10 レジスタバンクポインタ(RP)の構成

インタラプトレベルマスクレジスタ(ILM)

インタラプトレベルマスクレジスタ(ILM)は3bitから構成されており、CPUの割込みマスクのレベルを示します。この3bitにより示されるレベルより強いレベルの割込み要求のみが受付られます。表 2.4-1に示すように、レベルの強弱は"0"が最強で、"7"が最弱と定義されています。従って割込みが受付られるには、現状のILMの保持値より小さい値の要求でなければなりません。割込みが受付られるとその割込みのレベル値がILMにセットされ、これ以降の同じかそれより低い優先順位の割込みは受付られなくなります。ILMはリセットによりすべて"0"に初期化されます。命令上ではILMに8bitの即値が転送できますが、実際に使用されるのはそのデータの下位3bitのみです。

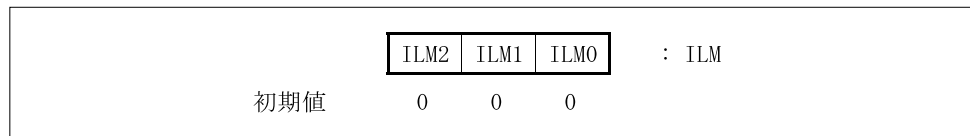


図 2.4-11 インタラプトレベルマスクレジスタ(ILM)の構成

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

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

2.4.4 プログラムカウンタ(PC)

プログラムカウンタ(PC)は、16bitのカウンタであり、CPUが実行すべき命令コードのメモリアドレスの下位16bitを示します。上位8bitアドレスはPCBにより示されます。

プログラムカウンタ(PC)

プログラムカウンタ(PC)は、条件分岐命令、サブルーチンコール命令、割込み、リセットなどにより内容が更新されます。また、オペランドアクセス時のベースポインタとして使用することもできます。

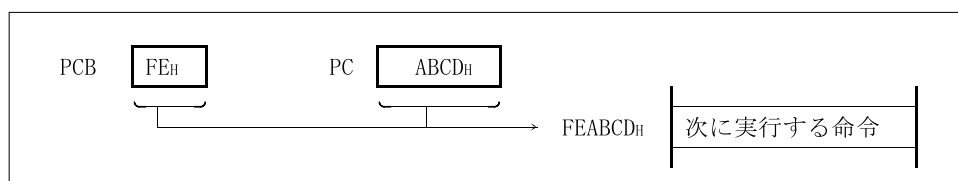


図 2.4-12 プログラムカウンタの構成

2.4.5 ダイレクトページレジスタ(DPR)

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

ダイレクトページレジスタ(DPR)

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

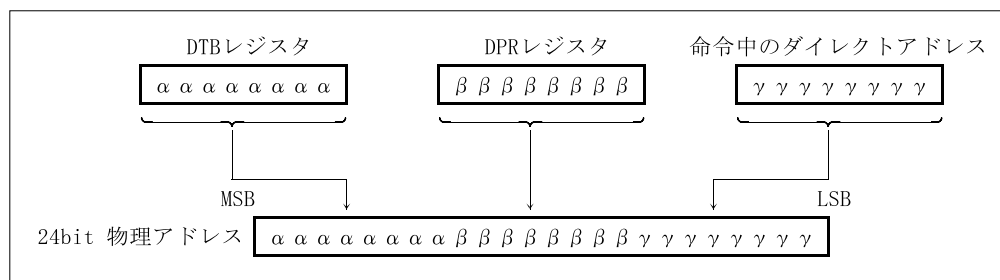


図 2.4-13 ダイレクトアドレッシングによる物理アドレスの生成

2.4.6 バンクレジスタ

バンクレジスタには、次の5種類があります。

- ・プログラムカウンタバンクレジスタ(PCB) <初期値：リセットベクタ中の値>
 - ・データバンクレジスタ(DTB) <初期値：00_H>
 - ・ユーザスタックバンクレジスタ(USB) <初期値：00_H>
 - ・システムスタックバンクレジスタ(SSB) <初期値：00_H>
 - ・アディショナルデータバンクレジスタ(ADB) <初期値：00_H>
-

バンクレジスタ

各々のバンクレジスタは、それぞれPC空間、DT空間、SP空間（ユーザ）、SP空間（システム）、AD空間の配置されるメモリバンクを示します。すべてのバンクレジスタはバイト長であり、リセットにより、PCBはリセットベクタにより00_Hに初期化されます。PCB以外のバンクレジスタは、読出し／書込みができます。PCBは読出しは可能ですが、書込みはできません。

16Mbyte全空間に分岐するJMPP,CALLP,RETP,RETIQ,RETF命令実行時、および、割込み時にPCBは書き換わります。

各レジスタの動作は「2.1 メモリ空間」を参照してください。

2.5 汎用レジスタ

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

汎用レジスタ

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

- R0～7 :8bitの汎用レジスタ
- RW0～7:16bitの汎用レジスタ
- RL0～3:32bitの汎用レジスタ

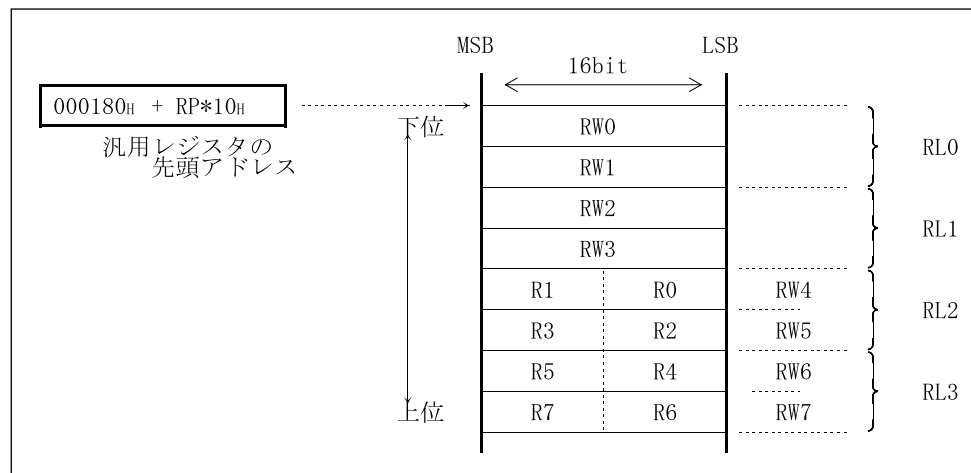


図 2.5-1 汎用レジスタ

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

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

また、RLiの上位・下位とRWの関係は、次の式で表すことができます。

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

レジスタバンク

レジスタバンクは、表 2.5-1に示すように、バイトレジスタ(R0～R7)、ワードレジスタ(RW0～RW7)、ロングワードレジスタ(RL0～RL3)の汎用レジスタとして各種演算、ポインタとして各種命令に使用できます。また、RL0～RL3は、全空間を直接アクセスするリニアポインタとしても使用できます。レジスタバンクの内容は、通常のRAMと同様に、リセットでは初期化されず、リセット前の状態が保持されます。ただし、パワーオン時は、不定値になります。

表 2.5-1 レジスタバンクの機能

レジスタ	機能
R0 ~ R7	各種命令のオペランドとして使用
RW0 ~ RW7	ポインタ, 各種命令のオペランドとして使用
RL0 ~ RL3	ロングポインタ, 各種命令のオペランドとして使用

< 注意事項 >

- ・ R0はバレルシフトのカウンタおよびノーマライズ命令のカウンタとしても使用します。
- ・ RW0はストリング命令のカウンタとしても使用します。

2.6 プリフィックスコード

プリフィックスコードには、バンクセレクトプリフィックス、コモンレジスタバンクプリフィックス、フラグ変化抑止プリフィックスの3種類があります。

命令の前に、これらのプリフィックスコードを置くことで、動作の一部を変更することが可能です。

バンクセレクトプリフィックス

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

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

表 2.6-1 バンクセレクトプリフィックス

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

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

ストリング命令〔MOVSW / MOVSW / SCEQ / SCWEQ / FILS / FILSW〕

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

スタック操作命令〔PUSHW / POPW〕

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

I/Oアクセス命令〔MOV A,io / MOV io,A / MOVX A,io / MOVW A,io / MOVW io,A / MOV io,#imm8 / MOVW io,#imm16 / MOV B A,io:bp / MOV B io:bp,A / SETB io:bp / CLRB io:bp BBC io:bp,rel / BBS io:bp,rel / WBTC, WBTS〕

プリフィックスの有無に関わらず、バンクのI/O空間が使用されます。

フラグ変更命令〔AND CCR,#imm8 / OR CCR,#imm8〕

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

POPW ps

プリフィックスの有無に関わらず、Sフラグに応じてSSBまたはUSBが使用されます。プリフィックスの効果が次の命令まで及びます。

MOV ILM,#imm8

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

RETI

プリフィックスの有無に関わらず、SSBが使用されます。

コモンレジスタバンクプリフィックス(CMR)

複数のタスク間でのデータ交換を容易にするためには、その場合のRPがどのような値でも、比較的簡単に定められた同一のレジスタバンクをアクセスする手段が必要です。コモンレジスタバンクプリフィックス(CMR)を、レジスタバンクをアクセスする命令の前に置くことで、現在のRPの値に関係なくその命令のレジスタアクセスをすべて000180H～00018FHにあるコモンバンク（RP=0の場合に選択されるレジスタバンク）に変更することができます。

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

ストリング命令〔MOVSW / MOVSW / SCEQ / SCWEQ / FILS / FILSW〕

プリフィックスコードを付加したストリング命令実行中に割込み要求が発生すると、割込み復帰後のストリング命令に対しては、プリフィックスが無効であるため誤動作となります。ストリング命令〔MOVSW,MOVSW,SCEQ,SCWEQ,FILS,FILSW〕に対してはCMRプリフィックスを付加しないでください。

フラグ変更命令〔AND CCR,#imm8 / OR CCR,#imm8 / POPW PS〕

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

MOV ILM,#imm8

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

フラグ変化抑止プリフィックス(NCC)

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

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

ストリング命令〔MOVSW / MOVSW / SCEQ / SCWEQ / FILS / FILSW〕

プリフィックスコードを付加したストリング命令の実行中に割込み要求が発生すると、割込み復帰後のストリング命令に対してはプリフィックスが無効であるために、誤動作となります。ストリング命令〔MOVSW,MOVSW,SCEQ,SCWEQ,FILS,FILSW〕には、NCCプリフィックスを付加しないでください。

フラグ変更命令〔AND CCR,#imm8 / OR CCR,#imm8 / POPW PS〕

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

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

プリフィックスの有無にかかわらず、CCRは命令の仕様どおり変化します。

JCTX @A

プリフィックスの有無にかかわらず、CCRは命令の仕様どおり変化します。

MOV ILM,imm8

命令動作は正常ですが、プリフィックスの効果が次の命令まで及びます。

2.6.1 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、次の3つの制約があります。

- ・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する。
- ・競合するプリフィックスコードが連続した場合、最後のものが有効となる。

プリフィックスコードと割込み／ホールド抑止命令

表 2.6-2に示す割込み／ホールド抑止命令とプリフィックスコードには、次のような制約があります。

表 2.6-2 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け付け 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 2.6-1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、受け付けません。この場合、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、初めてこれ以外の命令が実行された後になります。

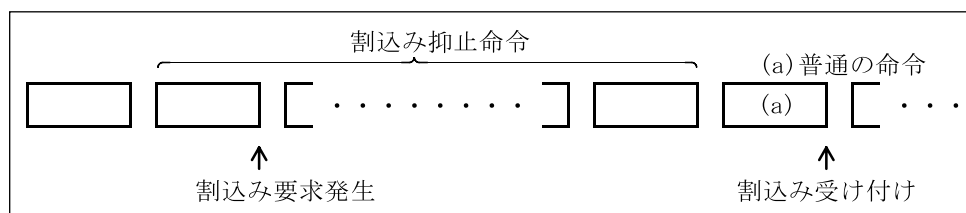


図 2.6-1 割込み／ホールドの抑止

プリフィックスコードの効果遅延

図 2.6-2に示すように、割り込み／ホールド抑止命令の前にプリフィックスコードを置いた場合、プリフィックスコードの効果は、割り込み／ホールド抑止命令後の最初の命令に対して有効となります。

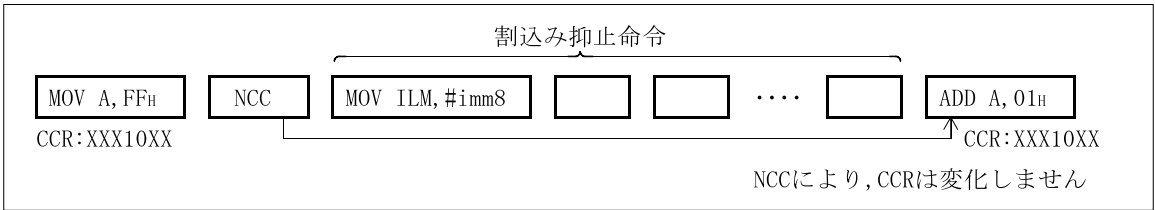


図 2.6-2 割り込み／ホールド抑止命令とプリフィックスコード

プリフィックスコードの連続

図 2.6-3に示すように、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、最後のものが有効になります。

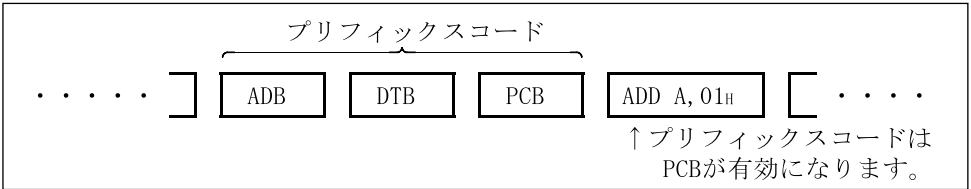


図 2.6-3 プリフィックスコードの連続

2.6.2 「DIV A,Ri」,「DIVW A,Rwi」命令の使用上の注意

命令の実行結果により得られる余りは、表 2.6-3に従った、メモリバンク領域(上位8ビット)の命令オペランドのレジスタに相当するアドレス(下位16ビット)に格納されます。このため、「DIV A,Ri」、「DIVW A,Rwi」命令は、対応するバンクレジスタを"00H"にして使用してください。

「DIV A,Ri」、「DIVW A,Rwi」命令の使用上の注意

表 2.6-3 「DIV A,Ri」、「DIVW A,Rwi」命令の使用上の注意

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 + CH: 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、除算の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、そこに余りが格納されます。

【例】

DTB=053H/RP=003Hの場合で「DIV A,R0」を実行すると、R0のアドレスは $0180H + RP(003H) \times 10H + 08H$ (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DTBですので、バンク指定053Hを付加したアドレスすなわち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、「2.5 汎用レジスタ」を参照してください)。

「DIV A,Ri」,「DIVW A,RWi」命令の注意事項の回避について

「DIV A,Ri」,「DIVW A,RWi」命令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,コンパイラでは上記命令を生成しないように変更し,アセンブラでは上記命令を同等の命令列に置き換える機能を追加しています。

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 割込み

この章では，割込みの機能と動作について説明します。

- 3.1 割込みの概要
- 3.2 割込み要因
- 3.3 割込みベクタ
- 3.4 ハードウェア割込み
- 3.5 ソフトウェア割込み
- 3.6 拡張インテリジェントI/Oサービス (EI²OS)
- 3.7 例外処理割込み

3.1 割込みの概要

F²MC-16LX はイベントなどの発生により、現在実行中の処理を中断して、別に定義したプログラムへ制御を移す割込み機能があります。

割込みの概要

割込み機能は以下の4つに分けることができます。

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

本章では、これら4種の割込み機能について述べます。

3.2 割込み要因

表 3.2-1に、割込み要因と割込みベクタ・割込み制御レジスタを示します。

割込み要因

表 3.2-1 割込み要因と割込みベクタ・割込み制御レジスタ

割込み要因	EI ² OS クリア	割込みベクタ		割込み制御レジスタ	
		番号	アドレス	番号	アドレス
リセット	×	# 08	FFFFDC _H	-	-
INT9命令	×	# 09	FFFFD8 _H	-	-
例外	×	# 10	FFFFD4 _H	-	-
A/Dコンバータ		# 11	FFFFD0 _H	ICR00	0000B0 _H
インプットキャプチャ0取込み		# 12	FFFFC0 _H		
DTP0 (外割り0)		# 13	FFFFC8 _H	ICR01	0000B1 _H
インプットキャプチャ1取込み		# 14	FFFFC4 _H		
アウトプットコンペア0一致		# 15	FFFFC0 _H	ICR02	0000B2 _H
アウトプットコンペア1一致		# 16	FFFFB0 _H		
アウトプットコンペア2一致		# 17	FFFFB8 _H	ICR03	0000B3 _H
アウトプットコンペア3一致		# 18	FFFFB4 _H		
I/Oシリアル0		# 19	FFFFB0 _H	ICR04	0000B4 _H
フリーランタイム	×	# 20	FFFFA0 _H		
I/Oシリアル1		# 21	FFFFA8 _H	ICR05	0000B5 _H
時計タイマ	×	# 22	FFFFA4 _H		
I/Oシリアル2		# 23	FFFFA0 _H	ICR06	0000B6 _H
DTP1 (外割り1)		# 24	FFFF90 _H		
DTP2/3 (外割り2/3)		# 25	FFFF98 _H	ICR07	0000B7 _H
8/16ビットPPG0カウンタボロー	×	# 26	FFFF94 _H		
DTP4/5 (外割り4/5)		# 27	FFFF90 _H	ICR08	0000B8 _H
8/16ビットPPG1カウンタボロー	×	# 28	FFFF80 _H		
アップダウンカウンタ0 ₀ 0 ₀ /オーバーフロー/反転		# 29	FFFF88 _H	ICR09	0000B9 _H
アップダウンカウンタ0コンペア一致		# 30	FFFF84 _H		
アップダウンカウンタ1 ₀ 0 ₀ /オーバーフロー/反転		# 31	FFFF80 _H	ICR10	0000BA _H
アップダウンカウンタ1コンペア一致		# 32	FFFF70 _H		
DTP6 (外割り6)		# 33	FFFF78 _H	ICR11	0000BB _H
タイムベースタイマ	×	# 34	FFFF74 _H		
DTP7 (外割り7)		# 35	FFFF70 _H	ICR12	0000BC _H
I2Cインタフェース	×	# 36	FFFF60 _H		
UART1 受信完了		# 37	FFFF68 _H	ICR13	0000BD _H
UART1 送信完了		# 38	FFFF64 _H		
UART0 受信完了		# 39	FFFF60 _H	ICR14	0000BE _H
UART0 送信完了		# 40	FFFF50 _H		
フラッシュメモリ	×	# 41	FFFF58 _H	ICR15	0000BF _H
遅延割込み	×	# 42	FFFF54 _H		

：EI²OS割込みクリア信号にて割込み要求フラグはクリアされます。ストップ要求できます。

：EI²OS割込みクリア信号にて割込み要求フラグはクリアされます。

×：EI²OS割込みクリア信号にて割込み要求フラグはクリアされません。

3.3 割込みベクタ

表 3.3-1に，割込みベクター一覧表を示します。

割込みベクタ

表 3.3-1 割込みベクター一覧表

ソフトウェア 割込み命令	ベクトル アドレスL	ベクトル アドレスM	ベクトル アドレスH	モード レジスタ	割込みNo	ハードウェア割込み
INT 0	FFFFFC _H	FFFFFD _H	FFFFFE _H	未使用	#0	なし
:	:	:	:	:	:	:
INT 7	FFFFE0 _H	FFFFE1 _H	FFFFE2 _H	未使用	#7	なし
INT 8	FFFFDC _H	FFFFDD _H	FFFFDE _H	FFFFDF	#8	(RESETベクトル)
INT 9	FFFFD8 _H	FFFFD9 _H	FFFFDA _H	未使用	#9	なし
INT 10	FFFFD4 _H	FFFFD5 _H	FFFFD6 _H	未使用	#10	<例外>
INT 11	FFFFD0 _H	FFFFD1 _H	FFFFD2 _H	未使用	#11	A/Dコンバータ
INT 12	FFFFC0 _H	FFFFC1 _H	FFFFC2 _H	未使用	#12	ICAP0
INT 13	FFFFC8 _H	FFFFC9 _H	FFFFCA _H	未使用	#13	DTP0
INT 14	FFFFC4 _H	FFFFC5 _H	FFFFC6 _H	未使用	#14	ICAD1
INT 15	FFFFC0 _H	FFFFC1 _H	FFFFC2 _H	未使用	#15	アウトプットコンペア#0
INT 16	FFFFB0 _H	FFFFB1 _H	FFFFB2 _H	未使用	#16	アウトプットコンペア#1
INT 17	FFFFB8 _H	FFFFB9 _H	FFFFBA _H	未使用	#17	アウトプットコンペア#2
INT 18	FFFFB4 _H	FFFFB5 _H	FFFFB6 _H	未使用	#18	アウトプットコンペア#3
INT 19	FFFFB0 _H	FFFFB1 _H	FFFFB2 _H	未使用	#19	拡張I/Oシリアル0
INT 20	FFFFA0 _H	FFFFA1 _H	FFFFA2 _H	未使用	#20	フリーランタイム
INT 21	FFFFA8 _H	FFFFA9 _H	FFFFAA _H	未使用	#21	拡張I/Oシリアル1
INT 22	FFFFA4 _H	FFFFA5 _H	FFFFA6 _H	未使用	#22	時計タイマ
INT 23	FFFFA0 _H	FFFFA1 _H	FFFFA2 _H	未使用	#23	拡張I/Oシリアル2
INT 24	FFFF90 _H	FFFF91 _H	FFFF92 _H	未使用	#24	DTP1
INT 25	FFFF98 _H	FFFF99 _H	FFFF9A _H	未使用	#25	DTP2/3
INT 26	FFFF94 _H	FFFF95 _H	FFFF96 _H	未使用	#26	PPG0
INT 27	FFFF90 _H	FFFF91 _H	FFFF92 _H	未使用	#27	DTP4/5
INT 28	FFFF80 _H	FFFF81 _H	FFFF82 _H	未使用	#28	PPG1
INT 29	FFFF88 _H	FFFF89 _H	FFFF8A _H	未使用	#29	アップダウン0ボロー
INT 30	FFFF84 _H	FFFF85 _H	FFFF86 _H	未使用	#30	アップダウン0コンペア
INT 31	FFFF80 _H	FFFF81 _H	FFFF82 _H	未使用	#31	アップダウン1ボロー
INT 32	FFFF70 _H	FFFF71 _H	FFFF72 _H	未使用	#32	アップダウン1コンペア
INT 33	FFFF78 _H	FFFF79 _H	FFFF7A _H	未使用	#33	DTP6
INT 34	FFFF74 _H	FFFF75 _H	FFFF76 _H	未使用	#34	タイムベース
INT 35	FFFF70 _H	FFFF71 _H	FFFF72 _H	未使用	#35	DTP7
INT 36	FFFF60 _H	FFFF61 _H	FFFF62 _H	未使用	#36	I ² Cインタフェース
INT 37	FFFF68 _H	FFFF69 _H	FFFF6A _H	未使用	#37	UART1 受信完了
INT 38	FFFF64 _H	FFFF65 _H	FFFF66 _H	未使用	#38	UART1 送信完了
INT 39	FFFF60 _H	FFFF61 _H	FFFF62 _H	未使用	#39	UART0 受信完了
INT 40	FFFF50 _H	FFFF51 _H	FFFF52 _H	未使用	#40	UART0 送信完了
INT 41	FFFF58 _H	FFFF59 _H	FFFF5A _H	未使用	#41	フラッシュメモリ
INT 42	FFFF54 _H	FFFF55 _H	FFFF56 _H	未使用	#42	遅延割込み
INT 43	FFFF50 _H	FFFF51 _H	FFFF52 _H	未使用	#43	なし
:	:	:	:	:	:	:
INT 254	FFFC04 _H	FFFC05 _H	FFFC06 _H	未使用	#254	なし
INT 255	FFFC00 _H	FFFC01 _H	FFFC02 _H	未使用	#255	なし

3.4 ハードウェア割込み

ハードウェア割込みは、内蔵リソースからの割込み要求信号に対応して、CPUがそれまで実行していたプログラムの実行を一時中断し、ユーザの定義した割込み処理用プログラムへ制御を移行する機能です。

ハードウェア割込みの概要

ハードウェア割込みの起動は、割込み要求がもつ割込みレベルと、CPUのプロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)の比較、およびコンディションコードレジスタ(CCR)内のIフラグの内容をハードウェアで参照したのち、発生条件に合致すれば発生します。

ハードウェア割込みが発生するときにCPUが行う処理は以下のものがあります。

- CPU内部のPC, PS, A, PCB, DTB, ADB, DPR レジスタをシステムスタックへ退避させます。
- プロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)の設定。自動的に現在要求している割込みレベルと同じになります。
- 対応する割込みベクトルの内容を取り込み、そこへ分岐します。

ハードウェア割込みの構造

ハードウェア割込みに関連する機構は、以下の4つの部分に分かれて存在します。

内蔵リソース

- 割込み許可ビット、割込み要求ビット：リソースからの割込み要求の制御

割込みコントローラ

- 割込み制御レジスタ(ICR)：割込みのレベル付け、同時要求割込みの優先度判定

CPU

- 割込み許可フラグ(I)、インタラプトマスクレジスタ(ILM)：要求割込みレベルと現レベルの比較、割込み許可状態の識別

マイクロコード：割込み処理用ステップ

メモリ上の"FFFC00_H" ~ "FFFFFF_H"

- 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。ソフトウェア割込みと共通で使用しています。

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

内蔵リソース領域への書き込み中のハードウェア割り込み要求

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

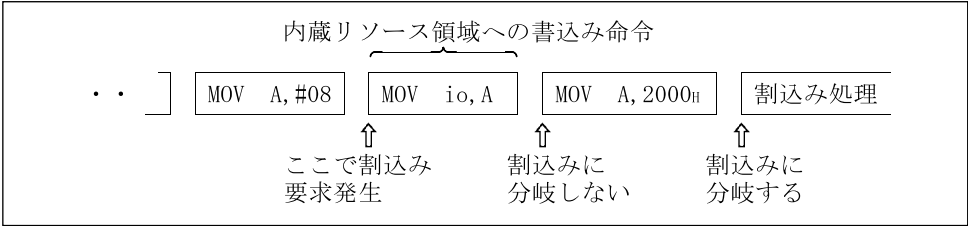


図 3.4-1 内蔵リソース領域への書き込み中のハードウェア割り込み要求

割り込み抑止命令

F²MC-16LXには、ハードウェア割り込みの要求の有無を検出しない、割り込み抑止命令があります。「2.6.1 プリフィックスコードに関する制約」を参照してください。

多重割り込み

F²MC-16LX CPU は多重割り込みをサポートしています。そのため、割り込み処理の実行中にその割り込みのもつ割り込みレベルより強いレベルの割り込みが発生すると、現在実行中の命令を終了したあとでそちらに制御が移ります。強いレベルでの割り込みが終了すると、もとの割り込み処理に戻ります。割り込み処理実行中に、その割り込みと同等かより弱いレベルの割り込み処理が発生した場合は、インタラプトレベルマスクレジスタ(ILM)の内容やコンディションコードレジスタ(CCR)内のIフラグの命令による変更を施さないかぎり現在の割り込み処理の終了まで、新しい割り込み要求は保留されます。なお、拡張インテリジェントI/Oサービスは多重に起動されることはなく、1つの拡張インテリジェントI/Oサービスの処理中は、他の割り込み要求及び拡張インテリジェントI/Oサービス要求は全て保留されます。

割り込み時におけるスタック内へのレジスタの退避

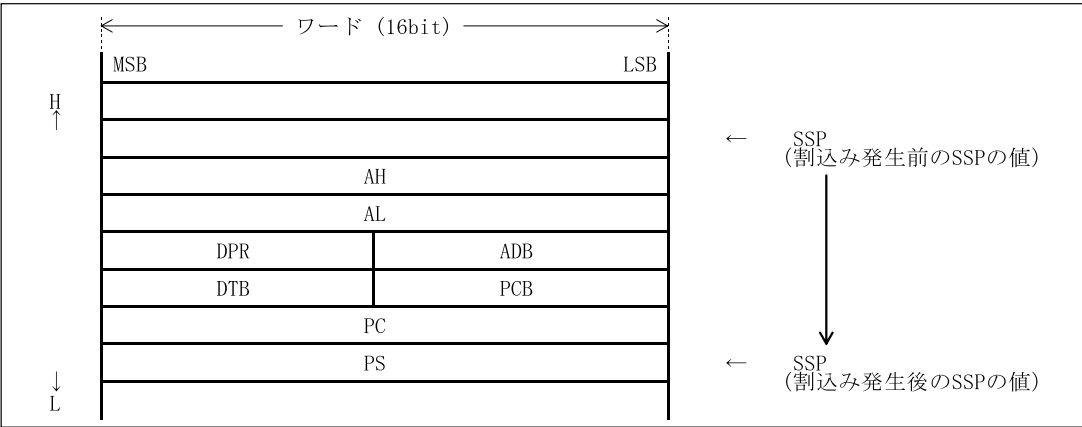


図 3.4-2 スタック内に退避されたレジスタ

ハードウェア割込みの使用上の注意

内蔵リソースによっては、制御レジスタやデータレジスタの読出し動作により、割込み要求をクリアするものがあります。割込み要求が発生し、割込み処理ハードウェアに制御が移る前に、読出し動作による割込みクリアが行われますと誤動作になります。そのため、レジスタリード動作により割込み要求をクリアする内蔵リソースを使用する場合には、割込み要求発生時のレジスタリードを行わないでください。

3.4.1 ハードウェア割込みの動作

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

ハードウェア割込みの動作

ハードウェア割込み要求の機能を備えた内蔵リソースには、割込み要求フラグと割込み許可フラグが存在します。割込み要求フラグは、割込み要求の有無を示し、割込み許可フラグは、CPUへの割込み要求の許可/禁止選択します。割込み要求フラグは、内蔵リソース固有のイベントの発生によりセットされ、割込み許可フラグが"許可"であった場合に、リソースは割込みコントローラへ割込み要求を発生します。

割込みコントロールは、同時に受け取った各割込み要求の割込みレベル(IL)同士を比較し、もっとも高いレベルの要求(ILの値の最も小さいもの)を採択してCPUへ通知します。同一レベルの要求が複数あった場合には、割込み番号が少ないものを優先します。各割込み要求と各ICRの関係はハードウェアで決まっています。

CPUは受け取った割込みレベル(IL)とプロセッサステータス(PS)内のインタラプトマスクレジスタ(ILM)とを比較し、割込みレベル(IL)<(ILM)で、コンディションコードレジスタ(CCR)内のIビットが"1"に設定されている場合、現在実行中の命令が終了後、割込み処理用マイクロコードを起動します。割込み処理マイクロコードの先頭で、割込み制御レジスタ(ICR)内のISEビットを参照し、ISEビットが"0"に設定されている場合、割込みシーケンスが起動されます。

割込み処理では最初にシステムスタック (SSBとSSPの示すシステムスタック空間)へ専用レジスタ(PS), PC, PCB, DTB, ADB, DPR, Aの12バイトの内容を退避します。次に、割込みベクタプログラムカウンタ(PC, PCB)へのロード。プロセッサステータス(PS)内のインタラプトマスクレジスタ(ILM)を、受け付けた割込み要求のレベル値に更新、コンディションコードレジスタ(CCR)内のSフラグを"1"に設定します。図 3.4-3の図中における、 ~ の内容を以下に説明します。

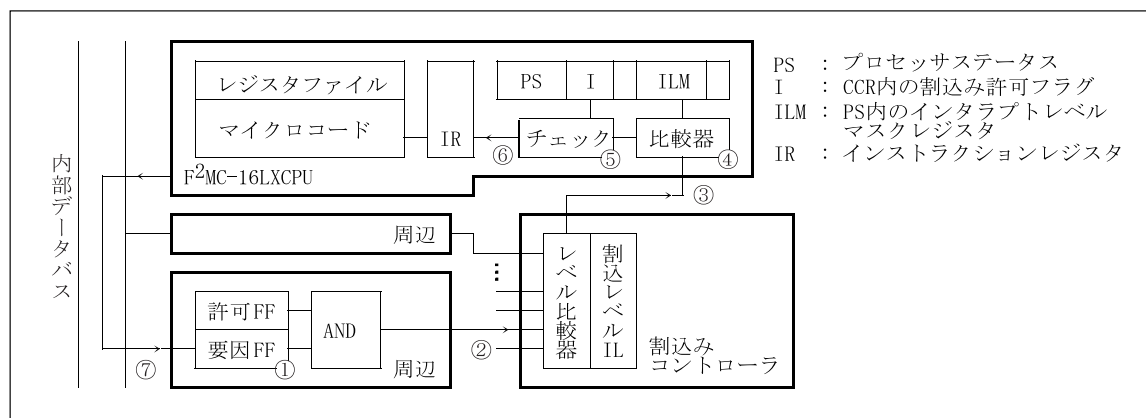


図 3.4-3 ハードウェア割込みの動作

周辺の内部で割込み要因が発生します。

周辺内の割込み許可ビットを参照し、割込み許可になっていれば、周辺から割込みコントローラへ割込み要求を発生します。

割込み要求を受けとった割込みコントローラは、同時に要求のあった割込みの優先順位を判定した上で、該当する割込み要求に対応する割込み

レベル(IL)をCPUへ転送します。

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)をプロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)と比較します。

比較の結果が現在の割込み処理レベルより優先順位が高い場合、同じコンディションコードレジスタ(CCR)内のIフラグの内容をチェックします。

のチェックの結果Iフラグが割込み許可状態である場合、現在実行中の命令の実行が終了するまで待ち、終了時点でILMレジスタに要求されたレベル(IL)を設定し、制御を割込み処理ルーチンへ移します。

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

および で、CPUが行う割込み処理の実行時間を以下に示します。割込みシーケンスへ移行するまでの時間は、スタックポインタの指しているアドレスによって異なります。

ハードウェア割込み処理時間

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

割込み要求サンプル待ち時間

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

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

割込みハンドリング時間(割込み処理準備に要する時間)

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

表 3.4-1 割込み処理時のサイクル数の補正值

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

3.4.2 ハードウェア割り込みの動作フロー

図 3.4-4に , ハードウェア割り込みの動作フローを示します。

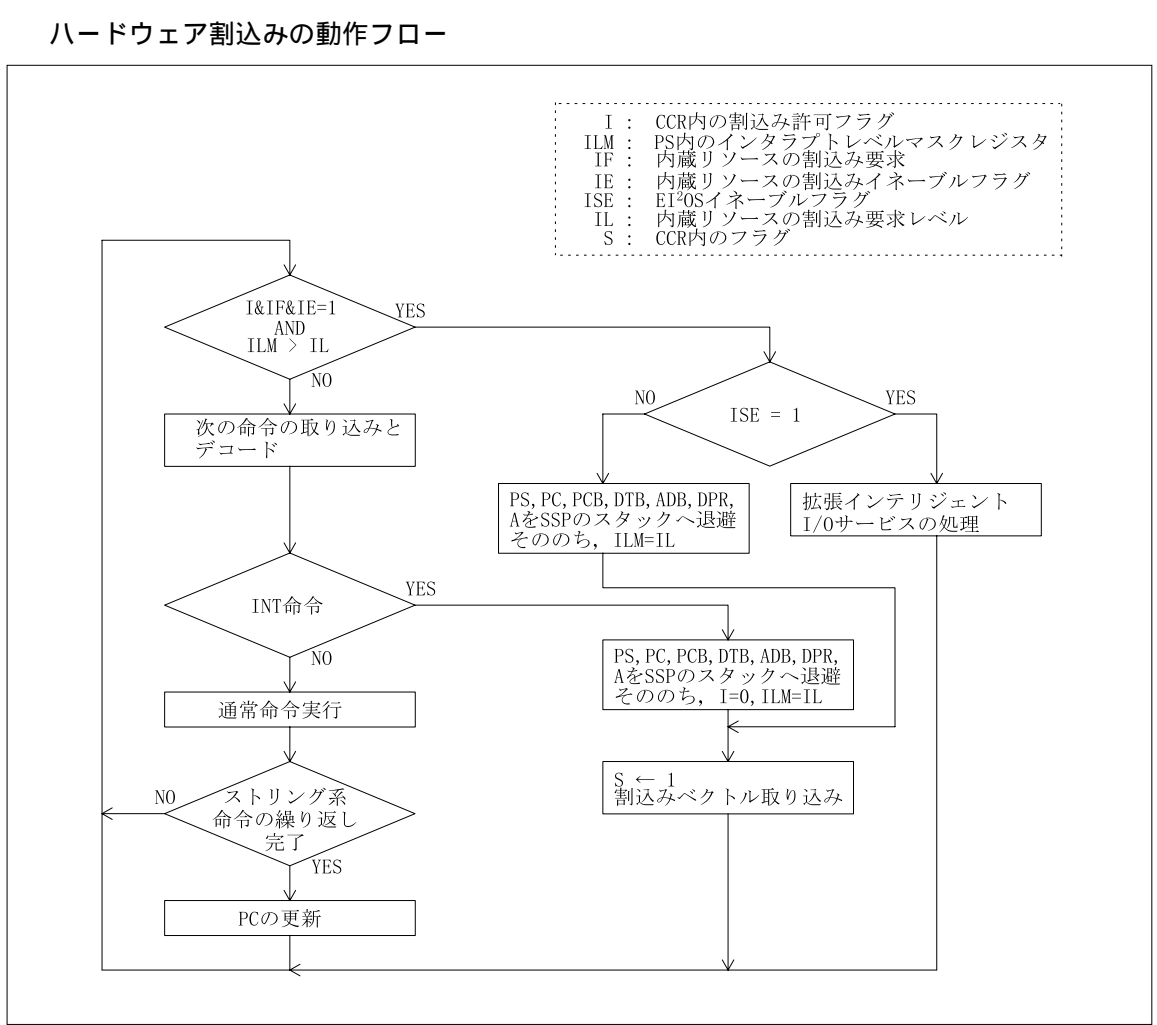


図 3.4-4 ハードウェア割り込み動作フロー

3.4.3 ハードウェア割込み使用手順

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

ハードウェア割込み使用手順

図 3.4-5に、ハードウェア割込み使用手順例を示します。

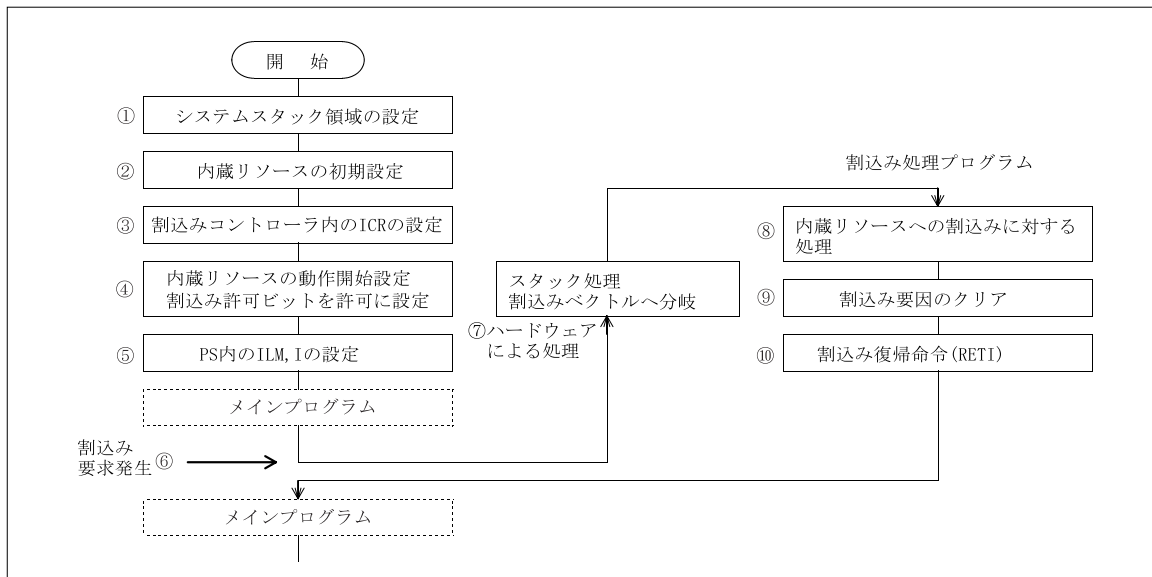


図 3.4-5 ハードウェア割込み使用手順例

システムスタック領域を設定する。

割込み要求を発生可能な内蔵リソースの初期設定を行う。

割込みコントローラ内の割込み制御レジスタ(ICR)を設定する。

内蔵リソースを動作開始状態にし、割込み許可ビットを許可に設定する。

プロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)、コンディションコードレジスタ(CCR)内のIフラグを割込み受け付け可能に設定する。

内蔵リソースの割込み発生で、ハードウェア割込みが要求が発生する。

割込み処理ハードウェアにより、レジスタの退避が行われ、割込み処理プログラムに分岐する。

割込み処理プログラムで、割込み発生に対する内蔵リソースへの処理を行う。

内蔵リソース回路の割込み要求を解除する。

割込み復帰命令を実行し、分岐前のプログラムに復帰する。

3.5 ソフトウェア割り込み

ソフトウェア割り込みは、専用の命令の実行に対応して、CPUがそれまで実行していたプログラムの実行から、ユーザの定義した割り込み処理用プログラムへ制御を移行する機能です。

ソフトウェア割り込みの概要

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

- CPU内部のPC, PS, A, PCB, DTB, ADB, DPRレジスタのシステムスタックへ退避させます。
- コンディションコードレジスタ(CCR)内のIフラグを設定します。割り込みは、自動的に禁止されます。
- 対応する割り込みベクトルの内容を取り込み、そこへ分岐します。

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

INT命令には割り込みレベルが存在しませんので、インタラプトレベルマスクレジスタ(ILM)の更新は行われません。INT命令実行中は、コンディションコードレジスタ(CCR)内のIフラグをクリアし、後続の割り込み要求を保留状態にします。

ソフトウェア割り込みの構造

ソフトウェア割り込みに関連する機構は、すべてCPUの内部に存在します。

CPU

マイクロコード.....割り込み処理用ステップ

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

表 3.3-1に示すように、割り込みベクタは、ハードウェア割り込みと同じ割り込み領域を共用します。たとえば、割り込み要求番号のINT 11は、ハードウェア割り込みのA/Dコンバータの割り込みで使用されると同時に、ソフトウェア割り込みのINT #11でも使用されます。したがって、A/DコンバータとINT #11は、同じ割り込み処理ルーチンと呼ばれます。

ソフトウェア割り込みの動作

CPUがソフトウェア割り込み命令を取り込み実行すると、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では最初にシステムスタックSSBとSSPの示すシステムスタック空間へ専用レジスタ(PSとPC, PCB, DTB, ADB, DPR, Aの12バイト)を退避させます。次に、マイクロコードは割り込みベクタを3バイト リードしてプログラムカウンタ(PC, PCB)へのロード、コンディションコードレジスタ(CCR)内のIフラグを"0"、Sフラグを"1"に設定します。その結果、次に実行する命令はユーザの定義した割り込み処理プログラムになります。

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

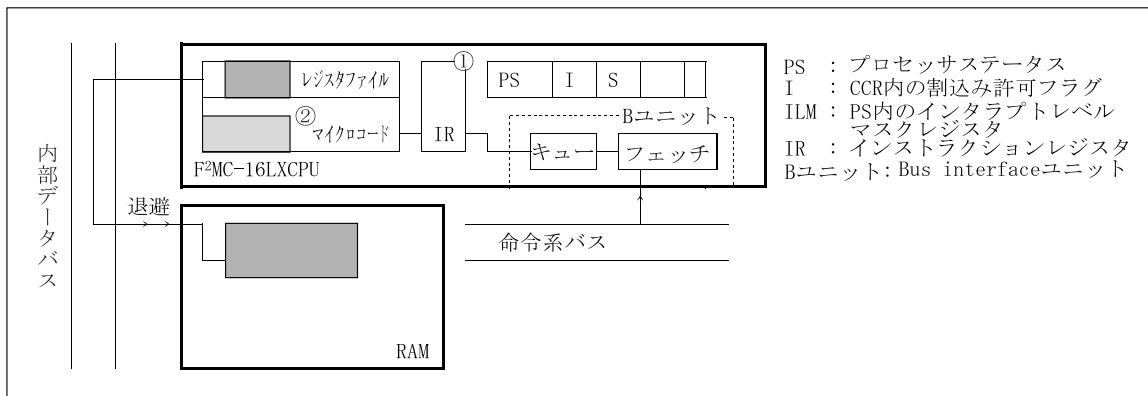


図 3.5-1 ソフトウェア割込みの発生と解除まで

ソフトウェア割込み命令を実行します。

ソフトウェア割込み命令に対応したマイクロコードにしたがって、レジスタファイル中のCPU内専用レジスタの退避を行います。

ユーザの割込み処理ルーチン内のRETI命令で割込み処理が終了します。

ソフトウェア割込みに関する注意事項

プログラムバンクレジスタ(PCB)がFF_Hの場合、CALLV命令のベクトル領域はINT #vct8命令のテーブルと重複します。ソフトウェアを作成する際は、CALLV命令とINT #vct8命令が同じアドレスを使用しないように注意してください。

3.6 拡張インテリジェントI/Oサービス (EI²OS)

拡張インテリジェントI/Oサービス (EI²OS) とは、ハードウェア割込み動作の一種で、I/Oとメモリとの間で自動データ転送を行う機能です。従来は、割込み処理プログラムにより、I/Oとメモリとのデータ転送を、DMA(ダイレクトメモリアクセス)のように行うことができます。

拡張インテリジェントI/Oサービス (EI²OS) の概要

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

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

また、EI²OSによるデータ転送が終了すると、終了条件を割込み制御レジスタ(ICR)内のS1, S0ビットに設定し、自動的に割込み処理ルーチンへ分岐します。したがって、割込み処理プログラムでEI²OSステータス(ICR: S1, S0)をチェックすることにより、EI²OSの終了条件の種類を判別することができます。

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

割込み制御レジスタ(ICR)

割込みコントローラ内にあり EI²OSの起動 EI²OSのチャネル指定 およびEI²OS終了時の状態表示を行います。

拡張インテリジェントI/Oサービスディスクリプタ(ISD)

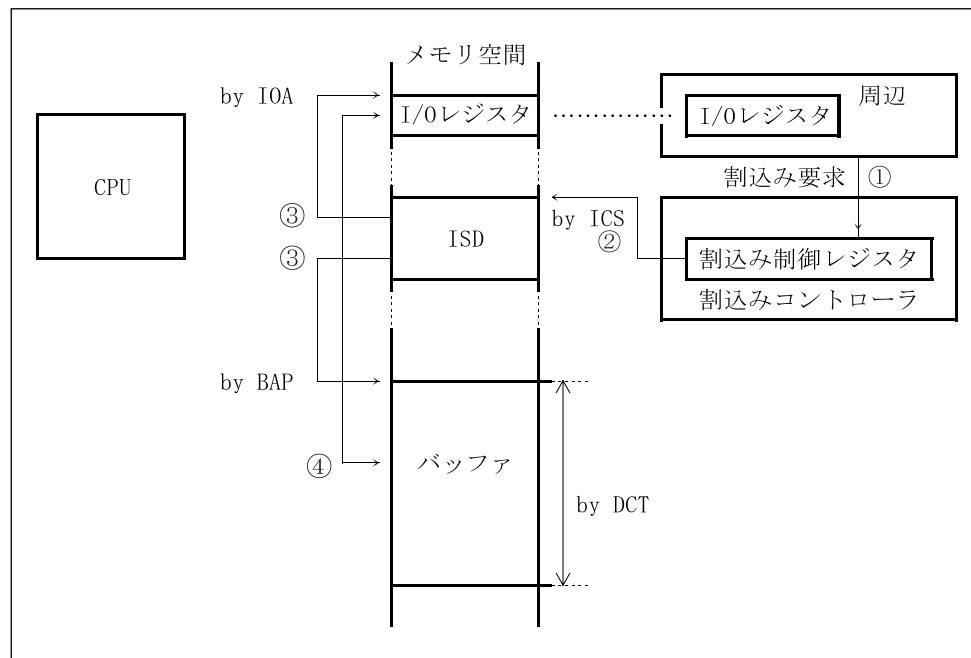
RAM上にあり、転送モード、I/Oのアドレスや転送数、バッファアドレスを保持します。

< 注意事項 >

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

拡張インテリジェントI/Oサービス(EI²OS)の動作

図 3.6-1に，EI²OSの動作を示します。



(注意)

- ・ IOAで指定できる領域は，000000H～00FFFFHです。
- ・ BAPで指定できる領域は，000000H～FFFFFFHです。
- ・ DTCで指定できる最大転送数は，65536個です。

図 3.6-1 拡張インテリジェントI/Oサービス(EI²OS)の動作

I/Oが転送を要求します。

割込みコントローラがディスクリプタを選択します。

転送元/先をディスクリプタから読出します。

I/Oとメモリ間で転送が行われます。

割込み要因は自動的にクリアされます。

拡張インテリジェントI/Oサービス(EI²OS)の構造

EI²OSに関連する機構は，以下の4つの部分に分かれて存在します。

内蔵リソース

割込み許可ビット，割込み要求ビット：リソースからの割込み要求の制御

割込みコントローラ

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

CPU

I, ILM：要求割込みレベルと現レベルの比較，割込み許可状態の識別

マイクロコード：EI²OS処理用ステップ

RAM

ディスクリプタ：EI²OSの転送情報を記述

3.6.1 割込み制御レジスタ(ICR)

割込み制御レジスタは割込みコントローラ内にあり，割込み機能を持つすべてのI/Oに対応して存在します。このレジスタには以下に示す三つの機能があります。

- ・対応する周辺の割込みレベルの設定
- ・対応する周辺の割込みを通常割込みにするか，拡張インテリジェントI/Oサービス(EI²OS)にするかの選択
- ・拡張インテリジェントI/Oサービス(EI²OS)のチャンネルの選択

このレジスタに対するリードモディファイライト系の命令でのアクセスは誤動作を引き起こしますので行わないでください。

割込み制御レジスタ(ICR)

割込み制御レジスタ(ICR)								↔ ビットNo.
アドレス : B0 _H ~BF _H	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0
リード/ライト ⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(1)	(1)	(1)
								↔ ビットNo.
アドレス : B0 _H ~BF _H	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
	—	—	S1	S0	ISE	IL2	IL1	IL0
リード/ライト ⇒	(-)	(-)	(R)	(R)	(R)	(R)	(R)	(R)
初期値 ⇒	(X)	(X)	(0)	(0)	(0)	(1)	(1)	(1)

図 3.6-2 割込み制御レジスタ(ICR)

< 注意事項 >

ICS3～0ビットは拡張インテリジェントI/Oサービス(EI²OS)を起動する場合のみ有効となります。EI²OSを起動する場合はISEビットを"1"に設定し，起動しない場合は"0"に設定してください。EI²OSを起動しない場合，ICS3～0ビットは何を設定してもかまいません。

ICS1ビットとICS0ビットは書込みのみ，S1ビットとS0ビットは読出しのみ，それぞれ有効です。

【ビット15～12, 7～4】ICS3～0 (EI²OSチャンネル選択ビット)

拡張インテリジェントI/Oサービス(EI²OS)チャンネル選択ビットです。

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

表 3.6-1 EI²OSチャンネル選択ビットとディスクリプタアドレスの対応

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

【ビット13～12, 5～4】S0, S1(EI²OSステータス)

拡張インテリジェントI/Oサービス(EI²OS)ステータスビットです。

読出し専用のビットで、EI²OS終了時にこのビットの値を調べることで、終了条件が何であったかを判別することができます。リセットにより"0"初期化されます。

表 3.6-2 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動作中あるいは非起動時
0	1	カウント終了による停止状態
1	0	リザーブ
1	1	内蔵リソースからの要求による停止状態

【ビット11, 3】ISE(EI²OS許可ビット)

拡張インテリジェントI/Oサービス(EI²OS)を許可します。

割込み要求の発生時にこのビットに"1"が設定されているとEI²OSが起動され、"0"が設定されていると割込みシーケンスが起動されます。また、EI²OSが終了(カウント終了による場合と内蔵リソースからの要求による場合)すると、ISEビットは"0"設定されます。対応する内蔵リソースにEI²OSの機能がないときには、ソフトウェアでISEビットを"0"に設定する必要があります。読み出しと書込みが可能なビットです。リセットにより"0"に初期化されます。

【ビット10～8，2～0】IL0，IL1，IL2(割込みレベル設定ビット)

割込みレベルを設定ビットです。

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

表 3.6-3 割込みレベル設定ビットの割込みレベルの対応

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

3.6.2 拡張インテリジェントI/Oサービスディスクリプタ(ISD)

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

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

拡張インテリジェントI/Oサービスディスクリプタ(ISD)

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

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

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

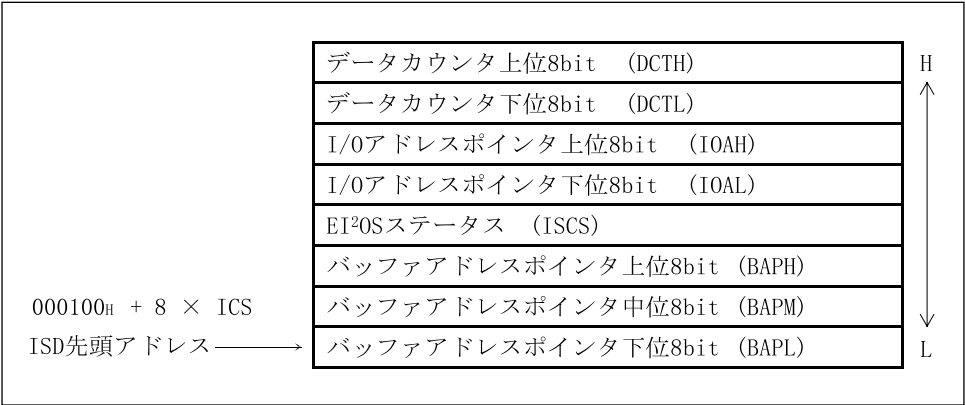


図 3.6-3 拡張インテリジェントI/Oサービスディスクリプタの構成

3.6.3 拡張インテリジェントI/Oサービスディスクリプタ(ISD)の各レジスタ

拡張インテリジェントI/Oサービスディスクリプタ(ISD)は、次のレジスタから構成されています。

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

データカウンタ (DCT)

データカウンタ (DCT)は、16ビット長のレジスタで、転送データ数に対応したカウンタとなります。各データの転送の後に、カウンタは1デクリメント（減値）します。このカウンタがゼロになるとEI²OSは終了します。

データカウンタ上位	15	14	13	12	11	10	9	8	⇐ ビットNo.
	B15	B14	B13	B12	B11	B10	B09	B08	DCTH
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
データカウンタ下位	7	6	5	4	3	2	1	0	⇐ ビットNo.
	B07	B06	B05	B04	B03	B02	B01	B00	DCTL
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 3.6-4 データカウンタ (DTC) の構成

I/Oレジスタアドレスポインタ (IOA)

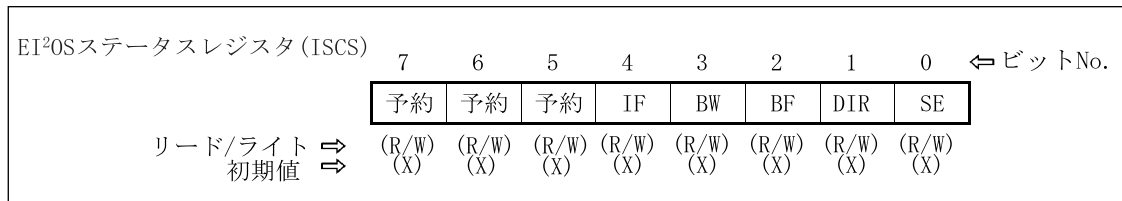
I/Oレジスタアドレスポインタ (IOA)は、16ビット長のレジスタで、I/Oレジスタアドレスポインタの16bitにより、バッファとデータ転送を行うI/Oレジスタの下位アドレス(A15～A0)を示します。上位アドレス(A23～A16)はすべて"0"であり、"000000_H"から"00FFFF_H"番地までの任意のI/Oをアドレスで指定できます。

I/Oアドレスポインタ上位	15	14	13	12	11	10	9	8	⇐ ビットNo.
	A15	A14	A13	A12	A11	A10	A09	A08	IOAH
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
I/Oアドレスポインタ下位	7	6	5	4	3	2	1	0	⇐ ビットNo.
	A07	A06	A05	A04	A03	A02	A01	A00	IOAL
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 3.6-5 I/Oレジスタアドレスポインタ (IOA)の構成

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

EI²OSステータスレジスタ (ISCS) は、8ビット長のレジスタで、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)転送方向を示します。

図 3.6-6 EI²OSステータスレジスタ (ISCS) の構成

【ビット7～5】予約ビット

必ず"0"に設定してください。

【ビット4】IF

I/Oレジスタアドレスポインタ(IOA)の更新 / 固定を指定します。

表 3.6-4 IFビットの機能

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

【ビット3】BW

転送データ長を指定します。

表 3.6-5 BWビットの機能

BW	機能
0	バイト
1	ワード

【ビット2】BF

バッファアドレスポインタ(BAP)の更新 / 固定を指定します。

表 3.6-6 BFビットの機能

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

< 注意事項 >

バッファアドレスポインタは、下位16ビットだけ変化し、インクリメントだけ可能です。

【ビット1】: DIR

データの転送方向を指定します。

表 3.6-7 DIRビットの機能

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

【ビット0】SE

内蔵リソースからの要求での拡張インテリジェントI/Oサービス (EI²OS)の終了を制御します。

表 3.6-8 EI²OS終了制御ビット

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

バッファアドレスポインタ (BAP)

バッファアドレスポインタ(BAP)は、24ビットのレジスタで、インテリジェントI/Oサービス(EI²OS)による次回の転送で使用するアドレスを保持します。BAPは、EI²OSの各チャネルに対してそれぞれ独立に存在しますので、EI²OSの各チャネルは、16Mバイトの任意のアドレスとI/Oの間でデータを転送することができます。

< 注意事項 >

EI²OSステータスレジスタ(ISCs)内のBFビットにより"更新あり"にした場合、BAPは下位16ビット(BAPM, BAPL)のみ変化し、上位8ビット(BAPH)は変化しません。

3.6.4 拡張インテリジェントI/Oサービス(EI²OS)の動作

周辺機能から割込み要求が発生し，対応する割込み制御レジスタ(ICR)でEI²OSの起動が設定されていれば，CPUはEI²OSによるデータ転送を行います。指定回数のデータ転送が終了すれば，自動的に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) の処理手順

図 3.6-7に,CPU内部のマイクロコードによる，EI²OSの動作フローを示します。

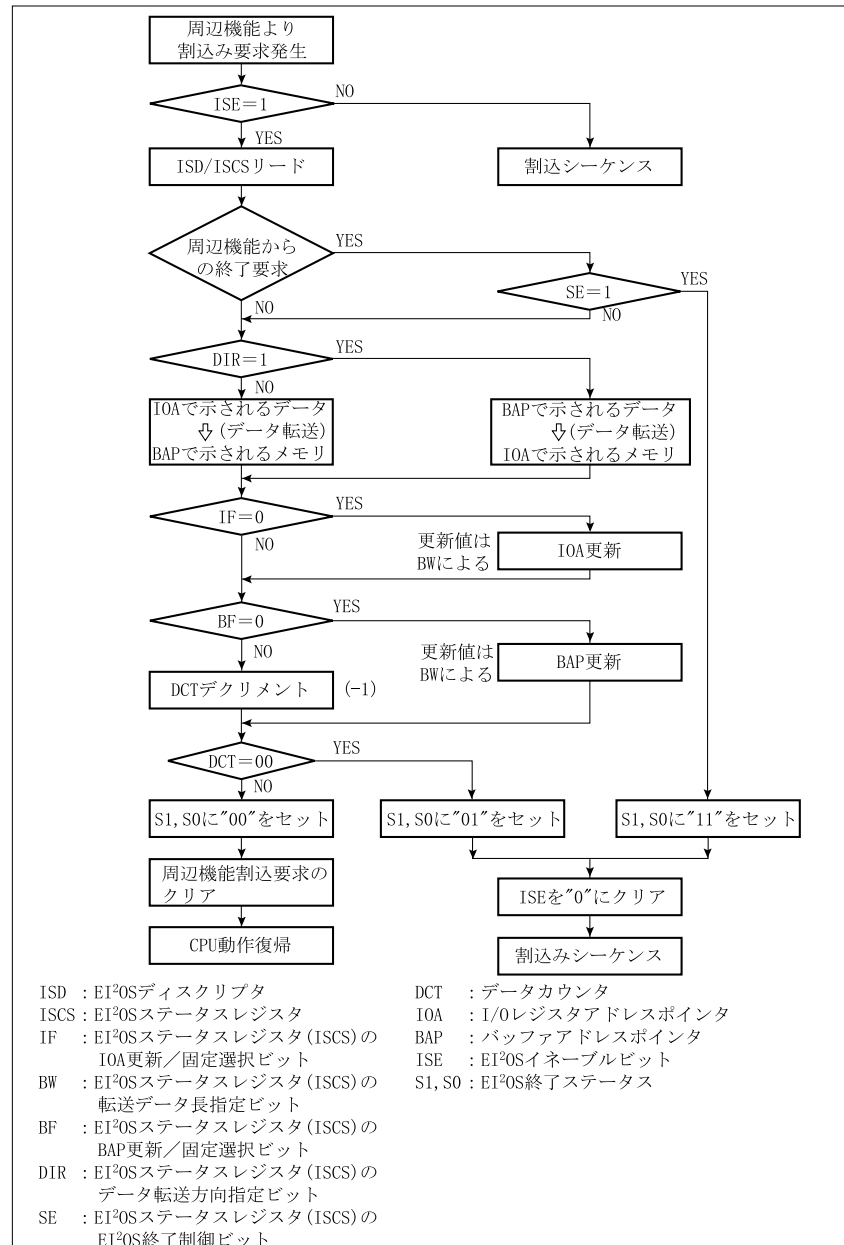


図 3.6-7 拡張インテリジェントI/Oサービス (EI²OS) の動作フロー

3.6.5 拡張インテリジェントI/Oサービス(EI²OS)の使用手順

拡張インテリジェントI/Oサービス(EI²OS)を使用するには、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、および割り込み制御レジスタ(ICR)などの設定が必要です。

拡張インテリジェントI/Oサービス(EI²OS)の使用手順

図 3.6-8に、EI²OSのソフトウェアとハードウェアによる処理を示します。

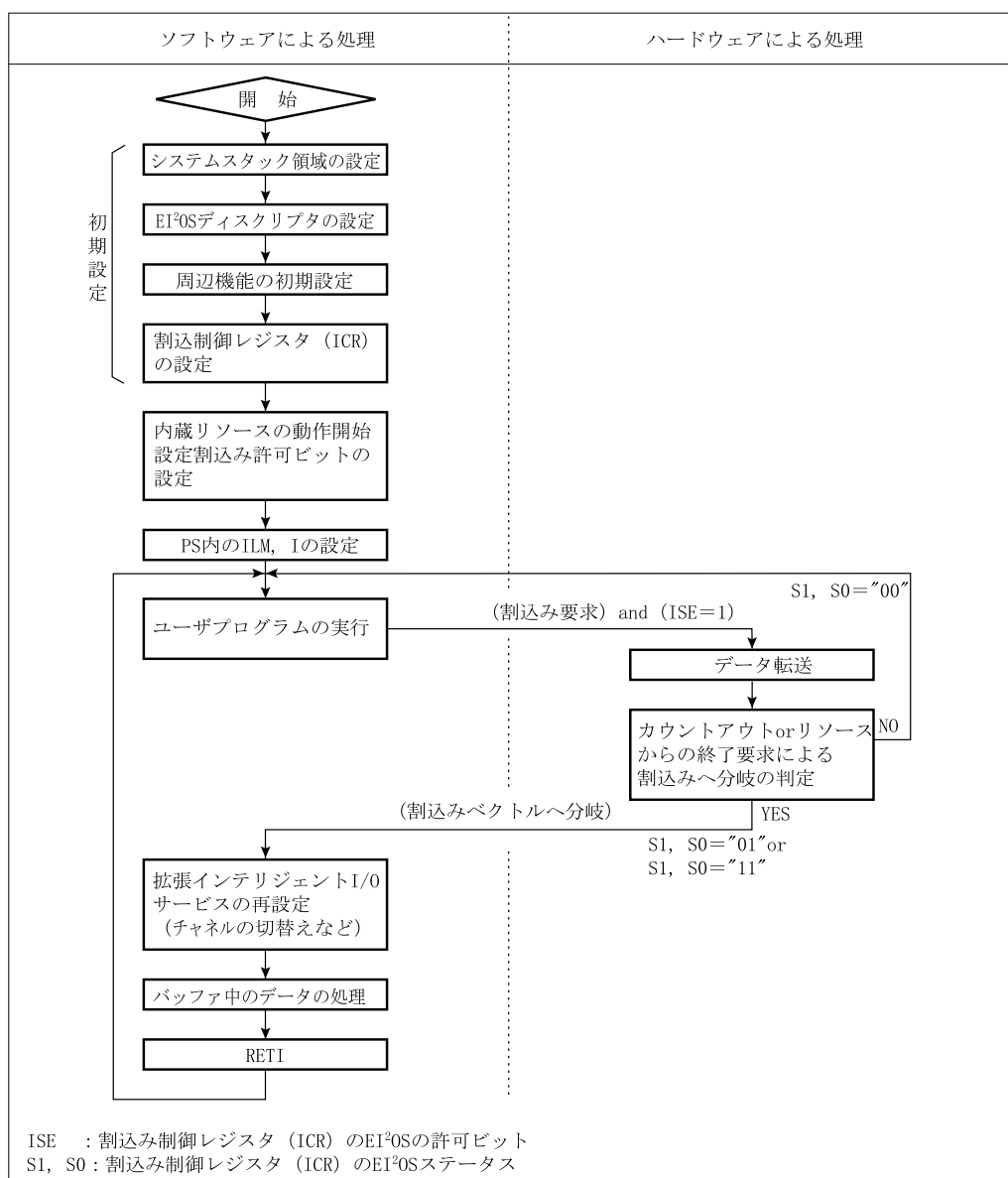


図 3.6-8 拡張インテリジェントI/Oサービス (EI²OS) 使用手順

3.6.6 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次の要因によって変化します。

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

また、EI²OSによるデータ転送終了時には、ハードウェア割込みが起動するため、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) の処理時間 (1回の転送時間)

データ転送継続時 (停止条件が成立しない場合)

データ転送継続時のEI²OS処理時間は、EI²OSステータスレジスタ (ISCS) の設定によって、表 3.6-9のようになります。

表 3.6-9 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 (ISCS, SE) の設定		"0" に設定		"1" に設定	
I/Oアドレスポインタの設定		固定	更新	固定	更新
BAPアドレス更新 / 固定 選択ビット (BF) 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。)

さらに、表 3.6-10に示すように、EI²OS実行時の条件によって補正が必要です。

表 3.6-10 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によるデータ転送終了時には、ハードウェア割込みが起動するため、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、次の式で算出されます。

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

割込みハンドリング時間は、スタックポインタの指しているアドレスによって異なります。表 3.6-11に、割込みハンドリング時間の補正值（Z）を示します。

表 3.6-11 割込みハンドリング時間の補正值（Z）

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

周辺機能（I/O）からの終了要求による終了時

周辺機能（I/O）からの終了要求で、EI²OSによるデータ転送を途中で終了した場合（ICR：S1, S0 = 11），データ転送は行わず、ハードウェア割込みを起動します。この場合のEI²OS処理時間は、次の式で算出されます。式中のZは、割込みハンドリング時間の補正值（表 3.6-11を参照）です。

途中終了時のEI²OS処理時間 = 36 + 6 × Z マシンサイクル

参考：

1マシンサイクルは、マシクロック（ ）の1クロック周期に相当します。

3.7 例外処理割込み

F²MC-16LXでは、未定義命令の実行により、例外処理が行われます。

例外処理は、基本的には割込みと同じものであり、命令の境目で例外事項が発生したことを検出した段階で、通常処理を中断して例外処理を行います。一般的に、例外処理は予想外の動作を行った結果で発生するものですので、デバッグ時や緊急時の復旧ソフトウェアの起動などのみに使用することを推奨します。

未定義命令の実行による例外発生

F²MC-16LXでは、命令マップで定義していないコードはすべて未定義命令として扱います。未定義命令を実行すると、ソフトウェア割込み命令の"INT 10"と等価な処理を行います。すなわち、AL,AH,DPR,DTB,ADB,PCB,PC,PSの内容をシステムスタックに退避し,コンディションコードレジスタ(CCR)内のIフラグを"0"に設定し,Sフラグを"1"に設定し,割込み番号10のベクタで示されるルーチンへ分岐します。スタックへ退避したプログラムカウンタ(PC)の値は、未定義命令を格納しているアドレスそのものです。2バイト以上の命令コードでは、未定義であることが識別できたコードを格納しているアドレスになります。このため、RETI命令で復帰させることは可能ですが、再び例外が発生するため、意味がありません。

第4章 クロックとリセット

この章では、クロックとリセットの機能と動作について説明します。

- 4.1 クロック発生部
- 4.2 クロック供給マップ
- 4.3 リセット要因
- 4.4 リセット解除後の動作

4.1 クロック発生部

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

クロック発生部に関する注意事項

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

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

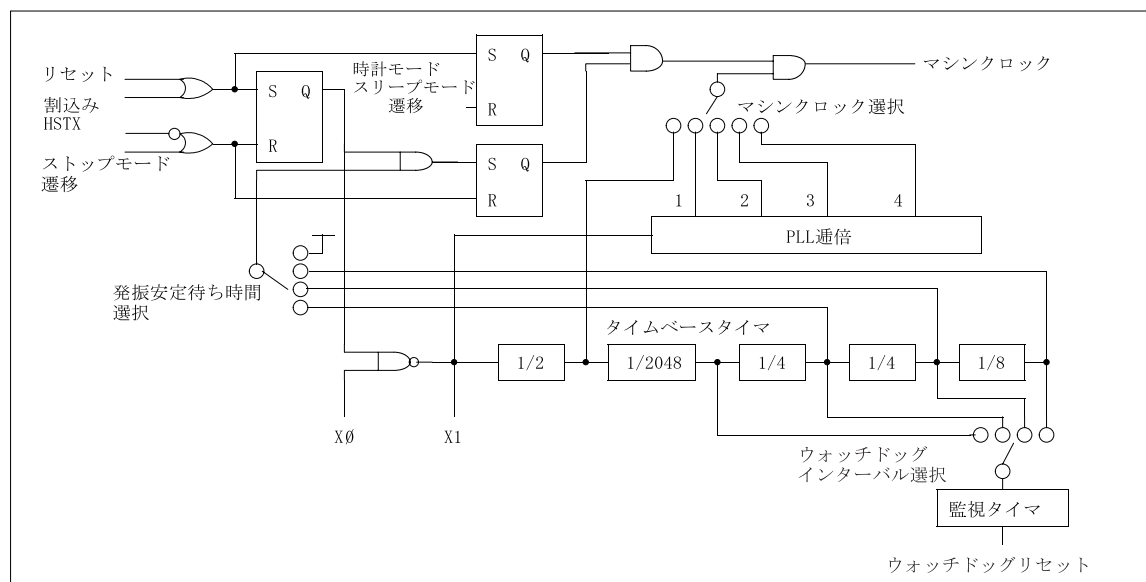


図 4.1-1 クロック発生回路ブロックダイヤグラム

4.2 クロック供給マップ

図 4.2-1に、クロック供給マップを示します。

クロック供給マップ

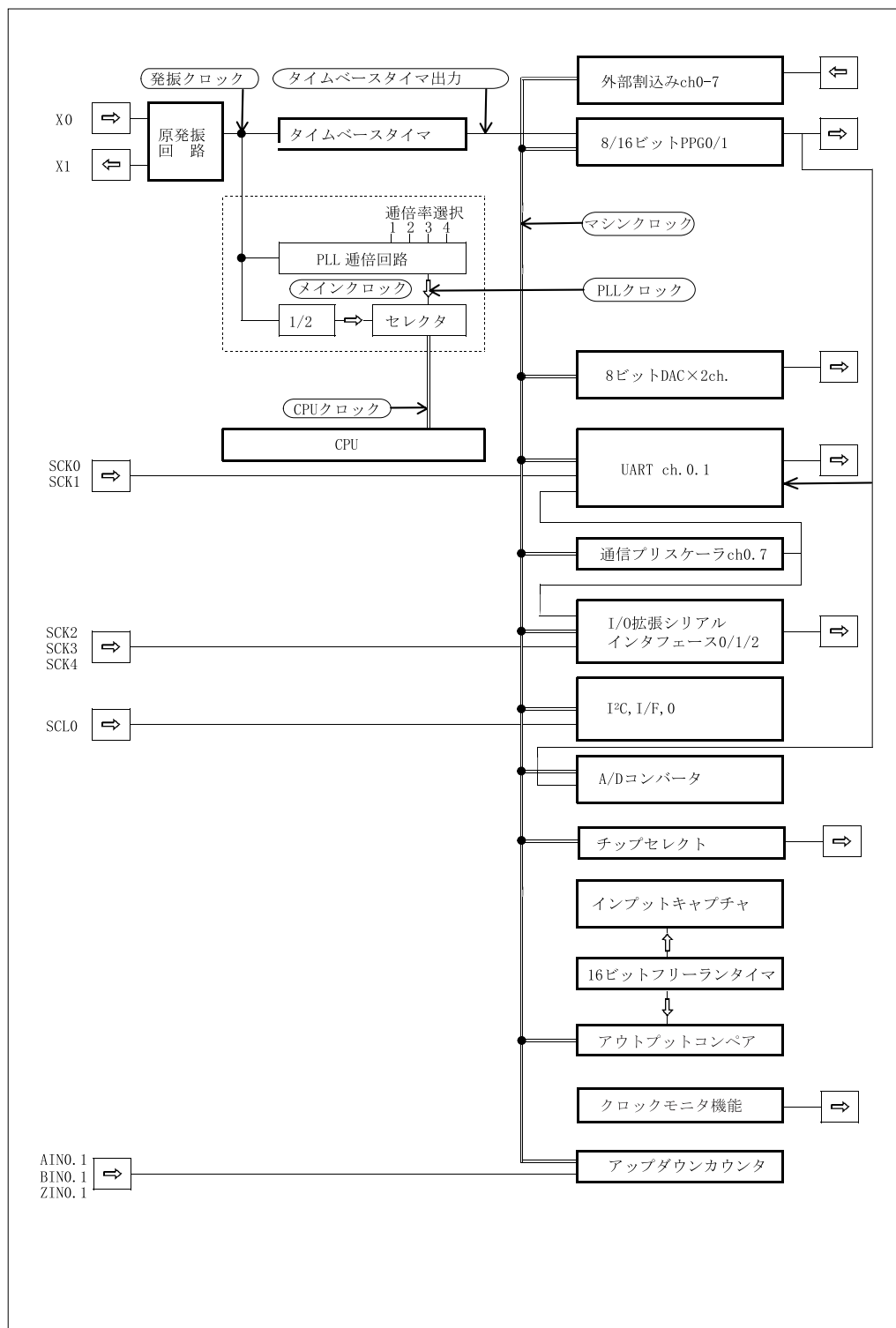


図 4.2-1 クロック供給マップ

4.3 リセット要因

リセットの要因には、以下に示す5種類があります。

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

リセット要因

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

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

< 注意事項 >

ストップモード以外は外部リセット入力を内部クロックでサンプルしていますので、外部から供給するクロックを停止した状態ではリセット入力は受け付けられません。

外部バスを使用している場合、リセット要因が発生するとリセット中にデバイスが発生するアドレスは不定となります。RDX、WRXなどの外部バスアクセス用の信号はすべてインアクティブになります。

表 4.3-1 リセット要因

リセット	発生要因	マシナクロック	ウォッチドッグタイマ	発振安定待ち
パワーオン	電源立ち上げ時	メインクロック	停止	あり
ハードウェアスタンバイ	HSTX端子への"L"レベル入力	メインクロック	停止	あり
ウォッチドッグタイマ	ウォッチドッグタイマオーバフロー	メインクロック	停止	あり
外部端子	RSTX端子への"L"レベル入力	前の状態を保持	前の状態を保持	なし
ソフトウェア	STBYC中のRSTビット"0"にライト	前の状態を保持	前の状態を保持	なし

- ・ストップ、ハードウェアスタンバイモード時のリセット入力は、リセット要因に関係なく発振安定待ち時間を取ります。
- ・パワーオンリセットの発振安定待ち時間は、評価品が原発振 2^{18} カウント固定で、FLASH品/MASK品が原発振 2^{17} カウント固定です。それ以外の発振安定待ち時間は、クロック選択レジスタ(CKSCR)のWS1/WS0により決定されます。

それぞれのリセット要因には対応したフリップフロップがあります。これらの内容はウォッチドッグタイマ制御レジスタを読み出すことで得ることができますので、リセット解除後にリセット発生要因を識別する必要がある場合には、ウォッチドッグタイマ制御レジスタを読み出した値をソフトウェアで処理した上で、適切なプログラムへ分岐するようにしてください。

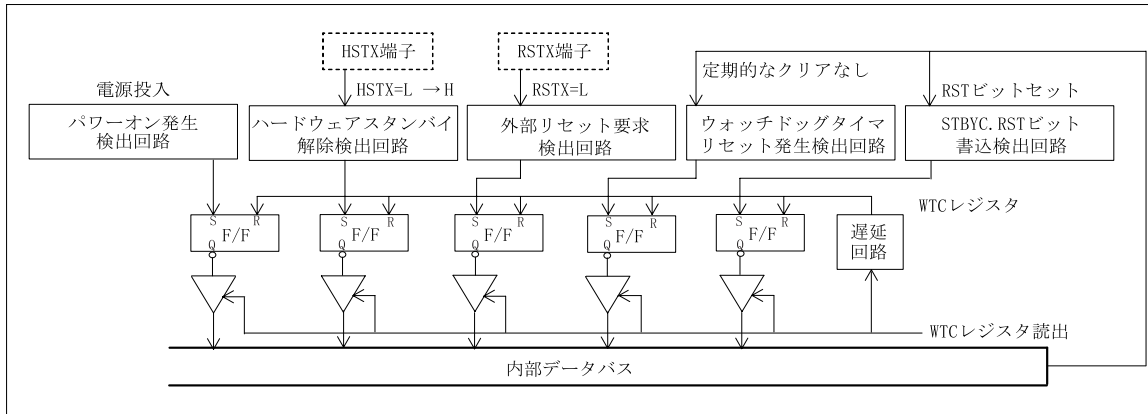


図 4.3-1 リセット要因ビットのブロックダイアグラム

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

ただし、パワーオンリセットの場合だけは別で、PONRビットが"1"である場合はそれ以外のリセット要因は不定となります。このため、PONRビットが"1"の場合はそれ以外のリセット要因ビットの内容は無視するようにソフトウェアを作成してください。

表 4.3-2 リセット要因ビットの内容とリセット要因の対応

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

*：前の値を保持します。

< 注意事項 >

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

ウォッチドッグタイマ制御レジスタの構成およびリセット要因ビットについては、「第10章 ウォッチドッグタイマ」を参照してください。

4.4 リセット解除後の動作

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

リセット解除後の動作

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

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

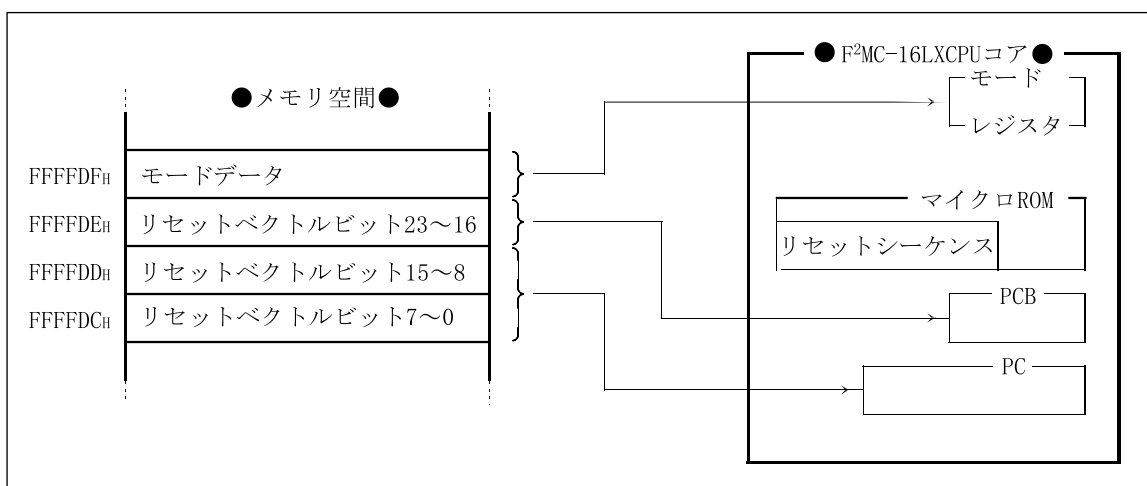


図 4.4-1 リセットベクトルとモードデータの格納場所と格納先

< 注意事項 >

モード・レジスタはリセット直後不定になります。必ず書き込みが行われるよう、メモリ空間に任意のモード・データを格納しておいてください。

第5章 低消費電力制御回路

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

- 5.1 低消費電力制御回路の概要
- 5.2 低消費電力制御回路のブロックダイアグラム
- 5.3 低消費電力制御回路のレジスタ
- 5.4 クロック選択の状態遷移

5.1 低消費電力制御回路の概要

低消費電力制御回路は、動作モードとして主に低消費電力モードを用います。また、レジスタのビット設定によりCPU間欠動作機能や発振安定待ち時間の設定を行うことができます。

全体のブロックダイアグラム中では、クロック制御回路の一部となります（「1.4 MB90570シリーズのブロックダイアグラム」を参照）。

低消費電力制御回路の動作モード

動作モードとして、PLLクロックモード・PLLスリープモード・PLL時計モード・擬似時計モード・メインクロックモード・メインスリープモード・メイン時計モード・メインストップモード・サブクロックモード・サブスリープモード・サブ時計モード・サブストップモード・ハードウェアスタンバイモードがあり、PLLクロックモード以外の動作モードが低消費電力モードになります。

CPU間欠動作機能

CPU間欠動作機能は、レジスタ、内蔵メモリ、内蔵リソース、アクセスを行うときにCPUに供給するクロックを一定期間停止させ、内蔵バスサイクルの起動を遅らせる機能であり、内蔵リソースに高速クロックを供給したままCPUの実行速度を下げることで、低消費電力で処理が行えます。低消費電力モード制御レジスタ(LPMCR)内のCG1,0ビットにより、CPUに供給するクロックの一時停止サイクル数の選択を行います。

なお、外部バス動作そのものは、リソースと同じクロックを使用して行います。

また、CPU間欠動作機能を使用した場合の命令実行時間の算出は、レジスタ、内蔵メモリ、内蔵リソース、アクセスを行う回数に、一時停止サイクル数を掛けた補正值を通常の実行時間に加えることで、求めることができます。

メインクロック発振安定待ち時間設定

クロック選択レジスタ(CKSCR)内のWS1,0ビットにより、ストップモード、およびハードウェアスタンバイモードの解除時のメインクロック発振安定待ち時間の選択を行います。X0, X1端子に接続する発振回路、発振素子の種類や特性に応じて発振安定待ち時間を選択してください。

パワーオンリセット以外のリセットは本ビットを初期化しません。パワーオンリセット発生時は本ビットを"11"に初期化します。そのため、パワーオン時のメインクロック発振安定待ち時間は、FLASH品/MASK品が原発振の約 2^{17} カウント評価品が原発振の約 2^{18} カウントになります。

マシクロックの切換え

メインクロック・PLLクロックの切換え

クロック選択レジスタ(CKSCR)内のMCSビットへの書込み動作により、メインクロックとPLLクロックの切換えを行います。

MCSビットを"1"から"0"に書き換えた場合には、PLLクロックの発振安定待ち時間（ 2^{12} マシクロック）後、メインクロックからPLLクロックに切り換わります。

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

MCSビットを書き換えても即座にマシクロックの切換えが行われないので、マシクロックに依存するリソースを操作する場合には、クロック選択レジスタ(CKSCR)内のMCMビットを参照してマシクロックの切換えが行われたことを

確認したのちに、リソースの操作を行ってください。

メインクロック・サブクロックの切換え

クロック選択レジスタ(CKSCR)内のSCSビットへの書込み動作により、発振クロックとサブクロックの切換えを行います。

SCSビットを"1"から"0"に書き換えた場合には、サブクロックのエッジが検出されたタイミングで発振クロックからサブクロックに切り換わります。

SCSビットを"0"から"1"に書き換えた場合には、発振クロックの発振安定待ち時間後に、サブクロックから発振クロックに切り換わります。

SCSビットを書き換えても即座にマシクロックの切換えが行われないので、マシクロックに依存することを確認した後に、リソースの操作を行ってください。

< 注意事項 >

電源を投入した場合、またはハードウェアスタンバイモード、ストップモードを解除した場合は、サブクロック発振安定待ち時間(約2秒)が発生します。この間に、メインクロックモードからサブクロックモードに切り換えた場合は、発振安定待ち時間が発生します。

マシクロック初期化

クロック選択レジスタ(CKSCR)内のMCSビットおよびSCSビットは、外部端子、低消費電力モード制御レジスタ(LPMCRR)内のRSTビットによるリセットでは、初期化されません。その他のリセットでは、"1"に初期化されます。

PLLクロック逡倍機能

PLLクロックの逡倍率は、CS1, 0ビットにより、2, 4, 6, 8逡倍の中から選択することができます。なお、マシクロックとしては、これらのクロックを2分周してから使用します。

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のブロックダイアグラムを示します。

低消費電力制御回路のブロックダイアグラム

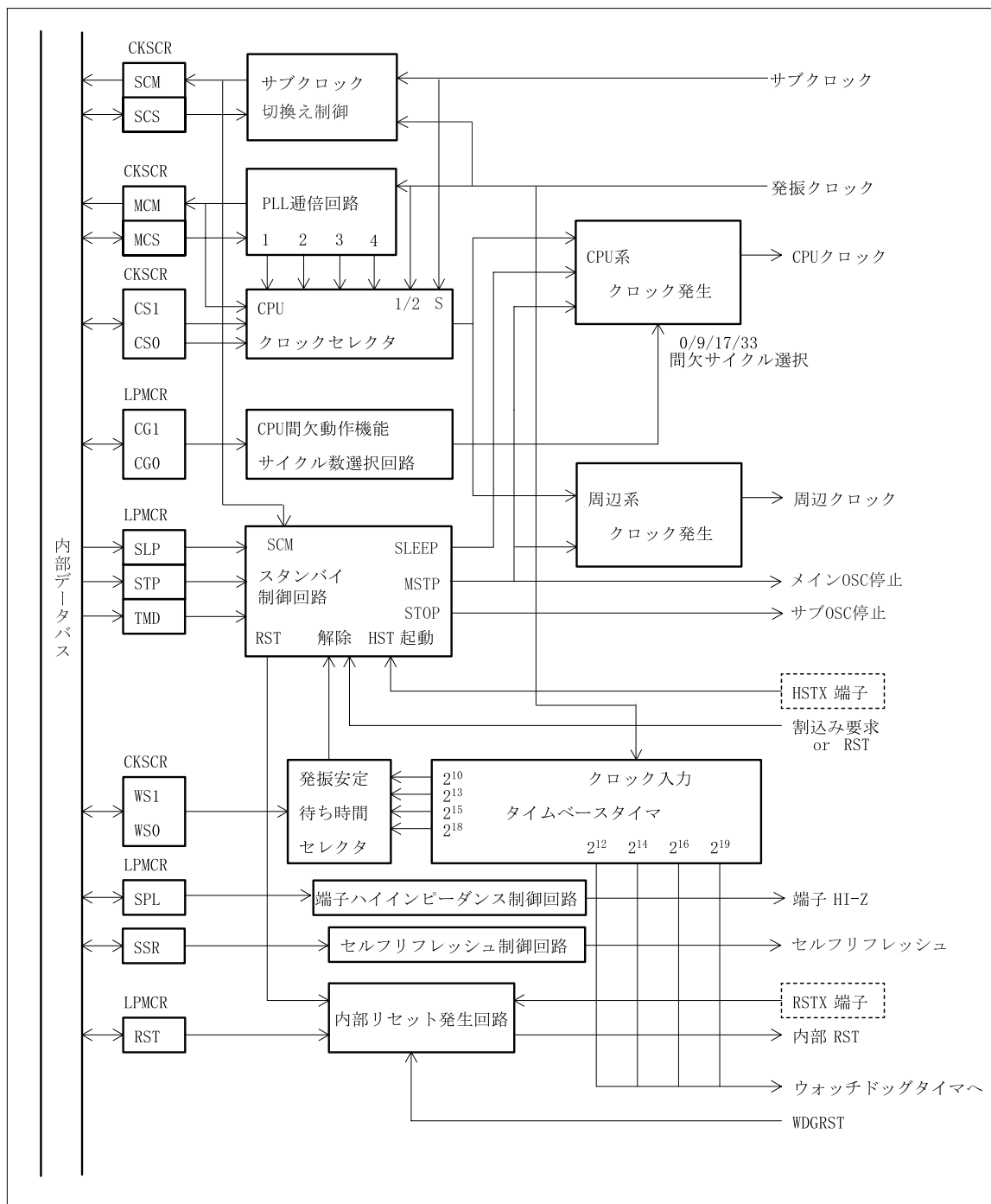


図 5.2-1 低消費電力制御回路のブロックダイアグラム

5.3 低消費電力制御回路のレジスタ

低消費電力制御回路のレジスタには、次の2種類があります。

- ・低消費電力モード制御レジスタ
- ・クロック選択レジスタ

低消費電力制御回路のレジスタ

低消費電力モード制御レジスタ								7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス : 0000A0H								STP	SLP	SPL	RST	TMD	CG1	CG0	SSR	LPMCR
リード/ライト ⇒								(W)	(W)	(R/W)	(W)	(W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒								(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)	
クロック選択レジスタ								15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス : 0000A1H								SCM	MCM	WS1	WS0	SCS	MCS	CS1	CS0	CKSCR
リード/ライト ⇒								(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒								(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	

図 5.3-1 低消費電力制御回路のレジスタ

5.3.1 低消費電力モード制御レジスタ(LPMCR)

低消費電力モード制御レジスタ(LPMCR) のレジスタ配置やビットの機能について説明します。

低消費電力モード制御レジスタ(LPMCR)

低消費電力モード制御レジスタ	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス : 0000A0H	STP	SLP	SPL	RST	TMD	CG1	CG0	SSR	LPMCR
リード/ライト ⇒	(W)	(W)	(R/W)	(W)	(W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)	

図 5.3-2 低消費電力モード制御レジスタ(LPMCR)

【ビット7】STP

"1"を書き込むことにより擬似時計モード(CKSCR, MCS=0 & SCS=1)またはストップモード(CKSCR : MCS=1 or SCS=0) に遷移します。"0"の書き込みでは動作に影響はありません。リセット, 時計解除またはストップ解除で"0"にクリアされます。書き込みのみ可能なビットです。読出し値は, 常に"0"です。

【ビット6】SLP

"1"を書き込むことによりスリープモードに遷移します。"0"の書き込みでは動作に影響はありません。リセット, またはスリープ解除, またはストップ解除で"0"にクリアされます。

STPビットとSLP ビットに同時に"1"を書き込んだ場合, 擬似時計モードまたはストップモードに遷移します。書き込みのみ可能なビットです。読出し値は, 常に"0"です。

【ビット5】SPL

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

【ビット4】RST

"0"を書き込むことにより3マシンサイクルの内部リセット信号を発生します。"1"の書き込みでは動作に影響はありません。読出し値は常に"1"です。

【ビット3】TMD

"0"を書き込むことにより時計モードに遷移します。"1"の書き込みでは動作に影響はありません。リセット, 時計解除, またはストップ解除で"1"にクリアされます。書き込みのみ可能なビットです。読出し値は, 常に"1"です。

【ビット2,1】CG1,CG0

CPU間欠動作機能のクロック一時停止サイクル数を設定します。

パワーオン，ハードウェアスタンバイ，ウォッチドッグによるリセットにより"00"に初期化されます。その他のリセット要因によるリセットでは初期化されません。読み書き可能なビットです。

表 5.3-1に，CGビット設定を示します。

表 5.3-1 CGビット設定

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

【ビット0】SSR

"1"の場合，スリープ（メイン・PLL）モード，時計モード，およびストップモード時にDRAMC のセルフリフレッシュ制御を行います。リセットで"0"にクリアされます。読み書き可能なビットです。

低消費電力モード制御レジスタ(LPMCR)へのアクセス

低消費電力モード制御レジスタの設定により，低消費電力モードへ遷移しますが，この場合に使用する命令は，表 5.3-2に示す命令を使用してください。この命令以外の命令で，低消費電力モードへ遷移した場合は，誤動作の原因となります。低消費電力モードへの遷移以外の設定を行う場合は，使用する命令に制限はありません。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，偶数アドレスで書込みが行われるようにしてください。奇数アドレスの書込みで低消費電力モードへ遷移した場合は，誤動作の原因となります。

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 dir,A	MOV eam,A
MOV @RLi+disp8,A	MOV P sddr24,A		
MOVW io,#imm16	MOVW dir,#imm16	MOVW eam,#imm16	MOVW eam,Rwi
MOVW io,A	MOVW dir,A	MOVW addr16,A	MOVW eam,A
MOVW @RLi+disp8,A	MOV PW addr24,A		
SETB io:bp	SETB dir:bp	SETB addr16:bp	

5.3.2 クロック選択レジスタ(CKSCR)

クロック選択レジスタ(CKSCR) のレジスタ配置やビットの機能について説明します。

クロック選択レジスタ(CKSCR)

クロック選択レジスタ	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス : 0000A1H	SCM	MCM	WS1	WS0	SCS	MCS	CS1	CS0	CKSCR
リード/ライト⇒	(R)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(1)	(1)	(1)	(1)	(1)	(1)	(0)	(0)	

図 5.3-3 クロック選択レジスタ(CKSCR)

【ビット15】SCM

マシクロックとして、発振クロックまたはサブクロックのどちらが選択されているかを表示するビットです。"0"の場合、サブクロックが選択されていることを示し、"1"の場合、発振クロックが選択されていることを示します。SCS = 1で、かつSCM = 1ならば、メインクロック発振安定待ち時間中であることを示します。

メインクロックからサブクロックに切り換えた場合は(CKSCR:SCM=1 0)、サブクロックに同期を取って(約130μs)サブクロックモードに切り換わります。

サブクロックモードからメインクロックに切り換えた場合(CKSCR:SCS=1 0)、メインクロックの発振安定待ち時間が発生した後、メインクロックモードに切り換わります。タイムベースタイマは自動的にクリアされます。

【ビット14】MCM

マシクロックとして、メインクロックまたはPLLクロックのどちらが選択されているかを表示するビットです。"0"の場合、PLLクロックが選択されていることを示し、"1"の場合、メインクロックが選択されていることを示します。MCS = 0で、かつMCM = 1ならば、PLLクロック発振安定待ち時間中であることを示します。なお、PLLクロックの発振安定待ち時間は、 2^{13} 発振クロックサイクル固定です。

【ビット13,12】WS1,WS0

ストップモード、ハードウェアスタンバイモード解除時のメインクロックの発振安定待ち時間を設定します。

パワーオンリセットにより"11"に初期化され、その他のリセット要因によるリセットでは初期化されません。読み書き可能なビットです。

表 5.3-3に、WSビット設定を示します。

表 5.3-3 WSビット設定

WS1	WS0	発振安定待ち時間(原発振4MHz時)
0	0	発振安定待ち時間なし
0	1	約 2.05 ms (原発振 2^{13} カウント)
1	0	約 8.19 ms (原発振 2^{15} カウント)
1	1	約 65.54 ms (原発振 2^{18} カウント) [初期値]

< 注意事項 >

電源を投入した場合、またはハードウェアスタンバイモード、ストップモードを解除した場合は、サブクロック発振安定待ち時間(約2秒)が発生します。この間に、メインクロックモードからサブクロックモードに切り換えた場合は、発振安定待ち時間が発生します。

【ビット11】SCS

マシニングロックとして、発振クロックまたはサブクロックのどちらを選択するかを指示するビットです。"0"を書き込むことにより、サブクロックを選択します。"1"を書き込むことにより、メインクロックを選択します。"1"の場合に、"0"を書き込むと、サブクロックに同期をとって(約130 μ s)サブクロックモードに切り換わります。"0"の場合に、"1"を書き込むと、メインクロックの発振安定待ち時間を発生した後、メインクロックモードに切り換わり、自動的にタイムベースタイマがクリアされます。SCSビットおよびMCSビットがともに"0"の場合には、SCSが優先され、サブクロックが選択されます。

【ビット10】MCS

マシニングロックとして、メインクロックまたはPLLクロックのどちらを選択するかを指示するビットです。"0"を書き込むことにより、PLL クロックを選択します。"1"を書き込むことにより、メインクロックを選択します。"1"の場合に、"0"を書き込むと、PLL クロックの発振安定待ち時間を発生するために、自動的にタイムベースタイマがクリアされます。なお、PLLクロックの発振安定待ち時間は、 2^{13} メインクロックサイクル固定です。

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

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

< 注意事項 >

MCSビットが、"1"の場合に"0"を書込む場合には、タイムベースタイマ制御レジスタ(TBTC)のTBIEビットまたは、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がマスクされている状態で行うようにしてください。

また、MCSビットに"1"を書込んでから8マシニングサイクルの間はこのビットに"0"が書込めない場合があります。8マシニングサイクル以上待ってから書込んでください。

【ビット9,8】CS1,CS0

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

MCSビットが"0"の場合には書込みが抑止されます。いったん、MCSビットを"1"にした(メインクロックモード)後に、CSビットを書き換えてください。読み書き可能なビットです。

表 5.3-4に、CSビットの設定を示します。

表 5.3-4 CSビットの設定

CS1	CS0	マシニングロック(原発振4MHz時)
0	0	4 MHz(動作周波数=OSC発振周波数)
0	1	8 MHz(動作周波数=OSC発振周波数×2)
1	0	12 MHz(動作周波数=OSC発振周波数×3)
1	1	16 MHz(動作周波数=OSC発振周波数×4)

5.4 クロック選択の状態遷移

図 5.4-1, 図 5.4-2にクロック選択の状態遷移を示します。

クロック選択の状態遷移

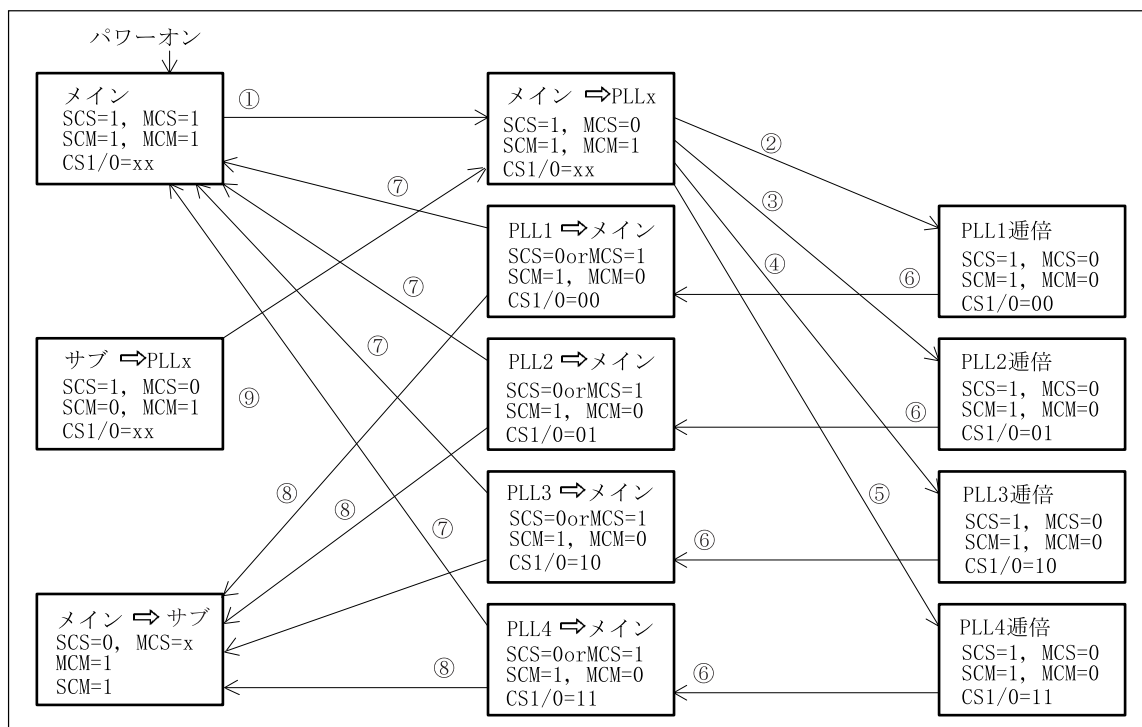


図 5.4-1 クロック選択の状態遷移図1

MCSビットクリアおよびSCSビットセット

PLLクロック発振安定待ち終了 & CS1/0=00

PLLクロック発振安定待ち終了 & CS1/0=01

PLLクロック発振安定待ち終了 & CS1/0=10

PLLクロック発振安定待ち終了 & CS1/0=11

MCSビットセットまたはSCSビットクリア

PLLクロックとメインクロックの同期タイミング & SCS=1

PLLクロックとメインクロックの同期タイミング & SCS=0

メインクロック発振安定待ち終了 & MCS=0

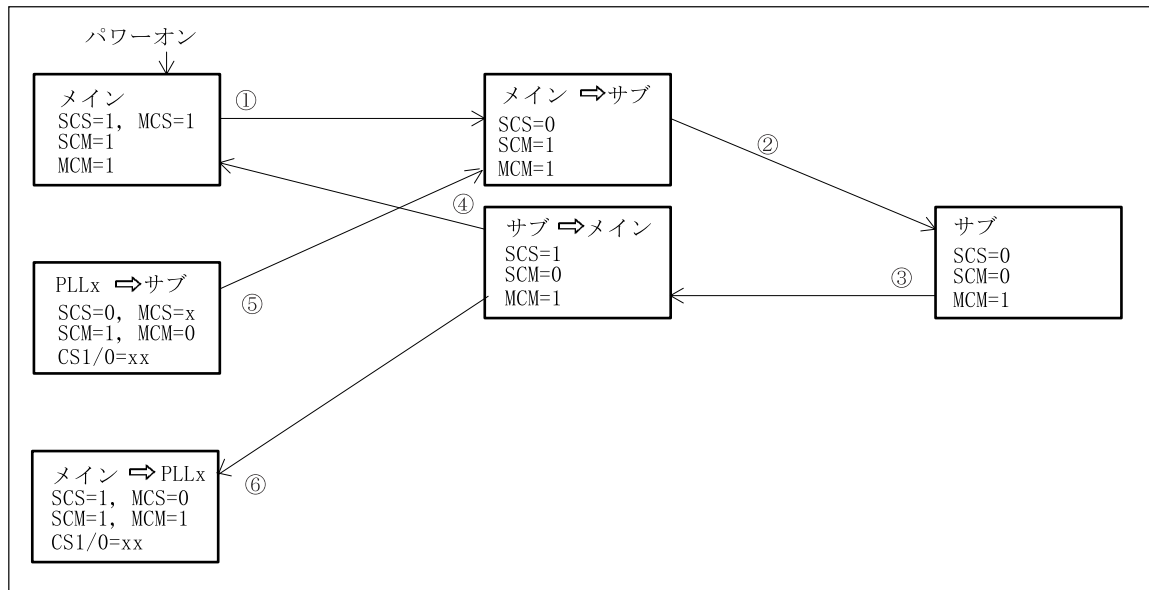


図 5.4-2 クロック選択の状態遷移図2

SCSビットクリア

サブクロックのエッジ検出タイミング

SCSビットセット

メインクロック発振安定待ち終了 & MCS=1

PLLクロックとメインクロックの同期タイミング & SCS=0

メインクロック発振安定待ち終了 & MCS=0

第6章 低消費電力モード

この章では、低消費電力モードの機能と動作について説明します。

- 6.1 低消費電力モード
- 6.2 低消費電力モードの遷移条件
- 6.3 低消費電力モード状態遷移図

6.1 低消費電力モード

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

低消費電力モード

メインクロックモード・メインスリープモード

メインクロック（メインOSC発振クロック、）およびサブクロック（サブOSC発振クロック）で動作させるモードです。動作クロックとしてメインクロックを2分周したクロックを使用し、時計用クロックとしてサブクロック（サブOSC発振クロックの4分周）を使用し、PLLクロック（VCO発振クロック）を停止させるモードです。

サブクロックモード・サブスリープモード

サブクロックのみで動作させるモードです。動作クロックとしてサブクロックの4分周を使用し、メインクロック、およびPLLクロックを停止させるモードです。

PLL スリープモード・メインスリープモード

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

擬似時計モード

時計タイマ、およびタイムベースタイマのみを動作させるモードです。

PLL時計モード・メイン時計モード・サブ時計モード

時計タイマのみを動作させるモードです。サブクロックの4分周で動作させるモードで、メインクロック、およびPLLクロックを停止させるモードです。なお、PLL時計モード・メイン時計モード・サブ時計モードの違いは、割込みによる復帰時の動作モードがそれぞれPLLクロックモード・メインクロックモード・サブクロックモードであることであり、時計モードの動作は違いありません。

メインストップモード・サブストップモードとハードウェアスタンバイモード

発振を停止させるモードであり、もっとも低消費電力でデータを保持できます。なお、メインストップモード・サブストップモードの違いは、割込みによる復帰時の動作モードがそれぞれメインクロックモード・サブクロックモードであることであり、ストップモードの動作は違いありません。

CPU間欠動作機能

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

PLLクロックの逡倍率は、CS1、0ビットにより、2、4、6、8逡倍の中から選択することができます。なお、マシクロックとしては、これらのクロックを2分周してから使用します。

低消費電力モード動作状態

表 6.1-1に、各動作モードにおけるチップ各部の状態を示します。

表 6.1-1 低消費電力モード動作状態

状態	遷移条件	サブ発振	メイン発振	クロック	CPU	周辺	端子	解除方法
サブクロック	SCS=0 MCS=x	動作	停止	動作	動作	動作	動作	リセット
サブスリープ	SCS=0 MCS=x SLP=1	動作	停止	動作	停止	動作	動作	リセット 割込み
メインスリープ	SCS=1 MCS=1 SLP=1	動作	動作	動作	停止	動作	動作	リセット 割込み
PLLスリープ	SCS=1 MCS=0 SLP=1	動作	動作	動作	停止	動作	動作	リセット 割込み
擬似時計 (SPL=0)	SCS=1 MCS=0 STP=1	動作	動作	停止	停止	停止	保持	リセット 割込み
擬似時計 (SPL=1)	SCS=1 MCS=0 STP=1	動作	動作	停止	停止	停止	HI-Z	リセット 割込み
時計 (SPL=0)	SCS=x MCS=x TMD=0	動作	停止	停止	停止	停止	保持	リセット 割込み
時計 (SPL=1)	SCS=x MCS=x TMD=0	動作	停止	停止	停止	停止	HI-Z	リセット 割込み
ストップ (SPL=0)	MCS=1 or SCS=0 STP=1	停止	停止	停止	停止	停止	保持	リセット 割込み
ストップ (SPL=1)	MCS=1 or SCS=0 STP=1	停止	停止	停止	停止	停止	HI-Z	リセット 割込み
ハードウェア スタンバイ	HSTX=L	停止	停止	停止	停止	停止	HI-Z	HSTX=H

< 注意事項 >

電源を投入した場合、またはハードウェアスタンバイモード、ストップモードを解除した場合は、サブクロック発振安定待ち時間(約2秒)が発生します。この間に、メインクロックモードからサブクロックモードに切り換えた場合は、発振安定待ち時間が発生します。

6.1.1 スリープモード

スリープモードとは、CPU に供給するクロックのみを停止するモードで、CPUは停止し、周辺回路は動作を続けます。

スリープモードへの遷移

低消費電力モード制御レジスタ(LPMCR)内のSLPビットに"1"，TMDビットに"1"，STPビットに"0"を書き込むことによりスタンバイ制御回路をスリープモードに遷移します。

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

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

スリープモードの解除

スタンバイ制御回路は，リセット入力または割込みの発生によりスリープモードを解除します。リセット要因によりスリープモードを解除した場合は，スリープモードを解除した上でリセット状態になります。

スリープモード中に周辺回路および内蔵リソースから割込みレベルが7より強い割込み要求が発生すると，スタンバイ制御回路はスリープモードを解除します。スリープモードの解除後は，通常の割込み処理と同じ扱いとなり，コンディションコードレジスタ(CCR)内のIフラグ，インタラプトレベルマスケレジスタ(ILM)と割込み制御レジスタ(ICR)の設定により割込みが受け付けられるとき，CPUはスタンバイ割込み命令の次の割込み保留でない命令を実行後に割込み処理を実行します。割込みが受け付けられないときは，スリープモードに入れた命令の次の命令から処理を続行します。

6.1.2 擬似時計モード

擬似時計モードとは、原発振（メインおよびサブ）と時計タイマとタイムベースタイマ以外の動作を止めるモードであり、チップのほぼ全機能が停止します。

擬似時計モードへの遷移

クロック選択レジスタ(CKSCR)のSCSビットに"1"，MCSビットに"0"，低消費電力モード制御レジスタ(LPMCR)内のTMDビットに"1"，STPビットに"1"を書き込むことによりスタンバイ制御回路を擬似時計モードに遷移します。

また、擬似時計モード中のI/O端子を直前の状態に保持するの、ハイインピーダンス状態にするの、低消費電力モード制御レジスタ(LPMCR)内のSPLビットにより制御できます。

低消費電力モード制御レジスタ(LPMCR)内のSTP ビットに"1"を書き込んだときに割り込み要求が発生している場合は、スタンバイ制御回路は擬似時計モードに遷移しません。

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

擬似時計モードの解除

スタンバイ制御回路は、リセット入力または割り込みの発生により擬似時計モードを解除します。リセット要因により擬似時計モードを解除した場合は、擬似時計モードを解除した上で、リセット状態になります。

擬似時計モードからの復帰の場合、スタンバイ制御回路はまず擬似時計モードを解除し、そののちPLLクロック発振安定待ち状態に遷移します。擬似時計モードからの解除がリセット要因であったときには、リセットシーケンスはメインクロックを使用して行われます。

擬似時計モード中に周辺回路などから割り込みレベルが7より強い割り込み要求が発生すると、スタンバイ制御回路は擬似時計モードを解除します。擬似時計モードの解除後は、通常の割り込み処理と同じ扱いとなり、コンディションコードレジスタ(CCR)のIフラグ、インタラプトレベルマスケレジスタ(ILM)と割り込み制御レジスタ(ICR)の設定により割り込みが受け付けられるときはCPUはスタンバイ書き込み命令の次の割り込み保留でない命令を実行後に割り込み処理を実行し、割り込みが受け付けられないときは擬似時計モードに入る前の次の命令から処理を続行します。

< 注意事項 >

- MB90V570, MB90F574, MB90573, MB90574の場合
外部割り込みで擬似時計モードを解除する場合は、外部割り込みの割り込み要求を"H"レベルに設定してください。"L"レベルの割り込み要求設定では、誤動作を起こす可能性があります。
また、エッジの割り込み要求では、擬似時計モードの解除はできません。
- MB90V570A, MB90F574A, MB90574Cの場合
擬似時計モードに入る前に設定した外部割り込みの割り込み要求を入力することにより、擬似時計モードを解除することができます。ch2～ch7の割り込み要求としては、"H"レベル、または"L"レベルが選択できます。ch0, ch1の割り込み要求としては、立上りエッジ、または立下りエッジが選択できます。

6.1.3 時計モード

時計モードとは、サブ原発振と時計タイマ以外の動作を止めるモードであり、チップのほぼ全機能が停止します。

時計モードへの遷移

低消費電力モード制御レジスタ(LPMCR)内のTMDビットに"0"を書き込むことによりスタンバイ制御回路を時計モードに遷移します。

また、時計モード中のI/O端子を直前の状態に保持するのか、ハイインピーダンス状態にするのかを、低消費電力モード制御レジスタ(LPMCR)内のSPLビットにより制御できます。

低消費電力モード制御レジスタ(LPMCR)内のTMDビットに"1"を書き込んだときに割込み要求が発生している場合は、スタンバイ制御回路は時計モードに遷移しません。

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

時計モードの解除

スタンバイ制御回路は、リセット入力または割込みの発生により時計モードを解除します。リセット要因により時計モードを解除した場合は、時計モードを解除した上で、リセット状態になります。

サブ時計モードからの復帰の場合、スタンバイ制御回路は時計モードを解除し、即座にサブクロックモードに遷移します。サブ時計モードからの解除がリセット要因であったときには、リセットシーケンスはサブクロックを使用して行われます。

メイン時計モード、およびPLL時計モードからの復帰の場合、スタンバイ制御回路はまず時計モードを解除し、そののちメインクロック発振安定待ち状態に遷移します。時計モードからの解除がリセット要因であったときには、リセットシーケンスはサブクロックを使用して行われます。

時計モード中に周辺回路などから割込みレベルが7より強い割込み要求が発生すると、スタンバイ制御回路は時計モードを解除します。時計モードの解除後は、通常の割込み処理と同じ扱いとなり、コンディションコードレジスタ(CCR)内のIフラグ、インタラプトレベルマスクレジスタ(ILM)と割込み制御レジスタ(ICR)の設定により割込みが受け付けられるときはCPUはスタンバイ書込み命令の次の割込み保留でない命令を実行後に割込み処理を実行し、割込みが受け付けられないときは時計モードに入る前の次の命令から処理を続行します。

< 注意事項 >

- MB90V570, MB90F574, MB90573, MB90574の場合

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

また、エッジの割込み要求では、時計モードの解除はできません。

- MB90V570A, MB90F574A, MB90574Cの場合

時計モードに入る前に設定した外部割込みの割込み要求を入力することにより、時計モードを解除することができます。ch2～ch7の割込み要求としては、"H"レベル、または"L"レベルが選択できます。ch0, ch1の割込み要求としては、立上りエッジ、または立下りエッジが選択できます。

6.1.4 ストップモード

ストップモードとは、原発振（メインおよびサブ）を止めるモードであり、チップの全機能が停止します。したがって、もっとも低消費電力でデータを保持することができます。

ストップモードへの遷移

クロック選択レジスタ(CKSCR)内のSCSビットに"0"，またはMCSビットに"1"，低消費電力モード制御レジスタ(LPMCR)内のSTPビットに"1"を書き込むことによりスタンバイ制御回路をストップモードに遷移します。

また，ストップモード中のI/O端子を，直前の状態に保持するのか，ハイインピーダンス状態にするのか，低消費電力モード制御レジスタ(LPMCR)内のSPLビットにより制御できます。

低消費電力モード制御レジスタ(LPMCR)内のSTPビットに"1"を書き込んだときに割込み要求が発生している場合は，スタンバイ制御回路はストップモードに遷移しません。

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

ストップモードの解除

スタンバイ制御回路は，リセット入力，または割込みの発生によりストップモードを解除します。リセット要因によりストップモードを解除した場合は，ストップモードを解除した上で，リセット状態になります。

サブストップモードからの復帰の場合，スタンバイ制御回路はまずサブクロック発振安定待ち状態に遷移し，そののちストップモードを解除します。ストップモードからの解除がリセット要因であったときには，リセットシーケンスはサブクロック発振安定待ち時間経過後実行されます。

メインストップモードからの復帰の場合，スタンバイ制御回路はまずメインクロック発振安定待ち状態に遷移し，そののちストップモードを解除します。ストップモードからの解除がリセット要因であったときには，リセットシーケンスはメインクロック発振安定待ち時間経過後実行されます。

ストップモード中に周辺回路などから割込みレベルが7より強い割込み要求が発生すると，スタンバイ制御回路はストップモードを解除します。サブストップモードの解除後は，サブクロックの発振安定待ち時間を経過したあとで，通常の割込み処理と同じ扱いとなり，コンディションコードレジスタ(CCR)内のIフラグ，インタラプトレベルマスクレジスタ(ILM)と割込み制御レジスタ(ICR)の設定により割込みが受け付けられるときは，CPUはスタンバイ書込み命令の次の割込み保留でない命令を実行後に割込み処理を実行し，割込みが受け付けられないときはストップモードに入る前の次の命令から処理を続行します。

メインストップモードの解除後は，クロック選択レジスタ(CKSCR)内のWS1,0ビットで指定されたメインクロックの発振安定待ち時間を経過したあとで，通常の割込み処理と同じ扱いとなり，コンディションコードレジスタ(CCR)内のIフラグ，インタラプトレベルマスクレジスタ(ILM)と割込み制御レジスタ(ICR)の設定により割込みが受け付けられるときはCPUはスタンバイ書込み命令の次の割込み保留でない命令を実行後に割込み処理を実行し，割込みが受け付けられないときはストップモードに入る前の次の命令から処理を続行します。

< 注意事項 >

-
- MB90V570, MB90F574, MB90573, MB90574の場合
外部割込みでストップモードを解除する場合は、外部割込みの割込み要求を"H"レベルに設定してください。"L"レベルの割込み要求設定では、誤動作を起こす可能性があります。
また、エッジの割込み要求では、ストップモードの解除はできません。
 - MB90V570A, MB90F574A, MB90574Cの場合
ストップモードに入る前に設定した外部割込みの割込み要求を入力することにより、ストップモードを解除することができます。ch2～ch7の割込み要求としては、"H"レベル、または"L"レベルが選択できます。ch0, ch1の割込み要求としては、立上りエッジ、または立下りエッジが選択できます。
-

6.1.5 ハードウェアスタンバイモード

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

ハードウェアスタンバイモードへの遷移

HSTX端子を"L"レベルにすることにより、どんな状態からでもスタンバイ制御回路をハードウェアスタンバイモードに遷移することができます。

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

ハードウェアスタンバイモードの解除

ハードウェアスタンバイモードの解除は、HSTX端子によってのみ解除することができます。HSTX端子を"H"レベルにすると、スタンバイ制御回路はハードウェアスタンバイモードを解除し、内部リセット信号を有効にしたのち、発振安定待ち状態に遷移します。さらに、メインクロック発振安定待ち時間の経過後、スタンバイ制御回路は内部リセットを解除し、その結果、CPUはリセットシーケンスから実行を開始します。

6.2 低消費電力モードの遷移条件

低消費電力モードでは、クロック選択レジスタや低消費電力モードレジスタの設定条件によって各状態が遷移します。

低消費電力モードの遷移条件

表 6.2-1に、遷移条件の一覧を示します。

以下に、表 6.2-1中の記号の意味を説明します。

- MCS：MCSビット（クロック選択レジスタ）（MCS = 0で、PLLクロックモード選択）
- SCS：SCSビット（クロック選択レジスタ）（SCS = 0で、サブクロックモード選択）
- STP：STPビット（低消費電力モードレジスタ）（STP = 0で、ストップモード選択）
- SLP：SLPビット（低消費電力モードレジスタ）（SLP = 0で、スリープモード選択）
- TMD：TMDビット（低消費電力モードレジスタ）（TMD = 0で、時計モード選択）
- MCM：MCMビット（クロック選択レジスタ）（MCM = 0で、PLLクロック使用中）
- SCM：SCMビット（クロック選択レジスタ）（SCM = 0で、サブクロック使用中）
- SCD：サブクロック発振停止（SCD = 1で、サブクロック発振停止）
- MCD：メインクロック発振停止（MCD = 1で、メインクロック発振停止）
- PCD：PLLクロック発振停止（PCD = 1で、PLLクロック発振停止）

表 6.2-1 遷移条件一覧表（1/3）

遷移前の状態	遷移条件	遷移後の状態
パワーオン	01 メイン発振安定時間待ち終了	メインモード
メイン発振安定	05 メイン発振安定時間待ち終了	メインモード
メインモード	06 SCS=0 書込み	MS遷移モード
	07 SCS=1 ・ MCS=0 書込み	MP遷移モード
	31 TMD=1 ・ STP=0 ・ SLP=1 書込み	メインスリープ
	32 TMD=0 書込み	メイン時計遷移
	33 TMD=1 ・ STP=1 書込み	メインストップ
PLLモード	21 SCS=0 書込み	PS遷移モード
	20 SCS=1 ・ MCS=1 書込み	PM遷移モード
	59 TMD=1 ・ STP=0 ・ SLP=1 書込み	PLLスリープ
	58 TMD=0 書込み	PLL時計遷移P
	57 TMD=1 ・ STP=1 書込み	擬似時計遷移
サブモード	10 SCS=1 ・ MCS=1 書込み	SM遷移モード
	12 SCS=1 ・ MCS=0 書込み	SP遷移モード
	11 リセット起動	メイン発振安定
	42 TMD=1 ・ STP=0 ・ SLP=1 書込み	サブスリープ
	43 TMD=0 書込み	サブ時計
	44 TMD=1 ・ STP=1 書込み	サブストップ

表 6.2-1 遷移条件一覧表 (2/3)

遷移前の状態	遷移条件	遷移後の状態
PM遷移モード	13 PLL メイン切換えタイミング待ち終了	メインモード
	38 TMD=1 ・STP=0 ・SLP=1 書込み	PM遷移スリープ
	39 TMD=0 書込み & PLL メイン切換え待ち終了	メイン時計遷移
	40 TMD=1 ・STP=1 書込み & PLL メイン切換え待ち終了	メインストップ
SM遷移モード	02 メイン発振安定時間待ち終了	メインモード
	03 リセット起動 or 割込み	メイン発振安定
	04 SCS=0 書込み	サブモード
	27 TMD=1 ・STP=0 ・SLP=1 書込み	SM遷移スリープ
	28 TMD=0 書込み & メイン発振安定時間待ち終了	メイン時計
	29 TMD=1 ・STP=1 書込み & メイン発振安定時間待ち終了	メインストップ
MP遷移モード	16 PLL 発振安定時間待ち終了	PLLモード
	14 SCS=1 ・MCS=1 書込み	メインモード
	15 SCS=0 書込み	MS遷移モード
	68 TMD=1 ・STP=0 ・SLP=1 書込み	MP遷移スリープ
	70 TMD=0 書込み	PLL時計遷移M
	69 TMD=1 ・STP=1 書込み	擬似時計モード
SP遷移モード	17 メイン発振安定待ち終了	MP遷移モード
	18 MCS=1 書込み	SM遷移モード
	19 リセット起動	メイン発振安定
	75 TMD=1 ・STP=0 ・SLP=1 書込み	SP遷移スリープ
	76 TMD=0 書込み	PLL時計
	78 TMD=1 ・STP=1 書込み & メイン発振安定待ち終了	擬似時計モード
MS遷移モード	09 メイン サブクロック切換えタイミング待ち終了	サブモード
	08 リセット起動	メインモード
	51 TMD=1 ・STP=0 ・SLP=1 書込み	MS遷移スリープ
	52 TMD=0 書込み & メイン サブ切換え待ち終了	サブ時計
	53 TMD=1 ・STP=1 書込み & メイン サブ切換え待ち終了	サブストップ
PS遷移モード	23 PLL メインクロック切換えタイミング待ち終了	MS遷移モード
	22 SCS=1 書込み	PM遷移モード
	56 TMD=1 ・STP=0 ・SLP=1 書込み	PS遷移スリープ
メインスリープ	26 割込み or リセット起動	メインモード
SM遷移スリープ	24 メイン発振安定待ち終了	メインスリープ
	25 割込み or リセット起動	SM遷移モード
PM遷移スリープ	34 PLL メインクロック切換えタイミング待ち終了	メインスリープ
	35 割込み or リセット起動	PM遷移モード
PLLスリープ	63 割込み or リセット起動	PLLモード
MP遷移スリープ	66 PLL 発振安定待ち終了	PLLスリープ
	67 割込み or リセット起動	MP遷移モード
SP遷移スリープ	73 メイン発振安定待ち終了	MP遷移スリープ
	74 割込み or リセット起動	SP遷移モード
サブスリープ	46 割込み or リセット起動	サブモード
MS遷移スリープ	49 メイン サブクロック切換えタイミング待ち終了	サブスリープ
	50 割込み or リセット起動	MS遷移モード
PS遷移スリープ	54 PLL メインクロック切換えタイミング待ち終了	MS遷移スリープ
	55 割込み or リセット起動	PS遷移モード
メイン時計	30 割込み or リセット起動	SM遷移モード
メイン時計遷移	36 メイン サブクロック切換えタイミング待ち終了	メイン時計
	37 割込み or リセット起動	メインモード
PLL時計	77 割込み or リセット起動	SP遷移モード

表 6.2-1 遷移条件一覧表 (3/3)

遷移前の状態	遷移条件	遷移後の状態
PLL時計遷移M	72メイン サブクロック切換えタイミング待ち終了	PLL時計
	71 割込み or リセット起動	MP遷移モード
PLL時計遷移P	65 PLL メインクロック切換えタイミング待ち終了	PLL時計遷移M
	64 割込み or リセット起動	PLLモード
サブ時計	47 割込み or リセット起動	サブモード
メインストップ	41 割込み or リセット起動	メイン発振安定
擬似時計	62 割込み or リセット起動	MP遷移モード
擬似時計遷移	61 PLL メインクロック切換えタイミング待ち終了	擬似時計モード
	60 割込み or リセット起動	PLLモード
サブストップ	48 割込み	サブ発振安定
	79 リセット起動	メイン発振安定
サブ発振安定	45 サブクロック発振安定待ち終了	サブモード
	80 リセット起動	メイン発振安定

6.3 低消費電力モード状態遷移図

図 6.3-1～図 6.3-4に状態遷移図を示します。状態遷移図では、複雑になるのを避けるために、同時に発生した事象を段階的に遷移する図になっています。実際には、即座に状態の遷移は行われます。

低消費電力モード状態遷移図

PLLクロックモードでMCS="1", SLP=1を同時に設定した場合に、状態遷移図では、いったんPM遷移モードに遷移した後に、PM遷移スリープに遷移することになっていますが、実際には、即座にPLLクロックモードからPM遷移スリープに遷移します。また、サブスリープモードで、リセットが起動された場合に、いったんサブモードに遷移した後に、メイン発振安定期間に遷移することになっていますが、実際には、即座にサブスリープモードからメイン発振安定期間に遷移します。

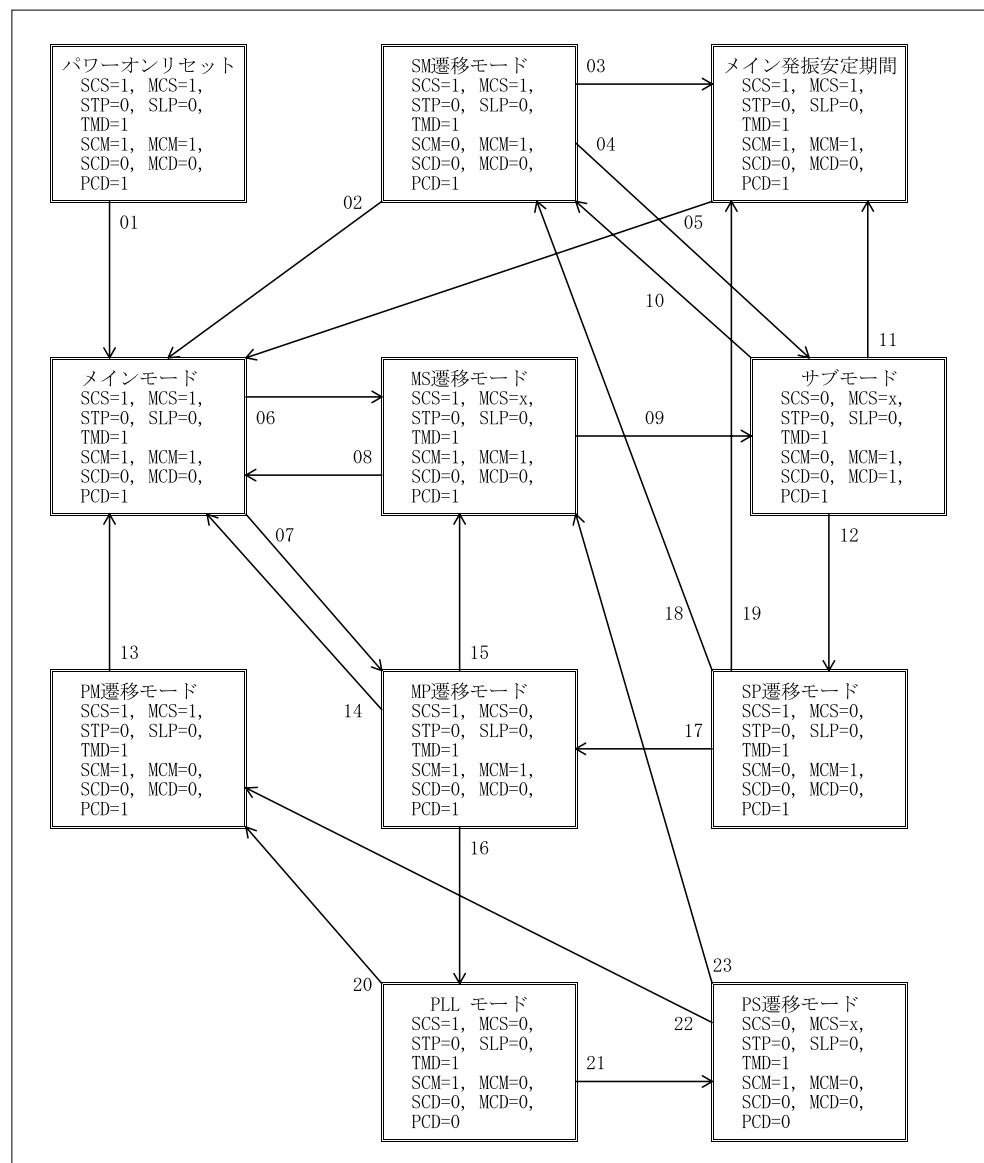


図 6.3-1 低消費電力モード状態遷移図A

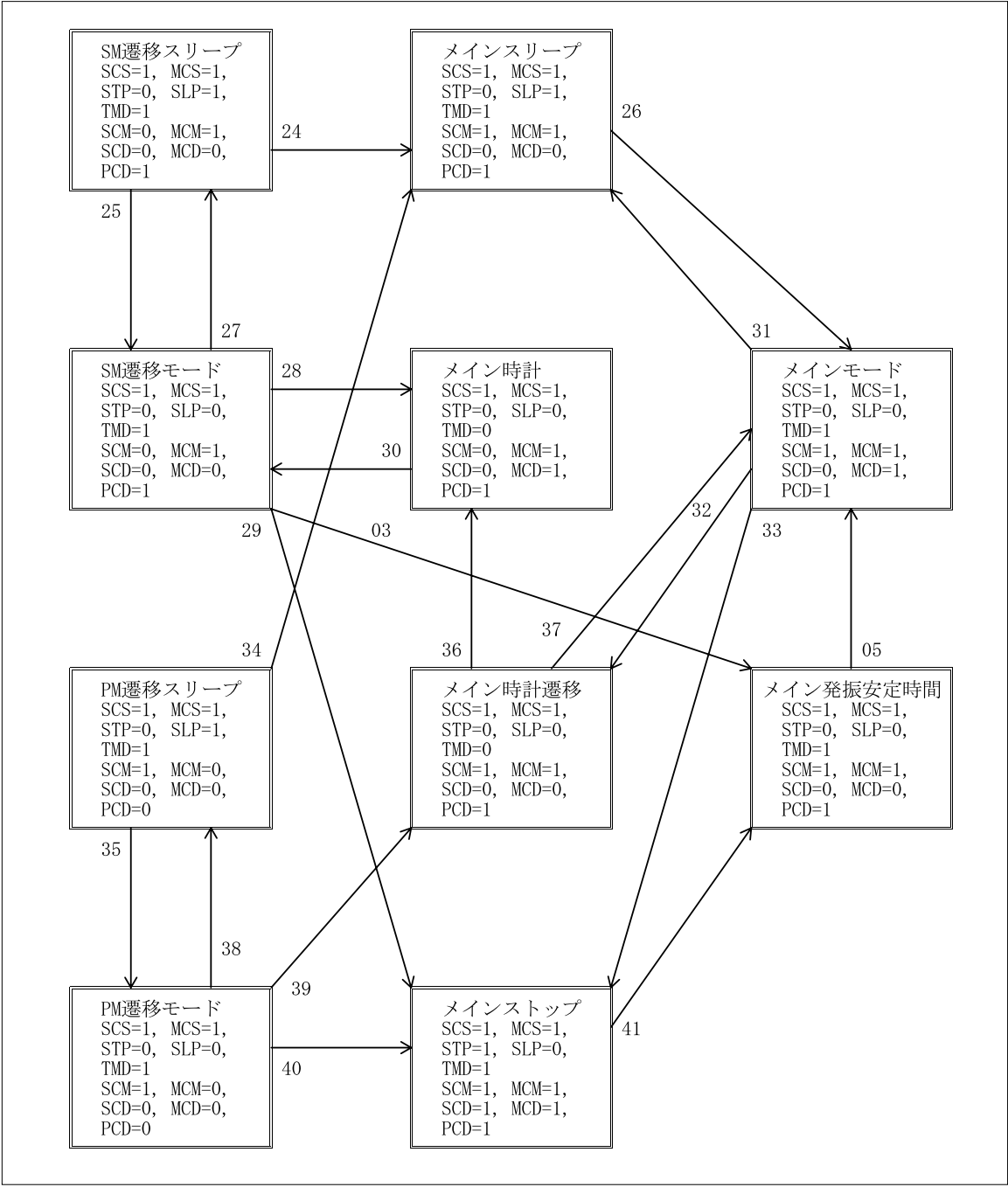


図 6.3-2 低消費電力モード状態遷移図B

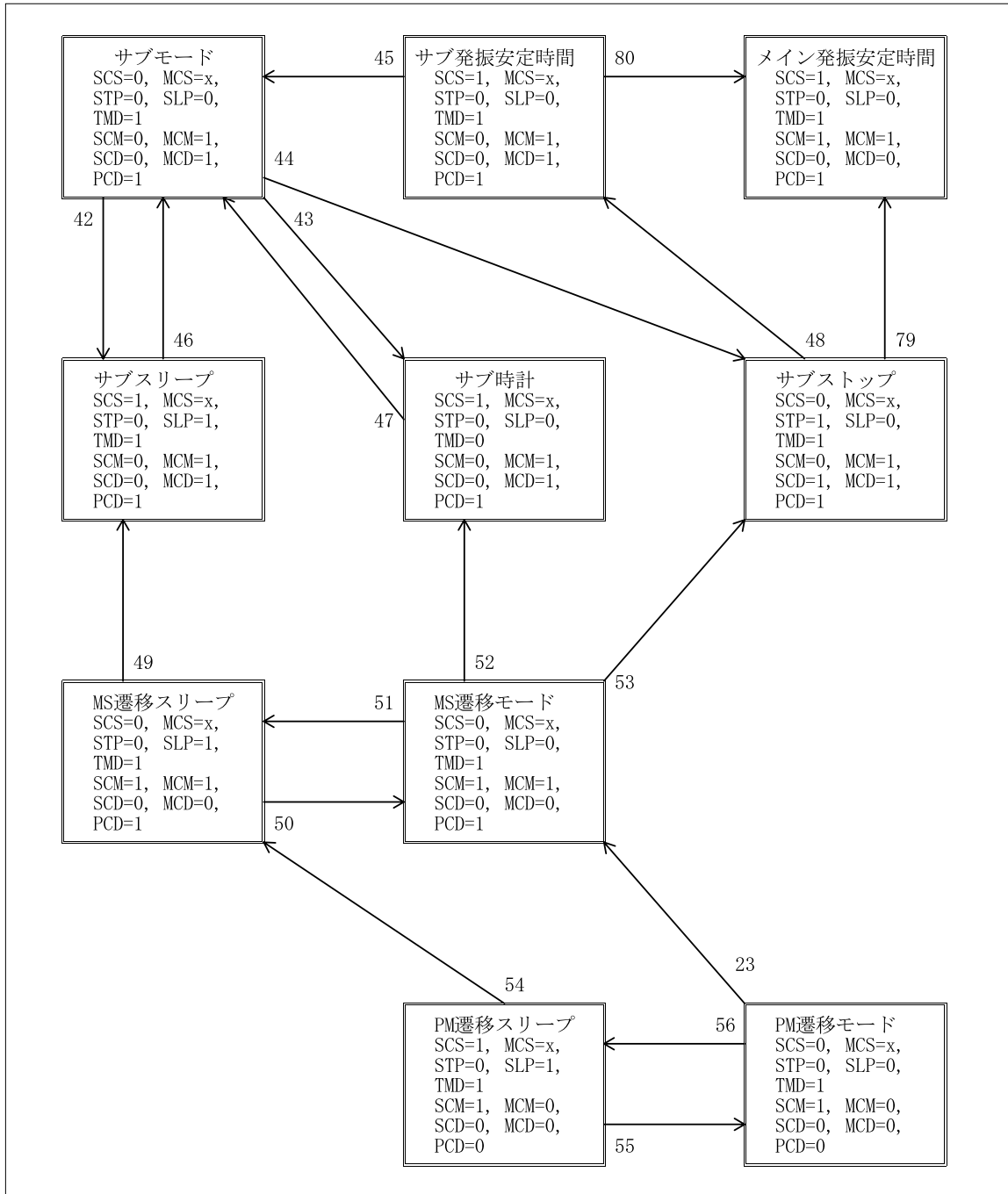


図 6.3-3 低消費電力モード状態遷移図C

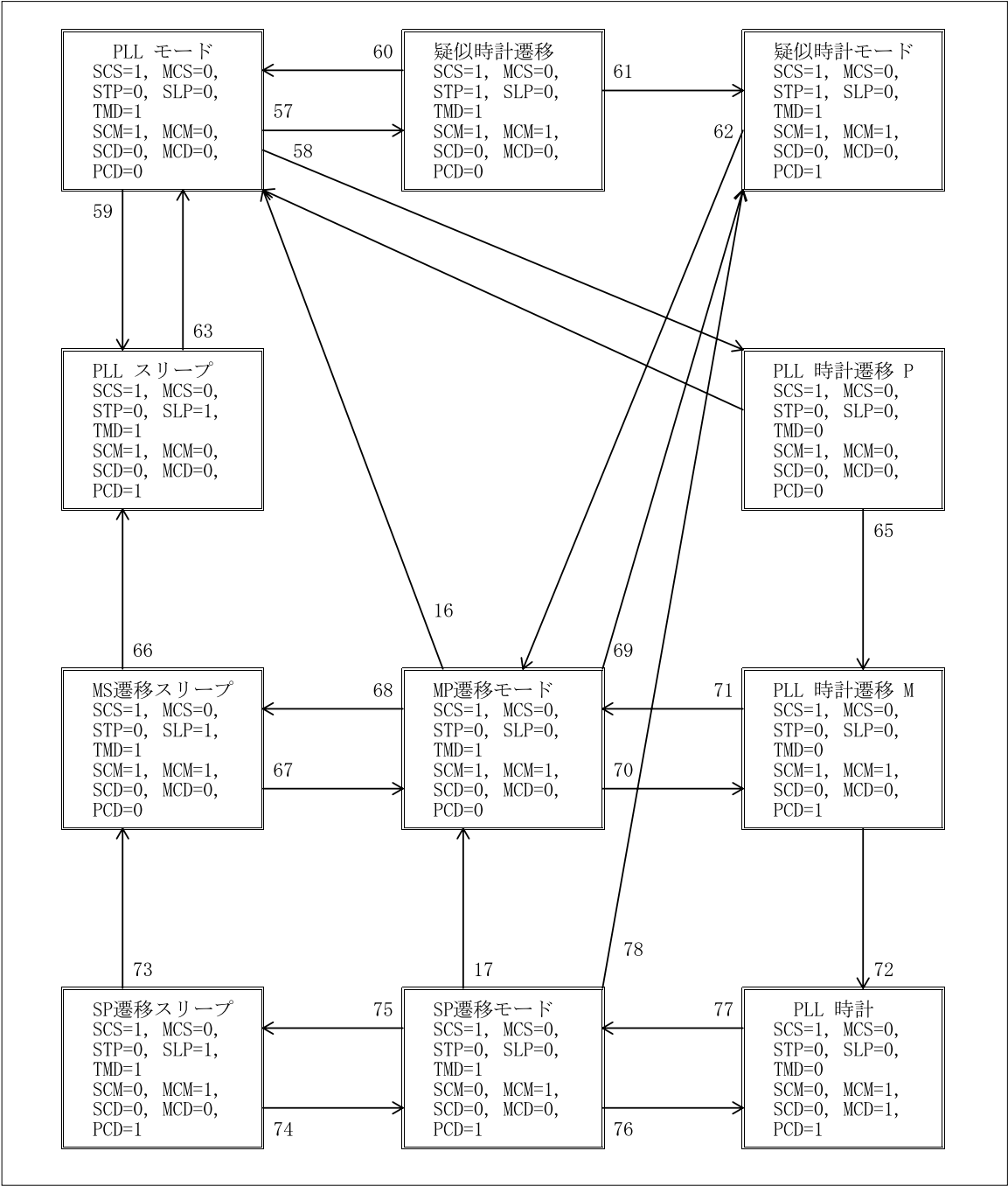


図 6.3-4 低消費電力モード状態遷移D

第7章 メモリアクセスモード

この章では，メモリアクセスモードの機能と動作について説明します。

7.1 メモリアクセスモード

7.2 外部メモリアクセス（外部バス端子制御回路）の概要

7.3 外部メモリアクセス制御信号の動作

7.1 メモリアクセスモード

F²MC-16LX では、アクセス方式、アクセス領域のおののおのについて、各種のモードがあります。

メモリアクセスモード

表 7.1-1 メモリアクセスモード

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

動作モード

動作モードとは、デバイスの動作状態を制御するモードを示すもので、モード設定用端子(MD_x)の内容で指定します。

動作モードを選択することで、通常動作の起動 / フラッシュメモリの書込みを行うことができます。

バスモード

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

モード設定用端子(MD_x) は、リセットベクタおよびモードデータを読み出すときのバスモードを指定するもので、モードデータ内のM_xビットは、通常動作時のバスモードを指定するものです。

アクセスモード

アクセスモードとは、外部データバス幅を制御するモードを示すもので、モード設定用端子 (MD_x) とモードデータ内のS0ビットで指定します。

アクセスモードを選択することで、外部データバスを8ビット長あるいは16ビット長のいずれかを指定します。

7.1.1 モード端子

MD2～0の3本の外部端子の設定の組合せでモードの指定ができます。

モード端子

表 7.1-2 モード端子と設定モードの関係

モード端子 設定 MD2 MD1 MD0	モード名	リセットベクトル アクセス領域	外部データ バス幅	備考
0 0 0	外部ベクトルモード0	外部	8ビット	-
0 0 1	外部ベクトルモード1	外部	16ビット	リセットベクトル 16ビットバス幅 アクセス
0 1 0	(指定禁止)			
0 1 1	内部ベクトルモード	内部	(モードデータ)	リセットシーケンス 以降はモードデータ で制御
1 0 0	(指定禁止)			
1 0 1				
1 1 0	フラッシュシリアル書込み*	-	-	-
1 1 1	フラッシュメモリモード	-	-	パラレルライタ使用 時のモードです。

*:フラッシュメモリのシリアル書込みは、モード端子の設定だけでは書込みができません。他の端子設定も必要です。詳細は、フラッシュシリアル書込み接続例をご参照ください。

< 注意事項 >

外部ベクトルモード"0"を選択した場合でもバス制御選択レジスタのIOBS, LMBSの初期値は"1"にセットされていますので、0000C0_H～0000FF_H, 002000_H～7FFFFFF_Hの領域では16ビット幅となります。この領域で8ビット幅としたい場合はバス制御選択レジスタのIOBS, LMBSに"0"を書き込んでください。

外部ベクトルモード"1"のときには、HMBSビットが"0"にセットされ、16ビットバス幅でアクセスが行われます。

フラッシュメモリのシリアル書込みは、モード端子の設定だけでは書込みができません。他の端子設定も必要です。詳細は、「第27章 MB90F574/Aシリアル書込み接続例」をご参照ください。

7.1.2 モードデータ

主記憶FFFFDF_H に置く，CPU の動作制御用のデータです。リセットシーケンス実行中にこのデータを取り込み，デバイス内部のモードレジスタに格納します。モードレジスタの内容を変更できるのはリセットシーケンスだけです。

本レジスタによる設定はリセットシーケンス以降に有効となります。

予約ビットは，必ず"0"を設定してください。

モードデータ

モードデータ	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス：FFFFDF _H	M1	M0	予約	予約	S0	予約	予約	予約	

図 7.1-1 モードデータ

【ビット7，6】：M1,M0 バスモード設定ビット

リセットシーケンス終了後の動作モードの指定を行うビットです。

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

【ビット3】：S0 アクセスモード設定ビット

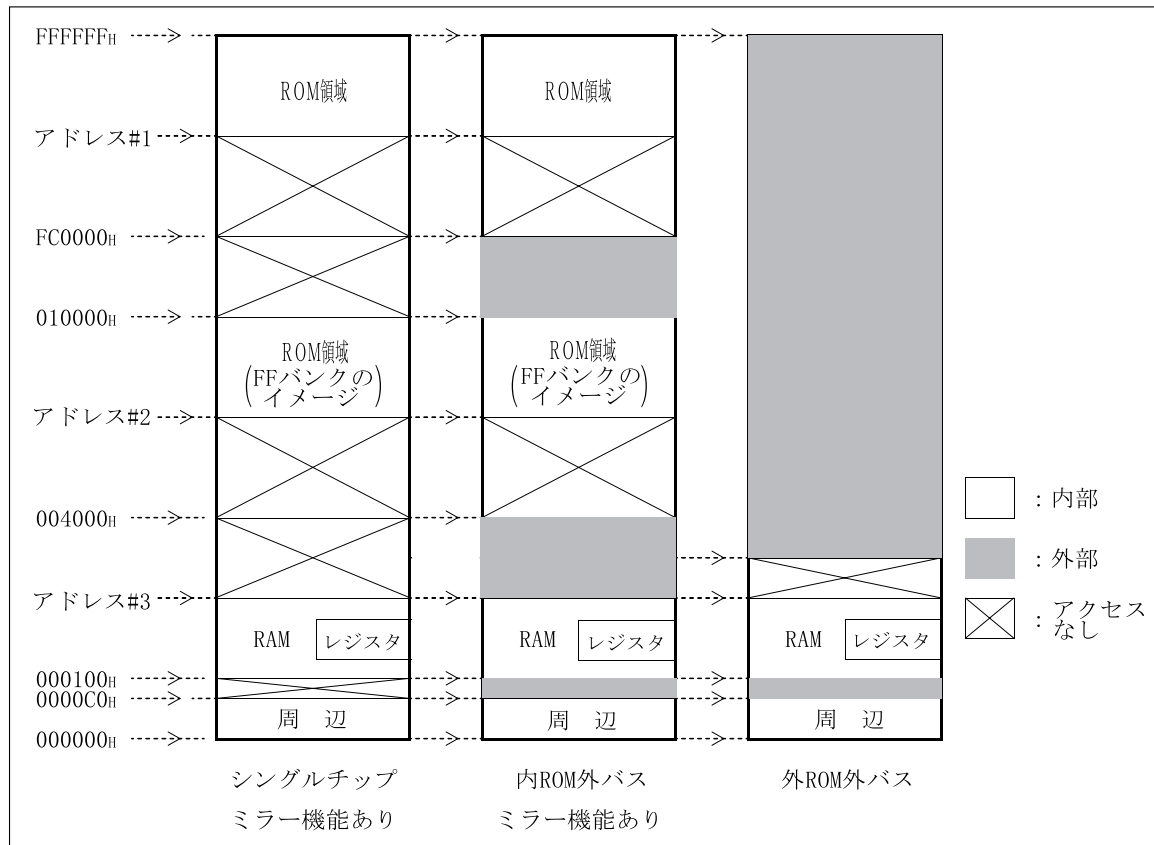
リセットシーケンス終了後のバスモード，アクセスモードの指定を行うビットです。

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

7.1.3 バスモード別メモリ空間

バスモードの指定による、アクセス領域と物理アドレスの対応を以下に示します。

バスモード別メモリ空間



品種	アドレス#1	アドレス#2	アドレス#3
MB90573	FE0000H	004000H	001900H
MB90574/C	FC0000H	004000H	002900H
MB90F574/A	FC0000H	004000H	002900H
MB90V570/A	(FC0000H)	004000H	002900H

図 7.1-2 MB90570のモード別メモリ空間

< 注意事項 >

ROMのミラー機能なしを選択した場合は、「第25章 ROMミラー機能選択モジュール」を参照してください。

00バンクの上位にFFバンクのROMがイメージで見えるようになっていますが、これはCコンパイラのスモールモデルを有効に生かすためです。下位16ビットは同じになるようにしてありますので、ポインタの宣言においてfar指定をしなくてもROM内のテーブルを参照することができます。

例えば、00C000Hをアクセスした場合に、実際には、FFC000HのROMの内容がアクセスされることになります。ここで、FFバンクのROM領域は、48Kバイトを越します。00バンクのイメージにすべての領域を見せることができません。したがって、FF4000H ~ FFFFFFFHのROMデータは004000H ~ 00FFFFHのイメージに見えますので、ROMデータテーブルはFF4000H ~ FFFFFFFHの領域に格納することを推奨いたします。

推奨設定例

表 7.1-3 モード端子とモードデータの推奨設定例

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

外部端子は、各種モードにより入出力する信号が変化します。

表 7.1-4 各種モードと関係する外部端子の動作

端子名	機能 シングルチップ	機能 外部バス拡張	
		外部バス拡張	
		8ビット	16ビット
P07-00	ポート	AD07-00	
P17-10		A15 ~ 08	
P27-20		A23 ~ 16*	AD15 ~ 08
P30		ALE	
P31		RDX	
P32		WRLX*	
P33		ポート	
P34		HRQ*	HRHX*
P35		HAKX*	
P36		RDY*	
P37		CLK*	

< 注意事項 >

アドレス上位とWRLX, WRHX, HAKX, HRQ, RDY, CLKは、機能選択によりポートとして使用することが可能です。詳細は、「7.2 外部メモリアクセス（外部バス端子制御回路）の概要」を参照ください。

7.2 外部メモリアクセス（外部バス端子制御回路）の概要

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

外部メモリアクセス（外部バス端子制御回路）

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

- CLK(P37)：マシンサイクルクロック（KBP）を出力
- RDY(P36)：外部レディ入力端子
- WRHX(P33)：データバス上位8ビットのライト信号
- WRLX(P32)：データバス下位8ビットのライト信号
- RDX (P31)：リード信号
- ALE (P30)：アドレスラッチイネーブル信号

外部メモリアクセスレジスター一覧

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

図 7.2-1 外部メモリアクセスレジスター一覧

外部メモリアクセスのブロックダイアグラム

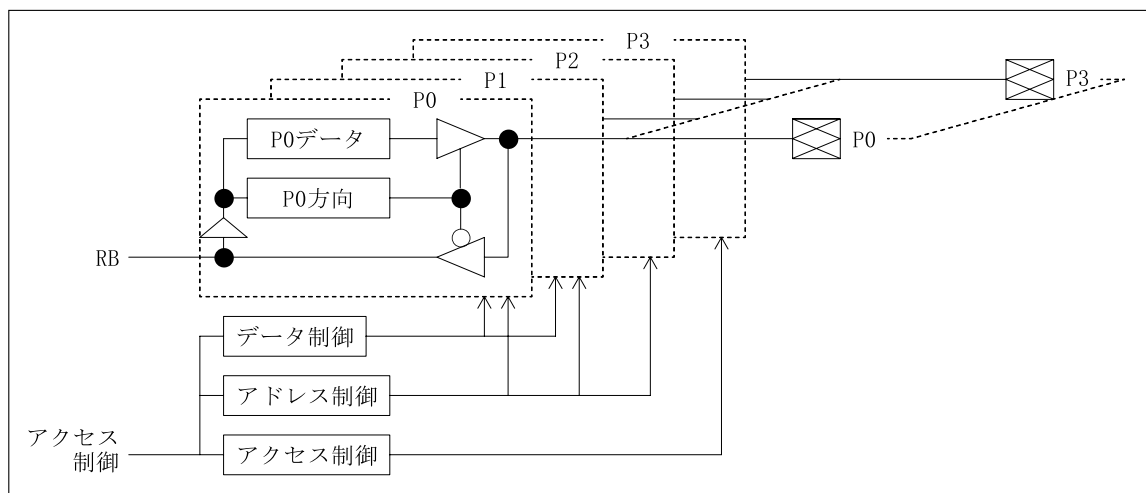


図 7.2-2 外部メモリアクセスのブロックダイアグラム

7.2.1 自動レディ機能選択レジスタ(ARSR)

自動レディ機能選択レジスタ(ARSR)は、外部アクセス時の領域ごとのメモリアクセスの自動ウェイト時間を設定します。

自動レディ機能選択レジスタ(ARSR)

自動レディ選択レジスタ	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス:0000A5 _H	IOR1	IOR0	HMR1	HMR0	—	—	LMR1	LMR0	ARSR
リード/ライト	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒
初期値	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒
	(W)	(W)	(W)	(W)	(-)	(-)	(W)	(W)	
	(0)	(0)	(1)	(1)	(-)	(-)	(0)	(0)	

図 7.2-3 自動レディ機能選択レジスタ(ARSR)

【ビット15,14】 IOR1, IOR0

IOR1, IOR0ビットでは、0000C0_H～0000FF_Hの領域に対する外部アクセスを行ったときの自動ウェイト機能を指定します。2ビットの組合せで表 7.2-1に示すような設定になります。

表 7.2-1 IOR1, IOR0ビットの設定

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

【ビット13,12】 HMR1, HMR0

HMR1, HMR0ビットでは、800000_H～FFFFFF_Hの領域に対する外部アクセスを行ったときの自動ウェイト機能を指定します。2ビットの組合せで表 7.2-2に示すような設定になります。

表 7.2-2 HMR1, HMR0ビットの設定

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

【ビット9,8】 LMR1, LMR0

LMR1, LMR0ビットでは、002000_H～7FFFFF_Hの領域に対する外部アクセスを行ったときの自動ウェイト機能を指定します。2ビットの組合せで、表 7.2-3に示すような設定になります。

表 7.2-3 LMR1,LMR0ビットの設定

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

7.2.2 外部アドレス出力制御レジスタ(HACR)

外部アドレス出力制御レジスタ(HACR)は、アドレス（A23～16）の外部への出力を制御するレジスタです。各ビットが、それぞれアドレスA23～16に対応し、各アドレス出力端子を表 7.2-4に示すように制御します。

外部アドレス出力制御レジスタ(HACR)

外部アドレス出力制御レジスタ									⇐ビットNo.
	7	6	5	4	3	2	1	0	
アドレス:0000A6H	E23	E22	E21	E20	E19	E18	E17	E16	HACR
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 7.2-4 外部アドレス出力制御レジスタ(HACR)

HACRレジスタは、デバイスがシングルチップモード時にはアクセスすることができません。その場合、本レジスタの値にかかわらず、全端子がI/Oポートとして機能します。

本レジスタの全ビットはすべて書込み専用で、読出しでは"1"となります。

表 7.2-4 外部アドレス出力制御レジスタ(A23～16ビット)の制御

A23～16	制御
0	対応する端子はアドレス出力（A××）になります。[初期値]
1	対応する端子はI/Oポート（P××）になります。

7.2.3 バス制御信号選択レジスタ(ECSR)

バス制御信号選択レジスタ(ECSR)は、外部バスモード時のバス動作の制御機能を設定するレジスタです。本レジスタは、デバイスがシングルチップモード時にはアクセスすることができません。その場合、本レジスタの値にかかわらず、全端子がI/Oポートとして機能します。

本レジスタの全ビットはすべて書込み専用で、読出しでは"1"となります。

バス制御信号選択レジスタ(ECSR)

制御信号選択レジスタ	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス:0000A7H	CKE	RYE	HDE	IOBS	HMBS	WRE	LMBS	—	ECSR
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(—)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(—)	

図 7.2-5 バス制御信号選択レジスタ

【ビット15】CKE

CKEビットでは、外部クロック（CLK）の出力を表 7.2-5に示すように制御します。

表 7.2-5 CKEビットの制御

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

【ビット14】RYE

RYEビットでは、外部レディ（RDY）の入力を表 7.2-6に示すように制御します。

表 7.2-6 RYEビットの制御

RYE	制御
0	I/Oポート（P36）動作（外部RDY 入力禁止）[初期値]
1	外部レディ（RDY）入力許可

【ビット13】HDE

HDEビットでは、ホールド関係の端子の入出力許可を指定します。本ビットの設定によりホールド要求入力（HRQ）とホールドアクノリッジ出力（HAKX）の2本を表 7.2-7に示すように制御します。

表 7.2-7 HDEビットの制御

HDE	制御
0	I/Oポート（P35,P34）動作（ホールド機能入出力禁止）[初期値]
1	ホールド要求（HRQ）入力 / ホールドアクノリッジ（HAKX）出力許可

【ビット12】IOBS

IOBSビットでは，外部データバス16ビットモード時に，0000C0_H～0000FF_Hの領域に対する外部アクセスを行ったときのバスサイズを指定します。本ビットの設定により，表 7.2-8に示すように制御します。

表 7.2-8 IOBSビットの設定

IOBS	制御
0	16ビットバスサイズアクセス [初期値]
1	8ビットバスサイズアクセス

【ビット11】HMBS

HMBSビットでは，外部データバス16ビットモード時に，800000_H～FFFFFF_Hの領域に対する外部アクセスを行ったときのバスサイズを指定します。本ビットの設定により，表 7.2-9に示すように制御します。

表 7.2-9 HMBSビットの制御

HMBS	制御
0	16ビットバスサイズアクセス [外部ベクトルモード1のときの初期値]
1	8ビットバスサイズアクセス [外部ベクトルモード0のときの初期値]

【ビット10】WRE

WREビットでは，外部ライト信号（16ビットバスモードの場合はWRHX / WRLX両端子，8ビットバスモードの場合はWRLX端子）の出力を，表 7.2-10に示すように制御します。

外部データバス8ビットモードの場合は，本ビットの設定値の如何によらずP33はI/Oポートとして機能します。

表 7.2-10 WREビットの制御

WRE	制御
0	I/Oポート（P33,P32）動作（ライト信号出力禁止） [初期値]
1	ライトストロープ信号（WRHX / WRLXまたはWRLXのみ）の出力許可

【ビット9】LMBS

LMBSビットでは，外部データバス16ビットモード時に，002000_H～7FFFF_Hの領域に対する外部アクセスを行ったときのバスサイズを指定します。本ビットの設定により，表 7.2-11に示すように制御します。

表 7.2-11 LMBSビットの制御

LMBS	制御
0	16ビットバスサイズアクセス [初期値]
1	8ビットバスサイズアクセス

< 注意事項 >

-
- ・ 16ビットバスモード時，WREビットにてWRHX,WRLX機能を許可する場合は，P33,P32を入力モードにしてください（DDR3のビット3，ビット2は"0"にしてください）。
 - ・ 8ビットバスモード時，WREビットにてWRX機能を許可する場合は，P32を入力モードにしてください（DDR3のビット2は"0"にしてください）。
- また，RYE,HDEビットにてRDY,HRQ入力を許可した場合でも，そのポートのI/Oポート機能は有効となります。そのため，DDR3中のそのポートに対応したビットは，必ず"0"(入力モード)にしてください。
-

7.3 外部メモリアクセス制御信号の動作

外部メモリへのアクセスは、レディ機能を使用しない場合に3サイクルで行います。外部16ビットバスモードでの8ビットバス幅アクセスは、8ビット幅の周辺チップと16ビット幅の周辺チップを混在して外部バスに接続した場合に、8ビット幅の周辺チップをリード/ライトするための機能です。

MB90570シリーズでは、アクセス方式、アクセス領域について各種のモードがあります。「7.1 メモリアクセスモード」を参照してください。

外部メモリアクセス制御信号

8ビットバス幅アクセスは、データバス下位8ビットを使用して実行されるので、8ビット幅の周辺チップは、データバスの下位8ビットに接続してください。

外部16ビットバスモードで16ビットバス幅アクセスを行うか8ビットバス幅アクセスを行うかは、EPCRのHMBS/LMBS/IOBSビットで指定することになります。

RDX/WRLX/WRHXをアサートしないことにより実際にはバス動作を行わない場合があります。ALE信号のみで、周辺チップへのアクセスを実行することがないようにしてください。

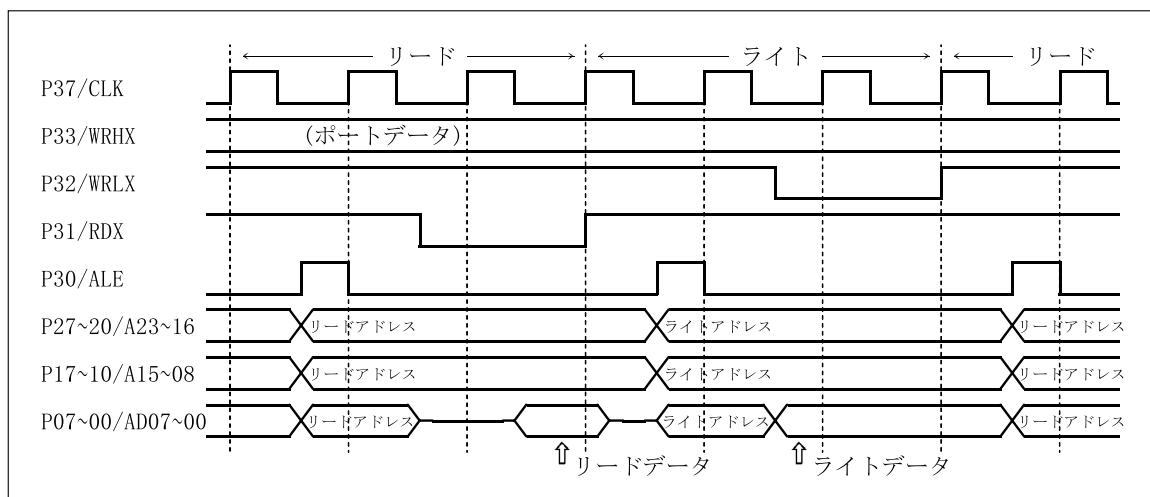
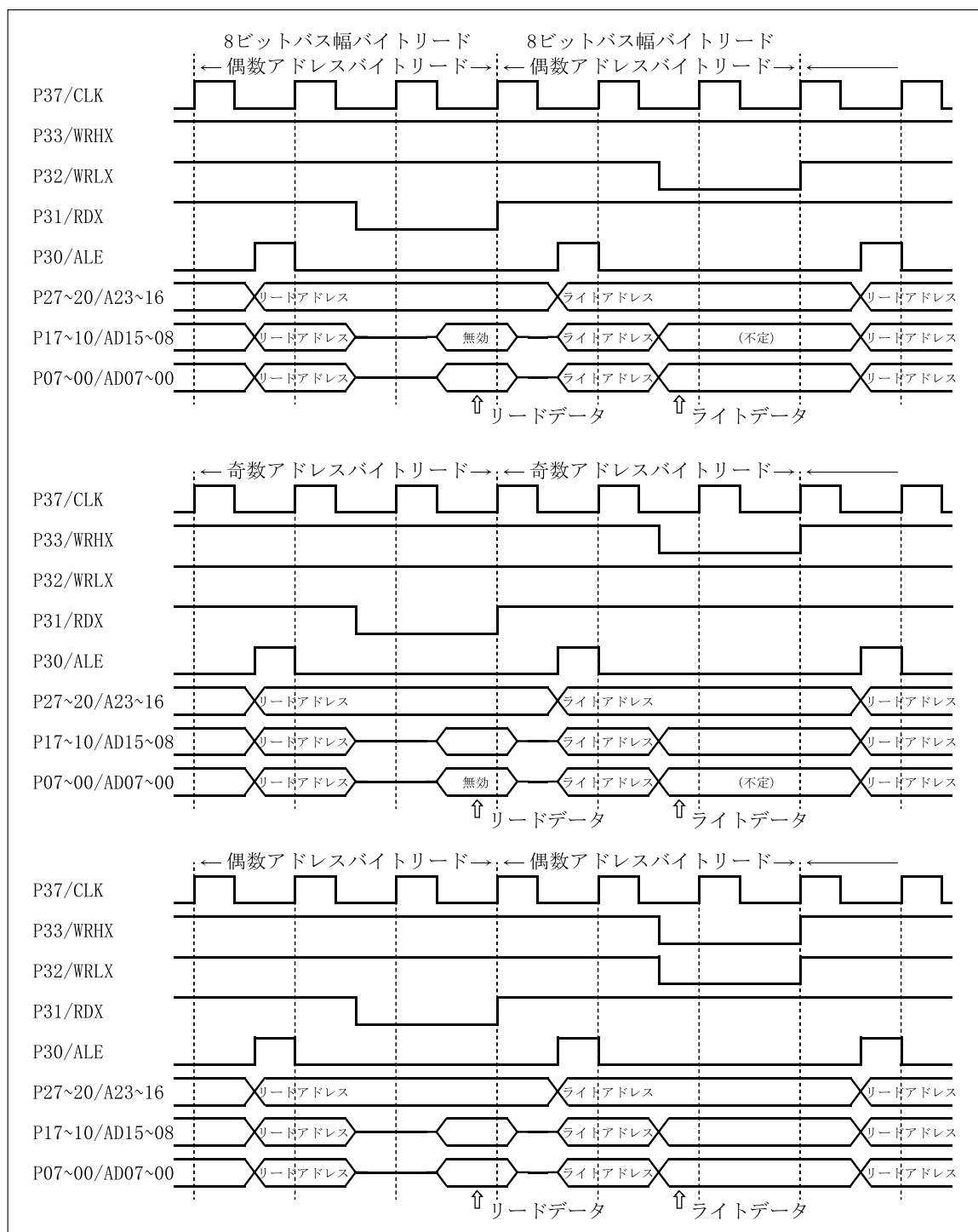


図 7.3-1 外部メモリアクセスタイミングチャート（外部8ビットバスモード）



(注意) 常にワードで読みだすように外部回路を設計してください。
P36/RXD端子, または, 自動レディ機能選択レジスタ(ARSR)の設定により, 低速のメモリや周辺回路へのアクセスが可能となります。

図 7.3-2 外部メモリアクセスタイミングチャート (外部16ビットバスモード)

7.3.1 レディ機能

P36/RDY 端子，または自動レディ機能選択レジスタ(ARSR)の設定により，低速のメモリや周辺回路へのアクセスが可能となります。

バス制御信号選択レジスタ(EPCR)中のRYE ビットが"1"に設定されている場合，外部領域へのアクセス時にP36/RDY 端子に L レベルが入力されている期間ウェイトサイクルとなり，アクセスサイクルを延ばすことが可能です。

レディ機能

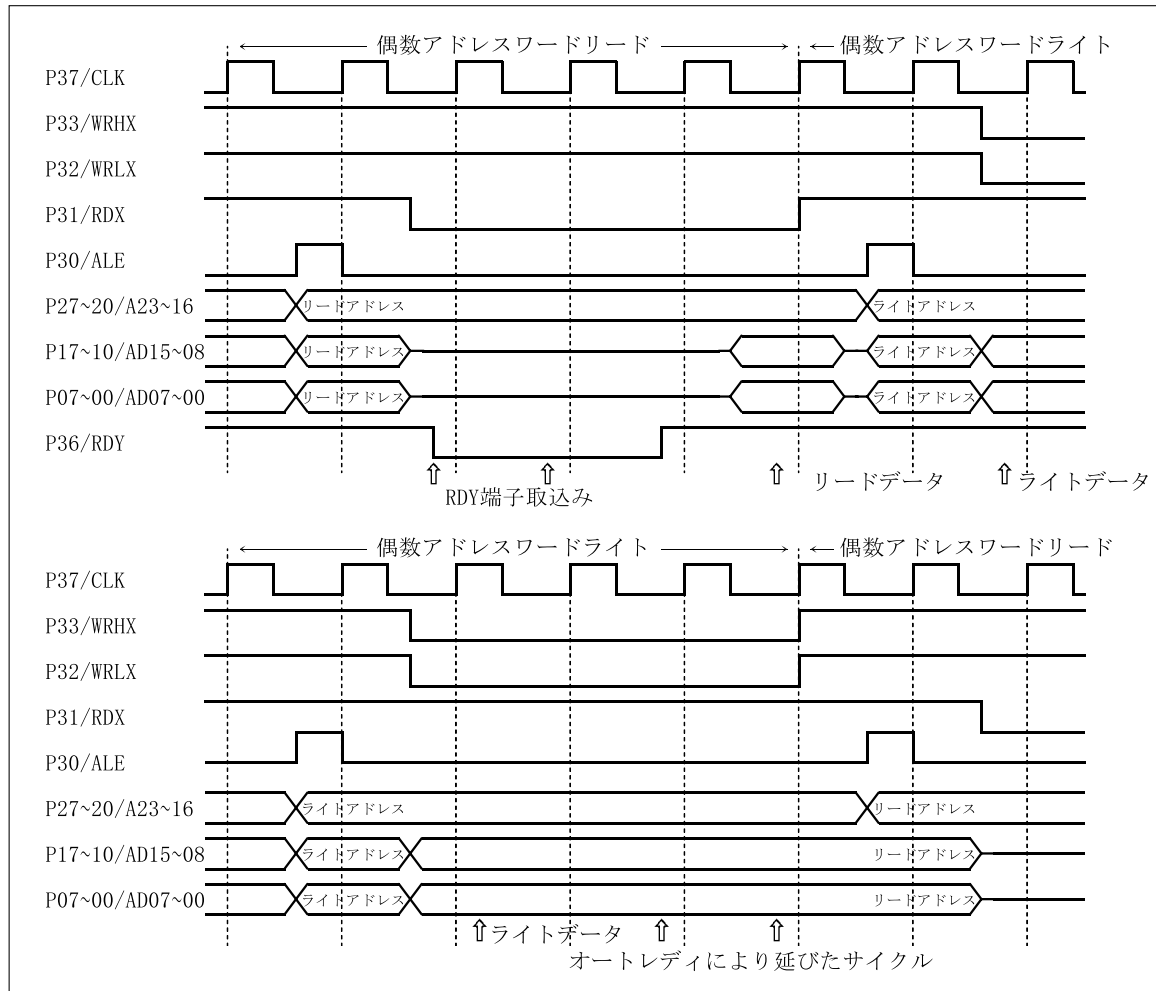


図 7.3-3 レディタイミングチャート

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

さらに、F²MC-16LXは外部I/O用のオートレディ機能をメモリ用とは独立に内蔵しています。アドレス0000C0_H～0000FF_H間の外部領域へのアクセス時に、外部回路なしで自動的に1～3サイクルのウェイトサイクルを挿入し、アクセスサイクルを延ばすことができる機能で、ARSR中のIOR1/IOR0ビットの設定により起動します。外部メモリ用、外部I/O用のオートレディどちらでもEPCR中のRYEビットが"1"に設定されている場合、上記オートレディ機能によるウェイトサイクル終了後にP36/RDY端子にLレベルが入力されているとそのままウェイトサイクルを継続します。

7.3.2 ホールド機能

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

ホールド機能

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

- アドレス出力：P23/A19～P20/A16
- アドレス／データ入出力：P17/D15～P00/D00
- バス制御信号：P30/ALE,P31/RDX,P32/WRLX,P33/WRHX

これにより，デバイス外部回路により外部バスを使用することが可能となります。P34/HRQ端子にLレベルを入力すると，P35/HAKX端子はHレベル出力となり，外部端子状態が復活してCPUは動作を再開します。

STOP状態においては，ホールド要求入力は受け付けません。

図 7.3-4に，外部バス16ビットモード時のホールドタイミングを示します。

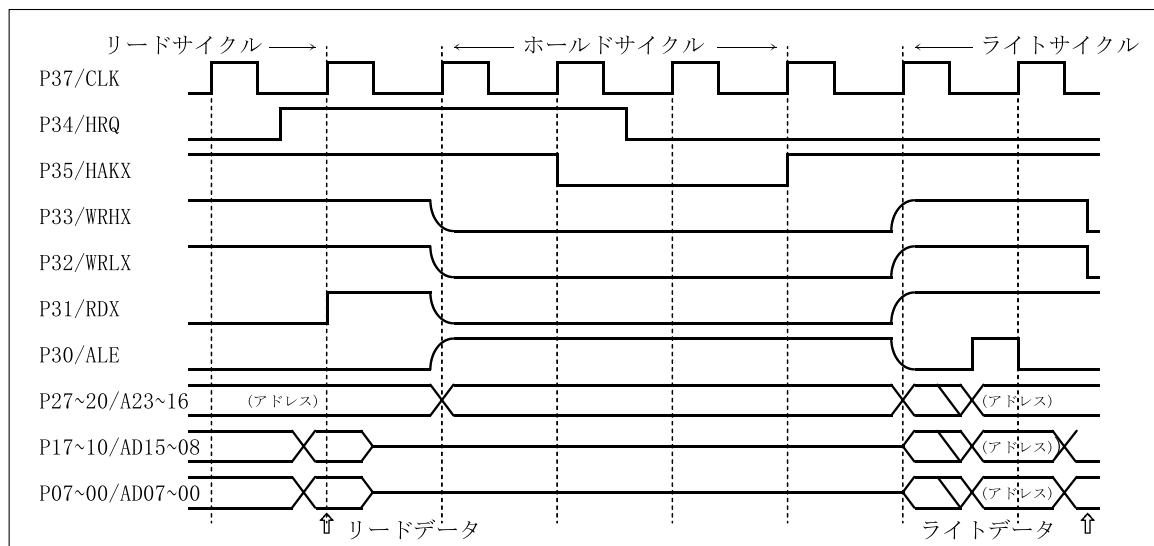


図 7.3-4 ホールドタイミング（外部バス16ビットモード時）

第8章 I/Oポート

この章では，I/Oポートの機能と動作について説明します。

8.1 I/Oポートの概要

8.2 I/Oポートのレジスタ一覧

8.1 I/Oポートの概要

それぞれのポートの各端子は，対応する周辺が端子を用いない設定となっているとき，1本ごとにポート方向レジスタ(DDR)の設定により入力，出力の指定を行うことができます。

I/Oポートの概要

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

制御出力として使用している場合にPDRレジスタを読み出した場合は，DDRレジスタの値に関係なく，制御出力として出力しているものが，読み出されます。

入力設定から出力設定に変更する場合，あらかじめPDRレジスタに出力データを設定するときに，リードモディファイライト系の命令（ビットセットなどの命令）を使用しますと，リードしてくるデータは端子よりの入力データであり，PDRレジスタのラッチの値ではありませんので，注意が必要です。

図 8.1-1に，入出力ポートのブロックダイヤグラムを示します。

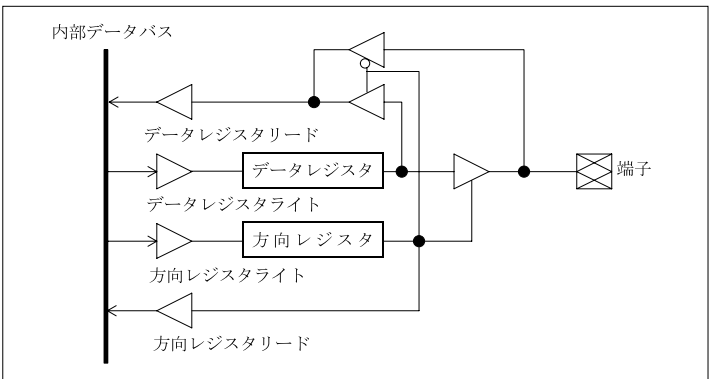


図 8.1-1 I/Oポートのブロックダイヤグラム

DDRレジスタにてポート出力に設定した場合のリソース兼用ポートの動作

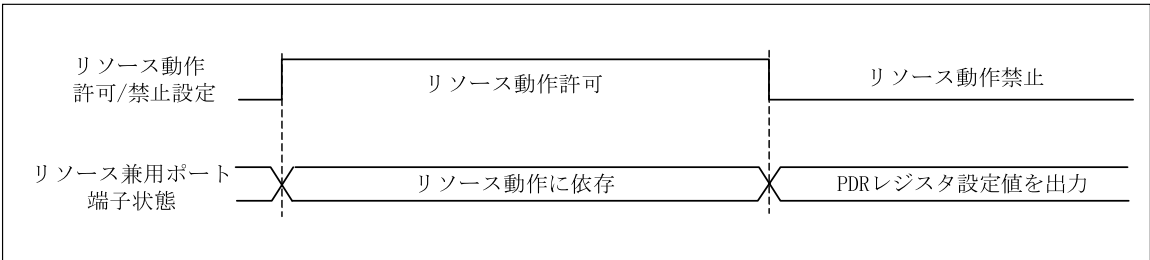


図 8.1-2 リソース動作許可 / 禁止時におけるリソース兼用ポートの端子状態

8.2 I/Oポートのレジスタ一覧

I/Oポートの各レジスタのビット構成を、図 8.2-1に示します。

I/Oポートのレジスタ一覧

ビットNo. ⇨	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス：000000H	P07	P06	P05	P04	P03	P02	P01	P00	ポート0データレジスタ (PDR0)
アドレス：000001H	P17	P16	P15	P14	P13	P12	P11	P10	ポート1データレジスタ (PDR1)
アドレス：000002H	P27	P26	P25	P24	P23	P22	P21	P20	ポート2データレジスタ (PDR2)
アドレス：000003H	P37	P36	P35	P34	P33	P32	P31	P30	ポート3データレジスタ (PDR3)
アドレス：000004H	P47	P46	P45	P44	P43	P42	P41	P40	ポート4データレジスタ (PDR4)
アドレス：000005H	P57	P56	P55	P54	P53	P52	P51	P50	ポート5データレジスタ (PDR5)
アドレス：000006H	P67	P66	P65	P64	P63	P62	P61	P60	ポート6データレジスタ (PDR6)
アドレス：000007H	—	—	—	P74	P73	P72	P71	P70	ポート7データレジスタ (PDR7)
アドレス：000008H	P87	P86	P85	P84	P83	P82	P81	P80	ポート8データレジスタ (PDR8)
アドレス：000009H	P97	P96	P95	P94	P93	P92	P91	P90	ポート9データレジスタ (PDR9)
アドレス：00000AH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	ポートAデータレジスタ (PDRA)
アドレス：00000BH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	ポートBデータレジスタ (PDRTB)
アドレス：00000CH	—	—	—	—	PC3	PC2	PC1	PC0	ポートCデータレジスタ (PDRTC)

ビットNo. ⇨	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス：000010H	D07	D06	D05	D04	D03	D02	D01	D00	ポート0方向レジスタ (DDR0)
アドレス：000011H	D17	D16	D15	D14	D13	D12	D11	D10	ポート1方向レジスタ (DDR1)
アドレス：000012H	D27	D26	D25	D24	D23	D22	D21	D20	ポート2方向レジスタ (DDR2)
アドレス：000013H	D37	D36	D35	D34	D33	D32	D31	D30	ポート3方向レジスタ (DDR3)
アドレス：000014H	D47	D46	D45	D44	D43	D42	D41	D40	ポート4方向レジスタ (DDR4)
アドレス：000015H	D57	D56	D55	D54	D53	D52	D51	D50	ポート5方向レジスタ (DDR5)
アドレス：000016H	D67	D66	D65	D64	D63	D62	D61	D60	ポート6方向レジスタ (DDR6)
アドレス：000017H	—	—	—	D74	D73	D72	D71	D70	ポート7方向レジスタ (DDR7)
アドレス：000018H	D87	D86	D85	D84	D83	D82	D81	D80	ポート8方向レジスタ (DDR8)
アドレス：000019H	D97	D96	D95	D94	D93	D92	D91	D90	ポート9方向レジスタ (DDR9)
アドレス：00001AH	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	ポートA方向レジスタ (DDRA)
アドレス：00001BH	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	ポートB方向レジスタ (DDRB)
アドレス：00001CH	—	—	—	—	DC3	DC2	DC1	DC0	ポートC方向レジスタ (DDRC)

ビットNo. ⇨	15	14	13	12	11	10	9	8	
アドレス：00001DH	OD47	OD46	OD45	OD44	OD43	OD42	OD41	OD40	ポート4端子レジスタ (ODR4)

ビットNo. ⇨	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス：00008CH	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	ポート0抵抗レジスタ (RDR0)
アドレス：00008DH	RD17	RD16	RD15	RD14	RD13	RD12	RD11	RD10	ポート1抵抗レジスタ (RDR1)
アドレス：00008EH	RD67	RD66	RD65	RD64	RD63	RD62	RD61	RD60	ポート6抵抗レジスタ (RDR6)

ビットNo. ⇨	15	14	13	12	11	10	9	8	
アドレス：00001EH	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	ポート8アナログ入力許可レジスタ (ADER)

図 8.2-1 I/Oポートのレジスタ一覧

8.2.1 ポートデータレジスタ(PDR)

ポートデータレジスタ(PDR)のビット構成の詳細を、図 8.2-2に示します。

ポートデータレジスタ(PDR)								初期値	アクセス
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDR0 アドレス：000000H	P07	P06	P05	P04	P03	P02	P01	P00	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDR1 アドレス：000001H	P17	P16	P15	P14	P13	P12	P11	P10	不定 R/W *
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDR2 アドレス：000002H	P27	P26	P25	P24	P23	P22	P21	P20	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDR3 アドレス：000003H	P37	P36	P35	P34	P33	P32	P31	P30	不定 R/W *
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDR4 アドレス：000004H	P47	P46	P45	P44	P43	P42	P41	P40	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDR5 アドレス：000005H	P57	P56	P55	P54	P53	P52	P51	P50	不定 R/W *
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDR6 アドレス：000006H	P67	P66	P65	P64	P63	P62	P61	P60	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDR7 アドレス：000007H	—	—	—	P74	P73	P72	P71	P70	不定 R/W *
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDR8 アドレス：000008H	P87	P86	P85	P84	P83	P82	P81	P80	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDR9 アドレス：000009H	P97	P96	P95	P94	P93	P92	P91	P90	不定 R/W *
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDRA アドレス：00000AH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	不定 R/W *
ビットNo. ⇄	15	14	13	12	11	10	9	8	
PDRB アドレス：00000BH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
ビットNo. ⇄	7	6	5	4	3	2	1	0	
PDRC アドレス：00000CH	—	—	—	—	PC3	PC2	PC1	PC0	
*：入出力ポートのR/Wは、メモリへのR/Wとは多少動作が異なり、次の通りになりますので注意してください。									
○入力モード									
リード時：対応する端子のレベルが読み出されます。									
ライト時：出力用のラッチに書き込まれます。									
○出力モード									
リード時：データレジスタラッチの値が読み出されます。									
ライト時：対応する端子に出力されます。									

図 8.2-2 ポートデータレジスタ(PDR)

< 注意事項 >

- ・ポート入力モード時に、ポートデータレジスタ(PDR)へRMW系の命令を実行するとリード時に端子レベルが読み込まれます。このため、ビット操作した以外の、同系列の入力ポートとして使用しているビット値が変化してしまう場合がありますので注意が必要です。
- ・リソース動作時に、リソースと兼用しているポートのデータレジスタ(PDR)へRMW系の命令を実行すると、リソースとして動作している端子については、READ時に端子レベルが読み込まれます。このため、ビット操作した以外の、同系列のリソースとして使用しているビット値が変化してしまう場合がありますので注意が必要です。

表 8.2-1 PDRレジスタへRMW系の命令を実行した時のREAD対象

	リソース動作許可時	リソース動作禁止時
ポート入力設定時 DDR=00h	端子レベル	端子レベル
ポート出力設定時 DDR=FFh	端子レベル	PDRレジスタ値

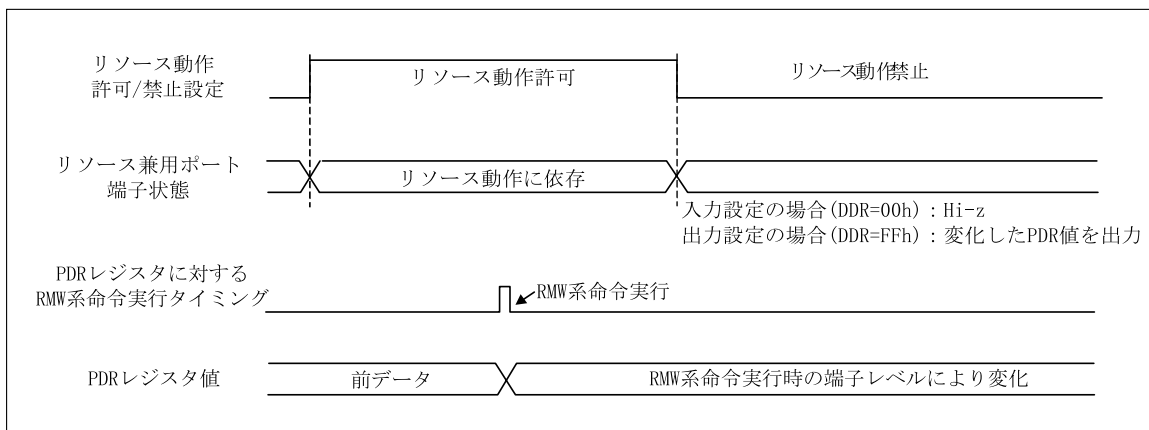


図 8.2-3 RMW命令実行時のリソース兼用ポートの端子状態

RMW系命令につきましては、「付録 B.8 F²MC-16LX命令一覧表」を参照してください。

8.2.2 ポート方向レジスタ(DDR)

ポート方向レジスタ(DDR)のビット構成を図 8.2-4に示します。

ポート方向レジスタ(DDR)

ビットNo. ⇨	7	6	5	4	3	2	1	0	初期値	アクセス
DDR0 アドレス：000010H	D07	D06	D05	D04	D03	D02	D01	D00	00000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDR1 アドレス：000011H	D17	D16	D15	D14	D13	D12	D11	D10	00000000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDR2 アドレス：000012H	D27	D26	D25	D24	D23	D22	D21	D20	00000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDR3 アドレス：000013H	D37	D36	D35	D34	D33	D32	D31	D30	00000000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDR4 アドレス：000014H	D47	D46	D45	D44	D43	D42	D41	D40	00000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDR5 アドレス：000015H	D57	D56	D55	D54	D53	D52	D51	D50	00000000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDR6 アドレス：000016H	D67	D66	D65	D64	D63	D62	D61	D60	00000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDR7 アドレス：000017H	—	—	—	D74	D73	D72	D71	D70	----000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDR8 アドレス：000018H	D87	D86	D85	D84	D83	D82	D81	D80	00000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDR9 アドレス：000019H	D97	D96	D95	D94	D93	D92	D91	D90	00000000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDRA アドレス：00001AH	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	--000000B	R/W
ビットNo. ⇨	15	14	13	12	11	10	9	8		
DDRB アドレス：00001BH	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	00000000B	R/W
ビットNo. ⇨	7	6	5	4	3	2	1	0		
DDRC アドレス：00001CH	—	—	—	—	DC3	DC2	DC1	DC0	----0000B	R/W

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

0：入力モード
1：出力モード
リセットで“0”になります。

図 8.2-4 ポート方向レジスタ(DDR)

8.2.3 出力端子レジスタ(ODR)

出力端子レジスタ(ODR) のビット構成を図 8.2-5に、ブロックダイアグラムを図 8.2-6に示します。

出力端子レジスタ(ODR)

ポート4端子レジスタ (ODR4)									
ビットNo. ⇨	7	6	5	4	3	2	1	0	初期値
アドレス : 00001DH	0D47	0D46	0D45	0D44	0D43	0D42	0D41	0D40	00000000B

図 8.2-5 出力端子レジスタ(ODR) のビット構成

出力端子レジスタ(ODR) のブロックダイアグラム

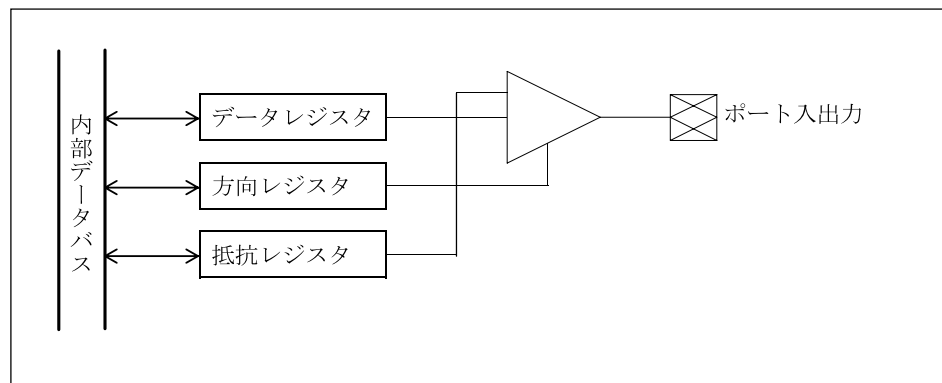


図 8.2-6 出力端子レジスタ(ODR)のブロックダイアグラム

出力端子レジスタ(ODR)に関する注意

出力端子レジスタ(ODR : R/W可能)は、出力モード時のオープンドレイン制御を行います。

- 0 : 出力モード時、標準出力ポートになります。
- 1 : 出力モード時、オープンドレイン出力ポートになります。
- 入力モード時は意味を持ちません。(出力Hi-z)
- 入出力モードは方向レジスタ(DDR) で決まります。
- ハードウェアスタンバイ時とストップ時 (SPL=1)のときはプルアップ抵抗なしになります。(ハイインピーダンス)
- バスで使用する場合、本機能は禁止です。本レジスタは書き込まないようにしてください。

8.2.4 入力抵抗レジスタ(RDR)

入力抵抗レジスタ(RDR)のビット構成を図 8.2-7に、ブロックダイアグラムを図 8.2-8に示します。

入力抵抗レジスタ(RDR)

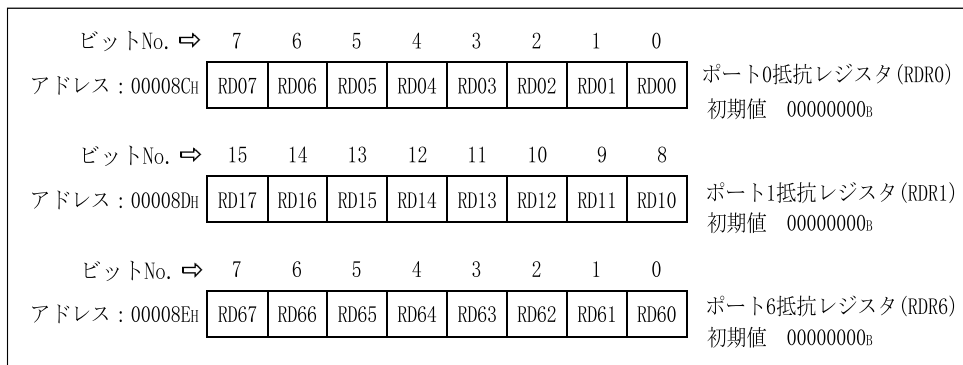


図 8.2-7 入力抵抗レジスタ(RDR)のビット構成

入力抵抗レジスタ(RDR)のブロックダイアグラム

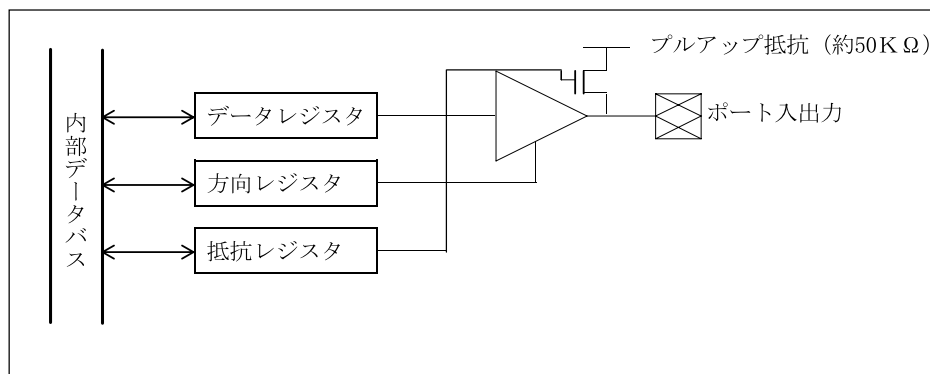


図 8.2-8 入力抵抗レジスタ(RDR)のブロックダイアグラム

入力抵抗レジスタ(PDR)に関連する注意事項

入力抵抗レジスタ(PDR : R/W可能)は、入力モード時のプルアップ抵抗制御を行います。

- 0 : 入力モード時プルアップ抵抗なし。
- 1 : 入力モード時プルアップ抵抗あり。
- 出力モード時は意味を持ちません (プルアップ抵抗なし)。
- 入出力モードはポート方向レジスタ(DDR)で決まります。
- ハードウェアスタンバイ時とストップモード時 (LPMCR : SPL=1) の場合はプルアップ抵抗なしになります (ハイインピーダンス)。
- 外バスで使用する時は本機能は禁止です。本レジスタは書き込まないようにしてください。

8.2.5 アナログ入力許可レジスタ(ADER)

アナログ入力許可レジスタ(ADER)のビット構成を、図 8.2-9に示します。

アナログ入力許可レジスタ(ADER)

ビットNo. ⇨	15	14	13	12	11	10	9	8	初期値
アドレス : 00001EH	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	11111111B
リード/ライト ⇨	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

図 8.2-9 アナログ入力許可レジスタ(ADER) のビット構成

アナログ入力許可レジスタ(ADER)は、ポート8の各端子を次のように制御します。

- 0 : ポート入力モード
- 1 : アナログ入力モード

リセットで"1"になります。

第9章 タイムベースタイマ

この章では、タイムベースタイマの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制御レジスタ (TBTC)
- 9.3 タイムベースタイマの動作

9.1 タイムベースタイマの概要

タイムベースタイマは、18ビットのタイマの他にインターバルな割込みを制御する回路から構成されています。なお、タイムベースタイマは、クロック選択レジスタ(CKSCR)中のMCSビットおよびSCSビットに関係なく、発振クロックを使用します。

タイムベースタイマのレジスター一覧

タイムベースタイマ制御レジスタ									← ビットNo.
	15	14	13	12	11	10	9	8	
アドレス : 0000A9h	予約	—	—	TBIE	TBOF	TBR	TBC1	TBC0	...
リード/ライト初期値	(-) (1)	(-) (-)	(-) (-)	(R/W) (0)	(R/W) (0)	(W) (1)	(R/W) (0)	(R/W) (0)	TBTC

図 9.1-1 タイムベースタイマのレジスタ一覧

タイムベースタイマのブロックダイヤグラム

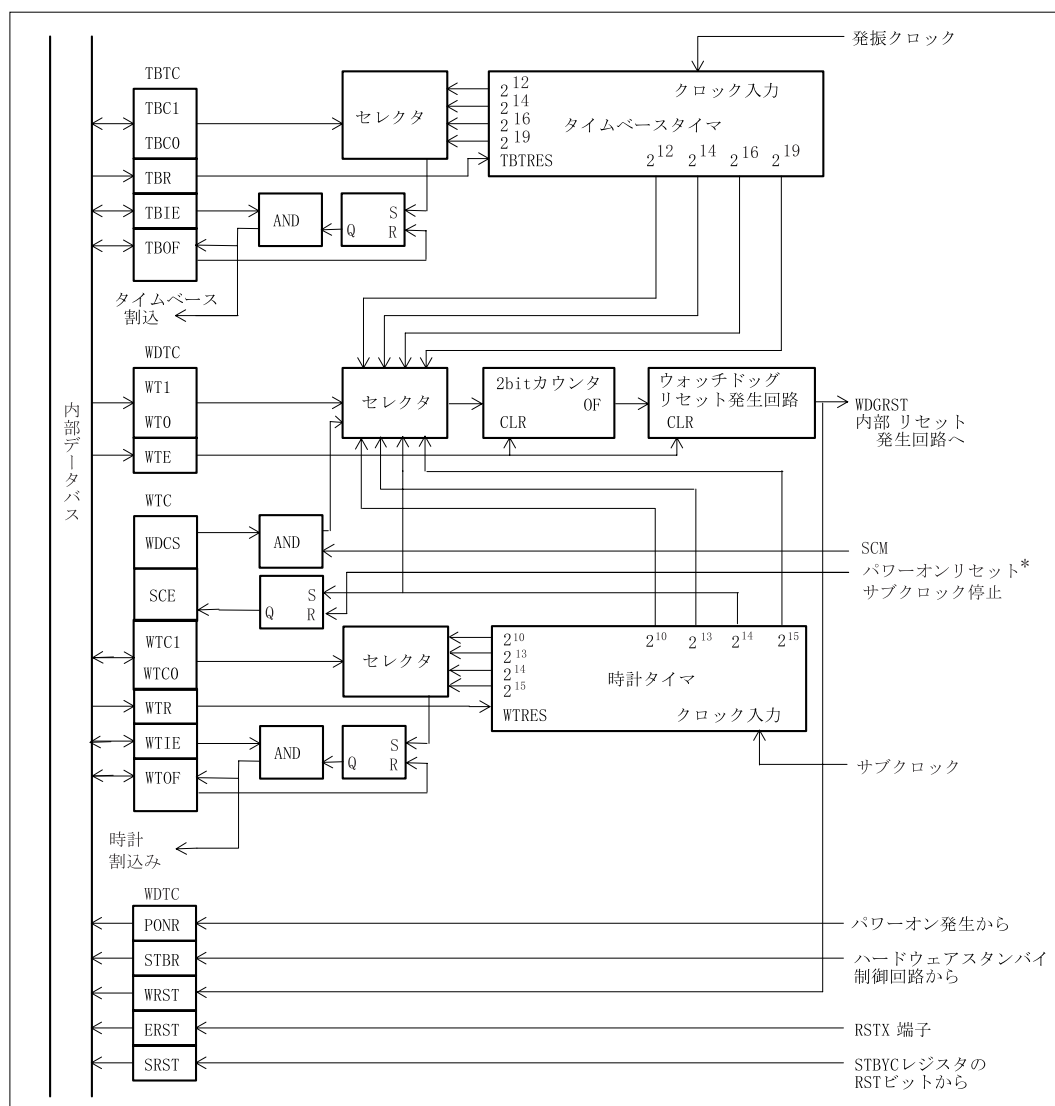


図 9.1-2 タイムベースタイマのブロックダイアグラム

9.2 タイムベースタイマ制御レジスタ (TBTC)

タイムベースタイマ制御レジスタ(TBTC)では、タイムベースタイマの動作を制御し、また、インターバル割込みの時間を制御します。

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									↔ ビットNo.
15	14	13	12	11	10	9	8		
予約	—	—	TBIE	TBOF	TBR	TBC1	TBC0		TBTC
リード/ライト 初期値 ⇒	(-) (1)	(-) (-)	(-) (-)	(R/W) (0)	(R/W) (0)	(W) (1)	(R/W) (0)	(R/W) (0)	

図 9.2-1 タイムベースタイマ制御レジスタ(TBTC)構成

< 注意事項 >

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

【ビット15】試験用ビット

試験用ビットです。書込み時には必ず"1"を書込んでください。

【ビット14, 13】未使用ビット

ビット14, 13は、空きビットです。

【ビット12】TBIE

タイムベースタイマによるインターバル割込みを許可するビットです。"1"のときには割込み許可、"0"のときは割込みを禁止します。リセットにて"0"に初期化します。読み書き可能なビットです。

【ビット11】TBOF

タイムベースタイマの割込み要求フラグです。TBIEビットが"1"のときTBOFが、"1"になると割込み要求を発生します。TBC1,0ビットで設定されたインターバルごとに"1"にセットされます。"0"の書込み、ストップモード、ハードウェアスタンバイモードへの遷移、およびリセットによりクリアされます。"1"の書込みは意味がありません。

リードモディファイライト系命令におけるリード時には"1"が読み出されず。

【ビット10】TBR

タイムベースタイマのカウンタを全ビット"0"にクリアするビットです。"0"を書き込むことによりタイムベースカウンタをクリアします。"1"の書き込みは動作に影響はありません。リード時は"1"が読めます。

【ビット9, 8】TBC1,0

タイムベースタイマのインターバル時間を設定するビットです。リセットにより"00"に初期化されます。読み書き可能なビットです。

表 9.2-1に、タイムベースタイマ インターバル時間の設定を示します。

表 9.2-1 タイムベースタイマ インターバル時間の設定

TBC1	TBC0	原発振4MHz時のインターバル時間
0	0	1.024 ms
0	1	4.096 ms
1	0	16.384 ms
1	1	131.072 ms

9.3 タイムベースタイマの動作

タイムベースタイマは、ウォッチドッグタイマのクロックソース、メインクロック、およびPLLクロックの発振安定待ち時間のタイマで、一定周期ごとに割込みを発生するインターバルタイマの機能があります。

タイムベースカウンタ

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

タイムベースカウンタのクリアは、パワーオンリセット、ストップモード、ハードウェアスタンバイモードへの遷移、クロック選択レジスタ(CKSCR)内のMCSビットによる発振クロックからPLLクロックへの遷移、クロック選択レジスタ(CKSCR)内のSCSビットによる発振クロックからサブクロックへの遷移、およびタイムベースタイマ制御レジスタレジスタ(TBTC)内のTBRビットへの"0"書き込みにより行われます。

タイムベースタイマの出力を使用している、ウォッチドッグタイマとインターバル割込みは、タイムベースカウンタのクリアにより影響を受けます。

タイムベースタイマのインターバル割込み機能

タイムベースカウンタの桁上がり信号により一定周期ごとに割込みを発生します。タイムベースタイマ制御レジスタレジスタ(TBTC)内のTBC1,0ビットで設定されるインターバル時間ごとにTBOFフラグをセットします。このフラグのセットは、最後にタイムベースタイマがクリアされた時間を基準にして行われます。

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

また、メインクロックモードからサブクロックモードに遷移すると、タイムベースタイマは発振クロックの発振安定待ちのタイマとして使用するために、タイムベースタイマがクリアされます。

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

第10章 ウォッチドッグタイマ

この章では、ウォッチドッグタイマの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制御レジスタ(WDTC)
- 10.3 ウォッチドッグタイマの動作

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、18ビットのタイムベースタイマ、または15ビットの時計タイマの桁上がり信号をクロックソースとする2ビットのウォッチドッグカウンタ、コントロールレジスタ、ウォッチドッグリセット制御部により構成されています。

ウォッチドッグタイマのレジスタ一覧

ウォッチドッグタイマ制御レジスタ									ビットNo.
アドレス : 0000A8H									
	7	6	5	4	3	2	1	0	
	PONR	STBR	WRST	ERST	SRST	WTE	WT1	WT0	WDTC
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(W)	(W)	(W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 10.1-1 ウォッチドッグタイマのレジスタ一覧

ウォッチドッグタイマのブロックダイアグラム

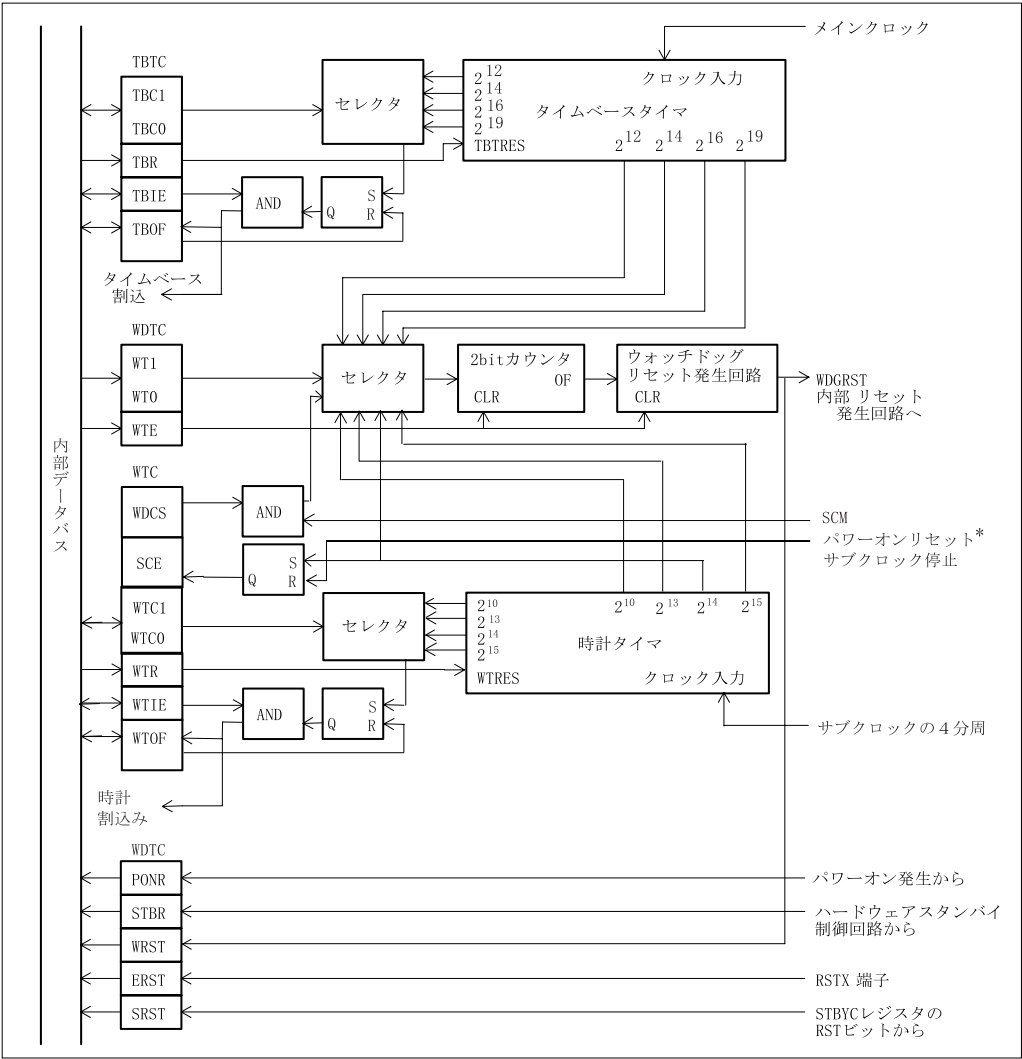


図 10.1-2 ウォッチドッグタイマのブロックダイアグラム

10.2 ウォッチドッグタイマ制御レジスタ(WDTC)

ウォッチドッグタイマ制御レジスタ(WDTC)は、ウォッチドッグタイマ関連の各種制御を行うビットと各種リセット要因を識別するビットで構成しています。

ウォッチドッグタイマ制御レジスタ(WDTC)

ウォッチドッグタイマ制御レジスタ								
	7	6	5	4	3	2	1	0 ← ビットNo.
アドレス : 0000A8H	PONR	STBR	WRST	ERST	SRST	WTE	WT1	WT0
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(W)	(W)	(W)
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

図 10.2-1 ウォッチドッグタイマ制御レジスタ(WDTC)

【ビット7～3】PONR, STBR, WRST, ERST, SRST

リセットの要因を示すフラグです。各種リセット要因が発生すると、本ビットは表 10.2-1に示すように設定されます。これらのビットはすべてWDTCレジスタのリード動作後に"0"にクリアされる読出し専用のレジスタです。なお、パワーオン時は、PONRビット以外のリセット要因ビットの内容は保証されません。したがって、PONRビットが"1"のときはこれ以外のビットの内容を無視するようにしてください。

表 10.2-1 リセット要因ビットとリセット要因

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

* : 前の値を保持します。

【ビット2】WTE

ウォッチドッグタイマが停止状態のときに本ビットに"0"を書き込むことによりウォッチドッグタイマを動作状態にします。2回目以降の"0"の書き込みではウォッチドッグタイマのカウンタをクリアします。"1"の書き込みは動作に影響はありません。

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

【ビット1, 0】WT1,0

ウォッチドッグタイマのインターバル時間を選択する，書き込み専用のビットです。

ウォッチドッグタイマ起動時に書き込まれたデータのみが有効です。ウォッチドッグ起動時以外の書き込みデータは無視されます。なお，時計タイマ制御レジスタ(WTC)内のWDCSビットとクロック選択レジスタ(LPMCR)内のSCMビットをANDした結果によりウォッチドッグタイマに入力するクロックの選択が行われます。つまり，WDCS="1"に設定するとマシナクロックにメインクロック，およびPLLクロックを選択している場合は，タイムベースタイマのインターバル時間を選択し，WDCS="0"またはサブクロックを選択している場合は，時計タイマのインターバル時間を選択します。

表 10.2-2にインターバル時間の設定を示します。

表 10.2-2 WT1,0(インターバル時間選択ビット)

WDCS SCM	WT1	WT0	インターバル時間*	
			最小	最大
1	0	0	約3.58ms	約4.61ms
1	0	1	約14.33ms	約18.43ms
1	1	0	約57.23ms	約73.73ms
1	1	1	約458.75ms	約589.82ms
0	0	0	0.438s	0.563s
0	0	1	3.500s	4.500s
0	1	0	7.000s	9.000s
0	1	1	14.00s	18.00s

*：原振がメインクロック4MHz，サブクロック32KHz時の場合

< 注意事項 >

インターバルの最大値は，ウォッチドッグ動作中にタイムベースカウンタまたは，時計カウンタがリセットされない場合の値です。

10.3 ウォッチドッグタイマの動作

ウォッチドッグタイマの機能は、プログラムの暴走を検出することができます。プログラムの暴走などにより、ウォッチドッグタイマのWTEビットに"0"書き込みが定められた時間内に行われなかった場合、ウォッチドッグタイマよりウォッチドッグリセットが発生します。

ウォッチドッグタイマの起動方法

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

ウォッチドッグタイマリセットの阻止

ウォッチドッグタイマが起動されると、プログラムで定期的に2ビットのウォッチドッグカウンタをクリアする必要があります。具体的には、ウォッチドッグタイマ制御レジスタ(WDTC)内のWTEビットに定期的に"0"を書き込む必要があります。ウォッチドッグカウンタは、タイムベースカウンタの桁上り信号をクロックソースとする2ビットのカウンタにより構成されています。したがって、タイムベースタイマがクリアされると、ウォッチドッグリセットの発生時間が設定より長くなることがあります。

ウォッチドッグタイマの動作を、図 10.3-1に示します。

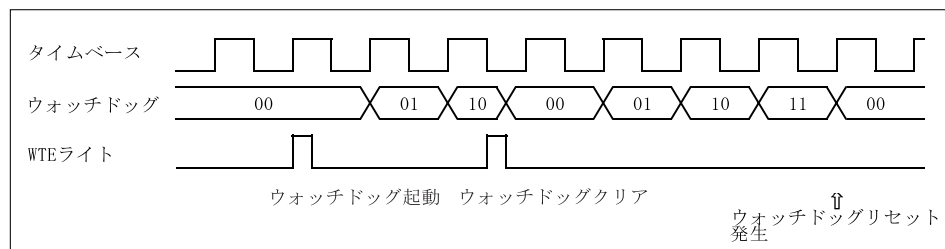


図 10.3-1 ウォッチドッグタイマ動作

ウォッチドッグ停止

ウォッチドッグタイマは一度起動すると、パワーオン、ハードウェアスタンバイ、ウォッチドッグによるリセットでのみ初期化され停止状態となります。

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

ウォッチドッグカウンタのクリア

ウォッチドッグカウンタは、ウォッチドッグタイマ制御レジスタ(WDTC)内のWTEビットへの書き込みによるクリアの他に、リセットの発生、スリープモード、ストップモードへの遷移、ホールドアクノリッジ記号によってクリアされます(なお、時計モードへの変遷では、クリアされません)。

第11章 時計タイマ

この章では、時計タイマの機能と動作について説明します。

11.1 時計タイマの概要

11.2 時計タイマ制御レジスタ(WTC)

11.3 時計タイマの動作

11.1 時計タイマの概要

時計タイマは、15ビットのタイマのほかに、インターバル割込みを制御する回路から構成されています。なお、時計タイマは、クロック選択レジスタ(CKSCR)内のMCSビット、およびSCSビットに関係なく、サブクロックを使用します。

時計タイマのレジスタ

時計タイマ制御レジスタ								ビットNo.
7	6	5	4	3	2	1	0	
WDCS	SCE	WTIE	WTOF	WTR	WTC2	WTC1	WTC0	WTC
(R/W)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
(1)	(X)	(0)	(0)	(0)	(0)	(0)	(0)	

図 11.1-1 時計タイマ制御レジスタ(WTC)

時計タイマのブロックダイアグラム

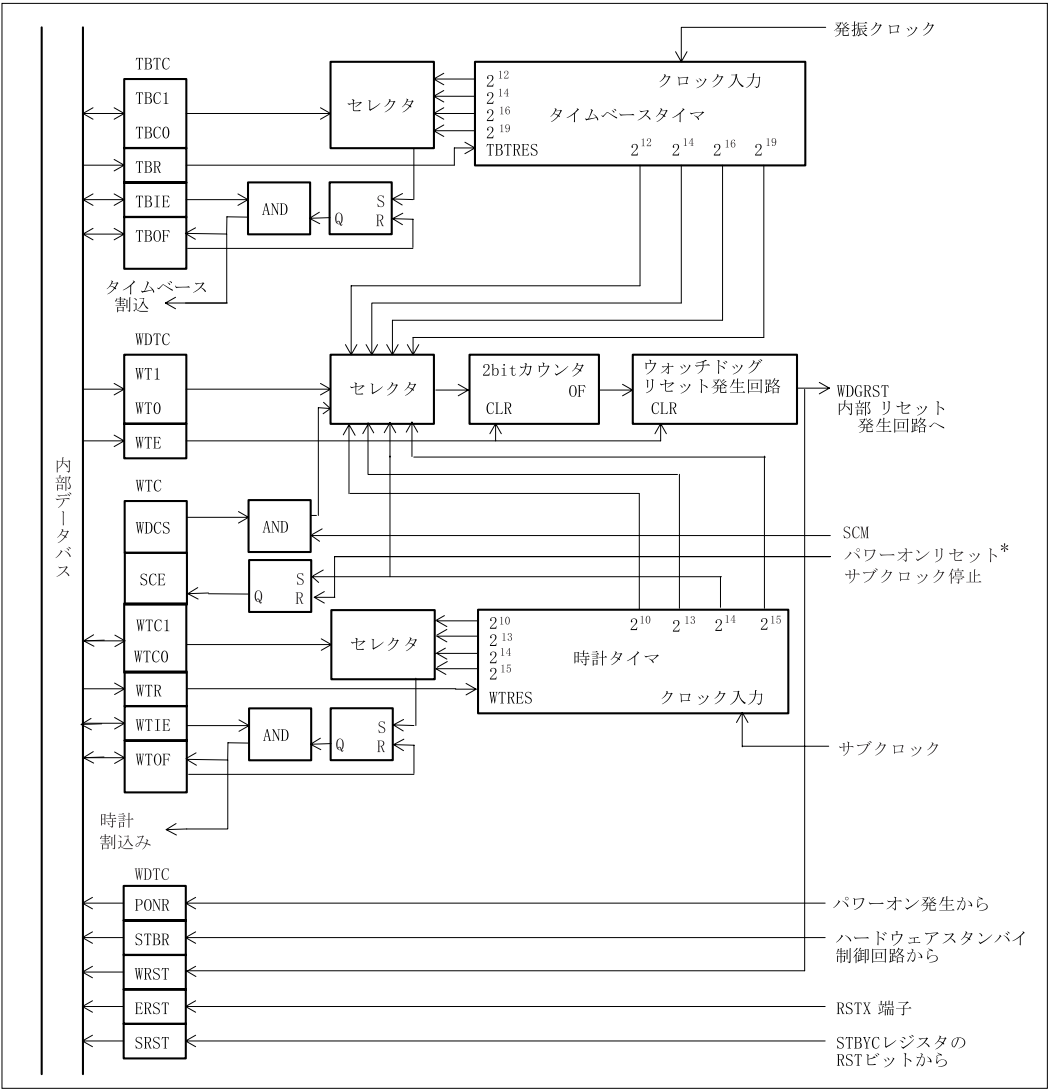


図 11.1-2 時計タイマのブロックダイアグラム

11.2 時計タイマ制御レジスタ(WTC)

時計タイマ制御レジスタ(WTC)は、時計タイマの動作を制御します。また、インターバル割込みの時間を制御します。

時計タイマ制御レジスタ(WTC)

時計タイマ制御レジスタ	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス : 0000AAH	WDCS	SCE	WTIE	WTOF	WTR	WTC2	WTC1	WTC0	WTC
リード/ライト ⇒	(R/W)	(R)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(1)	(X)	(0)	(0)	(0)	(0)	(0)	(0)	

図 11.2-1 時計タイマ制御レジスタ

【ビット7】WDCS

メインクロック、およびPLLクロックを選択しているときに、ウォッチドッグタイマの入力クロックとして、時計タイマのクロック信号を使用するのか、またはタイムベースタイマのクロック信号を使用するのかを選択するビットです。"0"のときは時計タイマのクロックを、"1"のときはタイムベースタイマのクロックを選択します。つまり、WDCS="1"に設定すると、メインクロックおよびPLLクロックを選択しているときにはタイムベースタイマの出力を選択し、サブクロックを選択しているときには、時計タイマを選択することができます。パワーオンリセットにより、本ビットは"1"に初期化されます。

< 注意事項 >

WDCSを"1"に設定した場合には、タイムベースタイマの出力と時計タイマの出力は非同期であるため、ウォッチドッグタイマのカウンタが進む可能性があります。このため、WDCSを"1"に設定した場合には、クロックモードを変更する前後で、ウォッチドッグタイマをクリアする必要があります。

【ビット6】SCE

サブクロックの発振安定待ち時間が経過したことを示すビットです。本ビットが"0"のとき、発振安定時間が現在進行中であることを示しています。発振安定時間は、 2^{16} サイクル(サブクロック)固定です。パワーオンリセット時およびストップ時、ウォッチドッグリセット時に、本ビットは"0"に初期化されます。

【ビット5】WTIE

時計タイマによるインターバル割込みを許可するビットです。本ビットが"1"のときに割込みを許可し、"0"のときに割込みを禁止します。リセットにより本ビットは"0"に初期化されます。読み書き可能なビットです。

【ビット4】WTOF

時計タイマの割込み要求フラグです。WTIEビットが"1"のときに、WTOFを"1"に設定すると割込み要求が発生します。本ビットは、WTC1,0ビットで設定したインターバル時間ごとに"1"に設定されます。本ビットは、"0"の書込み、ストップモード、ハードウェアスタンバイモードへの遷移、およびリセットによりクリアされます。"1"の書込みは動作に影響はありません。

リード/モディファイ/ライト命令によるリード時には、常に"1"が読み出されます。

【ビット3】WTR

時計タイマのカウンタを全ビット"0"にクリアするビットです。本ビットに"0"を書き込むことにより、時計カウンタをクリアします。"1"の書込みは動作に影響はありません。常に"1"が読み出されます。

【ビット2,1,0】WTC2, WTC1, WTC0

時計タイマのインターバル時間を設定するビットです。表 11.2-1にインターバル時間の設定を示します。リセットにより本ビットは"000"に初期化されます。読み書き可能なビットです。

本ビットに書き込むときには、ビット4(WT0F)も同時にクリアしてください。

表 11.2-1 時計タイマ インターバル時間の設定

WTC2	WTC1	WTC0	インターバル時間*
0	0	0	62.5 ms
0	0	1	125 ms
0	1	0	250 ms
0	1	1	500 ms
1	0	0	1.0 s
1	0	1	2.0 s
1	1	0	4.0 s
1	1	1	—

*：インターバル時間は、サブクロック 32KHz(動作クロック8KHz)時の値です。

11.3 時計タイマの動作

時計タイマは、ウォッチドッグカウンタのクロックソース、サブクロックの発振安定待ち時間のためのタイマで、一定周期ごとに割込みを発生するインターバルタイマの機能があります。

時計カウンタ

時計タイマは、マシンのクロックを作成するもとになる原振入力をカウントする15ビットのカウンタにより構成され、原振が入力されている間、常にカウント動作を続けます。時計タイマのクリアは、パワーオンリセット、ストップモード、ハードウェアスタンバイモードへの遷移、および 時計タイマ制御レジスタ(WTC)内のWTRビットへの"0"の書き込みにより行われます。

時計タイマの出力を使用しているウォッチドッグカウンタとインターバル割込みには、時計カウンタのクリアにより影響を受けます。

時計タイマのインターバル割込み機能

インターバル割込み機能は、時計カウンタの桁上がり信号により一定周期ごとに割込みを発生します。時計タイマ制御レジスタ(WTC)内のWTC1,0ビットで設定されるインターバル時間ごとに WTOFフラグをセットします。このフラグ設定のタイミングは、最後に時計タイマがクリアされた時間を基準にして行われます。

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

第12章 16ビット入出力タイマ

この章では、16ビット入出力タイマの機能と動作について説明します。

- 12.1 16ビット入出力タイマの概要
- 12.2 16ビット入出力タイマのブロックダイアグラム
- 12.3 16ビット入出力タイマのレジスタ
- 12.4 16ビットフリーランタイマ
- 12.5 アウトプットコンペア
- 12.6 インプットキャプチャ
- 12.7 16ビットフリーランタイマの動作
- 12.8 16ビットアウトプットコンペアの動作
- 12.9 16ビットインプットキャプチャの動作

12.1 16ビット入出力タイマの概要

16ビット入出力タイマは、16ビットフリーランタイマ1本、アウトプットコンペア4本、インプットキャプチャ2本のモジュールから構成されています。

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

16ビットフリーランタイマ(×1)

16ビットフリーランタイマは、16ビットのアップカウンタ、コントロールレジスタ、プリスケアラより構成されています。本タイマカウンタの出力値は、インプットキャプチャ、アウトプットコンペアの基本時間（ベースタイマ）として使用されます。

カウンタ動作クロック（4種類から選択可能）

内部クロック4種類： /4, /16, /32, /64
：マシクロック

割込み

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

初期化

リセット時、ソフトクリア時、またはコンペアレジスタ0との一致時にカウンタの値を"0000H"に初期化することができます。

アウトプットコンペア(×4)

アウトプットコンペアモジュールは、2本の16ビットコンペアレジスタ、コンペア出力用ラッチ、コントロールレジスタより構成されています。16ビットフリーランタイマの値が、コンペアレジスタの値と一致したとき、出力レベルを反転して割込みが発生します。

- 4本のコンペアレジスタを独立して動作させることができます。
 - 各コンペアレジスタに対応した出力端子と割込みフラグ
- 4本のコンペアレジスタをペアにして出力端子を制御することができます。
 - コンペアレジスタ4本を使用して出力端子の極性を反転することができます。
- 出力端子むけの初期値を設定することができます。
- 割込みはコンペア一致により発生可能です。

インプットキャプチャ (×2)

インプットキャプチャモジュールは、キャプチャレジスタ、コントロールレジスタから構成され、外部入力端子より入力された信号の任意エッジを検出することにより、16ビットフリーランタイム値をキャプチャレジスタに保持し、同時に割込みを発生することができます。

- 外部入力信号の検出エッジを選択可能です。
 - 立上りエッジ、立下りエッジ、両エッジから選択可能
- 2本のインプットキャプチャは独立して動作可能です。
- 割込みは外部入力信号の有効エッジにより発生可能です。
 - インプットキャプチャの割込みにより拡張インテリジェントI/Oサービスを起動できます。

16ビット入出力タイマ部全体のレジスター一覧

図 12.1-1に16ビット入出力タイマ部のレジスター一覧を示します。

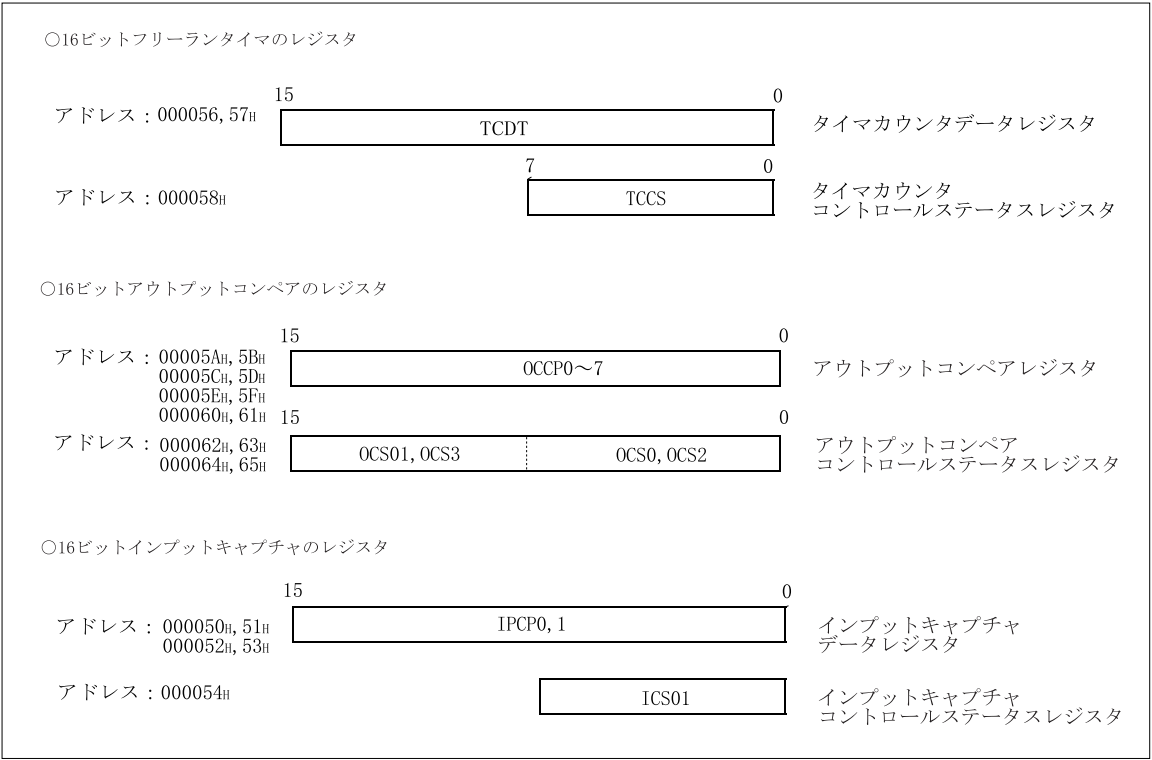


図 12.1-1 16ビットインプットキャプチャのレジスター一覧

12.2 16ビット入出力タイマのブロックダイアグラム

図 12.2-1に , 16ビット入出力タイマのブロックダイアグラムを示します。

16ビット入出力タイマのブロックダイアグラム

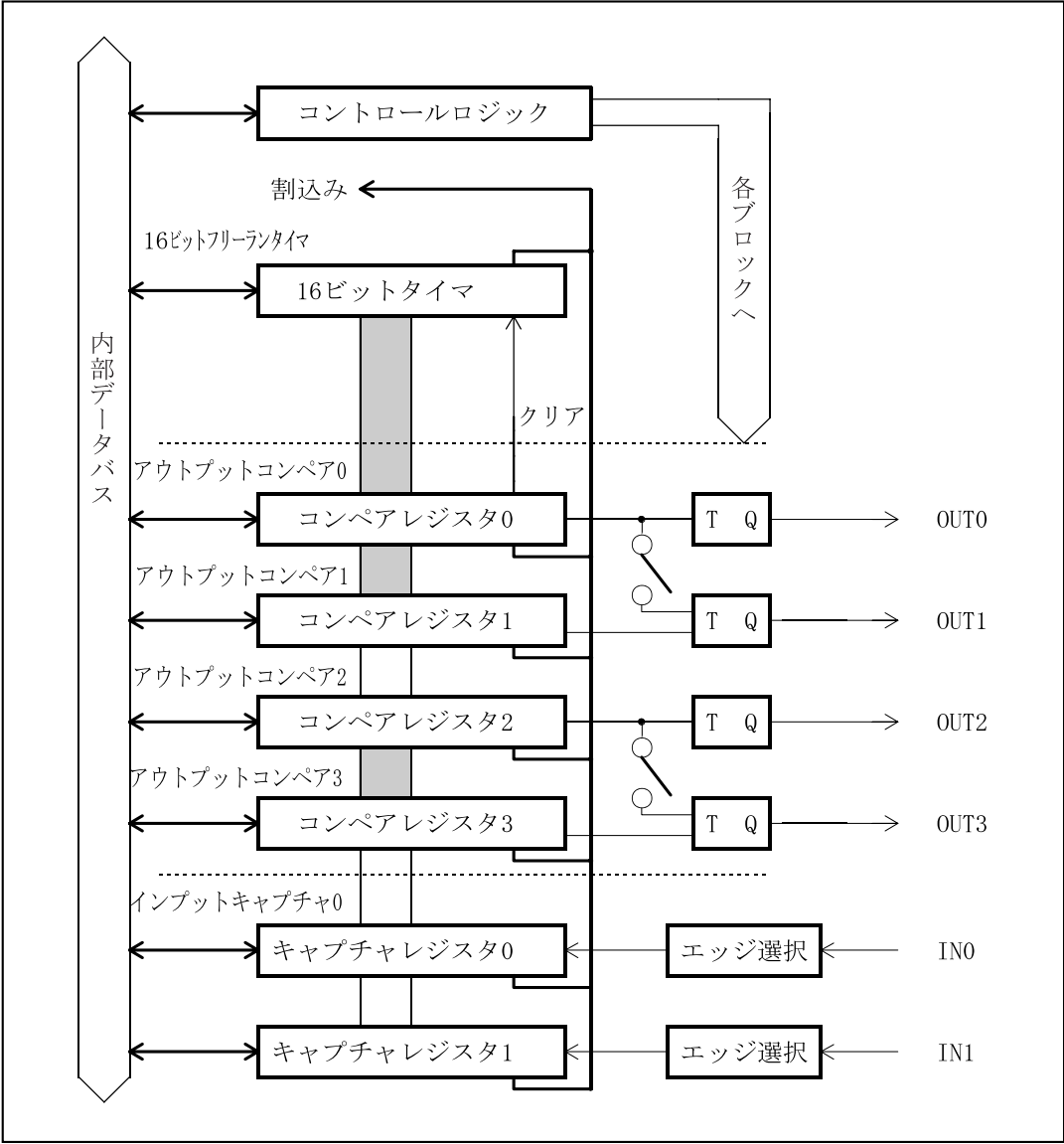


図 12.2-1 16ビット入出力タイマのブロックダイアグラム

12.3 16ビット入出力タイマのレジスタ

16ビット入出力タイマのレジスタには、次の6種類があります。

- ・タイマカウンタデータレジスタ(TCDT)
- ・タイマカウンタコントロールステータスレジスタ(TCCS)
- ・アウトプットコンペアレジスタ(OCCP0～3)
- ・アウトプットコンペアコントロールステータスレジスタ(OCS0～3)
- ・インプットキャプチャデータレジスタ(IPCP0,1)
- ・インプットキャプチャコントロールステータスレジスタ(ICS01)

16ビット入出力タイマのレジスタ

タイマカウンタデータレジスタ上位															
								15	14	13	12	11	10	9	8
アドレス:000057H								T15	T14	T13	T12	T11	T10	T09	T08
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
タイマカウンタデータレジスタ下位															
								7	6	5	4	3	2	1	0
アドレス:000056H								T07	T06	T05	T04	T03	T02	T01	T00
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
タイマカウンタコントロールステータスレジスタ															
								7	6	5	4	3	2	1	0
アドレス:000058H								予約	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
アウトプットコンペアレジスタ上位															
								15	14	13	12	11	10	9	8
アドレス:ch0 00005BH								C15	C14	C13	C12	C11	C10	C09	C08
ch1 00005DH															
ch2 00005FH															
ch3 000061H															
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
アウトプットコンペアレジスタ下位															
								7	6	5	4	3	2	1	0
アドレス:ch0 00005AH								C07	C06	C05	C04	C03	C02	C01	C00
ch1 00005CH															
ch2 00005EH															
ch3 000060H															
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)
アウトプットコンペアコントロールステータスレジスタ上位															
								15	14	13	12	11	10	9	8
アドレス:ch1 000063H								—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0
ch3 000065H															
リード/ライト ⇒								(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒								(-)	(-)	(-)	(0)	(0)	(0)	(0)	(0)
アウトプットコンペアコントロールステータスレジスタ下位															
								7	6	5	4	3	2	1	0
アドレス:ch0 000062H								ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0
ch2 000064H															
リード/ライト ⇒								(R/W)	(R/W)	(R/W)	(R/W)	(-)	(-)	(R/W)	(R/W)
初期値 ⇒								(0)	(0)	(0)	(0)	(-)	(-)	(0)	(0)

図 12.3-1 16ビット入出力タイマのレジスタ (続く)

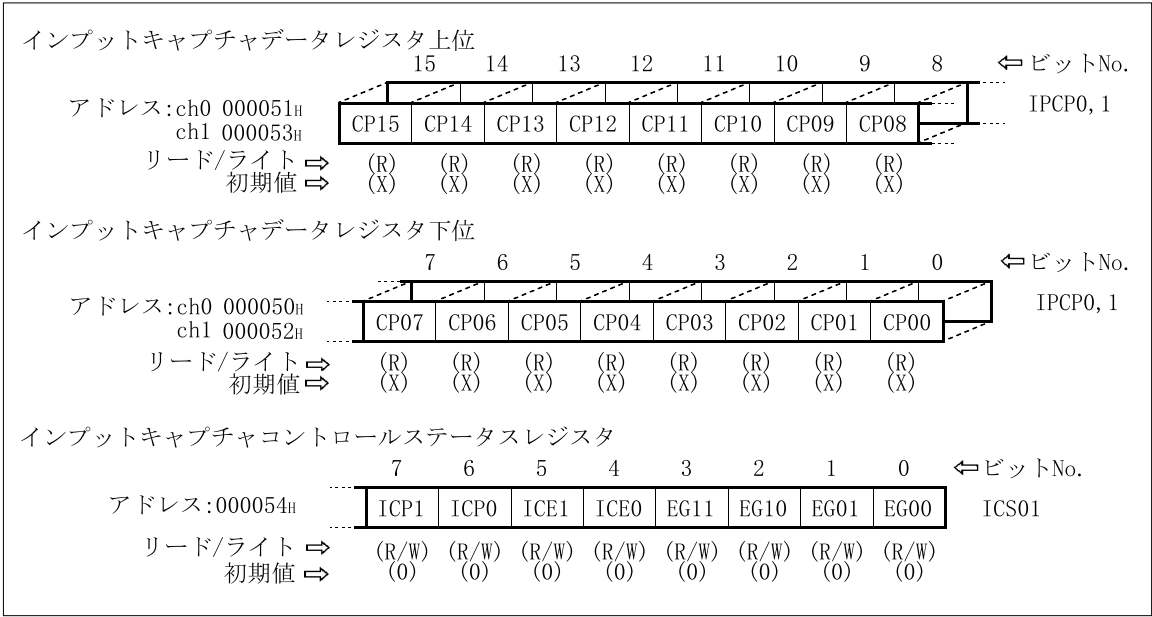


図 12.3-1 16ビット入出力タイマのレジスタ (続き)

12.4 16ビットフリーランタイム

16ビットフリーランタイムのレジスタには、次の2つがあります。

- ・ タイマカウンタデータレジスタ(TCDT)
- ・ タイマカウンタコントロールステータスレジスタ(TCCS)

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タ値はアウトプットコンペア、インプットキャプチャの基本時間（ベースタイム）として使用されます。

- カウントクロックは4種類から選択可能です。
- カウンタオーバーフロー割込みを発生することができます。
- モード設定によりアウトプットコンペアのコンペアレジスタ0の値との一致によるカウンタの初期化が可能です。

16ビットフリーランタイムのレジスタ一覧

16ビットフリーランタイムのレジスタ構成を以下に示します。

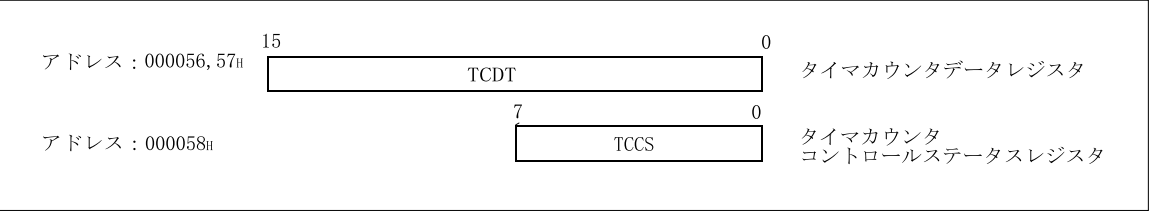


図 12.4-1 16ビットフリーランタイムのレジスタ一覧

16ビットフリーランタイムのブロックダイアグラム

図 12.4-2に16ビットフリーランタイムのブロックダイアグラムを示します。

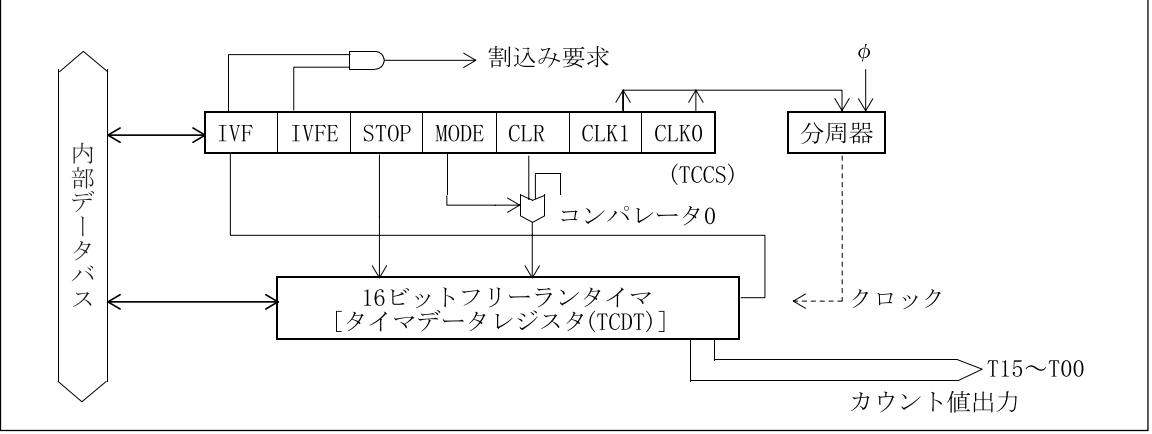


図 12.4-2 16ビットフリーランタイムのブロックダイアグラム

12.4.1 タイマカウンタデータレジスタ(TCDT)

タイマカウンタデータレジスタ(TCDT)は、16ビットフリーランタイマのカウンタ値を読出すことができるレジスタです。

タイマカウンタデータレジスタ(TCDT)

タイマカウンタデータレジスタ(TCDT)では、カウンタ値はリセット時に"0000_H"にクリアされます。このレジスタに書込むことで、タイマ値を設定できますが、必ず停止(STOP=1)状態で行ってください。

16ビットフリーランタイマの初期化は次の要因で行われます。

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

タイマカウンタデータレジスタ上位										
	15	14	13	12	11	10	9	8	⇐ビットNo.	
アドレス:000057H	T15	T14	T13	T12	T11	T10	T09	T08	TCDT (High)	
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)		
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)		
タイマカウンタデータレジスタ下位										
	7	6	5	4	3	2	1	0	⇐ビットNo.	
アドレス:000056H	T07	T06	T05	T04	T03	T02	T01	T00	TCDT (Low)	
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)		
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)		

図 12.4-3 タイマカウンタデータレジスタ(TCDT)

< 注意事項 >

TCDTレジスタは、ワードアクセスしてください。

12.4.2 タイマカウンタコントロールステータスレジスタ (TCCS)

タイマカウンタコントロールステータスレジスタ(TCCS)は、16ビットフリーランタイムのタイマカウンタを制御するレジスタです。

タイマカウンタコントロールステータスレジスタ(TCCS)

タイマカウンタコントロールステータスレジスタ								
	7	6	5	4	3	2	1	0 ⇐ビットNo.
アドレス:000058H	予約	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0 TCCS
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 12.4-4 タイマカウンタコントロールステータスレジスタ(TCCS)

【ビット7】予約ビット

ビット7は、予約ビットです。このビットには、必ず"0"を書き込んでください。

【ビット6】IVF

IVFは、16ビットフリーランタイムの割込み要求フラグです。
16ビットフリーランタイムがオーバーフローしたとき、またはモード設定によりコンペアレジスタ0との一致によってカウンタクリアされたときに"1"にセットされます。割込みが発生するのは、IVFEビット（ビット5）がセットされた場合です。
本ビットは"0"書込みによりクリアされます。"1"書込みは意味を持ちません。リードモディファイ系命令では"1"が読めます。

表 12.4-1 IVF(割込み要求フラグ)の機能

IVF	機能
0	割込み要求なし（初期値）
1	割込み要求あり

【ビット5】IVFE

IVFEは、16ビットフリーランタイムの割込み許可ビットです。
本ビットが"1"の場合、IVFビット（ビット5）が"1"にセットされると割込みが発生します。

表 12.4-2 IVFE(割込み許可ビット)の機能

IVFE	機能
0	割込み禁止（初期値）
1	割込み許可

【ビット4】STOP

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

表 12.4-3 STOP(カウントを停止ビット)の機能

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

(注) 16ビットフリーランタイマのカウントが停止すると、アウトプットコンペア動作も停止します。

【ビット3】MODE

MODEは、16ビットフリーランタイマの初期化条件を設定するビットです。"0"の場合はリセットとCLRビット(ビット2)でカウンタ値を初期化可能です。"1"の場合は、リセットとCLRビット(ビット2)の他にアウトプットコンペアのコンペアレジスタ0の値との一致によりカウンタ値を初期化することができます。

表 12.4-4 MODE(初期化条件設定ビット)の機能

MODE	機能
0	リセット, クリアビットによる初期化(初期値)
1	リセット, クリアビット, コンペアレジスタ0による初期化

(注) カウンタ値の初期化はカウント値の変化点で行われます。

【ビット2】CLR

CLRは、動作中の16ビットフリーランタイマ値を"0000_H"に初期化するビットです。"1"を書込み時にカウンタ値を"0000_H"に初期化します。"0"を書込んだり意味を持ちません。このビットから読み出すと、必ず"0"が読めます。カウンタ値の初期化はカウント値の変化点で行われます。

表 12.4-5 CLR(初期化ビット)の機能

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

(注) タイマ停止中にカウンタの値を初期化する場合は、データレジスタに"0000_H"を書き込んでください。

【ビット1, 0】CLK1, CLK0

CLK1, CLK0は、16ビットフリーランタイマのカウントクロックを選択するビットです。本ビットに値が書込まれた後すぐにクロックを更新します。したがって、これらのビットに値を書き込む前に、必ずアウトプットコンペア動作とインプットキャプチャ動作を停止してください。

表 12.4-6 CLK1, CLK0(カウントクロック選択ビット)

CLK1	CLK0	カウントクロック	=16MHz	= 8MHz	= 4MHz	= 1MHz
0	0	/4	0.25 μ s	0.5 μ s	1 μ s	4 μ s
0	1	/16	1 μ s	2 μ s	4 μ s	16 μ s
1	0	/64	4 μ s	8 μ s	16 μ s	64 μ s
1	1	/256	16 μ s	32 μ s	64 μ s	256 μ s

= マシンクロック

12.5 アウトプットコンペア

アウトプットコンペアは、次の2つより構成されます。

- ・ アウトプットコンペアレジスタ (OCCP0 ~ 3)
- ・ アウトプットコンペアコントロールステータスレジスタ (OCS0 ~ 3)

アウトプットコンペア

コンペアレジスタに設定された値と16ビットフリーランタイム値が一致したら、端子の出力レベルを反転すると共に割込みを発生することができます。

- コンペアレジスタは全部で2本あり、独立して動作させることができます。また設定により2本のコンペアレジスタを使い端子の出力を制御可能です。
- 端子の出力初期値を設定することができます。
- 割込みはコンペア一致により発生可能です。

アウトプットコンペアのレジスタ一覧

アウトプットコンペアのレジスタ構成を以下に示します。

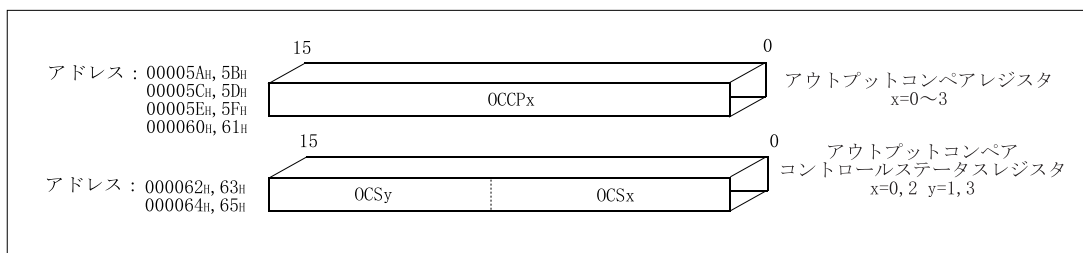


図 12.5-1 アウトプットコンペアのレジスタ一覧

アウトプットコンペアのブロックダイアグラム

図 12.5-2にアウトプットコンペアのブロックダイアグラムを示します。

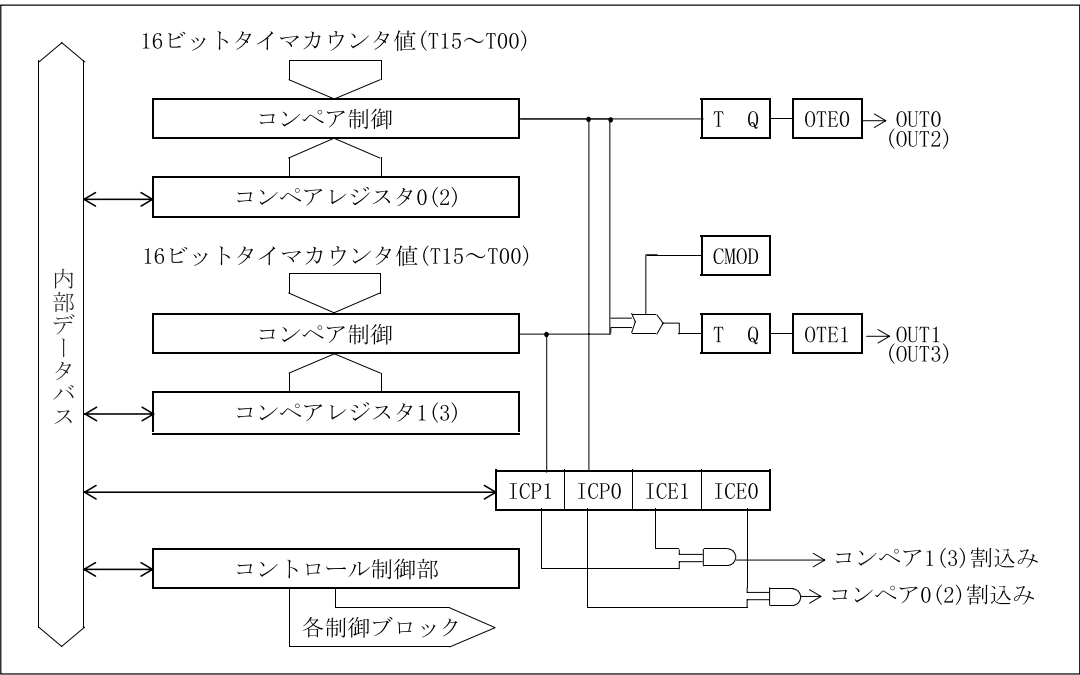


図 12.5-2 アウトプットコンペアのブロックダイアグラム

12.5.1 アウトプットコンペアレジスタ(OCCP0～3)

アウトプットコンペアレジスタ(OCCP0～3)は、16ビットフリーランタイムと比較する16ビット長のコンペアレジスタです。

アウトプットコンペアレジスタ(OCCP0～3)

アウトプットコンペアレジスタ(OCCP0～3)では、レジスタ値は初期値不定ですので設定してから起動許可してください。本レジスタ値と16ビットフリーランタイム値が一致した時コンペア信号が発生してアウトプットコンペア割込みフラグをセットします。また、出力許可をしている場合はコンペアレジスタに対応した出力レベルを反転します。

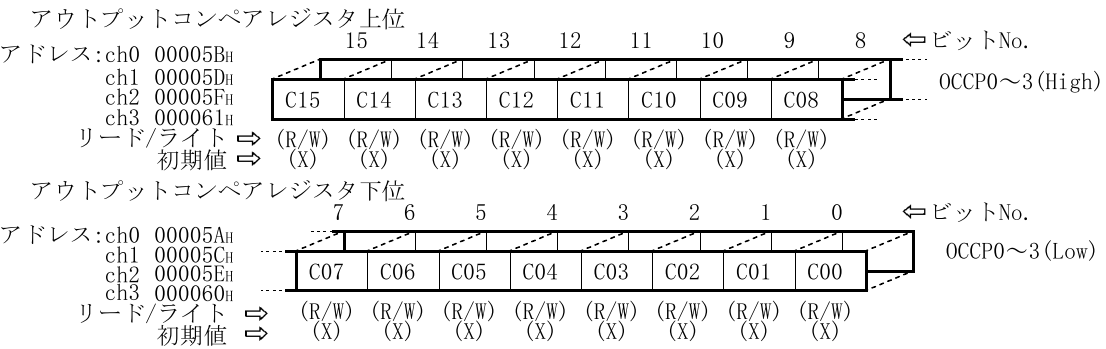


図 12.5-3 アウトプットコンペアレジスタ(OCCP0～3)

< 注意事項 >

OCCP0～3レジスタは、ワードアクセスしてください。

12.5.2 アウトプットコンペアコントロールステータスレジスタ(OCS0~3)

アウトプットコンペアコントロールステータスレジスタ(OCS0~3)は、16ビットフリーランタイムを制御するレジスタです。

アウトプットコンペアコントロールステータスレジスタ(OCS0~3)

アウトプットコンペアコントロールステータスレジスタ上位									
	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス:ch0 000063 _H	—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	OCS0, 2
ch1 000065 _H									
リード/ライト ⇒	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(-)	(-)	(0)	(0)	(0)	(0)	(0)	
アウトプットコンペアコントロールステータスレジスタ下位									
	7	6	5	4	3	2	1	0	⇐ビットNo.
アドレス:ch0 000062 _H									OCS1, 3
ch1 000064 _H	ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0	
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(-)	(-)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(-)	(-)	(0)	(0)	

図 12.5-4 アウトプットコンペアコントロールステータスレジスタ(OCS0~3)

【ビット15-13】未使用ビット

【ビット12】CMOD

CMODでは、端子出力を許可した場合(OTE1=1または、OTE0=1)のコンペア一致における端子出力レベル反転動作モードを切り換えます。

CMOD=0のとき（初期値）

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

- OUT0/2：コンペアレジスタ0の一致によりレベルを反転します。
- OUT1/3：コンペアレジスタ1の一致によりレベルを反転します。

CMOD=1のとき

CMOD=1のときは、コンペアレジスタ0はCMOD=0時と同じく出力レベルを反転しますが、コンペアレジスタ1に対応した端子(OUT1)の出力レベルは、コンペアレジスタ0の一致とコンペアレジスタ1の一致の両方で出力レベルを反転します。コンペアレジスタ0と1が同値の場合はコンペアレジスタ1本の時と同じ動作をします。

- OUT0/2：コンペアレジスタ0の一致によりレベルを反転します。
- OUT1/3：コンペアレジスタ0と1の一致によりレベルを反転します。

【ビット11, 10】OTE1, 0

OTE1, 0は、アウトプットコンペアの端子出力を許可するビットです。これらのビットの初期値は"0"です。

表 12.5-1 OTE1, 0(端子出力許可ビット)の機能

OTE1, 0	機能
0	汎用ポートとして動作します。[初期値]
1	アウトプットコンペア端子出力になります。

< 注意事項 >

OTE1 : アウトプットコンペア1/3に対応

OTE0 : アウトプットコンペア0/2に対応

【ビット9, 8】OTD1, OTD0

OTD1, OTD0ビットは、アウトプットコンペアの端子出力を許可した場合の端子出力レベルを変更する時に使用します。コンペア端子出力の初期値は"0"となります。書込み時はコンペア動作を停止してから行ってください。読み出し時はアウトプットコンペア端子出力値がよめます。

表 12.5-2 OTD1, OTD0(端子出力レベル変更ビット)の機能

OTD1, OTD0	機能
0	コンペア端子出力を"0"にします [初期値]
1	コンペア端子出力を"1"にします

< 注意事項 >

OTD1 : アウトプットコンペア1/3に対応

OTD0 : アウトプットコンペア0/2に対応

【ビット7, 6】ICP1, ICP0

ICP1, ICP0は、アウトプットコンペアの割込みフラグです。コンペアレジスタと16ビットフリーランタイム値が一致した場合に"1"にセットされます。割込み要求ビット(ICE1, ICE0) が許可されているときに本ビットがセットされるとアウトプットコンペア割込みが発生します。

本ビットは"0"書込みによりクリアされ、"1"書込みでは意味をもちません。リードモディファイ系命令では"1"が読めます。

表 12.5-3 ICP1, ICP0(アウトプットコンペア割込みビット)の機能

ICP1, ICP0	機能
0	コンペア一致なし [初期値]
1	コンペア一致あり

< 注意事項 >

ICP1 : アウトプットコンペア1/3に対応

ICP0 : アウトプットコンペア0/2に対応

【ビット5, 4】ICE1, ICE0

ICE1, ICE0は、アウトプットコンペアの割込み許可ビットです。本ビットが"1"の場合、割込みフラグ(ICP0, ICP1) がセットされるとアウトプットコンペア割込みが発生します。

表 12.5-4 ICE1, ICE0(アウトプットコンペアの割込み許可ビット)の機能

ICE1, ICE0	機能
0	アウトプットコンペア割込み禁止 [初期値]
1	アウトプットコンペア割込み許可

< 注意事項 >

ICE1 : アウトプットコンペア1/3に対応

ICE0 : アウトプットコンペア0/2に対応

【ビット3, 2】未使用ビット

【ビット1, 0】CST1, CST0

CST1, CST0は、16ビットフリーランタイムとの一致動作を許可するビットです。コンペア動作許可をする前にコンペアレジスタ値を設定してください。

表 12.5-5 CST1, CST0(16ビットフリーランタイムとの一致動作許可ビット)

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

< 注意事項 >

CST1 : アウトプットコンペア1/3に対応

CST0 : アウトプットコンペア0/2に対応

OCS0 ~ 3レジスタは、16ビットフリーランタイムのクロックと同期させているため、16ビットフリーランタイムを停止させるとコンペア動作も停止します。

12.6 インพุットキャプチャ

<p>インพุットキャプチャには、次の2つのレジスタがあります。</p> <ul style="list-style-type: none"> ・インพุットキャプチャデータレジスタ(IPCP0,1) ・インพุットキャプチャコントロールステータスレジスタ(ICS01)

<p>インพุットキャプチャ</p> <p>インพุットキャプチャは、インพุットキャプチャデータレジスタ、コントロールレジスタで構成されています。</p> <p>各インพุットキャプチャは、それぞれに対応した外部入力端子を持っています。</p> <ul style="list-style-type: none"> ● 外部入力の有効エッジを3種類から選択可能です。 <p>立上りエッジ () / 立下りエッジ () / 両エッジ ()</p> <ul style="list-style-type: none"> ● 外部入力の有効エッジを検出したときに割込みを発生することができます。

<p>インพุットキャプチャ全体のレジスタ一覧</p> <p>インพุットキャプチャ全体のレジスタ構成を以下に示します。</p>
--

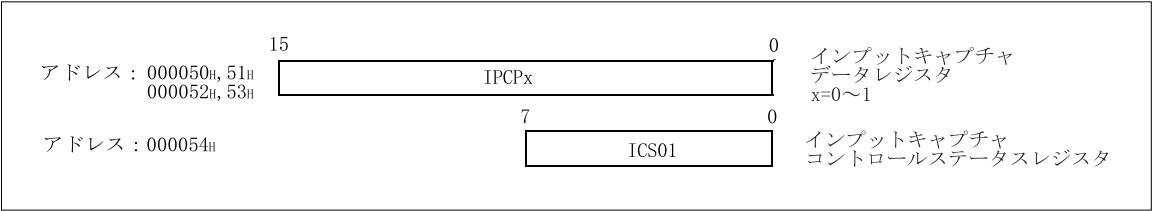


図 12.6-1 インพุットキャプチャ全体のレジスタ一覧

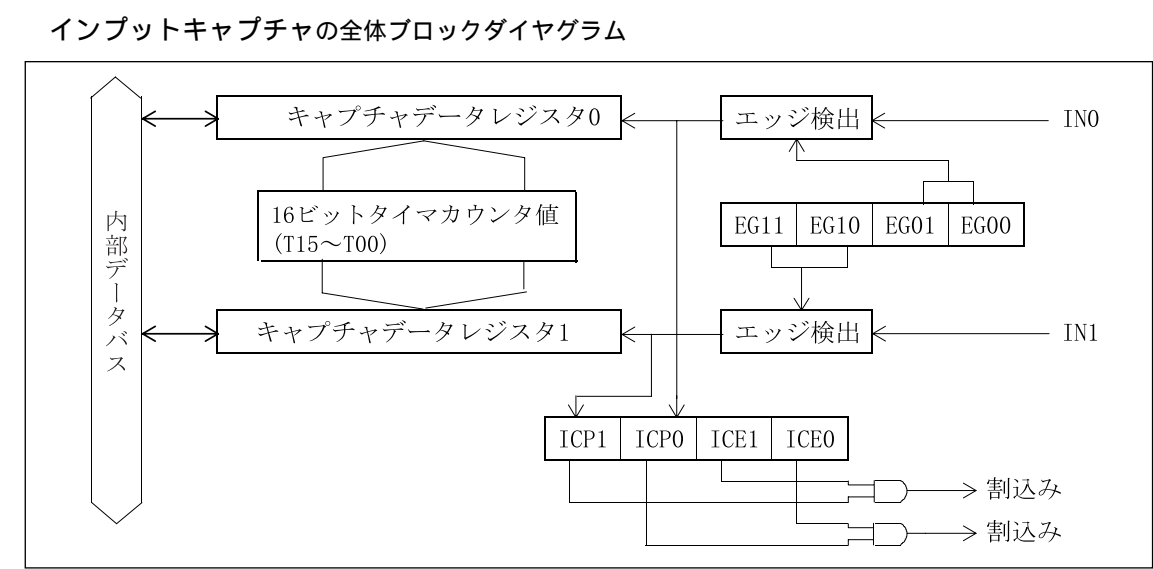


図 12.6-2 インพุットキャプチャのブロックダイアグラム

12.6.1 インプットキャプチャデータレジスタ(IPCP0,1)

インプットキャプチャデータレジスタ(IPCP0,1)は、対応した外部端子入力波形の有効エッジを検出したとき、16ビットフリーランタイム値を保持するレジスタです。

インプットキャプチャデータレジスタ(IPCP0,1)

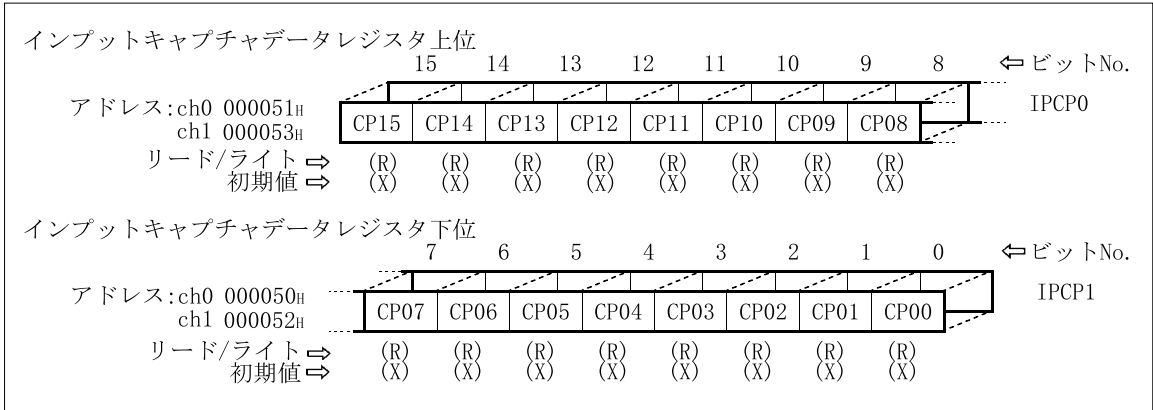


図 12.6-3 インプットキャプチャデータレジスタ(IPCP0,1)

< 注意事項 >

IPCP0,1レジスタは、ワードアクセスしてください。書き込みはできません。

12.6.2 インพุットキャプチャコントロールステータスレジスタ(ICS01)

インพุットキャプチャコントロールステータスレジスタ(ICS01)は、16ビットフ
リーランタイムを制御するレジスタです。

インพุットキャプチャコントロールステータスレジスタ(ICS01)

インพุットキャプチャコントロールステータスレジスタ								
	7	6	5	4	3	2	1	0
アドレス:000054h	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
								ICS01

図 12.6-4 インพุットキャプチャコントロールステータスレジスタ(ICS01)

< 注意事項 >

ICS01レジスタは、バイトアクセスしてください。

【ビット7, 6】ICP1, ICP0

ICP1, ICP0は、インพุットキャプチャ割込みフラグです。

外部入力端子の有効エッジを検出すると本ビットを"1"にセットします。割
込み許可ビット(ICE0, ICE1) がセットされていると有効エッジを検出すること
により割込みを発生することができます。

本ビットは"0"書き込みによりクリアされます。"1"書き込みは意味を持ちま
せん。リードモディファイライト系命令では"1"が読めます。

表 12.6-1 ICP1, 0(インพุットキャプチャ割込みフラグ)の機能

ICP1, 0	機能
0	有効エッジ検出なし (初期値)
1	有効エッジ検出あり

< 注意事項 >

ICP1：インพุットキャプチャ1に対応
ICP0：インพุットキャプチャ0に対応

【ビット5, 4】ICE1, ICE0

ICE1, ICE0は、インプットキャプチャ割込み許可ビットです。本ビットが"1"の場合、割込みフラグ(ICP0, ICP1) がセットされるとインプットキャプチャ割込みが発生します。

表 12.6-2 ICE1,0 (インプットキャプチャ割込み許可ビット)の機能

ICE1,0	機能
0	割込み禁止 (初期値)
1	割込み許可

< 注意事項 >

ICE1 : インプットキャプチャ1に対応

ICE0 : インプットキャプチャ0に対応

【ビット3, 2, 1, 0】EG11, EG10, EG01, EG00

EG11, EG10, EG01, EG00ビットは、外部入力の有効エッジ極性を指定します。インプットキャプチャ動作許可も兼用しています。

表 12.6-3 EGx1, EGx0(外部入力の有効エッジ極性指定ビット)の機能

EG11 EG01	EG10 EG00	エッジ検出極性
0	0	エッジ検出なし (停止状態) (初期値)
0	1	立上りエッジ検出
1	0	立下りエッジ検出
1	1	両エッジ検出

< 注意事項 >

EG01, EG00 : インプットキャプチャ0に対応

EG11, EG10 : インプットキャプチャ1に対応

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

16ビットフリーランタイムの動作

カウンタ値は、次の条件でクリアされます。

- オーバフローが発生した場合
- アウトプットコンペアレジスタ0値とコンペアマッチした場合（モード設定が必要）
- 動作中にTCCSレジスタのCLRビットに"1"を書き込んだ場合
- 停止中にTCDCレジスタに"0000H"を書き込んだ場合
- リセット時

割込みはオーバフローが発生したとき、コンペアレジスタ0値とコンペアマッチしてカウンタクリアされたとき発生することができます（コンペアマッチ割込みはモード設定が必要です）。

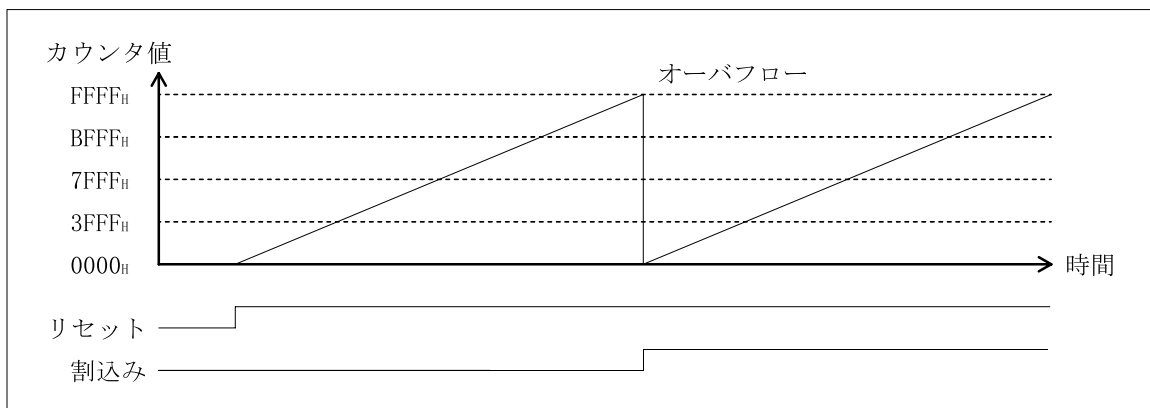


図 12.7-1 オーバフローによるカウンタクリア

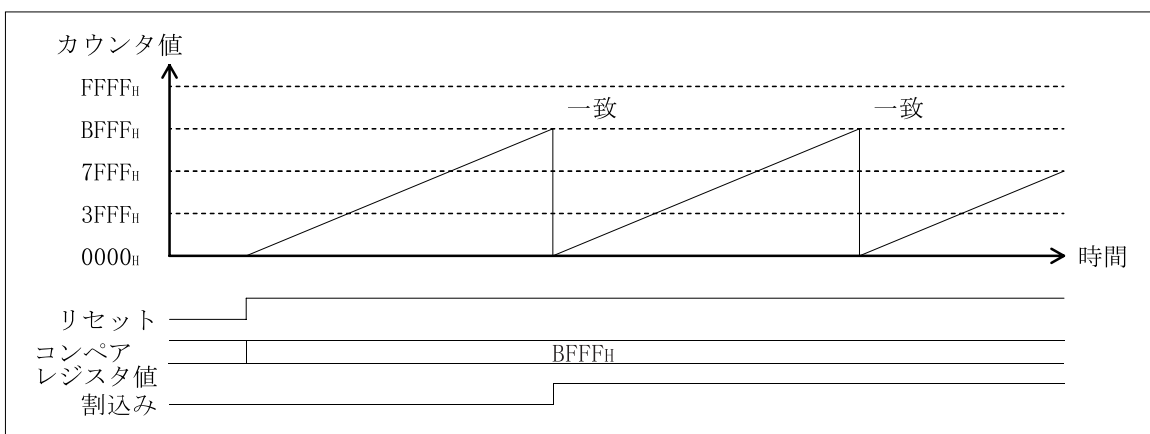


図 12.7-2 アウトプットコンペアレジスタ0値とコンペアマッチしたときのカウンタクリア

16ビットフリーランタイマのカウントタイミング

16ビットフリーランタイマは、入力されたクロック（内部または外部クロック）によりカウントアップされます。外部クロック選択時は立上りエッジでカウントされます。

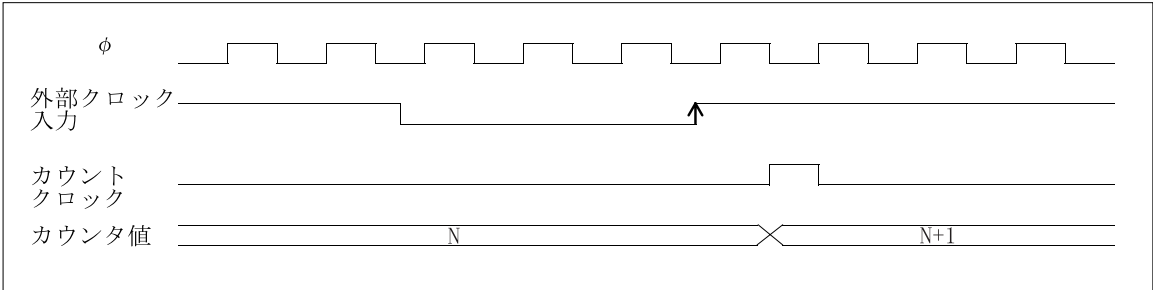


図 12.7-3 フリーランタイマのカウントタイミング

カウンタのクリアはリセット、ソフトウェアクリア時、コンペアレジスタ0との一致で行うことができます。リセットまたはソフトウェアクリアによって、カウンタはクリアされます。コンペアレジスタ0との一致によるカウンタクリアは、カウントタイミングに同期して行われます。

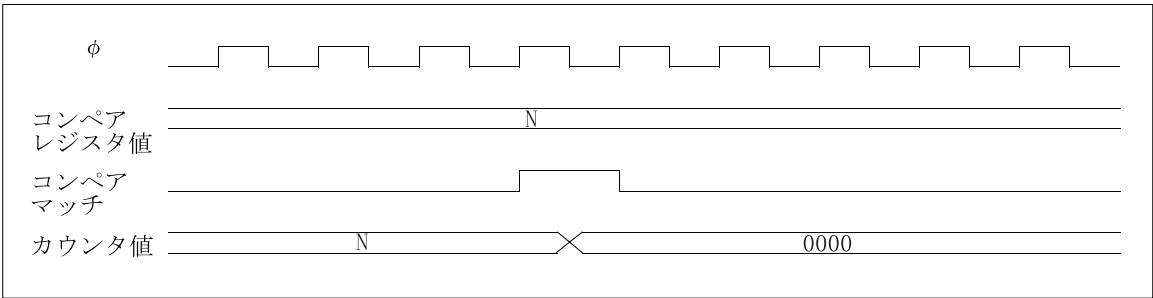


図 12.7-4 フリーランタイマのクリアタイミング（コンペアレジスタ0との一致）

12.8 16ビットアウトプットコンペアの動作

16ビットアウトプットコンペアは、設定されたコンペアレジスタ値と16ビットフリーランタイムとの値を比較して一致したら割込み要求フラグをセットするとともに、出力レベルを反転することができます。

16ビットアウトプットコンペアの動作

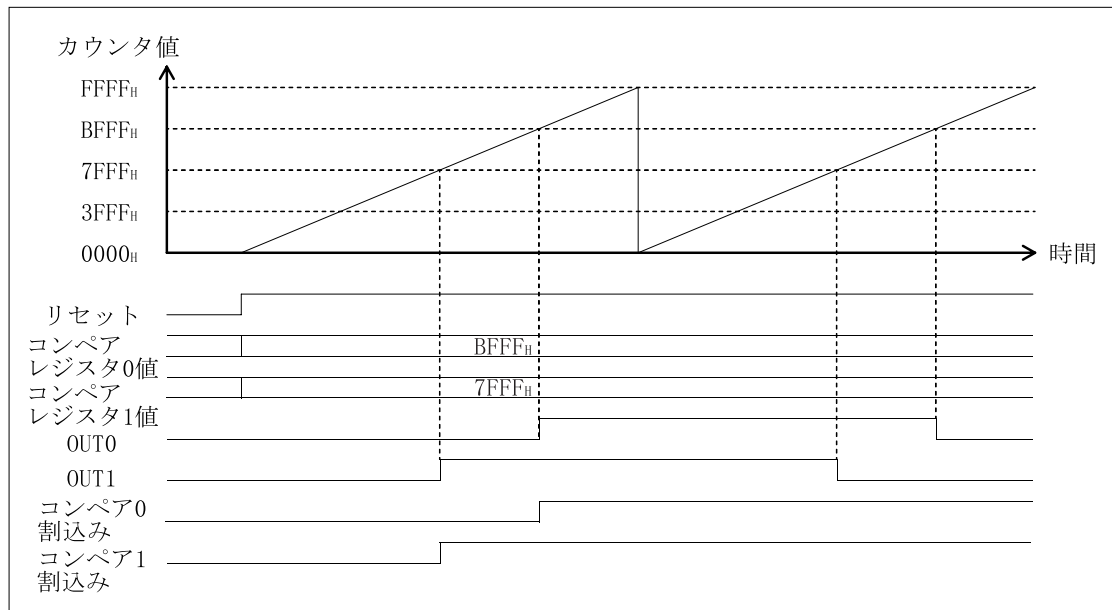


図 12.8-1 コンペアレジスタ0,1 を使用したときの出力波形例（出力の初期値は0とする）

図 12.8-2に示すように、2組のコンペアレジスタを使い出力レベルを変えることができます(CMOD=1 の時)。

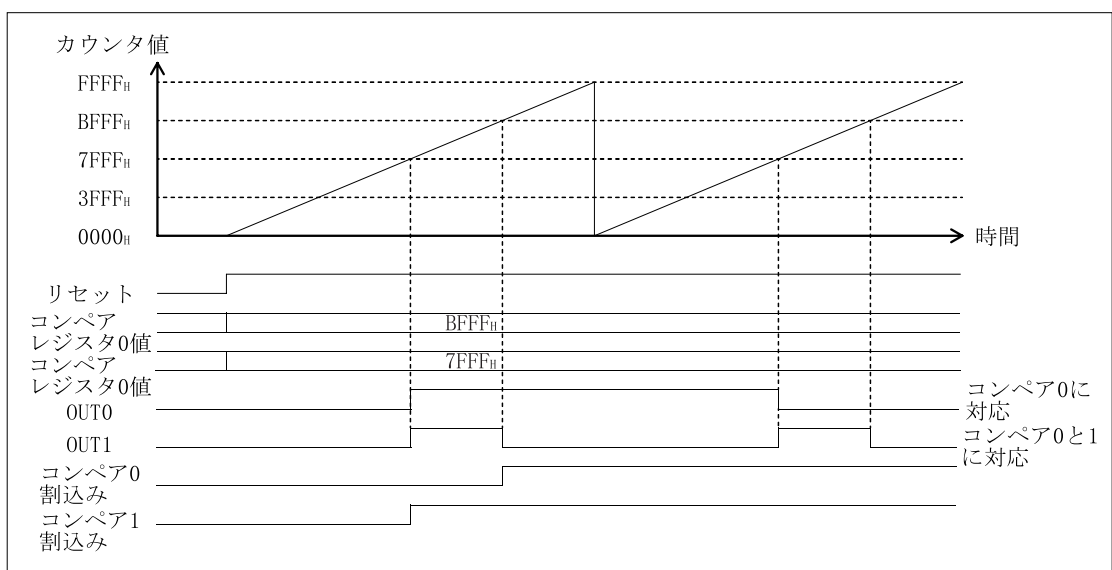


図 12.8-2 2組のコンペアレジスタによる出力波形例（出力の初期値は0とする）

16ビットアウトプットコンペアのタイミング

アウトプットコンペアは、フリーランタイムと設定したコンペアレジスタの値が一致したときにコンペアマッチ信号が発生して出力値を反転するとともに割込みを発生することができます。

コンペアマッチ時の出力反転タイミグはカウンタのカウントタイミグに同期して行われます。

コンペアレジスタ書換え時のカウンタ値とはコンペアしません。

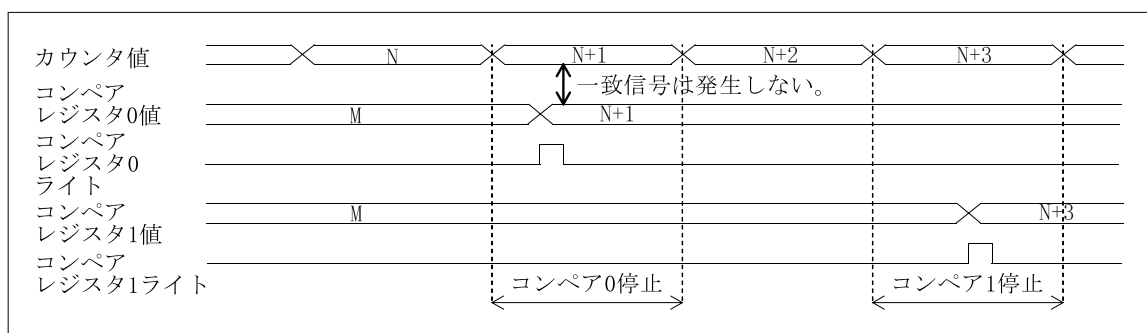


図 12.8-3 コンペアレジスタ書換え時のコンペア動作

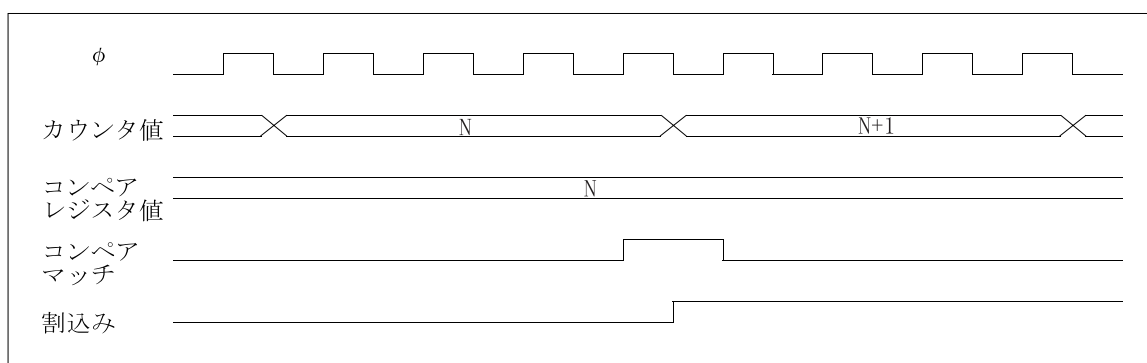


図 12.8-4 アウトプットコンペアの割込みタイミング

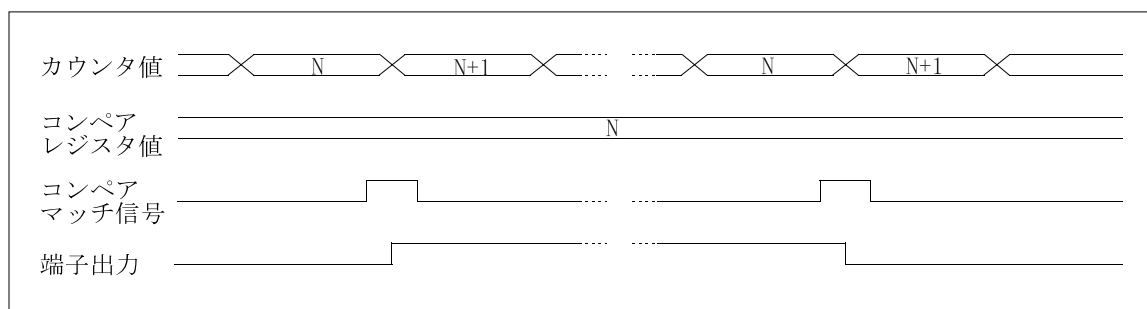


図 12.8-5 アウトプットコンペアの出力端子変化タイミング

12.9 16ビットインプットキャプチャの動作

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

16ビットインプットキャプチャの動作

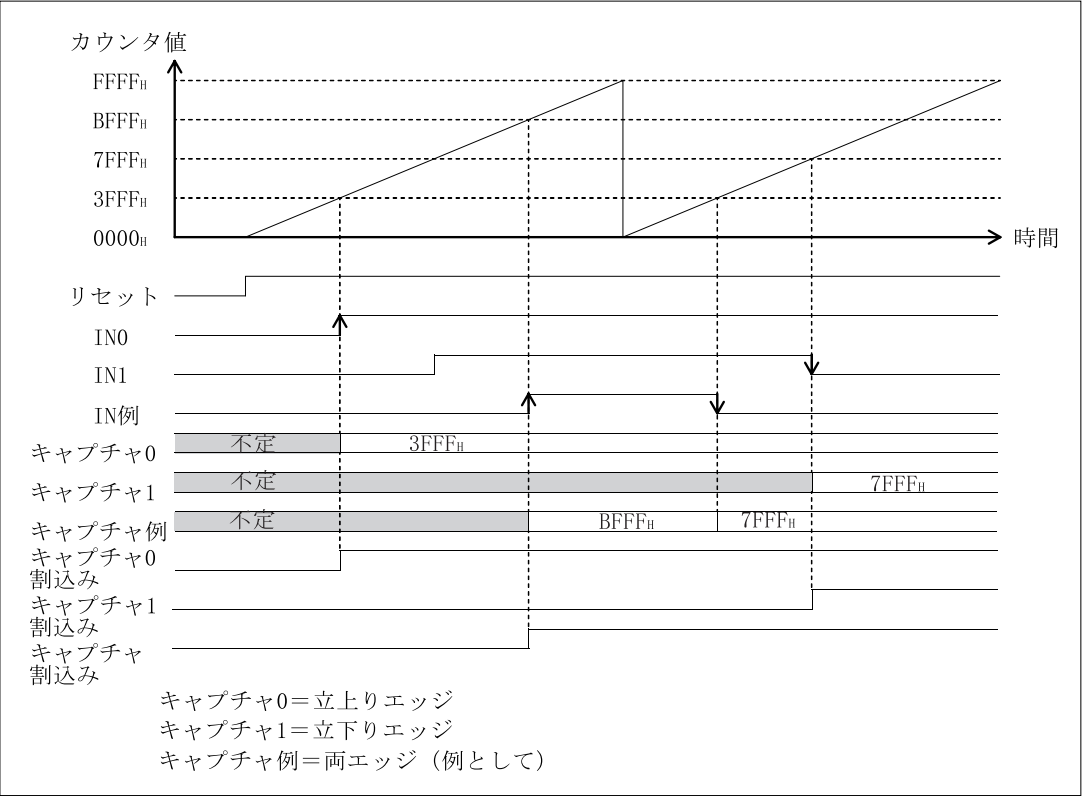


図 12.9-1 インプットキャプチャの取り込みタイミング例

インプットキャプチャ入力タイミング

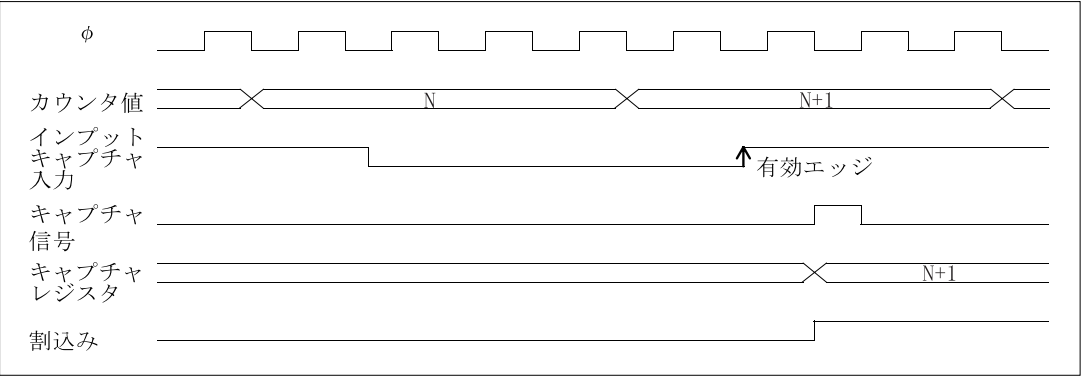


図 12.9-2 入力信号に対するキャプチャタイミング

第13章 8/16ビットPPG

この章では、8/16ビットPPGの機能と動作について説明します。

13.1 8/16ビットPPGの概要

13.2 8/16ビットPPGのブロックダイアグラム

13.3 8/16ビットPPGのレジスタ

13.4 8/16ビットPPGの動作

13.1 8/16ビットPPGの概要

8/16ビットPPGは、8/16ビットのリロードタイマモジュールで、タイマ動作に応じたパルス出力制御により、PPG出力を行います。

8/16ビットPPGの概要

8/16ビットPPGは、ハードウェアとして、8ビットダウンカウンタ2個、8ビットリロードレジスタ4個、16ビット制御レジスタ1個、外部パルス出力端子2本、割込み出力2本で構成されており、以下の機能を実現しています。

8ビットPPG出力2チャンネル独立動作モード

2チャンネルの独立したPPG出力動作が可能です。

16ビットPPG出力動作モード

1チャンネルの16ビットPPG出力動作が可能です。

8 + 8ビットPPG出力動作モード

ch0の出力を、ch1のクロック入力とすることにより、任意周期の8ビットPPG出力動作が可能です。

PPG出力動作

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

13.2 8/16ビットPPGのブロックダイアグラム

図 13.2-1に、8/16ビットPPG(ch0)のブロックダイアグラムを、図 13.2-2に、8/16ビットPPG(ch1)のブロックダイアグラムを示します。

8/16ビットPPGのブロックダイアグラム

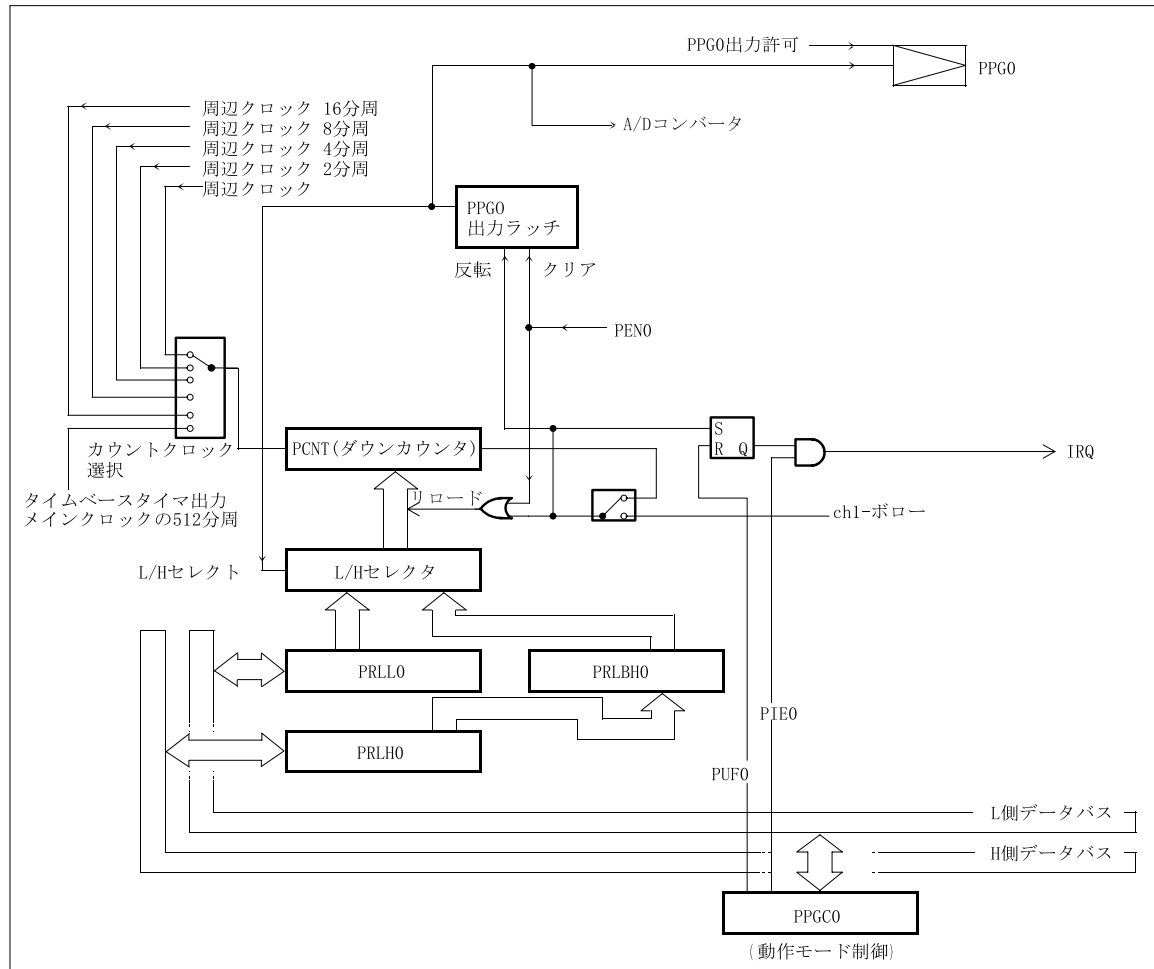


図 13.2-1 8/16ビットPPG(ch0)のブロックダイアグラム

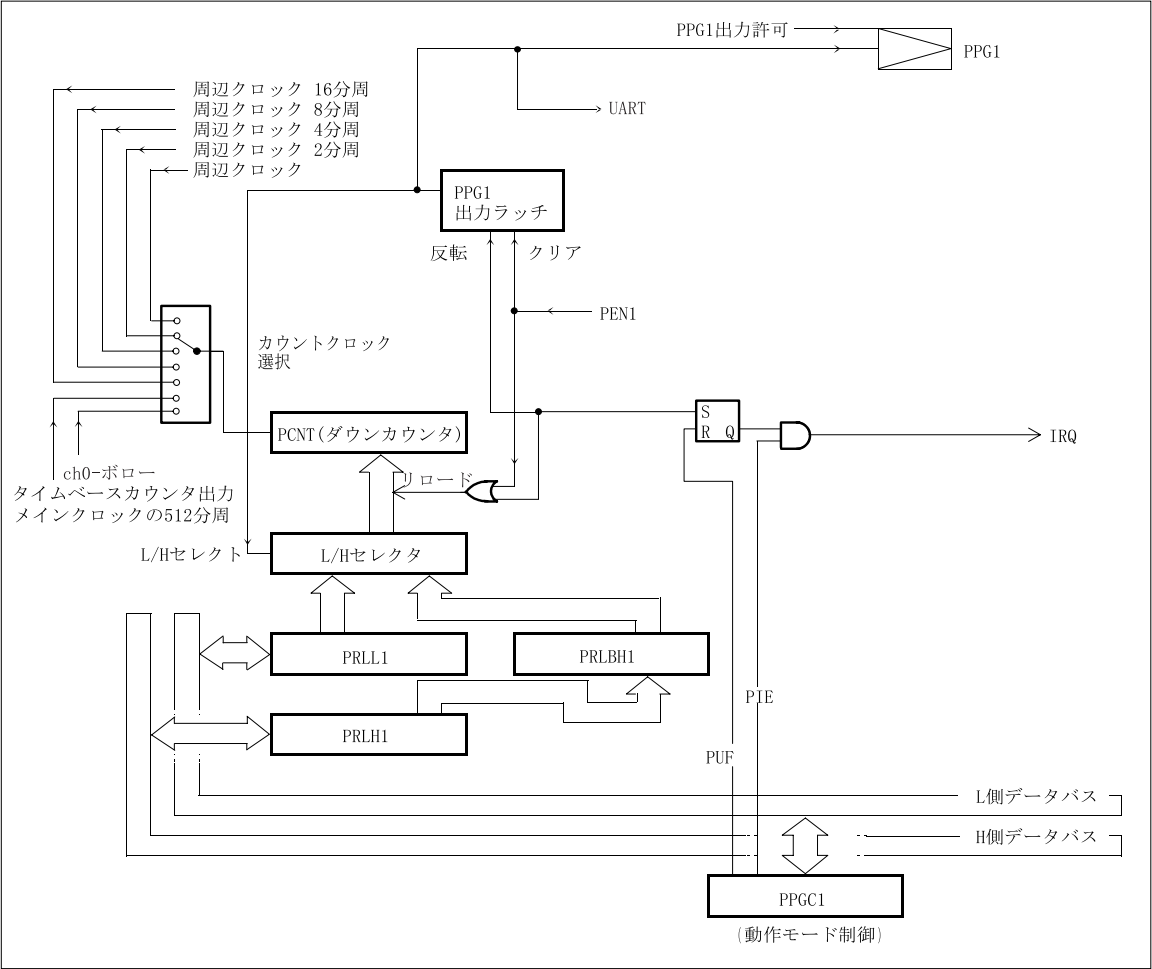


図 13.2-2 8/16ビットPPG(ch1)のブロックダイアグラム

13.3 8/16ビットPPGのレジスタ

8/16ビットPPGのレジスタには、次の3種類があります。

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

8/16ビットPPGのレジスタ

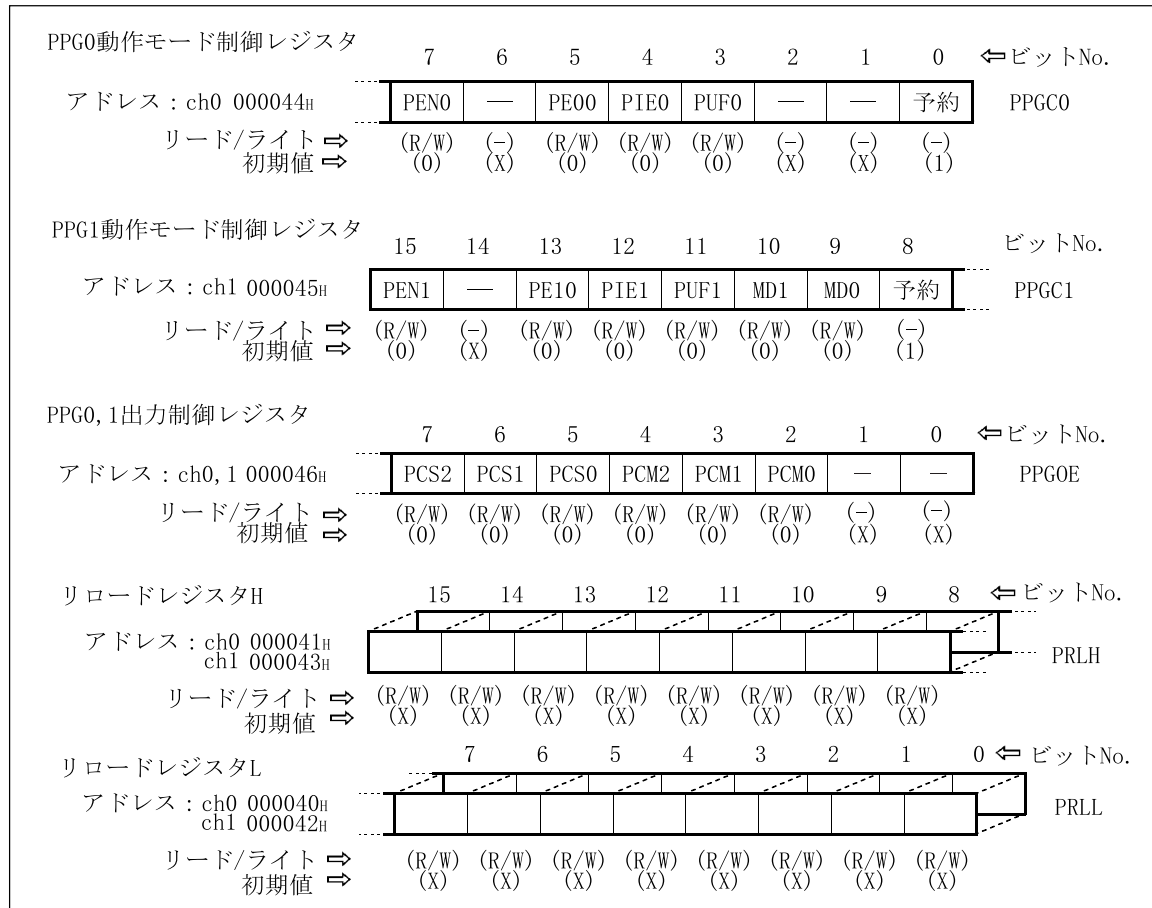


図 13.3-1 8/16ビットPPGのレジスタ

13.3.1 PPG0動作モード制御レジスタ(PPGC0)

PPG0動作モード制御レジスタ(PPGC0)は、8/16ビットPPGの動作モードの選択、端子出力制御、カウントクロック選択、トリガの制御を行うレジスタです。

PPG0動作モード制御レジスタ(PPGC0)

PPG0動作モード制御レジスタ	7	6	5	4	3	2	1	0	↔ビットNo.
アドレス : ch0 000044h	PEN0	—	PE00	PIE0	PUF0	—	—	予約	PPGC0
リード/ライト ⇒	(R/W)	(-)	(R/W)	(R/W)	(R/W)	(-)	(-)	(-)	
初期値 ⇒	(0)	(-)	(0)	(0)	(0)	(-)	(-)	(1)	

図 13.3-2 PPG0動作モード制御レジスタ(PPGC0)

【ビット7】PEN0(Ppg ENable)

PPGの動作開始および動作モードを、表 13.3-1で示すように選択します。本ビットに1を書き込むことでPPGはカウントを開始します。リセットにより、本ビットは"0"に初期化されます。読み出しと書込みが可能です。

表 13.3-1 PEN0(動作許可ビット)の機能

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

【ビット5】PE00(Ppg output Enable 00)

パルス出力外部端子PG00を、表 13.3-3で示すように制御します。リセットにより、本ビットは"0"に初期化されます。本ビットは、読み出しと書込みが可能です。

表 13.3-2 PE00(PPG0端子出力許可ビット)の機能

PE00	機能
0	汎用ポート端子(パルス出力禁止) [初期値]
1	PG00=パルス出力端子(パルス出力許可)

【ビット4】PIE0(Ppg Interrupt Enable)

PPGの割込み許可を、表 13.3-3で示すように制御します。
本ビットが"1"の場合、PUF0 = "1"になると割込み要求を発生します。
本ビットが"0"の場合は割込み要求を発生しません。リセットにより、本ビットは"0"に初期化されます。本ビットは、読み出しと書込みが可能です。

表 13.3-3 PIE0(PPGの割込み許可ビット)の機能

PIE0	機能
0	割込み禁止 [初期値]
1	割込み許可

< 注意事項 >

16ビットリロードタイマch0と同じ割込みベクトル番号が割り当てられていますので、16ビットリロードタイマch0で拡張インテリジェントI/Oサービス(EI²OS)を使用する場合には、PIE0ビットを"0"にしてください。

【ビット3】PUF0(Ppg Underflow Flag)

PPGのカウンタアンダフロービットを表 13.3-4に示すように制御します。

8ビットPPG2chモードおよび8ビットプリスケラ+8ビットPPGモードの場合には、ch0のカウンタの値が00_H~FF_Hになった時のアンダフローにより"1"にセットされます。16ビットPPG1chモードの場合には、ch1/ch0のカウンタの値が0000_H~FFFF_Hになったときのアンダフローにより"1"にセットされます。"0"の書き込みにより"0"になります。

このビットへの"1"書き込みは、意味がありません。

リードモディファイライトのリード時には"1"が読めます。

リセットにより、本ビットは、"0"に初期化されます。本ビットは、読み出しと書き込みが可能です。

表 13.3-4 PUF0(PPGカウンタアンダフロービット)の機能

PUF0	機能
0	PPGのカウンタアンダフローが検出していません。[初期値]
1	PPGのカウンタアンダフローを検出しました。

【ビット0】予約ビット

予約ビットです。PPGC0を設定するときは必ず"1"に設定してください。

13.3.2 PPG1動作モード制御レジスタ(PPGC1)

PPG1動作モード制御レジスタ(PPGC1), 8/16ビットPPGの動作モードの選択・端子出力制御・カウントクロック選択, トリガの制御を行うレジスタです。

PPG1動作モード制御レジスタ(PPGC1)

PPG1動作モード制御レジスタ								ビットNo.	
								15	14
								13	12
								11	10
								9	8
アドレス: ch1 000045 _H								PEN1	—
								PE10	PIE1
								PUF1	MD1
								MD0	予約
リード/ライト ⇒ (R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W) (R/W)								(-) (X)	(-) (1)
初期値 ⇒ (0) (0) (0) (0) (0) (0) (0) (0)									

図 13.3-3 PPG1動作モード制御レジスタ(PPGC1)

【ビット15】PEN1(Ppg ENable)

PPGの動作開始および動作モードを, 表 13.3-5で示すように選択します。本ビットに1を書き込むことでPWMはカウントを開始します。リセットにより, 本ビットは"0"に初期化されます。読出しと書込みが可能です。

表 13.3-5 動作許可ビット(PEN1)の機能

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

【ビット13】PE10(Ppg output Enable 10)

パルス出力外部端子PG10を, 表 13.3-6で示すように制御します。リセットにより, 本ビットは"0"に初期化されます。読出しと書込みが可能です。

表 13.3-6 PE10(PPG10端子出力許可ビット)の機能

PE10	機能
0	汎用ポート端子(パルス出力禁止) [初期値]
1	PPG1 = パルス出力端子(パルス出力許可)

【ビット12】PIE1(Ppg Interrupt Enable)

PPGの割込み許可を, 表 13.3-7で示すように制御します。本ビットが"1"の場合, PUF1 = "1"になると割込み要求を発生します。本ビットが"0"の場合は割込み要求を発生しません。

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

表 13.3-7 PIE1(PPGの割込み許可ビット)の機能

PIE1	機能
0	割込み禁止 [初期値]
1	割込み許可

< 注意事項 >

16ビットリロードタイマch1と同じ割込みベクトル番号が割り当てられていますので、16ビットリロードタイマch1で、拡張インテリジェントI/Oサービス(EI²OS)を使用する場合には、PIE1ビットを"0"にしてください。

【ビット11】PUF1(Ppg Underflow Flag)

PPGのカウンタアンダフロービットを、表 13.3-8に示すように制御します。

8ビットPPG2chモードおよび8ビットプリスケラ+8ビットPPGモードの場合には、ch1のカウンタの値が00_H~FF_Hへなった時のアンダフローにより"1"にセットされます。16ビットPPG1chモードの場合には、ch1/ch0のカウンタの値が0000_H~FFFF_Hへなったときのアンダフローにより"1"にセットされます。"0"の書き込みにより"0"になります。このビットへの"1"書き込みは、意味がありません。リードモディファイライトのリード時には"1"が読めます。

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

表 13.3-8 PUF1(PPGカウンタアンダフロービット)の機能

PUF1	機能
0	PPGのカウンタアンダフローを検出していません [初期値]
1	PPGのカウンタアンダフローを検出しました。

【ビット10, 9】MD2, 1(ppg count MoDe)

PPGタイマの動作モードを、表 13.3-9に示すように選択します。リセットにより、本ビットは"00"に初期化されます。

本ビットは、読出しと書き込みが可能です。

表 13.3-9 MD2, 1(動作モード選択ビット)の機能

MD1	MD0	動作モード
0	0	8ビットPPG2ch独立モード [初期値]
0	1	8ビットプリスケラ+8ビットPPG1chモード
1	0	予約 (設定禁止)
1	1	16ビットPPG1chモード

< 注意事項 >

- ・本ビットを"10"に設定しないでください。
- ・本ビットを"01"に設定する場合には、PPGC0レジスタのPEN0ビット/PPGC1レジスタのPEN1ビットを"01"に設定しないでください。また、PEN0ビットおよびPEN1ビットを同時に"11"または"00"にセットすることを推奨します。
- ・本ビットを"11"に設定する場合には、PPGC0/1をワード転送で書き換え、PEN0ビットおよびPEN1ビットを同時に"11"または"00"にセットしてください。

【ビット8】予約ビット

予約ビットです。必ず"1"に設定してください

13.3.3 PPG0,1出力端子制御レジスタ(PPGOE)

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

PPG0,1出力端子制御レジスタ(PPGOE)

PPG0,1出力制御レジスタ								
								↔ビットNo.
アドレス : ch0, 1 000046H	7	6	5	4	3	2	1	0
	PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	—	—
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(X)	(X)

図 13.3-4 PPG0,1出力端子制御レジスタ(PPGE)

【ビット7～5】PCS2～0(Ppg Count Select)

ch1のダウンカウンタの動作クロックを、表 13.3-10に示すように選択します。リセットにより、本ビットは"000"に初期化されます。本ビットは、読出しと書込みが可能です。

表 13.3-10 PCS2～0(カウントクロック選択ビット)の機能

PCS2	PCS1	PCS0	動作モード
0	0	0	周辺クロック (62.5ns/マシニングロック 16MHz時)
0	0	1	周辺クロック/2 (125 ns/マシニングロック 16MHz時)
0	1	0	周辺クロック/4 (250 ns/マシニングロック 16MHz時)
0	1	1	周辺クロック/8 (500 ns/マシニングロック 16MHz時)
1	0	0	周辺クロック/16(1μs/マシニングロック 16MHz時)
1	1	1	タイムベースタイマからの入力クロック (128μs/発振クロック 4MHz時)

< 注意事項 >

8ビットプリスケラ+8ビットPPGモードおよび16ビットPPGモードの場合には、ch1のPPGは、ch0からカウントクロックを受けて動作するため、PCS1ビットの指定は、無効になります。

【ビット4～2】PCM2～0(Ppg Count Mode)

ch0のダウンカウンタの動作クロックを表 13.3-11のように選択します。リセットにより本ビットは、"000"に初期化されます。本ビットは、読出しと書込みが可能です。

表 13.3-11 PCM2～0(カウントクロック選択ビット)の機能

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

【ビット1】PE11 (Ppg output Enable 11)

パルス出力外部端子PG11を表 13.3-12に示すように制御します。リセットにより本ビットは"0"に初期化されます。本ビットは、読出しと書込みが可能です。

表 13.3-12 PE11(PPG11端子出力許可ビット)

PE11	機能
0	汎用ポート端子 (パルス出力禁止)
1	PG11=パルス出力端子 (パルス出力許可)

【ビット0】PE01 (Ppg output Enable 01)

パルス出力外部端子PG01を表 13.3-13に示すように制御します。

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

表 13.3-13 PE01 (PPG01端子出力許可ビット)

PE01	機能
0	汎用ポート端子 (パルス出力禁止) [初期値]
1	PG01 = パルス出力端子 (パルス出力許可)

13.3.4 リロードレジスタ(PRLH/PRLH)

リロードレジスタ(PRLH/PRLH)は、ダウンカウンタPCNTへのリロード値を保持するレジスタです。

それぞれ以下に示す機能があります。

- ・ PRLH : H側リロード値保持
- ・ PRLH : L側リロード値保持

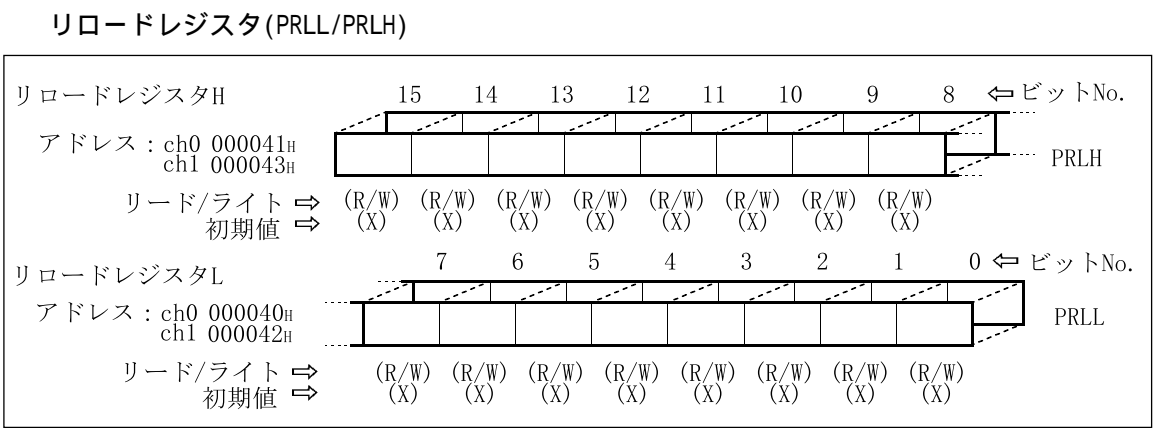


図 13.3-5 リロードレジスタ(PRLH/PRLH)

< 注意事項 >

8ビットプリスケラ+8ビットPPGモードで使用する場合には、ch0のPRLHとPRLHに異なる値を設定すると、ch1のPPG波形がサイクルごとに異なる場合がありますので、ch0のPRLHとPRLHは、同じ値に設定することを推奨します。

13.4 8/16ビットPPGの動作

8/16ビットPPGには、8ビット長のPPGユニットが2チャンネルあります。2チャンネル独立モード以外に、連結動作させることにより、8ビットプリスケラ+8ビットPPGモードと、16ビットPPG 1チャンネルモードの計3種類の動作を行うことができます。

8/16ビットPPGの動作

8ビット長のPPGユニットそれぞれは、8ビット長のリロードレジスタがL側とH側の2本あります（PRLH，PRLH）。このレジスタに書き込まれた値が、8ビットダウンカウンタ（PCNT）にL側／H側交互にリロードされてカウントクロックごとにダウンカウントされ、カウンタのボロー発生によるリロード時に、端子出力(PPG)の値を反転させます。この動作により、端子出力(PPG)はリロードレジスタ値に対応したL幅／H幅をもつパルス出力となります。

動作開始・再スタートはレジスタのビット書き込みによります。

リロード動作とパルス出力の関係を、表 13.4-1に示します。

表 13.4-1 リロード動作とパルス出力の関係

リロード動作		出力端子PG0，PG1の状態遷移
PRLH	PCNT	PG0x/1x [0 1] 立ち上がり
PRLH	PCNT	PG0x/1x [1 0] 立ち下がり

また、PPGの動作モード制御レジスタ(PPGC0)のビット4(PIE0)が"1"の場合および、PPG1動作モード制御レジスタ(PPGC1)のビット12(PIE1)が"1"の場合、それぞれのカウンタの00HからFFHへのボロー（16ビットPPGモードの場合には、0000HからFFFFHへのボロー）によって割込み要求が出力されます。

13.4.1 8/16ビットPPGの動作モード

8/16ビットPPGには、以下の3種類の動作モードがあります。

- ・2チャンネル独立モード
 - ・8ビットプリスケアラ+8ビットPPGモード
 - ・16ビットPPG 1チャンネルモード
-

8/16ビットPPGの動作モード

2チャンネル独立モード

2チャンネル独立モードは、8ビットPPGとして2チャンネル独立に動作させる動作モードです。PPG0端子は、ch0のPPG出力が接続され、PPG1端子は、ch1のPPG出力が接続されます。

8ビットプリスケアラ+8ビットPPGモード

8ビットプリスケアラ+8ビットPPGモードは、ch0を8ビットプリスケアラとして動作させ、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波形を出力できるようにする動作モードです。

PPG0端子は、ch0のプリスケアラ出力が接続され、PPG1端子は、ch1のPPG出力が接続されます。

16ビットPPG 1チャンネルモード

16ビットPPG 1チャンネルモードは、ch0とch1を連結させて、16ビットのPPGとして動作させる動作モードです。PPG0端子とPPG1端子は、両方とも16ビットPPG出力が接続されます。

13.4.2 8/16ビットPPGの出力動作

8/16ビットPPGは、ch0のPPGについては、PPG0動作モード制御レジスタ(PPGC0)のビット7(PEN0)を、ch1のPPGについては、PPG1動作モード制御レジスタ(PPGC1)のビット15(PEN1)を"1"にセットすることによって起動され、カウントを開始します。動作を開始した後は、PPGC0レジスタ内のPEN0ビットまたは、PPGC1レジスタ内のPEN1ビットに"0"を書き込むことによってカウント動作を停止し、停止した後、パルス出力はLレベルを保持します。

8/16ビットPPGの出力動作

8/16ビットPPGの出力動作時には、次の2つの点について注意してください。

- 8ビットプリスケラ+8ビットPPGモードの場合には、ch0を停止状態で、ch1を動作状態に設定しないでください。
- 16ビットPPGモードの場合には、PPGC0レジスタ内のPEN0ビットとPPGC1レジスタ内のPEN1ビットは、同時に開始・停止の制御を行ってください。

PPG動作時は、任意周波数・任意デューティ比（パルス波のHレベル期間とLレベル期間の比）のパルス波出力を連続して出力します。PPGはパルス波出力を開始し、動作停止を設定するまで停止しません。

図 13.4-1に、PPG出力動作の出力波形を示します。

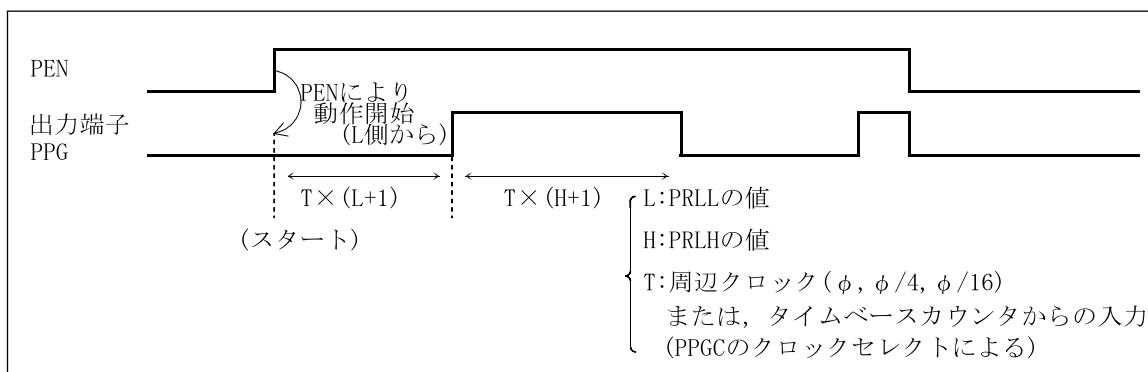


図 13.4-1 PPG出力動作の出力波形

リロード値とパルス幅の関係

リロードレジスタに書かれた値に+1した値に、カウントクロックの周期を掛けた値が、出力されるパルスの幅となります。つまり、8ビットPPG動作時のリロードレジスタ値が 00_{16} の場合および16ビットPPG動作時のリロードレジスタ値が 0000_{16} の場合は、カウントクロック1周期分のパルス幅をもつこととなりますので注意してください。また、8ビットPPG動作時のリロードレジスタ値が FF_{16} の場合、カウントクロック256周期分のパルス幅をもつことになり、16ビットPPG動作時のリロードレジスタ値が $FFFF_{16}$ の場合は、カウントクロック65536周期分のパルス幅をもつこととなりますので注意してください。

パルス幅の計算式を以下に示します。

$$\begin{aligned}
 &L: \text{PRLHの値} \\
 PI &= T \times (L+1) & H: \text{PRLHの値} \\
 Ph &= T \times (H+1) & T: \text{入力クロック周期} \\
 & & Ph: \text{ハイパルスの幅} \\
 & & PI: \text{ローパルスの幅}
 \end{aligned}$$

13.4.3 8/16ビットPPGのカウントクロックの選択

8/16ビットPPGの動作に使用するカウントクロックは、周辺クロックおよびタイムベースカウンタの入力を使用しており、6種類のカウントクロック入力を選択できます。

8/16ビットPPGのカウントクロックの選択

PPG0,1出力制御レジスタ (PPG0E) のビット4～2 (PCM2～0) でch0のクロックを、ビット7～5 (PCS2～0) でch1のクロックを選択します。クロックは、マシンのクロックの1/16～1倍のマシンのクロックとタイムベースタイマからの入力より選択できます。

ただし、8ビットプリスケラ + 8ビットPPGモードおよび16ビットPPGモードの場合には、ch1のPPGは、ch0からカウントクロックを受けて動作するため、PPG01レジスタ中のbit14 (PCS1) の値は、無効になります。

タイムベースカウンタの入力を使用した場合、トリガにより起動がかかる最初およびストップの後の最初のカウント周期がずれる可能性があります。また、本モジュール動作中にタイムベースカウンタのクリアを行うと周期がずれることがあります。

< 注意事項 >

8ビットプリスケラ + 8ビットPPGモード・ch0が動作状態・ch1が停止状態である時にch1の起動を行うと、最初のカウント周期がずれる可能性があります。

13.4.4 8/16ビットPPGのパルスの端子出力の制御

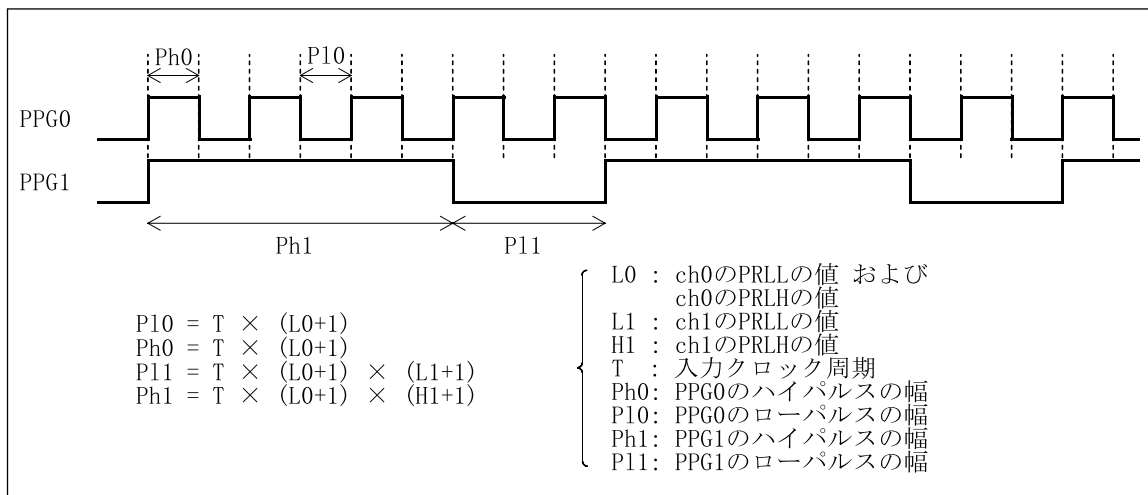
本モジュールの動作によって生成されたパルス出力は，外部端子PG00,PG01/PG10,PG11より出力させることができます。

8/16ビットPPGのパルス端子出力の制御

外部端子出力の許可は，PG00端子をPPG0動作モード制御レジスタ(PPGC0)のビット5(PE00)にて，PPG01端子をLPPG0,1制御レジスタ(PPGOE)のbit0(PE01)にて，PG10端子をPPG1動作モード制御レジスタのビット13(PE10)にて，PG11端子をPPGOEレジスタのビット1(PE11)によって行われます。本ビットが"0"の場合（初期値）は，パルス出力は外部端子より出力されずに，汎用ポートとして機能します。本ビットに"1"を設定すると，パルス出力が外部端子より出力されます。

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

8ビットプリスケラ + 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が出力され，PPG1端子は8ビットPPGの波形が出力されます。このモード時の出力波形の例を，図 13.4-2に示します。



(注意)ch0のPRLHとch0のPRLHは，同じ値を設定することを推奨します。

図 13.4-2 8 + 8PPG動作の出力波形

13.4.5 8/16ビットPPGのリロードレジスタへのライトタイミング

16ビットPPGモード以外のモードの場合には、リロードレジスタ(PRL, PRLH)への書込みはワード転送命令を使用することを推奨します。バイト転送命令2回にて書き込んだ場合、タイミングによっては、予想しないパルス幅の出力が発生する可能性があります。ご注意ください。

8/16ビットPPGのリロードレジスタへのライトタイミング

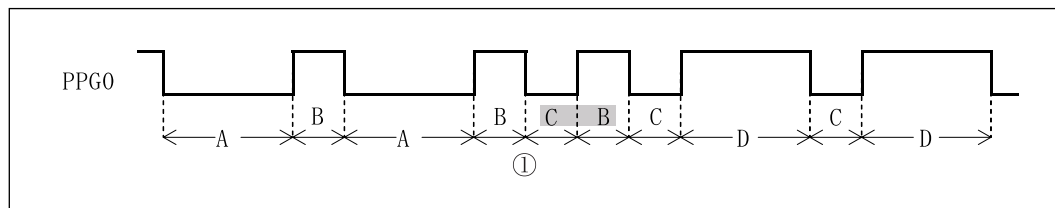


図 13.4-3 ライトタイミングタイムチャート図

図 13.4-3に示すタイムチャートにおいて、①のタイミングの前でPRLレジスタをAからCに書き換えて、②の後PRLHレジスタの値をBからDに書き換えた場合、③のタイミングでのリロードレジスタ(PRL)の値はPRL = C, PRLH = Bのため1回だけL側のカウンタ数C, H側のカウンタ数Bのパルスが発生されます。

同様に、16ビットPPGモードで使用する場合には、ch0とch1のPRLレジスタはロングワード転送で書き込むか、またはch0 ch1の順にワード転送で書き込んでください。このモードの場合には、ch0のPRLレジスタへの書き込みは、一時的にテンポラリに書き込まれ、その後ch1のPRLレジスタに書き込みを行った時点で、実際にch0のPRLレジスタへの書き込みが行われます。

なお、図 13.4-4に示すように、16ビットPPGモード以外ではch0とch1のPRLレジスタの書き込みは独立して行えます。

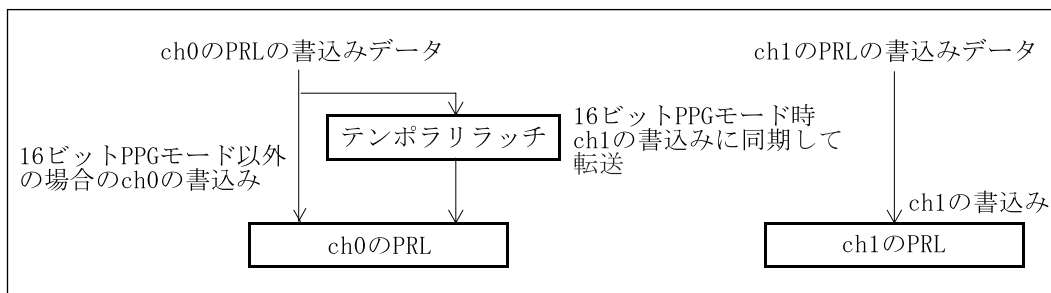


図 13.4-4 PRL書き込み部分のブロックダイアグラム

13.4.6 8/16ビットPPGの割込み

8/16ビットPPGの割込みは、リロード値がカウントアウトし、ボローが発生したときにアクティブになります。

8/16ビットPPGの割込み

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

13.4.7 8 / 16ビットPPGの各ハードウェアの初期値

8 / 16ビットPPGの各ハードウェアのリセット時の初期値を示します。

8 / 16ビットPPGの各ハードウェアの初期値

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

レジスタ

PPGC0 0X000001_B

PPGC1 00000001_B

PPG0E XXXXXX00_B

パルス出力

PPG0 "L"

PPG1 "L"

PE00 PPG0出力禁止

PE10 PPG1出力禁止

割込み要求

IRQ0 "L"

IRQ1 "L"

なお , 上記以外のハードウェアは初期化されません。

第14章 8/16ビットアップダウンカウンタ / タイマ

この章では、8/16ビットアップダウンカウンタ / タイマの機能と動作について説明します。

- 14.1 8/16ビットアップダウンカウンタ / タイマの概要
- 14.2 8/16ビットアップダウンカウンタ / タイマのブロックダイアグラム
- 14.3 8/16ビットアップダウンカウンタ / タイマのレジスタ一覧
- 14.4 8/16ビットアップダウンカウンタ / タイマのカウントモード選択
- 14.5 8/16ビットアップダウンカウンタ / タイマのリロード機能とコンペア機能
- 14.6 8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能の同時起動
- 14.7 8/16ビットアップダウンカウンタ / タイマのUDCRレジスタへのデータの書込み

14.1 8/16ビットアップダウンカウンタ / タイマの概要

8/16ビットアップダウンカウンタ / タイマは、イベント入力端子6本、8ビットアップ / ダウンカウンタ×2本、8ビットリロード / コンペアレジスタ×2本と、それらの制御回路から構成されています。

8/16ビットアップダウンカウンタ / タイマの機能

8/16ビットアップダウンカウンタ / タイマの主な機能を以下に示します。

カウント可能範囲

8ビットカウントレジスタにより、0～256の範囲でカウント可能
(16bit×1動作モードでは0～65535の範囲でカウント可能)。

カウントモード

カウントクロックの選択による4種類のカウントモードが選択可能です。

- タイマモード
- アップ / ダウンカウンタモード
- 位相差カウントモード (2通倍)
- 位相差カウントモード (8通倍)

カウントクロック (16MHz動作時)

タイマモード時に、内部クロック2種類を選択可能です。

- 125ns (8MHz:2分周)
- 1.0μs (2MHz:8分周)

検出エッジの選択

アップ / ダウンカウンタモード時に、外部端子入力信号で4種類の検出エッジが選択可能です。

- エッジ検出禁止
- 立下りエッジ検出
- 立上りエッジ検出
- 立下り / 立上りエッジ両エッジ検出

位相差カウントモード

位相差カウントモードは、モータなどのエンコーダのカウントに適し、エンコーダのA相、B相、Z相出力をそれぞれ入力することで、高精度で回転角度、回転数などのカウントを容易に行うことが可能です。

ZIN端子

ZIN端子は、2種類の機能が選択可能です。

- カウンタクリア機能
- ゲート機能

コンペア / リロード機能

8/16ビットアップダウンカウンタ / タイマでは、コンペア機能およびリロード機能を有し、それぞれの機能のみ、また組み合わせても使用可能です。両機能を起動することで、任意幅でのアップダウンカウントを行うことが可能です。

- コンペア機能（コンペア時割込み出力）
- コンペア機能（コンペア時割込み出力およびカウンタクリア）
- リロード機能（アンダフロー時割込み出力およびリロード）
- コンペア / リロード機能（コンペア時割込み出力、カウンタクリア、アンダフロー時割込み出力およびリロード）
- コンペア / リロード禁止

割込み制御

コンペア時、リロード（アンダフロー）時およびオーバフロー時に、割込みの発生をそれぞれ個別に制御可能です。

カウント方向の識別

カウント方向フラグにより、直前のカウント方向が識別可能です。

カウント方向と割込み

カウント方向が変わった場合に割込みを発生します。

14.2 8/16ビットアップダウンカウンタ / タイマのブロック ダイアグラム

8/16ビットアップダウンカウンタ / タイマのブロックダイアグラムを示します。

8/16ビットアップダウンカウンタ / タイマのブロックダイヤグラム

図 14.2-1に、8/16ビットアップダウンカウンタ / タイマ (ch.0) のブロックダイヤグラムを示します。

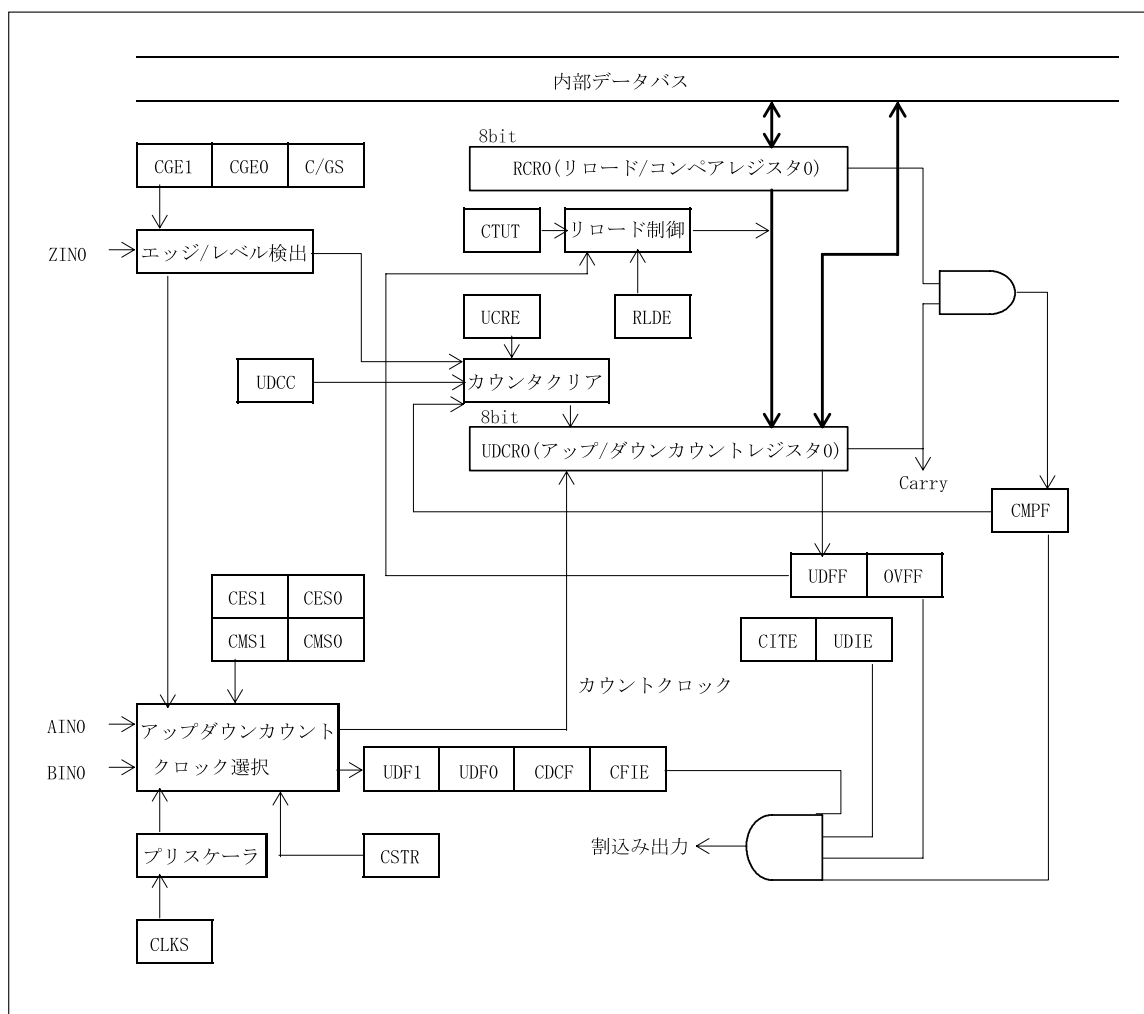


図 14.2-1 8/16ビットアップダウンカウンタ / タイマ(Ch.0)のブロックダイアグラム

図 14.2-2に、8/16ビットアップダウンカウンタ / タイマ(Ch.1)のブロックダイアグラムを示します。

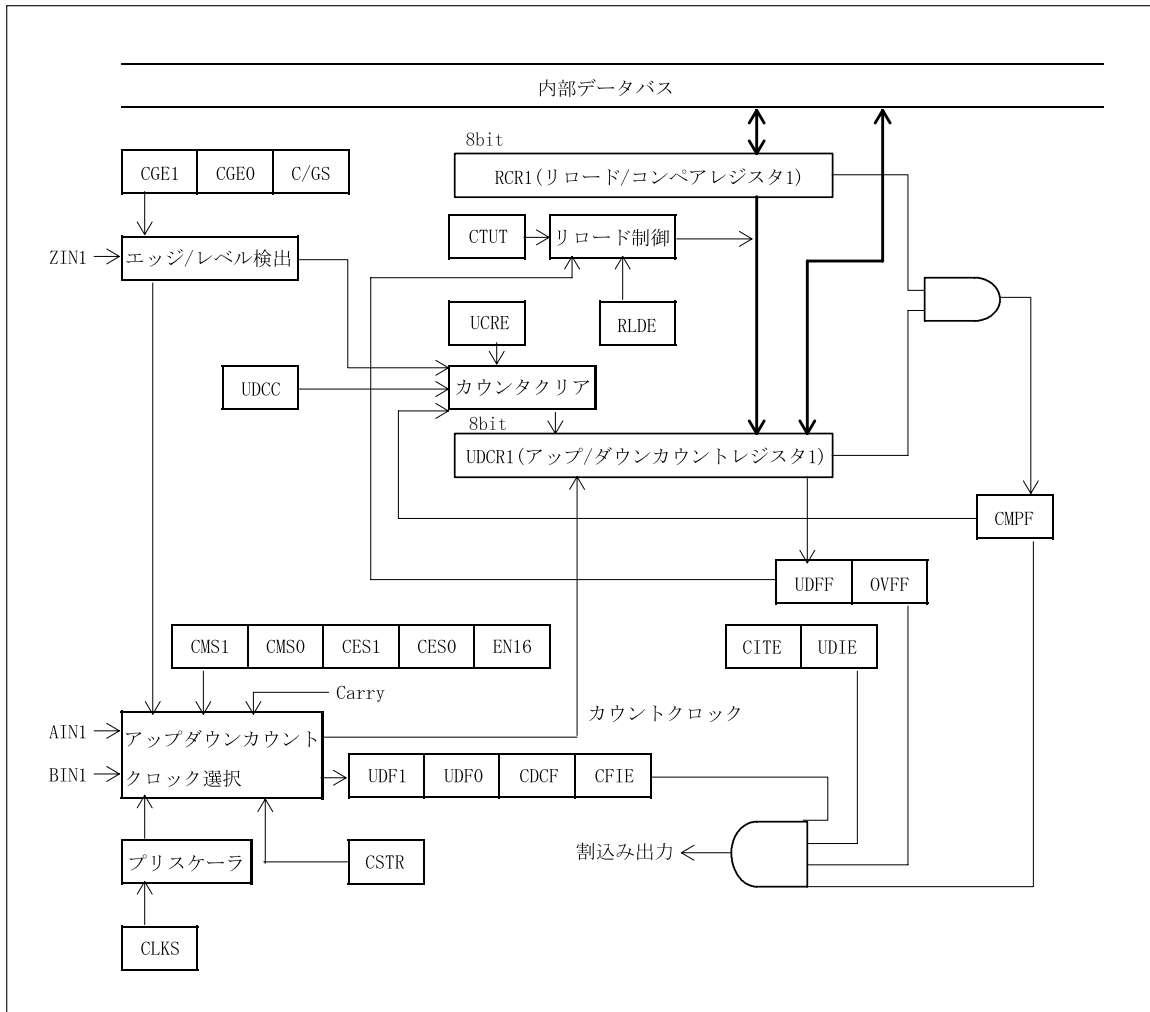


図 14.2-2 8/16ビットアップダウンカウンタ / タイマ(Ch.1)のブロックダイアグラム

14.3 8/16ビットアップダウンカウンタ / タイマのレジスタ一覧

8/16ビットアップダウンカウンタ / タイマには、図 14.3-1に示すレジスタがあります。

8/16ビットアップダウンカウンタ / タイマのレジスタ一覧

15								8 7 0							
UDCR1								UDCR0							
RCR1								RCR0							
予約領域								CSR0							
CCRHO								CCRL0							
予約領域								CSR1							
CCRHI								CCRL1							
8bit								8bit							

アップダウンカウンタレジスタ1								15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス:000071H								D17	D16	D15	D14	D13	D12	D11	D10	UDCR1
リード/ライト								(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
アップダウンカウンタレジスタ0								7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス:000070H								D07	D06	D05	D04	D03	D02	D01	D00	UDCR0
リード/ライト								(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
リロード/コンペアレジスタ1								15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス:000073H								D17	D16	D15	D14	D13	D12	D11	D10	RCR1
リード/ライト								(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
リロード/コンペアレジスタ0								7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス:000072H								D07	D06	D05	D04	D03	D02	D01	D00	RCR0
リード/ライト								(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
カウンタステータスレジスタ0/1								7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス:000074H アドレス:000078H								CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0	CSR0, 1
リード/ライト								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R)	(R)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
カウンタコントロールレジスタ0 上位								15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス:000077H								M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	CCRHO
リード/ライト								(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値								(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
カウンタコントロールレジスタ1 上位								15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス:00007BH								—	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	CCRHI
リード/ライト								(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値								(—)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
カウンタコントロールレジスタ0/1 下位								7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス:000076H アドレス:00007AH								—	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	CCRL0, 1
リード/ライト								(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値								(—)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 14.3-1 8/16ビットアップダウンカウンタ / タイマのレジスタ構成

14.3.1 アップダウンカウントレジスタ0,1(UDCR0,1)

アップダウンカウントレジスタ0,1(UDCR0,1)は、8ビットのカウントレジスタです。内部プリスケアラ、またはAIN端子、BIN端子の入力によってアップ / ダウンカウント動作を行います。

アップダウンカウントレジスタ0,1(UDCR0,1)

アップダウンカウントレジスタ1								
	15	14	13	12	11	10	9	8 ⇐ ビットNo.
アドレス:000071 _H	D17	D16	D15	D14	D13	D12	D11	D10 UDCR1
リード/ライト ⇨	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
初期値 ⇨	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
アップダウンカウントレジスタ0								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:000070 _H	D07	D06	D05	D04	D03	D02	D01	D00 UDCR0
リード/ライト ⇨	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
初期値 ⇨	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 14.3-2 アップダウンカウントレジスタch.0/1(UDCR0/1)

アップダウンカウントレジスタ(UDCR)は、8ビットのカウントレジスタです。内部プリスケアラ、またはAIN端子、BIN端子の入力によってアップ / ダウンカウント動作を行います。また16ビットのカウントモードでは、16ビットのカウントレジスタとして動作します。この場合上位8ビット側の制御レジスタの設定値は動作上無効になります。

UDCRレジスタは、直接書込み動作を行うことができません。本レジスタに書込みを行う場合は、リロードコンペアレジスタ(RCR)を介して行う必要があります。本レジスタに書き込みたい値をまずRCRレジスタに書き込み、その後カウントコントロールレジスタ下位(CCRL)内のCTUTビットに"1"を書き込むことでRCRレジスタから本レジスタに転送されます(ソフトウェアによるリロード)。

本レジスタは、16ビットモード起動時にはワードアクセスで読み出してください。

14.3.2 リロード / コンペアレジスタ0,1(RCR0,1)

リロード / コンペアレジスタ0,1(RCR0,1)は,8リロード / コンペアレジスタです。本レジスタにより,リロード値およびコンペア値を設定します。

リロード / コンペアレジスタ0,1(RCR0,1)

リロード/コンペアレジスタ1								
	15	14	13	12	11	10	9	8 ⇐ ビットNo.
アドレス:000073 _H	D17	D16	D15	D14	D13	D12	D11	D10 RCR1
リード/ライト	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
初期値	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
リロード/コンペアレジスタ0								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:000072 _H	D07	D06	D05	D04	D03	D02	D01	D00 RCR0
リード/ライト	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
初期値	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 14.3-3 リロード / コンペアレジスタ0,1(RCR0,1)

本レジスタは,8ビットのリロード / コンペアレジスタです。本レジスタにより,リロード値およびコンペア値を設定します。リロード値とコンペア値は同一であり,リロード機能およびコンペア機能を起動することで00_H ~ RCRレジスタの値(16ビット動作モード: 0000_H ~ RCRレジスタ値)の間でアップ / ダウンカウントが可能になります。

本レジスタは書込みのみ可能で,読出しはできません。カウント停止中にCCR0/1レジスタ内のCTUTビットに"1"を書き込むことで,本レジスタの値をアップダウンカウントレジスタ(UDCR)に転送する(ソフトウェアによるリロード)ことが可能です。

本レジスタは,ワードアクセスで書き込んでください。

14.3.3 カウンタステータスレジスタ0,1(CSR0,1)

カウンタステータスレジスタ0,1(CSR0,1)は、8bitモード時のch.0,1それぞれのイベントフラグ / 割込み動作制御の設定を行うレジスタです。

カウンタステータスレジスタ0,1(CSR0,1)

カウンタステータスレジスタ0,1(CSR0,1)のビット構成を、図 14.3-4に示します。

カウンタステータスレジスタ0/1								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:000074 _H	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0
アドレス:000078 _H								
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R)	(R)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 14.3-4 カウンタステータスレジスタch.0/1(CSR0/1)

【ビット7】CSTR

アップダウンカウンタレジスタ(UDCR)は、カウント動作の起動 / 停止を制御するビットです。

表 14.3-1 CSTR(カウント起動ビット)

CSTR	機能
0	カウント動作停止 (初期値)
1	カウント動作起動

【ビット6】CITE

CMPFビットがセットされた (コンペアが発生した) 場合に、CPUへの割込み出力を行うかどうかの許可 / 禁止を制御するビットです。

表 14.3-2 CITE (コンペア割込み出力制御ビット)

CITE	機能
0	コンペア割込み出力の禁止 (初期値)
1	コンペア割込み出力の許可

【ビット5】UDIE

OVFFビットおよびUDFFビットがセットされた (オーバフロー / アンダフローが発生した) 場合に、CPUへの割込み出力を行うかの許可 / 禁止を制御するビットです。

表 14.3-3 UDIE (オーバフロー / アンダフロー割込み出力制御ビット)

CITE	機能
0	オーバフロー / アンダフロー割込み出力の禁止 (初期値)
1	オーバフロー / アンダフロー割込み出力の許可

【ビット4】CMPF：コンペア検出フラグ

UDCRレジスタの値とリロード/コンペアレジスタ(RCR)の値の比較結果が等しくなったことを示すフラグです。カウント起動時にUDCRレジスタとRCRレジスタの値が一致していたら、起動と同時にセットされますので注意してください。"0"書込みのみ可能で、"1"書込みはできません。

表 14.3-4 CMPF (コンペア検出フラグ)

CMPF	機能
0	比較結果が一致していない (初期値)
1	比較結果が一致した

【ビット3】OVFF：オーバフロー検出フラグ

オーバフローの発生を示すフラグです。"0"書込みのみ可能で、"1"書込みはできません。

表 14.3-5 OVFF (オーバフロー検出フラグ)

OVFF	機能
0	オーバフローなし (初期値)
1	オーバフローあり

【ビット2】UDFF

アンダフローの発生を示すフラグです。"0"書込みのみ可能で、"1"書込みはできません。

表 14.3-6 UDFF (アンダフロー検出フラグ)

UDFF	機能
0	アンダフローなし (初期値)
1	アンダフローあり

【ビット1, 0】UDF1,UDF0

直前のカウント動作 (アップ / ダウン) を示すビットです。読出しのみ可能で、書込みはできません。

表 14.3-7 UDF1,UDF0 (アップダウンフラグ)

UDF1	UDF0	検出エッジ
0	0	入力なし (初期値)
0	1	ダウンカウント
1	0	アップカウント
1	1	アップ / ダウン同時発生

14.3.4 カウンタコントロールレジスタ0上位(CCRH0)

カウンタコントロールレジスタ0上位は、8bitモードch.0の動作制御の設定、および16ビットモードへの切換えを設定するレジスタです。カウンタコントロールレジスタ0下位(CCRLO)とあわせて動作設定を行います。

カウンタコントロールレジスタ0上位(CCRH0)

カウンタコントロールレジスタ0上位(CCRH0)のビット構成を図 14.3-5に示します。

カウンタコントロールレジスタ0上位								←ビットNo.	
	15	14	13	12	11	10	9	8	
アドレス:000077 _H	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	CCRHO
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 14.3-5 カウンタコントロールレジスタ0上位(CCRH0)

【ビット15】M16E

8bit×2ch/16bit×1ch動作モードの選択（切換え）ビットです。

表 14.3-8 M16E(16ビットモード許可設定ビット)

M16E	16ビットモード許可設定
0	8ビット×2ch動作モード（初期値）
1	16ビット×1ch動作モード

【ビット14】CDCF

カウント方向が換わった場合にセットされるフラグです。カウント起動中にカウント方向がアップ ダウン、またはダウン アップに換わった場合にセットされます。"0"書込みのみ可能で、"1"書込みはできません。

表 14.3-9 CDCF（カウント方向転換フラグ）

CDCF	方向転換検出
0	方向転換なし（初期値）
1	方向転換あり（1回以上）

【ビット13】CFIE

CDCFビットがセットされた場合、CPUに対しての割込み出力を制御するビットです。カウント起動中に一度でもカウント方向が換わった場合に割込みを発生します。

表 14.3-10 CFIE（カウント方向転換割込みイネーブルビット）

CFIE	方向転換割込み出力
0	方向転換割込み出力禁止（初期値）
1	方向転換割込み出力許可

【ビット12】CLKS

タイマモード選択時に、内蔵プリスケアラの周波数を選択するビットです。
タイマモードでのみ有効で、この場合はダウンカウントのみとなります。

表 14.3-11 CLKS (内蔵プリスケアラ選択ビット)

CLKS	選択内部クロック
0	2マシンサイクル (初期値)
1	8マシンサイクル

【ビット11, 10】CMS1, CMS0

カウントモードを選択するビットです。

表 14.3-12 CMS1, CMS0 (カウントモード選択ビット)

CMS1	CMS0	カウントモード
0	0	タイマモード〔ダウンカウント〕 (初期値)
0	1	アップ / ダウンカウントモード
1	0	位相差カウントモード (2 逡倍)
1	1	位相差カウントモード (4 逡倍)

【ビット9, 8】CES1, CES0

アップ / ダウンカウントモード時に、外部端子AINおよびBINの検出エッジを選択するビットです。アップ / ダウンカウントモード以外では、この設定は無効です。

表 14.3-13 CES1, CES0 (カウントクロックエッジ選択ビット)

CES1	CES0	選択エッジ
0	0	エッジ検出禁止 (初期値)
0	1	立下りエッジ検出
1	0	立上りエッジ検出
1	1	立上り / 立下り両エッジ検出

14.3.5 カウンタコントロールレジスタ1上位(CCRH1)

カウンタコントロールレジスタ1上位(CCRH1)は、8ビットモードch.1の動作制御の設定を行うレジスタです。カウンタコントロールレジスタ1下位(CCRL1)とあわせて動作設定を行います。

カウンタコントロールレジスタ1上位(CCRH1)

カウンタコントロールレジスタ1上位(CCRH1)のビット構成を図 14.3-6に示します。

カウンタコントロールレジスタ1上位									
	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス:00007BH	—	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0	CCRH1
リード/ライト ⇄	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇄	(—)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 14.3-6 カウンタコントロールレジスタ上位 ch.1(CCRH1)

【ビット15】未使用ビット

【ビット14】CDCF

カウント方向が換わった場合にセットされるフラグです。カウント起動中にカウント方向がアップ ダウン、またはダウン アップに換わった場合にセットされます。"0"書込みのみ可能で、"1"書込みはできません。

表 14.3-14 CDCF(カウント方向転換フラグ)

CDCF	方向転換検出
0	方向転換は行われていない(初期値)
1	方向転換が1回以上行われた

【ビット13】CFIE

CDCFビットがセットされた場合の、CPUに対しての割込み出力を制御するビットです。カウント起動中に一度でもカウント方向が換わった場合に割込みを発生します。

表 14.3-15 CFIE(カウント方向転換割込みイネーブルビット)

CFIE	方向転換割込み出力
0	方向転換割込み出力禁止(初期値)
1	方向転換割込み出力許可

【ビット12】CLKS

タイマモード選択時に、内蔵プリスケアラの周波数を選択するビットです。
タイマモードでのみ有効で、この場合はダウンカウントのみとなります。

表 14.3-16 CLKS(内蔵プリスケアラ選択ビット)

CLKS	選択内部クロック
0	2マシンサイクル (初期値)
1	8マシンサイクル

【ビット11, 10】CMS1, CMS0

カウントモードを選択するビットです。

表 14.3-17 CMS1, CMS0 (カウントモード選択ビット)

CMS1	CMS0	カウントモード
0	0	タイマモード〔ダウンカウント〕 (初期値)
0	1	アップ / ダウンカウントモード
1	0	位相差カウントモード2通倍
1	1	位相差カウントモード4通倍

【ビット9, 8】CES1, CES0

アップ / ダウンカウントモード時に、外部端子AINおよびBINの検出エッジを選択するビットです。アップ / ダウンカウントモード以外では、この設定は無効です。

表 14.3-18 CES1, CES0 (カウントクロックエッジ選択ビット)

CES1	CES0	選択エッジ
0	0	エッジ検出禁止 (初期値)
0	1	立下りエッジ検出
1	0	立上りエッジ検出
1	1	立上り / 立下り両エッジ検出

14.3.6 カウンタコントロールレジスタ0,1下位 (CCRL0,1)

カウンタコントロールレジスタ0,1下位(CCRL0,1)は、8ビットモードch.0,1の動作制御の設定を行うレジスタです。それぞれカウンタコントロールレジスタ0/1上位(CCRH0,1)とあわせて動作設定を行います。

カウンタコントロールレジスタ0,1下位 (CCRL0,1)

カウンタコントロールレジスタ0,1下位(CCRL0,1)のビット構成を図 14.3-7に示します。

カウンタコントロールレジスタ0/1下位									↔ビットNo.
	7	6	5	4	3	2	1	0	
アドレス:000076 _H アドレス:00007A _H	—	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0	CCRL0, 1
リード/ライト 初期値	⇒ ⇒	(-) (-)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

図 14.3-7 カウンタコントロールレジスタ下位 ch.0/1 (CCRL0/1)

【ビット7】未使用ビット

【ビット6】CTUT (カウンタライトビット)

RCRからUDCRへのデータ転送を行います。このビットに"1"を書き込むとロード/コンペアレジスタ(RCR)からアップダウンカウンタレジスタ(UDCR)にデータが転送されます。"0"書込みは無効であり、読出し値は常に"0"です。カウンタ起動中(カウンタステータスレジスタ(CSR0)内のCSTRビットが"1"の場合)に本ビットに"1"を書き込まないでください。

【ビット5】UCRE

コンペアによるUDCRレジスタのクリアを制御するビットです。コンペア発生によるクリア以外のUDCRレジスタクリア機能(ZIN端子によるものなど)には影響しません。

表 14.3-19 UCRE(UDCRクリアイネーブルビット)

UCRE	コンペアによるカウンタのクリア
0	カウンタクリア禁止 (初期値)
1	カウンタクリア許可

【ビット4】RLDE：リロードイネーブルビット

リロード機能の起動を制御するビットです。リロード機能起動時にUDCRレジスタがアンダフローを発生した場合にRCRレジスタの値をUDCRレジスタに転送します。

表 14.3-20 RLDE：リロードイネーブルビット

RLDE	リロード機能
0	リロード機能禁止（初期値）
1	リロード機能許可

【ビット3】UDCC：UDCRクリアビット

UDCRレジスタをクリアするビットです。このビットに"0"を書き込むとUDCRレジスタが0000_Hにクリアされます。"1"書込みは無効であり、読出し値は常に"1"です。

【ビット2】CGSC

外部端子ZINの機能を選択するビットです。

表 14.3-21 CGSC(カウンタクリア/ゲート選択ビット)

CGSC	ZINの機能
0	カウンタクリア機能（初期値）
1	ゲート機能

【ビット1～0】CGE1,CGE0

外部端子ZIN の検出エッジ/レベルを選択するビットです。

表 14.3-22 CGE1,CGE0（カウンタクリア/ゲートエッジ選択ビット）

CGE1	CGE0	カウンタクリア機能選択時	ゲート機能選択時
0	0	エッジ検出禁止（初期値）	レベル検出禁止 （カウントディセーブル）
0	1	立下りエッジ検出	LOWレベル
1	0	立上りエッジ検出	HIGHレベル
1	1	設定禁止	設定禁止

14.4 8/16ビットアップダウンカウンタ / タイマのカウントモード選択

8/16ビットアップダウンカウンタ / タイマは、4種類のカウントモードが選択できます。これらのカウントモードの選択は、カウンタコントロールレジスタ(CCRH)内のCMS1, CMS0ビットで制御します。

8/16ビットアップダウンカウンタ / タイマのカウントモード選択

表 14.4-1に、8/16ビットアップダウンカウンタ / タイマの4種類のカウントモードを示します。

表 14.4-1 8/16ビットアップダウンカウンタ / タイマのカウントモード選択

CMS1, CMS0	カウントモード
00 _B	タイマモード〔ダウンカウント〕
01 _B	アップ / ダウンカウントモード
10 _B	位相差カウントモード(2通倍)
11 _B	位相差カウントモード(4通倍)

タイマモード〔ダウンカウント〕

タイマモードでは、内部プリスケアラの出力をダウンカウントします。内蔵プリスケアラについては、CCRHレジスタ内のCLKSビットによって2マシンサイクル / 8マシンサイクルの選択が可能です。

アップ / ダウンカウントモード

アップダウンカウントモードでは、外部端子AINおよびBINの入力をカウントすることでアップ / ダウンカウントを行います。AIN端子の入力はアップカウントを、BIN端子の入力はダウンカウントをそれぞれ制御します。

AIN端子、BIN端子の入力はエッジ検出であり、CCRHレジスタ内のCES1, CES0ビットによって検出エッジの選択が可能です。表 14.4-2に検出エッジを示します。

表 14.4-2 8/16ビットアップダウンカウンタ / タイマの検出エッジ選択

CES1, CES0	選択エッジ
00 _B	エッジ検出禁止
01 _B	立下りエッジ検出
10 _B	立上りエッジ検出
11 _B	立上り / 立下り両エッジ検出

位相差カウントモード（2逓倍 / 4逓倍）

位相差カウントモードでは、エンコーダの出力信号A相、B相の位相差をカウントするため、AIN端子の入力エッジ検出時にBIN端子の入力レベルを、BIN端子の入力エッジ検出時にAIN端子の入力レベルを検出し、カウントを行います。

2逓倍 / 4逓倍モードでは、AIN端子入力とBIN端子入力の位相差について、AIN端子の方が早い場合にアップカウントを、BIN端子の方が早い場合にダウンカウントを行います。

2逓倍モードでは、BIN端子の立上り / 立下り両方のエッジのタイミングでAIN端子の値を検出することで次のカウントを行います。

- BIN端子の立上りエッジで検出したAIN端子の値が"H"の場合、アップカウント
- BIN端子の立上りエッジで検出したAIN端子の値が"L"の場合、ダウンカウント
- BIN端子の立下りエッジで検出したAIN端子の値が"H"の場合、ダウンカウント
- BIN端子の立下りエッジで検出したAIN端子の値が"L"の場合、アップカウント

図 14.4-1に、位相差カウントモード（2逓倍）の動作概略を示します。

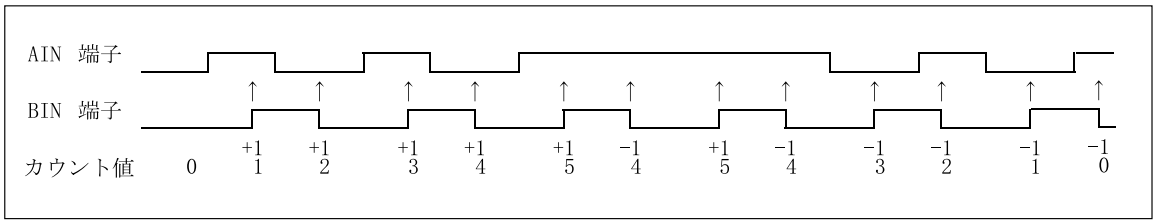


図 14.4-1 位相差カウントモード（2逓倍）の動作概略

4逓倍モードでは、BIN端子の立上り / 立下り両方のエッジのタイミングでAIN端子の値を検出し、またAIN端子の立上り / 立下り両方のエッジのタイミングでBIN端子の値を検出することで次のカウントを行います。

- BIN端子の立上りエッジで検出したAIN端子の値が"H"の場合、アップカウント
- BIN端子の立上りエッジで検出したAIN端子の値が"L"の場合、ダウンカウント
- BIN端子の立下りエッジで検出したAIN端子の値が"H"の場合、ダウンカウント
- BIN端子の立下りエッジで検出したAIN端子の値が"L"の場合、アップカウント
- AIN端子の立上りエッジで検出したBIN端子の値が"H"の場合、ダウンカウント
- AIN端子の立上りエッジで検出したBIN端子の値が"L"の場合、アップカウント
- AIN端子の立下りエッジで検出したBIN端子の値が"H"の場合、アップカウント
- AIN端子の立下りエッジで検出したBIN端子の値が"L"の場合、ダウンカウント

図 14.4-2に、位相差カウントモード（4逓倍）の動作概略を示します。

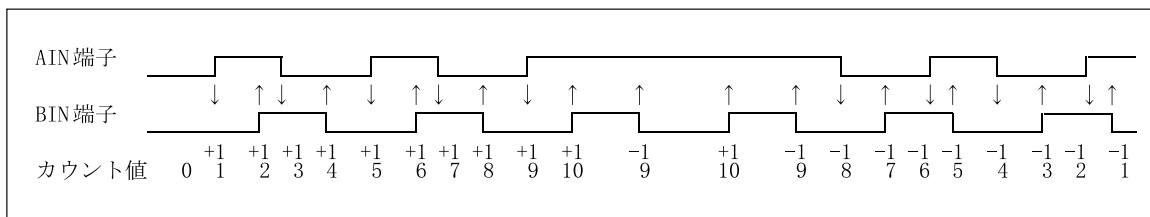


図 14.4-2 位相差カウントモード（4逓倍）動作概略

エンコーダ出力のカウントの際にはA相をAIN端子に、B相をBIN端子に、Z相をZIN端子に入力することで、高精度で回転角度や回転数のカウント、回転方向の検出などが可能です。

< 注意事項 >

本カウントモード選択時は、カウンタコントロールレジスタ(CCRH)内のCES1, CES0ビット、カウンタコントロールレジスタ下位(CCRL)内のCGE1, CGE0ビットによる検出エッジの選択は無効です。

14.5 8/16ビットアップダウンカウンタ / タイマのリロード機能とコンペア機能

8/16ビットアップダウンカウンタでは、リロード機能およびコンペア機能を内蔵しています。この二つの機能は組み合わせて処理を行うことが可能です。

8/16ビットアップダウンカウンタ / タイマのリロード機能とコンペア機能

8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能選択として、表 14.5-1に示すような設定が挙げられます。

表 14.5-1 8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能選択

RLDE,UCRE	リロード / コンペア機能
00 _B	コンペア / リロード禁止 (初期値)
01 _B	コンペア許可
10 _B	リロード許可
11 _B	コンペア / リロード許可

リロード機能

リロード機能起動時は、アンダフロー発生次のダウンカウントクロックのタイミングでリロード / コンペアレジスタ(RCR)の値をアップダウンカウントレジスタ(UDCR)に転送します。この場合、カウンタステータスレジスタ(CSR)内のUDFFビットがセットされるとともに割込み要求が発生します。

< 注意事項 >

ダウンカウントを行わないモードでは、本機能の起動は無効となります。

図 14.5-1に、リロード機能の動作概略を示します。

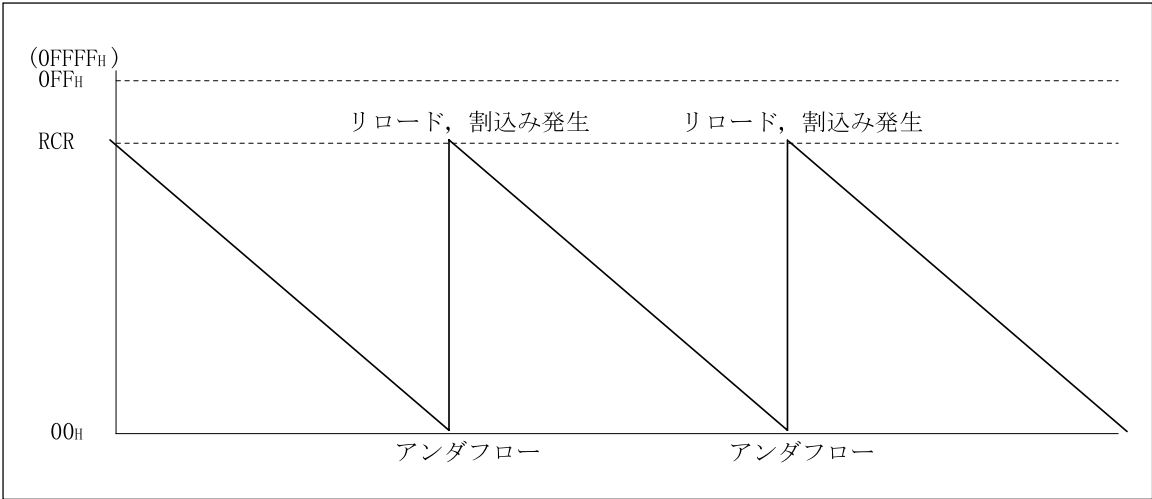


図 14.5-1 リロード機能の動作概略

コンペア機能

コンペア機能は、タイマモード以外のすべてのモードで使用可能です。コンペア機能起動時は、RCRレジスタとUDCRレジスタの値が一致した場合に、カウンタステータスレジスタ(SR)内のCMPFビットがセットされるとともに割込み要求が発生します。またコンペアクリア機能起動時は、その次のアップカウントクロックのタイミングでUDCRレジスタをクリアします。

< 注意事項 >

アップカウントを行わないモードでは、本機能の起動は無効となります。

図 14.5-2にコンペア機能の動作概略を示します。

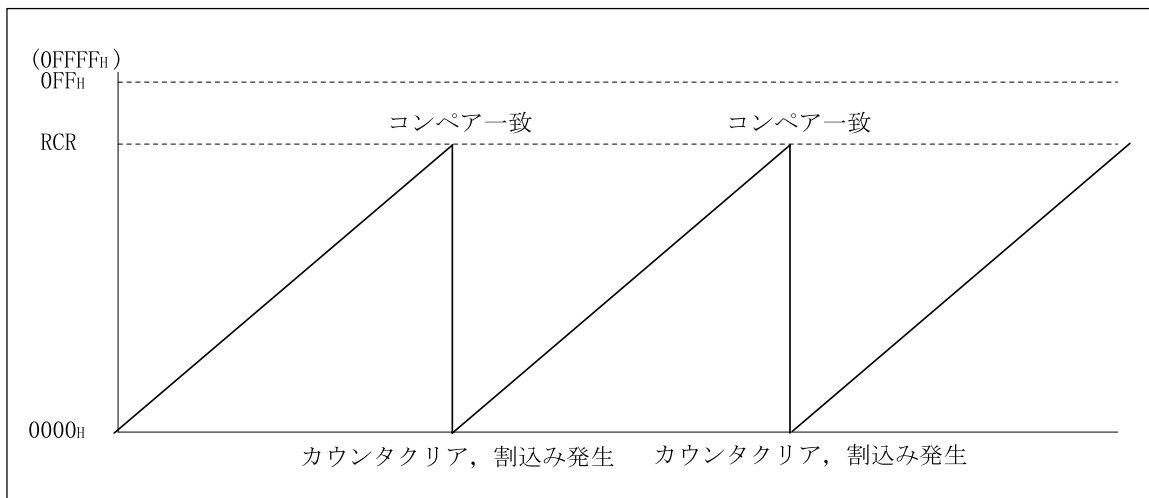


図 14.5-2 コンペア機能の動作概略

14.6 8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能の同時起動

リロード / コンペア機能の同時起動時は、任意幅でのアップ / ダウンカウントが可能です。

8/16ビットアップダウンカウンタ / タイマのリロード / コンペア機能の同時起動

リロード機能により、アンダフロー時に起動してリロード / コンペアレジスタ (RCR) の値をアップダウンカウンタレジスタ (UDCR) に転送します。また、コンペア機能により、RCRレジスタとUDCRレジスタの値が一致した場合に起動し、UDCRレジスタをクリアします。この両機能を利用して、00_H ~ RCR_Hの値の間でアップ / ダウンカウントを行います。

図 14.6-1に、リロード / コンペア機能同時起動時の動作概略を示します。

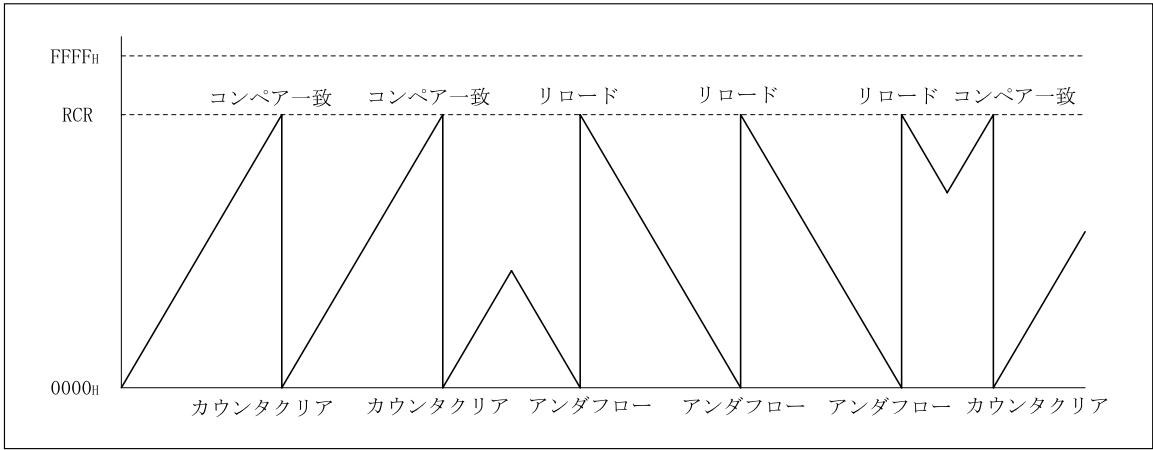


図 14.6-1 リロード / コンペア機能同時起動時の動作概略

コンペア一致時またはリロード（アンダフロー）時にCPUに割り込みを発生させることができます。また、これらの割り込み出力のイネーブルは個別に制御可能です。UDCRレジスタに対しリロードおよびクリアを行う場合のタイミングについて、カウント起動中と停止中では異なります。

カウント動作中のリロード / クリアタイミング

カウント動作中にリロードまたはクリアのイベントが発生した場合は、すべてカウントクロックに同期して行われます（図 14.6-2は80hをリロードした場合）。

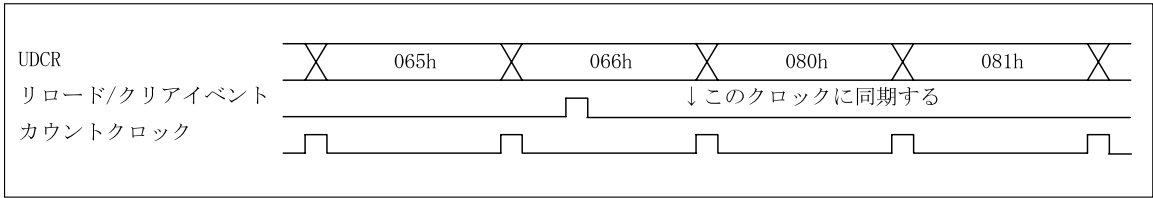


図 14.6-2 カウント動作中のリロード / クリアタイミング

リロードクリア後のカウントディセーブル時のカウント値

カウント動作中にリロードおよびクリアのイベントが発生した場合で、カウントクロック同期待ち（同期させるためのカウント入力を待っている状態）のままカウントを停止させたときは、停止した時点でリロードおよびクリアが行われます（図 14.6-3は、80_Hをリロードした場合）。

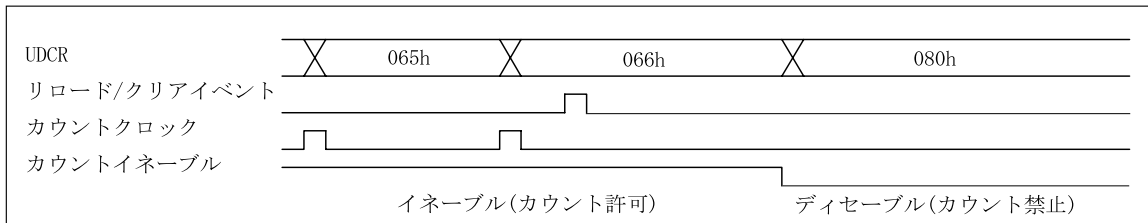


図 14.6-3 リロードクリア後のカウントディセーブル時のカウント値

カウント停止中のリロード/クリアタイミング

カウント停止中にリロードおよびクリアのイベントが発生した場合は、イベント発生時点で行われます（図 14.6-4は80_Hをリロードした場合）。

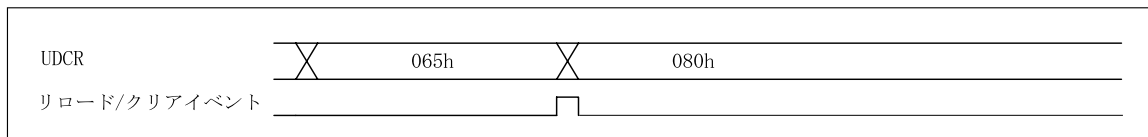


図 14.6-4 カウント停止中のリロード/クリアタイミング

コンペアによるクリアについては、UDCRレジスタとRCRレジスタの値が一致し、さらにアップカウントが行われた時にクリアが行われます。UDCRレジスタとRCRレジスタの値が一致した場合においても、その後ダウンカウントやカウント停止になった場合はクリアは行われません。クリア/リロードのタイミングについて、クリアについてはリセット入力以外のすべてのイベントで、またリロードもすべてのイベントにおいて上記タイミングにしたがいます。

クリアイベントとリロードイベントが同時期に発生した場合はクリアイベントが優先となります。

14.7 8/16ビットアップダウンカウンタ / タイマのUDCRレジスタへのデータの書込み

アップダウンカウンタレジスタ(UDCR)へデータバスから直接データを書き込むことはできません。UDCRレジスタに任意のデータを書き込む場合は、以下のような手順で書き込む必要があります。

- 1) UDCRレジスタに書き込むデータを、まずリロード/コンペアレジスタ(RCR)に書き込みます(RCRレジスタのデータは失われるので注意)。
- 2) カウンタコントロールレジスタ下位(CCRL)内のCTUTビットに"1"を書き込むことにより、RCRレジスタからUDCRレジスタにデータが転送されます。

以上の動作は、カウント停止中カウンタステータスレジスタ(CSR)内のCSTRビットが"0"の場合)に行ってください。

UDCRレジスタへのデータの書込み

カウンタのクリアについては、上記以外に以下のような方法があります。

- リセット入力によるクリア (初期化)
- ZIN端子からのエッジ入力によるクリア
- CCRLレジスタのUDCCビットに"0"を書き込むことによるクリア
- コンペア機能によるクリア

これらの書込みは、カウント起動 / 停止に関わらず行うことができます。

カウントクリア / ゲート機能

表 14.7-1に示すように、ZIN端子はCCRLレジスタ内のCGSCビットによって、カウントクリア機能またはゲート機能のいずれかとして選択して使用可能です。

表 14.7-1 ZIN端子機能の選択

CGSC	ZIN端子の機能
0 _B	カウントクリア機能
1 _B	ゲート機能

カウントクリア機能起動時は、ZIN端子からのエッジ入力によりカウンタのクリアを行います。ZIN端子入力信号のどのエッジでカウントクリアを行うかをカウンタコントロールレジスタ上位(CCRH)内のCGE1,CGE0ビットで選択します。本機能によりエンコードのZ相出力を本端子に入力することで、エンコードのカウントの起点でUDCRレジスタをクリアすることが可能です。

ゲート機能起動時は、ZIN端子からのレベル入力によりカウントのイネーブル / ディセーブルを行います。ZIN端子入力信号のどのレベルでイネーブルとするかをCCRHレジスタ内のCGE1,CGE0ビットで選択します。

本機能は、全カウントモードで使用可能です。

表 14.7-2に、ZIN端子入力信号による検出エッジの選択について示します。

表 14.7-2 ZIN端子入力信号による検出エッジの選択

CGE1,CGE0	カウントクリア機能時	ゲート機能
00 _B	検出禁止	検出禁止
01 _B	立上りエッジ	LOWレベル
10 _B	立下りエッジ	HIGHレベル

カウント方向フラグ，カウント方向転換フラグ

カウント方向フラグ(カウンタステータスレジスタ(CSR)内のUDF1,UDF0ビット)は、アップ/ダウンカウント時に、直前のカウントがアップカウントであったかダウンカウントであったかを示します。AIN,BIN両端子の入力から生成されたカウントクロックから判断して、カウントの度にフラグを書き換えます。

モータの制御等で、現在の回転方向を知りたい場合などにこのフラグをみることで回転方向がを判別できます。

本機能は、全カウントモードで使用可能です。

表 14.7-3に、カウント方向フラグを示します。

表 14.7-3 カウント方向フラグ

UDF1,UDF0	カウント方向
01 _B	ダウンカウント
10 _B	アップカウント
11 _B	アップ/ダウン同時発生(カウント動作は行われない)

カウント方向転換フラグ(カウンタコントロールレジスタ上位(CCRH)内のCDCFビットは、カウント方向がアップ ダウンで変わった場合にセットされます。このフラグがセットされると共にCPUに対して割り込みが発生します。この割り込みと、カウント方向フラグ(CERレジスタ内のUDF1,UDF0ビット)を参照することにより、カウント方向がどちらからどちらへ変化したかがわかります。ただし、方向転換の期間が短く、連続して発生した場合などは、方向転換後のフラグの示す方向が元に戻り同一方向となる場合がありますので注意が必要です。

表 14.7-4にカウント方向転換フラグを示します。

表 14.7-4 カウント方向転換フラグ

CDCF	カウント方向転換検出
0 _B	方向転換無し
1 _B	方向転換有り(1回以上)

コンペア検出フラグ

コンペア検出フラグ(カウンタステータスレジスタ(CSR)内のCMPFビット)は、カウント動作中にアップダウンカウンタレジスタ(UDCR)の値とリロード/コンペアレジスタ(RCR)の値が等しくなった時にセットされます。カウントアップ一致のほか、リロードイベント発生による一致、カウント起動時にすでに一致している場合もセットされます。

ただし、ダウンカウントにより一致した場合(アンダフローによるリロードでのコンペアは除く)は、一致と見なしません。この場合はフラグはセットされません。

8ビット×2ch動作，16ビット×1ch動作

このモジュールは、8ビットアップダウンカウンタ×2ch、または、16ビットアップダウンカウンタ×1chとして使用できます。カウンタコントロールレジスタ0上位(CCRH0)M16Eビットに"0"を書き込むことにより、8ビット×2chモードとなり、"1"を書き込むことにより16ビット×1chモードとなります。

16ビット×1chとして動作する場合、CSR0,CCRL0,CCRH0レジスタが有効となり、CSR1,CCRL1,CCRH1レジスタは使用できません。また入力端子は、AIN0,BIN0,ZIN0端子が有効となり、AIN1,BIN1,ZIN1端子は使用できません。

第15章 DTP/外部割込み

この章には，DTP/外部割込みの機能と動作について示します。

- 15.1 DTP/外部割込みの概要
- 15.2 DTP/外部割込みのレジスタ
- 15.3 DTP/外部割込みの動作
- 15.4 DTP/外部割込みの使用上の注意

15.1 DTP/外部割込みの概要

DTP(Data Transfer Peripheral)/外部割込みは、拡張インテリジェントI/Oサービス(EI²OS)あるいは外部割込み処理を起動させるための周辺です。

DTP/外部割込み機能

DTP/外部割込み回路は、デバイス外部に存在するペリフェラルとF²MC-16LX CPUとの間にあって、外部ペリフェラルが発生するDMA要求あるいは割込み要求を受け取り、これをF²MC-16LX CPUに伝えて拡張インテリジェントI/Oサービス(EI²OS)あるいは割込処理を起動させるための周辺です。要求レベルとして、拡張インテリジェントI/Oサービス(EI²OS)の場合は"H", "L"の2種が、外部割込み要求の場合は"H", "L"のほか立上りエッジと立下りエッジの計4種が選択可能です。Ch0 ~ ch1は、レベルによる要求入力できませんが、両エッジ入力が可能です。

*：外部ペリフェラルとは、MB90570シリーズのデバイスの外部に接続される周辺機能デバイスのことです。

< 注意事項 >

ch0, ch1は、インテリジェントI/Oサービスには使えません。

DTP/外部割込みのブロックダイアグラム

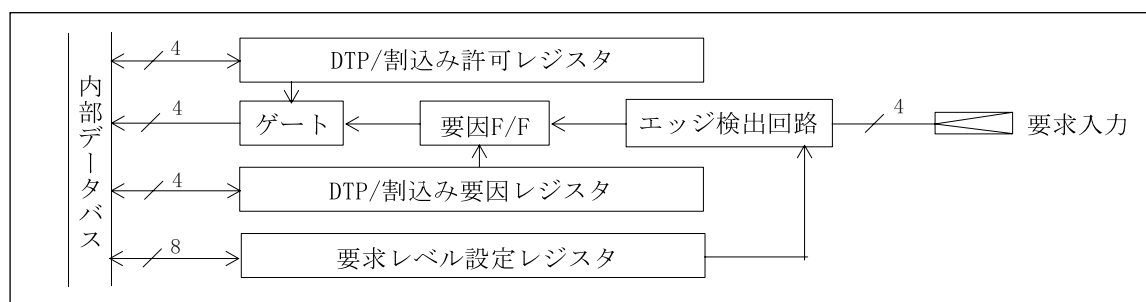


図 15.1-1 DTP/外部割込みのブロックダイアグラム

15.2 DTP/外部割込みのレジスタ

DTP/外部割込みのレジスタには、次の3種類のレジスタがあります。

- ・DTP/割込み許可レジスタ(ENIR)
- ・DTP/割込み要因レジスタ(EIRR)
- ・要求レベル設定レジスタ(ELVR)

DTP/外部割込みのレジスタ

DTP/割込み許可レジスタ									↔ビットNo.
	7	6	5	4	3	2	1	0	
アドレス:000030H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	ENIR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
DTP/割込み要因レジスタ									↔ビットNo.
	15	14	13	12	11	10	9	8	
アドレス:000031H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EIRR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
要求レベル設定レジスタ									↔ビットNo.
	15	14	13	12	11	10	9	8	
アドレス:000033H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
	7	6	5	4	3	2	1	0	↔ビットNo.
アドレス:000032H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 15.2-1 DTP/外部割込みのレジスタ

15.2.1 DTP/割込み許可レジスタ(ENIR)

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

DTP/割込み許可レジスタ(ENIR)

DTP/割込み許可レジスタ	7	6	5	4	3	2	1	0	↔ビットNo.
アドレス：000030 _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	ENIR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 15.2-2 DTP/割込み許可レジスタ(ENIR)

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

15.2.2 DTP/割込み要因レジスタ(EIRR)

DTP/割込み要因レジスタ(EIRR)は、読み出し時には対応するDTP/外部割込み要求があることを示し、書き込み時にはこの要求を示すフリップフロップ内容をクリアするレジスタです。

DTP/割込み要因レジスタ(EIRR)

DTP/割込み要因レジスタ	15	14	13	12	11	10	9	8	↔ビットNo.
アドレス:000031 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EIRR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 15.2-3 DTP/割込み要因レジスタ(EIRR)

このレジスタを読み出したときに"1"であった場合、このビットに対応する端子にDTP/外部割込み要求があることを示します。

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

15.2.3 要求レベル設定レジスタ(ELVR)

要求レベル設定レジスタ(ELVR)は、要求検出の選択を行うレジスタです。

要求レベル設定レジスタ(ELVR)

要求レベル設定レジスタ									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:000033H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス:000032H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 15.2-4 要求レベル設定レジスタ(ELVR)

1つの端子あたり2ビットが割当てられていて、LA2～LA7，LB2～LB7は表 15.2-1に示すような対応となり，LA0～LA1，LB0～LB1は表 15.2-1のような対応となります。要求入力レベルの場合，クリアしても入力がアクティブならば再びセットされます。

表 15.2-1 要求レベル設定レジスタ(ELVR)の動作 (LA2～LA7，LB2～LB7)

LBx	LAx	動作
0	0	Lレベルで要求あり
0	1	Hレベルで要求あり
1	0	立上り エッジで要求あり
1	1	立下り エッジで要求あり

表 15.2-2 要求レベル設定レジスタ(ELVR)の動作 (LA0～LA1，LB0～LB1)

LBx	LAx	動作
0	0	両エッジで要求あり
0	1	要求検出せず
1	0	立上り エッジで要求あり
1	1	立下り エッジで要求あり

15.3 DTP/外部割込みの動作

DTP/外部割込みは、外部割込み機能とDTP機能があります。各機能の動作について説明します。

DTP/外部割込みの動作

外部割込み要求を設定したのち、対応する端子に要求レベル設定レジスタ(ELVR)で設定された要求が入力されると本リソースは割込みコントローラに対して割込み要求信号を発生します。割込みコントローラ内で同時発生した割込みの優先順位を識別した結果、本リソースからの割込みが最も優先順位が高かった場合に、割込みコントローラはF²MC-16LX CPU に対して割込み要求を発生します。

F²MC-16LX CPUはプロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)と割込み要求を比較し、要求レベルがILMレジスタより高かった場合に現在実行中の命令が終了し次第、ハードウェア割込み処理マイクロプログラムを起動します。

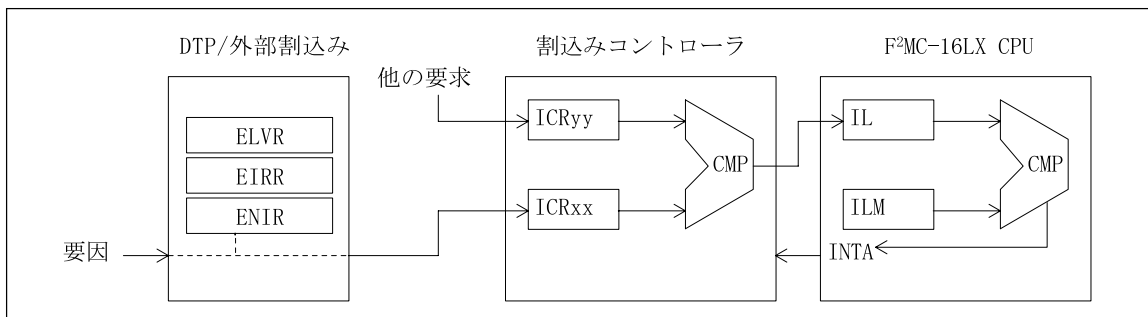


図 15.3-1 外部割込みの動作

ハードウェア割込み処理マイクロプログラムにおいて、CPUは割込み制御レジスタ(ICR)内のISEビットの情報を読み出し、これにより当該要求が割込み処理であることを識別し、割込み処理マイクロプログラムへ分岐します。割込み処理マイクロでは割込みベクトル領域の読出しと割込みコントローラへの割込みアクノリッジを発生し、ベクトルから生成したマクロ命令のジャンプ先アドレスをプログラムカウンタへ転送した上で、ユーザの割込み処理プログラムを実行します。

DTPの動作

初期化としてユーザプログラム内では拡張インテリジェントI/Oサービス(EI²OS)を起動するに当たって、拡張インテリジェントI/Oサービスディスクリプタ内のI/Oアドレスポインタに000000_Hから0000FF_Hに割りつけられているレジスタのアドレスを設定し、バッファアドレスポインタにメモリバッファの先頭アドレスを設定します。

DTPの動作シーケンスは外部割込みの場合とほぼ同じで、CPUがハードウェア割込み処理マイクロプログラムを起動するまでは全く同じです。DTPの場合はCPUがハードウェア割込み処理マイクロプログラム内で読出す割込み制御レジスタ(ICR)内のISEビットの内容がDTPを示しているので、拡張インテリジェントI/Oサービス処理マイクロプログラムへ制御を移します。拡張インテリジェントI/Oサービス(EI²OS)が起動されると、アドレッシングされている外部ペリフェラルにリードまたはライト信号が送られ、本チップとの転送が行われます。外部ペリフェラルはその転送が行われてから3マシンサイクル以内に本チップに対する割込み要求を取り下げてください。転送が終了するとディスクリプタの更新などが行われ、その後転送要因をクリアする信号を割込みコントローラに発生させます。転送要因を

クリアする信号を受けとった本リソースは要因を保持しているフリップフロップをクリアして端子からの次の要求に備えます。インテリジェントI/Oサービス処理の詳細は『F²MC[®]-16LX プログラミングマニュアル』を参照してください。

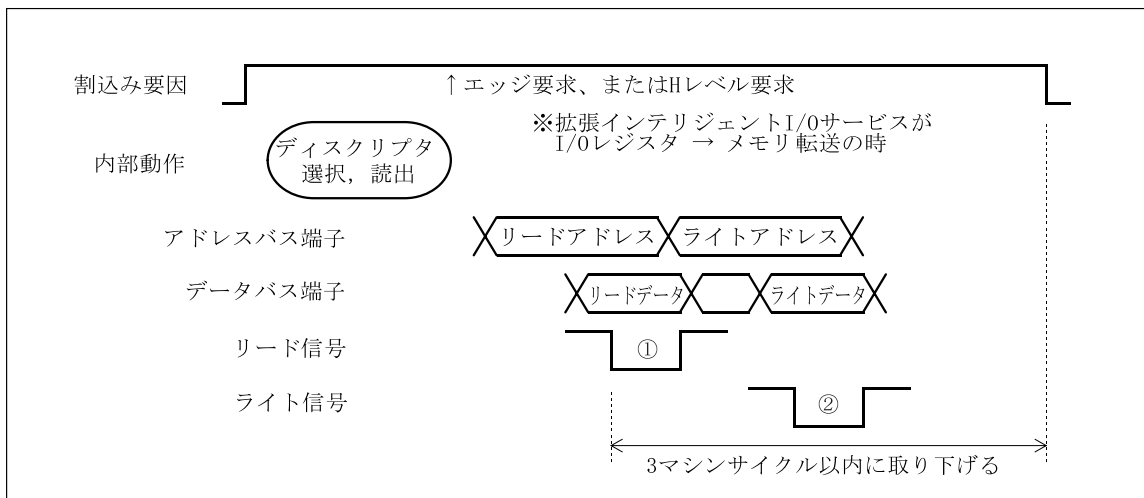


図 15.3-2 DTP動作終了時の外部割込み取り下げタイミング

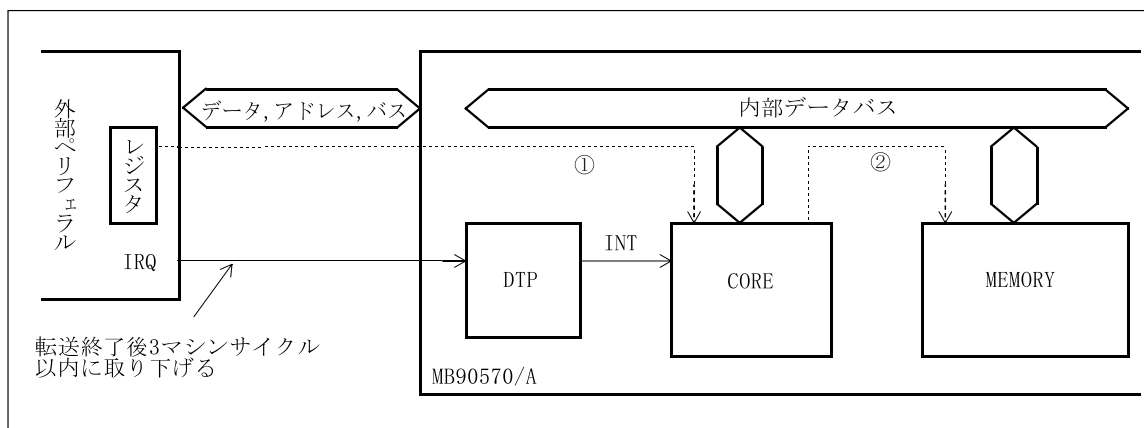


図 15.3-3 外部ペリフェラルとのインタフェース簡略例

外部割込み要求とDTP要求の切替え

外部割込み要求とDTP要求の切替えは、割込みコントローラの中にある、本リソースに対応する割込み制御レジスタ(ICR)内のISEビットの設定によって行います。

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

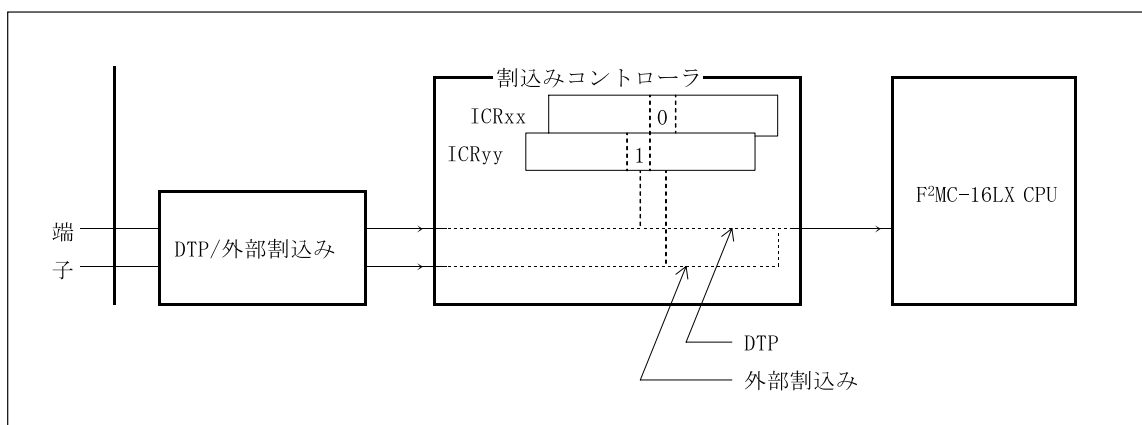


図 15.3-4 外部割込み要求とDTP要求の切替え

15.4 DTP/外部割込みの使用上の注意

DTP/外部割込みを使用する際には、次の4つの項目について特に注意が必要です。

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

DTPを用いた時の外部に接続するペリフェラルの条件

DTPがサポートできる外部ペリフェラルは、転送が行われたことにより自動的に要求をクリアするものでなければなりません。また、転送動作が開始してから3マシサイクル以内（暫定値）で転送要求を取り下げようになっていないと、本リソースは次の転送要求が発生したものととして扱ってしまいます。

スタンバイからの復帰

MB90V570, MB90F574, MB90573, MB90574の場合

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

MB90V570A, MB90F574A, MB90574Cの場合

クロック停止モードのスタンバイ状態からの復帰に外部割込みを使用する場合、ch2～ch7ではHレベル要求またはLレベル要求としてください。また、ch0～ch1ではエッジ要求としてください。

DTP/外部割込みの動作手順

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

- 1) DTP/外部割込み許可レジスタ(ENIR)の対象となるビットをディセーブル状態にする。
- 2) 要求レベル設定レジスタ(ELVR)の対象となるビットを設定する。
- 3) DTP/外部割込み要因レジスタ(EIRR)の対象となるビットをクリアする。
- 4) 許可レジスタの対象となるビットをイネーブル状態にする。

(ただし、3)と4)はワード指定による同時書き込み可。)

本リソース内のレジスタを設定する時には必ずENIRレジスタをディセーブル状態に設定しておかなくてはなりません。また、ENIRレジスタをイネーブル状態にする前に必ずENIRレジスタをクリアしておく必要があります。これは、レジスタ設定時や割込み許可状態時に誤って割込み要因が起こってしまうことを避けるためです。

外部割込み要求レベル

- 要求レベルがエッジ要求の場合，エッジが入力されたことを検出するためには，パルス幅は最小3マシンサイクル必要とします。
- 要求入力レベルがレベル検出の場合，外部より要求入力が入ってその後取り下げられても図 15.4-1に示すように，内部に要因保持回路が存在するので，割込みコントローラへの要求はアクティブのままです。割込みコントローラへの要求を取り下げるには，外部割込み要求フラグビットをクリアして，図 15.4-2に示すように要因保持回路をクリアする必要があります。

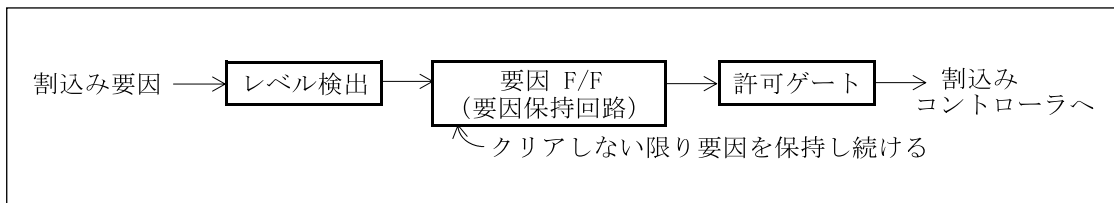


図 15.4-1 レベル設定時の要因保持回路のクリア

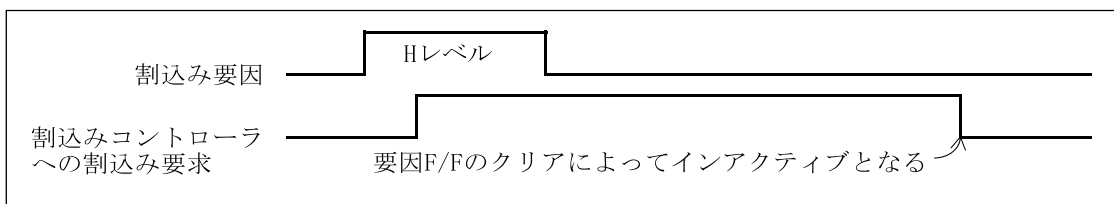


図 15.4-2 割込み許可時の割込み要因と割込みコントローラへの割込み要求

第16章 遅延割込み発生モジュール

この章では、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。

16.1 遅延割込み発生モジュールの概要

16.2 遅延割込み発生モジュールの動作

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

遅延割込み発生モジュールのブロックダイアグラム

図 16.1-1に、遅延割込み発生モジュールのブロックダイアグラムを示します。

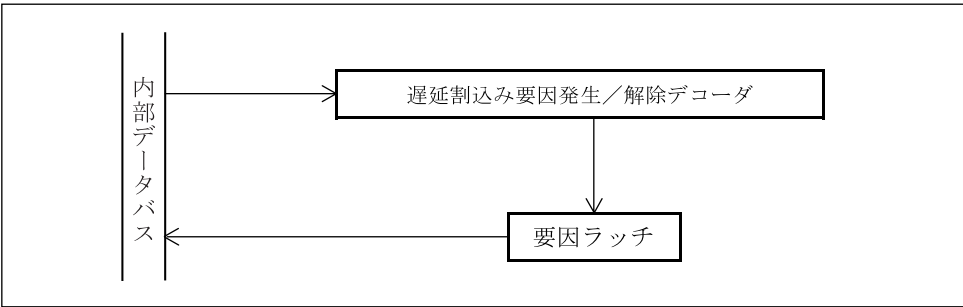


図 16.1-1 遅延割込み発生モジュールのブロックダイアグラム

遅延割込み発生モジュールのレジスタ

図 16.1-1に、遅延割込み発生モジュール [遅延割込み要因発生 / 解除レジスタ (DIRR: Delayed Interrupt Request Register)] のレジスタ構成を示します。
遅延割込み要因発生 / 解除レジスタ (DIRR) は遅延割込み要求の発生 / 解除を制御するレジスタで、このレジスタに対して "1" を書き込み時には遅延割込みの要求を発生させ、"0" を書き込み時には遅延割込みの要求を解除します。
リセット時には要因解除状態になります。
予約ビット領域は "0"、"1" どちらの書き込みも結構ですが、将来の拡張を考慮してこのレジスタをアクセスする際にはセットビット、クリアビット命令を使用することをお勧めします。

遅延割込み要因発生／解除レジスタ								↔ビットNo.	
	15	14	13	12	11	10	9	8	
アドレス:00009F _H	—	—	—	—	—	—	—	R0	DIRR
リード/ライト ↔	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
初期値 ↔	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	

図 16.1-2 遅延割込み要因発生 / 解除レジスタ (DIRR)

16.2 遅延割込み発生モジュールの動作

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

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

遅延割込み発生モジュールの動作

F²MC-16LX CPUはプロセッサステータス(PS)内のインタラプトレベルマスクレジスタ(ILM)と割込み要求を比較し、要求レベル(IL)がILMレジスタより高かったときに現在実行中の命令が終了し次第、ハードウェア割込み処理マイクロプログラムを起動します。この結果、本割込みに対する割込み処理ルーチンが実行されます。

割込み処理ルーチン内でDDIRレジスタの該当ビットの"0"を書込むことで本割込み要因をクリアし、合わせてタスクの切換えを行います。

図 16.2-1に、遅延割込み発生モジュールの動作を示します。

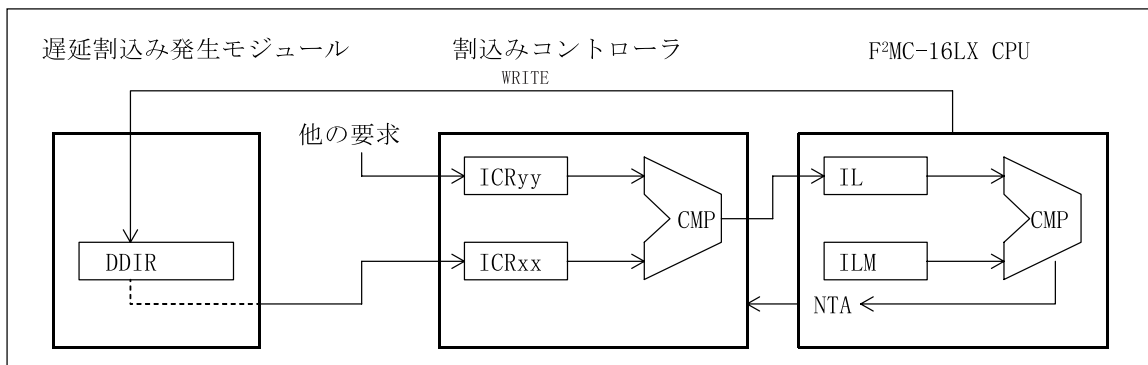


図 16.2-1 遅延割込み発生モジュールの動作

遅延割込み発生モジュールの使用上の注意

遅延割込み要求ラッチ

遅延割込み要求ラッチは、DIRRレジスタの該当するビットに"1"を書込むことでセットされ、同じビットに"0"を書込むことでクリアされます。したがって、割込み処理ルーチン内で要因をクリアするようにソフトウェアを作成しておかないと割込み処理から復帰した途端に再割込み処理を起動することになりますので注意してください。

第17章 A/Dコンバータ

この章では，A/Dコンバータの機能と動作について説明します。

- 17.1 A/Dコンバータの概要
- 17.2 A/Dコンバータのレジスタ
- 17.3 A/Dコンバータの動作
- 17.4 A/Dコンバータ使用上の注意
- 17.5 変換データ保護機能

17.1 A/Dコンバータの概要

A/Dコンバータは、アナログ入力電圧をデジタル値に変換するものです。

A/Dコンバータの概要

A/Dコンバータには、次の特長があります。

変換時間

最小26.3 μ s (マシクロック16MHz時)

サンプリング時間

1チャンネル当たり64マシサイクル/4096マシサイクル(4 μ s/256 μ s)の選択が可能 (マシクロック16MHz時)

コンペア時間

1チャンネルあたり、176マシサイクル/352マシサイクル
176マシクロックはマシクロック8MHz以下でご使用ください。

サンプル&ホールド回路付RC型逐次比較変換方式を採用

8/10ビットの分解能切換え可能

アナログ入力8チャンネルからプログラムで選択

- シングル変換モード:1チャンネルを選択変換
- スキャン変換モード:連続した複数のチャンネルを変換。最大8chプログラム可能
- 連続変換モード:指定チャンネルを繰り返し変換
- 停止変換モード:1チャンネルを変換したら一時停止して次の起動がかかるまで待機 (変換開始の同期が可能)

A/D変換終了時には、CPUに対してA/D変換終了の割り込み要求を発生させることができます。この割り込み発生でEI²OSを起動することができ、A/D変換結果データをメモリに転送できますので連続処理に適しています。

起動要因は、ソフト、外部トリガ (立下りエッジ)、タイマ (立上りエッジ) から選択できます。

17.2 A/Dコンバータのレジスタ

図 17.2-1に , A/Dコンバータのレジスタを示します。

A/Dコンバータのレジスタ

コントロールステータスレジスタ上位									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:000037 _H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	予約	ADCS2
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
コントロールステータスレジスタ下位									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス:000036 _H	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	ADCS1
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
データレジスタ上位									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:000039 _H	DSEL	ST1	ST0	CT1	CT0	—	D9	D8	ADCR2
リード/ライト⇒	(W)	(W)	(W)	(W)	(W)	(-)	(-)	(-)	
初期値⇒	(0)	(0)	(0)	(0)	(1)	(-)	(-)	(-)	
データレジスタ下位									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス:000038 _H	D7	D6	D5	D4	D3	D2	D1	D0	ADCR1
リード/ライト⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 17.2-1 A/Dコンバータのレジスタ

17.2.1 コントロールステータスレジスタ(ADCS1,2)

コントロールステータスレジスタ(ADCS1,2)は、A/Dコンバータの制御およびステータス表示を行います。

コントロールステータスレジスタ(ADCS1,2)

コントロールステータスレジスタ上位									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:000037H	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	予約	ADCS2
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
コントロールステータスレジスタ下位									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス:000036H	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	ADCS1
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

図 17.2-2 コントロール・ステータスレジスタ(ADCS1,2)

< 注意事項 >

ADCS1は、A/D変換動作中に書換えないようにしてください。

【ビット15】BUSY(Busy flag and stop)

A/Dコンバータ動作表示用のビットです。

リード時

このビットが"0"であれば、A/D変換停止中であることを示し、"1"であれば、A/D変換動作中であることを示します。

ライト時

A/D動作中に本ビットに"0"を書き込むと強制的に動作を停止します。連続、停止モード時の強制停止に利用します。動作表示用のビットに"1"を書き込むことはできません。

RMW系命令では"1"が読めます。単発モードではA/D変換終了でクリアされます。連発、停止モードでは"0"書き込みで停止するまでクリアされません。

< 注意事項 >

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

【ビット14】INT(Interrupt)

変換データがデータレジスタ(ADCR)に書き込まれると、このビットに"1"がセットされます。

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

本ビットは"0"書込みとEI²OS割込みのクリア信号でクリアされます。

< 注意事項 >

本ビットの"0"書込みによるクリアは、A/D停止中に行ってください。

【ビット13】INTE(INTerrupt Enable)

変換終了による割込みの許可・禁止を行うビットです。

EI²OSを使用するときは、本ビットに"1"をセットしてください。EI²OSは、割込み要求発生で起動します。

表 17.2-1 INTE(割込みの許可・不許可指定ビット)の機能

INTE	機能
0	割込みの禁止 [初期値]
1	割込みの許可

【ビット12】PAUS (a/d converter PAUSe)

A/D変換動作が一時的に停止した場合に"1"がセットされます。

A/D変換結果を格納するレジスタが1つのため、連続で変換した場合、変換結果をEI²OSで転送しなければ前データが壊れてしまいます。

これを保護するため、データレジスタの内容をEI²OSで転送しなければ次の変換データが格納されず、A/D変換動作は停止します。EI²OSで転送を終了するとA/Dは変換を再開します。

< 注意事項 >

PAUSビットは、EI²OSを使用したときのみ有効です。「17.5 変換データ保護機能」を参照してください。

【ビット11, 10】STS1,STS0(Start Source select)

A/D起動要因を選択します。

表 17.2-2 STS1,STS0(A/D起動要因選択ビット)の機能

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

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

外部端子トリガは立下りエッジを検出します。

外部トリガ入力レベルが"L"の場合に、本ビットを外部端子トリガ起動に設定すると、A/Dが起動する場合があります。

タイマ選択時は、16ビットリロードタイマ1の出力が選択されます。

【ビット9】STRT(StaRT)

STRTビットに"1"を書き込むことによりA/Dを起動します。再起動をかけるときは、再び書き込んでください。

停止モード時は、動作機能上再起動はかかりません。

< 注意事項 >

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

【ビット8】予約ビット

予約ビットです。必ず"0"を書込んでください。

【ビット7,6】MD1,MD0(a/d converter MoDe set)

動作モードを設定します。

表 17.2-3 MD1,MD0の動作モード

MD1	MD0	動作モード
0	0	単発モード，動作中の再起動は全て可能 [初期値]
0	1	単発モード，動作中の再起動不可能
1	0	連続モード，動作中の再起動不可能
1	1	停止モード，動作中の再起動不可能

単発モード

ANS2～ANS0ビットの設定チャンネルからANE2～ANE0ビットの設定チャンネルまでA/D変換を連続して行い、1回変換が終了したら停止します。

連続モード

ANS2～ANS0ビットの設定チャンネルからANE2～ANE0ビットの設定チャンネルまでA/D変換を繰り返し行います。

停止モード

ANS2～ANS0ビットの設定チャンネルからANE2～ANE0ビットの設定チャンネルまで1チャンネルごとにA/D変換を行い一時停止します。一時停止中の変換再開は、起動要因発生によって行われます。

< 注意事項 >

-
- ・連続モード，停止モードでA/D変換を起動すると，BUSYビットで停止するまで変換動作を続けます。
 - ・停止は，BUSYビットに"0"を書き込むことにより行われます。
 - ・単発，連続，停止の各モードの再起動の不可能はタイマ，外部トリガ，ソフト全ての起動に適用されます。
-

【ビット5,4,3】ANS2, ANS1, ANS0(ANalog Start channel set)

A/D変換の開始チャンネルを設定します。

A/Dコンバータを起動すると、このビットで選択されたチャンネルからA/D変換を開始します。A/D変換中は、変換チャンネルが読めます。停止モードで停止中は、前に変換したチャンネルが読めます。

表 17.2-4 ANS2, ANS1, ANS0ビットの開始チャンネル

ANS2	ANS1	ANS0	開始チャンネル
0	0	0	AN0 [初期値]
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

【ビット2, 1, 0】ANE2, ANE1, ANE0(ANalog End channel set)

A/D変換の終了チャンネルを設定します。

表 17.2-5 ANE2, ANE1, ANE0ビットの終了チャンネル

ANE2	ANE1	ANE0	終了チャンネル
0	0	0	AN0 [初期値]
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4
1	0	1	AN5
1	1	0	AN6
1	1	1	AN7

ANS2～ANS0ビットと同じチャンネルを設定すると1チャンネル変換になります(シングル変換)。

連続モード,停止モードを設定している場合は、本ビット群で設定されたチャンネルの変換が終わるとANS2～ANS0ビットで設定された開始チャンネルに戻ります。

設定チャンネルが $ANS < ANE$ の場合は、ANSより変換が始まり、7チャンネルまで変換したら0チャンネルに戻りANEまで変換します。

例 : チャンネル設定 $ANS = 6ch$, $ANE = 3ch$ で単発モードの場合

動作: 変換チャンネル 6ch 7ch 0ch 1ch 2ch 3ch

17.2.2 データレジスタ(ADCR1,2)

データレジスタ(ADCR1,2)では、A/D変換結果を格納するレジスタであり、A/D変換の分解能の選択やマシンサイクルを設定します。

データレジスタ(ADCR1,2)

データレジスタ上位		15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:000039 _H		DSEL	ST1	ST0	CT1	CT0	—	D9	D8	ADCR2
リード/ライト⇒		(W)	(W)	(W)	(W)	(W)	(—)	(R)	(R)	
初期値⇒		(0)	(0)	(0)	(0)	(1)	(—)	(X)	(X)	
データレジスタ下位		7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス:000038 _H		D7	D6	D5	D4	D3	D2	D1	D0	ADCR1
リード/ライト⇒		(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値⇒		(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 17.2-3 データレジスタ(ADCR1,2)

< 注意事項 >

ADCR2はリード値は不定です。

【ビット15】DSEL

8/10ビットの分解能を選択します。

表 17.2-6 SELB(分解能選択ビット)

SELB	機能
0	10ビットモード [初期値]
1	8ビットモード

【ビット14, 13】ST1, ST0(Sampling Time)

サンプリング時のマシンサイクル数を設定します。

表 17.2-7 ST1, ST0(サンプリング時のマシンサイクル数設定ビット)

ST1	ST0	サンプリング時マシンサイクル	サンプリング時間
0	0	64マシンサイクル	4 μ s / マシンクロック 16MHz
0	1	256マシンサイクル	16 μ s / マシンクロック 16MHz
1	0	1024マシンサイクル	64 μ s / マシンクロック 16MHz
1	1	4096マシンサイクル	256 μ s / マシンクロック 16MHz

【ビット12, 11】CT1, CT0(Compare Time)

コンペア時のマシンサイクル数を設定します。

表 17.2-8 CT1, CT0(コンペア時のマシンサイクル数設定ビット)

CT1	CT0	コンペア時マシンサイクル	コンペア時間
0	0	176マシンサイクル	22 μ s/マシクロック 8MHz
0	1	352マシンサイクル	22 μ s /マシクロック 16MHz
1	0	704マシンサイクル	44 μ s/マシクロック 16MHz
1	1	1408マシンサイクル	88 μ s/マシクロック 16MHz

< 注意事項 >

本ビットを"00"に設定するときは、マシクロック8MHz以下の場合にしてください。8MHzより高速の場合には、変換精度が保証されません。

【ビット9, 8】D9, D8

DSEL = 0の時、有効になります。変換結果であるデジタル値の上位2ビットが格納されます。

【ビット7~0】D7~D0

A/D変換格納レジスタで、変換結果であるデジタル値が格納されます。

このレジスタの値は一回の変換終了時ごとに更新されます。通常は最終変換値が格納されています。「17.5 変換データ保護機能」を参照してください。

このレジスタのリセット時は不定です。

< 注意事項 >

A/D動作中、このレジスタにデータを書き込まないようにしてください。

17.3 A/Dコンバータの動作

A/Dコンバータには、次の3種類の動作モードがあります。

- ・単発モード
- ・連続モード
- ・停止モード

動作説明

A/D変換器は、逐次比較方式で動作し、8/10ビットの分解切替え可能です。

このA/D変換器は変換結果格納用のレジスタが1つ（8ビット）しかないため、1回の変換終了とともに変換データレジスタ（ADCR0）が更新されてしまいます。

このため、A/D変換器単独では連続変換処理には適しませんのでEIOS機能を使って変換データをメモリに転送しながら変換することを推奨いたします。

単発モード

単発モードは、ANSビットとANEビットで設定されたアナログ入力を順に変換していき、ANEビットで設定された終了チャンネルまで変換が終わるとA/Dは動作を停止します。

開始チャンネルと終了チャンネルが同じ場合（ANS = ANE）は、ANSで指定したチャンネルのみの変換動作になります。

【例】:

```
ANS = 000, ANE = 011
開始  AN0  AN1  AN2  AN3  終了
ANS = 010, ANE = 010
開始  AN2  終了
```

連続モード

連続モードは、ANSビットとANEビットで設定されたアナログ入力を順に変換していきANEビットで設定された終了チャンネルまで変換が終わるとANSのアナログ入力に戻りA/D変換動作を続けます。

開始チャンネルと終了チャンネルが同じ場合（ANS = ANE）は、ANSで指定したチャンネルのみの変換を続けます。

【例】

```
ANS = 000, ANE = 011
開始  AN0  AN1  AN2  AN3  AN0  繰り返し
ANS = 010, ANE = 010
開始  AN2  AN2  AN2  繰り返し
```

連続モードで変換させると、コントロールステータスレジスタ（ADCS）内のBUSYビットに"0"を書き込むまで変換を繰り返し続けます（BUSYビットに"0"を書き込む 強制動作停止）。

強制動作停止を行うと変換中のものは途中で止まってしまうため、注意してください（強制動作停止した場合は、データレジスタ（ADCR）には変換完了した前データが格納されています）。

停止モード

停止モードは、ANSビットとANEビットで設定されたアナログ入力を順に変換していきますが1チャンネル変換する度に変換動作を一時停止します。一時停止の解除は、再起動により行われます。

ANEビットで設定された終了チャンネルまで変換が終わるとANSのアナログ入力に戻りA/D変換動作を続けます。

開始チャンネルと終了チャンネルが同じ場合（ANS = ANE）は、ANSで指定したチャンネルの変換をします。

【例】

ANS = 000, ANE = 011

開始 AN0 停止 起動 AN1 停止 起動 AN2 停止 起動 AN3 停止
起動 AN0 繰り返し

ANS = 010, ANE = 010

開始 AN2 停止 起動 AN2 停止 起動 AN2 繰り返し

この場合の起動要因は、ADCSレジスタ内のSTS1,0ビットで設定されたものです。

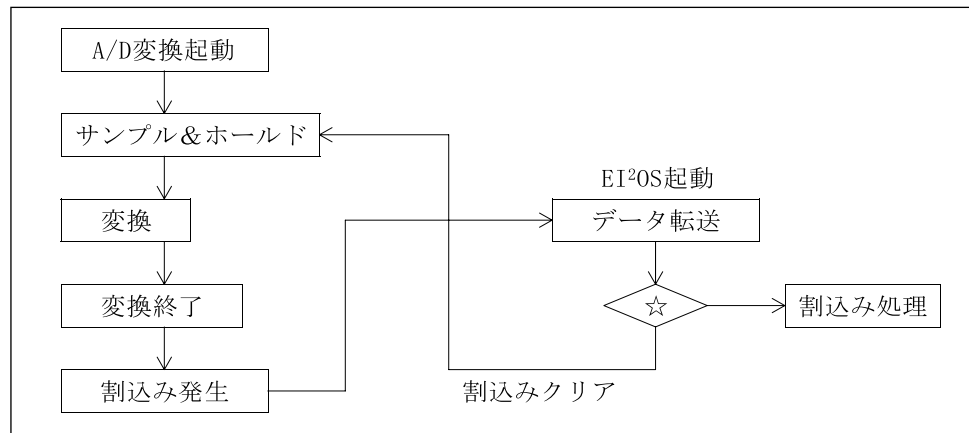
このモードを使用することにより、変換開始の同期をかけることが可能です。

17.3.1 EI²OSを使用した変換動作

A/Dコンバータは、拡張インテリジェントI/Oサービス（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

EI²OSを利用すると変換データ保護機能で、連続変換時にもデータを欠落することなく、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。



はEI²OSの設定で決まります。

図 17.3-1 A/D変換起動から変換データ転送までのフロー例（連続モード）

17.3.2 単発モード時のEI²OSの起動例

単発モード時のEI²OSは、次の手順で起動します。

- ・ アナログ入力(AN1 ~ AN3)の変換をして終了する。
- ・ 変換データは、200_H ~ 206_H番地に順に転送する。
- ・ 起動はソフトで行う。
- ・ 割込みレベル最強。

単発モード時のEI²OSの起動例

表 17.3-1 単発モード時のEI²OSの起動例

設定項目	プログラム例	動作説明
EI ² OSの設定	MOV ICR0, #08H	最強割込み設定, 割込み時EI ² OS起動, ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #02H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送, 転送後転送先番地をインクリメントする。I/Oからメモリへ転送。
	MOV IOAL, #3EH	A/Dコンバータの結果レジスタの設定
	MOV IOAH, #00H	
	MOV DCTL, #03H	EI ² OS転送を3回行う。変換回数と同じにする。
	MOV DCTH, #00H	
A/Dコンバータ設定	MOV ADCS1, #0BH	単発モード, 開始チャネルAN1, 終了チャネルAN3
	MOV ADCS2, #A2H	ソフト起動, A/D変換開始
その他の処理	:	:
EI ² OS終了割込みシーケンス	MOV ADCS2, #80H	-
	RETI	割込みからの復帰

ICR3 : 割込み制御レジスタ
BAPL : バッファアドレスポインタ下位
BAPM : バッファアドレスポインタ中位
BAPH : バッファアドレスポインタ上位
ISCS : EI²OSステータスレジスタ
IOAL : I/Oアドレスレジスタ下位
IOAH : I/Oアドレスレジスタ上位
DCTL : データカウンタ下位
DCTH : データカウンタ上位

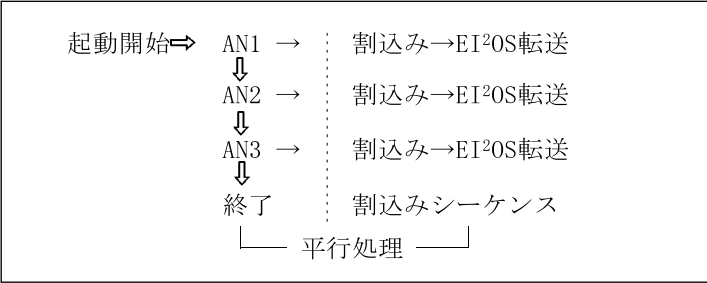


図 17.3-2 単発モード時のEI²OSの起動例

17.3.3 連続モード時のEI²OSの起動例

連続モード時のEI²OSは、次の手順で起動します。

- ・ アナログ入力(AN3 ~ AN5)の変換をして各チャネルの変換データを2つ取得する。
- ・ 変換データは600_H ~ 60C_H番地に順に転送する。
- ・ 起動は外部エッジ入力で行う。
- ・ 割込みレベル最強。

連続モード時のEI²OSの起動例

表 17.3-2 連続モード時のEI²OSの起動例

設定項目	プログラム例	動作説明
EI ² OSの設定	MOV ICR0, #08H	最強割込み設定, 割込み時EI ² OS起動, ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送, 転送後転送先番地をインクリメントする。I/Oからメモリへ転送。リソースからの要求で終了する。
	MOV IOAL, #3EH	転送元アドレス
	MOV IOAH, #00H	
	MOV DCTL, #06H MOV DCTH, #00H	EI ² OS転送を6回行う。3ch × 2分のデータ転送を行う。
A/Dコンバータ設定	MOV ADCS1, #9DH	単発モード, 開始チャネルAN3, 終了チャネルAN5
	MOV ADCS2, #A4H	外部エッジ起動, A/D変換開始
その他の処理	:	:
EI ² OS終了割込みシーケンス	MOV ADCS2, #80H	-
	RETI	割込みからの復帰

ICR3 : 割込み制御レジスタ

BAPL : バッファアドレスポインタ下位

BAPM : バッファアドレスポインタ中位

BAPH : バッファアドレスポインタ上位

ISCS : EI²OSステータスレジスタ

IOAL : I/Oアドレスレジスタ下位

IOAH : I/Oアドレスレジスタ上位

DCTL : データカウンタ下位

DCTH : データカウンタ上位

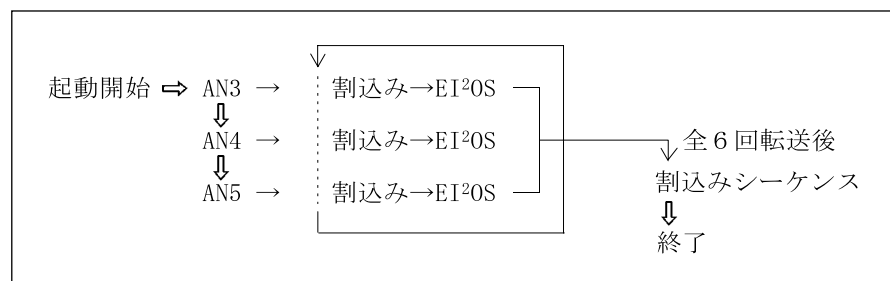


図 17.3-3 連続モード時のEI²OSの起動例

17.3.4 停止モード時のEI²OS起動例

停止モード時のEI²OSは、次の手順で起動します。

- ・アナログ入力(AN3)を一定期間おいて12回変換する。
- ・変換データは600_H～618_H番地に順に転送する。
- ・起動は外部エッジ入力で行う。
- ・割込みレベル最強

停止モード時のEI²OS起動例

表 17.3-3 停止モード時のEI²OS起動例

設定項目	プログラム例	動作説明
EI ² OSの設定	MOV ICR0, #08H	最強割込み設定, 割込み時EI ² OS起動, ディスクリプタアドレスの設定。
	MOV BAPL, #00H	変換データの転送先番地。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #19H	ワードデータ転送, 転送後転送先番地をインクリメントする。I/Oからメモリへ転送。リソースからの要求で終了する。
	MOV IOAL, #3EH	転送元アドレス
	MOV IOAH, #00H	
	MOV DCTL, #0CH	EI ² OS転送を12回行う。
	MOV DCTH, #00H	
A/Dコンバータ設定	MOV ADCS1, #DBH	連続モード, 開始チャネルAN3, 終了チャネルAN3(1ch変換)
	MOV ADCS2, #A4H	外部エッジ起動, A/D変換開始
その他の処理	:	:
EI ² OS終了割込みシーケンス	MOV ADCS2, #80H	-
	RET I	割込みからの復帰

ICR3 : 割込み制御レジスタ
BAPL : バッファアドレスポインタ下位
BAPM : バッファアドレスポインタ中位
BAPH : バッファアドレスポインタ上位
ISCS : EI²OSステータスレジスタ
IOAL : I/Oアドレスレジスタ下位
IOAH : I/Oアドレスレジスタ上位
DCTL : データカウンタ下位
DCTH : データカウンタ上位

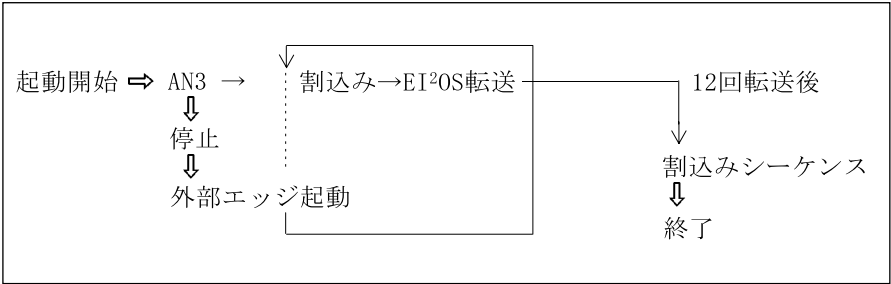


図 17.3-4 停止モード時のEI²OS起動例

17.4 A/Dコンバータ使用上の注意

A/Dコンバータを使用する場合の注意点を示します。

A/Dコンバータ使用上の注意

A/Dコンバータを外部トリガまたは内部タイマを使って起動する場合ADCS2レジスタのA/D起動要因ビットSTS1,0で設定しますが、この場合に外部トリガおよび内部タイマの入力値はインアクティブ側の状態で設定してください。アクティブ側にしておくと動作し始める場合があります。

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

アナログ入力に使用する端子は、必ずその対応するADERレジスタのビットを"1"にしてください。

詳細は、「第8章 I/Oポート」を参照ください。

17.5 変換データ保護機能

本A/Dコンバータは、変換データ保護機能を持ちEI²OSを使って連続変換と複数のデータを確保します。

変換データ保護機能

変換データレジスタは1つですので、連続でA/D変換をすると、1回の変換終了と共に変換データが格納されて前データが失われます。これを保護するために、A/Dコンバータは変換が終了しても前データがEI²OSを使ってメモリへ転送されていないと、変換データがレジスタに格納されずA/D変換動作は一時停止します。

一時停止の解除は、EI²OSでメモリへ前データが転送された後に、行われます。前データが転送されていれば、一時停止することなくA/Dは連続して変換します。

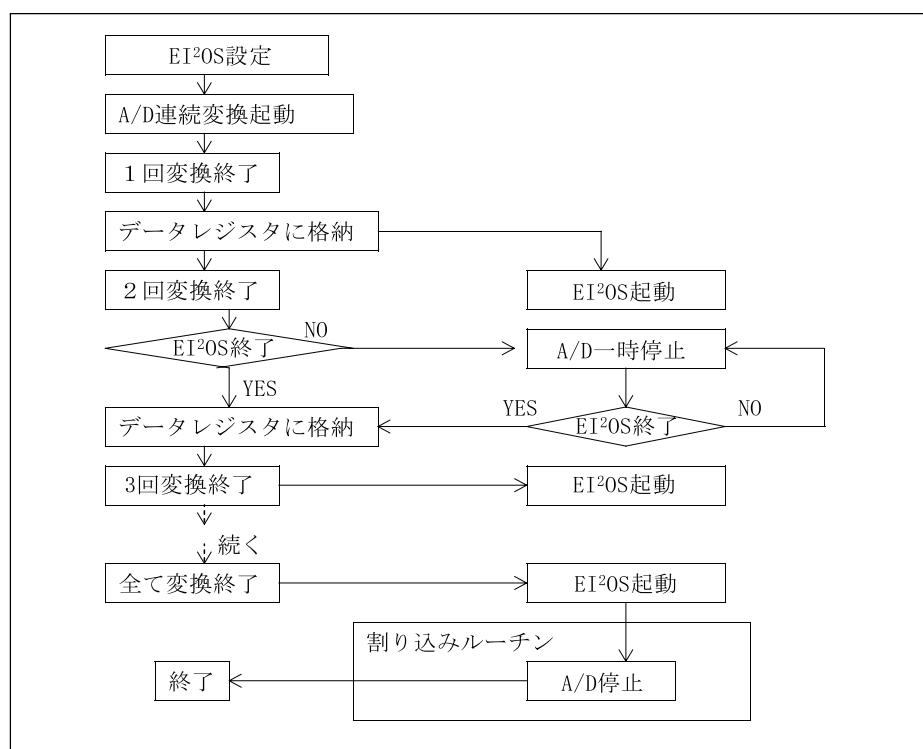


図 17.5-1 データ保護機能フロー (EI²OSを使用時)

< 注意事項 >

- この機能はADCS2レジスタのINTビット、INTEビットに関係しています。
- データ保護機能は割り込み許可 (INTE = "1") 状態でしか動作しないようになっています。
割り込み禁止 (INTE = "0") の場合は、本機能は動作せず、連続でA/D変換を行った場合は次々に変換データはレジスタに格納されて旧データは失われます。
- 割り込み許可 (INTE = "1") 状態でEI²OSを使わない場合はINTビットはクリアされないため、データ保護機能が働きA/Dは変換を一時停止状態にします。この場合は割り込みシーケンスでINTビットをクリアすると停止状態が解除されます。
- EI²OS動作中でA/Dが一時停止をしているときに割り込みを禁止にするとA/D変換が動作し、旧データの転送前に新データが書き込まれるときがあります。
- 一時停止中に再起動をかけると待機データが壊れます。

第18章 D/Aコンバータ

この章では，D/Aコンバータの機能と動作について説明します。

18.1 D/Aコンバータの概要

18.2 D/Aコンバータのレジスタ

18.3 D/Aコンバータの動作

18.1 D/Aコンバータの概要

このブロックは、R-2R方式のD/Aコンバータで、8ビットの分解能を備えています。D/Aコンバータは、2チャンネル内蔵しており、D/Aコントロールレジスタにて、それぞれ独立し出力制御を実行することができます。

D/Aコンバータのレジスタ

D/Aコンバータのレジスタ一覧を示します。

D/Aコンバータデータレジスタ1									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス : 00003B _H	DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	DADR1
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
D/Aコンバータデータレジスタ0									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス : 00003A _H	DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00	DADR0
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
D/Aコントロールレジスタ1									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス : 00003D _H	—	—	—	—	—	—	—	DAE1	DACR1
リード/ライト⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
初期値⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	
D/Aコントロールレジスタ0									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
アドレス : 00003C _H	—	—	—	—	—	—	—	DAE0	DACR0
リード/ライト⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
初期値⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	

図 18.1-1 D/Aコンバータのレジスタ

D/Aコンバータのブロックダイアグラム

図 18.1-2にD/Aコンバータのブロックダイアグラムを示します。

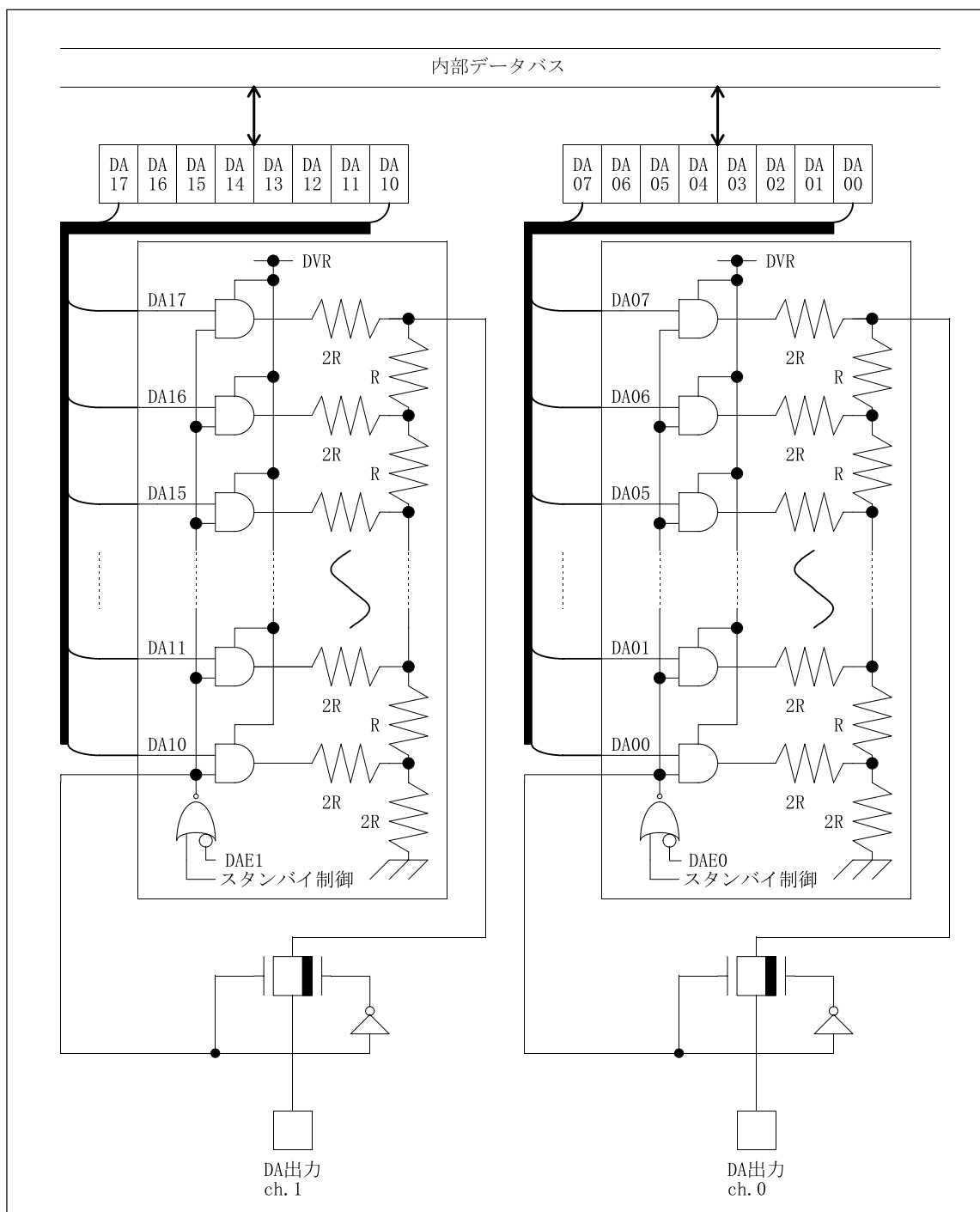


図 18.1-2 D/Aコンバータのブロックダイアグラム

18.2 D/Aコンバータのレジスタ

D/Aコンバータには、次の2種類のレジスタがあります。

- ・D/Aコンバータデータレジスタ(DADR0,1)
- ・D/Aコントロールレジスタ(DACR0,1)

D/Aコンバータデータレジスタ(DADR0,1)

D/Aコンバータデータレジスタ(DADR0,1)のレジスタ構成は、図 18.2-1のとおりです。

D/Aコンバータデータレジスタ1									
アドレス：00003Bh									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
	DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	DADR1
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
D/Aコンバータデータレジスタ0									
アドレス：00003Ah									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
	DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00	DADR0
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 18.2-1 D/Aデータレジスタ(DADR)

【ビット15～8】DA17-DA10

D/Aコンバータのチャンネル1の出力電圧を設定します。リセット時に、これらのビットは初期化されません。これらのビットは読出しと書込みが可能です。

【ビット7～0】DA07-DA00

D/Aコンバータのチャンネル0の出力電圧を設定します。リセット時に、これらのビットは初期化されません。これらのビットは読出しと書込みが可能です。

D/Aコントロールレジスタ(DACR0/1)

D/Aコントロールレジスタ(DACR0/1)のレジスタ構成は、図 18.2-2のとおりです。

D/Aコントロールレジスタ1									
アドレス：00003Dh									
	15	14	13	12	11	10	9	8	⇐ ビットNo.
	—	—	—	—	—	—	—	DAE1	DACR1
リード/ライト⇒	(—)	(—)	(—)	(—)	(—)	(—)	(—)	(R/W)	
初期値⇒	(—)	(—)	(—)	(—)	(—)	(—)	(—)	(0)	
D/Aコントロールレジスタ0									
アドレス：00003Ch									
	7	6	5	4	3	2	1	0	⇐ ビットNo.
	—	—	—	—	—	—	—	DAE0	DACR0
リード/ライト⇒	(—)	(—)	(—)	(—)	(—)	(—)	(—)	(R/W)	
初期値⇒	(—)	(—)	(—)	(—)	(—)	(—)	(—)	(0)	

図 18.2-2 D/Aコントロールレジスタ(DACR0/1)

【ビット8,0】DAE1,DAE0

D/Aコンバータの出力を許可または禁止します。DAE1はチャンネル1を制御し、DAE0はチャンネル0を制御します。

これらのビットに"1"が書き込まれると、D/A出力の許可が与えられます。"0"をセットすると、D/A出力を禁止します。

リセット時に、これらのビットは"0"に初期化されます。これらのビットは読出しと書込みが可能です。

18.3 D/Aコンバータの動作

D/A出力を開始するには，D/Aコントロールレジスタ(DACR)に属する対応D/A出力チャンネル用の許可ビットに"1"をセットします。

D/Aコンバータの動作

D/A出力を禁止すると，D/Aコンバータの各チャンネルの出力へ直列に挿入されたアナログスイッチがオフになります。さらに，D/Aコンバータの内部では"0"にクリアされ，直流電流の経路が遮断されます。以上のことはストップモードでも同様です。

D/Aコンバータの出力電圧は，0Vから $255/256 \times \text{DVR}$ の範囲となります。出力電圧の範囲を変更するには，外部でDVR電圧を調整します。

D/Aコンバータの出力は，内蔵バッファアンプを備えていません。また，アナログスイッチ(100)は出力へ向けて直列に挿入されているため，外部から出力に負荷を与えている場合，十分な整定時間を見込んでください。

表 18.3-1に，D/Aコンバータの出力電圧の理論値を示します。

表 18.3-1 D/Aコンバータの出力電圧の理論値

DA07～DA00 DA17～DA10 の設定値	出力電圧の理論値
00 _H	$0/256 \times \text{DVR} (=0\text{V})$
01 _H	$1/256 \times \text{DVR}$
02 _H	$2/256 \times \text{DVR}$
⋮	⋮
FD _H	$253/256 \times \text{DVR}$
FE _H	$254/256 \times \text{DVR}$
FF _H	$255/256 \times \text{DVR}$

第19章 UART

この章では，UARTの機能と動作について説明します。

- 19.1 UARTの概要
- 19.2 UARTのブロックダイアグラム
- 19.3 UARTのレジスタ
- 19.4 UARTのボーレート
- 19.5 UARTの動作
- 19.6 UARTのフラグと割込み発生要因
- 19.7 UARTの応用例と使用上の注意

19.1 UARTの概要

UARTは、非同期（調歩同期）通信，またはCLK同期通信を行うためのシリアルI/Oポートです。

UARTの特長

UARTには、以下の特長があります。

- 全二重ダブルバッファ
- 非同期（調歩同期），CLK同期通信が可能
- マルチプロセッサモードのサポート
- 専用ボーレートジェネレータ内蔵

表 19.1-1 ボーレート

動作	ボーレート*
非同期	31250/9615/4808/2404/1202 bps
CLK同期	2M/1M/500K/250K/125K/62.5K bps

*：内部マシニングクロックが 6,8,10,12,16MHz時の値です。

- 外部クロックによる自由なボーレートの設定が可能
- エラー検出機能（パリティ，フレーミング，オーバラン）
- 転送信号はNRZ符号
- 拡張インテリジェントI/Oサービス(EI²OS)のサポート

19.2 UARTのブロックダイアグラム

図 19.2-1に，UARTのブロックダイアグラムを示します。

UARTのブロックダイアグラム

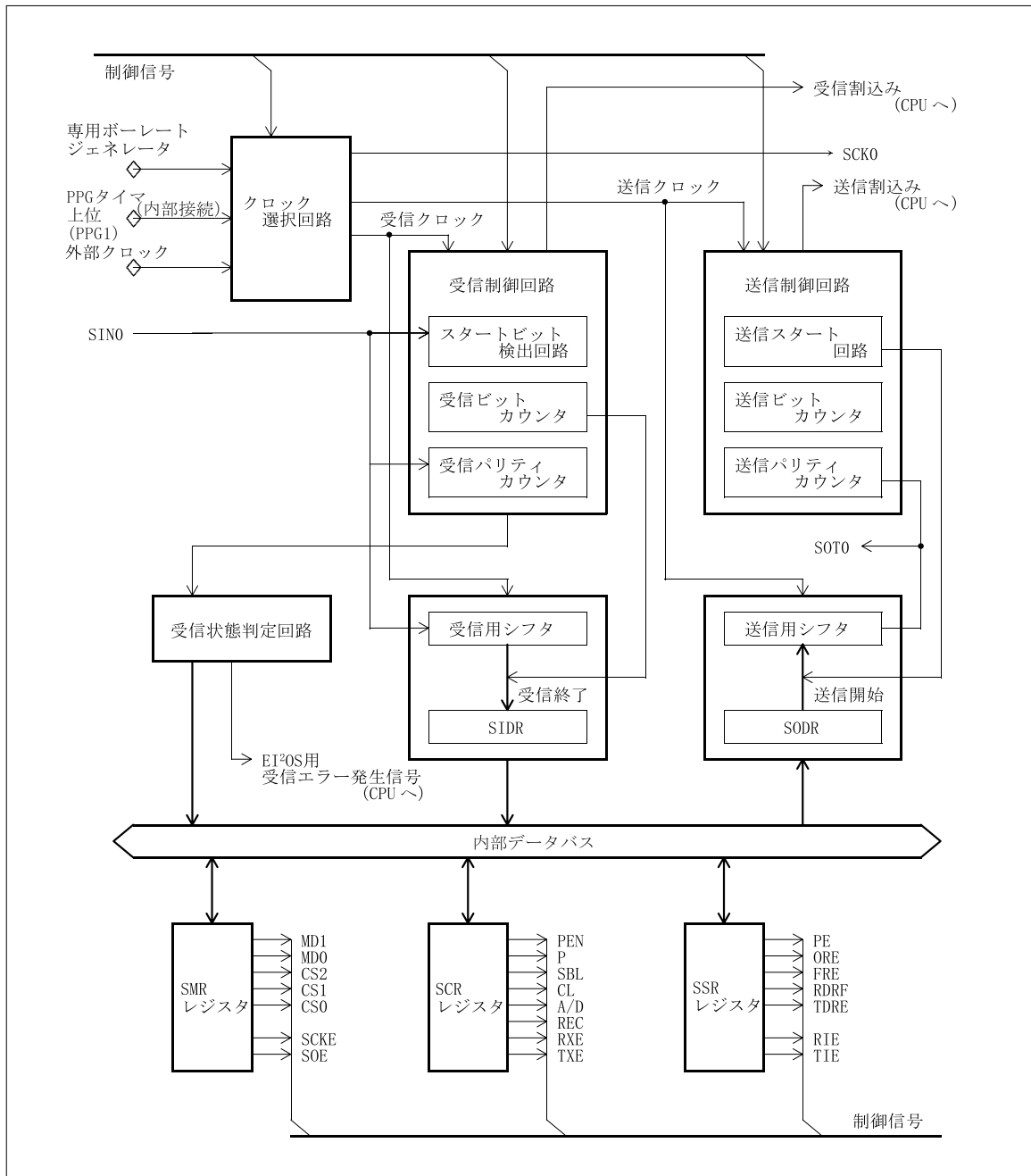


図 19.2-1 UARTのブロックダイアグラム

19.3 UARTのレジスタ

UARTのレジスタには、次の5種類があります。

- ・シリアルモードレジスタ(SMR)
- ・シリアルコントロールレジスタ(SCR)
- ・シリアルインプットデータレジスタ(SIDR)/シリアルアウトプットデータレジスタ(SODR)
- ・シリアルステータスレジスタ(SSR)
- ・通信プリスケールコントロールレジスタ(CDCR)

UARTのレジスタ

15	8	7	0	
—			CDCR	(R/W)
SCR			SMR	(R/W)
SSR			SIDR (R) / SODR (W)	(R/W)

シリアルモードレジスタ

	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス: 000020H 000024H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	SMR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

シリアルコントロールレジスタ

	15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス: 000021H 000025H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	SCR
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	

シリアルインプットデータレジスタ シリアルアウトプットデータレジスタ

	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス: 000022H 000026H	D7	D6	D5	D4	D3	D2	D1	D0	SIDR(read) SODR(write)
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

シリアルステータスレジスタ

	15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス: 000023H 000027H	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	SSR
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(—)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(1)	(—)	(0)	(0)	

通信プリスケールコントロールレジスタ

	7	6	5	4	3	2	1	0	↔ ビットNo.
アドレス: 000028H 00002AH	MD	—	—	—	DIV3	DIV2	DIV1	DIV0	CDCR
リード/ライト ⇒	(R/W)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(—)	(—)	(—)	(1)	(1)	(1)	(1)	

図 19.3-1 UARTのレジスタ

19.3.1 シリアルモードレジスタ(SMR)

シリアルモードレジスタ(SMR)は、UARTの動作モードを指定します。

動作モードの設定は動作停止中に行い、動作中にこのレジスタへの書込みは行わないでください。

シリアルモードレジスタ(SMR)

シリアルモードレジスタ									↔ビットNo.
	7	6	5	4	3	2	1	0	
アドレス: 000020 _H 000024 _H	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	SMR
リード/ライト ⇒ 初期値 ⇒	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

図 19.3-2 シリアルモードレジスタ(SMR)の構成

【ビット7, 6】MD1, MD0(MoDe select)

UARTの動作モードを選択します。

表 19.3-1 MD0,1(動作モード選択ビット)

モード	MD1	MD0	動作モード
0	0	0	非同期(調歩同期)ノーマルモード[初期値]
1	0	1	非同期(調歩同期)マルチプロセッサモード
2	1	0	CLK同期モード
-	1	1	設定禁止

< 注意事項 >

モード1のCLK非同期モード(マルチプロセッサ)とは、1台のホストCPUに数台のスレーブCPUが接続されるモードです。

本リソースでは、受信データのデータ形式を判別できません。従って、マルチプロセッサモードのマスタのみをサポートします。

また、パリティチェック機能は使用できませんのでシリアルコントロールレジスタ(SCR)内のPENビットは"0"に設定してください。

【ビット5～3】CS2,CS1,CS0(Clock Select)

ボーレートクロックソースを選択します。

専用ボーレートジェネレータを選択した場合には、同時にボーレートも決定されます。

表 19.3-2 CS0～2(ボーレートクロックソース選択ビット)

CS2	CS1	CS0	クロック入力
000 _B ～100 _B			専用ボーレートジェネレータ
1	0	1	予約
1	1	0	内部タイマ(16ビットリロードタイマ0)
1	1	1	外部クロック

< 注意事項 >

内部タイマを選択した場合、MB90570シリーズでは、16ビットリロードタイマ0の出力が選択されます。

【ビット2】予約ビット

必ず"0"に設定してください。

【ビット1】SCKE(SCLK Enable)

CLK同期モード(モード2)で通信する場合、SCK0端子をクロック入力端子にするか、クロック出力端子として使うかを指定します。

CLK非同期モード時または外部クロックモード時では"0"に設定してください。汎用ポート端子として機能します。

表 19.3-3 SCKE (SCLK Enable)ビットの機能

SCKE	機能
0	クロック入力端子として機能します。[初期値]
1	クロック出力端子として機能します。

< 注意事項 >

クロック入力端子として使うには、外部クロックソースが選択されている必要があります。

【ビット0】SOE(Serial Output Enable)

汎用I/Oポート端子と兼用されている外部端子(S0T0)をシリアル出力端子として使うか、I/Oポート端子として使うかを指定します。

表 19.3-4 SOE (Serial Output Enable) ビットの機能

SOE	機能
0	汎用I/Oポート端子として機能します。[初期値]
1	シリアルデータ出力端子(S0T0)として機能します。

19.3.2 シリアルコントロールレジスタ(SCR)

シリアルコントロールレジスタ(SCR)は、シリアル通信を行う場合の転送プロトコルを制御します。

シリアルコントロールレジスタ(SCR)

シリアルコントロールレジスタ									
	15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス:000021 _H 000025 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	SCR
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(R/W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	

図 19.3-3 シリアルコントロールレジスタ(SCR)の構成

【ビット15】PEN(Parity ENnable)

シリアル通信において、パリティを付加してデータ通信を行うかどうかを指定します。

表 19.3-5 PEN (Parity ENable)ピットの機能

PEN	機能
0	パリティなし [初期値]
1	パリティあり

< 注意事項 >

パリティを付加できるのは、非同期（調歩同期）通信モードのノーマルモード（モード0）のみです。マルチプロセッサモード（モード1）およびCLK同期通信（モード2）では、パリティを付加することはできません。

【ビット14】P(Parity)

パリティを付加してデータ通信を行うとき、偶数／奇数パリティを指定します。

表 19.3-6 P(偶数／奇数パリティ指定ビット)

P	機能
0	偶数パリティ [初期値]
1	奇数パリティ

【ビット13】SBL(Stop Bit Length)

SBLビットは、非同期（調歩同期）通信を行うときのフレームエンドマークである、ストップビットのビット長を指定します。

表 19.3-7 SBL(ストップビット長指定ビット)

SBL	機能
0	1ストップビット [初期値]
1	2ストップビット

【ビット12】CL (Character Length)

送受信する1フレームのデータ長を指定します。

表 19.3-8 CL(送受信データ長指定ビット)

CL	機能
0	7ビットデータ [初期値]
1	8ビットデータ

< 注意事項 >

7ビットデータを扱えるのは、非同期（調歩同期）通信のうちのノーマルモード（モード0）のみです。マルチプロセッサモード（モード1）およびCLK同期通信（モード2）では、8ビットデータとしてください。

【ビット11】A/D (Address/Data)

非同期（調歩同期）通信のマルチプロセッサモード（モード1）において、送受信するフレームのデータ形式を指定します。

表 19.3-9 A/D (Address/Data) ビットの機能

A/D	機能
0	データフレーム [初期値]
1	アドレスフレーム

【ビット10】REC (Receiver Error Clear)

SSRレジスタのエラーフラグ（PE, ORE, FRE）をクリアします。
"1"書込みは無効であり、読出し値は常に"1"になります。

【ビット9】RXE (Receiver Enable)

UARTの受信動作を制御します。

表 19.3-10 RXE (Receiver Enable) ビット

RXE	機能
0	受信動作を禁止します。 [初期値]
1	受信動作を許可します。

< 注意事項 >

受信中（受信シフトレジスタにデータが入力されているとき）に受信動作を禁止した場合には、そのフレームの受信を完了し、受信データバッファ(SIDR)に受信データをストアしたところで受信動作を停止します。

【ビット8】TXE (Transmitter Enable)

UARTの送信動作を制御します。

表 19.3-11 送信動作制御ビット(TXE)

TXE	機能
0	送信動作を禁止します。[初期値]
1	送信動作を許可します。

< 注意事項 >

送信中（送信レジスタからデータが出力されているとき）に送信動作を禁止した場合は，送信データバッファ(SODR)にデータがなくなった後に送信動作を停止します。

"0"の書込みは，SODRにデータを書込み後，一定期間を置いてから行ってください。

一定期間とは，クロック非同期転送モード時はボーレートの1/16時間です。クロック同期転送モード時は，ボーレートの時間です。

19.3.3 シリアルインプットデータレジスタ(SIDR)/シリアルアウトプットデータレジスタ(SODR)

シリアルインプットデータレジスタ(SIDR)は,シリアルデータ受信用レジスタで,シリアルアウトプットデータレジスタ(SODR)は,シリアルデータ送信用のレジスタです。

SIDR/SODRレジスタは,同一アドレスに配置されています。

シリアルインプットデータレジスタ(SIDR)/シリアルアウトプットデータレジスタ(SODR)

シリアルインプットデータレジスタ シリアルアウトプットデータレジスタ										
		7	6	5	4	3	2	1	0	← ビットNo.
アドレス: 000022H	D7	D6	D5	D4	D3	D2	D1	D0	SIDR(read) SODR(write)
000026H									
リード/ライト ⇒		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒		(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 19.3-4 シリアルインプットデータレジスタ(SIDR)/シリアルアウトプットデータレジスタ(SODR)の構成

データ長が7bitの場合は,上位1bit(D7)は無効データとなります。SODRレジスタへの書込みは,シリアルステータスレジスタ(SSR)内のTDREビットが"1"の場合に書込みを行ってください。

< 注意事項 >

このアドレスへのライトはSODRレジスタへの書き込みを,リードはSIDRレジスタの読み出しを意味します。

19.3.4 シリアルステータスレジスタ(SSR)

シリアルステータスレジスタ(SSR)は，UARTの動作状態を表すフラグで構成されています。

シリアルステータスレジスタ(SSR)

シリアルステータスレジスタ									
	15	14	13	12	11	10	9	8	↔ ビットNo.
アドレス: 000023 _H 000027 _H	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	SSR
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(-)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(1)	(-)	(0)	(0)	

図 19.3-5 シリアルステータスレジスタ(SSR)の構成

【ビット15】PE (Parity Error)

受信時にパリティエラーが発生したときにセットされる，割込み要求フラグです。一度セットされたフラグをクリアするには，シリアルコントロールレジスタ(SCR)内のRECビット(bit10)に"0"を書込みます。

本ビットがセットされた場合には，シリアルインプットデータレジスタ(SIDR)内のデータは無効データとなります。

表 19.3-12 PE (Parity Error)ビットの機能

PE	機能
0	パリティエラーなし [初期値]
1	パリティエラー発生

【ビット14】ORE (Over Run Error)

受信時にオーバランエラーが発生したときにセットされる，割込み要求フラグです。一度セットされたフラグをクリアするには，シリアルコントロールレジスタ(SCR)内のRECビット(bit 10)に"0"を書込みます。

本ビットがセットされた場合には，シリアルインプットデータレジスタ(SIDR)レジスタ内のデータは無効データとなります。

表 19.3-13 ORE (Over Run Error) の機能

ORE	機能
0	オーバランエラーなし [初期値]
1	オーバランエラー発生

【ビット13】FRE (FRaming Error)

受信時にフレーミングエラーが発生したときにセットされる，割り込み要求フラグです。一度セットされたフラグをクリアするには，シリアルコントロールレジスタ(SCR)内のRECビット(bit10)に"0"を書込みます。本ビットがセットされた場合には，シリアルインプットデータレジスタ(SIDR)内のデータは無効データとなります。

表 19.3-14 FRE (FRaming Error) の機能

FRE	機能
0	フレーミングエラーなし [初期値]
1	フレーミングエラー発生

【ビット12】RDRF (Receiver Data Register Full)

シリアルインプットデータレジスタ(SIDR)内に受信データがあることを示す，割り込み要求フラグです。SIDRレジスタに受信データがロードされるとセットされ，SIDRレジスタを読出すと自動的にクリアされます。

表 19.3-15 RDRF (Receiver Data Register Full) の機能

SIDR	機能
0	受信データなし [初期値]
1	受信データあり

【ビット11】TDRE (Transmitter Data Register Empty)

シリアルアウトプットレジスタ(SODR)内に送信データを書き込めることを示す，割り込み要求フラグです。SODRレジスタに送信データを書き込むとクリアされます。書き込んだデータが送信用シフトにロードされて転送が開始されると再びセットされ，次の送信データデータを書き込めることを表します。

表 19.3-16 TDRE (Transmitter Data Register Empty) の機能

TDRE	設定
0	送信データの書き込み禁止
1	送信データの書き込み許可 [初期値]

【ビット9】RIE (Receiver Interrupt Enable)

受信割り込みを制御します。

表 19.3-17 RIE (Receiver Interrupt Enable) の機能

RIE	機能
0	割り込みを禁止 [初期値]
1	割り込みを許可

< 注意事項 >

受信割り込み要因は，エラーフラグビット(PE,ORE,FRE)によるエラー発生その他，RDRFビットによる正常受信があります。

【ビット8】TIE (Transmitter Interrupt Enable)

送信割込みを制御します。

表 19.3-18 TIE (Transmitter Interrupt Enable) の機能

TIE	機能
0	割込みを禁止します。[初期値]
1	割込みを許可します。

< 注意事項 >

送信割込要因は、TDREビットによる送信要求があります。

19.3.5 通信プリスケアラコントロールレジスタ(CDCR)

通信プリスケアラコントロールレジスタ(CDCR)は、マシンのクロックの分周を制御するレジスタです。

通信プリスケアラコントロールレジスタ(CDCR)

UARTの動作クロックは、マシンのクロックを分周することにより得られます。この通信プリスケアラによって、種々のマシンのクロックに対して一定のボーレートが得られるように設計されています。

本通信プリスケアラ出力は、拡張I/O シリアルインタフェースの動作クロックにも使われています。

CDCRのビット構成を以下に示します。

通信プリスケアラコントロールレジスタ								
								↔ ビットNo.
アドレス:000028H 00002AH	7	6	5	4	3	2	1	0
	MD	—	—	—	DIV3	DIV2	DIV1	DIV0
リード/ライト ⇒	(R/W)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒	(0)	(—)	(—)	(—)	(1)	(1)	(1)	(1)

図 19.3-6 通信プリスケアラコントロールレジスタ(CDCR)の構成

< 注意事項 >

CDCR0 (アドレス : 000028H) は、UARTの0ch と、I/O 拡張シリアルインタフェース2,3,4ch用の通信プリスケアラレジスタです。したがって、UARTの0ch とI/O 拡張シリアルインタフェース2,3,4ch は、マシンのクロックの分周比が共通になります。それぞれ別々の分周比に設定することは出来ませんのでご注意ください。なお、CDCR1 (アドレス : 00002AH) はUARTの1ch用です。

【ビット7】MD(Machine clock divide mode select)

通信プリスケアラの動作許可ビットです。

表 19.3-19 MD(Machine clock divide mode select)ビットの機能

CDCR	機能
0	通信プリスケアラは停止します。 [初期値]
1	通信プリスケアラは動作します。

【ビット3, 2, 1, 0】DIV3~0(DIVide 3~0)

マシクロックの分周比を決定します。

表 19.3-20 DIV3~0(DIVide 3~0) ビットの機能

DIV3	DIV2	DIV1	DIV0	分周比(div)
1	1	1	1	設定禁止 [初期値]
1	1	1	0	2分周
1	1	0	1	3分周
1	1	0	0	4分周
1	0	1	1	5分周
1	0	1	0	6分周
1	0	0	1	7分周
1	0	0	0	8分周

(注意)

- ・実際の使用時は, "1111"以外に設定してください。
- ・分周比を変えた場合はクロックの安定時間として2周期分の時間を置いてから通信を行ってください。

通信プリスケアラレジスタの設定

使用するマシクロック によって, 通信プリスケアラレジスタの設定を以下のようにしてください。

表 19.3-21 通信プリスケアラレジスタの設定

マシクロック	Div	DIV3	DIV2	DIV1	DIV0	÷ div
4MHz	4	1	1	0	0	1MHz
6MHz	6	1	0	1	0	
8MHz	8	1	0	0	0	
6MHz	3	1	1	0	1	2MHz
8MHz	4	1	1	0	0	
10MHz	5	1	0	1	1	
12MHz	6	1	0	1	0	
14MHz	7	1	0	0	1	
16MHz	8	1	0	0	0	
8MHz	2	1	1	1	0	4MHz
12MHz	3	1	1	0	1	
16MHz	4	1	1	0	0	

上記以外のマシクロックとdivの設定で使用する場合, ÷ divは, 最大4.25Mを越えないようにしてください。

19.4 UARTのボーレート

UARTのクロックは、次の3種類から選択できます。

- ・専用ボーレートジェネレータ
- ・内部タイマ
- ・外部クロック

専用ボーレートジェネレータ

表 19.4-1および表 19.4-2に専用ボーレートジェネレータを選択した場合のボーレートを示します。ただし、(マシンのクロック)=16MHz, div(マシンのクロック分周比)=8で算出しています。また、表 19.4-3に通信プリスケアラの設定値を示します。

表 19.4-1 ボーレート(非同期時)

CS2	CS1	CS0	非同期時 (歩調同期)	算出式
0	0	0	9615 bps	$(\div \text{div}) / (8 \times 13 \times 2)$
0	0	1	4808 bps	$(\div \text{div}) / (8 \times 13 \times 2^2)$
0	1	0	2404 bps	$(\div \text{div}) / (8 \times 13 \times 2^3)$
0	1	1	1202 bps	$(\div \text{div}) / (8 \times 13 \times 2^4)$
1	0	0	31250 bps	$(\div \text{div}) / 2^6$

表 19.4-2 ボーレート(CLK同期時)

CS2	CS1	CS0	CLK同期時	算出式
0	0	0	1 M bps	$(\div \text{div}) / 2$
0	0	1	500K bps	$(\div \text{div}) / 2^2$
0	1	0	250K bps	$(\div \text{div}) / 2^3$
0	1	1	125K bps	$(\div \text{div}) / 2^4$
1	0	0	62.5K bps	$(\div \text{div}) / 2^5$

表 19.4-3 通信プリスケアラ設定値

MD	DIV3	DIV2	DIV1	DIV0	div	推奨マシンのクロック
1	1	1	0	1	3	6MHz
1	1	1	0	0	4	8MHz
1	1	0	1	1	5	10MHz
1	1	0	1	0	6	12MHz
1	1	0	0	0	8	16MHz

内部タイマ

シリアルモードレジスタ(SMR)内のCS2～0ビットを"110"に設定して内部タイマを選択した場合は、16ビットタイマ(タイマ0)をリロードモードで動作させます。この場合のボーレートの算出式は、次のとおりです。

非同期(調歩同期) $(\div N) / (16 \times 2 \times (n + 1))$

CLK同期 $(\div N) / (2 \times (n + 1))$

: マシニングロック

N: タイマのカウントクロックソース

n: タイマのリロード値

マシニングロックを 7.3728MHzとしたときのボーレートとリロード値(10進数)の関係を、表 19.4-4に示します。

表 19.4-4 ボーレートとリロード値(非同期)

ボーレート	リロード値	
	N = 2 ¹ (マシニングロックの2分周)	N = 2 ³ (マシニングロックの8分周)
38400	2	-
19200	5	-
9600	11	2
4800	23	5
2400	47	11
1200	95	23
600	191	47
300	383	95

ボーレートクロックソースとして内部タイマ(16ビットリロードタイマ0)を選択した場合には、16ビットタイマ0の出力 T00 は本コントローラ内部で既に接続されています。従って、16ビットタイマ0の外部端子 T00 から本UARTの外部クロック入力端子 SCK0へ外部で接続する必要はありません。また、タイマ0の出力端子を他に使用していなければI/Oポート端子として使用することができます。

外部クロック

SMRレジスタのCS2～0ビットを"111"に設定して外部クロックを選択した場合のボーレートは、外部クロックの周波数をfとすると、次のようになります。

非同期(調歩同期): $f / 16$

CLK同期 : f

ただし、fは最大1MHzまでです。

19.5 UARTの動作

UARTは、非同期モードとCLK同期モードの2種類の動作モードを持ち、シリアルモードレジスタ(SMR)、シリアルコントロールレジスタ(SCR)に値を設定することによりモードを切換えることができます。

UARTの動作

表 19.5-1 UARTの動作モード

モード	パリティ	データ長	動作モード	ストップビット長
0	有/無	7	非同期(調歩同期)ノーマルモード	1bitまたは2bit
	有/無	8		
1	無	8+1	非同期(調歩同期)マルチプロセッサ	無
2	無	8	CLK同期モード	

< 注意事項 >

非同期(調歩同期)モードでのストップビット長については、送信動作のみ指定が可能です。受信動作については常に1ビット長となります。上記モード以外では動作しませんので、設定しないでください。

19.5.1 非同期（調歩同期）モード

UARTを動作モード0（ノーマルモード）、または動作モード1（マルチプロセッサモード）で使用するときの転送方式は非同期となります。なお、UARTは、NRZ(Non Return to Zero)形式のデータのみを扱います。

転送データフォーマット

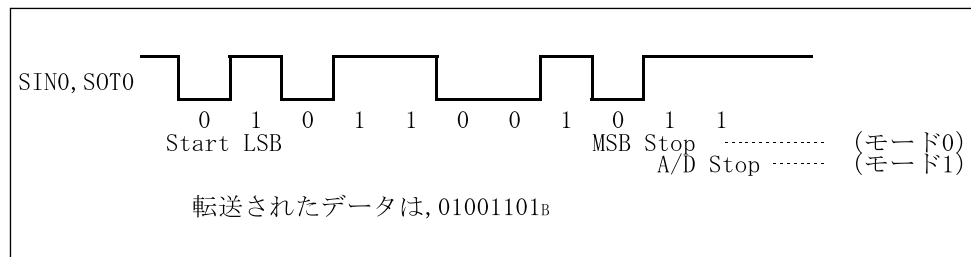


図 19.5-1 転送データフォーマット（モード0, 1）

図 19.5-1に示すように、転送データは必ずスタートビット（"L"レベルデータ）より始まり、LSBファーストで指定されたデータビット長転送が行われ、ストップビット（"H"レベルデータ）で終了します。外部クロックを選択している場合は、常にクロックを入力してください。

ノーマルモード（モード0）ではデータ長を7ビットまたは8ビットに設定することができますが、マルチプロセッサモード（モード1）では8ビットでなければなりません。また、マルチプロセッサモードではパリティを付加することはできません。そのかわり、A/D ビットが必ず付加されます。

受信動作

シリアルコントロールレジスタ(SCR)内のRXEビットが"1"ならば、常に受信動作が行われています。スタートビットを検出すると、SCRレジスタで決められたデータフォーマットに従い1フレームデータの受信が行われます。1フレームの受信が終わると、エラーが発生した場合にはエラーフラグのセットが行われた後、シリアルステータスレジスタ(SSR)内のRDRFフラグがセットされます。この場合同じSSRレジスタのRIEビットが"1"にセットされていればCPUに対して受信割込みが発生します。SSRレジスタの各フラグを調べ、正常受信ならシリアルインプットデータレジスタ(SIDR)をリードして、エラーが発生していれば必要なエラー処理を行うようにしてください。RDRFフラグは、SIDRレジスタをリードするとクリアされます。

送信動作

シリアルステータスレジスタ(SSR)内のTDREフラグが"1"の場合、シリアルアウトプットデータレジスタ(SODR)に送信データを書き込みます。ここで、シリアルコントロールレジスタ(SCR)内の TXEビットが"1"なら送信が行われます。

SODRレジスタにセットしたデータが送信用シフトレジスタにロードされて送信が開始されるとSSRレジスタ内のTDREフラグが再びセットされ、次の送信データをセットできるようになります。この場合同じSSRレジスタ内のTIEビットが"1"にセットされていればCPUに対して送信割込みが発生して、SODRレジスタに送信データをセットするように要求します。

SSRレジスタ内のTDREフラグは、SODRレジスタにデータをセットすると一旦クリアされます。

19.5.2 CLK同期モード

UARTを動作モード2(ノーマルモード)で使用する際の転送方式はクロック同期となります。なお、UARTは、NRZ(Non Return to Zero)形式のデータのみを扱います。

CLK同期モードの転送データフォーマット

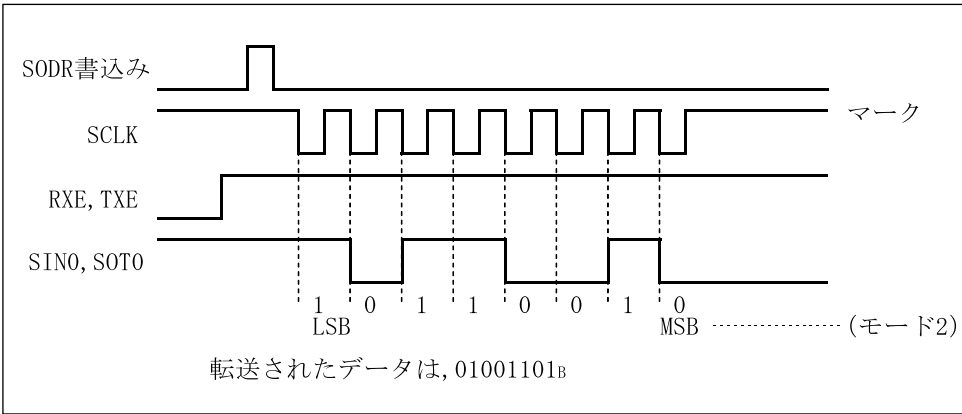


図 19.5-2 転送データフォーマット (モード2)

内部クロック (専用ボーレートジェネレータまたは内部タイマ) を選択している場合は、データを送信するとデータ受信用同期クロックが自動的に生成されます。

外部クロックを選択している場合は、送信側UARTの送信用データバッファSODRレジスタにデータがあること (SSRレジスタ内のTDREフラグが"0") を確かめた後、正確に1バイト分のクロックを供給する必要があります。また、送信開始前と終了後は、必ずマークレベル"H"にしてください。

データは8ビットのみとなり、パリティを付加することはできません。また、スタート/ストップビットがないのでオーバーランエラー以外のエラー検出は行われません。

CLK同期モードを使用する場合の各制御レジスタの設定値

CLK同期モードを使用する場合の、各制御レジスタの設定値を示します。

表 19.5-2 CLK同期モードを使用する場合の、各制御レジスタの設定値

レジスタ名	ビット名	設定
SMR レジスタ	MD1, MD0	"10"
	CS2, CS1, CS0	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タイマの場合"1", 外部クロックの場合"0"
	SOE	送信を行う場合"1", 受信のみの場合"0"
SCR レジスタ	PEN	"0"
	P, SBL, A/D	これらのビットは意味を持ちません
	CL	"1"
	REC	"0" (初期化するため)
	RXE, TXE	少なくとも、どちらか一方を"1"
SSR レジスタ	R1E	割込みを使用する場合"1", 割込みを使用しない場合"0"
	T1E	"0"

CLK同期モードの通信開始

シリアルアウトプットデータレジスタ(SODR)への書き込みによって通信を開始します。受信のみの場合でも、必ず仮の送信データをSODRレジスタに書き込む必要があります。

CLK同期モードの通信終了

シリアルステータスレジスタ(SSR)内のRDRFフラグが"1"に変化したことにより確認できます。

SSRレジスタのOREビットによって、通信が正常に行われたかを判断してください。

19.6 UARTのフラグと割込み発生要因

UARTには、5個のフラグ(PE/ORE/FRE/RDRF/TDRE)と2種類の割込み要因(受信用/送信用)があります。

UARTのフラグ (PE/ORE/FRE/RDRF/TDRE)

PE (パリティエラー) /ORE (オーバランエラー) /FRE (フレーミングエラー)

受信時エラーが発生した時にセットされ、シリアルコントロールレジスタ (SCR)内のRECビットに"0"を書込むとクリアされます。

RDRF

受信データがシリアルインプットデータレジスタ(SIDR)にロードされるとセットされ、SIDRレジスタを読み出すことでクリアされます。ただし、モード1ではパリティ検出機能、モード2ではパリティ検出機能とフレーミングエラー検出機能はありません。

TDRE

シリアルアウトプットデータレジスタ(SODR)が空になり書込み可能な状態になるとセットされ、SODRレジスタへ書込むとクリアされます。

UARTの割込み要因

UARTの割込み要因は、受信用のものと送信用のものがあります。受信時は、シリアルステータスレジスタ(SSR)内のPE/ORE/FRE/RDRFフラグにより割込みを要求します。送信時は、SSRレジスタ内のTDREフラグにより割込みを要求します。

各動作モード時の割込みとフラグのセットタイミングは「19.6.1 UARTの割込みとフラグのセットタイミング」に示します。参照してください。

19.6.1 UARTの割り込みとフラグのセットタイミング

各動作モード時の割り込みとフラグのセットタイミングを示します。

UARTの割り込みとフラグのセットタイミング

モード0の受信動作時

シリアルステータスレジスタ(SSR)内のPE,ORE,FRE,RDRFフラグは、受信転送が終了し最後のストップビットを検出する場合にセットされ、CPUへの割り込み要求が発生します。PE,ORE,FREフラグがアクティブな場合には、シリアルインプットデータレジスタ(SIDR)のデータは無効データとなります。

図 19.6-1に、PE,ORE,FRE,RDRFフラグのセットタイミング(モード0)を示します。

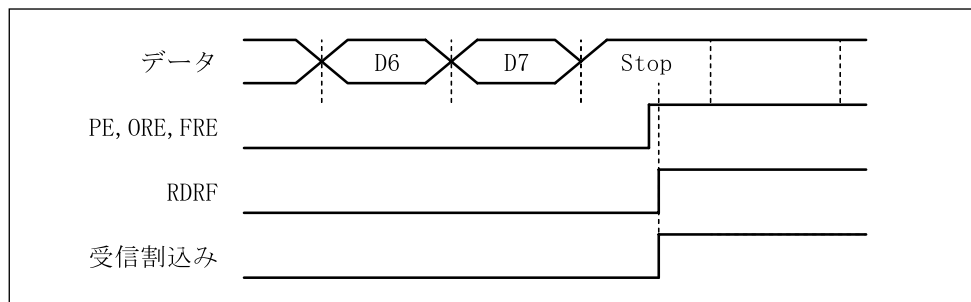


図 19.6-1 PE,ORE,FRE,RDRFフラグのセットタイミング(モード0)

モード1の受信動作時

シリアルステータスレジスタ(SSR)内のORE,FRE,RDRFフラグは受信転送が終了し、最後のストップビットを検出する場合にフラグがセットされ、CPUへの割り込み要求が発生します。また、受信可能なデータ長は8bitのため、最後の9bit目のアドレス/データを示すデータは無効データとなります。ORE,FREフラグがアクティブな場合には、シリアルインプットデータレジスタ(SIDR)のデータは無効データとなります。

図 19.6-2に、ORE,FRE,RDRFフラグのセットタイミング(モード1)を示します。

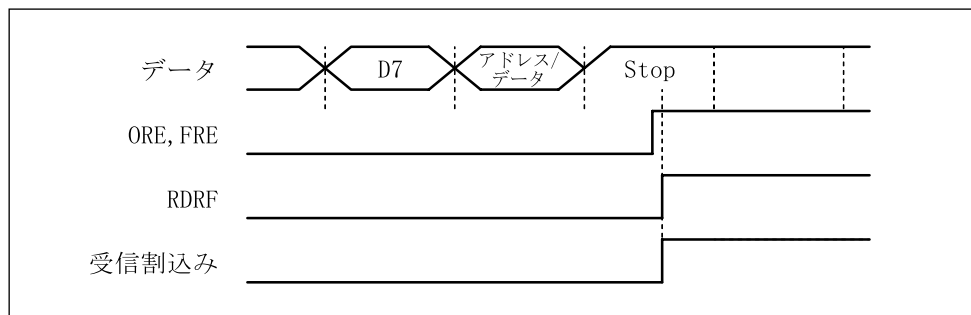


図 19.6-2 ORE,FRE,RDRFフラグのセットタイミング(モード1)

モード2の受信動作時

シリアルステータスレジスタ(SSR)内のORE,RDRFフラグは、受信転送が終了し、最後のデータ(D7)を検出する場合にセットされ、CPUへの割込み要求が発生します。OREがアクティブな場合には、シリアルインプットデータレジスタ(SIDR)のデータは無効データとなります。

図 19.6-3に、ORE、RDRFフラグのセットタイミング(モード2)を示します。

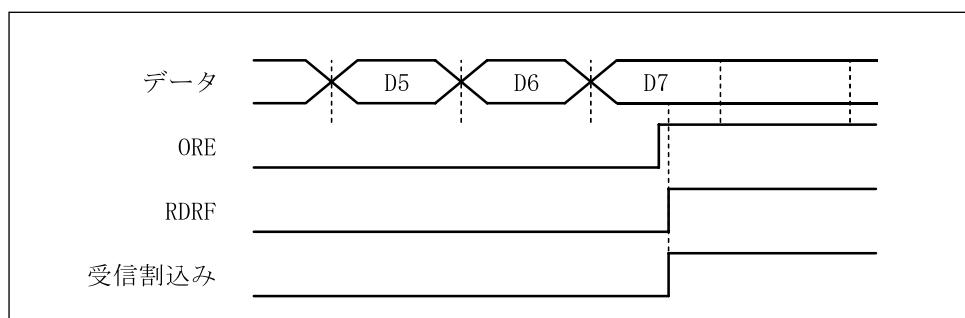


図 19.6-3 ORE, RDRF フラグのセットタイミング(モード2)

モード0, モード1, モード2の送信動作時

シリアルステータスレジスタ(SSR)内のTDREフラグは、シリアルアウトプットデータレジスタ(SODR)へ送信データが書込まれるとクリアされます。また、SODRレジスタ値が内部のシフトレジスタに転送されるとSODRレジスタが書込み可能な状態になるため、TDREフラグがセットされます。このフラグがセットされると、CPUへの割り込み要求が発生します。送信動作中にシリアルコントロールレジスタ(SCR)内のTXEビットに"0" (モード2の場合はRXEも含む)を書込むと、SSRレジスタ内のTDREフラグが"1"となり、送信用のシフトが停止してからUARTの送信動作を禁止します。送信動作中にSCRレジスタのTXEに"0" (モード2の場合はRXEも含む)を書込んだ後、送信が停止する前にSODRレジスタへ書込まれたデータは送信されます。

図 19.6-4に、TDREフラグのセットタイミング(モード0,1)を示します。

また、図 19.6-5に、TDREフラグのセットタイミング(モード2)を示します。

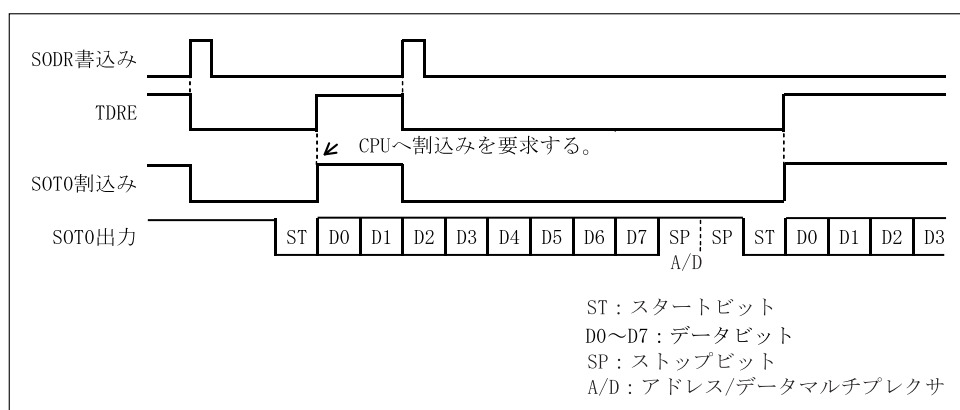


図 19.6-4 TDREフラグのセットタイミング(モード0,1)

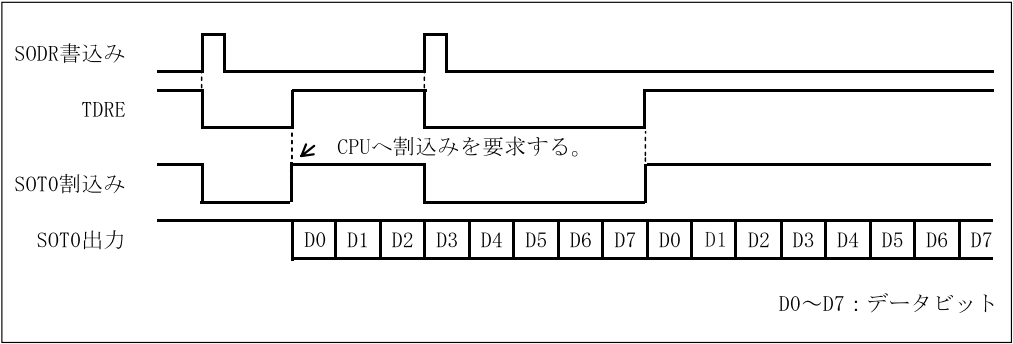


図 19.6-5 TDREフラグのセットタイミング(モード2)

19.7 UARTの応用例と使用上の注意

UARTの応用例として、モード1使用時のシステム構築例と通信フローチャートを示します。

UARTの応用例(モード1使用時のシステム構築例)

モード1は、1台のホストCPUに数台のスレーブCPUが接続されるような場合に使用されます。このリソースでは、ホスト側の通信インタフェースのみサポートします。図 19.7-1に、モード1使用時のシステム構築例を示します。

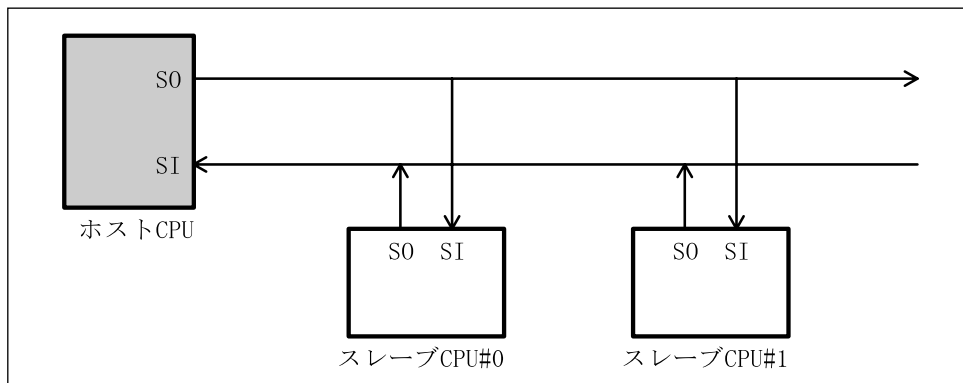


図 19.7-1 モード1使用時のシステム構築例

UARTの通信フローチャート

図 19.7-2に示すように、通信は、ホストCPUがアドレスデータを転送することによって始まります。アドレスデータとは、シリアルコントロールレジスタ(SCR)内のA/Dビットが"1"の場合のデータで、それにより通信先となるスレーブCPUが選択され、ホストCPUとの通信が可能になります。通常データは、SCRレジスタ内のA/Dビットが"0"の場合のデータです。

このモードにおいては、パリティチェック機能は使用できませんのでSCRレジスタ内のPENビットは"0"としてください。

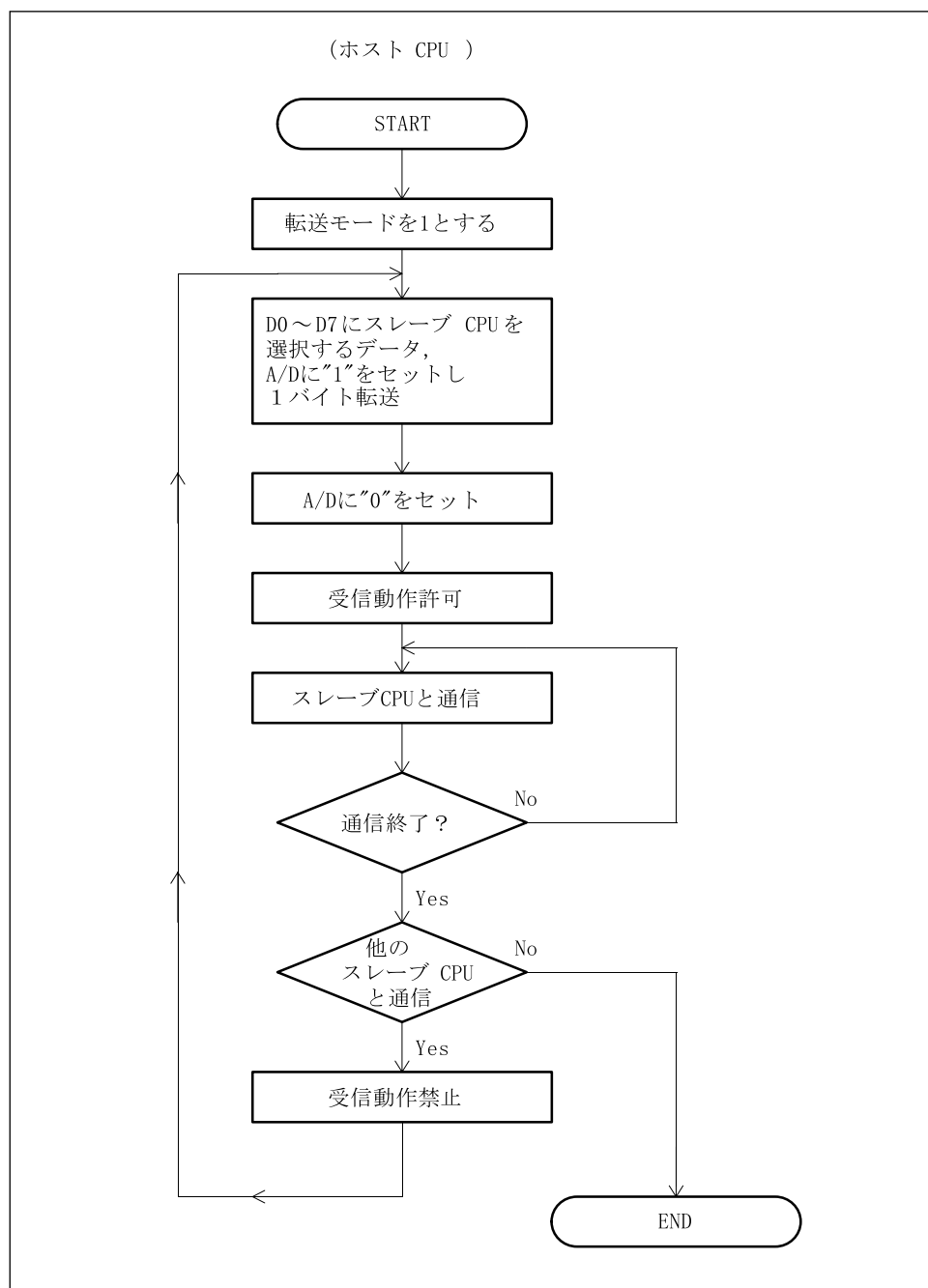


図 19.7-2 モード1使用時の通信フローチャート

拡張インテリジェントI/Oサービス(EI²OS)

EI²OSについては、「3.6 拡張インテリジェントI/Oサービス(EI²OS)」を参照してください。

UARTの使用上の注意

通信モードの設定は、動作停止中に行ってください。モード設定時の送受信データは保証されません。

第20章 I/O拡張シリアルインタフェース

この章では，I/O拡張シリアルインタフェースの機能と動作について説明します。

- 20.1 I/O拡張シリアルインタフェースの概要
- 20.2 I/O拡張シリアルインタフェースのレジスタ
- 20.3 I/O拡張シリアルインタフェースの動作

20.1 I/O拡張シリアルインタフェースの概要

I/O拡張シリアルインタフェースは、8bit × 3ch. 構成のクロック同期式によるデータ転送可能なシリアルI/Oインタフェースです。また、データ転送においてLSBファースト/MSBファーストの選択が可能です。

I/O拡張シリアルインタフェースの概要

I/O拡張シリアルインターフェイスの動作モードには、以下の2種類があります。

内部シフトクロックモード

内部クロック（通信プリスケアラ）に同期してデータを転送します。

外部シフトクロックモード

外部端子（SCK）から入力されるクロックに同期してデータを転送します。このモードで外部端子(SCK)を共有している汎用ポートを操作することにより、CPUの命令による転送動作も可能です。

本シリーズは、I/O拡張シリアルインタフェースを3ch内蔵しています。

I/O拡張シリアルインタフェースのブロックダイアグラム

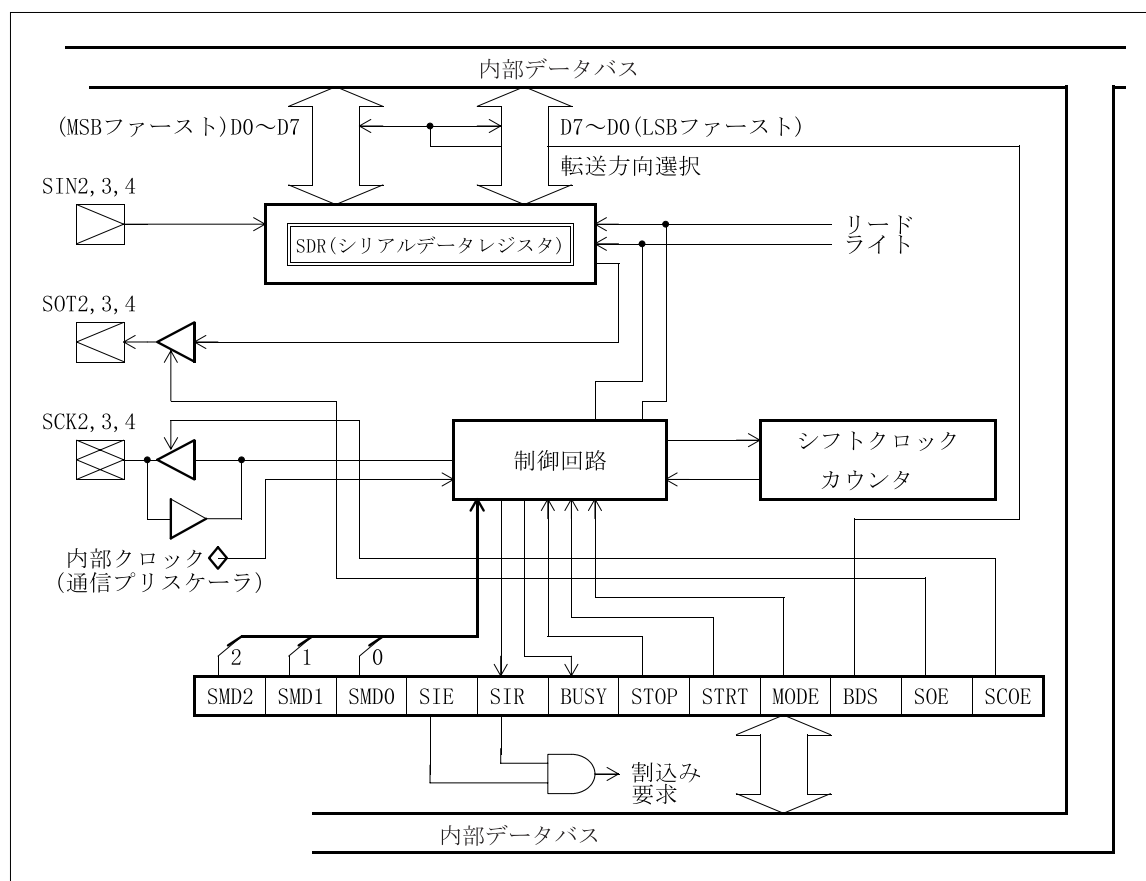


図 20.1-1 I/O拡張シリアルインタフェースブロックダイアグラム

20.2 I/O拡張シリアルインタフェースのレジスタ

I/O拡張シリアルインタフェースには、次の3種類のレジスタがあります。

- ・シリアルモードコントロールステータスレジスタ上位
- ・シリアルモードコントロールステータスレジスタ下位
- ・シリアルデータレジスタ

I/O拡張シリアルインタフェースのレジスタ

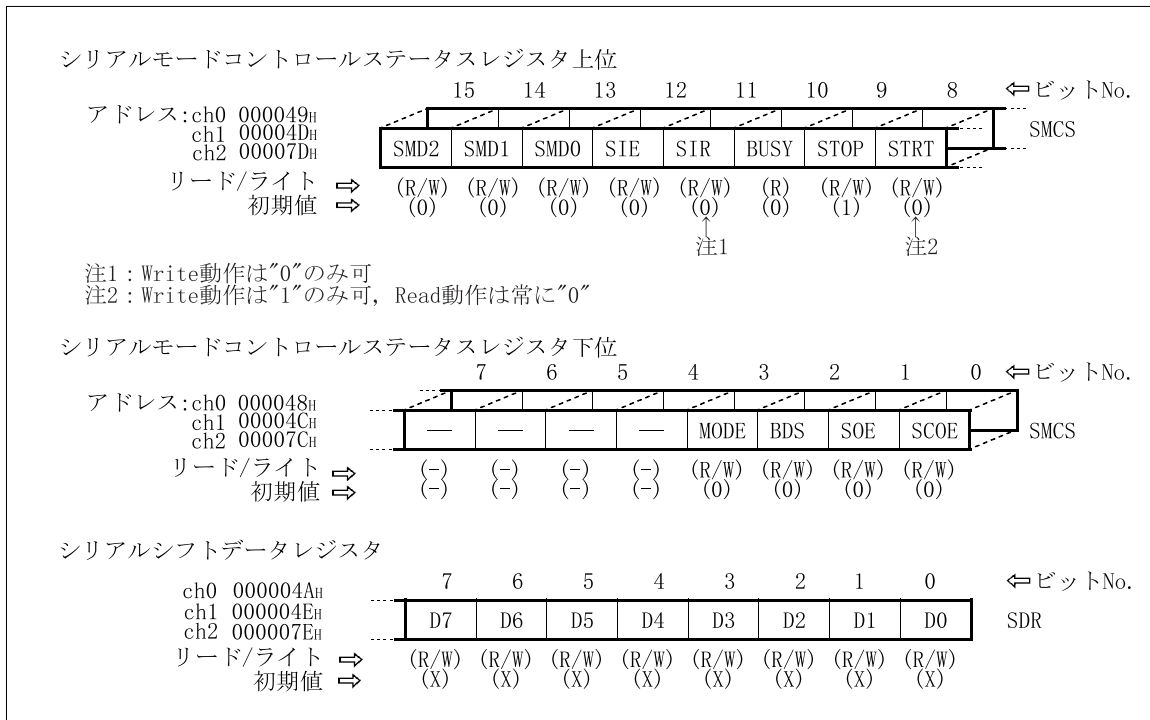
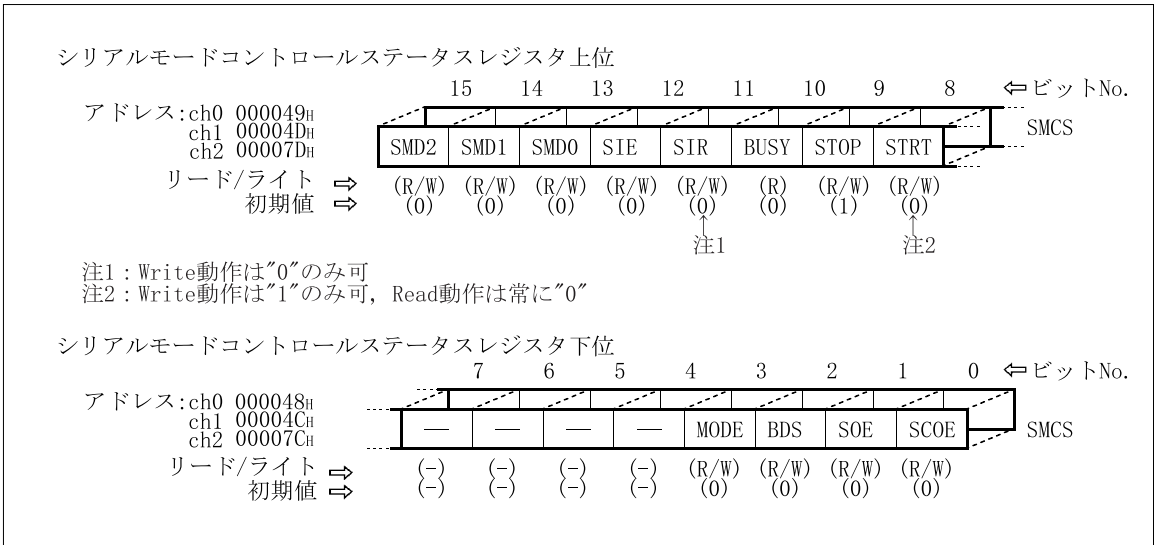


図 20.2-1 I/O拡張シリアルインタフェースのレジスタ

20.2.1 シリアルモードコントロールステータスレジスタ (SMCS)

シリアルモードコントロールステータスレジスタ (SMCS) は、シリアルI/Oの転送動作モードを制御するレジスタです。

シリアルモードコントロールステータスレジスタ (SMCS)



(注意) SIR(ビット11)のWrite動作は,"0"のみ可
STRT(ビット8)のWrite動作は,"1"のみ可, Read動作は常に"0"

図 20.2-2 シリアルモードコントロールステータスレジスタ (SMCS)

以下に各ビットの機能について説明します。

【ビット3】MODE

MODEビットは、停止状態からの起動条件を選択します。ただし、動作中の書換えは禁止します。リセットにより、"0"に初期化されます。
本ビットは、リード/ライトが可能です。
拡張インテリジェントI/Oサービスを起動する際には"1"にしておいてください。

表 20.2-1 MODE(起動条件選択ビット)の機能

MODE	機能
0	STRT=1にすることにより起動します。[初期値]
1	シリアルデータレジスタのリード/ライトにより起動します。

【ビット2】BDS(Bit Direction Select)

シリアルデータの入出力時に、最下位ビット側から先に転送するか (LSBファースト)、最上位ビット側から先に転送するか (MSBファースト) を、図 20.2-2 に示すように選択します。リセットにより、"0"に初期化されます。
本ビットは、リード/ライト可能です。

表 20.2-2 BDS(Bit Direction Select)ビットの機能

BDS	機能
0	LSBファースト [初期値]
1	MSBファースト

< 注意事項 >

転送方向ビット(BDS) は、SDRレジスタにデータを書き込む前に設定してください。

【ビット1】SOE(Serial Out Enable)

SOEビットは、シリアルI/O出力外部端子 (SOT2,3,4) の出力を表 20.2-3 に示すように制御します。リセットにより"0"に初期化されます。

本ビットは、リード/ライトが可能です。

表 20.2-3 SOE(Serial Out Enable)ビットの機能

SOE	機能
0	汎用ポート端子 [初期値]
1	シリアルデータ出力

【ビット0】SCOE(SCLK Output Enable)

SCOEビットは、シフトクロック用入出力外部端子 (SCK2,3,4) の出力を表 20.2-4で示すように制御します。

外部シフトクロックモードで命令ごとに転送を行うときは"0"に設定してください。リセットにより"0"に初期化されます。リード/ライトが可能です。

表 20.2-4 SCOE(SCLK Output Enable)ビットの機能

SCOE	機能
0	汎用ポート端子 [初期値]
1	シフトクロック出力端子

【ビット15,14,13】SMD2, SMD1, SMD0(Serial Shift Clock Mode)

SMD2, SMD1, SMD0ビットは、シリアルシフトクロックモードを表 20.2-5に示すように選択します。

リセットにより"000"に初期化されます。本ビットの転送中の書換えは禁止します。

シフトクロックは内部シフトクロック5種類と、外部シフトクロックが選択できます。

SMD2,1,0 = "110", "111"はリザーブですので設定しないでください。

クロック選択でSCOE = "0"とし、SCK1, 2端子を共有しているポートを操作することによって命令ごとにシフト動作させることも可能です。

表 20.2-5 SMD0~2(シリアルシフトクロックモード選択ビット)の機能

SMD2	SMD1	SMD0	分周値 A	=16MHz div=8	=8MHz Div=4	=4MHz div=4
0	0	0	2	1MHz	1MHz	500KHz
0	0	1	4	500KMHz	500KMHz	250KMHz
0	1	0	16	125KMHz	125KMHz	62.5KMHz
0	1	1	32	62.5KMHz	62.5KMHz	31.25KMHz
1	0	0	64	31.25MHz	31.25MHz	15.625MHz
1	0	1	1	外部シフトクロックモード		
1	1	0	-	予約		
1	1	1	-	予約		

表 20.2-6 通信プリスケアラ(CDCR)の設定による推奨マシンサイクル

Div*	マシンクロック					推奨マシンサイクル
	MD	D3	D2	D1	D0	
3	1	1	1	0	1	6MHz
4	1	1	1	0	0	8MHz
5	1	1	0	1	1	10MHz
6	1	1	0	1	0	12MHz
7	1	1	0	0	1	14MHz
8	1	1	0	0	0	16MHz

注意：通信プリスケアラコントロールレジスタ(CDCR)の項を参照してください。
D3~D0は、DIV0~DIV3の略です。

【ビット12】SIE (Serial I/O Interrupt Enable)

SIEビットは、シリアルI/Oの割り込み要求を表 20.2-7に示すように制御します。

リセットにより"0"に初期化されます。本ビットは、リード/ライトが可能です。

表 20.2-7 SIE(シリアルI/O割り込み要求制御ビット) の機能

SIE	機能
0	シリアルI/O割り込み禁止 [初期値]
1	シリアルI/O割り込み許可

【ビット11】SIR(Serial i/o Interrupt Request)

SIRビットは、シリアルデータの転送が終了すると"1"にセットされ、割り込み許可時 (SIE = "1") にこのビットが"1"になると、CPUへ割り込み要求を発生します。クリア条件は、MODEビットによって異なります。MODEビットが"0"の場合、SIR ビットへの"0"書き込みによりクリアされます。MODEビットが"1"の場合、SDRレジスタの読出し、または書き込み動作によりクリアされます。

MODEビットの値に関わらずリセット、またはSTOPビットへの"1"書き込み動作によりクリアされます。

本ビットに"1"を書き込んでも意味がありません。リードモディファイライト系命令のリード時は、常に"1"がリードされます。

【ビット10】BUSY

BUSYビットは、シリアル転送が実行中か否かを示します。リセットにより"0"

に初期化されます。本ビットは、リードのみ可能です。

表 20.2-8 BUSYビットの機能

BUSY	機能
0	停止またはシリアルデータレジスタ R/W 待機状態 [初期値]
1	シリアル転送状態

【ビット9】STOP

STOPビットは、シリアル転送を強制的に中断させるビットです。

本ビットを"1"にするとSTOP = "1"による停止状態となります。

リセットにより、"1"に初期化されます。本ビットは、リード/ライトが可能です。

表 20.2-9 STOPビットの機能

STOP	機能
0	通常動作
1	STOP=1による転送停止 [初期値]

【ビット8】STRT

STRTビットは、シリアル転送を起動します。停止状態で"1"を書くことによって転送を開始します。シリアル転送動作中およびシリアルシフトレジスタR/W待機中の"1"のライトは無視され、"0"のライトは意味がありません。

リード時には常に"0"が読み出されます。

20.2.2 シリアルシフトデータレジスタ(SDR)

シリアルシフトデータレジスタ(SDR)は、シリアルI/Oの転送データを保持します。
転送中のSDRレジスタへの書き込み、および読出しは禁止です。

シリアルシフトデータレジスタ (SDR)										
シリアルシフトデータレジスタ										
ch0	000004Ah	7	6	5	4	3	2	1	0	⇐ ビットNo.
ch1	000004Eh	D7	D6	D5	D4	D3	D2	D1	D0	SDR
ch2	000007Eh									
リード/ライト	⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値	⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 20.2-3 シリアルシフトデータレジスタ(SDR)

20.3 I/O拡張シリアルインタフェースの動作

I/O拡張シリアルインタフェースは、SMCSレジスタ、SDRレジスタにより構成され、8bitのシリアルデータの入出力に使用します。

I/O拡張シリアルインタフェースの動作

シリアルデータの入出力は、シフトレジスタの内容がシリアルシフトクロック（外部クロック、内部クロック）の立ち下がりに同期して、ビット直列にシリアル出力端子（SOT1端子）に出力され、立ち上がりに同期してシリアル入力端子（SIN1端子）からビット直列にSDRレジスタに入力されます。シフトの方向（MSBからの転送または、LSBからの転送）は、SMCSレジスタのBDSビットにより指定することができます。

転送が終了すると、SMCSレジスタのMODEビットによって停止状態、または、データレジスタR/W待機状態に入ります。各々の状態から転送状態に移るには、それぞれ次のことを行います。

- 停止状態からの復帰の場合は、STOPビットに"0"、STRTビットに"1"を書き込みます（STOPとSTRTは同時設定が可能です）。
- SDRレジスタR/W待機状態からの復帰時は、データレジスタをリード、またはライトします。

20.3.1 シフトクロック

シフトクロックは内部シフトクロックモード，外部シフトクロックモードの2種類があり，SMCSレジスタの設定で指定します。モードの切り換えは，シリアルI/Oが停止した状態で行ってください。停止状態の確認は，BUSYビットを読み出すことで可能です。

内部シフトクロックモード

通信プリスケアラの出力により動作し，同期タイミング出力としてデューティ比50%のシフトクロックがSCK端子より出力可能です。データが1クロックごとに1bit転送されます。

転送の速度は，次の式で表せます。

$$\text{転送速度(s)} = \frac{A}{\text{内部クロックのマシンサイクル(Hz)}}$$

AはSMCSレジスタのSMDビットで示す分周比で2，2²，2⁴，2⁵，2⁶です。

外部シフトクロックモード

外部シフトクロックモードでは，SCK端子から入力される外部シフトクロックに同期して，データが1クロックごとに1bit転送されます。転送速度はDCから1/(5マシンサイクル)まで可能です。例えば，"1マシンサイクル=0.1μs"のとき，2MHzまで可能です。

命令ごとに転送することも出来，次のような設定をすることによって実現します。

- 1) 外部シフトクロックモードを選択し，SMCSレジスタのSCOEビットを"0"に設定します。
- 2) SCK端子を共有しているポートの方向レジスタに"1"を書き込み，ポートを出力モードに設定します。

以上の設定をした後，ポートのデータレジスタ(PDR)に"1"，"0"を書き込むと，SCK端子に出力されるポートの値を外部クロックとして取り込み，転送動作を行います。シフトクロックの開始は"H"からスタートさせてください。

< 注意事項 >

シリアルI/O動作中のSMCSレジスタ，SDRレジスタへの書き込みは禁止です。

20.3.2 I/O拡張シリアルインタフェースの動作状態

I/O拡張シリアルインタフェースの動作状態として、次の4種類の状態があります。

- ・ STOP状態
- ・ 停止状態
- ・ SDRレジスタのR/W待機状態
- ・ 転送状態

STOP状態

RESET時，またはSMCSのSTOPビットに"1"を書き込んだときの状態で，シフトカウンタは初期化され，SIR = "0"となります。STOP状態からの復帰は，STOP = "0"，STRT = "1"（同時設定可）とすることによって行われます。STOPビットは，STRTビットより優先ビットです。STOP = "1"の場合は，STRT = "1"としても転送動作は行われません。

停止状態

MODEビットが"0"の場合，転送が終了したことによってSMCSレジスタがBUSY = "0" SIR = "1"となり，カウンタが初期化され停止状態に入ります。停止状態からの復帰はSTRT = "1"にすると転送動作が再開されます。

シリアルデータレジスタR/W待機状態

SMCSレジスタのMODEビットが"1"の場合，シリアル転送が終了すると，BUSY = "0"，SIR = "1"となりSDRレジスタR/W待機状態に入ります。割込み許可レジスタが許可状態ならば，本ブロックより割込み信号を出します。

R/W待機状態からの復帰はSDRレジスタが読み出されたり，書き込まれたりするとBUSY = "1"となり転送動作が再開されます。

転送状態

BUSY = "1"でシリアル転送を行っている状態です。MODEビットによりそれぞれ停止状態およびR/W待機状態に遷移します。

図 20.3-1に，I/O拡張シリアルインタフェース動作遷移図を，図 20.3-2にシリアルデータレジスタへのリード，ライト概念図を示します。

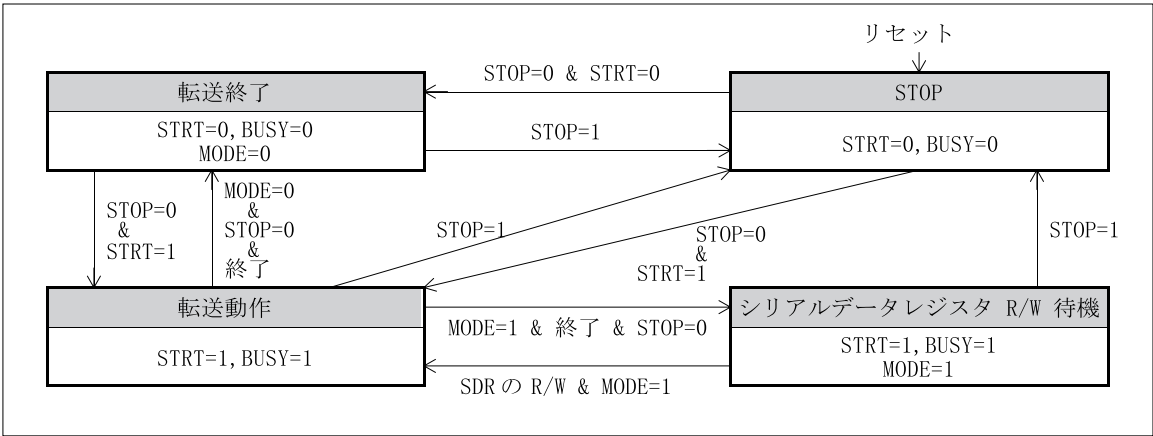


図 20.3-1 I/O拡張シリアルインタフェース動作遷移図

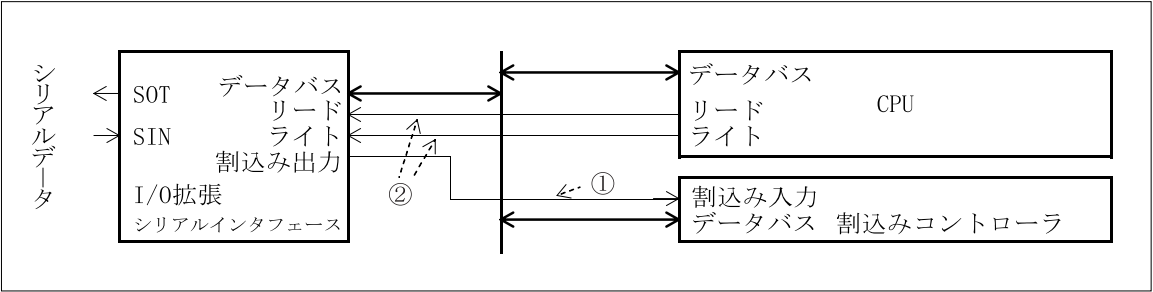


図 20.3-2 シリアルデータレジスタへのリード、ライト概念図

図 20.3-2の図中における、①と②について説明します。

MODE = "1"の場合、シフトクロックカウンタによって転送終了し、SIR = "1" となってリード/ライト待機状態に入ります。SIE ビットが"1"ならば割り込み信号を発生します。ただし、SIE がインアクティブの場合やSTOP へ"1"を書き込むことによる転送中断の場合は割り込み信号は発生しません。

SDRレジスタがリード/ライトされると、割り込み要求はクリアされシリアル転送を開始します。

20.3.3 シフト動作のスタート/ストップタイミング

シフト動作をスタート/ストップする場合は、以下のとおり設定します。

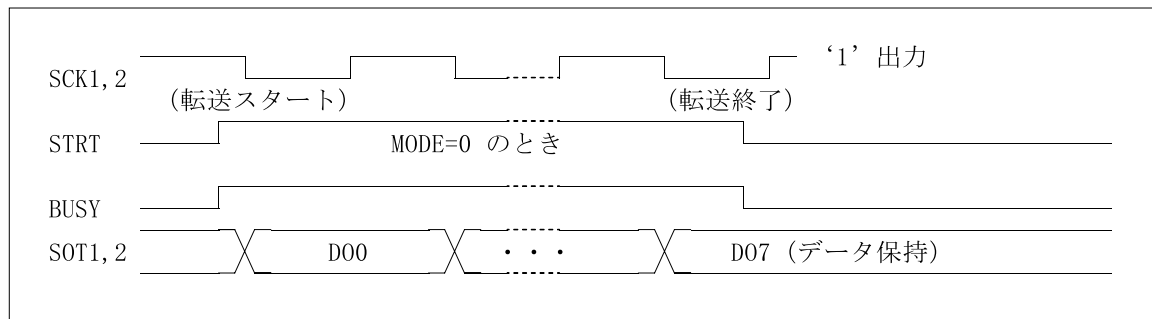
スタート：SMCSレジスタのSTOPビットを"0"，STRTビットを"1"に設定します。

ストップ：転送終了によって停止する場合とSTOP = "1"によって停止する場合があります。STOP = "1"によって停止 MODEビットに係わらずSIR = "0"のまま停止します。転送終了で停止 MODEビットに係わらずSIR = "1"となり停止します。

シフト動作のスタート/ストップタイミング

BUSYビットはMODEビットに係わらずシリアル転送状態の場合に"1"となり、停止状態、またはR/W待機状態の場合に"0"となります。転送状態を確認したい場合はこのビットをリードしてください。

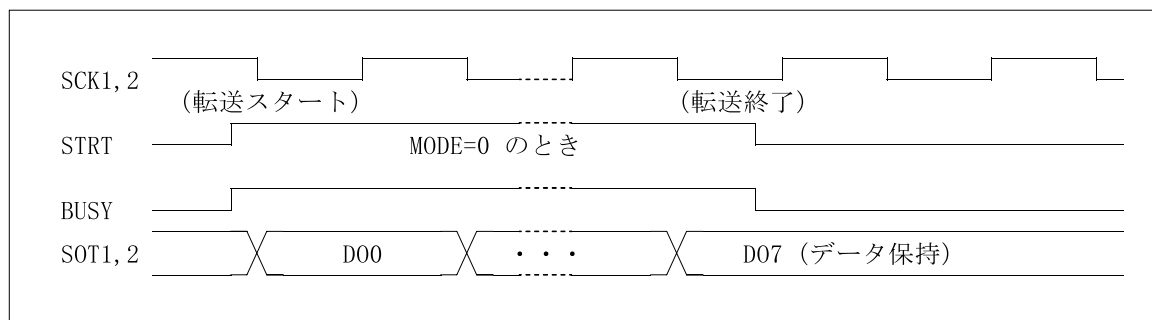
内部シフトクロックモード（LSBファースト）



(注意) D07～D00は出力データを示します。

図 20.3-3 シフト動作のタイミング（内部クロック）

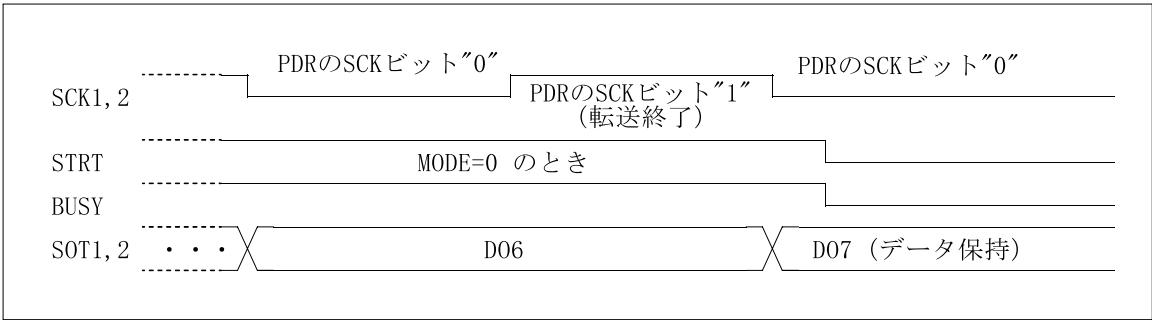
外部シフトクロックモード（LSBファースト）



(注意) D07～D00は出力データを示します。

図 20.3-4 シフト動作のタイミング（外部クロック）

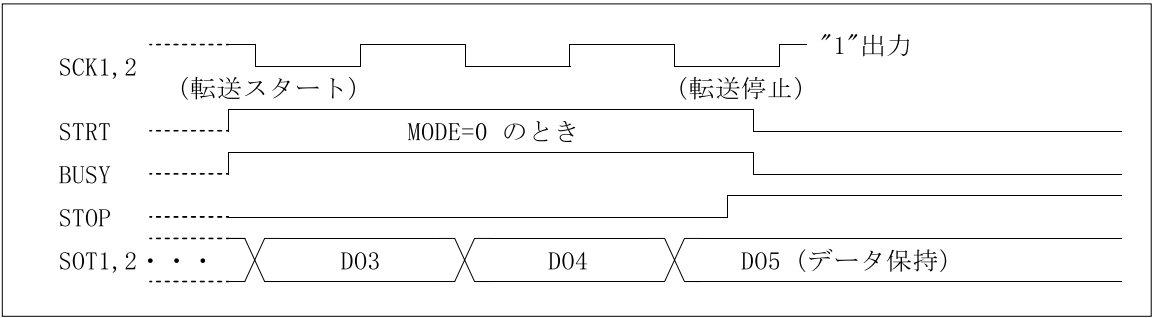
外部シフトクロックモードで命令シフトを行った場合（LSBファースト）



(注意) D07～D00は出力データを示します。
命令シフトでは、PDRのSCKに対応するビットに"1"が書き込まれた場合は"H"が出力され、"0"が書き込まれた場合は"L"が出力されます。(ただし、外部シフトクロックモード選択で、SCOE = "0"の場合)

図 20.3-5 シフト動作のタイミング（外部シフトクロックモードで命令ごとにシフトさせた場合）

STOP = "1"による停止（LSBファースト，内部クロック時）



(注意) D07～D00は出力データを示します。

図 20.3-6 STOPビットを"1"とした場合のストップタイミング

20.3.4 シリアルデータの入出力のタイミング

シリアルデータの転送中は、シフトクロックの立ち下がりでシリアル出力端子（SOT2）からのデータが出力され、立ち上がりでシリアル入力端子（SIN0）のデータが入力されます。

シリアルデータの入出力のタイミング

図 20.3-7に、シリアルデータの入出力シフトタイミングを示します。

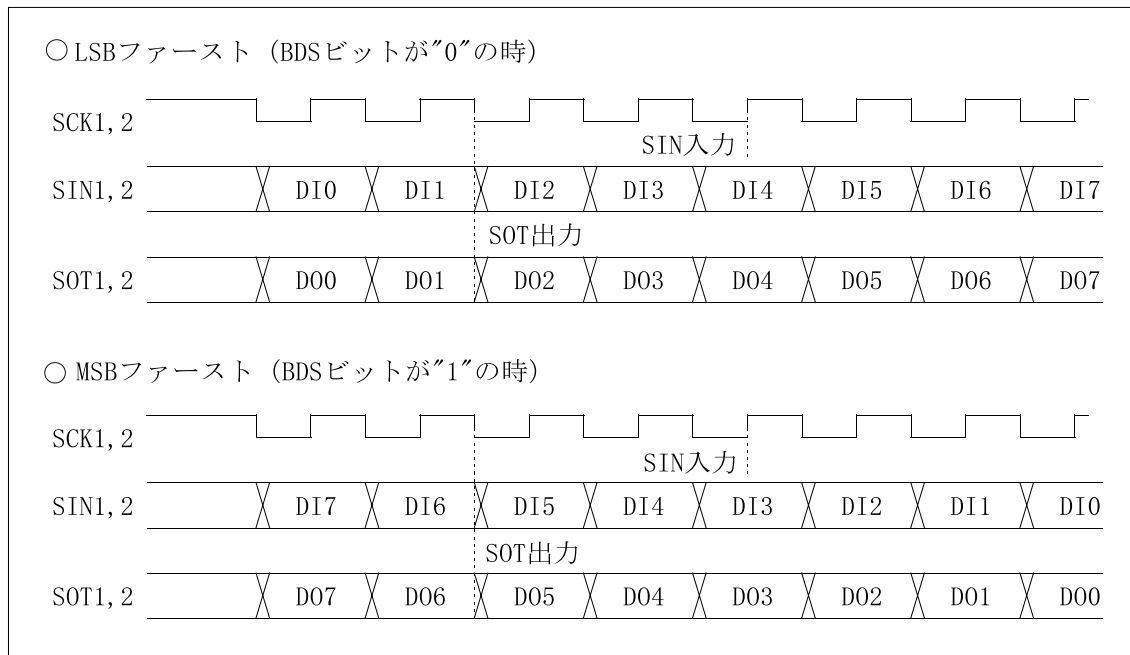


図 20.3-7 シリアルデータの入出力のシフトタイミング

20.3.5 I/O拡張シリアルインタフェースの割り込み機能

I/O拡張シリアルインタフェースは，CPUに対し割り込み要求を発生することができます。データの転送終了時に割り込みフラグであるSIRビットがセットされ，割り込みを許可するSMCSレジスタのSIEビットが"1"の場合，CPUへ割り込み要求を出力します。

I/O拡張シリアルインタフェースの割り込み機能

図 20.3-8に，I/O拡張シリアルインタフェースの割り込み信号出力タイミングを示します。

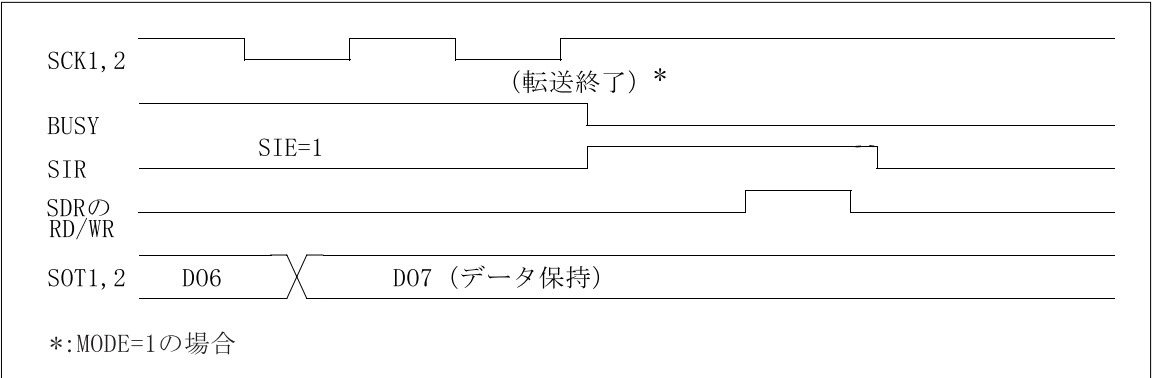


図 20.3-8 I/O拡張シリアルインタフェースの割り込み信号出力タイミング

第21章 I²Cインタフェース

この章では、I²Cインタフェースの機能と概要について説明します。

- 21.1 I²Cインタフェースの概要
- 21.2 I²Cインタフェースのブロックダイアグラム
- 21.3 I²Cインタフェースのレジスタ
- 21.4 I²Cインタフェースの動作

21.1 I²Cインタフェースの概要

I²Cインタフェースは、Inter IC BUSをサポートするシリアルI/Oポートで、I²Cバス上のマスター/スレーブデバイスとして動作し、以下の特長があります。

I²Cインタフェースの特長

MB90570シリーズでは、I²Cインタフェースを1ch内蔵しています。

以下に、I²Cインタフェースの特長を示します。

- マスター/スレーブ送受信
- アービトレーション機能
- クロック同期化機能
- スレーブアドレス/ゼネラルコールアドレス検出機能
- 転送方向検出機能
- スタートコンディションの繰り返し発生および検出機能
- バスエラー検出機能
- 本I²Cは、I²Cの標準モード[(シリアルクロック周波数100kHz(MAX))]に対応

21.2 I²Cインタフェースのブロックダイアグラム

図 21.2-1に、I²Cインタフェースのブロックダイアグラムを示します。

I²Cインタフェースのブロックダイアグラム

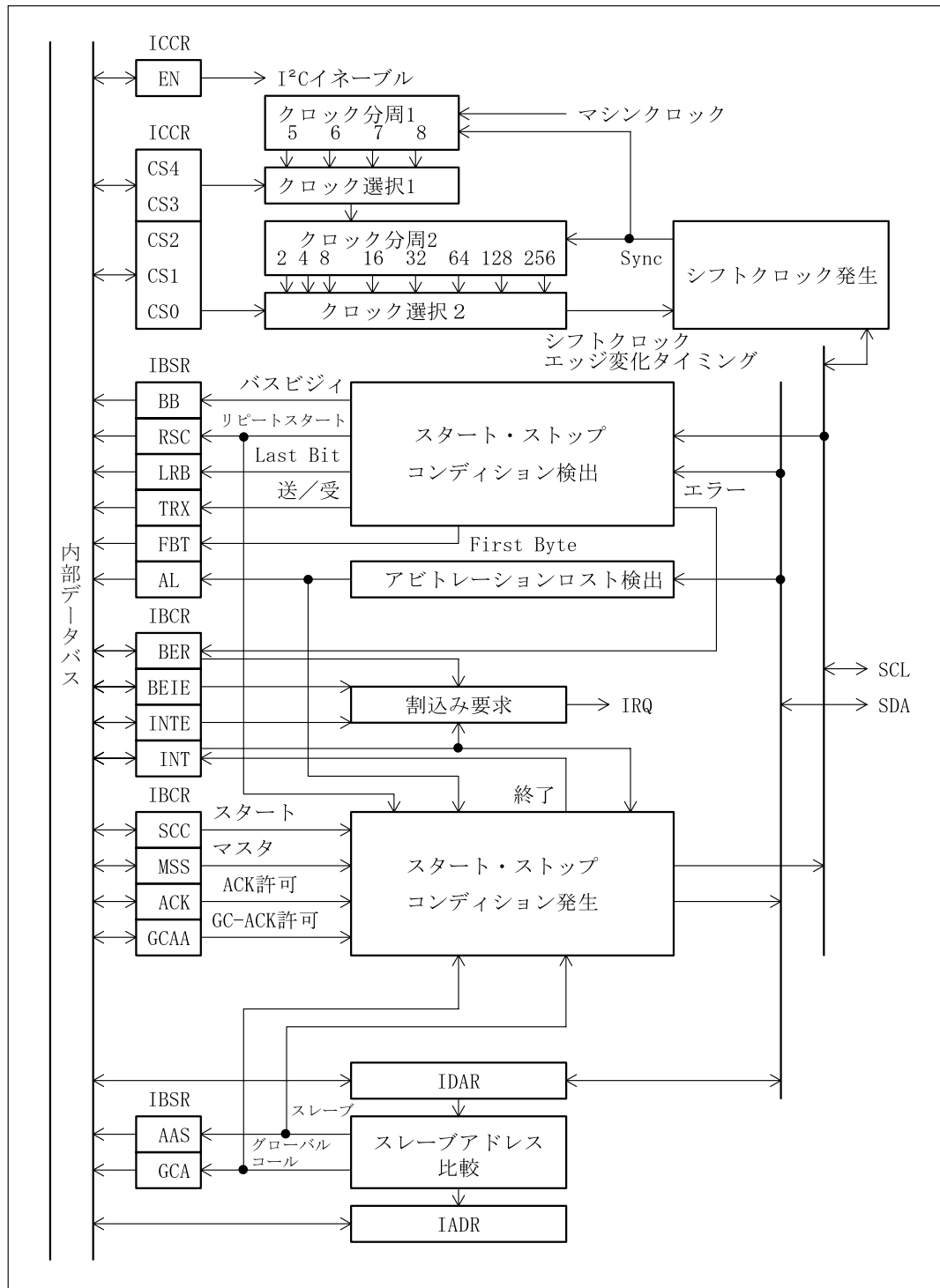


図 21.2-1 I²Cインタフェースのブロックダイアグラム

21.3 I²Cインタフェースのレジスタ

I²Cインタフェースのレジスタには、次の5種類があります。

- ・バスステータスレジスタ
- ・バスコントロールレジスタ
- ・クロックコントロールレジスタ
- ・アドレスレジスタ
- ・データレジスタ

I²Cインタフェースのレジスタ

・バスステータスレジスタ								
	7	6	5	4	3	2	1	0 ⇐ビットNo.
アドレス:000068 _H	BB	RSC	AL	LRB	TRX	AAS	GCA	IBSR
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
・バスコントロールレジスタ								
	15	14	13	12	11	10	9	8 ⇐ビットNo.
アドレス:000069 _H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
・クロックコントロールレジスタ								
	7	6	5	4	3	2	1	0 ⇐ビットNo.
アドレス:00006A _H	—	—	EN	CS4	CS3	CS2	CS1	CS0
リード/ライト ⇒	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(-)	(0)	(X)	(X)	(X)	(X)	
・アドレスレジスタ								
	15	14	13	12	11	10	9	8 ⇐ビットNo.
アドレス:00006B _H	—	A6	A5	A4	A3	A2	A1	A0
リード/ライト ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(X)	(X)	(X)	(X)	(X)	(X)	
・データレジスタ								
	7	6	5	4	3	2	1	0 ⇐ビットNo.
アドレス:00006C _H	D7	D6	D5	D4	D3	D2	D1	D0
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

図 21.3-1 I²Cインタフェースのレジスタ

21.3.1 バスステータスレジスタ(IBSR)

バスステータスレジスタ(IBSR)には、以下の機能があります。

- ・ I²Cバスの状態を示す
- ・ 繰り返しスタートコンディション検出
- ・ アドレクションロスト検出
- ・ アクノリッジ格納
- ・ 第1バイト検出
- ・ アドレッシング検出
- ・ ゼネラルコールアドレス検出
- ・ データ転送

バスステータスレジスタ(IBSR)

・ バスステータスレジスタ								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:000068H	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT IBSR
リード/ライト ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 21.3-2 バスステータスレジスタ(IBSR)

【ビット7】 BB(Bus Busy)

BBは、I²Cバスの状態を示すビットです。

表 21.3-1 BB(Bus Busy) ビットの機能

BB	機 能
0	ストップコンディションを検出した。[初期値]
1	スタートコンディションを検出した。(バスは使用されている)

【ビット6】 RSC (Repeated Start Condition)

RSCは、繰り返しスタートコンディションを検出するビットです。INTビットへの"0"書込み、スレープ時にアドレスされなかった場合、バス停止中のスタートコンディション検出、またはストップコンディション検出でクリアされます。

表 21.3-2 RSC (Repeated Start Condition) ビットの機能

RSC	機 能
0	繰り返しスタートコンディションは検出されていない。[初期値]
1	バス使用中に、再びスタートコンディションを検出した

【ビット5】AL (Arbitration Lost)

ALは、アービトレーションロストを検出するビットです。
INT ビットへの"0"書込みでクリアされます。

表 21.3-3 AL (Arbitration Lost) ビットの機能

AL	機 能
0	アービトレーションロスト検出されていない。[初期値]
1	マスター送信中にアービトレーションロストが発生した、または他のシステムがバスを使用中であるときに、MSS ビットに"1"書込みを行った場合

【ビット4】LRB (Last Recieved Bit)

LRBは、アクノリッジ格納ビットで、受信側からのアクノリッジを格納します。スタートコンディションまたはストップコンディションの検出でクリアされます。

【ビット3】TRX (Transfer / Recieve)

TRXは、データ転送の送受信を示すビットです。

表 21.3-4 TRX (Transfer / Recieve) ビットの機能

TRX	機 能
0	受信状態 [初期値]
1	送信状態

【ビット2】AAS (Addressed As Slave)

AASは、アドレッシング検出ビットです。
スタートコンディションまたはストップコンディションの検出でクリアされます。

表 21.3-5 AAS (Addressed As Slave) ビットの機能

AAS	機 能
0	スレーブ時に、アドレッシングされていない。[初期値]
1	スレーブ時に、アドレッシングされた。

【ビット1】GCA (General Call Address)

GCAは、ゼネラルコールアドレス(00_H)検出ビットです。スタートコンディションまたはストップコンディションの検出でクリアされます。

表 21.3-6 GCA (General Call Address) ビットの機能

GCA	機 能
0	スレーブ時に、ゼネラルコールアドレスは受信されていない。[初期値]
1	スレーブ時に、ゼネラルコールアドレスを受信した。

【ビット0】FBT (First Byte Transfer)

FBTは、第1バイト検出ビットです。スタートコンディションの検出で"1"にセットされても、INTビットの"0"書込み、またはスレーブ時にアドレスされなかった場合クリアされます。

表 21.3-7 FBT (First Byte Transfer)ビットの機能

FTB	機 能
0	受信データが第1バイト以外である。[初期値]
1	受信データが第1バイト (アドレスデータ) である。

21.3.2 バスコントロールレジスタ(IBCR)

バスコントロールレジスタ(IBCR) には、以下の機能があります。

- ・ 割込み要求/割込み許可
- ・ スタートコンディション発生
- ・ マスタ/スレーブの選択
- ・ アクノリッジ発生許可

バスコントロールレジスタ(IBCR)									
・ バスコントロールレジスタ									
アドレス:000069H	15	14	13	12	11	10	9	8	⇐ ビットNo.
	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR
	リード/ライト ⇒ 初期値 ⇒	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

図 21.3-3 IBCR (バスコントロールレジスタ)

【ビット15】BER (Bus ERror)

BERは、バスエラー割込み要求フラグビットです。このビットがセットされた場合、CCRレジスタのENビットはクリアされ、I²Cインタフェースは停止状態になり、データ転送は中断されます。

表 21.3-8 BER (Bus ERror)ビットの機能

BER		機 能
ライト時	0	バスエラー割込み要求フラグをクリアする。[初期値]
	1	関係なし
リード時	0	バスエラーは検出されていない。[初期値]
	1	データ転送中に、不正なスタート、ストップコンディションを検出した。

【ビット14】BEIE (Bus Error Interrupt Enable)

BEIEは、バスエラー割込み許可ビットです。このビットが"1"の場合、BERビットが"1"なら割込みを発生します。

表 21.3-9 BEIE (Bus Error Interrupt Enable) ビットの機能

BEIE	機 能
0	バスエラー割込み禁止 [初期値]
1	バスエラー割込み許可

【ビット13】SCC (Start Condition Continue)

SCCビットは、スタートコンディション発生ビットです。このビットの読み出し値は常に"0"です。

表 21.3-10 SCC (Start Condition Continue) ビットのライト時の機能

SCC	機 能
0	関係なし [初期値]
1	マスタ転送時に再びスタートコンディションを発生し、アドレスデータ転送を開始します。

【ビット12】MSS (Master Slave Select)

MSSビットは、マスタ/スレーブの選択ビットです。このビットは、マスター送信中にアービトレーションロストが発生した場合クリアされ、スレーブモードになります。

表 21.3-11 MSS (Master Slave Select) の機能

MSS	機 能
0	ストップコンディションを発生し、転送終了後スレーブモードとなります。 [初期値]
1	マスタモードとなりスタートコンディションを発生し、アドレスデータ転送を開始します。

【ビット11】ACK (ACKnowledge)

データを受信した場合のアクノリッジ発生許可ビットです。このビットは、スレーブ時のアドレスデータ受信時には無効となります。

表 21.3-12 ACK (ACKnowledge) の機能

ACK	機 能
0	アクノリッジ発生しない。 [初期値]
1	アクノリッジ発生する。

【ビット10】GCAA (General Call Address Acknowledge)

ゼネラルコールアドレスを受信した場合のアクノリッジ発生許可ビットです。

表 21.3-13 GCAA (General Call Address Acknowledge) の機能

GCAA	機 能
0	アクノリッジ発生しない。 [初期値]
1	アクノリッジ発生する。

【ビット9】INTE (INTerrupt Enable)

INTEは、割込み許可ビットです。このビットが"1"の場合、INTビットが"1"なら割込みを発生します。

表 21.3-14 INTE (INTerrupt Enable) の機能

INTE	機 能
0	割込み禁止 [初期値]
1	割込み許可

【ビット8】INT (INTerrupt)

INTは、転送終了割込み要求フラグビットです。このビットが"1"の場合、SCLラインは"L"レベルに保たれます。このビットへの"0"書込みによりクリアされ、SCLラインを開放し、次バイトの転送を行います。また、マスタ時にスタートコンディションまたはストップコンディションの発生により"0"にリセットされます。

表 21.3-15 INT (INTerrupt) の機能

INT		機 能
ライト時	0	転送終了割込み要求フラグをクリアする。[初期値]
	1	関係なし
リード時	0	転送が終了していない。[初期値]
	1	アクノリッジビットを含めた1バイト転送が終了した時に次の条件に該当する場合セットされます。 <ul style="list-style-type: none"> ・バスマスタである。 ・アドレスされたスレーブである。 ・ゼネラルコールアドレスを受信した。 ・アービトレーションロストが起った。 ・他のシステムがバスを使用中にスタートコンディションを発生しようとした。

SCC , MSS , INTビットの競合についての注意

SCC , MSS , INT ビットの同時書込みにより、次バイト転送、スタートコンディション発生、ストップコンディション発生の競合が起こります。この時の優先度は次のようになります。

- 1) 次バイト転送とストップコンディション発生
 - INTビットに"0", MSSビットに"0"を書き込むと、MSSビットの"0"書込みが優先され、ストップコンディションが発生されます。
- 2) 次バイト転送とスタートコンディション発生
 - INTビットに"0", SCCビットに"1"を書き込むと、SCCビットの"1"書込みが優先され、スタートコンディションが発生されます。
- 3) スタートコンディション発生とストップコンディション発生
 - SCCビットに"1", MSS ビットに"0"の同時書込みは禁止します。

21.3.3 クロックコントロールレジスタ (ICCR)

クロックコントロールレジスタ (ICCR) には、以下の機能があります。

- ・I²Cインタフェース動作許可
- ・シリアルクロックの周波数の設定

クロックコントロールレジスタ (ICCR)

・ クロックコントロールレジスタ								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:00006Ah	—	—	EN	CS4	CS3	CS2	CS1	CS0 ICCR
リード/ライト ⇒	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇒	(-)	(-)	(0)	(X)	(X)	(X)	(X)	(X)

図 21.3-4 クロックコントロールレジスタ (ICCR)

【ビット7,6】未使用ビット

【ビット5】EN (ENable)

ENは、I²Cインタフェース動作許可ビットです。このビットが"0"の場合、BSRレジスタ、BCRレジスタ(BER、BEIEビットを除く)の各ビットはクリアされます。BERビットがセットされた場合、このビットはクリアされます。

表 21.3-16 EN(Enable)ビットの機能

EN	機 能
0	動作禁止 [初期値]
1	動作許可

【ビット4～0】CS4～0(Clock Period Select 4-0)

シリアルクロックの周波数を設定するビットです。
シフトクロックの周波数 (f_{sck}) は次式のように設定されます。

$$\text{fsck} = \frac{\phi}{m \times n + 4} \quad \phi: \text{マシンのクロック}$$

< 注意事項 >

- ・シリアルクロック周波数を100kHz以上に設定しないでください。
- ・+4のサイクルは、SCL端子の出力レベルが変化しているための最小のオーバーヘッドです。SCL端子の立ち上がりのディレイが大きい場合やスレーブデバイスでクロックを引き延ばしている場合には、この値より大きくなります。

m, nは、CS4~0に対して、表 21.3-17, 表 21.3-18のようになります。

表 21.3-17 シリアルクロック周波数設定 (CS4,3)

m	CS4	CS3
5	0	0
6	0	1
7	1	0
8	1	1

表 21.3-18 シリアルクロック周波数設定 (CS2~0)

n	CS2	CS1	CS0
4	0	0	0
8	0	0	1
16	0	1	0
32	0	1	1
64	1	0	0
128	1	0	1
256	1	1	0
512	1	1	1

例えば, $f_{clk} = 16\text{MHz}$ のとき, $m = 5$, $n = 32$ を選択した場合, シリアルクロック周波数は, 97.561KHz になります。

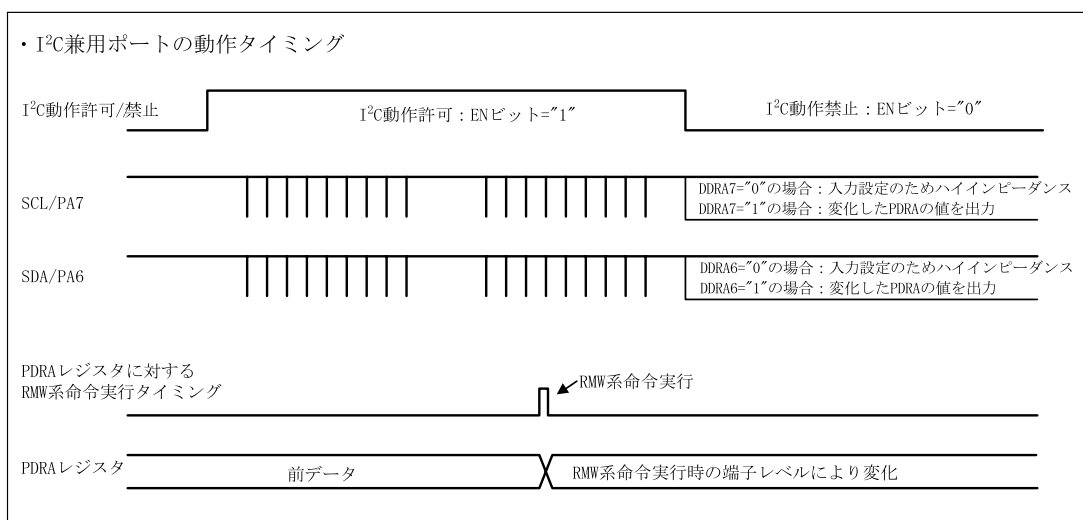
< 注意事項 >

I²C動作許可ビット (ENビット) の設定によって, 以下のようにI²C兼用ポート端子の出力は変化します。

- ENbit=1 (動作許可) の場合
DDRA:bit6, DDRA:bit7 の設定値 (入力設定, 出力設定) に関わらず SDA/PA6, SCL/PA7 端子には I²C 出力信号が出力されます。
- ENbit=0 (動作禁止) の場合
DDRA:bit6=1, DDRA:bit7=1 (出力設定) の場合, SDA/PA6, SCL/PA7 端子には PDRA レジスタの PA6, PA7 の設定値が出力されます。

< 注意事項 >

I²C動作中にI²C端子と同系列のポートデータレジスタ (PDRA) に対してRMW系の命令を実行すると, リード時にPDRAのbit6, bit7には端子レベルが読み込まれます。このため, PA7/SCL, PA6/SDA 端子レベルによって, PDRAのbit6, bit7値が変化してしまう場合がありますので注意が必要です。I²C兼用ポートの変化タイミングを以下に示します。



RMW命令系については, 「付録 B.8 F²MC-16LX命令一覧表」にてご確認ください。

21.3.4 アドレスレジスタ(IADR)

アドレスレジスタ(IADR)は、スレーブアドレスを指定します。

アドレスレジスタ(IADR)

・アドレスレジスタ								
	15	14	13	12	11	10	9	8 ←ビットNo.
アドレス:00006B _H	—	A6	A5	A4	A3	A2	A1	A0 IADR
リード/ライト ⇔	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇔	(-)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

図 21.3-5 IADR (アドレスレジスタ)

【ビット14～8】A6～A0(スレーブアドレスビット)

A6～A0は、スレーブアドレスを指定するレジスタです。スレーブ時、アドレスデータ受信後にDARレジスタとの比較が行われ、一致している場合はマスターに対してアクノリッジを送信します。

21.3.5 データレジスタ(IDAR)

データレジスタ(IDAR)は、シリアル転送に使用されるデータレジスタです。

データレジスタ(IDAR)

• データレジスタ								
	7	6	5	4	3	2	1	0 ⇐ ビットNo.
アドレス:00006CH	D7	D6	D5	D4	D3	D2	D1	D0 IDAR
リード/ライト ⇨	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
初期値 ⇨	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

図 21.3-6 データレジスタ(IDAR)

【ビット7～0】D7～D0(データビット)

D7～D0(データビット)は、シリアル転送に使用されるデータレジスタであり、MSBから転送されます。データ受信時(TRX = "0")は、データ出力値は"1"になります。

このレジスタの書込み側はダブルバッファになっており、バスが使用中(BB = "1")である場合、書込みデータは各バイト転送時にシリアル転送用のレジスタにロードされます。読み出し時はシリアル転送用のレジスタを直接読み出すため、受信データはINT ビットがセットされている場合のみ有効です。

21.4 I²Cインタフェースの動作

I²Cバスは、1本のシリアルデータライン(SDA)と1本のシリアルクロックライン(SCL)の2本の双方向バスラインにより通信が行われます。I²Cインタフェースはそれに対して2本のオープンドレイン入出力端子(SDA,SCL)を有し、ワイヤード論理を可能にします。

スタートコンディション

バスが開放されている状態(BB = "0", MSS = "0")でMSSビットに"1"を書き込むと、I²Cインターフェースはマスタモードとなり、同時にスタートコンディションを発生します。マスタモードでは、バスが使用状態(BB = "1")であっても、SCCビットに"1"を書き込むことで再びスタートコンディションを発生させることができます。

スタートコンディションを発生させる条件として、次の2通りがあります。

- 1) バスが使用されていない状態(MSS=0*BB=0*INT=0*AL=0)でのMSSビットへの"1"書込み。
- 2) バスマスタ時の割り込み状態(MSS=1*BB=1*INT=1*AL=0)でのSCCビットへの"1"書込み。

他のシステム(アイドル状態中)がバス使用中に、MSSビットへ"1"書込みを行うと、ALビットが"1"にセットされます。1), 2) 以外でのMSSビット, SCCビットへの"1"書込みは無視されます。

ストップコンディション

マスタモード(MSS = "1")時に、MSSビットに"0"を書き込むと、ストップコンディションを発生し、スレーブモードになります。

ストップコンディションを発生させる条件は、次のとおりです。

- バスマスタ時の割り込み状態(MSS=1*BB=1*INT=1*AL=0)でのMSSビットへの"0"書込み。

上記以外での、MSS ビットへの"0"書込みは、無視されます。

アドレッシング

マスタモードでは、スタートコンディション発生後、BB = "1", TRX = "1"にセットされ、IDARレジスタの内容をMSBから出力します。アドレスデータ送信後、スレーブからアクノリッジを受信すると、送信データのビット0(送信後のIDARレジスタのビット0)を反転してTRXビットへ格納します。

スレーブモードでは、スタートコンディション発生後、BB = "1", TRX = "0"にセットされ、マスタからの送信データをIDARレジスタへ受信します。アドレスデータ受信後IDARレジスタとIADRレジスタとの比較が行われ、一致している場合、AAS = "1"にセットし、マスタに対してアクノリッジを送信します。その後、受信データのビット0(受信後のIDARレジスタのビット0)をTRXビットへ格納します。

アービトレーション

マスタ送信時に、他のマスタも同時にデータを送信している場合、アービトレーションが起こります。I²Cインターフェースの送信データが"1", SDAライン上のデータが"L"レベルの場合、I²Cインターフェースはアービトレーションを失ったと見なし、AL = "1"にセットします。また、前述のようにバスが使用状態の時に、スタートコンディションを発生させようとした場合もAL = "1"にセットされます。

AL = "1"にセットされると、MSS = "0", TRX = "0"となり、スレーブ受信モードとなります。

アクノリッジ

アクノリッジは、受信側が送信側に対して送信します。データ受信時は、ACKビットによってアクノリッジの有無を選択することができます。データ送信時は、受信側からのアクノリッジがLRBビットに格納されます。

スレーブ送信時に、マスタ受信側からアクノリッジを受信しなかった場合、TRX="0"となりスレーブ受信モードになります。これにより、マスタはスレーブがSCLラインを開放した場合に、ストップコンディションを発生することができます。

バスエラー

以下の条件が成立した場合はバスエラーと判断され、I²Cインタフェースは停止状態になります。

- データ転送中（ACKビット含む）のI²Cバス上の基本規定違反の検出
- マスター時のストップコンディション検出
- バスアイドル時のI²Cバス上の基本規定違反の検出

SDA="L"、SCL="L"時におけるスタートコンディション発生命令の実行

SDA="L"、SCL="L"時に、スタートコンディション発生命令を実行（MSSビットへの"1"書込み）した場合、BB="0"、AL="1"の状態になります。この場合、転送が終了していないため、転送終了割込要求フラグ（INTビット）は立ちません。このため、本状態の検出は、プログラム上でBBビットとALビットを監視して行う必要があります。

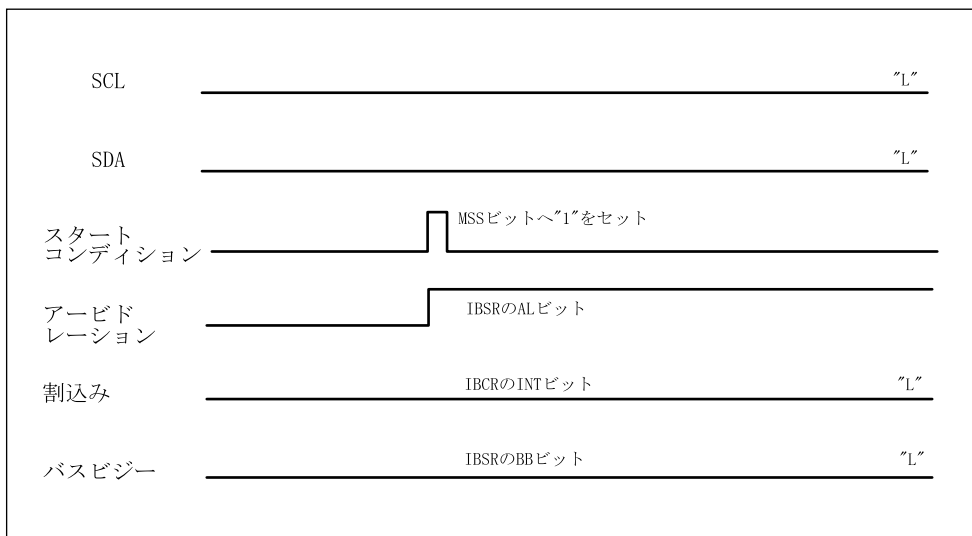


図 21.4-1 SDA="L"、SCL="L"時におけるスタートコンディション発生命令実行時の各フラグの変化タイミング

21.4.1 I²Cインタフェースの動作フロー

I²Cインタフェースのマスタ送受信プログラムのフロー例（割込み使用）を図 21.4-2に示します。また、I²Cインタフェースのスレーブプログラムのフロー例（割込み使用）を図 21.4-3に示します。

I²Cインタフェースの動作フロー

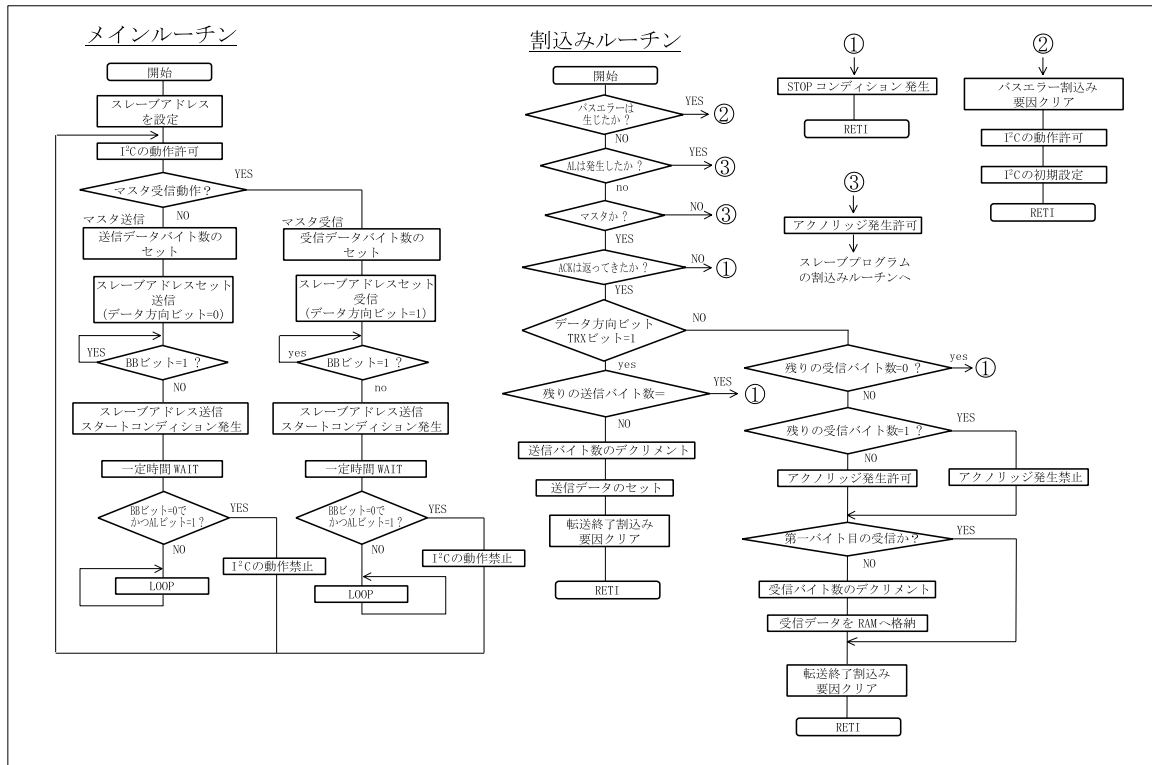


図 21.4-2 I²Cマスタ送受信プログラムのフロー例（割込み使用）

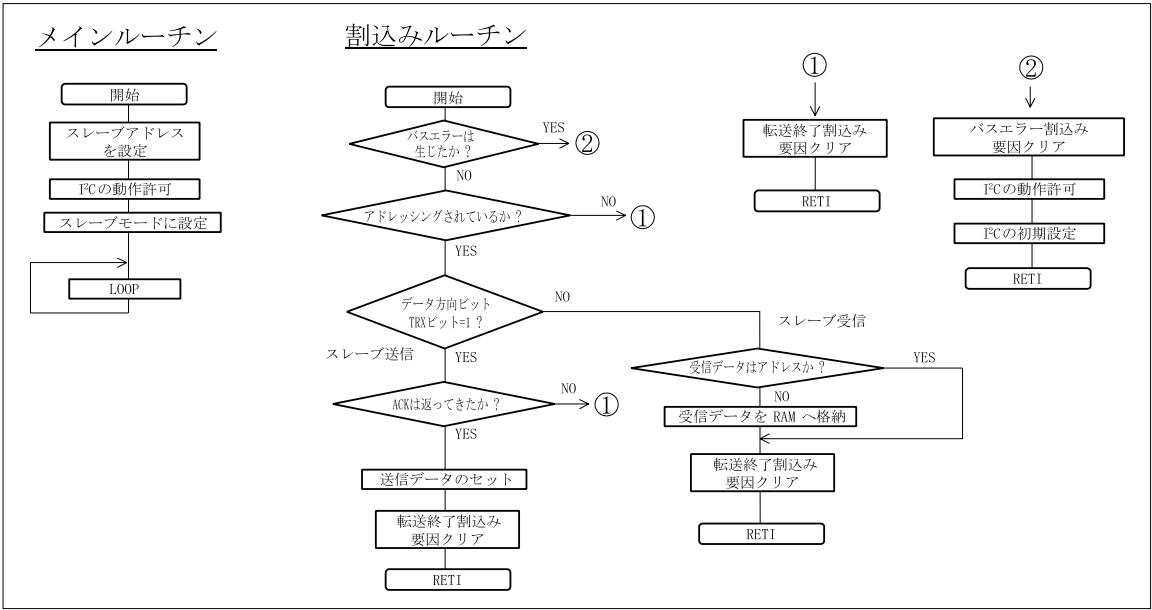


図 21.4-3 I²Cスレーププログラムのフロー例（割り込み使用）

第22章 チップセレクト機能

この章では，チップセレクトの機能と動作について説明しています。

- 22.1 チップセレクト機能の概要
- 22.2 チップセレクト機能のレジスタ
- 22.3 チップセレクト機能の動作
- 22.4 チップセレクト機能のデコードアドレス空間

22.1 チップセレクト機能の概要

メモリやI/Oの接続を容易にするためのチップセレクト信号を発生するモジュールです。

チップセレクト機能の概要

8本のチップセレクト出力端子を持ち、ハードウェアにて設定された領域を各端子レジスタに設定し、アドレスに対するアクセスを検出するとセレクト信号を端子から出力します。

チップセレクト機能のブロックダイアグラム

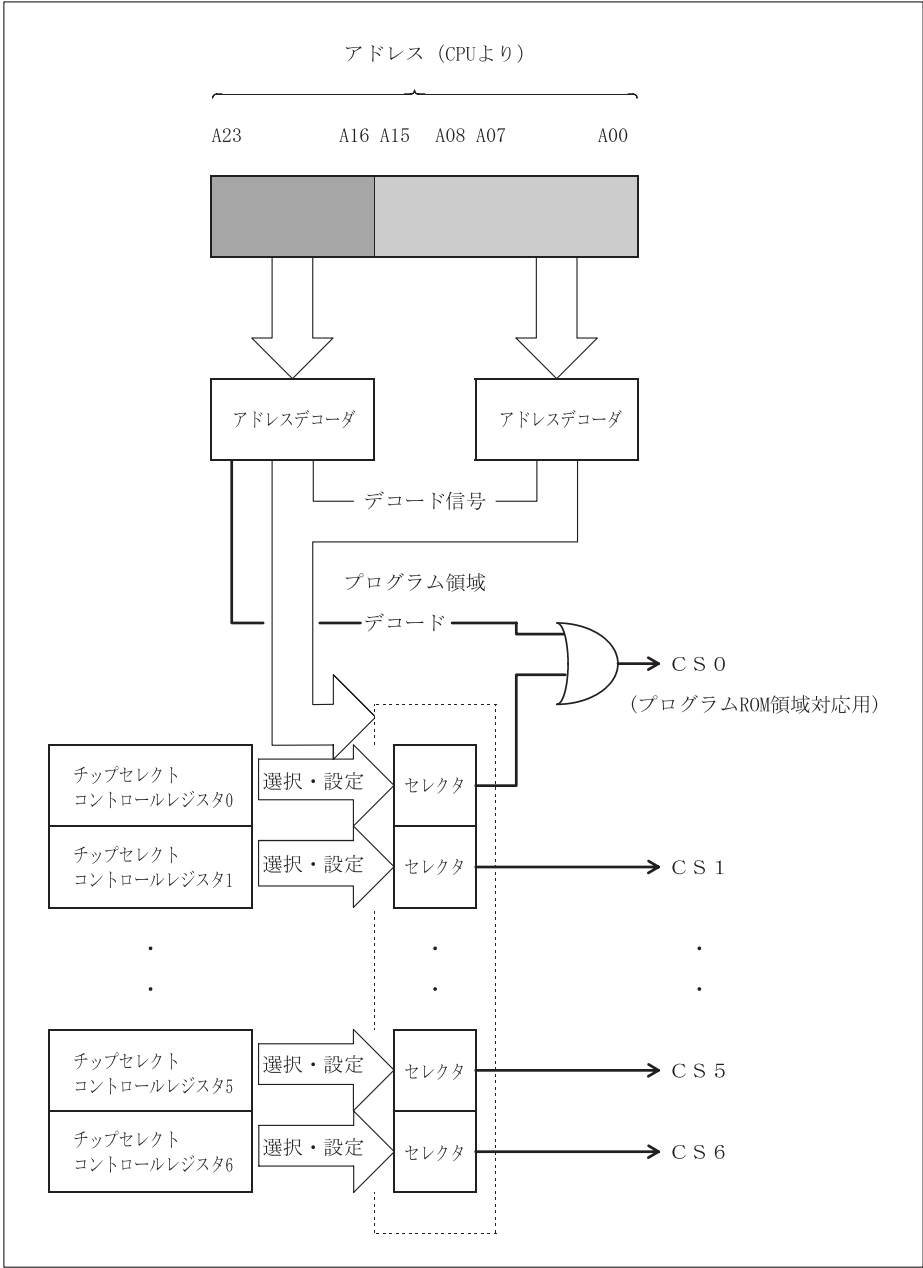


図 22.1-1 チップセレクト機能のブロックダイアグラム

22.2 チップセレクト機能のレジスタ

チップセレクト機能のレジスタには、次の1種類があります。

- ・チップセレクトコントロールレジスタ（CSCR0～6）

チップセレクトコントロールレジスタ（CSCR0～6）									
アドレス:000081 _H :000083 _H :000085 _H	15	14	13	12	11	10	9	8	チップセレクトコントロールレジスタ (奇数:CSCR1/3/5/7)
	—	—	—	—	ACTL	OPEL	CSA1	CSA0	
アドレス:000080 _H :000082 _H :000084 _H :000086 _H	7	6	5	4	3	2	1	0	チップセレクトコントロールレジスタ (偶数:CSCR0/2/4/6)
	—	—	—	—	ACTL	OPEL	CSA1	CSA0	

図 22.2-1 チップセレクトコントロールレジスタ（CSCR0～6）

- 【bit 15/07～12/04】: 未使用ビットです。
- 空きビットです。読出し値は不定です。
- 【bit 11/03】: ACTL
- CS0～6の各端子それぞれのアクティブレベルを設定するビットです。以下の
ように動作設定されます。
- "0":CS0～6の各端子は、それぞれデコード時に"L"を出力します。
- "1":CS0～6の各端子は、それぞれデコード時に"H"を出力します。
- 【bit10/02】: OPEL
- CS0～6の各端子の出力を外部に出力するか否かを設定するビットです。以下
のようになります。
- "0": CS0～6の各端子からのデコード出力を禁止します。
- "1": CS0～6の各端子からのデコード出力を許可します。
- 【bit09/01～08/00】: CSA1,CSA0
- 各チップセレクト端子のアドレスデコード範囲（ROM/RAM/外付け回路）の選
択が可能です。各容量／範囲は、表 22.4-1をご参照ください。

22.3 チップセレクト機能の動作

CPUがプログラムまたはデータにアクセスする際のアドレスの最上位バイト・最下位バイトをデコードすることにより，出力設定レジスタの設定にしたがってチップセレクトをアクティブにします。

チップセレクト機能の動作

このモジュールのチップセレクト機能は，表 22.4-1に示す領域を出力設定レジスタのCSA1～0ビットにて設定し，CPUがその領域をアクセスするとCS0～7の各端子信号がアクティブとなります。さらに出力設定レジスタのOPELビットにて出力をマスクすることもできます。

またCS0～7のアクティブレベルを出力設定レジスタのACTLにて変更することが可能です。

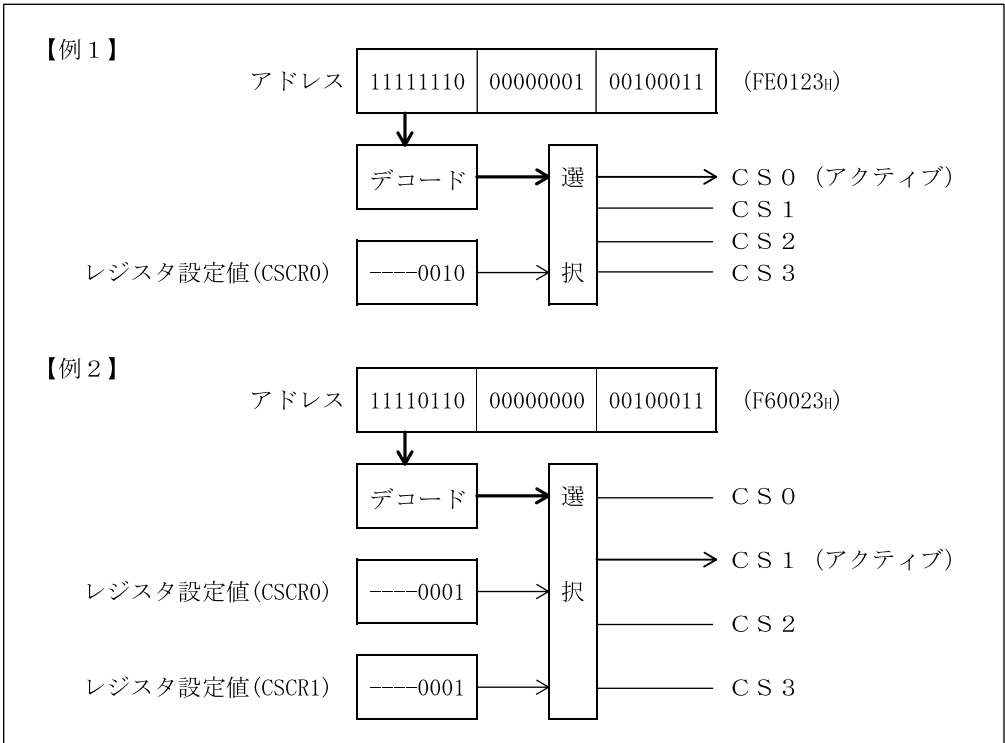


図 22.3-1 チップセレクト機能の動作例

22.4 チップセレクト機能のデコードアドレス空間

表 22.4-1にチップセレクト機能のデコードアドレス空間一覧を、図 22.4-1にデコードアドレス空間マップを、図 22.4-2にリセット解除後のCS0出力を示します。

チップセレクト機能のデコードアドレス空間一覧

表 22.4-1 デコードアドレス空間一覧

端子名	CSA1	CSA0	デコード空間	領域バイト数	備考
CS0	0	0	F00000 _H ~ FFFFFFF _H	1Mbyte	プログラム領域，プログラムベクタフェッチ時にアクティブとなります。
	0	1	F80000 _H ~ FFFFFFF _H	512Kbyte	
	1	0	FE0000 _H ~ FFFFFFF _H	128Kbyte	
	1	1	-	使用禁止	
CS1	0	0	E00000 _H ~ EFFFFFF _H	1Mbyte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	F00000 _H ~ F7FFFF _H	512Kbyte	
	1	0	FC0000 _H ~ FDFFFF _H	128Kbyte	
	1	1	68FF80 _H ~ 68FFFF _H	128byte	
CS2	0	0	003000 _H ~ 003FFF _H	4Kbyte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	FA0000 _H ~ FBFFFF _H	128Kbyte	
	1	0	68FF80 _H ~ 68FFFF _H	128byte	
	1	1	68FF00 _H ~ 68FF7F _H	128byte	
CS3	0	0	F80000 _H ~ F9FFFF _H	128Kbyte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	68FF00 _H ~ 68FF7F _H	128byte	
	1	0	68FE80 _H ~ 68FEFF _H	128byte	
	1	1	-	使用禁止	
CS4	0	0	002800 _H ~ 002FFF _H	2Kbyte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	68FE80 _H ~ 68FEFF _H	128byte	
	1	0	-	使用禁止	
	1	1	-	使用禁止	
CS5	0	0	68FF80 _H ~ 68FFFF _H	128byte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	-	使用禁止	
	1	0	-	使用禁止	
	1	1	-	使用禁止	
CS6	0	0	68FF00 _H ~ 68FF7F _H	128byte	データROM・RAM接続領域，および外部回路接続用途向き
	0	1	-	使用禁止	
	1	0	-	使用禁止	
	1	1	-	使用禁止	
CS7	*	*	-	使用禁止	使用禁止です。

< 注意事項 >

- ・モード端子を外ROMと設定した場合のCS0 端子は，アドレスF00000_H ~ FFFFFFF_H (1Mbyte) の空間 (プログラムROM用領域) において，リセット直後 (初期状態) にプログラムベクタフェッチのためにデコード信号を出力しますので，本端子は必ずプログラムROM専用端子としてご使用ください。なお本端子のアクティブレベルは" L "出力となります。
- ・初期状態においてチップセレクトデコード信号端子は，ポート機能で入力状態となっていますので，各設定レジスタを書き換えてご使用ください。

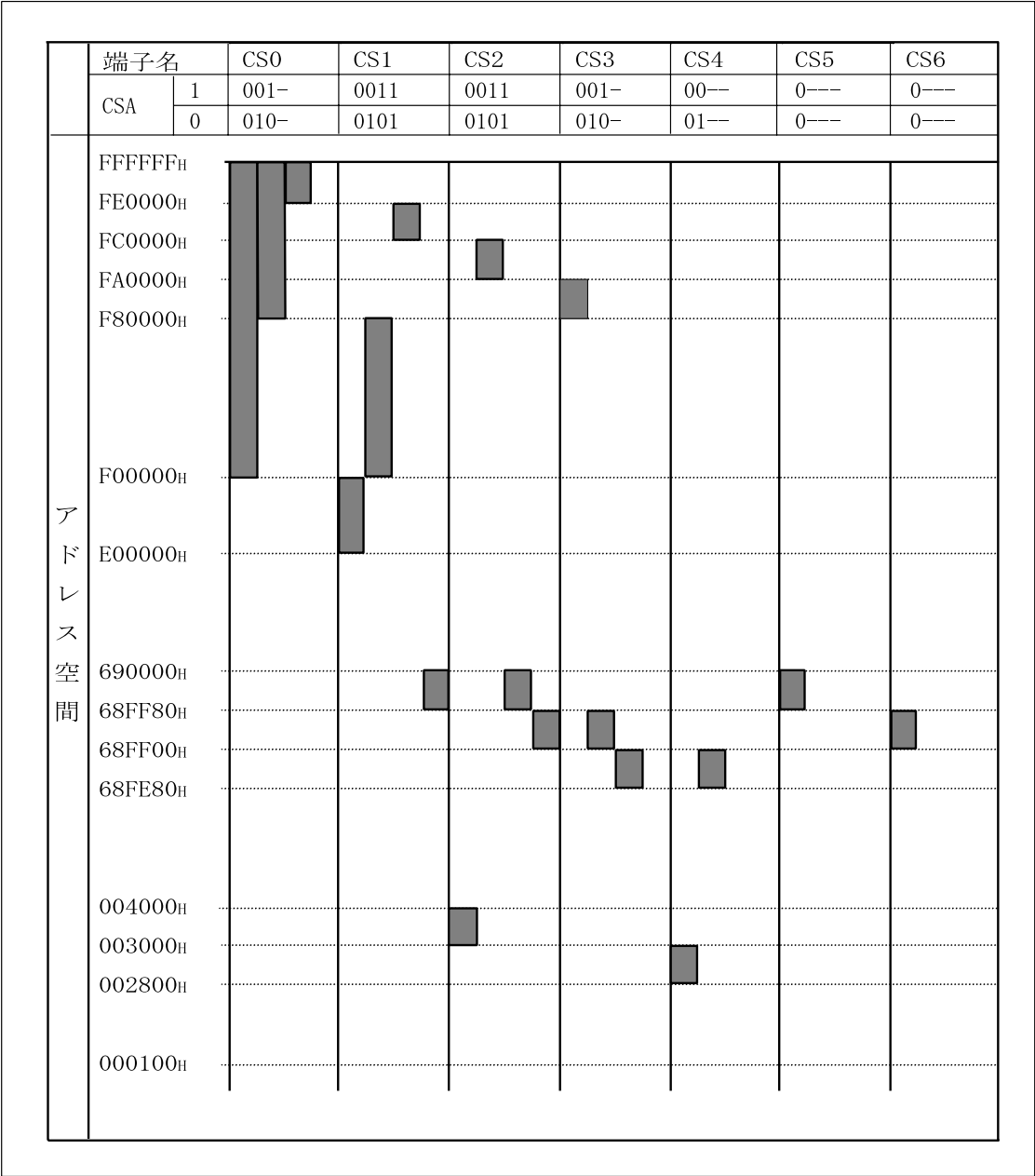


図 22.4-1 デコードアドレス空間マップ

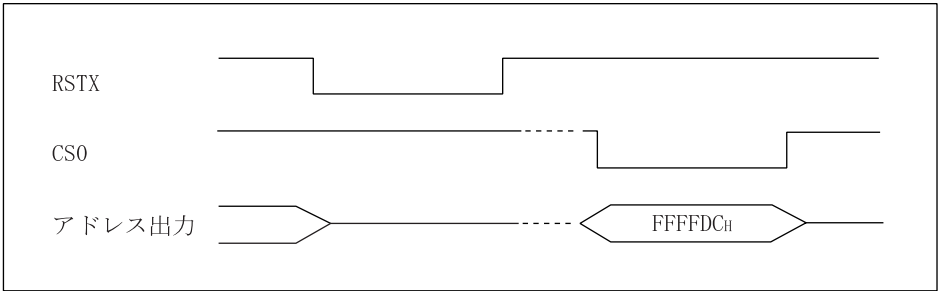


図 22.4-2 リセット解除後のCS0出力

第23章 クロックモニタ機能

この章では、クロックモニタの機能と動作について説明します。

23.1 クロックモニタ機能の概要

23.2 クロック出力許可レジスタ(CLKR)

23.1 クロックモニタ機能の概要

クロックモニタ機能は、CK0T端子からマシンのクロックの分周クロック（モニタ用のクロック）を出力します。

クロックモニタ機能のブロックダイアグラム

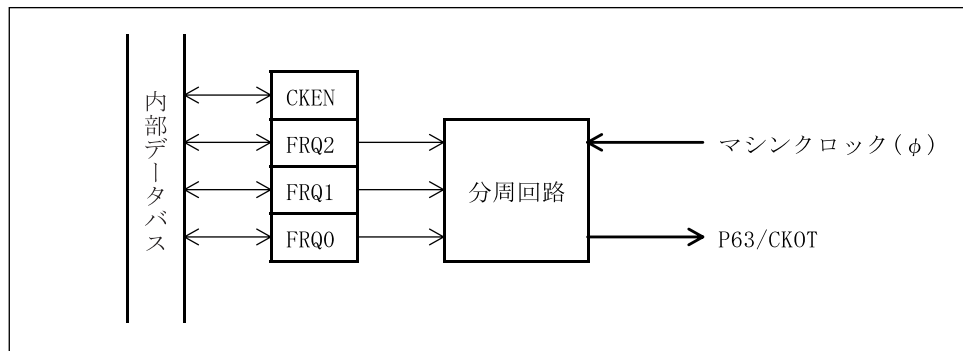


図 23.1-1 クロックモニタ機能のブロックダイアグラム

23.2 クロック出力許可レジスタ(CLKR)

クロック出力許可レジスタ(CLKR)は、CKOT出力許可やクロック出力周波数を選択します。

クロック出力許可レジスタ(CLKR)

クロック出力許可レジスタ	7	6	5	4	3	2	1	0	← ビットNo.
アドレス:0003Eh	—	—	—	—	CKEN	FRQ2	FRQ1	FRQ0	CLKR
リード/ライト ⇒	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(-)	(-)	(-)	(0)	(0)	(0)	(0)	

図 23.2-1 クロック出力許可レジスタ(CLKR)

【ビット7,6,5】未使用ビット

【ビット3】CKEN

CKENは、CKOT出力許可ビットです。

表 23.2-1 CKENビットの機能

CKEN	機 能
0	通常ポートとなります。
1	CKOT出力となります。

【ビット2,1,0】FRQ2, FRQ1, FRQ0

クロック出力周波数を選択するビットです。

表 23.2-2 FRQ2, FRQ1, FRQ0(クロック出力周波数を選択ビット)の機能

FRQ2	FRQ1	FRQ0	出力クロック	=16MHz	= 8MHz	= 4MHz
0	0	0	2 ¹ /	125ns	250ns	500ns
0	0	1	2 ¹ /	250ns	500ns	1 μs
0	1	0	2 ³ /	500ns	1 μs	2 μs
0	1	1	2 ⁴ /	1 μs	2 μs	4 μs
1	0	0	2 ⁵ /	2 μs	4 μs	8 μs
1	0	1	2 ⁶ /	4 μs	8 μs	16 μs
1	1	0	2 ⁷ /	8 μs	16 μs	32 μs
1	1	1	2 ⁸ /	16 μs	32 μs	64 μs

第24章 アドレス一致検出機能

この章は、アドレス一致検出の機能と動作について説明します。

- 24.1 アドレス一致検出機能の概要
- 24.2 アドレス一致検出機能のレジスタ
- 24.3 アドレス一致検出機能の動作
- 24.4 アドレス一致検出機能の使用例
- 24.5 プログラムパッチ処理例とフロー

24.1 アドレス一致検出機能の概要

アドレスが、アドレス検出レジスタに設定された値と等しい場合に、CPUに読み込まれる命令コードを強制的にINT9命令のコード(01H)に置き換えます。その結果、CPUが設定された命令を実行する時に、INT9命令を実行します。INT #9割り込みルーチンで処理を行うことにより、プログラムのパッチ当て機能を実現することができます。

アドレス検出レジスタは2本用意されており、各レジスタごとに割り込み許可ビットがあります。割り込み許可ビットが"1"の場合、アドレス検出レジスタに設定された値とアドレスを比較します。アドレスが一致した場合は、CPUに読み込まれる命令コードを強制的にINT9命令のコードに置き換えます。

アドレス一致検出機能のブロックダイアグラム

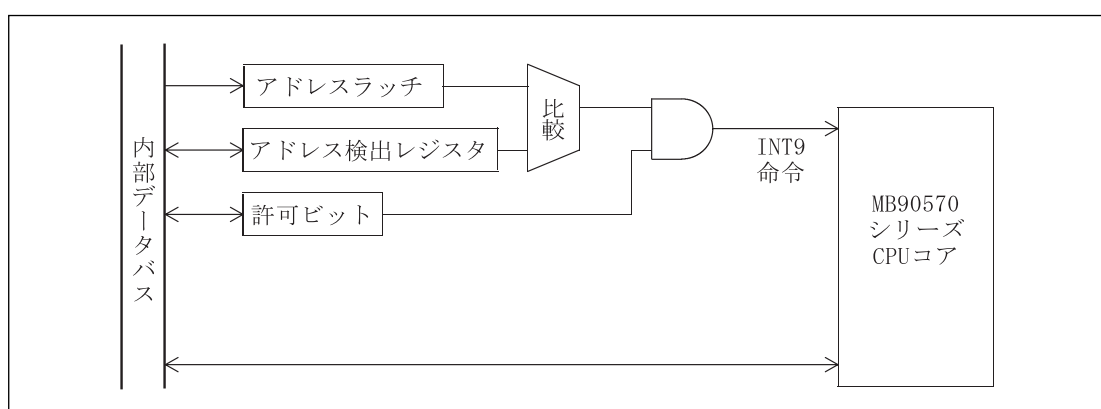


図 24.1-1 アドレス一致検出機能のブロックダイアグラム

< 注意事項 >

アドレス検出レジスタに、命令の1バイト目のアドレス以外を設定した場合は、この機能は正しく動作しません。設定したアドレスのデータが"01H"に替わることになるので、違う命令を実行したり、違うアドレスをアクセスすることになります。

アドレス検出レジスタの変更は割り込み許可ビットを"0"にした状態で行ってください。割り込み許可ビットが"1"の状態を書き込んだ場合は、書き込みの途中で誤ってアドレス検出を行ってしまい、誤動作を引き起こす可能性があります。

24.2 アドレス一致検出機能のレジスタ

アドレス一致検出機能には、次の2種類のレジスタがあります。

- ・プログラムアドレス検出レジスタ(PADRO,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)

アドレス一致検出機能のレジスタ

・プログラムアドレス検出レジスタ									
		bit23	bit16	bit15	bit8	bit7	bit0	アクセス	初期値
PADRO	アドレス:1FF2 _H /1FF1 _H /1FF0 _H							R/W	不定
		bit23	bit16	bit15	bit8	bit7	bit0		
PADR1	アドレス:1FF5 _H /1FF4 _H /1FF3 _H							R/W	不定
・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									
		7	6	5	4	3	2	1	0 ← ビットNo.
アドレス:009E _H		予約	予約	予約	予約	AD1E	予約	AD0E	予約
リード/ライト⇒		(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒		(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
		PACSR							

図 24.2-1 アドレス一致検出機能のレジスタ

24.2.1 プログラムアドレス検出レジスタ(PADRO,1)

プログラムアドレス検出レジスタ(PADRO,1)は、PACSRレジスタの対応する割込み許可ビットが"1"の場合、各レジスタに書込まれた値とアドレスを比較します。アドレスが一致した場合は、CPUに対してINT9命令の発生を要求します。
対応する割込み許可ビットが"0"の場合は、何も行いません。

プログラムアドレス検出レジスタ(PADRO,1)

・プログラムアドレス検出レジスタ						
	bit23	bit16	bit15	bit8	bit7	bit0
PADRO アドレス:1FF2 _H /1FF1 _H /1FF0 _H						
	bit23	bit16	bit15	bit8	bit7	bit0
PADR1 アドレス:1FF5 _H /1FF4 _H /1FF3 _H						

図 24.2-2 プログラムアドレス検出レジスタ(PADRO/PADR1)

表 24.2-1に、PADRO/PADR1レジスタとPACSRレジスタとの対応を示します。

表 24.2-1 PADRO/1レジスタとPACSRレジスタとの対応

アドレス検出レジスタ (PADRO/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)
	割込み許可ビット
PADRO	AD0E
PADR1	AD1E

24.2.2 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、アドレス検出機能の動作を制御します。

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)

・ 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								
	7	6	5	4	3	2	1	0 ← ビットNo.
アドレス: 009Eh	予約	予約	予約	予約	AD1E	予約	AD0E	予約 PACSR
リード/ライト⇒	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)
初期値⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

図 24.2-3 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)

【ビット7～4】予約ビット

予約ビットです。必ず"0"を設定してください。

【ビット3】AD1E(Address Detect register 1 Enable)

PADR1の割込み許可ビットです。

このビットが"1"の場合、PADR1レジスタとアドレスの比較を行い、アドレスが一致した場合は、INT9命令を発生します。

【ビット2】予約ビット

予約ビットです。必ず"0"を設定してください。

【ビット1】AD0E(Address Detect register 0 Enable)

PADR0の割込み許可ビットです。

このビットが"1"の場合PADR0レジスタとアドレスの比較を行い、アドレスが一致した場合には、INT9命令を発生します。

【ビット0】予約ビット

予約ビットです。必ず"0"を設定してください。

24.3 アドレス一致検出機能の動作

アドレス一致検出機能は、アドレスが、アドレス検出レジスタに設定された値と等しい場合に、CPUに読み込まれる命令コードを強制的にINT9命令のコード(01_H)に置き換えます。その結果、CPUが設定された命令を実行する時に、INT9命令を実行します。INT #9割込みルーチンで処理を行うことにより、プログラムのパッチ当て機能を実現することができます。

アドレス一致検出機能の動作

アドレス検出レジスタは2本用意されており、各レジスタごとに割込み許可ビットがあります。割込み許可ビットが"1"の場合、アドレス検出レジスタに設定された値とアドレスを比較します。アドレスが一致した場合は、CPUに読み込まれる命令コードを強制的にINT9命令のコードに置き換えます。

アドレス一致検出機能の動作上の注意

アドレス検出レジスタに、命令の1バイト目のアドレス以外を設定した場合は、この機能は正しく動作しません。設定したアドレスのデータが"01_H"に替わることになるので、違う命令を実行したり、違うアドレスをアクセスすることになります。

また、アドレス検出レジスタの変更は割込み許可ビットを"0"にした状態で行ってください。割込み許可ビットが"1"の状態で書き込んだ場合は、書き込みの途中で誤ってアドレス検出を行ってしまい、誤動作を引き起こす可能性が有ります。

24.4 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として、システム構成例を示します。

アドレス一致検出機能のシステム構成例

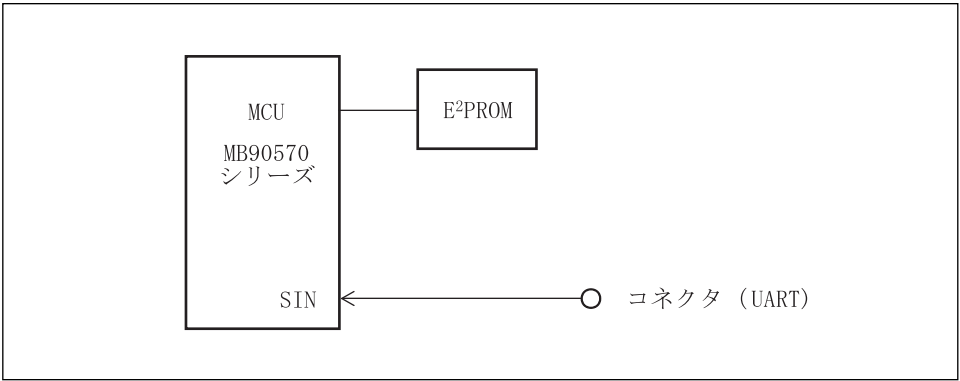


図 24.4-1 アドレス一致検出機能のシステム構成例

E²PROMメモリマップ

表 24.4-1 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無し)
0001 _H	プログラムアドレスNo.0 bit7-0
0002 _H	プログラムアドレスNo.0 bit15-8
0003 _H	プログラムアドレスNo.0 bit24-16
0004 _H	パッチプログラムNo.1バイト数 (0の場合プログラムミス無し)
0005 _H	プログラムアドレスNo.1 bit7-0
0006 _H	プログラムアドレスNo.1 bit15-8
0007 _H	プログラムアドレスNo.1 bit24-16
~ 0010 _H + パッチプログラムNo.0バイト数	パッチプログラムNo.0本体

初期状態

E²PROMはすべて"0"とします。

プログラムミスが発生した場合

コネクタ (UART)を通して、MCUにパッチプログラムの本体およびプログラムアドレスを転送します。MCUはその情報をE²PROMに書き込みます。

リセットシーケンス

MCUはリセット後E²PROMの値を読み出します。パッチプログラムのバイト数が"0"でなかった場合は、パッチプログラムの本体を読み出しRAMに書き出します。プログラムアドレスをPADR0,1レジスタのいずれかに設定して動作を許可します。RAMに書かれたプログラムの先頭番地は、各アドレス検出レジスタごとに定まったRAMに保存します。

INT9割込み

割込みルーチン中では，どのアドレス検出で割込みがかかったかをスタックに退避されたプログラムカウンタ(PC)の値より判断し，割込み要求が出力されたプログラムへ分岐します。プログラムへ分岐した場合は，割込みでスタックされた情報は無効となります。

24.5 プログラムパッチ処理例とフロー

図 24.5-1にプログラムパッチ処理例を、図 24.5-2にプログラムパッチ処理フローを示します。

プログラムパッチ処理例とフロー

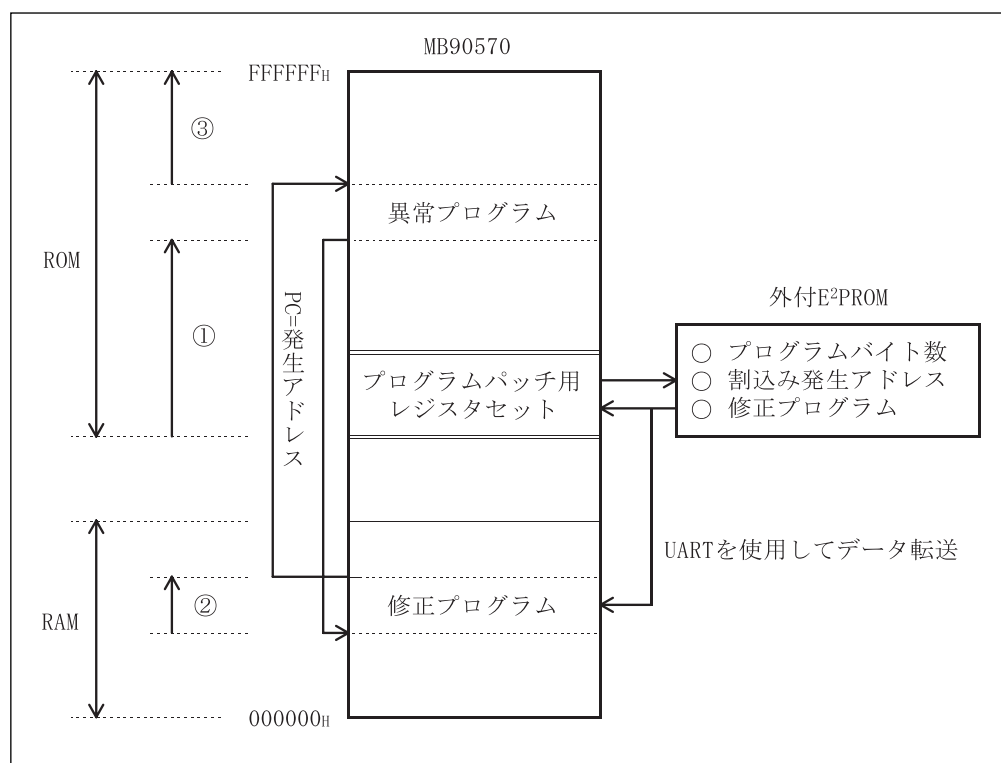


図 24.5-1 プログラムパッチ処理例

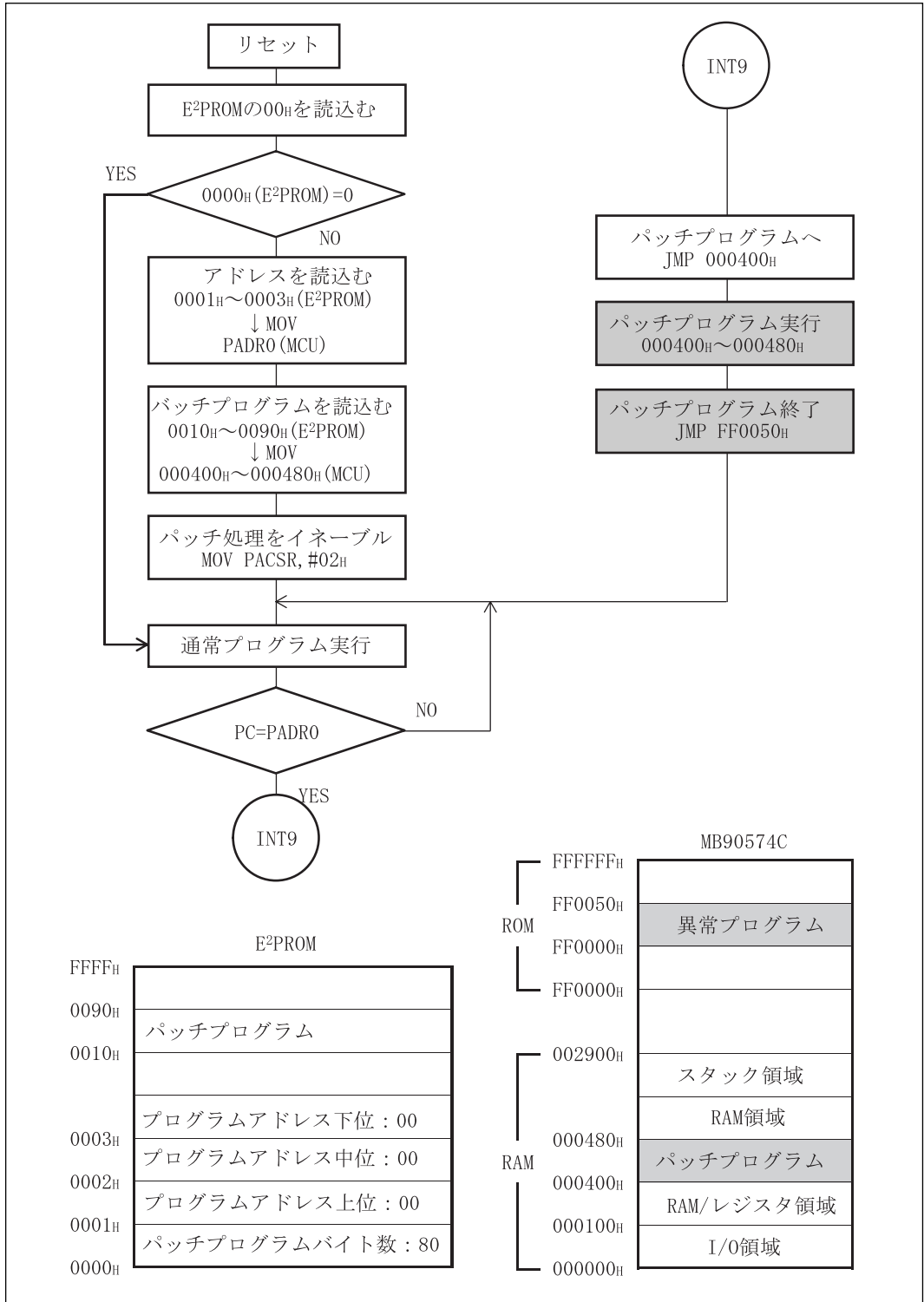


図 24.5-2 プログラムパッチ処理フロー

第25章 ROMミラー機能選択モジュール

この章では，ROMミラー機能選択モジュールの機能と動作について説明します。

25.1 ROMミラー機能選択モジュールの概要

25.2 ROMミラー機能選択レジスタ (ROMM)

25.1 ROMミラー機能選択モジュールの概要

ROMミラー機能選択モジュールは、ROMが配置されているFFバンクが00バンクであることをレジスタの設定で選択することができます。

ROMミラー機能選択モジュールのブロックダイアグラム

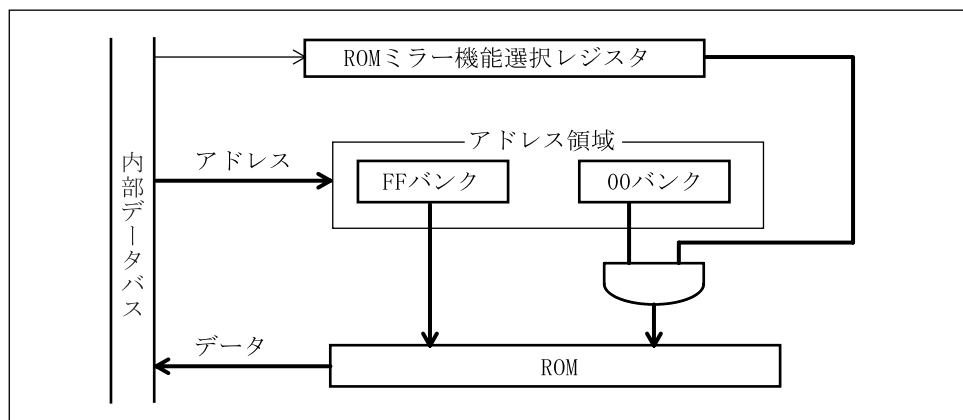


図 25.1-1 ROMミラー機能選択モジュールのブロックダイアグラム

25.2 ROMミラー機能選択レジスタ(ROMM)

ROMミラー機能選択レジスタ(ROMM)について説明します。

ROMミラー機能選択レジスタ(ROMM)

・ROMミラー機能選択レジスタ

	15	14	13	12	11	10	9	8	⇐ ビットNo.
アドレス:00006F _H	—	—	—	—	—	—	—	MI	ROMM
リード/ライト⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(W)	
初期値⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(1)	

図 25.2-1 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番地の動作中にアクセスしないでください。

【ビット15～9】未使用ビット

【ビット8】MI

"1"書込み時には、FFバンクのROMのデータが00バンクでも読み出せるようになります。"0"書込み時には、00バンクでこの機能は働きません。本ビットは、書込みのみ可能です。

図 25.2-2に、シングルチップモード時のメモリ空間を示します。

< 注意事項 >

00バンクは004000_H～00FFFF_H番地のみで、FF4000_H～FFFF_H番地をミラーします。このため、FFF000_H～FF3FFF_H番地はROMのミラー機能を選択してもミラーされません。

表 25.2-1 メモリ空間のアドレス

	MB90573	MB90574/C	MB90F574/A	MB90V570/A
アドレス1	FE0000 _H	FC0000 _H	FC0000 _H	(FC0000 _H)
アドレス2	001800 _H	002900 _H	002900 _H	002900 _H

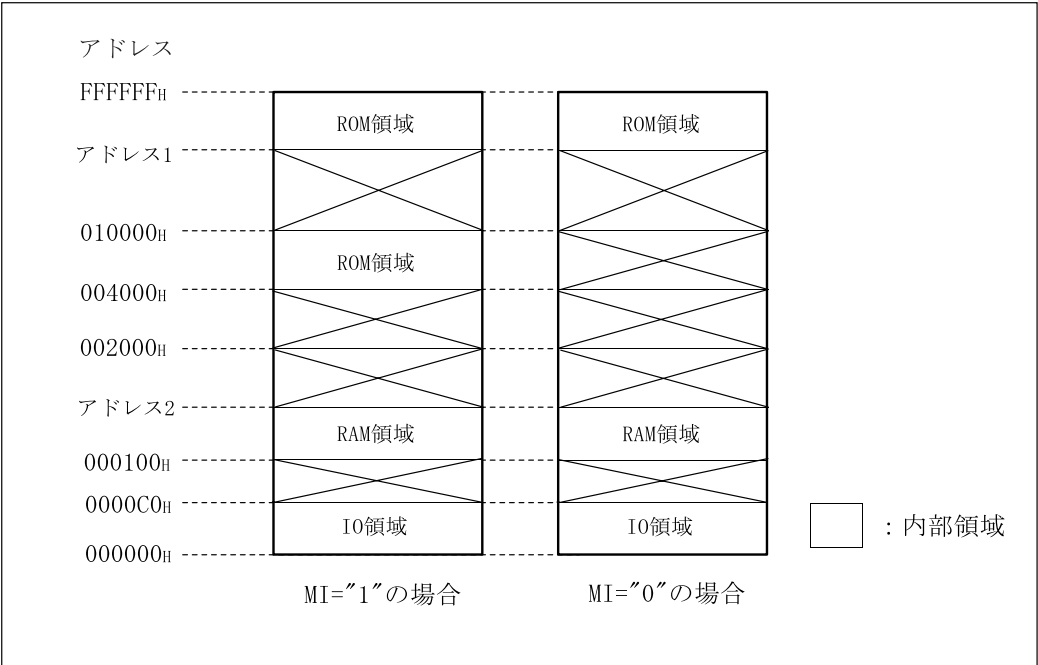


図 25.2-2 シングルチップモード時のメモリ空間

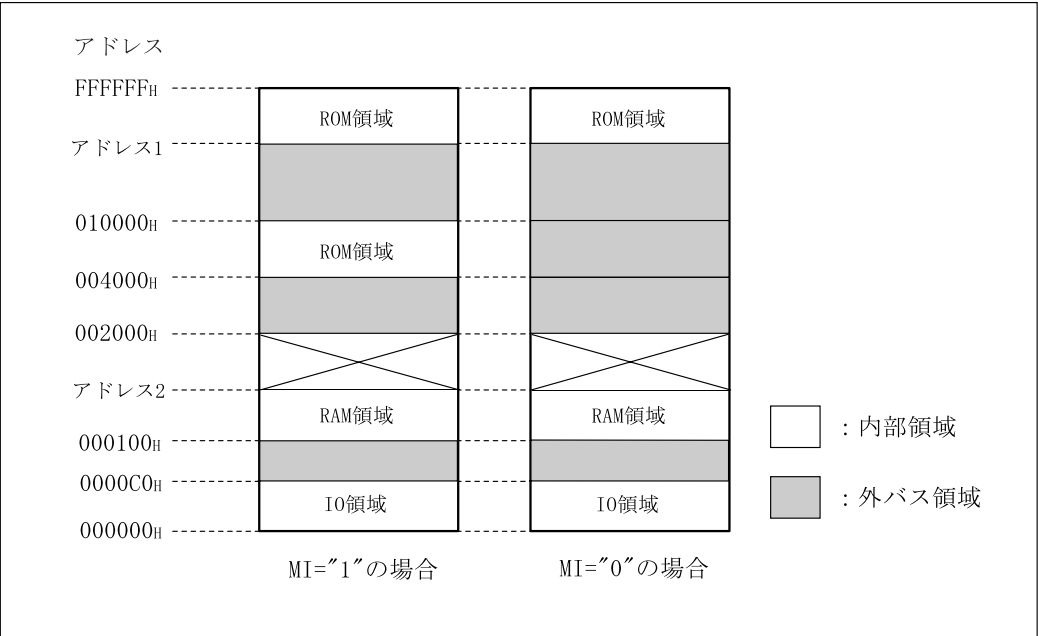


図 25.2-3 内ROM外バスモード時のメモリ空間

第26章 2Mビットフラッシュメモリ

この章では、2M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。
フラッシュメモリへのデータ書込み / 消去の方法には、以下の3とおりがあります。

- 1) パラレルライタ
- 2) シリアル専用ライタ
- 3) プログラム実行による書込み / 消去

ここでは、"3) プログラム実行による書込み / 消去"について説明します。

- 26.1 2Mビットフラッシュメモリの概要
- 26.2 2Mビットフラッシュメモリのセクタ構成
- 26.3 フラッシュメモリコントロールステータスレジスタ (FMCS)
- 26.4 フラッシュメモリ自動アルゴリズムの起動方法
- 26.5 自動アルゴリズム実行状態の確認
- 26.6 フラッシュメモリ書込み / 消去の詳細説明
- 26.7 フラッシュメモリのプログラム例

26.1 2Mビットフラッシュメモリの概要

2Mビットフラッシュメモリは、CPUメモリマップ上のFC～FFバンクに配置され、フラッシュメモリインタフェース回路の機能により、マスクROMと同様にCPUからリードアクセスおよびプログラムアクセスが可能です。

フラッシュメモリへの書込み/消去は、フラッシュメモリインタフェース回路を介してCPUからの命令動作で行えます。このため、内蔵CPUの制御による実装状態での書き換えが可能となり、プログラムおよびデータの改善が効率よく行えます。

2Mビットフラッシュメモリの特長

- 256Kワード×8 / 128Kワード×16ビット (16K+512×2+7K+8K+32K+64K+64K+64K)セクタ構成
- 自動プログラムアルゴリズム (Embedded Algorithm : MBM29F400TAと同様)
- 消去一時停止 / 消去再開機能の搭載
- データポーリング, トグルビットによる書込み / 消去完了検出
- 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
- セクタ毎の消去が可能 (セクタ組み合わせ自由)
- 書込み / 消去回数 (最小) 10,000回

Embedded 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。つまり、フラッシュメモリにデータ書込み／消去動作を行う場合には、フラッシュメモリ上にあるプログラムをRAMにいったんコピーし、RAM実行することで、フラッシュメモリからプログラムアクセスをせず、書込み動作だけを行うことが可能になります。

フラッシュメモリのレジスタ

フラッシュメモリコントロールステータスレジスタ (FMCS)

・フラッシュメモリコントロールステータスレジスタ (FMCS)

	7	6	5	4	3	2	1	0	ビットNo.
アドレス : 0000AEh	INTE	RDYINT	WE	RDY	予約	—	—	LPM	FMCS
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(W)	(W)	(R/W)	
初期値⇒	(0)	(0)	(0)	(1)	(0)	(-)	(-)	(0)	

図 26.1-1 フラッシュメモリコントロールステータスレジスタ (FMCS)

26.2 2Mビットフラッシュメモリのセクタ構成

図 26.2-1に、2Mビットフラッシュメモリのセクタ構成を示します。図中のアドレスには、各セクタの上位アドレスと下位アドレスを示します。

2Mビットフラッシュメモリのセクタ構成

CPUからアクセスする場合、FCバンクレジスタにSA0、FDバンクレジスタにSA1、FEバンクレジスタにSA2、FFバンクレジスタにはSA3～SA8が配置されています。

フラッシュメモリ	CPUアドレス	ライタアドレス*
SA8 (16Kバイト)	FFFFFF _H	7FFFF _H
	FFC000 _H	7C000 _H
SA7 (512バイト)	FFBFFF _H	7BFFF _H
	FFBE00 _H	7BE00 _H
SA6 (512バイト)	FFBDFE _H	7BDFE _H
	FFBC00 _H	7BC00 _H
SA5 (7Kバイト)	FFBBFF _H	7BBFF _H
	FFA000 _H	7A000 _H
SA4 (8Kバイト)	FF9FFF _H	79FFF _H
	FF8000 _H	78000 _H
SA3 (32Kバイト)	FF7FFF _H	77FFF _H
	FF0000 _H	70000 _H
SA2 (64Kバイト)	FEFFFF _H	6FFFF _H
	FE0000 _H	60000 _H
SA1 (64Kバイト)	FDFFFF _H	5FFFF _H
	FD0000 _H	50000 _H
SA0 (64Kバイト)	FCFFFF _H	4FFFF _H
	FC0000 _H	40000 _H

* : ライタアドレスとは、フラッシュメモリにパラレルライタでデータ書込みを行う場合、CPUアドレスに相当するアドレスです。汎用ライタを使用し書込み/消去を行う場合は、このアドレスで書込み/消去を実行します。

図 26.2-1 2Mビットフラッシュメモリのセクタ構成

26.3 フラッシュメモリコントロールステータスレジスタ (FMCS)

コントロールステータスレジスタ(FMCS)は、フラッシュメモリインタフェース回路にあるレジスタで、フラッシュメモリの書込み/消去の際に使用します。

フラッシュメモリコントロールステータスレジスタ(FMCS)

・フラッシュメモリコントロールステータスレジスタ (FMCS)								↔ ビットNo.
アドレス : 0000AE _H								FMCS
	7	6	5	4	3	2	1	0
	INTE	RDYINT	WE	RDY	予約	—	—	LPM
リード/ライト⇒	(R/W)	(R/W)	(R/W)	(R/W)	(W)	(W)	(W)	(R/W)
初期値⇒	(0)	(0)	(0)	(1)	(0)	(—)	(—)	(0)

図 26.3-1 フラッシュメモリコントロールステータスレジスタ (FMCS)

【bit7】 INTE(INTerrupt Enable)

フラッシュメモリの書込み/消去の終了でCPUに割込みを発生させるビットです。

INTEビットが"1"でかつRDYINTビットが"1"の場合に、CPUへ割込みが発生します。INTEビットが"0"であれば割込みは発生しません。

0	書込み/消去終了での割込み禁止
1	書込み/消去終了での割込み許可

【bit6】 RDYINT(ReaDY INTerrupt)

フラッシュメモリの動作状態を表すビットです。

フラッシュメモリの書込み/消去が終って"1"となります。フラッシュメモリの書込み/消去後、このビットが"0"の間は、フラッシュメモリへの書込み/消去はできません。書込み/消去が終了して"1"になった後は、フラッシュメモリへの書込み/消去が可能になります。このビットは"0"書込みによって、"0"にクリアされ、"1"の書込みは無視されます。フラッシュメモリ自動アルゴリズム終了タイミングで、"1"がセットされます(フラッシュメモリ自動アルゴリズムについては「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照してください)。

リードモディファイライト (RMW) 命令使用時は、必ず"1"が読めます。

0	書込み/消去動作実行中
1	書込み/消去動作終了(割込み要求発生)

【bit5】 WE(Write Enable)

フラッシュメモリ領域へのライトイネーブルビットです。

WEビットが"1"の場合、FC～FFバンクへのコマンドシーケンス発行後のライトは、フラッシュメモリ領域へのライトになります(フラッシュメモリ自動アルゴリズムについては「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照してください)。WEビットが"0"の場合、書込み/消去の信号は発生されません。

このビットは、フラッシュメモリの書込み／消去のコマンドを起動する際に使用します。書込み／消去を行わない場合は、誤ってフラッシュメモリにデータを書き込まないように、常に"0"に設定することを推奨します。

0	フラッシュメモリの書込み／消去禁止
1	フラッシュメモリの書込み／消去許可

【bit4】RDY(ReaDY)

フラッシュメモリの書込み／消去許可ビットです。

RDYビットが"0"の間は、フラッシュメモリへの書込み／消去はできません。なお、この状態でも、読出し／リセットコマンド、セクタ消去一時停止などのサスペンドコマンドは受け付けられます。

0	書込み／消去動作実行中
1	書込み／消去動作終了（次データ書込み／消去許可）

【bit3】予約ビット

試験用予約ビットです。通常使用時は必ず"0"に設定してください。

【bit2, 1】空きビット

通常使用時は必ず"0"に設定してください。

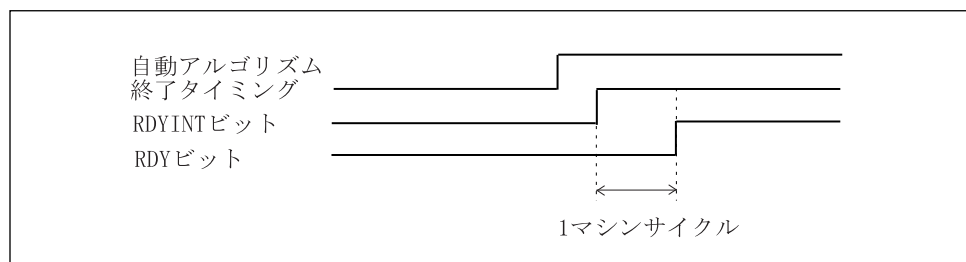
【bit0】LPM(Low Power Mode)

LPMビットを"1"に設定すると、フラッシュメモリアクセス時のフラッシュメモリへのセレクト信号が最小となり、フラッシュメモリ本体の消費電力が抑えられます。しかし、アクセスタイムはLPM="0"の場合に比べて大幅に大きくなりますので、CPUの高速動作時にはメモリアクセスが不可能になります。このモードを使用する場合は、CPUを4MHz以下の周波数で動作させてください。サブクロックモードへ移行する際は、自動的にLPM="1"の状態となりますのでソフトウェアでの設定は不要です。

0	通常消費電力モード
1	低消費電力モード（内部動作周波数4MHz以下で動作）

< 注意事項 >

RDYINTビットとRDYビットは同時には変化しません。どちらか片方のビットで判定するようにプログラムを作成してください。



26.4 フラッシュメモリ自動アルゴリズムの起動方法

フラッシュメモリの自動アルゴリズムを起動するコマンドには、読出し/リセット、書込み、チップ消去の4種類があり、セクタ消去については一時停止と再開の制御が可能です。

コマンドシーケンス表

表 26.4-1に、フラッシュメモリの書込み/消去時に使用するコマンドの一覧を示します。コマンドレジスタにライトするデータはすべてバイトですが、ワードアクセスでライトするようにしてください。この場合、上位バイト分のデータは無視されます。

表 26.4-1 コマンドシーケンス表

コマンド シーケンス	バス ライト サイクル	1stバス サイクル		2ndバス サイクル		3rdバス サイクル		4thバス サイクル		5thバス サイクル		6thバス サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し /リセット ^{*1}	1	FxXXXX	XXF0	-	-	-	-	-	-	-	-	-	-
読出し /リセット ^{*1}	4	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XXF0	RA	RD	-	-	-	-
書込み プログラム	4	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XXA0	PA (even)	PD (word)	-	-	-	-
チップ 消去	6	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX80	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX10
セクタ 消去	6	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XX80	FxAAAA	XXAA	Fx5554	XX55	SA (even)	XX30
セクタ消去一時停止		Address "FxXXXX"Data(xxB0H)の入力で、セクタ消去中の消去一時停止											
セクタ消去再開		Address "FxXXXX"Data(xx30H)の入力で、セクタ消去一時停止後、消去再開											

(注) 表中のアドレスFxは、FF,FE,FD,FCを意味します。それぞれの操作時にはアクセス対象バンクの値としてください。

表中のアドレスは、CPUメモリマップ上の値です。アドレスおよびデータはすべて16進数表記しています。ただし、"X"は任意値です。

RA：読出しアドレス

PA：書込みアドレス、偶数アドレスのみ指定可。

SA：セクタアドレス、「26.2 2Mビットフラッシュメモリのセクタ構成」を参照願います。

RD：読出しデータ

PD：書込みデータ、ワードデータのみ指定可。

*1：2種類の読出し/リセットコマンドは、どちらもフラッシュメモリを読出しモードにリセットすることができます。

26.5 自動アルゴリズム実行状態の確認

フラッシュメモリでは、書込み／消去のフローを自動アルゴリズムで行うため、フラッシュメモリ内部の動作状態や動作完了したことを知らせるハードウェアを持ちます。この自動アルゴリズムは、以下のハードウェアシーケンスによって内蔵フラッシュメモリの動作状態の確認ができます。

ハードウェアシーケンスフラグ

ハードウェアシーケンスフラグは、DQ7、DQ6、DQ5、DQ3の4ビットの出力で構成されます。それぞれデータボーリングフラグ(DQ7)、トグルビットフラグ(DQ6)、タイミングリミット超過フラグ(DQ5)、セクタ消去タイムフラグ(DQ3)の機能を持ちます。これにより、書込み／チップ・セクタ消去終了、消去コードライトが有効かの確認をすることができます。

ハードウェアシーケンスフラグを参照するには、コマンドシーケンス(「26.4 フラッシュメモリ自動アルゴリズムの起動方法」の表 26.4-1を参照)設定後に、フラッシュメモリ内部の対象セクタのアドレスにリードアクセスすることで参照できます。

表 26.5-1に、ハードウェアシーケンスフラグのビット割当てを示します。

表 26.5-1 ハードウェアシーケンスフラグのビット割当て

ビットNo.	7	6	5	4	3	2	1	0
ハードウェア シーケンスフラグ	DQ7	DQ6	DQ5	-	DQ3	-	-	-

自動書込み／チップ・セクタ消去が実行中であるかの判断は、ハードウェアシーケンスフラグを確認するか、フラッシュメモリコントロールレジスタ(FMCS)のRDYビットを確認することで、書込みが終了しているかを知ることができます。書込み／消去終了後は、読出し／リセット状態に戻ります。

実際にプログラムを作成する場合には、いずれかのフラグで自動書込み／消去終了を確認後に、データの読出しなどの次処理を行ってください。また、2回目以降のセクタ消去コードライトが有効であるかについても、ハードウェアシーケンスフラグによって確認することができます。表 26.5-2にハードウェアシーケンスフラグについて説明します。

表 26.5-2 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)	DQ7 DATA:7	Toggle DATA:6	0 DATA:5	0 DATA:3
	チップ・セクタ消去動作 消去完了	0 1	Toggle Stop	0 1	1
	セクタ消去ウェイト 消去開始	0	Toggle	0	0 1
	消去動作 セクタ消去一時停止 (消去中のセクタ)	0 1	Toggle 1	0	1 0
	セクタ消去一時停止 消去再開 (消去中のセクタ)	1 0	1 Toggle	0	0 1
	セクタ消去一時停止中 (消去中でないセクタ)	DATA:7	DATA:6	DATA:5	DATA:3
異常動作	書込み動作	DQ7	Toggle	1	0
	チップ・セクタ消去動作	0	Toggle	1	1

次項で各ハードウェアシーケンスフラグについて説明します。

26.5.1 データポーリングフラグ(DQ7)

データポーリングフラグは、自動アルゴリズム実行が進行中、もしくは終了状態であることをデータポーリング機能によって知らせるためのフラグです。

書込み動作時

自動書込みアルゴリズム実行中にリードアクセスすると、フラッシュメモリはアドレスの指し示す番地によらず、最後に書き込まれたデータのビット7の反転データを出力します。自動書込みアルゴリズム終了時にリードアクセスすると、フラッシュメモリはアドレスの指し示す番地の読出し値のビット7を出力します。

チップ消去 / セクタ消去動作時

チップ消去 / セクタ消去アルゴリズム実行中にリードアクセスすると、チップ消去時はアドレスの指し示す番地に関係なく、セクタ消去時は現在消去しているセクタから、フラッシュメモリは"0"を出力します。同様に終了時には"1"を出力します。

セクタ消去一時停止時

セクタ消去一時停止時にリードアクセスすると、フラッシュメモリはアドレスの指し示す番地が消去中であれば"1"を出力し、消去中のセクタでなければアドレスの指し示す番地の読出し値のビット7(DATA:7)を出力します。トグルビットフラグ(DQ6)とともに参照することで、現在、セクター一時停止状態であるか、どのセクタが消去中であるかの判定が可能です。

データポーリングフラグの状態遷移

表 26.5-3と表 26.5-4に、データポーリングフラグの状態遷移を示します。

表 26.5-3 正常動作時のデータポーリングフラグの状態変化

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 26.5-4 異常動作時のデータポーリングフラグの状態変化

動作状態	書込み動作	チップ・セクタ 消去動作
DQ7	$\overline{\text{DQ7}}$	0

< 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出力が可能となります。このため、自動アルゴリズム終了後のデータ読出しは、データポーリングを確認したリードアクセスの次に行うようにしてください。

26.5.2 トグルビットフラグ(DQ6)

トグルビットフラグはデータポーリングフラグと同様に，主に自動アルゴリズム実行が進行中，もしくは終了状態であることをトグルビット機能によって知らせるためのフラグです。

書込み / チップ・セクタ消去時

自動書込みアルゴリズムおよびチップ・セクタ消去アルゴリズム実行中に，連続したリードアクセスを行うと，フラッシュメモリはアドレスの指し示す番地によらず，リードごとに"1"と"0"を交互に出力するトグル状態を出力します。自動書込みアルゴリズムおよびチップ・セクタ消去アルゴリズム終了時に連続したリードアクセスを行うとフラッシュメモリはビット6のトグル動作を止め，アドレスの指し示す番地の読出し値のビット6(DATA:6)を出力します。

セクタ消去一時停止時

セクタ消去一時停止時にリードアクセスすると，フラッシュメモリはアドレスの指し示す番地が消去中のセクタに属するならば，"1"を出力します。消去中のセクタに属さないのであれば，アドレスの指し示す番地の読出し値のビット6(DATA:6)を出力します。

参考：

書込みの際，書き込もうとしているセクタが書き換え保護されているセクタの場合は，約2 μ sのトグル動作をした後，データを書き換えることなくトグル動作を終わります。
消去の際，選択されたすべてのセクタが書き換え保護されている場合，トグルビットは約100 μ sのトグル動作をし，その後データを書き換えしないで読出し / リセット状態に戻ります。

トグルビットフラグの状態遷移

表 26.5-5と表 26.5-6に，トグルビットフラグの状態遷移を示します。

表 26.5-5 正常動作時のトグルビットフラグの状態変化

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 26.5-6 異常動作時のトグルビットフラグの状態変化

動作状態	書込み動作	チップ・セクタ 消去動作
DQ6	Toggle	Toggle

26.5.3 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）を越えてしまったことを知らせるフラグです。

書込み / チップ・セクタ消去時

書込みまたはチップ・セクタ消去自動アルゴリズム起動後にリードアクセスすると、規定時間（書込み / 消去に要する時間）内であれば"0"を、規定時間を超えてしまっている場合は"1"を出力します。これは、自動アルゴリズムが実行中か、終了状態にあるかは無関係ですので、書込み / 消去が成功したか、失敗したかの判定が可能です。すなわち、このフラグが"1"を出力した場合、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まだ実行中であれば、書込みが失敗していると判断することができます。

例えば、"0"が書き込まれているフラッシュメモリアドレスに"1"を書き込もうとするとフェイルが発生します。この場合フラッシュメモリはロックされ、自動アルゴリズムは終了しません。したがって、データポーリングフラグ(DQ7)から有効なデータが出力されません。また、トグルビットビットフラグ(DQ6)はトグル動作を止めず、タイムリミットを越え、タイミングリミット超過フラグ(DQ5)は"1"を出力します。この状態は、フラッシュメモリが不良ではなく、正しく使用されなかったということを表しています。この状態が発生した場合は、リセットコマンドを実行してください。

タイミングリミット超過フラグの状態遷移

表 26.5-7と表 26.5-8に、タイミングリミット超過フラグの状態遷移を示します。

表 26.5-7 正常動作時のタイミングリミット超過フラグの状態変化

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 26.5-8 異常動作時のタイミングリミット超過フラグの状態変化

動作状態	書込み動作	チップ・セクタ 消去動作
DQ5	1	1

26.5.4 セクタ消去タイマフラグ(DQ3)

セクタ消去タイマフラグは，セクタ消去コマンド起動後，セクタ消去ウェイト期間中であるか否かを知らせるフラグです。

セクタ消去動作時

セクタ消去コマンド起動後にリードアクセスすると，フラッシュメモリはコマンドを発行したセクタのアドレス信号の指し示す番地によらず，セクタ消去ウェイト期間中であれば"0"を，セクタ消去ウェイト期間を超えてしまっている場合は"1"を出力します。

データポーリング機能やトグルビット機能により消去アルゴリズムが実行中を示している場合，このフラグが"1"であれば内部で制御される消去が始まっています。続けてのセクタ消去コードのライト，または消去一時停止以外のコマンドは，消去が終了されるまで無視されます。

このフラグが"0"であればフラッシュメモリは，追加のセクタ消去コードのライトを受け付けます。このことを確認するために，引き続きセクタ消去コードのライトに先立ち，このフラグの状態をチェックすることを推奨します。もし，2回目のチェックで"1"であったなら追加セクタの消去コードは受け付けられない可能性があります。

セクタ消去動作時

セクタ消去一時停止中にリードアクセスすると，フラッシュメモリはアドレスの指し示す番地が，消去中のセクタに属するならば"1"を出力します。消去中のセクタに属さないのであれば，アドレスの指し示す番地の読出し値のビット3(DATA:3)を出力します。

セクタ消去タイマフラグの状態遷移

表 26.5-9と表 26.5-10に，セクタ消去タイマフラグの状態遷移を示します。

表 26.5-9 正常動作時のセクタ消去タイマフラグの状態変化

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 26.5-10 異常動作時のセクタ消去タイマフラグの状態変化

動作状態	書込み動作	チップ・セクタ 消去動作
DQ3	0	1

26.6 フラッシュメモリ書込み／消去の詳細説明

ここでは、自動アルゴリズムを起動するコマンドを発行し、フラッシュメモリに読出し／リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、セクタ消去再開のそれぞれの動作を行う手順を説明します。

フラッシュメモリ書込み／消去の詳細説明

フラッシュメモリは読出し／リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、セクタ消去再開の動作がコマンドシーケンス（「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照）のバスへのライトサイクルを行うことで、自動アルゴリズムを実行することが可能です。それぞれのバスへのライトサイクルは、必ず続けて行う必要があります。また、自動アルゴリズムはデータポーリング機能などで終了時を知ることができます。正常終了後は、読出し／リセット状態に戻ります。

各動作について、次項より以下の順に説明します。

- 1) 読出し／リセット状態にする（「26.6.1項」を参照）
- 2) データ書込みを行う（「26.6.2項」を参照）
- 3) 全データ消去を行う（チップ消去）（「26.6.3項」を参照）
- 4) 任意データ消去を行う（セクタ消去）（「26.6.4項」を参照）
- 5) セクタ消去を一時停止する（「26.6.5項」を参照）
- 6) セクタ消去を再開する（「26.6.6項」を参照）

26.6.1 フラッシュメモリの読出し／リセット状態

ここでは、読出し／リセットコマンドを発行し、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。

フラッシュメモリの読出し／リセット状態

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（「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照）の読出し／リセットコマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。

読出し／リセットコマンドには、1回と3回のバス動作を行う2通りのコマンドシーケンスがありますが、これらの本質的な違いはありません。

読出し／リセット状態は、フラッシュメモリの初期状態であり、電源投入時、コマンドの正常終了時は常に読出し／リセット状態になります。読出し／リセット状態は、他のコマンドの入力待ち状態です。

読出し／リセット状態では、通常のリードアクセスでデータを読出せます。マスクROMと同様にCPUからのプログラムアクセスが可能です。通常読出しでのデータ読出しに、このコマンドは必要ありません。このコマンドは、主に何らかの理由でコマンドが正常に終了しなかった場合など、自動アルゴリズムを初期化する場合に使用します。

26.6.2 フラッシュメモリへのデータ書込み

ここでは、書込みコマンドを発行し、フラッシュメモリにデータ書込みを行う手順について説明します。

フラッシュメモリへのデータ書込み

フラッシュメモリのデータ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（「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照）の書込みコマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。4サイクル目に目的のアドレスへのデータライトが終了した時点で、自動アルゴリズムが起動され、自動書込みを開始します。

アドレス指定方法

書込みデータサイクルの中で指定する書込みアドレスは、偶数アドレスのみが可能です。奇数アドレスを指定すると正しく書き込むことができません。つまり、偶数アドレスへのワードデータ単位での書込みが必要となります。

書込みはどのようなアドレスの順番でも、また、セクタの境界を超えても可能ですが、1回の書込みコマンドによって書き込まれるデータは1ワードのみです。

データ書込み上の注意

書込みによって、データ"0"をデータ"1"に戻すことはできません。データ"0"にデータ"1"を書き込むと、データポーリングアルゴリズム(DQ7)またはトグル動作(DQ6)が終了せず、フラッシュメモリ素子が不良と判定され、書込み規定時間を超えタイミングリミット超過フラグ(DQ6)がエラーと判定するか、あるいは見かけ上データ"1"が書き込まれたように見えるかのどちらかになります。しかし、読出し/リセット状態でデータを読出すとデータは"0"のままです。消去動作のみが"0"データを"1"にすることができます。

自動書込み実行中は、すべてのコマンドが無視されます。書込み中にハードウェアリセットが起動されると、書き込んでいるアドレスのデータは保証されませんので注意が必要です。

フラッシュメモリの書込み手順

ハードウェアシーケンスフラグ（「26.5 自動アルゴリズム実行状態の確認」を参照）を用いることでフラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。

図 26.6-1にフラッシュメモリの書込み手順の例を示します。ここでは、書込み終了の確認にデータポーリングフラグ(DQ7)を用いています。

フラグチェックのために読み込むデータは、最後に書込みを行ったアドレスからの読込みとなります。

データポーリングフラグ(DQ7)は、タイミングリミット超過フラグ(DQ5)と同時に変わりますので、たとえタイミングリミット超過フラグ(DQ5)が"1"であっても、データポーリングフラグビット(DQ7)は再チェックする必要があります。

トグルビットフラグ(DQ6)でも同様に、タイミング超過フラグビット(DQ5)が"1"に変わるのと同時にトグル動作を止めますので、トグルビットフラグ(DQ6)を再チェックする必要があります。

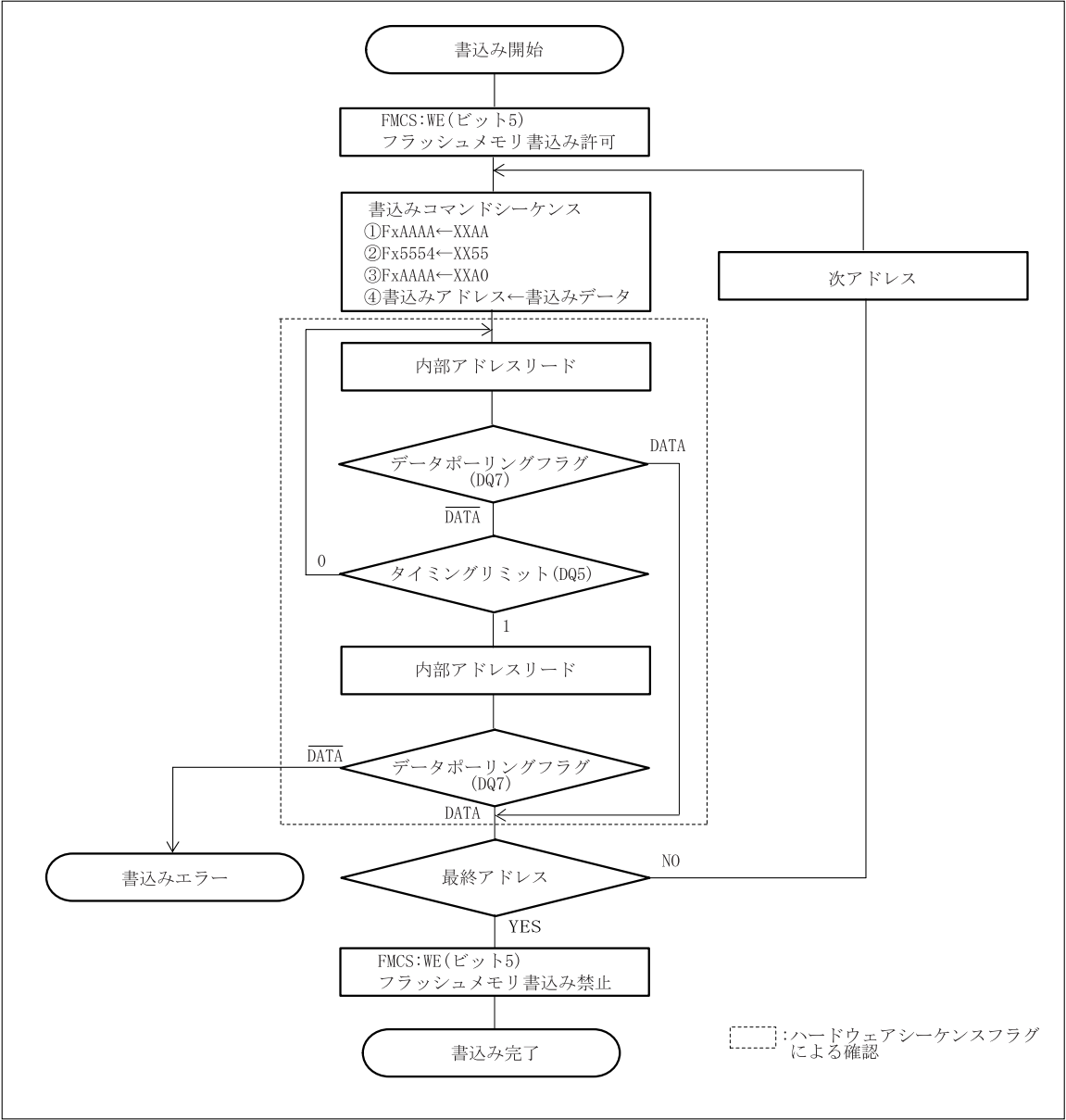


図 26.6-1 フラッシュメモリ書き込み手順の例

26.6.3 フラッシュメモリの全データ消去(チップ消去)

ここでは、チップ消去コマンドを発行し、フラッシュメモリの全データ消去を行う手順について説明します。

フラッシュメモリの全データ消去(チップ消去)

フラッシュメモリからすべてのデータを消去するには、コマンドシーケンス表(「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照)のチップ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。

チップ消去コマンドは6回のバス動作で行われます。6サイクル目のライトが完了した時点で、チップ消去動作が開始します。チップ消去では、消去前にユーザがフラッシュメモリに書込みを行う必要はありません。自動アルゴリズム実行中には、フラッシュメモリは自動的にすべてのセルを消去する前に"0"を書き込んで検証します。

26.6.4 フラッシュメモリの任意データ消去(セクタ消去)

ここでは、セクタ消去コマンドを発行し、フラッシュメモリの任意のセクタ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。

フラッシュメモリの任意データ消去(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(「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照)のセクタ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。

セクタ指定方法

セクタ消去コマンドは6回のバス動作で行われます。6サイクル目に目的のセクタ内のアクセス可能な任意の偶数の1アドレスへ、セクタ消去コード(30H)をライトすることにより50 μ sのセクタ消去ウェイトが開始します。複数のセクタ消去を行う場合は、上述の処理に引き続き消去する目的のセクタ内のアドレスに消去コード(30H)をライトします。

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ライトから50 μ sのセクタ消去ウェイト期間終了により、消去が開始します。つまり、複数のセクタを同時に消去する場合は、次の消去セクタのアドレスと消去コード(コマンドシーケンス 6サイクル目)をそれぞれ50 μ s以内に入力する必要があり、それ以降では受け付けられないことがあります。引き続くセクタ消去コードのライトが有効かどうかはセクタ消去タイマ(ハードウェアシーケンスフラグ DQ3)によって調べることができます。なお、この場合、セクタ消去タイマをリードするアドレスは、消去しようとしているセクタを指すようにします。

フラッシュメモリのセクタ消去手順

ハードウェアシーケンスフラグ(「26.5 自動アルゴリズム実行状態の確認」を参照)を用いることでフラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。

図 26.6-2にフラッシュメモリのセクタ消去手順の例を示します。ここでは、消去終了の確認にトグルビットフラグ(DQ6)を用いています。

フラグチェックのために読み込むデータは、消去しようとしているセクタからの読み込みとなりますので、注意が必要です。

トグルビットフラグ(DQ6)は、タイミングリミット超過フラグ(DQ5)が"1"に変わるのと同時にトグル動作を止めます。たとえタイミングリミット超過フラグ"1"であっても、トグルビットフラグ(DQ6)は再チェックする必要があります。

データポーリングフラグ(DQ7)でも同様に、タイミングリミット超過フラグ(DQ5)と同時に変わりますので、データポーリングフラグ(DQ7)を再チェックする必要があります。

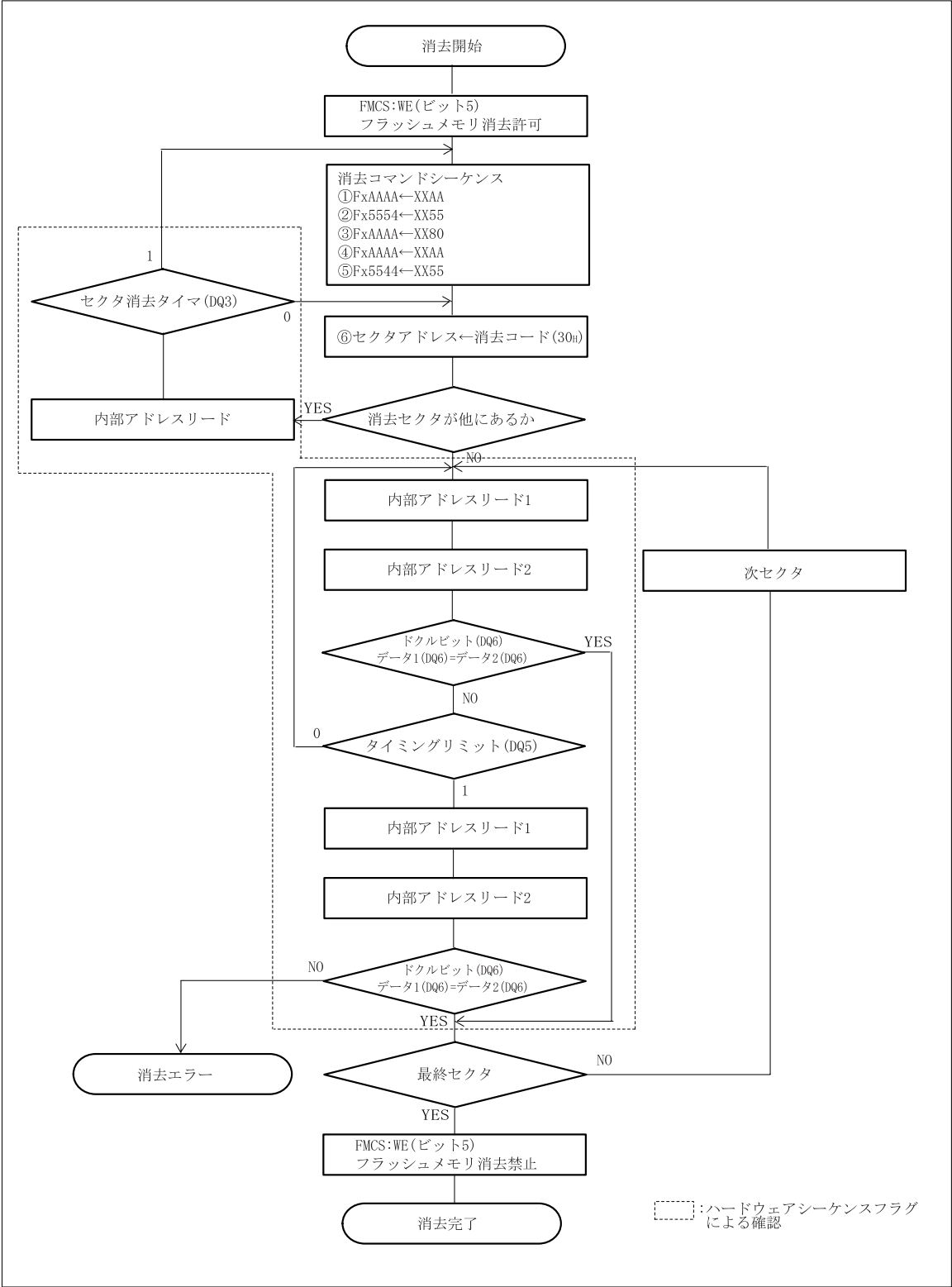


図 26.6-2 フラッシュメモリのセクタ消去手順の例

26.6.5 フラッシュメモリのセクタ消去の一時停止

ここでは、セクタ消去一時停止コマンドを発行し、フラッシュメモリのセクタ消去の一時停止を行う手順について説明します。消去中でないセクタから、データを読み出すことが可能です。

フラッシュメモリのセクタ消去の一時停止

フラッシュメモリのセクタ消去を一時停止するには、コマンドシーケンス表（「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照）のセクタ消去一時停止コマンドを、フラッシュメモリ内に続けて送ることで実行可能です。

セクタ消去一時停止コマンドは、セクタ消去中に消去を一時停止し、消去中でないセクタからのデータ読み出しを可能にするものです。この状態では、読み出しのみが可能で書込みはできません。このコマンドは消去ウェイト時間を含むセクタ消去中のみ有効で、チップ消去中や書込み動作中は無視されます。

消去一時停止コード(B0H)のライトを行うことで実施されますが、この場合アドレスはフラッシュメモリ内の任意のアドレスを指すようにします。消去一時停止での、再度の消去一時停止コマンドは無視されます。

セクタ消去ウェイト期間中に、セクタ消去一時停止コマンドが入力されると、直ちにセクタ消去ウェイトを終了し、消去動作を中断して消去停止状態になります。セクタ消去ウェイト期間後のセクタ消去動作中に、消去一時停止コマンドが入力されると、最大15 μ sの時間の後、消去一時停止状態に入ります。

26.6.6 フラッシュメモリのセクタ消去の再開

ここでは、セクタ消去再開コマンドを発行し、一時停止したフラッシュメモリセクタの消去を再開する手順について説明します。

フラッシュメモリのセクタ消去の再開

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（「26.4 フラッシュメモリ自動アルゴリズムの起動方法」を参照）のセクタ消去再開コマンドを、フラッシュメモリ内に続けて送ることで実行可能です。

セクタ消去再開コマンドは、セクタ消去一時停止コマンドによるセクタ消去一時停止状態から、セクタ消去を再開するためのコマンドです。このコマンドは消去再開コード(30_H)のライトを行うことで実施されますが、この場合のアドレスはフラッシュメモリ領域内の任意のアドレスを指すようにします。

なお、セクタ消去中のセクタ消去再開コマンドの発行は無視されます。

26.7 フラッシュメモリのプログラム例

ここでは、フラッシュメモリのプログラム例を掲載します。

フラッシュメモリのプログラム例

```

NAME    FLASHWE
TITLE   FLASHWE
;-----
;MB90F574/A-FLASH テストプログラム
;
;1: FLASHにあるプログラム(アドレス FFBC00H セクタ SA6)をRAM(アドレス 001500H)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 FD0000Hセクタ SA1)に書き込む。
;4: 書き込んだ値(アドレス FD0000H セクタ SA1)を読み出しPDR2に出力する。
;5: 書き込んだセクタ(SA1)を消去する。
;6: 消去データ確認の出力
;条件
;
;   ・ RAM転送バイト数: 100H(256B)
;   ・ 書き込み, 消去の終了判定
;       DQ5(タイミングリミット超過フラグ)での判定
;       DQ6(トグルビットフラグ)での判定
;       RDY(FMCS)での判定
;   ・ エラー時の処理
;       P00 ~ P07にHiを出力する
;       リセットコマンド発行
;-----
;
RESOUS  IOSEG  ABS=00          ;"RESOUS" I/Oセグメントの定義
        ORG    0000H
PDR0    RB     1
PDR1    RB     1
PDR2    RB     1
PDR3    RB     1
        ORG    0010H
DDR0    RB     1
DDR1    RB     1
DDR2    RB     1
DDR3    RB     1
        ORG    00A1H
CKSCR   RB     1
        ORG    00AEH
FMCS    RB     1
        ORG    006FH
ROMM    RB     1
RESOUS  ENDS
;
SSTA    SSEG
        RW     0127H
STA_T   RW     1
SSTA    ENDS
;
DATA    DSEG   ABS=0FFH       ;FLASHコマンドアドレス
        ORG    5554H
COMADR2 RW     1

```

```

        ORG      0AAAAH
COMADR1 RW      1
DATA     ENDS
;//////////
;メインプログラム(SA3)
;//////////
CODE     CSEG
START:
;
;   初期化
;
;   初期化
MOV      CKSCR,#0BAH      ;3倍に設定
MOV      RP,#0
MOV      A,#!STA_T
MOV      SSB,A
MOVW     A,#STA_T
MOVW     SP,A
MOV      ROMM,#00H        ;ミラーOFF
MOV      PDR0,#00H        ;エラー確認用
MOV      DDR0,#0FFH
MOV      PDR1,#00H        ;データ入力用ポート
MOV      DDR1,#00H
MOV      PDR2,#00H        ;データ出力用ポート
MOV      DDR2,#0FFH
;
;   RAM(1500H番地)に"FLASH書き込み消去プログラム(FFBC00H)"を転送する
;   初期化
MOVW     A,#1500H          ;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001500Hへ100H転送
CALLP    001500H            ;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
;
;   データ出力
OUT      MOV      A,#0FDH
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(SA6)
;//////////
RAMPRG   CSEG      ABS=0FFH
        ORG      0BC00H
;
;   初期化
;
;   初期化
MOVW     RW0,#0500H        ;RW0:入力データ確保用RAM 空間      00:0500~
MOVW     RW2,#0000H        ;RW2:フラッシュメモリ書き込みアドレス  FD:0000~
MOV      A,#00H            ;DTB変更
MOV      DTB,A            ;@RW0用バンク指定
MOV      A,#0FDH          ;ADB変更1
MOV      ADB,A            ;書き込みモード指定アドレス用バンク指定
MOV      PDR3,#00H        ;スイッチ初期化

```

```

        MOV     DDR3,#00H
;
;
WAIT1   BBC     PDR3:0,WAIT1           ;PDR3:0 Hiで書き込みスタート
;
;
;////////////////////////////////////
; 書き込み(SA1)
;////////////////////////////////////
        MOV     A,PDR1
        MOVW    @RW0+00,A             ;RAMにPDR1データを確保
        MOV     FMCS,#20H             ;書き込みモード設定
        MOVW    ADB:COMADR1,#00AAH    ;フラッシュ書き込みコマンド1
        MOVW    ADB:COMADR2,#0055H    ;フラッシュ書き込みコマンド2
        MOVW    ADB:COMADR1,#00A0H    ;フラッシュ書き込みコマンド3
;
;
        MOVW    A,@RW0+00             ;入力データ(RW0)をフラッシュメモリ(RW2)
;                                       ;に書き込む
        MOVW    @RW2+00,A
WRITE   ;待ち時間チェック
;
;   //////////////////////////////////////
;   タイムリミット超過チェック-フラグが立ちトグル動作中である場合 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
;
;   //////////////////////////////////////
;   書き込み終了チェック(FMCS-RDY)
;   //////////////////////////////////////
NTOW    MOVW    A,FMCS
        AND     A,#10H                 ;FMCS RDYビット(4bit)抽出
        BZ      WRITE                 ;書き込み終了か?
        MOV     FMCS,#00H             ;書き込みモード解除
;
;   //////////////////////////////////////
;   書き込みデータ出力
;   //////////////////////////////////////
        MOVW    RW2,#0000H            ;書き込みデータ出力
        MOVW    A,@RW2+00
        MOV     PDR2,A
;
;
WAIT2   BBC     PDR3:1,WAIT2           ;PDR3:1 Hiでセクタ消去スタート
;
;
;////////////////////////////////////
;セクタ消去(SA1)
;////////////////////////////////////
        MOVW    RW2,#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  リード毎Hi Lowが交互出力される
XORW     A                ;AHとALのXOR(DQ6の値が違えば1 書き込み動作
;                          ;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      PDRO,#0FFH        ;エラー処理の確認
MOV      FMCS,#00H        ;FLASHモード解除
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
;

```

第27章 MB90F574/Aシリアル書込み接続例

この章では、株式会社ワイ・ディ・シー製 AF220/AF210/AF120/AF110フラッシュマイコンプログラマを用いた場合の、シリアル書込みの接続例について説明します。

- 27.1 MB90F574/Aシリアル書込み接続の基本構成
- 27.2 シリアル書込み接続例（ユーザ電源使用時）
- 27.3 シリアル書込み接続例（ライターから電源供給時）
- 27.4 フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用時）
- 27.5 フラッシュマイコンプログラマとの最小限の接続例（ライターから電源供給時）

27.1 MB90F574/Aシリアル書込み接続の基本構成

MB90F574/Aでは，フラッシュROMのシリアルオンボード書込み(富士通標準)をサポートしています。その仕様について以下に解説します。

MB90F574/Aシリアル書込み接続の基本構成

富士通標準シリアルオンボード書込みには，株式会社ワイ・ディ・シー製 AF220/AF210/AF120/AF110フラッシュマイコンプログラマを使用します。図 27.1-1 に，F²MC-16LX MB90F574/Aシリアル書込み接続の基本構成を示します。

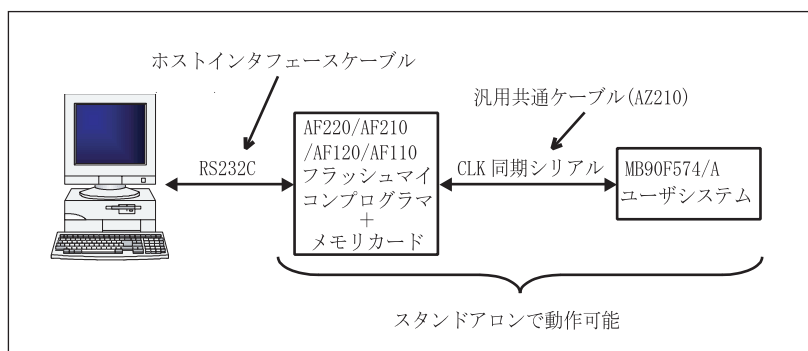


図 27.1-1 MB90F574/Aシリアル書込み接続の基本構成

< 注意事項 >

AF220/AF210/AF120/AF110フラッシュマイコンプログラマの機能，操作方法，および接続用汎用共通ケーブル(AZ210)，コネクタにつきましては，株式会社ワイ・ディ・シー製殿にお問い合わせください。

表 27.1-1 富士通標準シリアルオンボード書込みに使用する端子

端 子	機 能	補足説明
MD2, MD1, MD0	モード端子	フラッシュマイコンプログラマから，書込みモードに制御します。
X0, X1	発振子用端子	書込みモード時に，CPU内部動作クロックはPLLクロック1逓倍となっております。従いまして，発振クロック周波数が，内部動作クロックとなりますので，シリアル書き換え時に使用する発振子は1MHz～16MHzとなります。
P00, P01	書込みプログラム起動端子	-
RSTX	リセット端子	-
SINO	シリアルデータ入力端子	UART0をCLK同期モードとして使用します。
SOTO	シリアルデータ出力端子	
SCK0	シリアルクロック入力端子	
C	C端子	電源安定化の容量端子です。外部に0.1μF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(5V±10%)をユーザシステムから供給する場合にはフラッシュマイコンプログラマとの接続は必要ありません。接続時にはユーザ側の電源と短絡しない様にしてください。
VSS	GND端子	フラッシュマイコンプログラマのGNDと共通にします。
HSTX	ハードウェアスタンバイ端子	シリアル書込みモード中は"H"レベルを入力してください。

なお、P00, SINO, SOTO, SCK0端子をユーザシステムでも使用する場合には、図 27.1-2 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ムの/TICS信号により、シリアル書込み中はユーザ回路を切り離すことができます)。

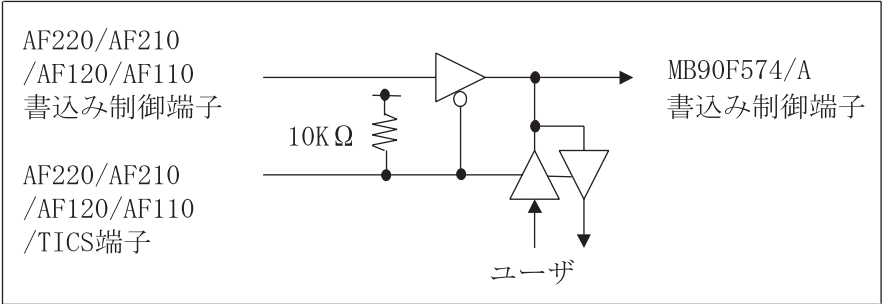


図 27.1-2 制御回路

27.2節以降に、次の4つ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。

- 内部ベクトルモード(シングルチップモード、内ROM外バスモード) ユーザー電源使用時
- 内部ベクトルモード(シングルチップモード、内ROM外バスモード) ライタから電源供給時
- フラッシュマイコンプログラマとの最小限の接続例 ユーザー電源使用時
- フラッシュマイコンプログラマとの最小限の接続例 ライタから電源供給時

表 27.1-2 フラッシュマイコンプログラマシステム構成 (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フェース内蔵モデル /100V～220V 電源アダプタ
	AF210/AC4P	スタンダードモデル /100V～220V 電源アダプタ
	AF120/AC4P	単キーイーサネットインタフェース内蔵モデル /100V～220V 電源アダプタ
	AF110/AC4P	単キーモデル /100V～220V 電源アダプタ
AZ221		ライター専用 PC/AT用RS232Cケーブル
AZ210		標準ターゲットプローブ(a) 長さ：1m
FF201		富士通製 F ² MC-16LX フラッシュマイコン用コントロールモジュール
AZ290		リモートコントローラ
/P2		2MB PC Card(Optional) FLASHメモリ容量 ～128KB対応
/P4		4MB PC Card(Optional) 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は終息製品ですが、コントロールモジュールFF201を用いることにより使用できます。シリアル書込み接続例を27.2節以降に示します。

発振クロック周波数とシリアルクロック入力周波数

MB90F574/Aの入力可能なシリアルクロック周波数は、以下の計算により求められることができます。ご使用の発振クロック周波数に合わせて、シリアルクロック入力周波数をフラッシュマイコンプログラマに設定してください。

入力可能なシリアルクロック周波数=0.125×発振クロック周波数

表 27.1-3 入力可能なシリアルクロック周波数の例

発振クロック 周波数	マイコンの 入力可能な最大シリアル クロック周波数	AF220/AF210/AF120/AF110の 設定可能な最大シリアル クロック周波数	AF200の 設定可能な最大シリアル クロック周波数
4MHz時	500kHz	500kHz	500kHz
8MHz時	1MHz	800kHz	500kHz
16MHz時	2MHz	1.25MHz	500kHz

27.2 シリアル書き込み接続例（ユーザ電源使用時）

図 27.2-1に、マイコンの電源電圧をユーザ電源より供給する場合の、シリアル書き込み接続例を示します。なお、モード端子MD2,MD0には、AF220/AF210/AF120/AF110のTAUX3,TMODEより、MD2=1,MD0=0が入力されます。

シリアル書き換えモード：MD2,MD1,MD0=110

シリアル書き込み接続例（ユーザ電源使用時）

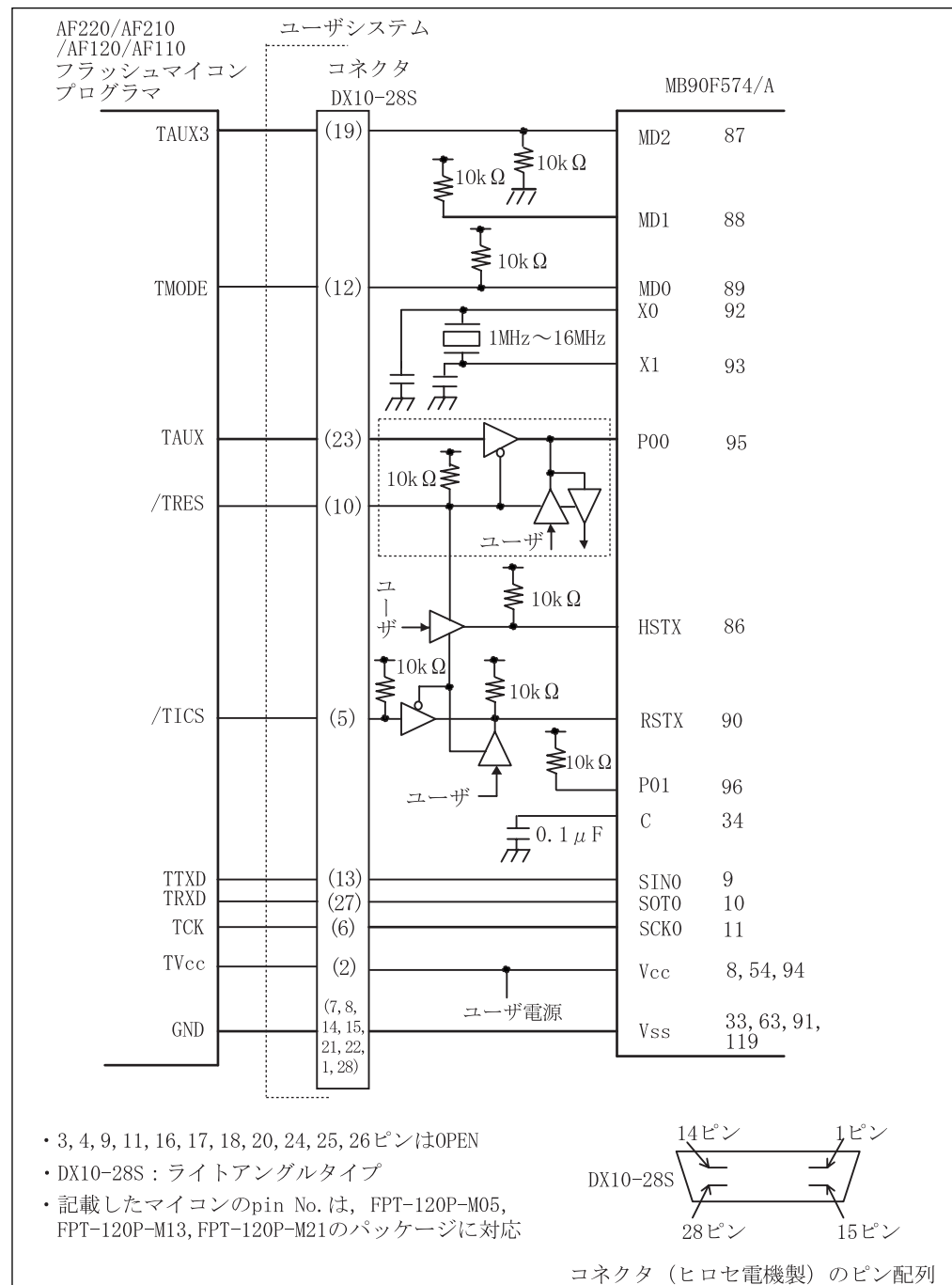
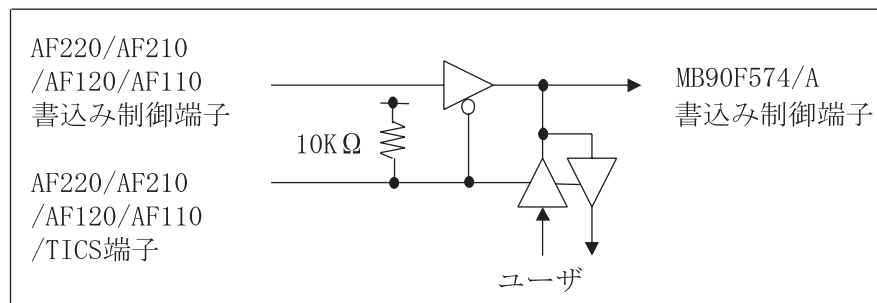


図 27.2-1 MB90F574/A内部ベクトルモード時 シリアル書き込み接続例（ユーザ電源使用時）

- SIN0, SOT0, SCK0端子をユーザシステムでも使用する場合には, P00と同様に, 下図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ムの /TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。

27.3 シリアル書き込み接続例（ライターから電源供給時）

図 27.3-1に、マイコンの電源電圧をライター電源より供給する場合の、シリアル書き込み接続例を示します。なお、モード端子MD2,MD0には、AF220/AF210/AF120/AF110のTAUX3,TMODEより、MD2=1,MD0=0が入力されます。

シリアル書き換えモード：MD2,MD1,MD0=110

シリアル書き込み接続例（ライターから電源供給時）

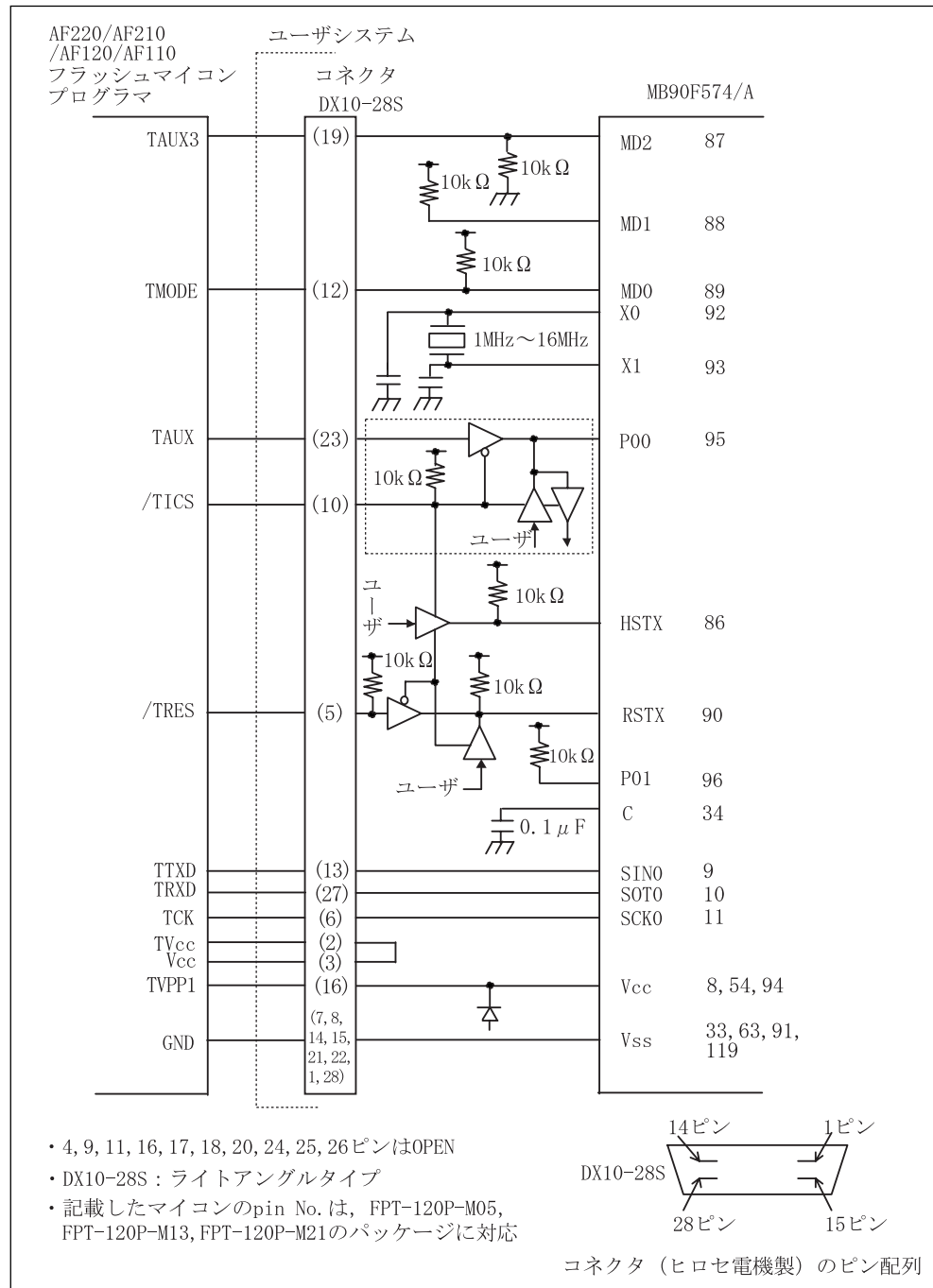
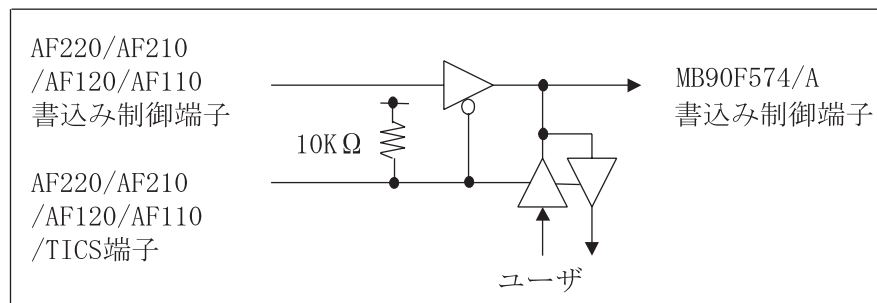


図 27.3-1 MB90F574/A内部ベクトルモード時 シリアル書き込み接続例（ライターから電源供給時）

- SIN0, SOT0, SCK0端子をユーザシステムでも使用する場合には, P00と同様に, 下図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ムの/TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給する場合はユーザ電源と短絡しないでください。

27.4 フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用時）

図 27.4-1に、マイコンの電源電圧をユーザ電源より供給する場合の、フラッシュマイコンプログラマとの最小限の接続例を示します。

シリアル書き換えモード：MD2,MD1,MD0=110

フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用時）

フラッシュメモリ書込み時に、各端子を図 27.4-1に示すように設定して頂ければ、MD2,MD1,MD0,P00とフラッシュマイコンプログラマとの接続は必要ありません。

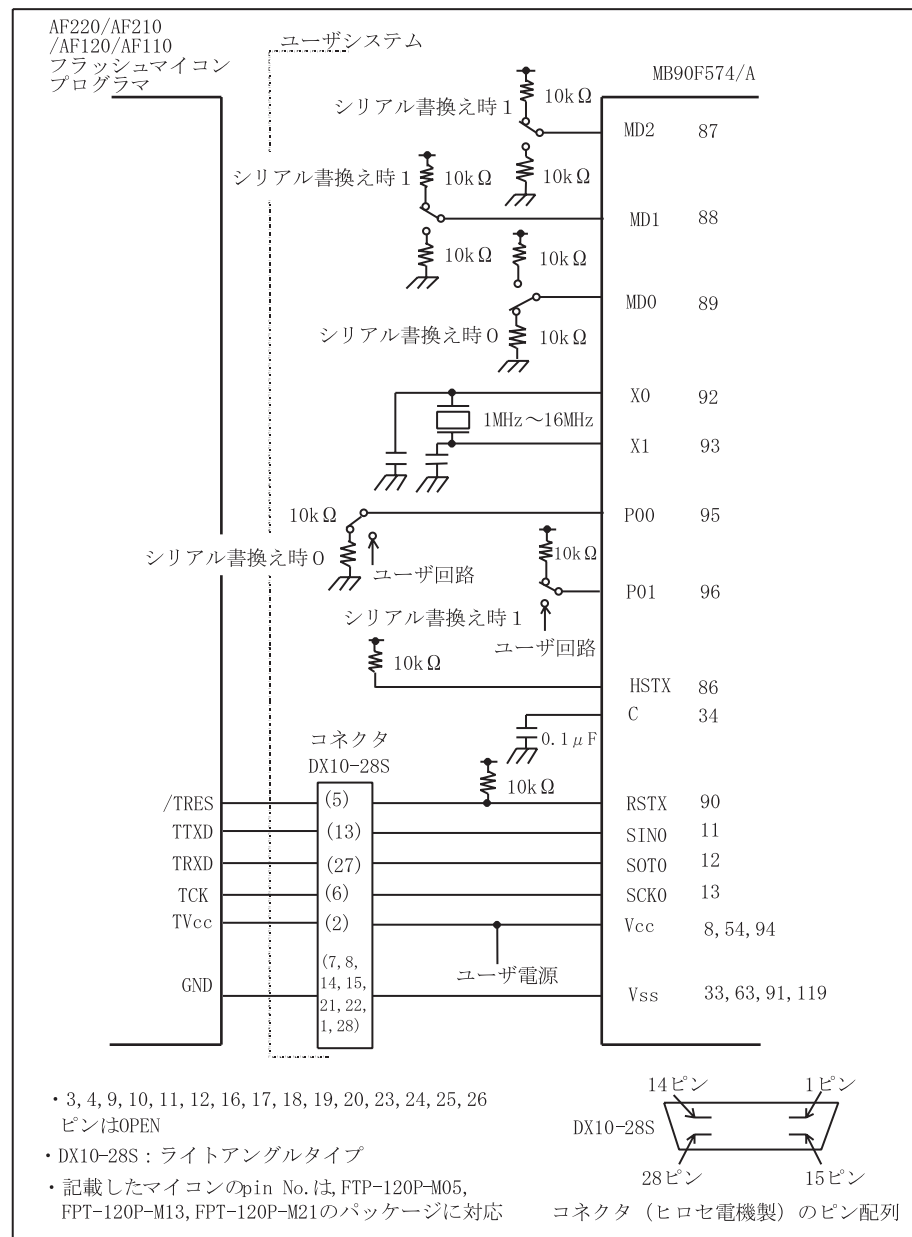
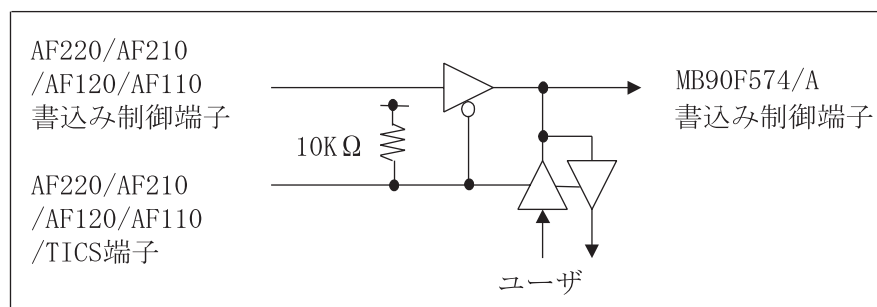


図 27.4-1 フラッシュマイコンプログラマとの最小限の接続例(ユーザ電源使用時)

- SINO, SOTO, SCK0端子をユーザシステムでも使用する場合には、下図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ムの /TICS信号により、シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。

27.5 フラッシュマイコンプログラマとの最小限の接続例（ライターから電源供給時）

図 27.5-1に、マイコンの電源電圧をライター電源より供給する場合の、フラッシュマイコンプログラマとの最小限の接続例を示します。

シリアル書き換えモード：MD2,MD1,MD0=110

フラッシュマイコンプログラマとの最小限の接続例（ライターから電源供給時）

フラッシュメモリ書込み時に各端子を図 27.5-1に示すように設定して頂ければ、MD2,MD1,MD0,P00とフラッシュマイコンプログラマとの接続は必要ありません。

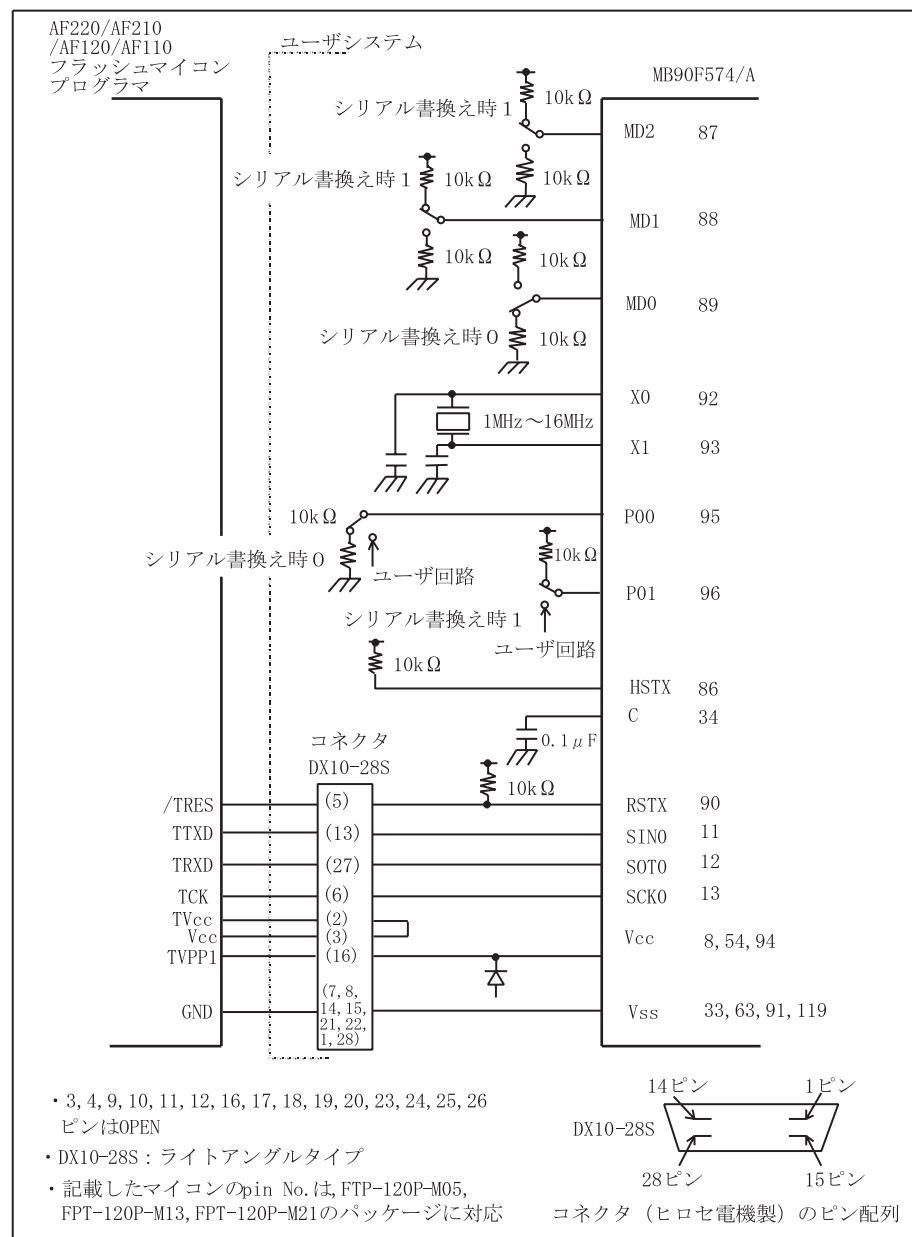
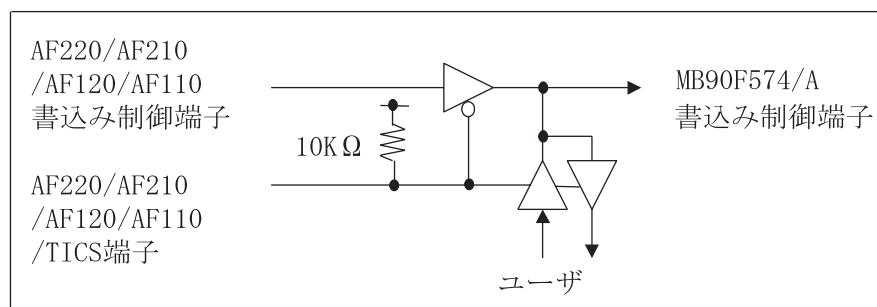


図 27.5-1 フラッシュマイコンプログラマとの最小限の接続例（ライターから電源供給時）

- SINO, SOTO, SCK0端子をユーザシステムでも使用する場合には, 下図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ムの /TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給する場合はユーザ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

MB90570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	アクセス	リソース	初期値
00 _H	ポート0データレジスタ	PDR0	R/W	ポート0	XXXXXXXX
01 _H	ポート1データレジスタ	PDR1	R/W	ポート1	XXXXXXXX
02 _H	ポート2データレジスタ	PDR2	R/W	ポート2	XXXXXXXX
03 _H	ポート3データレジスタ	PDR3	R/W	ポート3	XXXXXXXX
04 _H	ポート4データレジスタ	PDR4	R/W	ポート4	XXXXXXXX
05 _H	ポート5データレジスタ	PDR5	R/W	ポート5	XXXXXXXX
06 _H	ポート6データレジスタ	PDR6	R/W	ポート6	XXXXXXXX
07 _H	ポート7データレジスタ	PDR7	R/W	ポート7	XXXXXXXX
08 _H	ポート8データレジスタ	PDR8	R/W	ポート8	XXXXXXXX
09 _H	ポート9データレジスタ	PDR9	R/W	ポート9	XXXXXXXX
0A _H	ポートAデータレジスタ	PDRA	R/W	ポートA	XXXXXXXX
0B _H	ポートBデータレジスタ	PDRB	R/W	ポートB	XXXXXXXX
0C _H	ポートCデータレジスタ	PDRC	R/W	ポートC	XXXXXXXX
0D _H ~ 0F _H	使用禁止				
10 _H	ポート0方向レジスタ	DDR0	R/W	ポート0	00000000
11 _H	ポート1方向レジスタ	DDR1	R/W	ポート1	00000000
12 _H	ポート2方向レジスタ	DDR2	R/W	ポート2	00000000
13 _H	ポート3方向レジスタ	DDR3	R/W	ポート3	00000000
14 _H	ポート4方向レジスタ	DDR4	R/W	ポート4	00000000
15 _H	ポート5方向レジスタ	DDR5	R/W	ポート5	00000000
16 _H	ポート6方向レジスタ	DDR6	R/W	ポート6	00000000
17 _H	ポート7方向レジスタ	DDR7	R/W	ポート7	-----000
18 _H	ポート8方向レジスタ	DDR8	R/W	ポート8	00000000
19 _H	ポート9方向レジスタ	DDR9	R/W	ポート9	00000000
1A _H	ポートA方向レジスタ	DDRA	R/W	ポートA	--000000
1B _H	ポートB方向レジスタ	DDRB	R/W	ポートB	00000000
1C _H	ポートC方向レジスタ	DDRC	R/W	ポートC	----0000
1D _H	ポート4出力端子レジスタ	ODR4	R/W	ポート4	00000000
1E _H	アナログ入力許可レジスタ	ADER	R/W	ポート8, A/D	11111111
1F _H	使用禁止				
20 _H	シリアルモードレジスタ0	SMR0	R/W	UART0	00000000
21 _H	シリアルコントロールレジスタ0	SCR0	R/W		00000100
22 _H	シリアルインプット/シリアルアウトプットレジスタ0	SIDR0/ SODR0	R/W		XXXXXXXX
23 _H	シリアルステータスレジスタ0	SSR0	R/W		00001-00
24 _H	シリアルモードレジスタ1	SMR1	R/W	UART1	00000000
25 _H	シリアルコントロールレジスタ1	SCR1	R/W		00000100
26 _H	シリアルインプット/シリアルアウトプットレジスタ1	SIDR1/ SODR1	R/W		XXXXXXXX
27 _H	シリアルステータスレジスタ1	SSR1	R/W		00001-00
28 _H	通信プリスケラコントロールレジスタ0	CDCR0	R/W	UART0	0---1111
29 _H	使用禁止				

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ	略称	アクセス	リソース	初期値
2A _H	通信ブリスケラコントロールレジスタ1	CDCR1	R/W	UART1	0---1111
2B _H ~ 2F _H	使用禁止				
30 _H	割込み/DTP許可レジスタ	ENIR	R/W	DTP/外部割込み	00000000
31 _H	割込み/DTP要因レジスタ	EIRR	R/W		XXXXXXXX
32 _H	要求レベル設定レジスタ	ELVR	R/W		00000000
33 _H					00000000
34 _H ~ 35 _H	使用禁止				
36 _H	コントロールステータスレジスタ	ADCS1	R/W	A/Dコンバータ	00000000
37 _H		ADCS2			00000000
38 _H	データレジスタ	ADCR1	R		XXXXXXXX
39 _H		ADCR2			00001-XX
3A _H	D/A・コンバータデータレジスタ0	DADRO	R/W	D/Aコンバータ	XXXXXXXX
3B _H	D/A・コンバータデータレジスタ1	DADR1	R/W		XXXXXXXX
3C _H	D/A コントロールレジスタ0	DACR0	R/W		-----0
3D _H	D/A コントロールレジスタ1	DACR1	R/W		-----0
3E _H	クロック出力許可レジスタ	CLKR	R/W	クロックモニタ機能	----0000
3F _H	使用禁止				
40 _H	リロードレジスタL (ch.0)	PRLLO	R/W	8/16bit PPG0/1	XXXXXXXX
41 _H	リロードレジスタH (ch.0)	PRLHO	R/W		XXXXXXXX
42 _H	リロードレジスタL (ch.1)	PRLLO	R/W		XXXXXXXX
43 _H	リロードレジスタH (ch.1)	PRLHO	R/W		XXXXXXXX
44 _H	PPG0動作モード制御レジスタ	PPGCO	R/W		0X000XX1
45 _H	PPG1動作モード制御レジスタ	PPGCI	R/W		0X000001
46 _H	PPG0,1出力制御レジスタ	PPGOE	R/W		000000XX
47 _H	使用禁止				
48 _H	シリアルモードコントロールステータスレジスタ0	SMCS0	R/W	拡張シリアルI/O インタフェース0	----0000
49 _H					00000010
4A _H	シリアルデータレジスタ0	SDRO	R/W		XXXXXXXX
4B _H	使用禁止				
4C _H	シリアルモードコントロールステータスレジスタ1	SMCS1	R/W	拡張シリアルI/O インタフェース1	----0000
4D _H					00000010
4E _H	シリアルデータレジスタ1	SDR1	R/W		XXXXXXXX
4F _H	使用禁止				
50 _H	インプットキャプチャレジスタch.0	IPCP0	R	16ビット入出力 タイマ(インプットキャ プチャ部)	XXXXXXXX
51 _H					XXXXXXXX
52 _H	インプットキャプチャレジスタch.1	IPCP1	R		XXXXXXXX
53 _H					XXXXXXXX
54 _H	インプットキャプチャコントロール レジスタ	ICSO1	R/W		00000000
55 _H	使用禁止				
56 _H	タイマデータレジスタ	TCDT	R/W	16ビット入出力タイマ (フリーランタイマ部)	00000000
57 _H					00000000
58 _H	タイマコントロールステータスレジスタ	TCCS	R/W		00000000
59 _H	使用禁止				
5A _H	コンペアレジスタch.0	OCCP0	R/W	16ビット入出力タイマ (アウトプットコンペア 部)	XXXXXXXX
5B _H					XXXXXXXX
5C _H	コンペアレジスタch.1	OCCP1	R/W		XXXXXXXX
5D _H					XXXXXXXX
5E _H	コンペアレジスタch.2	OCCP2	R/W		XXXXXXXX
5F _H					XXXXXXXX

付表 A-1 I/Oマップ (続く)

アドレス	レジスタ	略 称	アクセス	リソース	初 期 値
60 _H	コンペアレジスタch.3	OCCP3	R/W	16ビット入出力タイマ (アウトプットコンペア部)	XXXXXXXX
61 _H					XXXXXXXX
62 _H	コンペアコントロールステータス レジスタch.0	OCS0	R/W		0000--00
63 _H	コンペアコントロールステータス レジスタch.1	OCS1	R/W		---00000
64 _H	コンペアコントロールステータス レジスタch.2	OCS2	R/W		0000--00
65 _H	コンペアコントロールステータス レジスタch.3	OCS3	R/W		---00000
66 _H ~ 67 _H	使用禁止				
68 _H	バスステータスレジスタ	IBSR	R/W	I ² Cインタフェース	00000000
69 _H	バスコントロールレジスタ	IBCR	R/W		00000000
6A _H	クロックコントロールレジスタ	ICCR	R/W		--0XXXXX
6B _H	アドレスレジスタ	IADR	R/W		-XXXXXXXX
6C _H	データレジスタ	IDAR	R/W		XXXXXXXX
6D _H ~ 6E _H	使用禁止				
6F _H	ROMミラー機能選択レジスタ	ROMM	W	ROMミラー機能	-----1
70 _H	アップダウンカウントレジスタ0	UDCR0	R	8/16ビットアップ ダウタイムカウンタ	00000000
71 _H	アップダウンカウントレジスタ1	UDCR1			00000000
72 _H	リロードコンペアレジスタ0	RCR0	W	8/16ビットアップ ダウタイムカウンタ	00000000
73 _H	リロードコンペアレジスタ1	RCR1			00000000
74 _H	カウンタステータスレジスタ0	CSR0	R/W		00000000
75 _H	予約領域	-	-		-----
76 _H	カウンタコントロールレジスタ0	CCRL0	R/W		-0000000
77 _H		CCRHO			00000000
78 _H	カウンタステータスレジスタ1	CSR1	R/W		00000000
79 _H	予約領域	-	-		-----
7A _H	カウンタコントロールレジスタ1	CCRL1	R/W		-0000000
7B _H		CCRH1			-0000000
7C _H	シリアルモードコントロールステータス レジスタ2	SMCS2	R/W	拡張シリアルI/O インタフェース2	----0000
7D _H		SMCS2	R/W		00000010
7E _H	シリアルデータレジスタ2	SDR2	R/W		XXXXXXXX
7F _H	使用禁止				
80 _H	チップセレクトコントロールレジスタ0	CSCR0	R/W	チップセレクト機能	----0000
81 _H	チップセレクトコントロールレジスタ1	CSCR1	R/W		----0000
82 _H	チップセレクトコントロールレジスタ2	CSCR2	R/W		----0000
83 _H	チップセレクトコントロールレジスタ3	CSCR3	R/W		----0000
84 _H	チップセレクトコントロールレジスタ4	CSCR4	R/W		----0000
85 _H	チップセレクトコントロールレジスタ5	CSCR5	R/W		----0000
86 _H	チップセレクトコントロールレジスタ6	CSCR6	R/W		----0000
87 _H ~ 8B _H	使用禁止				
8C _H	ポート0抵抗レジスタ	RDR0	R/W	ポート0	00000000
8D _H	ポート1抵抗レジスタ	RDR1	R/W	ポート1	00000000
8E _H	ポート6抵抗レジスタ	RDR6	R/W	ポート6	00000000
8F _H ~ 9D _H	使用禁止				
9E _H	プログラムアドレス検出コントロール ステータスレジスタ	PACSR	R/W	アドレス一致検出機能	00000000
9F _H	遅延割込み要因発生 / 解除レジスタ	DIRR	R/W	遅延割込み発生 モジュール	-----0

付表 A-1 I/Oマップ (続き)

アドレス	レジスタ	略称	アクセス	リソース	初期値
A0 _H	低消費電力モードレジスタ	LPMCR	R/W	低消費電力制御回路	00011000
A1 _H	クロック選択レジスタ	CKSCR	R/W		11111100
A2 _H ~ A4 _H	使用禁止				
A5 _H	自動レディ機能選択レジスタ	ARSR	W	外部バス端子制御回路	0011 - - 00
A6 _H	外部アドレス出力制御レジスタ	HACR	W		00000000
A7 _H	バス制御信号選択レジスタ	ECSR	W		00000000
A8 _H	ウォッチドッグタイマ制御レジスタ	WDTC	R/W	ウォッチドッグタイマ	XXXXXXXXXX
A9 _H	タイムベースタイマ制御レジスタ	TBTC	R/W	タイムベースタイマ	1 - - 00100
AA _H	時計タイマ制御レジスタ	WTC	R/W	時計タイマ	1X000000
AB _H ~ AD _H	使用禁止				
AE _H	フラッシュメモリコントロール ステータスレジスタ	FMCS	R/W	フラッシュメモリ	00010 - - 0
AF _H	使用禁止				
B0 _H	割込み制御レジスタ00	ICR00	R/W	割込みコントローラ	00000111
B1 _H	割込み制御レジスタ01	ICR01	R/W		00000111
B2 _H	割込み制御レジスタ02	ICR02	R/W		00000111
B3 _H	割込み制御レジスタ03	ICR03	R/W		00000111
B4 _H	割込み制御レジスタ04	ICR04	R/W		00000111
B5 _H	割込み制御レジスタ05	ICR05	R/W		00000111
B6 _H	割込み制御レジスタ06	ICR06	R/W		00000111
B7 _H	割込み制御レジスタ07	ICR07	R/W		00000111
B8 _H	割込み制御レジスタ08	ICR08	R/W		00000111
B9 _H	割込み制御レジスタ09	ICR09	R/W		00000111
BA _H	割込み制御レジスタ10	ICR10	R/W		00000111
BB _H	割込み制御レジスタ11	ICR11	R/W		00000111
BC _H	割込み制御レジスタ12	ICR12	R/W		00000111
BD _H	割込み制御レジスタ13	ICR13	R/W		00000111
BE _H	割込み制御レジスタ14	ICR14	R/W		00000111
BF _H	割込み制御レジスタ15	ICR15	R/W		00000111
CO _H ~ FF _H	外部領域				
100 _H ~ # _H	RAM領域				
# _H ~ 1FE _{FH}	予約領域				
1FF0 _H	プログラムアドレス検出レジスタ0	PADR0	R/W	アドレス一致検出機能	XXXXXXXXXX
1FF1 _H	プログラムアドレス検出レジスタ1		R/W		XXXXXXXXXX
1FF2 _H	プログラムアドレス検出レジスタ2		R/W		XXXXXXXXXX
1FF3 _H	プログラムアドレス検出レジスタ3	PADR1	R/W		XXXXXXXXXX
1FF4 _H	プログラムアドレス検出レジスタ4		R/W		XXXXXXXXXX
1FF5 _H	プログラムアドレス検出レジスタ5		R/W		XXXXXXXXXX
1FF6 _H ~ 1FFF _H	予約領域				

(注) 書込み可能なビットに関しては、リセットにより初期化される値が初期値として記述されています。読出し時の値ではありませんので注意してください。

また、LPMCR/CKSCR/WDTCでは、リセットの種類により、初期化される場合と、初期化されない場合がありますが、初期化される場合の初期値が記述されています。

- ・ 00FF_H以下のアドレスは予約領域です。外バスアクセス信号は出ません。
- ・ RAM領域と予約領域の境界"#_H"は、MB90573の場合は1900_H、MB90574、MB90574Cの場合は1FE_{FH}(上記の予約領域はなし)になります。
- ・ 表中の項目「アクセス」の表記説明
R/W: リードライト可 R: リードオンリー W: ライトオンリー
- ・ 表中の項目「初期値」の表記説明
0: このビットの初期値は"0"。 1: このビットの初期値は"1"。
X: このビットの初期値は不定。 -: このビットは未使用。初期値は不定。

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。

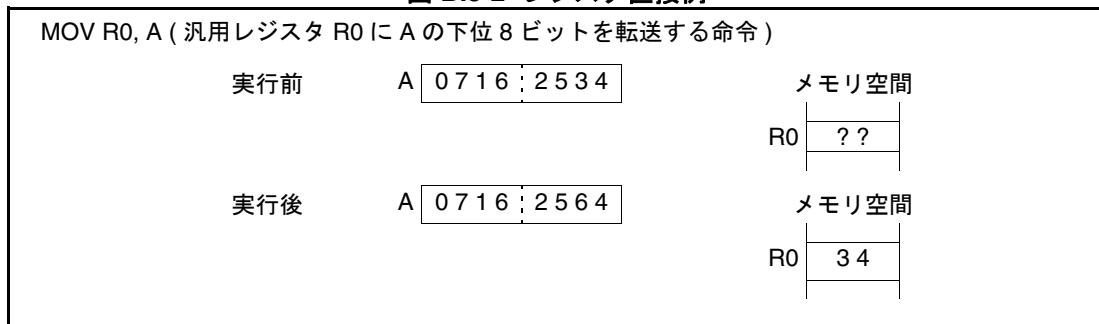
表 B.3-1 レジスタ直接

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。

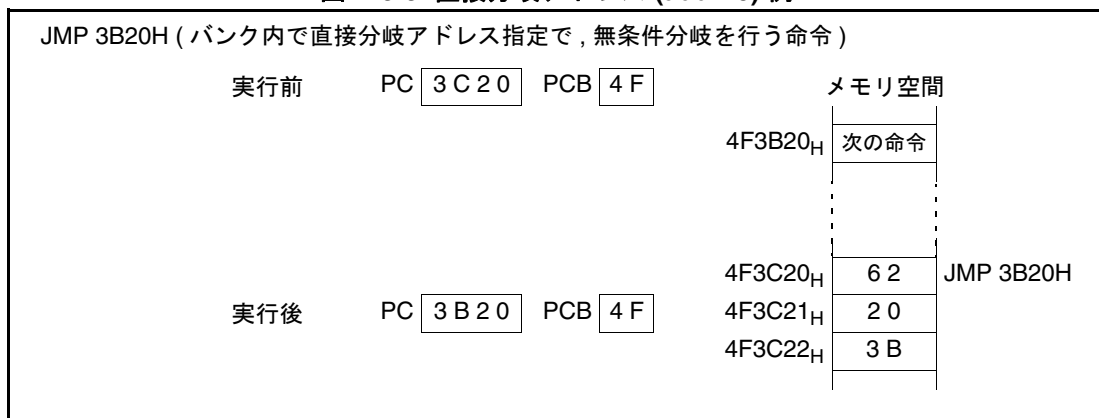
図 B.3-2 レジスタ直接例



● 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。

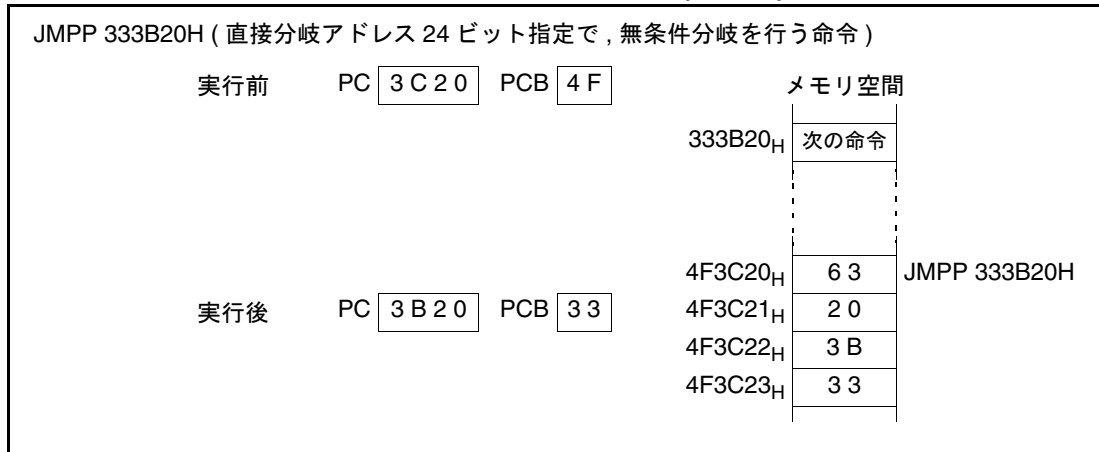
図 B.3-3 直接分岐アドレス (addr16) 例



● 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イースメントで直接指定します。ディスプレイースメントのデータ長は 24 ビットです。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。図 B.3-4 に例を示します。

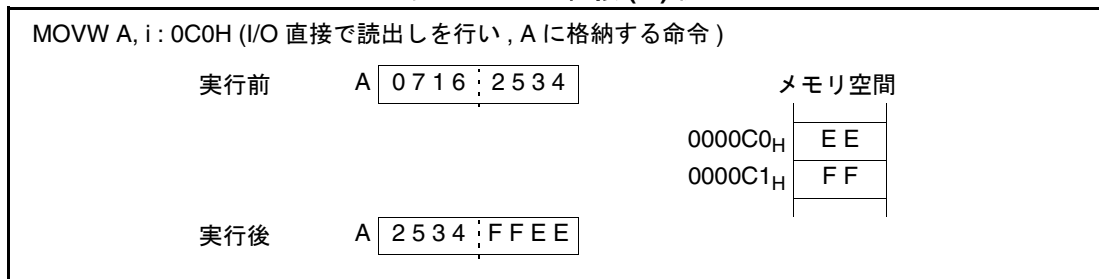
図 B.3-4 物理直接分岐アドレス (addr24) 例



● I/O 直接 (io)

オペランドのメモリアドレスを 8 ビットのディスプレイースメントで直接指定します。データバンクレジスタ (DTB)、ダイレクトページレジスタ (DPR) の値にかかわらず、物理アドレス "000000_H" ~ "0000FF_H" の空間の I/O 空間がアクセスされます。I/O 直接アドレス指定を使用した命令の前にバンク指定用のバンクセレクトプリフィックスを記述しても無効です。図 B.3-5 に例を示します。

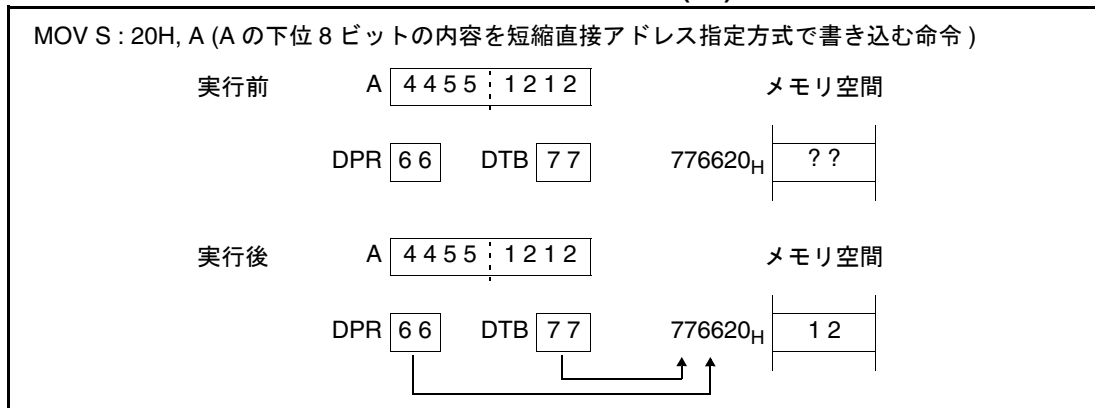
図 B.3-5 I/O 直接 (io) 例



● 短縮直接アドレス (dir)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit15 ～ bit8 はダイレクトページレジスタ (DPR) により指定されます。アドレスの bit23 ～ bit16 はデータバンクレジスタ (DTB) により指定されます。図 B.3-6 に例を示します。

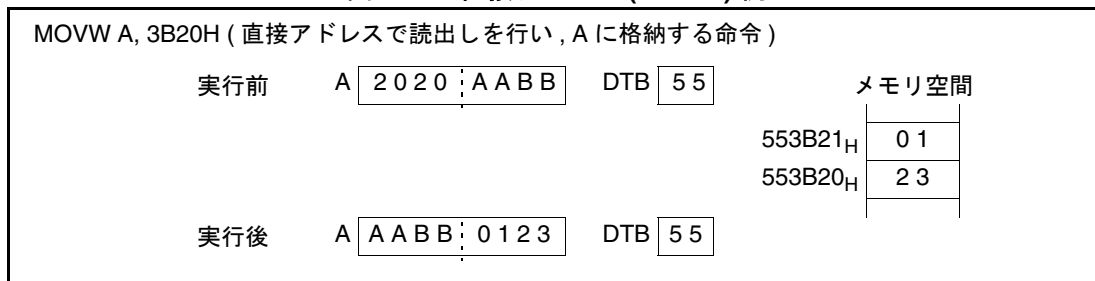
図 B.3-6 短縮直接アドレス (dir) 例



● 直接アドレス (addr16)

オペランドで、メモリアドレス下位 16 ビットを直接指定します。アドレスの bit23 ～ bit16 はデータバンクレジスタ (DTB) で指定されます。直接アドレスのアドレッシングに対しては、アクセス空間指定用のプリフィックス命令は無効です。図 B.3-7 に例を示します。

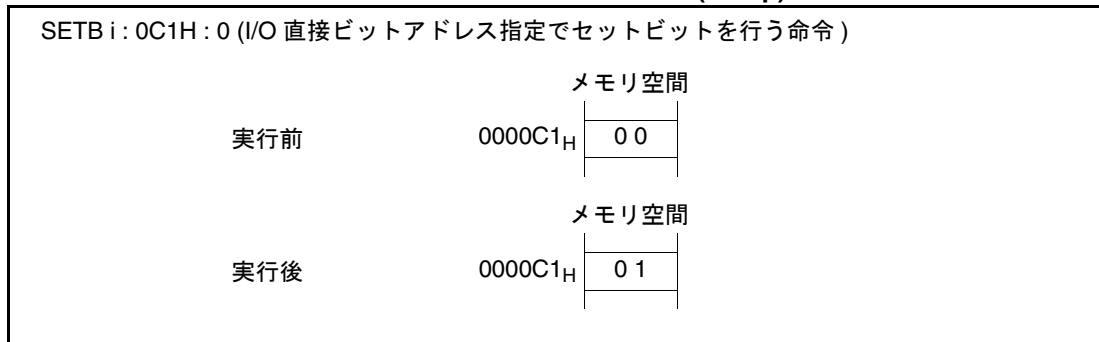
図 B.3-7 直接アドレス (addr16) 例



● I/O 直接ビットアドレス (io: bp)

物理アドレス "000000_H" ~ "0000FF_H" 内のビットを直接指定します。ビットの位置は ": bp" で表され, 数字の大きいほうが最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。図 B.3-8 に例を示します。

図 B.3-8 I/O 直接ビットアドレス (io: bp) 例



● 短縮直接ビットアドレス (dir: bp)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit15 ~ bit8 はダイレクトページレジスタ (DPR) で指定されます。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され, 数字の大きい方が最上位ビット, 小さい方が最下位ビットとなります。図 B.3-9 に例を示します。

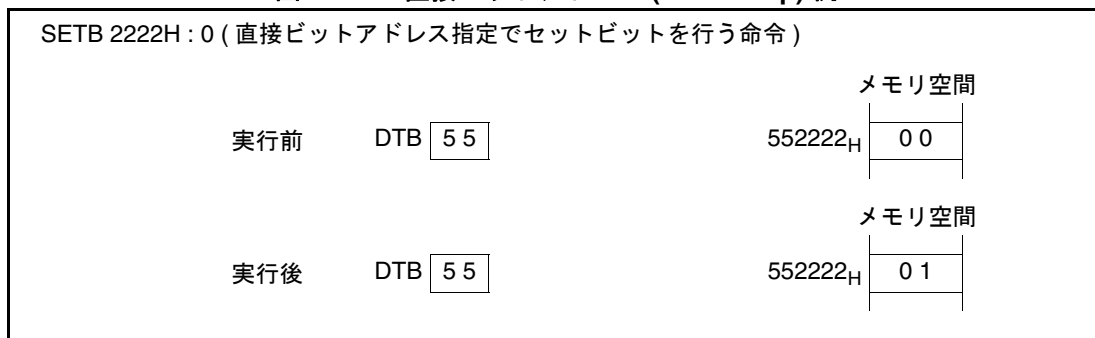
図 B.3-9 短縮直接ビットアドレス (dir: bp) 例



● 直接ビットアドレス (addr16: bp)

64K バイト内の任意のビットに対し直接指定します。アドレスの bit23 ～ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され、数字の大きい方が最上位ビット、小さい方が最下位ビットとなります。図 B.3-10 に例を示します。

図 B.3-10 直接ビットアドレス (addr16: bp) 例



● ベクタアドレス (#vct)

分岐先のアドレスは指定されたベクタの内容となります。ベクタ番号のデータ長には 4 ビットと 8 ビットの 2 種類があります。サブルーチンコール命令、ソフトウェア割り込み命令に使用します。図 B.3-11 に例を示します。

図 B.3-11 ベクタアドレス (#vct) 例

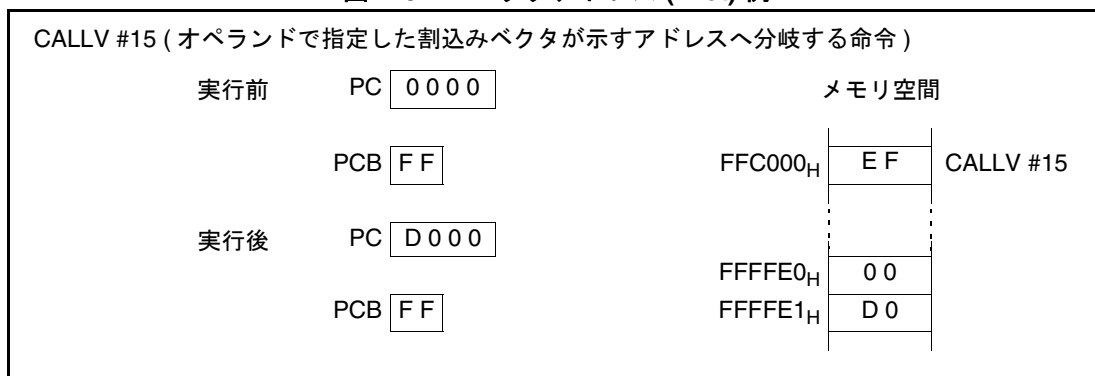


表 B.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE _H	XXFFFF _H
CALLV #1	XXFFFC _H	XXFFFD _H
CALLV #2	XXFFFA _H	XXFFFB _H
CALLV #3	XXFFF8 _H	XXFFF9 _H
CALLV #4	XXFFF6 _H	XXFFF7 _H
CALLV #5	XXFFF4 _H	XXFFF5 _H
CALLV #6	XXFFF2 _H	XXFFF3 _H
CALLV #7	XXFFF0 _H	XXFFF1 _H
CALLV #8	XXFFEE _H	XXFFEF _H
CALLV #9	XXFFEC _H	XXFFED _H
CALLV #10	XXFFEA _H	XXFFEB _H
CALLV #11	XXFFE8 _H	XXFFE9 _H
CALLV #12	XXFFE6 _H	XXFFE7 _H
CALLV #13	XXFFE4 _H	XXFFE5 _H
CALLV #14	XXFFE2 _H	XXFFE3 _H
CALLV #15	XXFFE0 _H	XXFFE1 _H

(注意事項) XX には PCB レジスタの値が入ります。

<注意事項>

プログラムカウンタバンクレジスタ (PCB) が "FF_H" の場合ベクタ領域は INT #vct8 (#0 ~ #7) のベクタ領域と共有しているため、使用する場合は注意が必要です。(表 B.3-2 を参照してください)

B.4 間接アドレッシング

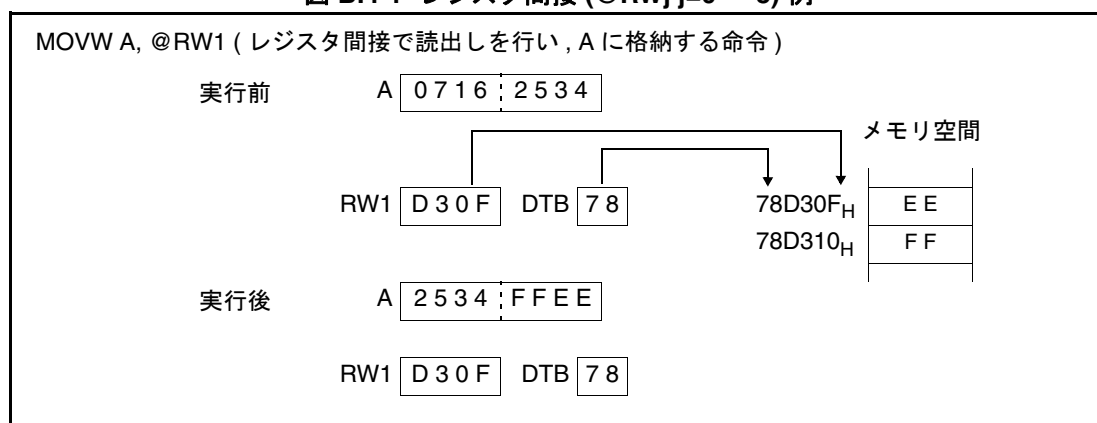
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

■ 間接アドレッシング

● レジスタ間接 (@RWj j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-1 に例を示します。

図 B.4-1 レジスタ間接 (@RWj j=0 ~ 3) 例



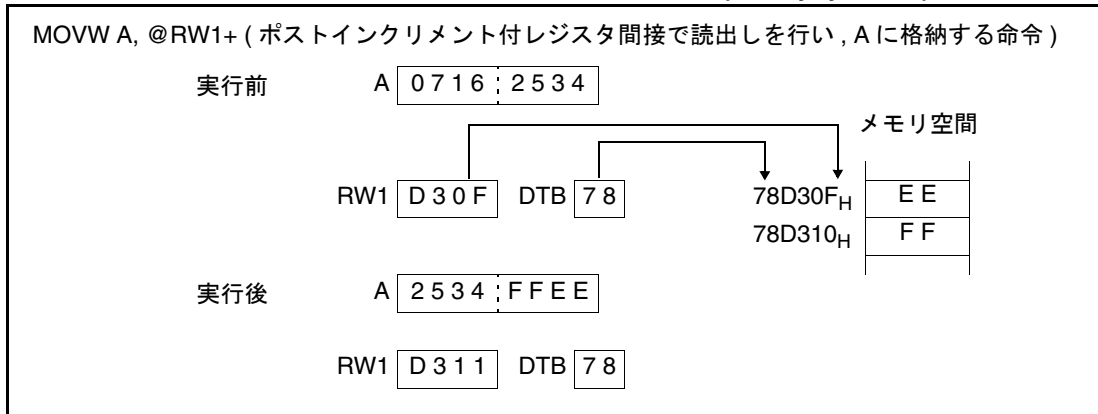
● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。オペランド操作後RWjはオペランドデータ長 (バイトの場合は1, ワードの場合は2, ロングワードの場合は4) 分だけ加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。

ポストインクリメントした結果がインクリメント指定したレジスタ自身のアドレスだった場合は、参照される値はインクリメントした値となり、命令が書込みだった場合は命令による書込みが優先されるので、インクリメントするはずだったレジスタは書込みデータとなります。

図 B.4-2 に例を示します。

図 B.4-2 ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3) 例

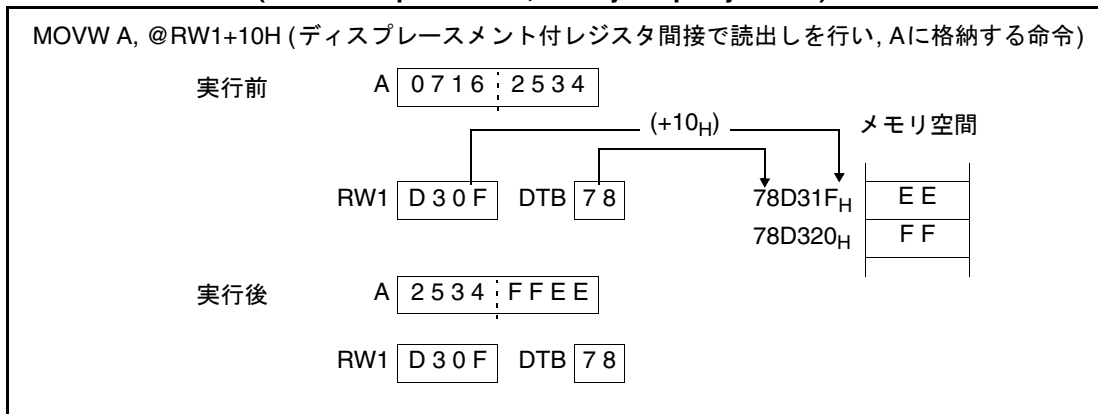


● ディスプレースメント付レジスタ間接

(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)

汎用レジスタ RWj の内容にディスプレースメントを加算したものをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは、バイトとワードの2種類があり、符号付数値として加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1, RW4, RW5 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3, RW7 を用いた場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2, RW6 を用いた場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-3 に例を示します。

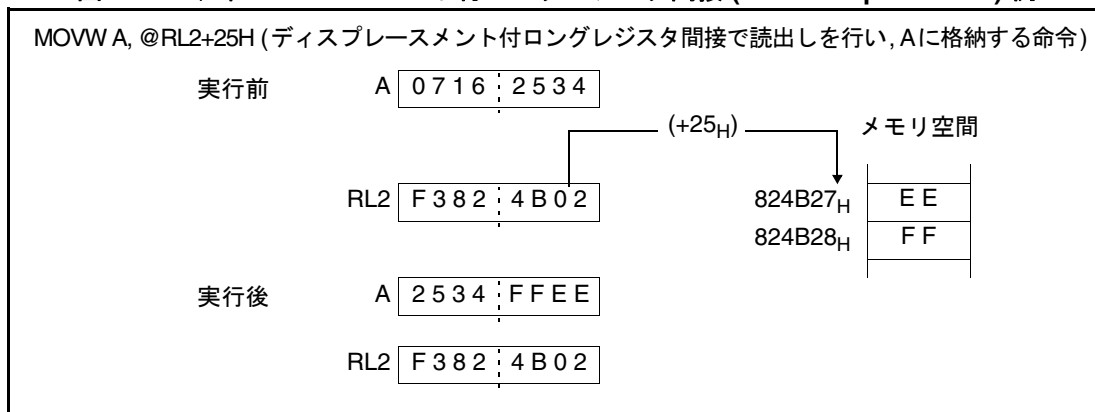
図 B.4-3 ディスプレースメント付レジスタ間接
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● ディスプレースメント付ロングレジスタ間接 (RLi+disp8 i=0 ~ 3)

汎用レジスタ RLi の内容にディスプレースメントを加算した結果の下位 24 ビットをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは 8 ビットで符号付数値として加算されます。図 B.4-4 に例を示します。

図 B.4-4 ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3) 例



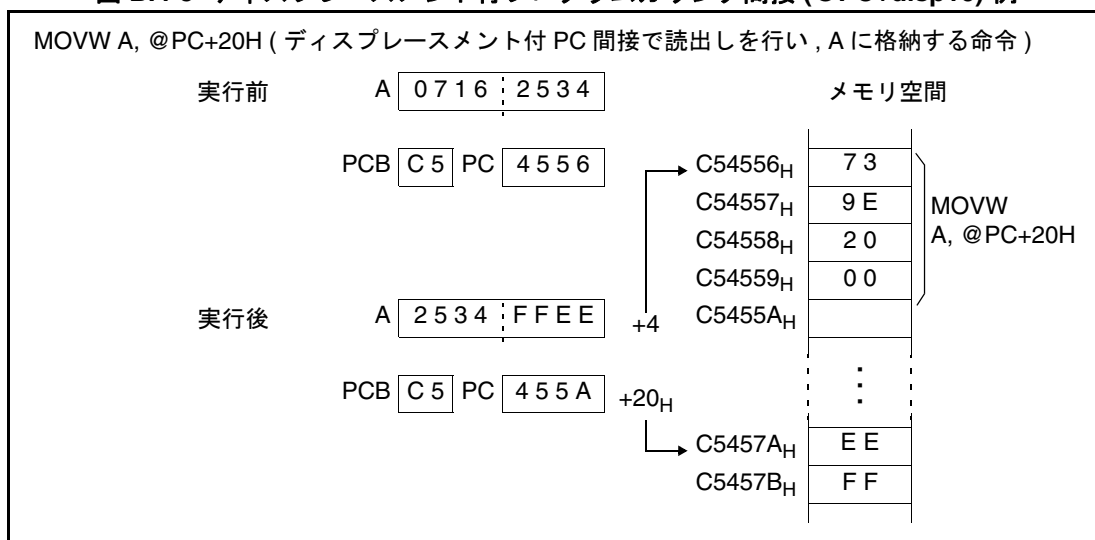
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(命令のアドレス+4+disp16) で示されるアドレスのメモリをアクセスするアドレッシングです。ディスプレースメントはワード長です。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) により指定されます。次に示す命令のオペランドアドレスは, (次の命令のアドレス +disp16) とは, みなしませんでしたので注意してください。

- DBNZ eam, rel
- CBNE eam, #imm8, rel
- MOV eam, #imm8
- DWBNZ eam, rel
- CWBNE eam, #imm16, rel
- MOVW eam, #imm16

図 B.4-5 に例を示します。

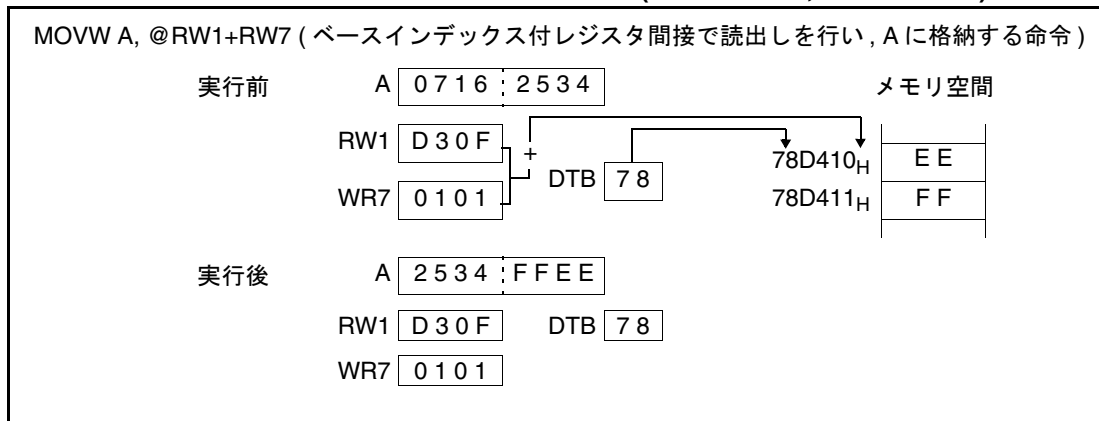
図 B.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例



● ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7)

汎用レジスタ RW7 の内容に, RW0 あるいは RW1 を加算したものをアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で示されます。図 B.4-6 に例を示します。

図 B.4-6 ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7) 例

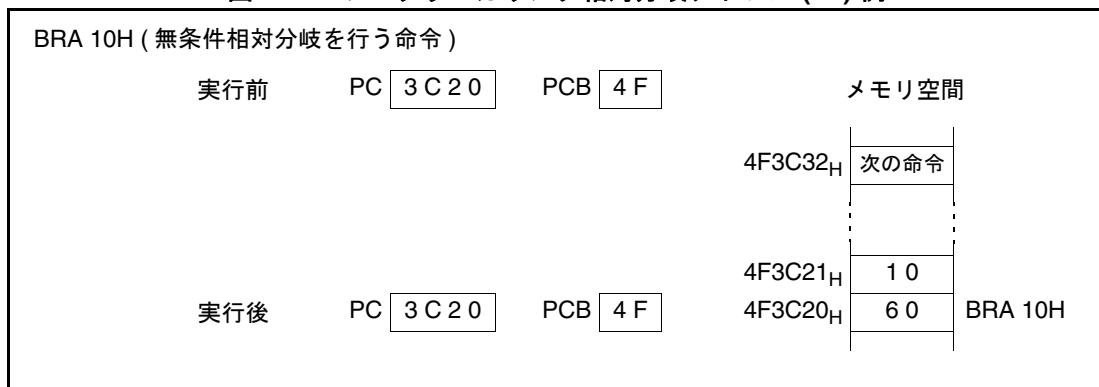


● プログラムカウンタ相対分岐アドレス (rel)

分岐先のアドレスはプログラムカウンタ (PC) の値と 8 ビットのディスプレースメントを加算した値となります。加算の結果が 16 ビットを超えた場合は, バンクレジスタのインクリメントまたはデクレメントをせずに, 超えた分は無視されますので, 64K バイトのバンク内で閉じたアドレスとなります。プログラムカウンタ相対分岐アドレスによるアドレッシングは, 無条件 / 条件分岐命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で示されます。

図 B.4-7 に例を示します。

図 B.4-7 プログラムカウンタ相対分岐アドレス (rel) 例



● レジスタリスト (rlst)

スタックに対するプッシュ/ポップの対象となるレジスタを指定します。図 B.4-8 にレジスタリストの構成, 図 B.4-9 に例を示します。

図 B.4-8 レジスタリストの構成

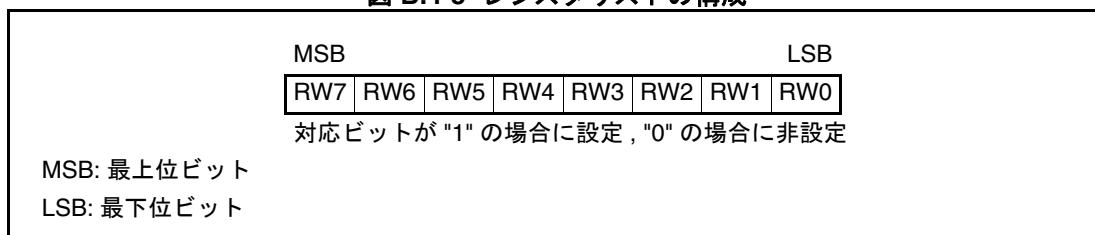
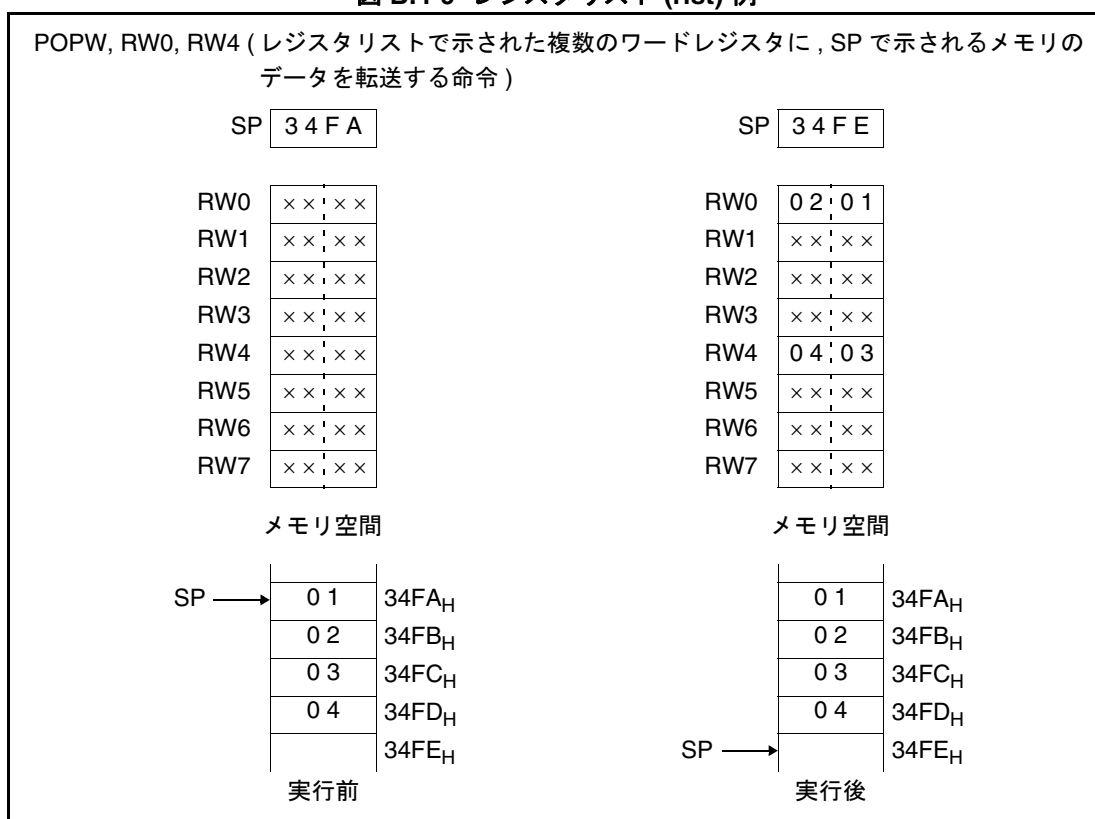


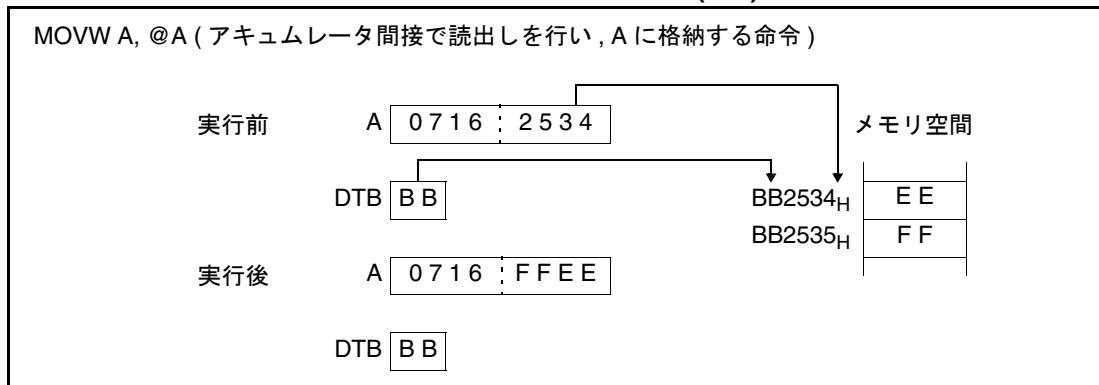
図 B.4-9 レジスタリスト (rlst) 例



● アキュムレータ間接 (@A)

アキュムレータの下位バイト (AL) の内容 (16 ビット) で示されるアドレスのメモリをアクセスするアドレス方式です。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) によりニーモニックで指定されます。図 B.4-10 に例を示します。

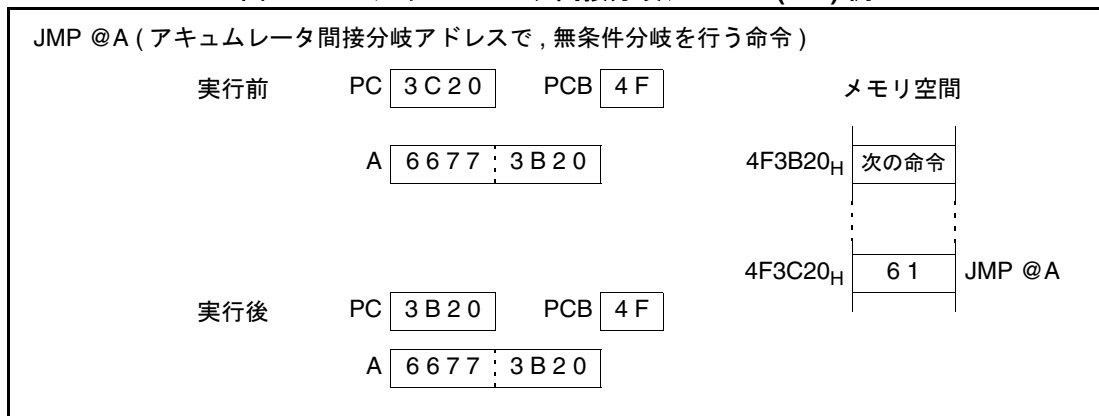
図 B.4-10 アキュムレータ間接 (@A) 例



● アキュムレータ間接分岐アドレス (@A)

分岐先のアドレスは, アキュムレータの下位バイト (AL) の内容 (16 ビット) となります。バンク空間内での分岐先を示し, アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されますが, JCTX (Jump Context) 命令の場合は, アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。アキュムレータ間接分岐アドレスによるアドレッシングは, 無条件分岐命令に使用します。図 B.4-11 に例を示します。

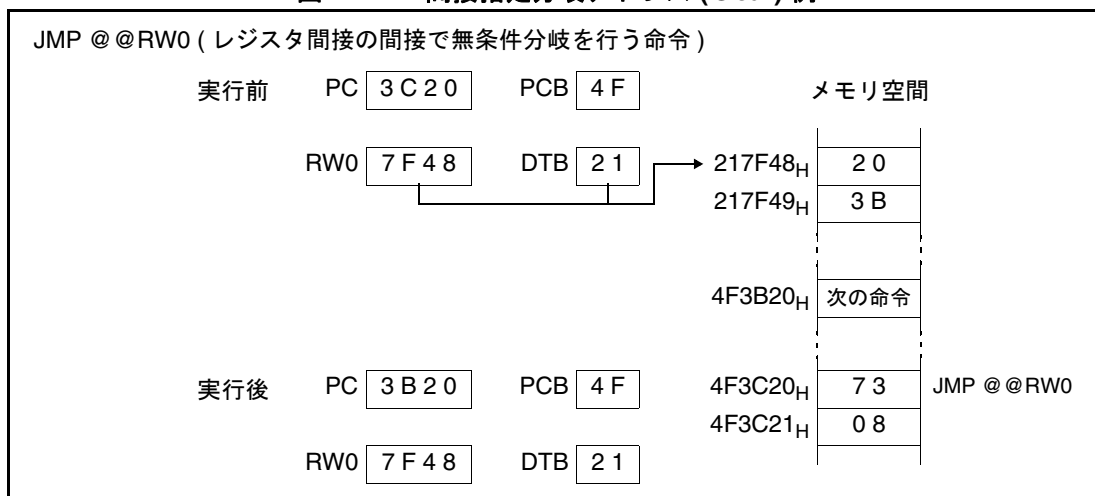
図 B.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 間接指定分岐アドレス (@ear)

ear で示される番地のワードデータが分岐先アドレスとなります。図 B.4-12 に例を示します。

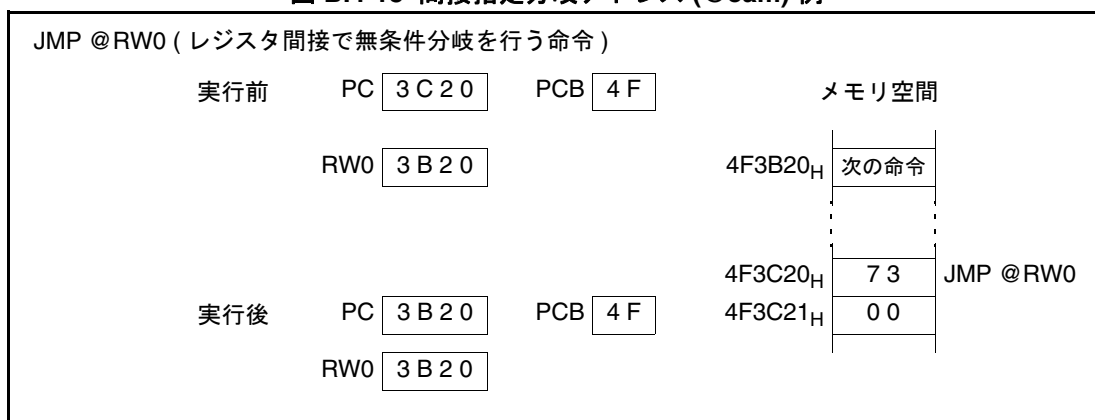
図 B.4-12 間接指定分岐アドレス (@ear) 例



● 間接指定分岐アドレス (@eam)

eam で示される番地のワードデータが分岐先アドレスとなります。図 B.4-13 に例を示します。

図 B.4-13 間接指定分岐アドレス (@eam) 例



B.5 実行サイクル数

命令の実行に要するサイクル数 (実行サイクル数) は、各命令の「サイクル数」の値と、条件で決まる「補正值」の値およびプログラムフェッチの「サイクル数」の値を加算することで得られます。

■ 実行サイクル数

内蔵 ROM などの 16 ビットバスに接続されたメモリ上のプログラムをフェッチする場合には、実行中の命令がワード境界を越えるごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

外部データバスの 8 ビットバスに接続されたメモリ上のプログラムをフェッチする場合は、実行中の命令の 1 バイトごとにプログラムフェッチをするため、データのアクセスなどに干渉すると実行サイクル数が増大します。

CPU 間欠動作時は、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスのアクセスをすると、低消費電力モード制御レジスタの CG0, CG1 ビットで指定されるサイクル数分 CPU へ供給されるクロックが一時停止しますので、CPU 間欠動作の命令の実行に要するサイクル数は、通常の実行サイクル数に、汎用レジスタ、内蔵 ROM、内蔵 RAM、内蔵 I/O、外部データバスの「アクセス回数」×一時停止の「サイクル数」の値を「補正值」として加算してください。

■ 実行サイクル数計算方法

表 B.5-1、表 B.5-2、表 B.5-3 に命令実行サイクル数および補正值のデータを示します。

表 B.5-1 各種アドレッシングに対する実行サイクル数

コード	オペランド	(a) *	各種アドレッシングに対するレジスタアクセス回数
		各種アドレッシングに対する実行サイクル数	
00 ~ 07	Ri RWi RLi	命令一覧表に記載	命令一覧表に記載
08 ~ 0B	@RWj	2	1
0C ~ 0F	@RWj+	4	2
10 ~ 17	@RWi+disp8	2	1
18 ~ 1B	@RWi+disp16	2	1
1C	@RW0+RW7	4	2
1D	@RW1+RW7	4	2
1E	@PC+disp16	2	0
1F	addr16	1	0

*: (a) は「B.8 F2MC-16LX 命令一覧表」の～ (サイクル数)、B (補正值) で使用されています。"～" と "B" の意味については「B.7 命令一覧表の読み方」を参照してください。

表 B.5-2 実サイクル数算出用サイクル数の補正值

オペランド	(b) バイト * ¹		(c) ワード * ¹		(d) ロング * ¹	
	サイクル数	アクセス回数	サイクル数	アクセス回数	サイクル数	アクセス回数
内部レジスタ	+0	1	+0	1	+0	2
内部メモリ 偶数アドレス	+0	1	+0	1	+0	2
内部メモリ 奇数アドレス	+0	1	+2	2	+4	4
外部データバス * ² 16 ビット偶数アドレス	+1	1	+1	1	+2	2
外部データバス * ² 16 ビット奇数アドレス	+1	1	+4	2	+8	4
外部データバス * ² 8 ビット	+1	1	+4	2	+8	4

*1: (b), (c), (d) は「B.8 F2MC-16LX 命令一覧表」の～ (サイクル数), B (補正值) で使用されています。

*2: 外部データバスを使用した場合は, レディ入力および自動レディによりウェイトしたサイクル数も加算する必要があります。

表 B.5-3 プログラムフェッチサイクル数算出用サイクル数の補正值

命令	バイト境界	ワード境界
内部メモリ	—	+2
外部データバス 16 ビット	—	+3
外部データバス 8 ビット	+3	—

(注意事項) ・ 外部データバスを使用した場合は, レディ入力および自動レディでウェイトしたサイクル数も加算する必要があります。
 ・ 実際にはすべてのプログラムフェッチで, 命令実行が遅くなるわけではないので, この補正值は最悪ケースを算出する場合に使用してください。

B.6 実効アドレスフィールド

表 B.6-1 に実効アドレスフィールドを示します。

■ 実効アドレスフィールド

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記			アドレス形式	アドレス拡張部の バイト数*
00	R0	RW0	RL0	レジスタ直接 ea は左から順に バイト ワード ロングワード の型に対応する。	—
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0+			ポストインクリメント付 レジスタ間接	0
0D	@RW1+				
0E	@RW2+				
0F	@RW3+				
10	@RW0+disp8			8 ビットディスプレースメント付 レジスタ間接	1
11	@RW1+disp8				
12	@RW2+disp8				
13	@RW3+disp8				
14	@RW4+disp8				
15	@RW5+disp8				
16	@RW6+disp8				
17	@RW7+disp8				

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記	アドレス形式	アドレス拡張部の バイト数*
18	@RW0+disp16	16 ビットディスプレースメント付 レジスタ間接	2
19	@RW1+disp16		
1A	@RW2+disp16		
1B	@RW3+disp16		
1C	@RW0+RW7	インデックス付レジスタ間接	0
1D	@RW1+RW7	インデックス付レジスタ間接	0
1E	@PC+disp16	16 ビットディスプレースメント付 PC 間接	2
1F	addr16	直接アドレス	2

*: アドレス拡張部のバイト数は、「B.8 F2MC-16LX 命令一覧表」の#(バイト数)の "+" に当てはまります。
 "#" の意味については「B.7 命令一覧表の読み方」を参照してください。

B.7 命令一覧表の読み方

「B.8 F2MC-16LX 命令一覧表」で使用している項目の説明を表 B.7-1 に、記号の説明を表 B.7-2 に示します。

■ 命令の表示記号の説明

表 B.7-1 命令一覧表の項目の説明

項目	説明
ニーモニック	英大文字、記号：アセンブラ上もそのまま表記します。 英小文字：アセンブラ上では、書き替えて記述します。 英小文字の後の数：命令中のビット幅を示します。
#	バイト数を示します。
～	サイクル数を示します。
RG	命令実行時のレジスタアクセス回数を示します。 CPU 間欠動作時の補正値を算出するのに使用します。
B	命令実行時の実サイクル数の算出用補正値を示します。 命令実行時の実サイクルは～欄の数値を加算したものとなります。
オペレーション	命令の動作を示します。
LH	アキュムレータの bit15 ～ bit8 に対する特殊動作を示します。 Z: 0 を転送する。 X: 符号を拡張して転送する。 －: 転送しない。
AH	アキュムレータの上位 16 ビットに対する特殊動作を示します。 *: AL から AH へ転送する。 －: 転送しない。 Z: AH へ 00 _H を転送する。 X: AL の符号拡張で AH へ 00 _H または FF _H を転送する。
I	I (割込み許可), S (スタック), T (ステッキビット), N (ネガティブ), Z (ゼロ), V (オーバフロー), C (キャリー) の各フラグの状態を示します。 *: 命令の実行で変化する。 －: 変化しない。 S: 命令の実行でセットされる。 R: 命令の実行でリセットされる。
S	
T	
N	
Z	
V	
C	
RMW	
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) であるかどうかを示します。 *: リードモディファイライト命令である。 －: リードモディファイライト命令ではない。 (注意事項) 読み書きで意味の異なるアドレスには使用できません。

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
A	32 ビットアキュムレータ 命令により, 使用されるビット長が変わります。 バイト: AL の下位 8 ビット ワード: AL の 16 ビット ロング: AL: AH の 32 ビット
AH	A の上位 16 ビット
AL	A の下位 16 ビット
SP	スタックポインタ (USP or SSP)
PC	プログラムカウンタ
PCB	プログラムカウンタバンクレジスタ
DTB	データバンクレジスタ
ADB	アディショナルデータバンクレジスタ
SSB	システムスタックバンクレジスタ
USB	ユーザスタックバンクレジスタ
SPB	カレントスタックバンクレジスタ (SSB or USB)
DPR	ダイレクトページレジスタ
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir	短縮直接アドレス指定
addr16	直接アドレス指定
addr24	物理直接アドレス指定
ad24 0 ~ 15	addr24 の bit0 ~ bit15
ad24 16 ~ 23	addr24 の bit16 ~ bit23
io	I/O 領域 (000000 _H ~ 0000FF _H)
#imm4	4 ビット即値データ
#imm8	8 ビット即値データ
#imm16	16 ビット即値データ
#imm32	32 ビット即値データ
ext (imm8)	8 ビット即値データを符号拡張した 16 ビットデータ

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
disp8	8 ビットディスプレースメント
disp16	16 ビットディスプレースメント
bp	ビットオフセット値
vct4	ベクタ番号 (0 ～ 15)
vct8	ベクタ番号 (0 ～ 255)
() b	ビットアドレス
rel	PC 相対分岐指定
ear	実効アドレス指定 (コード 00 ～ 07)
eam	実効アドレス指定 (コード 08 ～ 1F)
rlst	レジスタ並び

B.8 F²MC-16LX 命令一覧表

F²MC-16LX で使用している命令の一覧を示します。

■ F²MC-16LX 命令一覧表

表 B.8-1 転送系命令 (バイト) 41 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A,dir	2	3	0	(b)	byte (A) ← (dir)	Z	*	-	-	-	*	*	-	-	-
MOV A,addr16	3	4	0	(b)	byte (A) ← (addr16)	Z	*	-	-	-	*	*	-	-	-
MOV A,Ri	1	2	1	0	byte (A) ← (Ri)	Z	*	-	-	-	*	*	-	-	-
MOV A,ear	2	2	1	0	byte (A) ← (ear)	Z	*	-	-	-	*	*	-	-	-
MOV A,eam	2+	3 + (a)	0	(b)	byte (A) ← (eam)	Z	*	-	-	-	*	*	-	-	-
MOV A,io	2	3	0	(b)	byte (A) ← (io)	Z	*	-	-	-	*	*	-	-	-
MOV A,#imm8	2	2	0	0	byte (A) ← imm8	Z	*	-	-	-	*	*	-	-	-
MOV A,@A	2	3	0	(b)	byte (A) ← ((A))	Z	-	-	-	-	*	*	-	-	-
MOV A,@RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	Z	*	-	-	-	*	*	-	-	-
MOVN A,#imm4	1	1	0	0	byte (A) ← imm4	Z	*	-	-	-	R	*	-	-	-
MOVX A,dir	2	3	0	(b)	byte (A) ← (dir)	X	*	-	-	-	*	*	-	-	-
MOVX A,addr16	3	4	0	(b)	byte (A) ← (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX A,Ri	2	2	1	0	byte (A) ← (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX A,ear	2	2	1	0	byte (A) ← (ear)	X	*	-	-	-	*	*	-	-	-
MOVX A,eam	2+	3 + (a)	0	(b)	byte (A) ← (eam)	X	*	-	-	-	*	*	-	-	-
MOVX A,io	2	3	0	(b)	byte (A) ← (io)	X	*	-	-	-	*	*	-	-	-
MOVX A,#imm8	2	2	0	0	byte (A) ← imm8	X	*	-	-	-	*	*	-	-	-
MOVX A,@A	2	3	0	(b)	byte (A) ← ((A))	X	-	-	-	-	*	*	-	-	-
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) ← ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX A,@RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOV dir,A	2	3	0	(b)	byte (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV addr16,A	3	4	0	(b)	byte (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,A	1	2	1	0	byte (Ri) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV ear,A	2	2	1	0	byte (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV eam,A	2+	3 + (a)	0	(b)	byte (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV io,A	2	3	0	(b)	byte (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV @RLi+disp8,A	3	10	2	(b)	byte ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,ear	2	3	2	0	byte (Ri) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOV Ri,eam	2+	4 + (a)	1	(b)	byte (Ri) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOV ear,Ri	2	4	2	0	byte (ear) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV eam,Ri	2+	5 + (a)	1	(b)	byte (eam) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV Ri,#imm8	2	2	1	0	byte (Ri) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV io,#imm8	3	5	0	(b)	byte (io) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV dir,#imm8	3	5	0	(b)	byte (dir) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV ear,#imm8	3	2	1	0	byte (ear) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV eam,#imm8	3+	4 + (a)	0	(b)	byte (eam) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV @AL,AH	2	3	0	(b)	byte ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCH A,ear	2	4	2	0	byte (A) ↔ (ear)	Z	-	-	-	-	-	-	-	-	-
XCH A,eam	2+	5 + (a)	0	2 × (b)	byte (A) ↔ (eam)	Z	-	-	-	-	-	-	-	-	-
XCH Ri,ear	2	7	4	0	byte (Ri) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCH Ri,eam	2+	9 + (a)	2	2 × (b)	byte (Ri) ↔ (eam)	-	-	-	-	-	-	-	-	-	-

(注意事項) 表中の (a), (b) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-2 転送系命令 (ワード・ロングワード) 38 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A,dir	2	3	0	(c)	word (A) ← (dir)	-	*	-	-	-	*	*	-	-	-
MOVW A,addr16	3	4	0	(c)	word (A) ← (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW A,SP	1	1	0	0	word (A) ← (SP)	-	*	-	-	-	*	*	-	-	-
MOVW A,RWi	1	2	1	0	word (A) ← (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW A,ear	2	2	1	0	word (A) ← (ear)	-	*	-	-	-	*	*	-	-	-
MOVW A,eam	2+	3 + (a)	0	(c)	word (A) ← (eam)	-	*	-	-	-	*	*	-	-	-
MOVW A,io	2	3	0	(c)	word (A) ← (io)	-	*	-	-	-	*	*	-	-	-
MOVW A,@A	2	3	0	(c)	word (A) ← ((A))	-	-	-	-	-	*	*	-	-	-
MOVW A,#imm16	3	2	0	0	word (A) ← imm16	-	*	-	-	-	*	*	-	-	-
MOVW A,@RWi+disp8	2	5	1	(c)	word (A) ← ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW A,@RLi+disp8	3	10	2	(c)	word (A) ← ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW dir,A	2	3	0	(c)	word (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW addr16,A	3	4	0	(c)	word (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW SP,A	1	1	0	0	word (SP) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,A	1	2	1	0	word (RWi) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW ear,A	2	2	1	0	word (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW eam,A	2+	3 + (a)	0	(c)	word (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW io,A	2	3	0	(c)	word (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RWi+disp8,A	2	5	1	(c)	word ((RWi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RLi+disp8,A	3	10	2	(c)	word ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,ear	2	3	2	0	word (RWi) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,eam	2+	4 + (a)	1	(c)	word (RWi) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVW ear,RWi	2	4	2	0	word (ear) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW eam,RWi	2+	5 + (a)	1	(c)	word (eam) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,#imm16	3	2	1	0	word (RWi) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW io,#imm16	4	5	0	(c)	word (io) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW ear,#imm16	4	2	1	0	word (ear) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW eam,#imm16	4+	4 + (a)	0	(c)	word (eam) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW @AL,AH	2	3	0	(c)	word ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCHW A,ear	2	4	2	0	word (A) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCHW A,eam	2+	5 + (a)	0	2 × (c)	word (A) ↔ (eam)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, ear	2	7	4	0	word (RWi) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, eam	2+	9 + (a)	2	2 × (c)	word (RWi) ↔ (eam)	-	-	-	-	-	-	-	-	-	-
MOVL A,ear	2	4	2	0	long (A) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVL A,eam	2+	5 + (a)	0	(d)	long (A) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVL A,#imm32	5	3	0	0	long (A) ← imm32	-	-	-	-	-	*	*	-	-	-
MOVL ear,A	2	4	2	0	long (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVL eam,A	2+	5 + (a)	0	(d)	long(eam) ← (A)	-	-	-	-	-	*	*	-	-	-

(注意事項) 表中の (a), (c), (d) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-3 加減算命令 (バイト・ワード・ロングワード) 42 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A,#imm8	2	2	0	0	byte (A) \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 × (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 × (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 × (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 × (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 × (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 × (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 × (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 × (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。

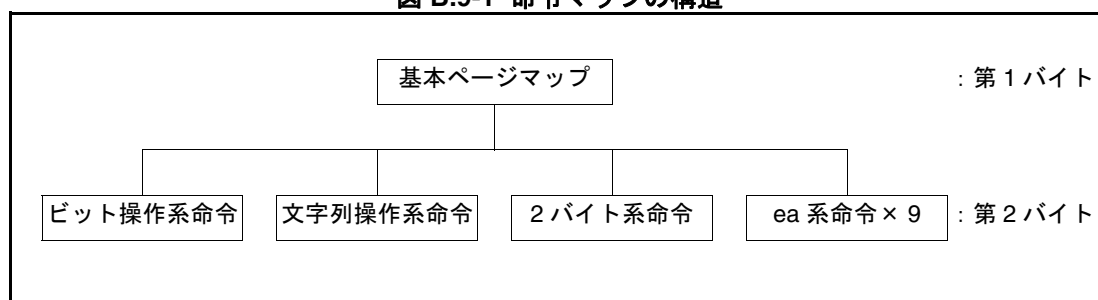
B.9 命令マップ

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

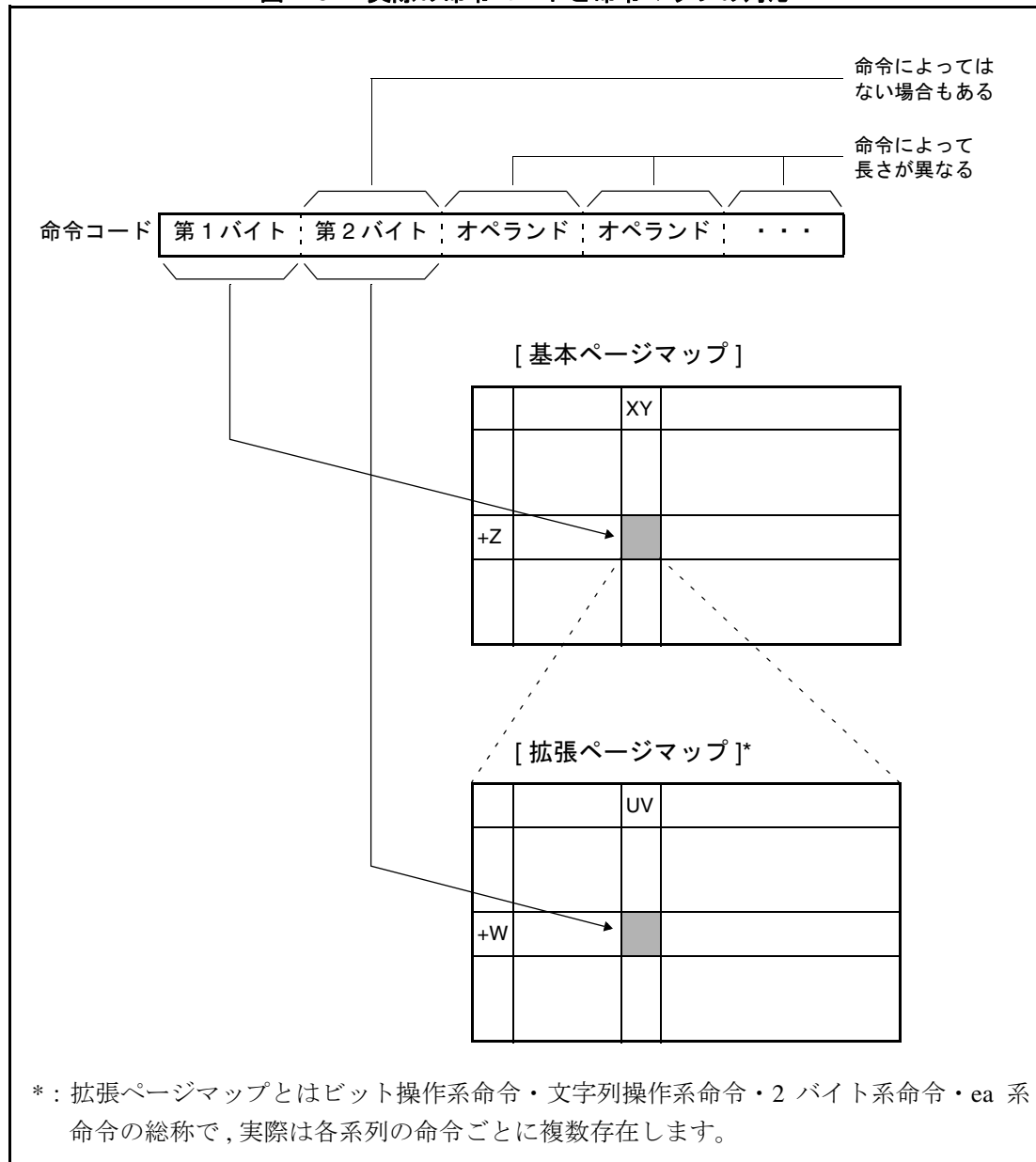
図 B.9-1 命令マップの構造



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	CBNE 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, RWI	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+1, @RW0+1	SUBL A, A, @RW0+1, @RW0+1	SUBL A, A, @RW0+1, @RW0+1	SUBL A, A, @RW0+1, @RW0+1	@RW0+1, @RW0+1 #16, rel	CMPL A, A, @RW0+1, @RW0+1	CMPL A, A, @RW0+1, @RW0+1	CMPL A, A, @RW0+1, @RW0+1	ANDL A, A, @RW0+1, @RW0+1	ANDL A, A, @RW0+1, @RW0+1	ORL A, A, @RW0+1, @RW0+1	ORL A, A, @RW0+1, @RW0+1	XORL A, A, @RW0+1, @RW0+1	XORL A, A, @RW0+1, @RW0+1	@RW0+1, @RW0+1 #8, rel	@RW0+1, @RW0+1 #8, rel
+D	ADDL A, A, @RW1+1, @RW1+1	SUBL A, A, @RW1+1, @RW1+1	SUBL A, A, @RW1+1, @RW1+1	SUBL A, A, @RW1+1, @RW1+1	@RW1+1, @RW1+1 #16, rel	CMPL A, A, @RW1+1, @RW1+1	CMPL A, A, @RW1+1, @RW1+1	CMPL A, A, @RW1+1, @RW1+1	ANDL A, A, @RW1+1, @RW1+1	ANDL A, A, @RW1+1, @RW1+1	ORL A, A, @RW1+1, @RW1+1	ORL A, A, @RW1+1, @RW1+1	XORL A, A, @RW1+1, @RW1+1	XORL A, A, @RW1+1, @RW1+1	@RW1+1, @RW1+1 #8, rel	@RW1+1, @RW1+1 #8, rel
+E	ADDL A, A, @PC+d16	SUBL A, A, @PC+d16	SUBL A, A, @PC+d16	SUBL A, A, @PC+d16	@PC+d16, @PC+d16 #16, rel	CMPL A, A, @PC+d16, @PC+d16	CMPL A, A, @PC+d16, @PC+d16	CMPL A, A, @PC+d16, @PC+d16	ANDL A, A, @PC+d16, @PC+d16	ANDL A, A, @PC+d16, @PC+d16	ORL A, A, @PC+d16, @PC+d16	ORL A, A, @PC+d16, @PC+d16	XORL A, A, @PC+d16, @PC+d16	XORL A, A, @PC+d16, @PC+d16	@PC+d16, @PC+d16 #8, rel	@PC+d16, @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, @RW3+, #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	@RW3+, addr16 #8, rel	@RW3+, 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 @ RL0, #8, @ @RW0+d8	MOVEA A, @ RW0, @ @RW0+d8	MOVEA A, @ RW0, @ @RW0+d8
+1	JMPP @ RL0, @ @RW1+d8	JMPP @ @ RL0, @ @RW1+d8	CALLP @ RL0, @ @RW1+d8	CALLP @ @ RL0, @ @RW1+d8	INCL @ RL0, @ @RW1+d8	INCL @ RL0, @ @RW1+d8	DECL @ RL0, @ @RW1+d8	DECL @ RL0, @ @RW1+d8	MOVL A, @ RL0, @ @RW1+d8	MOVL A, @ RL0, @ @RW1+d8	MOVL @ RL0, A, @ @RW1+d8	MOVL @ RL0, A, @ @RW1+d8	MOV R1, #8, @ @RW1+d8	MOV @ RL0, #8, @ @RW1+d8	MOVEA A, @ RW1, @ @RW1+d8	MOVEA A, @ RW1, @ @RW1+d8
+2	JMPP @ RL1, @ @RW2+d8	JMPP @ @ RL1, @ @RW2+d8	CALLP @ RL1, @ @RW2+d8	CALLP @ @ RL1, @ @RW2+d8	INCL @ RL1, @ @RW2+d8	INCL @ RL1, @ @RW2+d8	DECL @ RL1, @ @RW2+d8	DECL @ RL1, @ @RW2+d8	MOVL A, @ RL1, @ @RW2+d8	MOVL A, @ RL1, @ @RW2+d8	MOVL @ RL1, A, @ @RW2+d8	MOVL @ RL1, A, @ @RW2+d8	MOV R2, #8, @ @RW2+d8	MOV @ RL1, #8, @ @RW2+d8	MOVEA A, @ RW2, @ @RW2+d8	MOVEA A, @ RW2, @ @RW2+d8
+3	JMPP @ RL1, @ @RW3+d8	JMPP @ @ RL1, @ @RW3+d8	CALLP @ RL1, @ @RW3+d8	CALLP @ @ RL1, @ @RW3+d8	INCL @ RL1, @ @RW3+d8	INCL @ RL1, @ @RW3+d8	DECL @ RL1, @ @RW3+d8	DECL @ RL1, @ @RW3+d8	MOVL A, @ RL1, @ @RW3+d8	MOVL A, @ RL1, @ @RW3+d8	MOVL @ RL1, A, @ @RW3+d8	MOVL @ RL1, A, @ @RW3+d8	MOV R3, #8, @ @RW3+d8	MOV @ RL1, #8, @ @RW3+d8	MOVEA A, @ RW3, @ @RW3+d8	MOVEA A, @ RW3, @ @RW3+d8
+4	JMPP @ RL2, @ @RW4+d8	JMPP @ @ RL2, @ @RW4+d8	CALLP @ RL2, @ @RW4+d8	CALLP @ @ RL2, @ @RW4+d8	INCL @ RL2, @ @RW4+d8	INCL @ RL2, @ @RW4+d8	DECL @ RL2, @ @RW4+d8	DECL @ RL2, @ @RW4+d8	MOVL A, @ RL2, @ @RW4+d8	MOVL A, @ RL2, @ @RW4+d8	MOVL @ RL2, A, @ @RW4+d8	MOVL @ RL2, A, @ @RW4+d8	MOV R4, #8, @ @RW4+d8	MOV @ RL2, #8, @ @RW4+d8	MOVEA A, @ RW4, @ @RW4+d8	MOVEA A, @ RW4, @ @RW4+d8
+5	JMPP @ RL2, @ @RW5+d8	JMPP @ @ RL2, @ @RW5+d8	CALLP @ RL2, @ @RW5+d8	CALLP @ @ RL2, @ @RW5+d8	INCL @ RL2, @ @RW5+d8	INCL @ RL2, @ @RW5+d8	DECL @ RL2, @ @RW5+d8	DECL @ RL2, @ @RW5+d8	MOVL A, @ RL2, @ @RW5+d8	MOVL A, @ RL2, @ @RW5+d8	MOVL @ RL2, A, @ @RW5+d8	MOVL @ RL2, A, @ @RW5+d8	MOV R5, #8, @ @RW5+d8	MOV @ RL2, #8, @ @RW5+d8	MOVEA A, @ RW5, @ @RW5+d8	MOVEA A, @ RW5, @ @RW5+d8
+6	JMPP @ RL3, @ @RW6+d8	JMPP @ @ RL3, @ @RW6+d8	CALLP @ RL3, @ @RW6+d8	CALLP @ @ RL3, @ @RW6+d8	INCL @ RL3, @ @RW6+d8	INCL @ RL3, @ @RW6+d8	DECL @ RL3, @ @RW6+d8	DECL @ RL3, @ @RW6+d8	MOVL A, @ RL3, @ @RW6+d8	MOVL A, @ RL3, @ @RW6+d8	MOVL @ RL3, A, @ @RW6+d8	MOVL @ RL3, A, @ @RW6+d8	MOV R6, #8, @ @RW6+d8	MOV @ RL3, #8, @ @RW6+d8	MOVEA A, @ RW6, @ @RW6+d8	MOVEA A, @ RW6, @ @RW6+d8
+7	JMPP @ RL3, @ @RW7+d8	JMPP @ @ RL3, @ @RW7+d8	CALLP @ RL3, @ @RW7+d8	CALLP @ @ RL3, @ @RW7+d8	INCL @ RL3, @ @RW7+d8	INCL @ RL3, @ @RW7+d8	DECL @ RL3, @ @RW7+d8	DECL @ RL3, @ @RW7+d8	MOVL A, @ RL3, @ @RW7+d8	MOVL A, @ RL3, @ @RW7+d8	MOVL @ RL3, A, @ @RW7+d8	MOVL @ RL3, A, @ @RW7+d8	MOV R7, #8, @ @RW7+d8	MOV @ RL3, #8, @ @RW7+d8	MOVEA A, @ RW7, @ @RW7+d8	MOVEA A, @ RW7, @ @RW7+d8
+8	JMPP @ @ @RW0, @ @RW0+d16	JMPP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	INCL @ @RW0, @ @RW0+d16	INCL @ @RW0, @ @RW0+d16	DECL @ @RW0, @ @RW0+d16	DECL @ @RW0, @ @RW0+d16	MOVL A, @ @RW0, @ @RW0+d16	MOVL A, @ @RW0, @ @RW0+d16	MOVL @ @RW0, A, @ @RW0+d16	MOVL @ @RW0, A, @ @RW0+d16	MOV @ @RW0, #8, @ @RW0+d16	MOV @ @RW0, #8, @ @RW0+d16	MOVEA A, @ @RW0, @ @RW0+d16	MOVEA A, @ @RW0, @ @RW0+d16
+9	JMPP @ @ @RW1, @ @RW1+d16	JMPP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	INCL @ @RW1, @ @RW1+d16	INCL @ @RW1, @ @RW1+d16	DECL @ @RW1, @ @RW1+d16	DECL @ @RW1, @ @RW1+d16	MOVL A, @ @RW1, @ @RW1+d16	MOVL A, @ @RW1, @ @RW1+d16	MOVL @ @RW1, A, @ @RW1+d16	MOVL @ @RW1, A, @ @RW1+d16	MOV @ @RW1, #8, @ @RW1+d16	MOV @ @RW1, #8, @ @RW1+d16	MOVEA A, @ @RW1, @ @RW1+d16	MOVEA A, @ @RW1, @ @RW1+d16
+A	JMPP @ @ @RW2, @ @RW2+d16	JMPP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	INCL @ @RW2, @ @RW2+d16	INCL @ @RW2, @ @RW2+d16	DECL @ @RW2, @ @RW2+d16	DECL @ @RW2, @ @RW2+d16	MOVL A, @ @RW2, @ @RW2+d16	MOVL A, @ @RW2, @ @RW2+d16	MOVL @ @RW2, A, @ @RW2+d16	MOVL @ @RW2, A, @ @RW2+d16	MOV @ @RW2, #8, @ @RW2+d16	MOV @ @RW2, #8, @ @RW2+d16	MOVEA A, @ @RW2, @ @RW2+d16	MOVEA A, @ @RW2, @ @RW2+d16
+B	JMPP @ @ @RW3, @ @RW3+d16	JMPP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	INCL @ @RW3, @ @RW3+d16	INCL @ @RW3, @ @RW3+d16	DECL @ @RW3, @ @RW3+d16	DECL @ @RW3, @ @RW3+d16	MOVL A, @ @RW3, @ @RW3+d16	MOVL A, @ @RW3, @ @RW3+d16	MOVL @ @RW3, A, @ @RW3+d16	MOVL @ @RW3, A, @ @RW3+d16	MOV @ @RW3, #8, @ @RW3+d16	MOV @ @RW3, #8, @ @RW3+d16	MOVEA A, @ @RW3, @ @RW3+d16	MOVEA A, @ @RW3, @ @RW3+d16
+C	JMPP @ @ @RW0+, @ @RW0+RW7	JMPP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	INCL @ @RW0+, @ @RW0+RW7	INCL @ @RW0+, @ @RW0+RW7	DECL @ @RW0+, @ @RW0+RW7	DECL @ @RW0+, @ @RW0+RW7	MOVL A, @ @RW0+, @ @RW0+RW7	MOVL A, @ @RW0+, @ @RW0+RW7	MOVL @ @RW0+, A, @ @RW0+RW7	MOVL @ @RW0+, A, @ @RW0+RW7	MOV @ @RW0+, #8, @ @RW0+RW7	MOV @ @RW0+, #8, @ @RW0+RW7	MOVEA A, @ @RW0+, @ @RW0+RW7	MOVEA A, @ @RW0+, @ @RW0+RW7
+D	JMPP @ @ @RW1+, @ @RW1+RW7	JMPP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	INCL @ @RW1+, @ @RW1+RW7	INCL @ @RW1+, @ @RW1+RW7	DECL @ @RW1+, @ @RW1+RW7	DECL @ @RW1+, @ @RW1+RW7	MOVL A, @ @RW1+, @ @RW1+RW7	MOVL A, @ @RW1+, @ @RW1+RW7	MOVL @ @RW1+, A, @ @RW1+RW7	MOVL @ @RW1+, A, @ @RW1+RW7	MOV @ @RW1+, #8, @ @RW1+RW7	MOV @ @RW1+, #8, @ @RW1+RW7	MOVEA A, @ @RW1+, @ @RW1+RW7	MOVEA A, @ @RW1+, @ @RW1+RW7
+E	JMPP @ @ @PC+d16	JMPP @ @ @PC+d16	CALLP @ @ @PC+d16	CALLP @ @ @PC+d16	INCL @ @PC+d16	INCL @ @PC+d16	DECL @ @PC+d16	DECL @ @PC+d16	MOVL A, @ @PC+d16	MOVL A, @ @PC+d16	MOVL @ @PC+d16, A	MOVL @ @PC+d16, A	MOV @ @PC+d16, #8, @ @PC+d16	MOV @ @PC+d16, #8, @ @PC+d16	MOVEA A, @ @PC+d16, #8	MOVEA A, @ @PC+d16, #8
+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	ROL	R0' @RW0+d8	RORC	R0' @RW0+d8	INC	R0' @RW0+d8	DEC	R0' @RW0+d8	MOV	A, R0' @RW0+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+1	ROL	R1' @RW1+d8	RORC	R1' @RW1+d8	INC	R1' @RW1+d8	DEC	R1' @RW1+d8	MOV	A, R1' @RW1+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+2	ROL	R2' @RW2+d8	RORC	R2' @RW2+d8	INC	R2' @RW2+d8	DEC	R2' @RW2+d8	MOV	A, R2' @RW2+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+3	ROL	R3' @RW3+d8	RORC	R3' @RW3+d8	INC	R3' @RW3+d8	DEC	R3' @RW3+d8	MOV	A, R3' @RW3+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+4	ROL	R4' @RW4+d8	RORC	R4' @RW4+d8	INC	R4' @RW4+d8	DEC	R4' @RW4+d8	MOV	A, R4' @RW4+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+5	ROL	R5' @RW5+d8	RORC	R5' @RW5+d8	INC	R5' @RW5+d8	DEC	R5' @RW5+d8	MOV	A, R5' @RW5+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+6	ROL	R6' @RW6+d8	RORC	R6' @RW6+d8	INC	R6' @RW6+d8	DEC	R6' @RW6+d8	MOV	A, R6' @RW6+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+7	ROL	R7' @RW7+d8	RORC	R7' @RW7+d8	INC	R7' @RW7+d8	DEC	R7' @RW7+d8	MOV	A, R7' @RW7+d8	MOV	MOV	MOVX	MOVX	XCH	XCH
+8	ROL	R0' @RW0+d16	RORC	R0' @RW0+d16	INC	R0' @RW0+d16	DEC	R0' @RW0+d16	MOV	A, R0' @RW0+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+9	ROL	R1' @RW1+d16	RORC	R1' @RW1+d16	INC	R1' @RW1+d16	DEC	R1' @RW1+d16	MOV	A, R1' @RW1+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+A	ROL	R2' @RW2+d16	RORC	R2' @RW2+d16	INC	R2' @RW2+d16	DEC	R2' @RW2+d16	MOV	A, R2' @RW2+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+B	ROL	R3' @RW3+d16	RORC	R3' @RW3+d16	INC	R3' @RW3+d16	DEC	R3' @RW3+d16	MOV	A, R3' @RW3+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+C	ROL	R0' @RW0+RW7	RORC	R0' @RW0+RW7	INC	R0' @RW0+RW7	DEC	R0' @RW0+RW7	MOV	A, R0' @RW0+RW7	MOV	MOV	MOVX	MOVX	XCH	XCH
+D	ROL	R1' @RW1+RW7	RORC	R1' @RW1+RW7	INC	R1' @RW1+RW7	DEC	R1' @RW1+RW7	MOV	A, R1' @RW1+RW7	MOV	MOV	MOVX	MOVX	XCH	XCH
+E	ROL	R2' @RW2+PC+d16	RORC	R2' @RW2+PC+d16	INC	R2' @RW2+PC+d16	DEC	R2' @RW2+PC+d16	MOV	A, R2' @RW2+PC+d16	MOV	MOV	MOVX	MOVX	XCH	XCH
+F	ROL	R3' @RW3+addr16	RORC	R3' @RW3+addr16	INC	R3' @RW3+addr16	DEC	R3' @RW3+addr16	MOV	A, R3' @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	ADD A, R0, @RW0+d8	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	CMP A, R0, @RW0+d8	CMP A, R0, @RW0+d8	AND A, R0, @RW0+d8	AND A, R0, @RW0+d8	OR A, R0, @RW0+d8	OR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	DBNZ R0, @RW0+d8, r	DBNZ R0, @RW0+d8, r
+1	ADD A, R1, @RW1+d8	ADD A, R1, @RW1+d8	SUB A, R1, @RW1+d8	SUB A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	ADDC A, R1, @RW1+d8	CMP A, R1, @RW1+d8	CMP A, R1, @RW1+d8	AND A, R1, @RW1+d8	AND A, R1, @RW1+d8	OR A, R1, @RW1+d8	OR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	XOR A, R1, @RW1+d8	DBNZ R1, @RW1+d8, r	DBNZ R1, @RW1+d8, r
+2	ADD A, R2, @RW2+d8	ADD A, R2, @RW2+d8	SUB A, R2, @RW2+d8	SUB A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	ADDC A, R2, @RW2+d8	CMP A, R2, @RW2+d8	CMP A, R2, @RW2+d8	AND A, R2, @RW2+d8	AND A, R2, @RW2+d8	OR A, R2, @RW2+d8	OR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	XOR A, R2, @RW2+d8	DBNZ R2, @RW2+d8, r	DBNZ R2, @RW2+d8, r
+3	ADD A, R3, @RW3+d8	ADD A, R3, @RW3+d8	SUB A, R3, @RW3+d8	SUB A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	ADDC A, R3, @RW3+d8	CMP A, R3, @RW3+d8	CMP A, R3, @RW3+d8	AND A, R3, @RW3+d8	AND A, R3, @RW3+d8	OR A, R3, @RW3+d8	OR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	XOR A, R3, @RW3+d8	DBNZ R3, @RW3+d8, r	DBNZ R3, @RW3+d8, r
+4	ADD A, R4, @RW4+d8	ADD A, R4, @RW4+d8	SUB A, R4, @RW4+d8	SUB A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	ADDC A, R4, @RW4+d8	CMP A, R4, @RW4+d8	CMP A, R4, @RW4+d8	AND A, R4, @RW4+d8	AND A, R4, @RW4+d8	OR A, R4, @RW4+d8	OR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	XOR A, R4, @RW4+d8	DBNZ R4, @RW4+d8, r	DBNZ R4, @RW4+d8, r
+5	ADD A, R5, @RW5+d8	ADD A, R5, @RW5+d8	SUB A, R5, @RW5+d8	SUB A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	ADDC A, R5, @RW5+d8	CMP A, R5, @RW5+d8	CMP A, R5, @RW5+d8	AND A, R5, @RW5+d8	AND A, R5, @RW5+d8	OR A, R5, @RW5+d8	OR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	XOR A, R5, @RW5+d8	DBNZ R5, @RW5+d8, r	DBNZ R5, @RW5+d8, r
+6	ADD A, R6, @RW6+d8	ADD A, R6, @RW6+d8	SUB A, R6, @RW6+d8	SUB A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	ADDC A, R6, @RW6+d8	CMP A, R6, @RW6+d8	CMP A, R6, @RW6+d8	AND A, R6, @RW6+d8	AND A, R6, @RW6+d8	OR A, R6, @RW6+d8	OR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	XOR A, R6, @RW6+d8	DBNZ R6, @RW6+d8, r	DBNZ R6, @RW6+d8, r
+7	ADD A, R7, @RW7+d8	ADD A, R7, @RW7+d8	SUB A, R7, @RW7+d8	SUB A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	ADDC A, R7, @RW7+d8	CMP A, R7, @RW7+d8	CMP A, R7, @RW7+d8	AND A, R7, @RW7+d8	AND A, R7, @RW7+d8	OR A, R7, @RW7+d8	OR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	XOR A, R7, @RW7+d8	DBNZ R7, @RW7+d8, r	DBNZ R7, @RW7+d8, r
+8	ADD A, @RW0, @RW0+d16	ADD A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	SUB A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	ADDC A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	CMP A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	AND A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	OR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	XOR A, @RW0, @RW0+d16	DBNZ @RW0, @RW0+d16, r	DBNZ @RW0, @RW0+d16, r
+9	ADD A, @RW1, @RW1+d16	ADD A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	SUB A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	ADDC A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	CMP A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	AND A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	OR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	XOR A, @RW1, @RW1+d16	DBNZ @RW1, @RW1+d16, r	DBNZ @RW1, @RW1+d16, r
+A	ADD A, @RW2, @RW2+d16	ADD A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	SUB A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	ADDC A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	CMP A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	AND A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	OR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	XOR A, @RW2, @RW2+d16	DBNZ @RW2, @RW2+d16, r	DBNZ @RW2, @RW2+d16, r
+B	ADD A, @RW3, @RW3+d16	ADD A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	SUB A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	ADDC A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	CMP A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	AND A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	OR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	XOR A, @RW3, @RW3+d16	DBNZ @RW3, @RW3+d16, r	DBNZ @RW3, @RW3+d16, r
+C	ADD A, @RW0+, @RW0+RW7	ADD A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	SUB A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	ADDC A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	CMP A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	AND A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	OR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	XOR A, @RW0+, @RW0+RW7	DBNZ @RW0+, @RW0+RW7, r	DBNZ @RW0+, @RW0+RW7, r
+D	ADD A, @RW1+, @RW1+RW7	ADD A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	SUB A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	ADDC A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	CMP A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	AND A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	OR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	XOR A, @RW1+, @RW1+RW7	DBNZ @RW1+, @RW1+RW7, r	DBNZ @RW1+, @RW1+RW7, r
+E	ADD A, @RW2+, @PC+d16	ADD A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	SUB A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	ADDC A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	CMP A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	AND A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	OR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	XOR A, @RW2+, @PC+d16	DBNZ @RW2+, @PC+d16, r	DBNZ @RW2+, @PC+d16, r
+F	ADD A, @RW3+, A, addr16	ADD A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	SUB A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	ADDC A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	CMP A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	AND A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	OR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	XOR A, @RW3+, A, addr16	DBNZ @RW3+, 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, R0' @RW0+d8	MULU A, R0' @RW0+d8	MULUW A, RW0' @RW0+d8	MULUW A, RW0' @RW0+d8	MUL A, R0' @RW0+d8	MUL A, R0' @RW0+d8	MULW A, RW0' @RW0+d8	MULW A, RW0' @RW0+d8	DIVU A, R0' @RW0+d8	DIVU A, R0' @RW0+d8	DIVUW A, RW0' @RW0+d8	DIVUW A, RW0' @RW0+d8	DIV A, R0' @RW0+d8	-DIV A, R0' @RW0+d8	DIVW A, RW0' @RW0+d8	DIVW A, RW0' @RW0+d8
+1	MULU A, R1' @RW1+d8	MULU A, R1' @RW1+d8	MULUW A, RW1' @RW1+d8	MULUW A, RW1' @RW1+d8	MUL A, R1' @RW1+d8	MUL A, R1' @RW1+d8	MULW A, RW1' @RW1+d8	MULW A, RW1' @RW1+d8	DIVU A, R1' @RW1+d8	DIVU A, R1' @RW1+d8	DIVUW A, RW1' @RW1+d8	DIVUW A, RW1' @RW1+d8	DIV A, R1' @RW1+d8	-DIV A, R1' @RW1+d8	DIVW A, RW1' @RW1+d8	DIVW A, RW1' @RW1+d8
+2	MULU A, R2' @RW2+d8	MULU A, R2' @RW2+d8	MULUW A, RW2' @RW2+d8	MULUW A, RW2' @RW2+d8	MUL A, R2' @RW2+d8	MUL A, R2' @RW2+d8	MULW A, RW2' @RW2+d8	MULW A, RW2' @RW2+d8	DIVU A, R2' @RW2+d8	DIVU A, R2' @RW2+d8	DIVUW A, RW2' @RW2+d8	DIVUW A, RW2' @RW2+d8	DIV A, R2' @RW2+d8	-DIV A, R2' @RW2+d8	DIVW A, RW2' @RW2+d8	DIVW A, RW2' @RW2+d8
+3	MULU A, R3' @RW3+d8	MULU A, R3' @RW3+d8	MULUW A, RW3' @RW3+d8	MULUW A, RW3' @RW3+d8	MUL A, R3' @RW3+d8	MUL A, R3' @RW3+d8	MULW A, RW3' @RW3+d8	MULW A, RW3' @RW3+d8	DIVU A, R3' @RW3+d8	DIVU A, R3' @RW3+d8	DIVUW A, RW3' @RW3+d8	DIVUW A, RW3' @RW3+d8	DIV A, R3' @RW3+d8	-DIV A, R3' @RW3+d8	DIVW A, RW3' @RW3+d8	DIVW A, RW3' @RW3+d8
+4	MULU A, R4' @RW4+d8	MULU A, R4' @RW4+d8	MULUW A, RW4' @RW4+d8	MULUW A, RW4' @RW4+d8	MUL A, R4' @RW4+d8	MUL A, R4' @RW4+d8	MULW A, RW4' @RW4+d8	MULW A, RW4' @RW4+d8	DIVU A, R4' @RW4+d8	DIVU A, R4' @RW4+d8	DIVUW A, RW4' @RW4+d8	DIVUW A, RW4' @RW4+d8	DIV A, R4' @RW4+d8	-DIV A, R4' @RW4+d8	DIVW A, RW4' @RW4+d8	DIVW A, RW4' @RW4+d8
+5	MULU A, R5' @RW5+d8	MULU A, R5' @RW5+d8	MULUW A, RW5' @RW5+d8	MULUW A, RW5' @RW5+d8	MUL A, R5' @RW5+d8	MUL A, R5' @RW5+d8	MULW A, RW5' @RW5+d8	MULW A, RW5' @RW5+d8	DIVU A, R5' @RW5+d8	DIVU A, R5' @RW5+d8	DIVUW A, RW5' @RW5+d8	DIVUW A, RW5' @RW5+d8	DIV A, R5' @RW5+d8	-DIV A, R5' @RW5+d8	DIVW A, RW5' @RW5+d8	DIVW A, RW5' @RW5+d8
+6	MULU A, R6' @RW6+d8	MULU A, R6' @RW6+d8	MULUW A, RW6' @RW6+d8	MULUW A, RW6' @RW6+d8	MUL A, R6' @RW6+d8	MUL A, R6' @RW6+d8	MULW A, RW6' @RW6+d8	MULW A, RW6' @RW6+d8	DIVU A, R6' @RW6+d8	DIVU A, R6' @RW6+d8	DIVUW A, RW6' @RW6+d8	DIVUW A, RW6' @RW6+d8	DIV A, R6' @RW6+d8	-DIV A, R6' @RW6+d8	DIVW A, RW6' @RW6+d8	DIVW A, RW6' @RW6+d8
+7	MULU A, R7' @RW7+d8	MULU A, R7' @RW7+d8	MULUW A, RW7' @RW7+d8	MULUW A, RW7' @RW7+d8	MUL A, R7' @RW7+d8	MUL A, R7' @RW7+d8	MULW A, RW7' @RW7+d8	MULW A, RW7' @RW7+d8	DIVU A, R7' @RW7+d8	DIVU A, R7' @RW7+d8	DIVUW A, RW7' @RW7+d8	DIVUW A, RW7' @RW7+d8	DIV A, R7' @RW7+d8	-DIV A, R7' @RW7+d8	DIVW A, RW7' @RW7+d8	DIVW A, RW7' @RW7+d8
+8	MULU A, @RW0' @RW0+d16	MULU A, @RW0' @RW0+d16	MULUW A, @RW0' @RW0+d16	MULUW A, @RW0' @RW0+d16	MUL A, @RW0' @RW0+d16	MUL A, @RW0' @RW0+d16	MULW A, @RW0' @RW0+d16	MULW A, @RW0' @RW0+d16	DIVU A, @RW0' @RW0+d16	DIVU A, @RW0' @RW0+d16	DIVUW A, @RW0' @RW0+d16	DIVUW A, @RW0' @RW0+d16	DIV A, @RW0' @RW0+d16	-DIV A, @RW0' @RW0+d16	DIVW A, @RW0' @RW0+d16	DIVW A, @RW0' @RW0+d16
+9	MULU A, @RW1' @RW1+d16	MULU A, @RW1' @RW1+d16	MULUW A, @RW1' @RW1+d16	MULUW A, @RW1' @RW1+d16	MUL A, @RW1' @RW1+d16	MUL A, @RW1' @RW1+d16	MULW A, @RW1' @RW1+d16	MULW A, @RW1' @RW1+d16	DIVU A, @RW1' @RW1+d16	DIVU A, @RW1' @RW1+d16	DIVUW A, @RW1' @RW1+d16	DIVUW A, @RW1' @RW1+d16	DIV A, @RW1' @RW1+d16	-DIV A, @RW1' @RW1+d16	DIVW A, @RW1' @RW1+d16	DIVW A, @RW1' @RW1+d16
+A	MULU A, @RW2' @RW2+d16	MULU A, @RW2' @RW2+d16	MULUW A, @RW2' @RW2+d16	MULUW A, @RW2' @RW2+d16	MUL A, @RW2' @RW2+d16	MUL A, @RW2' @RW2+d16	MULW A, @RW2' @RW2+d16	MULW A, @RW2' @RW2+d16	DIVU A, @RW2' @RW2+d16	DIVU A, @RW2' @RW2+d16	DIVUW A, @RW2' @RW2+d16	DIVUW A, @RW2' @RW2+d16	DIV A, @RW2' @RW2+d16	-DIV A, @RW2' @RW2+d16	DIVW A, @RW2' @RW2+d16	DIVW A, @RW2' @RW2+d16
+B	MULU A, @RW3' @RW3+d16	MULU A, @RW3' @RW3+d16	MULUW A, @RW3' @RW3+d16	MULUW A, @RW3' @RW3+d16	MUL A, @RW3' @RW3+d16	MUL A, @RW3' @RW3+d16	MULW A, @RW3' @RW3+d16	MULW A, @RW3' @RW3+d16	DIVU A, @RW3' @RW3+d16	DIVU A, @RW3' @RW3+d16	DIVUW A, @RW3' @RW3+d16	DIVUW A, @RW3' @RW3+d16	DIV A, @RW3' @RW3+d16	-DIV A, @RW3' @RW3+d16	DIVW A, @RW3' @RW3+d16	DIVW A, @RW3' @RW3+d16
+C	MULU A, @RW0+ @RW0+RW7	MULU A, @RW0+ @RW0+RW7	MULUW A, @RW0+ @RW0+RW7	MULUW A, @RW0+ @RW0+RW7	MUL A, @RW0+ @RW0+RW7	MUL A, @RW0+ @RW0+RW7	MULW A, @RW0+ @RW0+RW7	MULW A, @RW0+ @RW0+RW7	DIVU A, @RW0+ @RW0+RW7	DIVU A, @RW0+ @RW0+RW7	DIVUW A, @RW0+ @RW0+RW7	DIVUW A, @RW0+ @RW0+RW7	DIV A, @RW0+ @RW0+RW7	-DIV A, @RW0+ @RW0+RW7	DIVW A, @RW0+ @RW0+RW7	DIVW A, @RW0+ @RW0+RW7
+D	MULU A, @RW1+ @RW1+RW7	MULU A, @RW1+ @RW1+RW7	MULUW A, @RW1+ @RW1+RW7	MULUW A, @RW1+ @RW1+RW7	MUL A, @RW1+ @RW1+RW7	MUL A, @RW1+ @RW1+RW7	MULW A, @RW1+ @RW1+RW7	MULW A, @RW1+ @RW1+RW7	DIVU A, @RW1+ @RW1+RW7	DIVU A, @RW1+ @RW1+RW7	DIVUW A, @RW1+ @RW1+RW7	DIVUW A, @RW1+ @RW1+RW7	DIV A, @RW1+ @RW1+RW7	-DIV A, @RW1+ @RW1+RW7	DIVW A, @RW1+ @RW1+RW7	DIVW A, @RW1+ @RW1+RW7
+E	MULU A, @RW2+ @PC+d16	MULU A, @RW2+ @PC+d16	MULUW A, @RW2+ @PC+d16	MULUW A, @RW2+ @PC+d16	MUL A, @RW2+ @PC+d16	MUL A, @RW2+ @PC+d16	MULW A, @RW2+ @PC+d16	MULW A, @RW2+ @PC+d16	DIVU A, @RW2+ @PC+d16	DIVU A, @RW2+ @PC+d16	DIVUW A, @RW2+ @PC+d16	DIVUW A, @RW2+ @PC+d16	DIV A, @RW2+ @PC+d16	-DIV A, @RW2+ @PC+d16	DIVW A, @RW2+ @PC+d16	DIVW A, @RW2+ @PC+d16
+F	MULU A, @RW3+ addr16	MULU A, @RW3+ addr16	MULUW A, @RW3+ addr16	MULUW A, @RW3+ addr16	MUL A, @RW3+ addr16	MUL A, @RW3+ addr16	MULW A, @RW3+ addr16	MULW A, @RW3+ addr16	DIVU A, @RW3+ addr16	DIVU A, @RW3+ addr16	DIVUW A, @RW3+ addr16	DIVUW A, @RW3+ addr16	DIV A, @RW3+ addr16	-DIV A, @RW3+ addr16	DIVW A, @RW3+ addr16	DIVW A, @RW3+ addr16

表 B.9-15 MOVEA RWi, ea 命令 (第1バイト=79_H)[illegible]

表 B.9-16 MOV Ri, ea 命令 (第1バイト = 7A_H)

	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 R2, 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 R2, 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 R2, 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 R2, 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 R2, 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 R2, 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 R2, 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 R2, 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 R2, @RW0+d16	MOV R3, @RW0	MOV R3, @RW0+d16	MOV R4, @RW0	MOV R4, @RW0+d16	MOV R5, @RW0	MOV R5, @RW0+d16	MOV R6, @RW0	MOV R6, @RW0-d16	MOV R7, @RW0	MOV R7, @RW0+d16
+9	MOV R0, @RW1	MOV R0, @RW1+d16	MOV R1, @RW1	MOV R1, @RW1-d16	MOV R2, @RW1	MOV R2, @RW1+d16	MOV R3, @RW1	MOV R3, @RW1+d16	MOV R4, @RW1	MOV R4, @RW1+d16	MOV R5, @RW1	MOV R5, @RW1+d16	MOV R6, @RW1	MOV R6, @RW1-d16	MOV R7, @RW1	MOV R7, @RW1+d16
+A	MOV R0, @RW2	MOV R0, @RW2+d16	MOV R1, @RW2	MOV R1, @RW2-d16	MOV R2, @RW2	MOV R2, @RW2+d16	MOV R3, @RW2	MOV R3, @RW2+d16	MOV R4, @RW2	MOV R4, @RW2+d16	MOV R5, @RW2	MOV R5, @RW2+d16	MOV R6, @RW2	MOV R6, @RW2-d16	MOV R7, @RW2	MOV R7, @RW2+d16
+B	MOV R0, @RW3	MOV R0, @RW3+d16	MOV R1, @RW3	MOV R1, @RW3-d16	MOV R2, @RW3	MOV R2, @RW3+d16	MOV R3, @RW3	MOV R3, @RW3+d16	MOV R4, @RW3	MOV R4, @RW3+d16	MOV R5, @RW3	MOV R5, @RW3+d16	MOV R6, @RW3	MOV R6, @RW3-d16	MOV R7, @RW3	MOV R7, @RW3+d16
+C	MOV R0, @RW0+	MOV R0, @RW0-RW7	MOV R1, @RW0+	MOV R1, @RW0-RW7	MOV R2, @RW0+	MOV R2, @RW0-RW7	MOV R3, @RW0+	MOV R3, @RW0-RW7	MOV R4, @RW0+	MOV R4, @RW0-RW7	MOV R5, @RW0+	MOV R5, @RW0-RW7	MOV R6, @RW0+	MOV R6, @RW0-RW7	MOV R7, @RW0+	MOV R7, @RW0-RW7
+D	MOV R0, @RW1+	MOV R0, @RW1+RW7	MOV R1, @RW1+	MOV R1, @RW1+RW7	MOV R2, @RW1+	MOV R2, @RW1+RW7	MOV R3, @RW1+	MOV R3, @RW1+RW7	MOV R4, @RW1+	MOV R4, @RW1+RW7	MOV R5, @RW1+	MOV R5, @RW1+RW7	MOV R6, @RW1+	MOV R6, @RW1+RW7	MOV R7, @RW1+	MOV R7, @RW1+RW7
+E	MOV R0, @RW2+	MOV R0, @PC+d16	MOV R1, @RW2+	MOV R1, @PC+d16	MOV R2, @RW2+	MOV R2, @PC+d16	MOV R3, @RW2+	MOV R3, @PC+d16	MOV R4, @RW2+	MOV R4, @PC+d16	MOV R5, @RW2+	MOV R5, @PC+d16	MOV R6, @RW2+	MOV R6, @PC+d16	MOV R7, @RW2+	MOV R7, @PC+d16
+F	MOV R0, @RW3+	MOV R0, addr16	MOV R1, @RW3+	MOV R1, addr16	MOV R2, @RW3+	MOV R2, addr16	MOV R3, @RW3+	MOV R3, addr16	MOV R4, @RW3+	MOV R4, addr16	MOV R5, @RW3+	MOV R5, addr16	MOV R6, @RW3+	MOV R6, addr16	MOV R7, @RW3+	MOV R7, addr16

表 B.9-17 MOVW RWi, ea 命令 (第1バイト=7B_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW R0,0, @R0+0	MOVW R0,10, @R0+0	MOVW R1,1, @R0+0	MOVW R1,11, @R0+0	MOVW R2,1, @R0+0	MOVW R2,11, @R0+0	MOVW R3,1, @R0+0	MOVW R3,11, @R0+0	MOVW R4,1, @R0+0	MOVW R4,11, @R0+0	MOVW R5,1, @R0+0	MOVW R5,11, @R0+0	MOVW R6,1, @R0+0	MOVW R6,11, @R0+0	MOVW R7,1, @R0+0	MOVW R7,11, @R0+0
+1	MOVW R0,1, @R1+0	MOVW R0,11, @R1+0	MOVW R1,2, @R1+0	MOVW R1,12, @R1+0	MOVW R2,2, @R1+0	MOVW R2,12, @R1+0	MOVW R3,2, @R1+0	MOVW R3,12, @R1+0	MOVW R4,2, @R1+0	MOVW R4,12, @R1+0	MOVW R5,2, @R1+0	MOVW R5,12, @R1+0	MOVW R6,2, @R1+0	MOVW R6,12, @R1+0	MOVW R7,2, @R1+0	MOVW R7,12, @R1+0
+2	MOVW R0,2, @R2+0	MOVW R0,12, @R2+0	MOVW R1,3, @R2+0	MOVW R1,13, @R2+0	MOVW R2,3, @R2+0	MOVW R2,13, @R2+0	MOVW R3,3, @R2+0	MOVW R3,13, @R2+0	MOVW R4,3, @R2+0	MOVW R4,13, @R2+0	MOVW R5,3, @R2+0	MOVW R5,13, @R2+0	MOVW R6,3, @R2+0	MOVW R6,13, @R2+0	MOVW R7,3, @R2+0	MOVW R7,13, @R2+0
+3	MOVW R0,3, @R3+0	MOVW R0,13, @R3+0	MOVW R1,4, @R3+0	MOVW R1,14, @R3+0	MOVW R2,4, @R3+0	MOVW R2,14, @R3+0	MOVW R3,4, @R3+0	MOVW R3,14, @R3+0	MOVW R4,4, @R3+0	MOVW R4,14, @R3+0	MOVW R5,4, @R3+0	MOVW R5,14, @R3+0	MOVW R6,4, @R3+0	MOVW R6,14, @R3+0	MOVW R7,4, @R3+0	MOVW R7,14, @R3+0
+4	MOVW R0,4, @R4+0	MOVW R0,14, @R4+0	MOVW R1,5, @R4+0	MOVW R1,15, @R4+0	MOVW R2,5, @R4+0	MOVW R2,15, @R4+0	MOVW R3,5, @R4+0	MOVW R3,15, @R4+0	MOVW R4,5, @R4+0	MOVW R4,15, @R4+0	MOVW R5,5, @R4+0	MOVW R5,15, @R4+0	MOVW R6,5, @R4+0	MOVW R6,15, @R4+0	MOVW R7,5, @R4+0	MOVW R7,15, @R4+0
+5	MOVW R0,5, @R5+0	MOVW R0,15, @R5+0	MOVW R1,6, @R5+0	MOVW R1,16, @R5+0	MOVW R2,6, @R5+0	MOVW R2,16, @R5+0	MOVW R3,6, @R5+0	MOVW R3,16, @R5+0	MOVW R4,6, @R5+0	MOVW R4,16, @R5+0	MOVW R5,6, @R5+0	MOVW R5,16, @R5+0	MOVW R6,6, @R5+0	MOVW R6,16, @R5+0	MOVW R7,6, @R5+0	MOVW R7,16, @R5+0
+6	MOVW R0,6, @R6+0	MOVW R0,16, @R6+0	MOVW R1,7, @R6+0	MOVW R1,17, @R6+0	MOVW R2,7, @R6+0	MOVW R2,17, @R6+0	MOVW R3,7, @R6+0	MOVW R3,17, @R6+0	MOVW R4,7, @R6+0	MOVW R4,17, @R6+0	MOVW R5,7, @R6+0	MOVW R5,17, @R6+0	MOVW R6,7, @R6+0	MOVW R6,17, @R6+0	MOVW R7,7, @R6+0	MOVW R7,17, @R6+0
+7	MOVW R0,7, @R7+0	MOVW R0,17, @R7+0	MOVW R1,8, @R7+0	MOVW R1,18, @R7+0	MOVW R2,8, @R7+0	MOVW R2,18, @R7+0	MOVW R3,8, @R7+0	MOVW R3,18, @R7+0	MOVW R4,8, @R7+0	MOVW R4,18, @R7+0	MOVW R5,8, @R7+0	MOVW R5,18, @R7+0	MOVW R6,8, @R7+0	MOVW R6,18, @R7+0	MOVW R7,8, @R7+0	MOVW R7,18, @R7+0
+8	MOVW R0,8, @R0+16	MOVW R0,18, @R0+16	MOVW R1,9, @R0+16	MOVW R1,19, @R0+16	MOVW R2,9, @R0+16	MOVW R2,19, @R0+16	MOVW R3,9, @R0+16	MOVW R3,19, @R0+16	MOVW R4,9, @R0+16	MOVW R4,19, @R0+16	MOVW R5,9, @R0+16	MOVW R5,19, @R0+16	MOVW R6,9, @R0+16	MOVW R6,19, @R0+16	MOVW R7,9, @R0+16	MOVW R7,19, @R0+16
+9	MOVW R0,9, @R1+16	MOVW R0,19, @R1+16	MOVW R1,10, @R1+16	MOVW R1,20, @R1+16	MOVW R2,10, @R1+16	MOVW R2,20, @R1+16	MOVW R3,10, @R1+16	MOVW R3,20, @R1+16	MOVW R4,10, @R1+16	MOVW R4,20, @R1+16	MOVW R5,10, @R1+16	MOVW R5,20, @R1+16	MOVW R6,10, @R1+16	MOVW R6,20, @R1+16	MOVW R7,10, @R1+16	MOVW R7,20, @R1+16
+A	MOVW R0,10, @R2+16	MOVW R0,20, @R2+16	MOVW R1,11, @R2+16	MOVW R1,21, @R2+16	MOVW R2,11, @R2+16	MOVW R2,21, @R2+16	MOVW R3,11, @R2+16	MOVW R3,21, @R2+16	MOVW R4,11, @R2+16	MOVW R4,21, @R2+16	MOVW R5,11, @R2+16	MOVW R5,21, @R2+16	MOVW R6,11, @R2+16	MOVW R6,21, @R2+16	MOVW R7,11, @R2+16	MOVW R7,21, @R2+16
+B	MOVW R0,11, @R3+16	MOVW R0,21, @R3+16	MOVW R1,12, @R3+16	MOVW R1,22, @R3+16	MOVW R2,12, @R3+16	MOVW R2,22, @R3+16	MOVW R3,12, @R3+16	MOVW R3,22, @R3+16	MOVW R4,12, @R3+16	MOVW R4,22, @R3+16	MOVW R5,12, @R3+16	MOVW R5,22, @R3+16	MOVW R6,12, @R3+16	MOVW R6,22, @R3+16	MOVW R7,12, @R3+16	MOVW R7,22, @R3+16
+C	MOVW R0,12, @R0+R7	MOVW R0,22, @R0+R7	MOVW R1,13, @R0+R7	MOVW R1,23, @R0+R7	MOVW R2,13, @R0+R7	MOVW R2,23, @R0+R7	MOVW R3,13, @R0+R7	MOVW R3,23, @R0+R7	MOVW R4,13, @R0+R7	MOVW R4,23, @R0+R7	MOVW R5,13, @R0+R7	MOVW R5,23, @R0+R7	MOVW R6,13, @R0+R7	MOVW R6,23, @R0+R7	MOVW R7,13, @R0+R7	MOVW R7,23, @R0+R7
+D	MOVW R0,13, @R1+R7	MOVW R0,23, @R1+R7	MOVW R1,14, @R1+R7	MOVW R1,24, @R1+R7	MOVW R2,14, @R1+R7	MOVW R2,24, @R1+R7	MOVW R3,14, @R1+R7	MOVW R3,24, @R1+R7	MOVW R4,14, @R1+R7	MOVW R4,24, @R1+R7	MOVW R5,14, @R1+R7	MOVW R5,24, @R1+R7	MOVW R6,14, @R1+R7	MOVW R6,24, @R1+R7	MOVW R7,14, @R1+R7	MOVW R7,24, @R1+R7
+E	MOVW R0,14, @PC+16	MOVW R0,24, @PC+16	MOVW R1,15, @PC+16	MOVW R1,25, @PC+16	MOVW R2,15, @PC+16	MOVW R2,25, @PC+16	MOVW R3,15, @PC+16	MOVW R3,25, @PC+16	MOVW R4,15, @PC+16	MOVW R4,25, @PC+16	MOVW R5,15, @PC+16	MOVW R5,25, @PC+16	MOVW R6,15, @PC+16	MOVW R6,25, @PC+16	MOVW R7,15, @PC+16	MOVW R7,25, @PC+16
+F	MOVW R0,15, @R0+16	MOVW R0,25, @R0+16	MOVW R1,16, @R0+16	MOVW R1,26, @R0+16	MOVW R2,16, @R0+16	MOVW R2,26, @R0+16	MOVW R3,16, @R0+16	MOVW R3,26, @R0+16	MOVW R4,16, @R0+16	MOVW R4,26, @R0+16	MOVW R5,16, @R0+16	MOVW R5,26, @R0+16	MOVW R6,16, @R0+16	MOVW R6,26, @R0+16	MOVW R7,16, @R0+16	MOVW R7,26, @R0+16

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)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW @RW0, RW0, @RW0+d8, RW0	MOVW @RW0, RW0, @RW0+d8, RW0	MOVW @RW0, RW1, @RW0+d8, RW1	MOVW @RW0, RW1, @RW0+d8, RW1	MOVW @RW0, RW2, @RW0+d8, RW2	MOVW @RW0, RW2, @RW0+d8, RW2	MOVW @RW0, RW3, @RW0+d8, RW3	MOVW @RW0, RW3, @RW0+d8, RW3	MOVW @RW0, RW4, @RW0+d8, RW4	MOVW @RW0, RW4, @RW0+d8, RW4	MOVW @RW0, RW5, @RW0+d8, RW5	MOVW @RW0, RW5, @RW0+d8, RW5	MOVW @RW0, RW6, @RW0+d8, RW6	MOVW @RW0, RW6, @RW0+d8, RW6	MOVW @RW0, RW7, @RW0+d8, RW7	MOVW @RW0, RW7, @RW0+d8, RW7
+1	MOVW @RW1, RW0, @RW1+d8, RW0	MOVW @RW1, RW0, @RW1+d8, RW0	MOVW @RW1, RW1, @RW1+d8, RW1	MOVW @RW1, RW1, @RW1+d8, RW1	MOVW @RW1, RW2, @RW1+d8, RW2	MOVW @RW1, RW2, @RW1+d8, RW2	MOVW @RW1, RW3, @RW1+d8, RW3	MOVW @RW1, RW3, @RW1+d8, RW3	MOVW @RW1, RW4, @RW1+d8, RW4	MOVW @RW1, RW4, @RW1+d8, RW4	MOVW @RW1, RW5, @RW1+d8, RW5	MOVW @RW1, RW5, @RW1+d8, RW5	MOVW @RW1, RW6, @RW1+d8, RW6	MOVW @RW1, RW6, @RW1+d8, RW6	MOVW @RW1, RW7, @RW1+d8, RW7	MOVW @RW1, RW7, @RW1+d8, RW7
+2	MOVW @RW2, RW0, @RW2+d8, RW0	MOVW @RW2, RW0, @RW2+d8, RW0	MOVW @RW2, RW1, @RW2+d8, RW1	MOVW @RW2, RW1, @RW2+d8, RW1	MOVW @RW2, RW2, @RW2+d8, RW2	MOVW @RW2, RW2, @RW2+d8, RW2	MOVW @RW2, RW3, @RW2+d8, RW3	MOVW @RW2, RW3, @RW2+d8, RW3	MOVW @RW2, RW4, @RW2+d8, RW4	MOVW @RW2, RW4, @RW2+d8, RW4	MOVW @RW2, RW5, @RW2+d8, RW5	MOVW @RW2, RW5, @RW2+d8, RW5	MOVW @RW2, RW6, @RW2+d8, RW6	MOVW @RW2, RW6, @RW2+d8, RW6	MOVW @RW2, RW7, @RW2+d8, RW7	MOVW @RW2, RW7, @RW2+d8, RW7
+3	MOVW @RW3, RW0, @RW3+d8, RW0	MOVW @RW3, RW0, @RW3+d8, RW0	MOVW @RW3, RW1, @RW3+d8, RW1	MOVW @RW3, RW1, @RW3+d8, RW1	MOVW @RW3, RW2, @RW3+d8, RW2	MOVW @RW3, RW2, @RW3+d8, RW2	MOVW @RW3, RW3, @RW3+d8, RW3	MOVW @RW3, RW3, @RW3+d8, RW3	MOVW @RW3, RW4, @RW3+d8, RW4	MOVW @RW3, RW4, @RW3+d8, RW4	MOVW @RW3, RW5, @RW3+d8, RW5	MOVW @RW3, RW5, @RW3+d8, RW5	MOVW @RW3, RW6, @RW3+d8, RW6	MOVW @RW3, RW6, @RW3+d8, RW6	MOVW @RW3, RW7, @RW3+d8, RW7	MOVW @RW3, RW7, @RW3+d8, RW7
+4	MOVW @RW4, RW0, @RW4+d8, RW0	MOVW @RW4, RW0, @RW4+d8, RW0	MOVW @RW4, RW1, @RW4+d8, RW1	MOVW @RW4, RW1, @RW4+d8, RW1	MOVW @RW4, RW2, @RW4+d8, RW2	MOVW @RW4, RW2, @RW4+d8, RW2	MOVW @RW4, RW3, @RW4+d8, RW3	MOVW @RW4, RW3, @RW4+d8, RW3	MOVW @RW4, RW4, @RW4+d8, RW4	MOVW @RW4, RW4, @RW4+d8, RW4	MOVW @RW4, RW5, @RW4+d8, RW5	MOVW @RW4, RW5, @RW4+d8, RW5	MOVW @RW4, RW6, @RW4+d8, RW6	MOVW @RW4, RW6, @RW4+d8, RW6	MOVW @RW4, RW7, @RW4+d8, RW7	MOVW @RW4, RW7, @RW4+d8, RW7
+5	MOVW @RW5, RW0, @RW5+d8, RW0	MOVW @RW5, RW0, @RW5+d8, RW0	MOVW @RW5, RW1, @RW5+d8, RW1	MOVW @RW5, RW1, @RW5+d8, RW1	MOVW @RW5, RW2, @RW5+d8, RW2	MOVW @RW5, RW2, @RW5+d8, RW2	MOVW @RW5, RW3, @RW5+d8, RW3	MOVW @RW5, RW3, @RW5+d8, RW3	MOVW @RW5, RW4, @RW5+d8, RW4	MOVW @RW5, RW4, @RW5+d8, RW4	MOVW @RW5, RW5, @RW5+d8, RW5	MOVW @RW5, RW5, @RW5+d8, RW5	MOVW @RW5, RW6, @RW5+d8, RW6	MOVW @RW5, RW6, @RW5+d8, RW6	MOVW @RW5, RW7, @RW5+d8, RW7	MOVW @RW5, RW7, @RW5+d8, RW7
+6	MOVW @RW6, RW0, @RW6+d8, RW0	MOVW @RW6, RW0, @RW6+d8, RW0	MOVW @RW6, RW1, @RW6+d8, RW1	MOVW @RW6, RW1, @RW6+d8, RW1	MOVW @RW6, RW2, @RW6+d8, RW2	MOVW @RW6, RW2, @RW6+d8, RW2	MOVW @RW6, RW3, @RW6+d8, RW3	MOVW @RW6, RW3, @RW6+d8, RW3	MOVW @RW6, RW4, @RW6+d8, RW4	MOVW @RW6, RW4, @RW6+d8, RW4	MOVW @RW6, RW5, @RW6+d8, RW5	MOVW @RW6, RW5, @RW6+d8, RW5	MOVW @RW6, RW6, @RW6+d8, RW6	MOVW @RW6, RW6, @RW6+d8, RW6	MOVW @RW6, RW7, @RW6+d8, RW7	MOVW @RW6, RW7, @RW6+d8, RW7
+7	MOVW @RW7, RW0, @RW7+d8, RW0	MOVW @RW7, RW0, @RW7+d8, RW0	MOVW @RW7, RW1, @RW7+d8, RW1	MOVW @RW7, RW1, @RW7+d8, RW1	MOVW @RW7, RW2, @RW7+d8, RW2	MOVW @RW7, RW2, @RW7+d8, RW2	MOVW @RW7, RW3, @RW7+d8, RW3	MOVW @RW7, RW3, @RW7+d8, RW3	MOVW @RW7, RW4, @RW7+d8, RW4	MOVW @RW7, RW4, @RW7+d8, RW4	MOVW @RW7, RW5, @RW7+d8, RW5	MOVW @RW7, RW5, @RW7+d8, RW5	MOVW @RW7, RW6, @RW7+d8, RW6	MOVW @RW7, RW6, @RW7+d8, RW6	MOVW @RW7, RW7, @RW7+d8, RW7	MOVW @RW7, RW7, @RW7+d8, RW7
+8	MOVW @RW0, RW0, +d16, RW0	MOVW @RW0, RW0, +d16, RW0	MOVW @RW0, RW1, +d16, RW1	MOVW @RW0, RW1, +d16, RW1	MOVW @RW0, RW2, +d16, RW2	MOVW @RW0, RW2, +d16, RW2	MOVW @RW0, RW3, +d16, RW3	MOVW @RW0, RW3, +d16, RW3	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW4, +d16, RW4	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW5, +d16, RW5	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW6, +d16, RW6	MOVW @RW0, RW7, +d16, RW7	MOVW @RW0, RW7, +d16, RW7
+9	MOVW @RW1, RW0, +d16, RW0	MOVW @RW1, RW0, +d16, RW0	MOVW @RW1, RW1, +d16, RW1	MOVW @RW1, RW1, +d16, RW1	MOVW @RW1, RW2, +d16, RW2	MOVW @RW1, RW2, +d16, RW2	MOVW @RW1, RW3, +d16, RW3	MOVW @RW1, RW3, +d16, RW3	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW4, +d16, RW4	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW5, +d16, RW5	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW6, +d16, RW6	MOVW @RW1, RW7, +d16, RW7	MOVW @RW1, RW7, +d16, RW7
+A	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW7, +d16, RW7
+B	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW7, +d16, RW7
+C	MOVW @RW0, RW0, +RW7, RW0	MOVW @RW0, RW0, +RW7, RW0	MOVW @RW0, RW1, +RW7, RW1	MOVW @RW0, RW1, +RW7, RW1	MOVW @RW0, RW2, +RW7, RW2	MOVW @RW0, RW2, +RW7, RW2	MOVW @RW0, RW3, +RW7, RW3	MOVW @RW0, RW3, +RW7, RW3	MOVW @RW0, RW4, +RW7, RW4	MOVW @RW0, RW4, +RW7, RW4	MOVW @RW0, RW5, +RW7, RW5	MOVW @RW0, RW5, +RW7, RW5	MOVW @RW0, RW6, +RW7, RW6	MOVW @RW0, RW6, +RW7, RW6	MOVW @RW0, RW7, +RW7, RW7	MOVW @RW0, RW7, +RW7, RW7
+D	MOVW @RW1, RW0, +RW7, RW0	MOVW @RW1, RW0, +RW7, RW0	MOVW @RW1, RW1, +RW7, RW1	MOVW @RW1, RW1, +RW7, RW1	MOVW @RW1, RW2, +RW7, RW2	MOVW @RW1, RW2, +RW7, RW2	MOVW @RW1, RW3, +RW7, RW3	MOVW @RW1, RW3, +RW7, RW3	MOVW @RW1, RW4, +RW7, RW4	MOVW @RW1, RW4, +RW7, RW4	MOVW @RW1, RW5, +RW7, RW5	MOVW @RW1, RW5, +RW7, RW5	MOVW @RW1, RW6, +RW7, RW6	MOVW @RW1, RW6, +RW7, RW6	MOVW @RW1, RW7, +RW7, RW7	MOVW @RW1, RW7, +RW7, RW7
+E	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW0, +d16, RW0	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW1, +d16, RW1	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW2, +d16, RW2	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW3, +d16, RW3	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW4, +d16, RW4	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW5, +d16, RW5	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW6, +d16, RW6	MOVW @RW2, RW7, +d16, RW7	MOVW @RW2, RW7, +d16, RW7
+F	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW0, +d16, RW0	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW1, +d16, RW1	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW2, +d16, RW2	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW3, +d16, RW3	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW4, +d16, RW4	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW5, +d16, RW5	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW6, +d16, RW6	MOVW @RW3, RW7, +d16, RW7	MOVW @RW3, RW7, +d16, RW7

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×1ch動作
 - 8ビット×2ch, 16ビット×1ch動作..... 225
- 16ビットアウトプットコンペア
 - 16ビットアウトプットコンペアのタイミング..... 178
 - 16ビットアウトプットコンペアの動作... 177
- 16ビットインプットキャプチャ
 - 16ビットインプットキャプチャの動作... 179
- 16ビット入出力タイマ
 - 16ビット入出力タイマのブロックダイアグラム..... 158
 - 16ビット入出力タイマのレジスタ..... 159
- 16ビット入出力タイマ部
 - 16ビット入出力タイマ部全体のレジスタ一覧..... 157
- 16ビットフリーランタイマ
 - 16ビットフリーランタイマ..... 161
 - 16ビットフリーランタイマ(×1) 156
 - 16ビットフリーランタイマのカウントタイミング..... 176
 - 16ビットフリーランタイマの動作..... 175
 - 16ビットフリーランタイマのブロックダイアグラム..... 161
 - 16ビットフリーランタイマのレジスタ一覧..... 161

8

- 8/16ビットPPG
 - 8/16ビットPPGの概要..... 182
 - 8/16ビットPPGの動作..... 193
 - 8/16ビットPPGの動作モード..... 194
 - 8/16ビットPPGのレジスタ..... 185
 - 8/16ビットPPGの割込み..... 199
- 8/16ビットアップダウンカウンタ／タイマ
 - 8/16ビットアップダウンカウンタ／タイマのカウントモード選択..... 217
 - 8/16ビットアップダウンカウンタ／タイマの機能..... 202
 - 8/16ビットアップダウンカウンタ／タイマのブロックダイアグラム..... 204
 - 8/16ビットアップダウンカウンタ／タイマのリロード機能とコンペア機能..... 220
 - 8/16ビットアップダウンカウンタ／タイマのレジスタ一覧..... 206
- 8ビット×2ch動作
 - 8ビット×2ch, 16ビット×1ch動作..... 225
- 8ビットPPG
 - 8ビットPPGのブロックダイアグラム..... 183

<アルファベット>

A

- A
 - アキュムレータ(A)..... 28
- A/Dコンバータ
 - A/Dコンバータ使用上の注意..... 259
 - A/Dコンバータの概要..... 244
 - A/Dコンバータのブロックダイアグラム... 245
 - A/Dコンバータのレジスタ..... 246
- ADB
 - アディショナルバンクレジスタ(ADB)..... 24
- ADCR
 - データレジスタ(ADCR1, ADCR0)..... 251
- ADCS
 - コントロールステータスレジスタ(ADCS0, 1)..... 247
- ADE
 - アナログ入力許可レジスタ(ADER)..... 135
- ARSR
 - 自動レディ機能選択レジスタ(ARSR)..... 115

B

- BAP
 - バッファアドレスポインタ (BAP)..... 66

C

- C
 - キャリーフラグ(C)..... 32
- CCR
 - コンディションコードレジスタ(CCR)..... 31
- CDCR
 - 通信プリスケアラレジスタ(CDCR)..... 280
- CLKR
 - クロック出力許可レジスタ(CLKR)..... 337
- CLK同期モード
 - CLK同期モードの転送データフォーマット..... 286
- CLK同期モード
 - CLK同期モードを使用する場合..... 286
- CMR
 - コモンレジスタバンクプリフィックス(CMR)..... 40
- CPU間欠動作機能
 - CPU間欠動作機能..... 80
- CSCRO～6
 - チップセレクトコントロールレジスタ(CSCRO～6)..... 331

D

- D/Aコントロールレジスタ
 - D/Aコントロールレジスタ (DACR0/1)..... 264
- D/Aコンバータ
 - D/Aコンバータの動作..... 266
 - D/Aコンバータのブロックダイアグラム... 263
 - D/Aコンバータのレジスタ一覧..... 262
- D/Aコンバータレジスタ
 - D/Aコンバータレジスタ (DADR0, 1)..... 264
- DACR
 - D/Aコントロールレジスタ (DACR0/1)..... 264
- DADR
 - D/Aコンバータレジスタ (DADR0, 1)..... 264
- DCT
 - データカウンタ (DCT)..... 64
- DDR
 - ポート方向レジスタ (DDR)..... 132
- DIV A, Ri
 - 「DIV A, Ri」, 「DIVW A, RWi」命令の使用上の注意..... 43
 - 「DIV A, Ri」, 「DIVW A, RWi」命令の注意事項の回避について..... 44
- DIVW A, RWi
 - 「DIV A, Ri」¥, 「DIVW A, RWi」命令の使用上の注意..... 43
 - 「DIV A, Ri」, 「DIVW A, RWi」命令の注意事項の回避について..... 44
- DPR
 - ダイレクトページレジスタ (DPR)..... 35
- DQ3
 - セクタ消去タイムフラグ (DQ3)..... 363
- DQ5
 - タイミングリミット超過フラグ (DQ5)..... 362
- DQ6
 - トグルビットフラグ (DQ6)..... 361
- DQ7
 - データポーリングフラグ (DQ7)..... 360
- DTB
 - データバンクレジスタ (DTB)..... 24
- DTP
 - DTP/外部割込みの動作..... 233
 - DTP/外部割込みのレジスタ..... 229
- DTP
 - DTP/外部割込み機能..... 228
 - DTP/外部割込みのブロックダイアグラム..... 228
 - DTPの動作..... 233
 - DTP/外部割込みの動作..... 236
- DTP要求
 - 外部割込み要求とDTP要求の切替え..... 234
- DTP/割込み許可レジスタ
 - DTP/割込み許可レジスタ (ENIR)..... 230
- DTP/割込み要因レジスタ
 - DTP/割込み要因レジスタ (EIRR)..... 231

E

- E²PROMメモリマップ
 - E²PROMメモリマップ..... 345
- ECSR
 - バス制御信号選択レジスタ (ECSR)..... 118
- EI²OS
 - 停止モード時のEI20S起動例..... 258
- EI²OS
 - EI²OSを使った変換動作..... 255
 - 拡張インテリジェントI/Oサービス (I20S)..... 293
 - 拡張インテリジェントI/Oサービス (EI²OS) の概要..... 58
 - 拡張インテリジェントI/Oサービス (EI20S) の構造..... 59
 - 拡張インテリジェントI/Oサービス (EI²OS) の動作..... 59
 - 単発モード時のEI20Sの..... 256
 - 連続モード時のEI20Sの起動例..... 257
- EI²OSステータスレジスタ (ISCS)
 - EI²OSステータスレジスタ (ISCS)..... 65
- EIRR
 - DTP/割込み要因レジスタ (EIRR)..... 231
- ELVR
 - 要求レベル設定レジスタ (ELVR)..... 232
- ENIR
 - DTP/割込み許可レジスタ (ENIR)..... 230

F

- F²MC-16LX
 - F²MC-16LX命令一覧表..... 419
- 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)..... 356
- FPT-120P-M05
 - FPT-120P-M05の外形寸法図..... 7
- FPT-120P-M13
 - FPT-120P-M13の外形寸法図..... 8
- FPT-120P-M21
 - FPT-120P-M21の外形寸法図..... 9

H

- HACR
 - 外部アドレス出力制御レジスタ (HACR).... 117

I

- I
 - 割込み許可フラグ (I)..... 31
- I/O拡張シリアルインタフェース
 - I/O拡張シリアルインタフェースの概要... 296
 - I/O拡張シリアルインタフェースの動作... 303

I/O拡張シリアルインタフェースのブロックダイアグラム..... 296
 I/O拡張シリアルインタフェースのレジスタ..... 297
 I/O拡張シリアルインタフェースの割り込み機能..... 310

I/Oポート
 I/Oポートの概要..... 128
 I/Oポートのレジスタ一覧..... 129

I/Oマップ
 I/Oマップ..... 390

I/Oレジスタアドレスポインタ (IOA)
 I/Oレジスタアドレスポインタ (IOA)..... 64

I²Cインタフェース
 I²Cインタフェースの動作..... 325
 I²Cインタフェースの動作フロー..... 327
 I²Cインタフェースの特長..... 312
 I²Cインタフェースのブロックダイアグラム..... 313
 I²Cインタフェースのレジスタ..... 314

IADR
 アドレスレジスタ (IADR) 323

IBCR
 バスコントロールレジスタ (IBCR)..... 318

IBSR
 バスステータスレジスタ (IBSR)..... 315

ICCR
 クロックコントロールレジスタ (ICCR)..... 321

ICR
 割り込み制御レジスタ (ICR) 60

ICS01
 インพุットキャプチャコントロールステータスレジスタ (ICS01)..... 173

IDAR
 データレジスタ (IDAR)..... 324

ILM
 インタラプトレベルマスクレジスタ (ILM)..... 33

INT
 SCC, MSS, INTビットの競合についての注意..... 320

INT9
 INT9割り込み..... 346

IOA
 I/Oレジスタアドレスポインタ (IOA)..... 64

IPCP0,1
 インพุットキャプチャデータレジスタ (IPCP0,1)..... 172

ISCS
 EI²OSステータスレジスタ (ISCS) 65

ISD
 拡張インテリジェントI/Oサービスディスクリプタ (ISD) 63

L

LPMCR
 低消費電力モード制御レジスタ (LPMCR)..... 84

M

MB90F574/A
 MB90F574/Aシリアル書き込み接続の基本構成..... 378

MSS

SCC, MSS, INTビットの競合についての... 320

N**N**

ネガティブフラグ (N)..... 31

NCC

フラグ変化抑止プリフィックス (NCC)..... 40

O**OCCP0~3**

アウトプットコンペアレジスタ (OCCP0~3)..... 167

OCS0~3

アウトプットコンペアコントロールステータスレジスタ (OCS0~3)..... 168

ODR

出力端子レジスタ (ODR)..... 133
 出力端子レジスタ (ODR)に関する注意..... 133
 出力端子レジスタ (ODR) のブロックダイアグラム..... 133

P**PACSR**

プログラムアドレス検出コントロールレジスタ (PACSR)..... 343

PADR

プログラムアドレス検出レジスタ (PADR0,1)..... 342

PC

プログラムカウンタ (PC)..... 34

PCB

プログラムバンクレジスタ (PCB)..... 23

PDR

ポートデータレジスタ (PDR)..... 130

PPG0,1出力端子制御レジスタ

PPG0,1出力端子制御レジスタ (PPGOE)..... 190

PPG0動作モード制御レジスタ

PPG0動作モード制御レジスタ (PPGC0)..... 186

PPG1動作モード制御レジスタ

PPG1動作モード制御レジスタ (PPGC1)..... 188

PPGC0

PPG0動作モード制御レジスタ (PPGC0)..... 186

PPGC1

PPG1動作モード制御レジスタ (PPGC1)..... 188

PPGOE

PPG0,1出力端子制御レジスタ (PPGOE)..... 190

PPG出力動作

8/16ビットPPGの出力動作..... 195

PRLL/PRLH

リロードレジスタ (PRLL/PRLH)..... 192

PS

プロセッサステータス (PS)..... 31

R

RDR

入力抵抗レジスタ (RDR)..... 134

ROMM

ROMミラー機能選択レジスタ (ROMM)..... 351

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..... 350

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)..... 351

RP

レジスタバンクポインタ (RP)..... 32

S

S

スタックフラグ (S)..... 31

SCC

SCC, MSS, INTビットの競合についての注意
..... 320

SCR

シリアルコントロールレジスタ (SCR)..... 273

SDR

シリアルシフトデータレジスタ (SDR)..... 302

SIDR

シリアルインプットデータレジスタ (SIDR)/シ
リアルアウトプットデータレジスタ
(SODR)..... 276

SMCS

シリアルモードコントロールステータスレジ
スタ (SMCS)..... 298

SMR

シリアルモードレジスタ (SMR)..... 271

SODR

シリアルインプットデータレジスタ (SIDR)/シ
リアルアウトプットデータレジスタ
(SODR)..... 276

SSB

システムスタックバンクレジスタ (SSB) ... 24

SSP

ユーザスタックポインタ (USP) とシステムス
タックポインタ (SSP)..... 30

SSR

シリアルステータスレジスタ (SSR)..... 277

STOP状態

STOP状態..... 305

T

T

スティッキイビットフラグ (T)..... 31

TBTC

タイムベースタイマ制御レジスタ (TBTC)
..... 139

TCCS

タイマカウンタコントロールステータスレジ
スタ (TCCS)..... 163

TCDT

タイマカウンタデータレジスタ (TCDT).... 162

U

UART

UARTの割込み要因..... 288

UART

UARTの応用例 (モード1使用時のシステム構築
例)..... 292
UARTの使用上の注意..... 293
UARTの通信フローチャート..... 292
UARTの動作..... 284
UARTの特長..... 268
UARTのフラグ (PE/ORE/FRE/RDRF/TD)..... 288
UARTのブロックダイヤグラム..... 269
UARTのレジスタ..... 270
UARTの割込みとフラグのセットタイミング
..... 289

UDCRレジスタ

UDCRレジスタへのデータの書込み..... 224

USB

ユーザスタックバンクレジスタ (USB)..... 24

USP

ユーザスタックポインタ (USP) とシステムス
タックポインタ (SSP)..... 30

V

V

オーバフローフラグ (V)..... 32

W

WDTC

ウォッチドッグタイマ制御レジスタ (WDTC)
..... 145

WTC

時計タイマ制御レジスタ (WTC)..... 151

Z

Z

ゼロフラグ (Z)..... 31

<五十音>

あ

アービトレーション	
アービトレーション.....	325
アウトプットコンペア	
アウトプットコンペア.....	165
アウトプットコンペア (×4)	156
アウトプットコンペアのブロックダイアグラム.....	166
アウトプットコンペアのレジスタ一覧....	165
アウトプットコンペアコントロールステータスレジスタ	
アウトプットコンペアコントロールステータスレジスタ (OCS0~3).....	168
アウトプットコンペアレジスタ	
アウトプットコンペアレジスタ (OCCP0~3)	167
アキュムレータ	
アキュムレータ (A).....	28
アクセスモード	
アクセスモード.....	108
アクノリッジ	
アクノリッジ.....	326
アップダウンカウンタレジスタ	
アップダウンカウンタレジスタ 0, 1 (UDCR0, 1)	207
アディショナルバンクレジスタ	
アディショナルバンクレジスタ (ADB).....	24
アドレス一致検出機能	
アドレス一致検出機能のシステム構成例	345
アドレス一致検出機能の動作.....	344
アドレス一致検出機能の動作上の注意....	344
アドレス一致検出機能のブロックダイアグラム.....	340
アドレス一致検出機能のレジスタ.....	341
アドレスレジスタ	
アドレスレジスタ (IADR)	323
アドレッシング	
アドレッシング.....	325, 396
間接アドレッシング	405
直接アドレッシング	398
アナログ入力許可レジスタ	
アナログ入力許可レジスタ (ADER).....	135

い

インターバル割込み機能	
インターバル割込み機能.....	141
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)	33
インプットキャプチャ	
インプットキャプチャ.....	171

インプットキャプチャ (×2)	157
インプットキャプチャの全体ブロックダイアグラム.....	171
インプットキャプチャ全体のレジスタ一覧	171
インプットキャプチャ入力タイミング....	179
インプットキャプチャコントロールステータスレジスタ	
インプットキャプチャコントロールステータスレジスタ (ICS01).....	173
インプットキャプチャデータレジスタ	
インプットキャプチャデータレジスタ (IPCP0, 1).....	172

う

ウォッチドッグ	
ウォッチドッグ停止.....	147
ウォッチドッグタイマ	
ウォッチドッグタイマの起動方法.....	147
ウォッチドッグタイマのクリア.....	147
ウォッチドッグタイマのブロックダイアグラム.....	144
ウォッチドッグタイマのレジスタ一覧....	144
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)	145
ウォッチドッグタイマリセット	
ウォッチドッグタイマリセットの阻止....	147

お

オーバフローフラグ	
オーバフローフラグ (V).....	32

か

外形寸法図	
FPT-120P-M05の外形寸法図.....	7
FPT-120P-M13の外形寸法図.....	8
FPT-120P-M21の外形寸法図.....	9
外部アドレス出力制御レジスタ	
外部アドレス出力制御レジスタ (HACR)....	117
外部クロック	
外部クロック.....	283
外部シフトクロックモード	
外部シフトクロックモード.....	304
外部バス端子制御回路	
外部メモリアクセス (外部バス端子制御回路)	113
外部メモリアクセス	
外部メモリアクセス (外部バス端子制御回路)	113
外部メモリアクセスのブロックダイアグラム	114

外部メモリアクセス制御信号	
外部メモリアクセス制御信号.....	121
外部メモリアクセスレジスタ	
外部メモリアクセスレジスタ一覧.....	113
外部割込み	
DTP/外部割込みの動作.....	233
DTP/外部割込み機能.....	228
DTP/外部割込みの動作手順.....	236
DTP/外部割込みのブロックダイアグラム	
.....	228
DTP/外部割込みのレジスタ.....	229
外部割込み要求	
外部割込み要求とDTP要求の切替え.....	234
外部割込み要求レベル	
外部割込み要求レベル.....	237
カウンタコントロールレジスタ	
カウンタコントロールレジスタ0, 1 下位	
(CCRLO, 1).....	215
カウンタコントロールレジスタ0上位 (CCRHO)	
.....	211
カウンタコントロールレジスタ1上位 (CCRHI)	
.....	213
カウンタステータスレジスタ	
カウンタステータスレジスタ0, 1 (CSR0, 1)	
.....	209
カウントクリア／ゲート機能	
カウントクリア／ゲート機能.....	224
カウントクロック	
8/16ビットPPGのカウントクロックの選択	
.....	196
カウント方向	
カウント方向フラグ, カウント方向転換フラグ	
.....	225
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 (EI20S)	
.....	293
拡張インテリジェントI/Oサービス (EI ² OS) の	
概要.....	58
拡張インテリジェントI/Oサービス (EI20S) の	
構造.....	59
拡張インテリジェントI/Oサービス (EI20S) の	
使用手順.....	68
拡張インテリジェントI/Oサービス (EI20S) の	
処理時間 (1回の転送時間)	69
拡張インテリジェントI/Oサービス (EI20S) の処	
理手順.....	67
拡張インテリジェントI/Oサービス (EI ² OS) の	
動作.....	59
拡張インテリジェントI/Oサービスディスクリプタ	
拡張インテリジェントI/Oサービスディスクリ	
プタ (ISD).....	63
間接アドレッシング	
間接アドレッシング.....	405

き

擬似時計モード	
擬似時計モードの解除.....	95
擬似時計モードへの遷移.....	95
キャリーフラグ	
キャリーフラグ (C).....	32

く

クロック	
クロック選択の状態遷移.....	88
クロック供給マップ	
クロック供給マップ.....	75
クロックコントロールレジスタ	
クロックコントロールレジスタ (ICCR)	321
クロック出力許可レジスタ	
クロック出力許可レジスタ (CLKR)	337
CKSCR	
クロック選択レジスタ (CKSCR)	86
クロック選択レジスタ	
クロック選択レジスタ (CKSCR)	86
クロック発生部	
クロック発生部に関する注意事項.....	74
クロックモニタ	
クロックモニタ機能のブロックダイアグラム	
.....	336

こ

コマンドシーケンス	
コマンドシーケンス表.....	358
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)	
.....	40
コンディションコードレジスタ	
コンディションコードレジスタ (CCR)	31
コントロールステータスレジスタ	
コントロールステータスレジスタ (ADCS0, 1)	
.....	247
コンペア機能	
8/16ビットアップダウンカウンタ／タイマの	
リロード機能とコンペア機能	
.....	220, 222
コンペア検出フラグ	
コンペア検出フラグ.....	225

し

システムスタックバンクレジスタ	
システムスタックバンクレジスタ (SSB) ...	24
システムスタックポインタ	
ユーザスタックポインタ (USP) とシステムス	
タックポインタ (SSP)	30

実効アドレス	
実効アドレスフィールド.....	414
実効アドレスフィールド	
実効アドレスフィールド.....	397
実行サイクル数	
実行サイクル数.....	412
実行サイクル数計算方法.....	412
自動レディ機能選択レジスタ	
自動レディ機能選択レジスタ (ARSR).....	115
シフト動作	
シフト動作のスタート/ストップタイミング	
.....	307
受信動作	
受信動作.....	285
出力端子レジスタ	
出力端子レジスタ (ODR).....	133
出力端子レジスタ (ODR) のブロックダイヤグラ	
ム.....	133
初期状態	
初期状態.....	345
シリアル	
MB90F574/Aシリアル書込み接続の基本構成	
.....	378
シリアル書込み接続例 (ユーザ電源使用時)	
.....	381
シリアル書込み接続例 (ライタから電源供給	
時).....	383
シリアルアウトデータレジスタ	
シリアルインプットデータレジスタ (SIDR)/シ	
リアルアウトデータレジスタ	
(SODR).....	276
シリアルインプットデータレジスタ	
シリアルインプットデータレジスタ (SIDR)/シ	
リアルアウトデータレジスタ	
(SODR).....	276
シリアルクロック入力周波数	
発振クロック周波数とシリアルクロック入力	
周波数.....	380
シリアルコントロールレジスタ	
シリアルコントロールレジスタ (SCR).....	273
シリアルシフトデータレジスタ (SDR)	
シリアルシフトデータレジスタ (SDR).....	302
シリアルステータスレジスタ	
シリアルステータスレジスタ (SSR).....	277
シリアルデータ	
シリアルデータの入出力のタイミング....	309
シリアルデータレジスタR/W待機状態	
シリアルデータレジスタR/W待機状態.....	305
シリアルモードコントロールステータスレジスタ	
シリアルモードコントロールステータスレジ	
スタ (SMCS).....	298
シリアルモードレジスタ	
シリアルモードレジスタ (SMR).....	271

す

推奨設定例	
推奨設定例.....	112
スタートコンディション	
SDA=L, SCL=L時におけるスタートコンディショ	
ン発生命令の実行.....	326
スタートコンディション.....	325
スタックフラグ	
スタックフラグ (S).....	31
スタンバイ	
スタンバイからの復帰.....	236
スティッキビットフラグ	
スティッキビットフラグ (T).....	31
ストップコンディション	
ストップコンディション.....	325
ストップモード	
ストップモードの解除.....	97
ストップモードへの遷移.....	97
スリープモード	
スリープモードの解除.....	94
スリープモードへの遷移.....	94

せ

セクタ消去タイマフラグ	
セクタ消去タイマフラグ (DQ3).....	363
セクタ消去タイマフラグの状態遷移.....	363
ゼロフラグ	
ゼロフラグ (Z).....	31
専用ポーレートジェネレータ	
専用ポーレートジェネレータ.....	282
専用レジスタ	
専用レジスタ.....	26

そ

送信動作	
送信動作.....	285
ソフトウェア割込み	
ソフトウェア割込みに関する注意事項.....	57
ソフトウェア割込みの概要.....	56
ソフトウェア割込みの構造.....	56
ソフトウェア割込みの動作.....	56

た

タイマカウンタコントロールステータスレジスタ	
タイマカウンタコントロールステータスレジ	
スタ (TCCS).....	163
タイマカウンタデータレジスタ	
タイマカウンタデータレジスタ (TCDT).....	162
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).....	362

タイミングリミット超過フラグの状態遷移	362
タイムベースタイマ	
タイムベースタイマの動作.....	141
タイムベースタイマのブロックダイアグラム	138
タイムベースタイマのレジスタ一覧.....	138
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)	139
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).....	35
多重割込み	
多重割込み.....	50
多バイト長データ	
多バイト長データのアクセス.....	25
メモリ空間における多バイト長データの配置	25
端子	
端子機能説明.....	10
端子配列図.....	6
単発モード	
単発モード.....	253
単発モード時のEI20Sの起動例.....	256
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	241
遅延割込み発生モジュールの動作.....	241
遅延割込み発生モジュールのブロックダイア	240
グラム.....	
遅延割込み発生モジュールのレジスタ....	240
遅延割込み要求ラッチ	
遅延割込み要求ラッチ.....	241
チップセレクト	
チップセレクト機能の概要.....	330
チップセレクト機能のデコードアドレス空間	333
一覧.....	
チップセレクト機能の動作.....	332
チップセレクト機能のブロックダイアグラム	330
チップセレクトコントロールレジスタ	
チップセレクトコントロールレジスタ (CSCRO	331
～6).....	
注意	
「DIV A, Ri」, 「DIVW A, RWi」 命令の使用上の	43
注意.....	
「DIV A, Ri」, 「DIVW A, RWi」 命令の注意事項	44
の回避について.....	
SCC, MSS, INT ビットの競合についての注意	320
.....	
デバイスの取り扱いに関する注意事項.....	17
直接アドレッシング	
直接アドレッシング.....	398

つ

通信開始	
CLK同期モードの通信開始.....	287
通信終了	
CLK同期モードの通信終了.....	287
通信プリスケアラレジスタ	
通信プリスケアラレジスタ (CDCR).....	280
通信プリスケアラレジスタの設定.....	281

て

停止状態	
停止状態.....	305
停止モード	
停止モード.....	254
停止モード時のEI20S起動例.....	258
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	82
.....	
低消費電力制御回路のレジスタ.....	83
低消費電力制御回路	
低消費電力制御回路の動作モード.....	80
低消費電力モード	
低消費電力モード.....	92
低消費電力モード動作状態.....	93
低消費電力モード状態遷移図.....	103
低消費電力モードの遷移条件.....	100
低消費電力モード制御レジスタ	
低消費電力モード制御レジスタ (LPMCR)	84
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) へのア	85
クセス.....	
データカウンタ	
データカウンタ (DCT).....	64
データバンクレジスタ	
データバンクレジスタ (DTB).....	24
データポーリングフラグ	
データポーリングフラグ (DQ7).....	360
データポーリングフラグの状態遷移.....	360
データレジスタ	
データレジスタ (ADCR1, ADCR0).....	251
データレジスタ (IDAR).....	324
デバイスの取り扱い	
デバイスの取り扱いに関する注意事項.....	17
転送状態	
転送状態.....	305
転送データフォーマット	
CLK同期モードの転送データフォーマット.....	286
転送データフォーマット.....	285

と

動作モード	
動作モード.....	108
特長	
特長	2
トグルビットフラグ	
トグルビットフラグ (DQ6).....	361
トグルビットフラグの状態遷移.....	361
時計カウンタ	
時計カウンタ.....	153
時計タイマ	
時計タイマのインターバル割込み機能....	153
時計タイマのブロックダイアグラム.....	150
時計タイマのレジスタ.....	150
時計タイマ制御レジスタ	
時計タイマ制御レジスタ (WTC).....	151
時計モード	
時計モードの解除.....	96
時計モードへの遷移.....	96

な

内部シフトクロックモード	
内部シフトクロックモード.....	304
内部タイマ	
内部タイマ.....	283

に

入出力回路	
入出力回路形式.....	14
入力抵抗レジスタ	
入力抵抗レジスタ (RDR).....	134
入力抵抗レジスタ (RDR) のブロックダイアグラム.....	134

ね

ネガティブフラグ	
ネガティブフラグ (N).....	31

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.....	359
ハードウェアスタンバイモード	
ハードウェアスタンバイモードの解除.....	99
ハードウェアスタンバイモードへの遷移... ..	99
ハードウェア割込み	
ハードウェア割込み使用手順.....	55
ハードウェア割込み処理時間.....	53
ハードウェア割込みの概要.....	49
ハードウェア割込みの構造.....	49
ハードウェア割込みの使用上の注意.....	51

ハードウェア割込みの動作.....	52
ハードウェア割込みの動作フロー.....	54
ハードウェア割込み要求	
内蔵リソース領域への書き込み中のハードウェア割込み要求.....	50
バスエラー	
バスエラー.....	326
バスコントロールレジスタ	
バスコントロールレジスタ (IBCR)	318
バスステータスレジスタ	
バスステータスレジスタ (IBSR).....	315
バス制御信号選択レジスタ	
バス制御信号選択レジスタ (ECSR).....	118
バスモード	
バスモード.....	108
バスモード別メモリ空間.....	111
発振クロック周波数	
発振クロック周波数とシリアルクロック入力周波数.....	380
バッファアドレスポインタ	
バッファアドレスポインタ (BAP).....	66
パルス	
8/16ビットPPGのパルスの端子出力の制御.....	197
パルス幅	
リロード値とパルス幅の関係.....	195
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.....	39
バンク方式	
バンク方式によるアドレス指定.....	23
バンクレジスタ	
バンクレジスタ.....	36
汎用レジスタ	
汎用レジスタ.....	37

ひ

品種	
品種構成.....	4

ふ

フラグ変化抑止プリフィックス	
フラグ変化抑止プリフィックス (NCC).....	40
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 (ユーザ電源使用时)	385
フラッシュマイコンプログラマとの最小限の接続例 (ライタから電源供給時) ...	387
フラッシュメモリ	
2Mビットフラッシュメモリのセクタ構成.....	355
2Mビットフラッシュメモリの特長.....	354

フラッシュメモリ書込み/消去の詳細説明	364
.....	364
フラッシュメモリ書込み/消去の方法.....	354
フラッシュメモリの書込み手順.....	366
フラッシュメモリのセクタ消去手順.....	369
フラッシュメモリのセクタ消去の一時停止	
.....	371
フラッシュメモリのセクタ消去の再開.....	372
フラッシュメモリの全データ消去(チップ消去).....	368
フラッシュメモリの任意データ消去(セクタ消去).....	369
フラッシュメモリのプログラム例.....	373
フラッシュメモリの読出し/リセット状態	
.....	365
フラッシュメモリのレジスタ.....	354
フラッシュメモリへのデータ書込み.....	366
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).....	356
プリフィックスコード	
プリフィックスコード.....	39
プリフィックスコードと割込み/ホールド抑止命令.....	41
プリフィックスコードの連続.....	42
プログラム	
プログラムパッチ処理例とフロー.....	347
プログラムミスが発生した場合.....	345
プログラムアドレス検出コントロールレジスタ	
プログラムアドレス検出コントロールレジスタ (PACSR).....	343
プログラムアドレス検出レジスタ	
プログラムアドレス検出レジスタ (PADR0, 1).....	342
プログラムカウンタ	
プログラムカウンタ (PC).....	34
プログラムバンクレジスタ	
プログラムバンクレジスタ (PCB).....	23
プロセッサステータス	
プロセッサステータス (PS).....	31
ブロックダイヤグラム	
16ビット入出力タイマのブロックダイヤグラム.....	158
16ビットフリーランタイマのブロックダイヤグラム.....	161
8/16ビットアップダウンカウンタ/タイマのブロックダイヤグラム.....	204
I/O拡張シリアルインタフェースのブロックダイヤグラム.....	296
アウトプットコンペアのブロックダイヤグラム.....	166
インプットキャプチャの全体ブロックダイヤグラム.....	171
外部メモリアクセスのブロックダイヤグラム	
.....	114
クロックモニタ機能のブロックダイヤグラム	
.....	338

出力端子レジスタ (ODR) のブロックダイアグラム	133
タイムベースタイマのブロックダイアグラム	138
チップセレクト機能のブロックダイアグラム	330
低消費電力制御回路のブロックダイアグラム	82
入力抵抗レジスタ (RDR) のブロックダイアグラム	134
ブロックダイアグラム	5

 \wedge

ペリフェラル

DTPを用いた時の外部に接続するペリフェラル
の条件..... 236

変換データ保護機能

変換データ保護機能.....260

ほ

ポートデータレジスタ

ポートデータレジスタ (PDR)..... 130

ポート方向レジスタ

ポート方向レジスタ (DDR)..... 132

ホールド機能

ホールド機能..... 125

ま

マシクロック

マシクロックの切換え..... 80

み

未定義命令

未定義命令の実行による例外発生..... 71

ゆ

命令

命令の種類	395
命令マップ ^o の構造	433

命令一覽表

F ² MC-16LX命令一覽表	419
-----------------------------------	-----

メインクロック

メインクロック発振安定待ち時間設定..... 80

メモリアクセスモード

メモリアクセスモード..... 108

メモリ空間

バスモード別メモリ空間..... 111
メモリ空間..... 22

メモリ空間における多バイト長データの配置	25
----------------------	----

も

モード端子	
モード端子	109
モードデータ	
モードデータ	110

ゆ

ユーザスタックバンクレジスタ	
ユーザスタックバンクレジスタ (USB)	24
ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP)	
ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP)	30

よ

要求レベル設定レジスタ	
要求レベル設定レジスタ (ELVR)	232

ら

ライトタイミング	
8/16ビットPPGのリロードレジスタへのライトタイミング	198

り

リセット	
フラッシュメモリの読出し/リセット状態	365
リセット解除後の動作	78
リセットシーケンス	
リセットシーケンス	345
リセット要因	
リセット要因	76
リニア方式	
リニア方式によるアドレス指定	23
リロード/コンペアレジスタ	
リロード/コンペアレジスタ0, 1 (RCR0, 1)	208
リロード機能	
8/16ビットアップダウンカウンタ/タイマのリロード機能とコンペア機能	220, 222
リロード値	
リロード値とパルス幅の関係	195
リロードレジスタ	
リロードレジスタ (PRL/PLRH)	192

れ

例外発生

未定義命令の実行による例外発生	71
-----------------	----

レジスタ

8/16ビットアップダウンカウンタ/タイマのレジスタ一覧	206
I ² Cインタフェースのレジスタ	314
ROMミラー機能選択レジスタ (ROMM)	351
アディショナルバンクレジスタ (ADB)	24
アドレス一致検出機能のレジスタ	341
アドレスレジスタ (IADR)	323
クロックコントロールレジスタ (ICCR)	321
クロック出力許可レジスタ (CLKR)	337
システムスタックバンクレジスタ (SSB)	24
専用レジスタ	26
チップセレクトコントロールレジスタ (CSCRO ~6)	331
低消費電力制御回路のレジスタ	83
データバンクレジスタ (DTB)	24
データレジスタ (IDAR)	324
バスコントロールレジスタ (IBCR)	318
バスステータスレジスタ (IBSR)	315
バンクレジスタ	36
汎用レジスタ	37
フラッシュメモリコントロールステータスレジスタ (FMCS)	356
フラッシュメモリのレジスタ	354
プログラムアドレス検出コントロールレジスタ (PACSR)	343
プログラムアドレス検出レジスタ (PADR0, 1)	342
プログラムバンクレジスタ (PCB)	23
ユーザスタックバンクレジスタ (USB)	24

レジスタバンク

レジスタバンク	37
---------	----

レジスタバンクポインタ

レジスタバンクポインタ (RP)	32
------------------	----

レディ機能

レディ機能	123
-------	-----

連続モード

連続モード	253
連続モード時のEI20Sの起動例	257

わ

割込み

8/16ビットPPGの割込み	199
I/O拡張シリアルインタフェースの割込み機能	310
INT9割込み	346
UARTの割込みとフラグのセットタイミング	289
割込み時におけるスタック内へのレジスタの回避	50
割込みの概要	46

割り込み／ホールド抑止命令

プリフィックスコードと割り込み／ホールド抑
止命令..... 41

割り込み許可フラグ

割り込み許可フラグ (I)..... 31

割り込み制御レジスタ

割り込み制御レジスタ (ICR)..... 60

割り込みハンドリング時間

割り込みハンドリング時間..... 53

割り込みベクタ

割り込みベクタ..... 48

割り込み要因

UARTの割り込み要因..... 288

割り込み要因..... 47

割り込み要求サンプル待ち時間

割り込み要求サンプル待ち時間..... 53

割り込み抑止命令

割り込み抑止命令..... 50

CM44-10102-8

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

MB90570series

ハードウェアマニュアル

2008年7月 第8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 2 - 7 *