



本ドキュメントはCypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、仕様の開発元企業として「スパンション」, 「Spansion」, 「富士通」または「Fujitsu」の名が記載されておりますが、これらの製品は Cypress が新規および既存のお客様に引き続き提供してまいります。

#### 商品仕様の継続性について

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

#### オーダ型格および品名について

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

#### 詳しいお問い合わせ先

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

#### サイプレスについて

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

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

16ビット・マイクロコントローラ

**MB90580Cseries**

ハードウェアマニュアル



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

16ビット・マイクロコントローラ

**MB90580Cseries**

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。

MB90580Cシリーズは、ASIC(Application Specific IC)対応が可能なオリジナル16ビット・ワンチップマイクロコントローラであるF<sup>2</sup>MC<sup>®</sup>-16LXシリーズの汎用品の1つとして開発された製品です。

本書は、実際にこの半導体を使って製品を設計する技術者の方を対象に、MB90580Cシリーズの機能や動作について記載しています。本書をご一読ください。

## ■ 商標

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

## ■ 本書の全体構成

本書は、以下に示す26の章と付録から構成されています。

### 第1章 概要

この章では、MB90580C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、F<sup>2</sup>MC-16LXシリーズのCPUの内部構成、およびMB90580Cシリーズに内蔵されているハードウェアの仕様について説明します。

### 第3章 割込み

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の機能と動作について説明します。

### 第6章 メモリアクセスモード

この章では、F<sup>2</sup>MC<sup>®</sup>-16LXシリーズの内部メモリアクセスモードと外部メモリアクセスについて説明します。

### 第7章 I/Oポート

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8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9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10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11章 PWCタイマ

この章では、PWC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 16ビットリロードタイマ（イベントカウント機能付）

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

## 第14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15章 DTP/外部割込み

この章では、DTP/外部割込みの機能と動作および使用上の注意について説明します。

## 第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、および、使用上の注意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、および、使用上の注意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、および、使用上の注意について説明します。

## 第19章 通信プリスケアラ

この章では、通信プリスケアラの機能と動作について説明します。

## 第20章 UART

この章では、UARTの機能と動作、使用上の注意、および、応用例について説明します。

## 第21章 IEBus™コントローラ

この章では、IEBus™コントローラの機能と動作について説明します。

## 第22章 クロックモニタ機能

この章では、クロックモニタ機能について説明します。

## 第23章 アドレス一致検出機能

この章では、アドレス一致検出機能とその動作について説明します。

## 第24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25章 1Mビットフラッシュメモリ

この章では、1Mビットフラッシュメモリの機能と動作について説明します。

## 第26章 MB90F583C/CAシリアル書き込み接続例

この章では、MB90F583C/CAのシリアル書き込み接続例について説明します。

## 付録

付録では、I/Oマップ、命令について示します。

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

Copyright ©2004-2008 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	5
1.3	MB90580Cシリーズのブロックダイアグラム	6
1.4	パッケージ外形寸法図	7
1.5	端子配列図	9
1.6	端子機能説明	11
1.7	入出力回路形式	16
1.8	デバイスの取扱いに関する注意事項	19
第2章	CPU	23
2.1	メモリ空間	24
2.2	アドレス指定	25
2.2.1	メモリ空間における多バイト長データの配置	27
2.3	専用レジスタ	28
2.3.1	アキュムレータ(A)	30
2.3.2	ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)	32
2.3.3	プロセッサステータス(PS)	33
2.3.4	プログラムカウンタ(PC)	35
2.3.5	ダイレクトページレジスタ(DPR)	36
2.3.6	バンクレジスタ	37
2.4	汎用レジスタ	38
2.5	プリフィックスコード	40
2.6	割込み抑止命令とプリフィックスコード	42
2.7	「DIV A,Ri」,「DIW A,RWi」命令の使用上の注意	44
第3章	割込み	47
3.1	割込みの概要	48
3.2	割込み要因	49
3.3	割込みベクタ	51
3.4	ハードウェア割込み	53
3.4.1	ハードウェア割込みの動作	55

3.4.2	ハードウェア割込みの動作フロー .....	57
3.4.3	ハードウェア割込み使用手順例 .....	58
3.5	ソフトウェア割込み .....	59
3.6	拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) .....	61
3.6.1	割込み制御レジスタ(ICR) .....	63
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) .....	66
3.6.3	拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) の動作 .....	69
3.6.4	拡張インテリジェントI/Oサービス(EI <sup>2</sup> OS)の実行時間 .....	71
3.7	未定義命令の実行による例外発生 .....	72
<b>第4章</b>	<b>クロックとリセット .....</b>	<b>73</b>
4.1	クロック発生部 .....	74
4.2	リセット要因 .....	75
4.3	リセット解除後の動作 .....	77
<b>第5章</b>	<b>低消費電力制御回路 .....</b>	<b>81</b>
5.1	低消費電力制御回路の概要 .....	82
5.2	低消費電力モード制御レジスタ(LPMC <sub>R</sub> ) .....	84
5.3	クロック選択レジスタ(CKSCR) .....	86
5.4	低消費電力制御回路の動作 .....	89
5.4.1	スリープモード .....	91
5.4.2	擬似時計モード .....	92
5.4.3	時計モード .....	93
5.4.4	ストップモード .....	94
5.4.5	ハードウェアスタンバイモード .....	95
5.5	CPU間欠動作機能 .....	96
5.6	メインクロック発振安定待ち時間設定 .....	97
5.7	マシクロックの切り替え .....	98
5.8	状態遷移 .....	101
5.9	低消費電力モードの状態遷移図 .....	104
<b>第6章</b>	<b>メモリアクセスモード .....</b>	<b>113</b>
6.1	メモリアクセスモードの概要 .....	114
6.1.1	モード端子 .....	115
6.1.2	モードデータ .....	116

6.1.3	バスモード別メモリ空間 .....	117
6.2	外部メモリアクセス（外部バス端子制御回路） .....	119
6.2.1	外部メモリアクセス（外部バス端子制御回路）のレジスタ .....	120
6.2.2	自動レディ機能選択レジスタ (ARSR) .....	121
6.2.3	外部アドレス出力制御レジスタ (HACR) .....	123
6.2.4	バス制御信号選択レジスタ (ECSR) .....	124
6.3	外部メモリアクセス制御信号の動作 .....	127
6.3.1	レディ機能 .....	129
6.3.2	ホールド機能 .....	131
<b>第7章</b>	<b>I/Oポート .....</b>	<b>133</b>
7.1	I/Oポートの概要 .....	134
7.2	I/Oポートブロックダイアグラム .....	135
7.3	I/Oポートのレジスタ .....	137
7.3.1	ポートデータレジスタ (PDRx) .....	139
7.3.2	ポート方向レジスタ (DDRx) .....	141
7.3.3	ポート4出力端子レジスタ (ODR4) .....	143
7.3.4	入力プルアップ抵抗設定レジスタ (RDR0,1,6) .....	144
7.3.5	ポート5アナログ入力許可レジスタ (ADER) .....	145
<b>第8章</b>	<b>タイムベースタイマ .....</b>	<b>147</b>
8.1	タイムベースタイマの概要 .....	148
8.2	タイムベースタイマ制御レジスタ (TBTC) .....	150
8.3	タイムベースタイマの動作 .....	152
<b>第9章</b>	<b>ウォッチドッグタイマ .....</b>	<b>153</b>
9.1	ウォッチドッグタイマの概要 .....	154
9.2	ウォッチドッグタイマ制御レジスタ (WDTC) .....	155
9.3	ウォッチドッグタイマの動作 .....	157
<b>第10章</b>	<b>時計タイマ .....</b>	<b>159</b>
10.1	時計タイマの概要 .....	160
10.2	時計タイマ制御レジスタ (WTC) .....	162
10.3	時計タイマの動作 .....	164



第11章	PWCタイマ	165
11.1	PWCタイマの概要	166
11.2	PWCタイマのブロックダイアグラム	167
11.3	PWCタイマのレジスタ	168
11.3.1	PWCコントロールステータスレジスタ(PWCSR)	169
11.3.2	PWCデータバッファレジスタ(PWCR)	174
11.3.3	分周比コントロールレジスタ(DIVR)	175
11.3.4	PWCノイズフィルタレジスタ(RNCR)	176
11.4	PWCタイマの動作	177
11.4.1	カウントクロックの選択	179
11.4.2	動作モードの選択	180
11.4.3	タイマ/パルス幅測定の開始と停止, タイマのクリア	181
11.5	タイマモードの動作の詳細	183
11.6	タイマモード動作のフローチャート	185
11.7	パルス幅測定モード動作の詳細	186
11.7.1	測定モードと測定動作	189
11.7.2	パルス幅測定動作のフローチャート	191
11.8	PWCタイマ取扱い上の注意事項	192
第12章	16ビット入出力タイマ	195
12.1	16ビット入出力タイマの概要	196
12.2	16ビット入出力タイマのブロックダイアグラム	198
12.3	16ビット入出力タイマのレジスタ	199
12.3.1	16ビットフリーランタイマ	201
12.3.2	アウトプットコンペア	204
12.3.3	インプットキャプチャ	207
12.4	16ビットフリーランタイマの動作	209
12.5	16ビットアウトプットコンペアの動作	211
12.6	16ビットインプットキャプチャの動作	213
第13章	16ビットリロードタイマ ( イベントカウント機能付 )	215
13.1	16ビットリロードタイマ ( イベントカウント機能付 ) の概要	216
13.2	16ビットリロードタイマ ( イベントカウント機能付 ) のレジスタ	217
13.2.1	タイマコントロールステータスレジスタ(TMCSR)	218

13.2.2	16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR) .....	221
13.3	クロック動作 .....	222
13.4	アンダフロー動作 .....	223
13.5	入出力端子機能(内部クロックモード時) .....	224
13.6	カウンタの動作状態 .....	226
<b>第14章</b>	<b>8/16ビットPPG .....</b>	<b>227</b>
14.1	8/16ビットPPGの概要 .....	228
14.2	8/16ビットPPGのブロックダイアグラム.....	229
14.3	8/16ビットPPGのレジスタ .....	231
14.3.1	PPG0動作モード制御レジスタ(PPGC0) .....	232
14.3.2	PPG1動作モード制御レジスタ(PPGC1) .....	234
14.3.3	PPG0,1出力端子制御レジスタ(PPGOE) .....	236
14.3.4	リロードレジスタ(PRLL/PRLH) .....	237
14.4	8/16ビットPPGの動作 .....	238
14.4.1	8/16ビットPPGの動作モード .....	240
14.4.2	PPG出力動作 .....	241
14.4.3	カウントクロックの選択 .....	242
14.4.4	パルスの端子出力の制御 .....	243
14.4.5	リロードレジスタへのライトタイミング .....	244
<b>第15章</b>	<b>DTP/外部割込み .....</b>	<b>245</b>
15.1	DTP/外部割込みの概要 .....	246
15.2	DTP/外部割込みのレジスタ .....	247
15.3	DTP/外部割込みの動作 .....	249
15.4	DTP/外部割込みの使用上の注意 .....	251
<b>第16章</b>	<b>遅延割込み発生モジュール .....</b>	<b>253</b>
16.1	遅延割込み発生モジュールの概要 .....	254
16.2	遅延割込み発生モジュールの動作 .....	255
<b>第17章</b>	<b>A/Dコンバータ .....</b>	<b>257</b>
17.1	A/Dコンバータの概要 .....	258
17.2	A/Dコンバータのロックダイアグラム.....	259
17.3	A/Dコンバータのレジスタ .....	260

17.3.1	コントロール・ステータスレジスタ(ADCS1,2)	261
17.3.2	データレジスタ(ADCR1,2)	265
17.4	A/Dコンバータの動作	267
17.4.1	単発モード時のEI <sup>2</sup> OSの起動例	269
17.4.2	連続モード時のEI <sup>2</sup> OSの起動例	270
17.4.3	停止モード時のEI <sup>2</sup> OS起動例	271
17.5	変換データ保護機能	272
第18章	D/Aコンバータ	273
18.1	D/Aコンバータの概要	274
18.2	D/Aコンバータのレジスタ	276
18.3	D/Aコンバータの動作	278
第19章	通信プリスケアラ	279
19.1	通信プリスケアラの概要	280
19.2	通信プリスケアラの動作	282
第20章	UART	283
20.1	UARTの概要	284
20.2	UARTのブロックダイアグラム	285
20.3	UARTのレジスタ	286
20.3.1	シリアルモードレジスタ(SMR0~4)	287
20.3.2	シリアルコントロールレジスタ(SCR0~4)	289
20.3.3	シリアルインプットデータレジスタ(SIDR0~4)/シリアルアウトプットデータレジスタ(SODR0~4)	291
20.3.4	シリアルステータスレジスタ(SSR0~4)	292
20.4	UARTの動作	294
20.4.1	UARTのクロック選択	295
20.4.2	非同期(調歩同期)モード	297
20.4.3	CLK同期モード	299
20.4.4	割込み発生およびフラグのセットタイミング	301
20.5	UART(モード1使用時)の応用例	304
第21章	IEBus <sup>TM</sup> コントローラ	307
21.1	IEBus <sup>TM</sup> コントローラの概要	308

21.2	IEBus™コントローラのブロックダイアグラム	309
21.3	IEBus™コントローラのレジスタ	310
21.3.1	自局アドレス設定レジスタ(MAWH, MAWL)	312
21.3.2	スレーブアドレス設定レジスタ(SAWH, SAWL)	313
21.3.3	同報コントロールビット設定レジスタ(DCWR)	314
21.3.4	電文長ビット設定レジスタ(DEWR)	315
21.3.5	コマンドレジスタ上位8ビット(CMRH)	316
21.3.6	コマンドレジスタ下位8ビット(CMRL)	318
21.3.7	ステータスレジスタ上位8ビット(STRH)	320
21.3.8	ステータスレジスタ下位8ビット(STRL)	322
21.3.9	ロックリードレジスタ(LRRH, LRRL)	324
21.3.10	マスタアドレスリードレジスタ(MARH, MARL)	325
21.3.11	同報コントロールビットリードレジスタ(DCRR)	326
21.3.12	電文長ビットリードレジスタ下位8ビット(DERR)	327
21.3.13	リードデータバッファ(RDB)	328
21.3.14	ライトデータバッファ(WDB)	329
21.4	IEBus™の送信制御	330
21.5	IEBus™の受信制御	333
21.6	通信制御のステータス	335
21.7	IEBus™コントローラのメイン/割り込み処理フロー例	337
21.7.1	初期設定処理ルーチン	338
21.7.2	マスタ送信処理ルーチン	339
21.7.3	スレーブデータ送信処理ルーチン	340
21.7.4	マスタ受信処理ルーチン	341
21.8	IEBus™コントローラの送信時の動作	343
21.9	IEBus™プロトコル動作	345
21.10	伝送プロトコル	348
21.10.1	伝送プロトコルのヘッダ	349
21.10.2	伝送プロトコルのマスタアドレスフィールド	350
21.10.3	伝送プロトコルのスレーブアドレスフィールド	351
21.10.4	伝送プロトコルのコントロールフィールド	352
21.10.5	電文長フィールド	353
21.10.6	データフィールド	354
21.10.7	パリティビット	355

21.10.8	アクノリッジビット	356
21.11	伝送データ	358
21.12	ビットフォーマット	361
第22章	クロックモニタ機能	363
22.1	クロックモニタ機能の概要	364
22.2	クロック出力許可レジスタ(CCLKR)	365
第23章	アドレス一致検出機能	367
23.1	アドレス一致検出機能の概要	368
23.2	アドレス一致検出機能のレジスタ	369
23.3	アドレス一致検出機能の動作	371
23.4	アドレス一致検出機能の使用例	372
第24章	ROMミラー機能選択モジュール	375
24.1	ROMミラー機能選択モジュールの概要	376
24.2	ROMミラー機能選択レジスタ(ROMM)	377
第25章	1Mビットフラッシュメモリ	379
25.1	1Mビットフラッシュメモリの概要	380
25.2	フラッシュメモリのセクタ構成	381
25.3	フラッシュメモリコントロールステータスレジスタ(FMCS)	382
25.4	フラッシュメモリ自動アルゴリズム起動方法	384
25.5	自動アルゴリズム実行状態の確認	385
25.5.1	データポーリングフラグ(DQ7)	386
25.5.2	トグルビットフラグ(DQ6)	387
25.5.3	タイミングリミット超過フラグ(DQ5)	388
25.5.4	セクタ消去タイマフラグ(DQ3)	389
25.6	フラッシュメモリ書込み/消去の詳細説明	390
25.6.1	フラッシュメモリを読み出し/リセット状態にする	391
25.6.2	フラッシュメモリヘデータを書き込む	392
25.6.3	フラッシュメモリの全データを消去する(チップ消去)	394
25.6.4	フラッシュメモリの任意のデータを消去する(セクタ消去)	395
25.6.5	フラッシュメモリのセクタ消去を一時停止する	397
25.6.6	フラッシュメモリのセクタ消去を再開する	398

25.7	1Mビットフラッシュメモリのプログラム例.....	399
<b>第26章</b>	<b>MB90F583C/CAシリアル書き込み接続例.....</b>	<b>403</b>
26.1	MB90F583C/CAシリアル書き込み接続の基本構成.....	404
26.2	シリアル書き込み接続例（ユーザ電源使用时）.....	407
26.3	シリアル書き込み接続例（ライターから電源供給時）.....	409
26.4	フラッシュマイコンプログラマとの最小限の接続例（ユーザ電源使用时） .....	411
26.5	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） .....	413
<b>付録</b>	<b>.....</b>	<b>415</b>
A	I/Oマップ.....	416
B	命令.....	421
B.1	命令の種類.....	422
B.2	アドレッシング.....	423
B.3	直接アドレッシング.....	425
B.4	間接アドレッシング.....	432
B.5	実行サイクル数.....	439
B.6	実効アドレスフィールド.....	441
B.7	命令一覧表の読み方.....	443
B.8	F <sup>2</sup> MC-16LX命令一覧表.....	446
B.9	命令マップ.....	460
<b>索引</b>	<b>.....</b>	<b>483</b>



## 本版での主な変更内容

ページ	変更内容（詳細は本文を参照してください。）
421 ~ 481	「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、MB90580Cシリーズの品種構成、および品種の概要について説明します。

---

- 1.1 特長
- 1.2 品種構成
- 1.3 MB90580Cシリーズのブロックダイアグラム
- 1.4 パッケージ外形寸法図
- 1.5 端子配列図
- 1.6 端子機能説明
- 1.7 入出力回路形式
- 1.8 デバイスの取扱いに関する注意事項

## 1.1 特長

MB90580Cシリーズは、民生機器等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。命令体系は、F<sup>2</sup>MC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IEBus™コントローラを内蔵しており、機器間の通信が容易に実現できます。コンポーネント型のオーディオやVTRシステム等に適した製品です。

### MB90580Cシリーズの特長

#### 最小命令実行時間

62.5ns/4MHz原発振 4通倍(PLL クロック通倍方式)

#### 最大メモリ空間

- 16Mbyte
- リニア、バンク方式にてアクセス

#### コントローラ用途に最適化された命令体系

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

#### 高級言語(C言語)/マルチタスクに対応をとった命令体系

- システムスタックポインタの採用
- 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)

#### 実行速度の向上：4byteのキュー

#### 強力な割込み機能

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

#### CPUに依存しないデータ転送機能

- 最大 16chまでの拡張インテリジェントI/Oサービス
- DTP要求入力：8本

#### 内蔵ROM

- FLASH版：128Kbyte
- MASKROM版：128Kbyte(MB90583C/CA)，64Kbyte(MB90587C/CA)

#### 内蔵RAM

- FLASH版：6Kbyte
- MASKROM版：6Kbyte(MB90583C/CA)，4Kbyte(MB90587C/CA)

#### 汎用ポート

最大77本(うち入力プルアップ抵抗設定可能：22本/出力オープンドレイン設定可能：8本)

#### IEBus™コントローラ\*

異なる3つのデータ転送実行速度が選択可能

- モード0：3.9Kbps(16byte/frame)
- モード1：17.0Kbps(32byte/frame)
- モード2：26.0Kbps(128byte/frame)

\*：IEBus™コントローラは日本電気(株)の商標です。

#### A/Dコンバータ(RC 逐次比較方式)：8ch

- 分解能：8/10bit
- 変換時間：34.7μs(最小)，12MHz動作時

#### D/Aコンバータ：2ch

- 分解能：8bit
- 設定時間：12.5μs

#### UART：5ch

#### 8/16bit PPG：1ch

8bit×2ch，16bit×1ch モード切り替え機能付き

#### 16bit リロードタイマ：3ch

#### 16bit PWCタイマ：1ch

ノイズフィルタ搭載，パルス幅カウンタ用に使用可

#### 16bit 入出力タイマ

- インพุットキャプチャ：4ch
- アウトプットコンペア：2ch
- フリーランタイマ：1ch

#### クロックジェネレータ内蔵

#### タイムベースカウンタ/ウォッチドッグタイマ：18bit

#### クロックモニタ機能組込み

低消費電力モード

- スリープ
- ストップ
- ハードウェアスタンバイモード
- CPU間欠機能

パッケージ

- LQFP-100
- QFP-100

## 1.2 品種構成

MB90580Cシリーズの品種構成を、表 1.2-1に示します。ROM/RAM容量、クロック以外の機能は共通となっています。ただし、MB90587C/CAにIEBus™コントローラはありません。

### 品種構成

表 1.2-1 MB90580Cシリーズの品種構成

項目	MB90583C	MB90583CA	MB90587C	MB90587CA	MB90F583C	MB90F583CA	MB90V580B
ROM容量	マスクROM 128Kバイト	マスクROM 128Kバイト	マスクROM 64Kバイト	マスクROM 64Kバイト	FLASH ROM 128Kバイト	FLASH ROM 128Kバイト	-
RAM容量	6Kバイト	6Kバイト	4Kバイト	4Kバイト	6Kバイト	6Kバイト	6Kバイト
クロック	2系統	1系統	2系統	1系統	2系統	1系統	2系統
IEBus™ コントローラ	あり	あり	なし	なし	あり	あり	あり
エミュレータ 専用電源	-	-	-	-	-	-	無

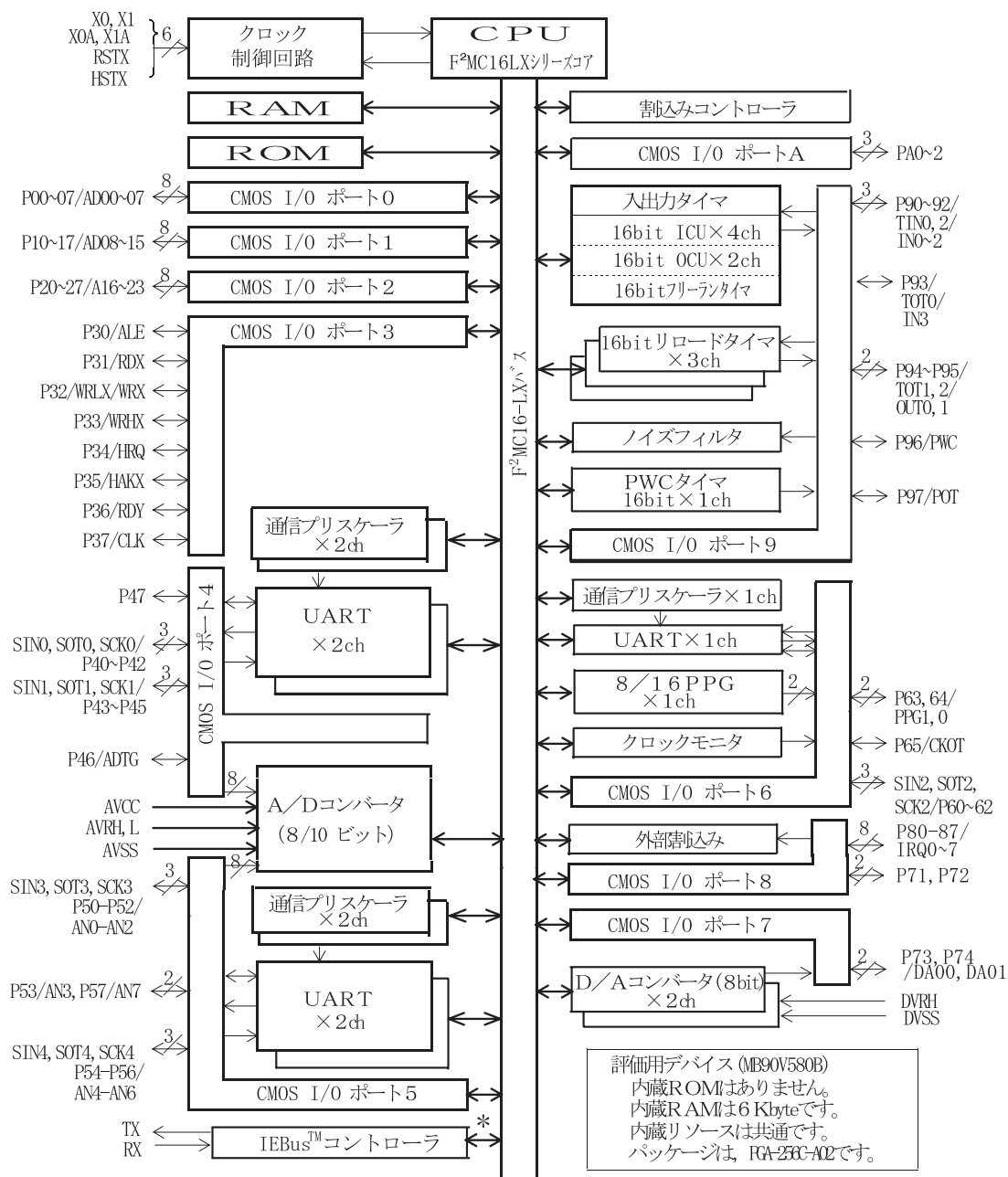
\* :エミュレーションボットMB2145-507をご使用頂く際のディップスイッチS2の設定です。詳細につきましては、MB2145-507ハードウェアマニュアル（2.7 エミュレータ専用電源端子）をご参照ください。

(注) 評価用デバイスにはMB90V580Bをご使用ください。なお、この時1系統で使用する場合はツール側よりX0A、X1Aにクロックを供給してください。

## 1.3 MB90580Cシリーズのブロックダイアグラム

図 1.3-1に、MB90580Cシリーズのブロックダイアグラムを示します。

MB90580Cシリーズのブロックダイアグラム



P00~07 (8ch) : 入力プルアップ抵抗として使用可能なレジスタ付  
 P10~17 (8ch) : 入力プルアップ抵抗として使用可能なレジスタ付  
 P60~65 (6ch) : 入力プルアップ抵抗として使用可能なレジスタ付  
 P40~47 (8ch) : オープンドレインとして使用可能なレジスタ付

\* : MB90587C/CAでは、IEBus™ コントローラはありません。  
 TX, RXは、N.C.になります。

その他の端子  
 MOD 2~0  
 C, VCC, VSS

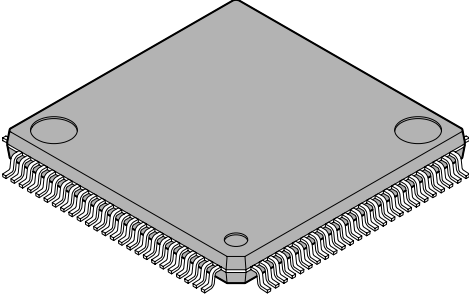
図 1.3-1 MB90580Cシリーズのブロックダイアグラム

## 1.4 パッケージ外形寸法図

図 1.4-1に、FPT-100P-M05(LQFP-100)のパッケージ外形寸法図を、図 1.4-2に、FPT-100P-M06(QFP-100)のパッケージ外形寸法図を示します。

なお、本外形寸法は参考用です。正式版につきましては別途ご相談ください。

FPT-100P-M05パッケージ外形寸法図

<p>プラスチック・LQFP, 100 ピン</p>  <p>(FPT-100P-M05)</p>	リードピッチ	0.50mm
	パッケージ幅 × パッケージ長さ	14.0 × 14.0mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	1.70mm MAX
	質量	0.65g
	コード (参考)	P-LFQFP100-14×14-0.50

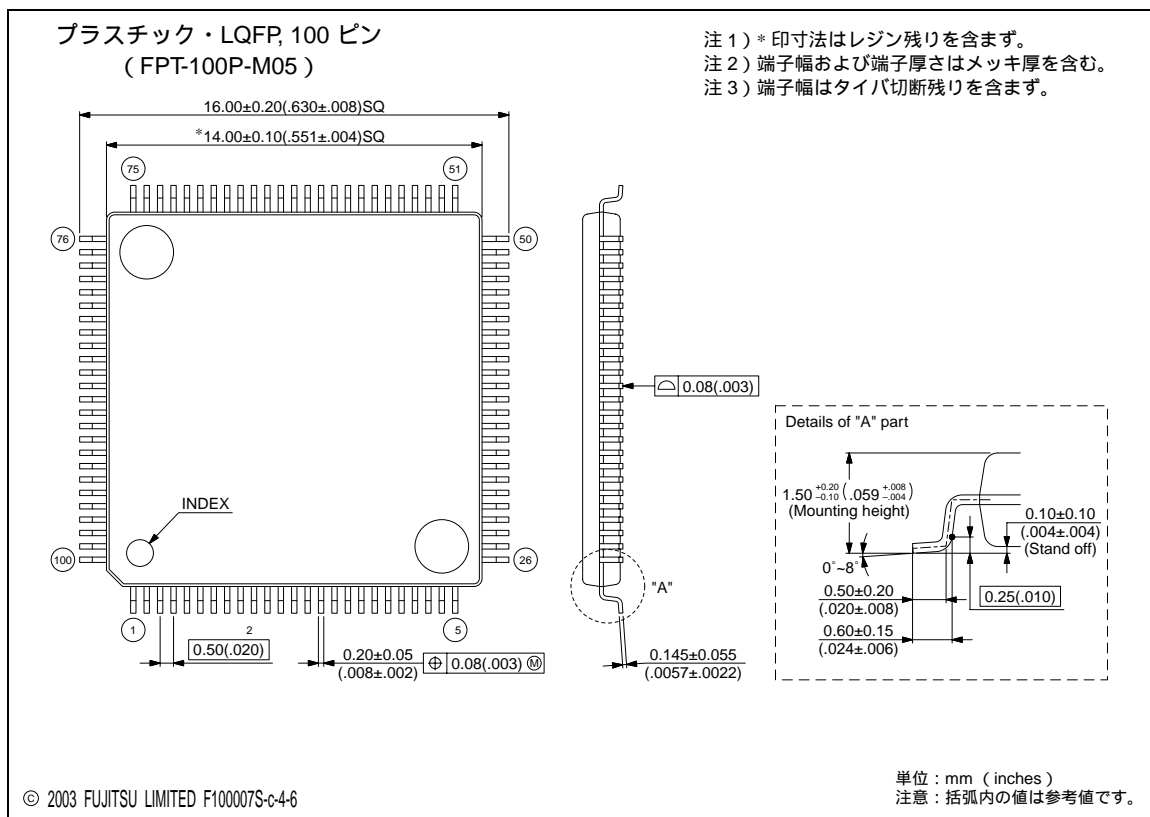
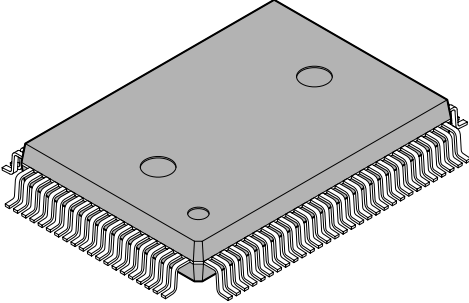


図 1.4-1 FPT-100P-M05(LQFP-100)のパッケージ外形寸法図



FPT-100P-M06パッケージ外形寸法図

<p>プラスチック・QFP, 100 ピン</p>  <p>(FPT-100P-M06)</p>	リードピッチ	0.65mm
	パッケージ幅× パッケージ長さ	14.00 × 20.00mm
	リード形状	ガルウィング
	封止方法	プラスチックモールド
	取付け高さ	3.35mm MAX
	コード (参考)	P-QFP100-14×20-0.65

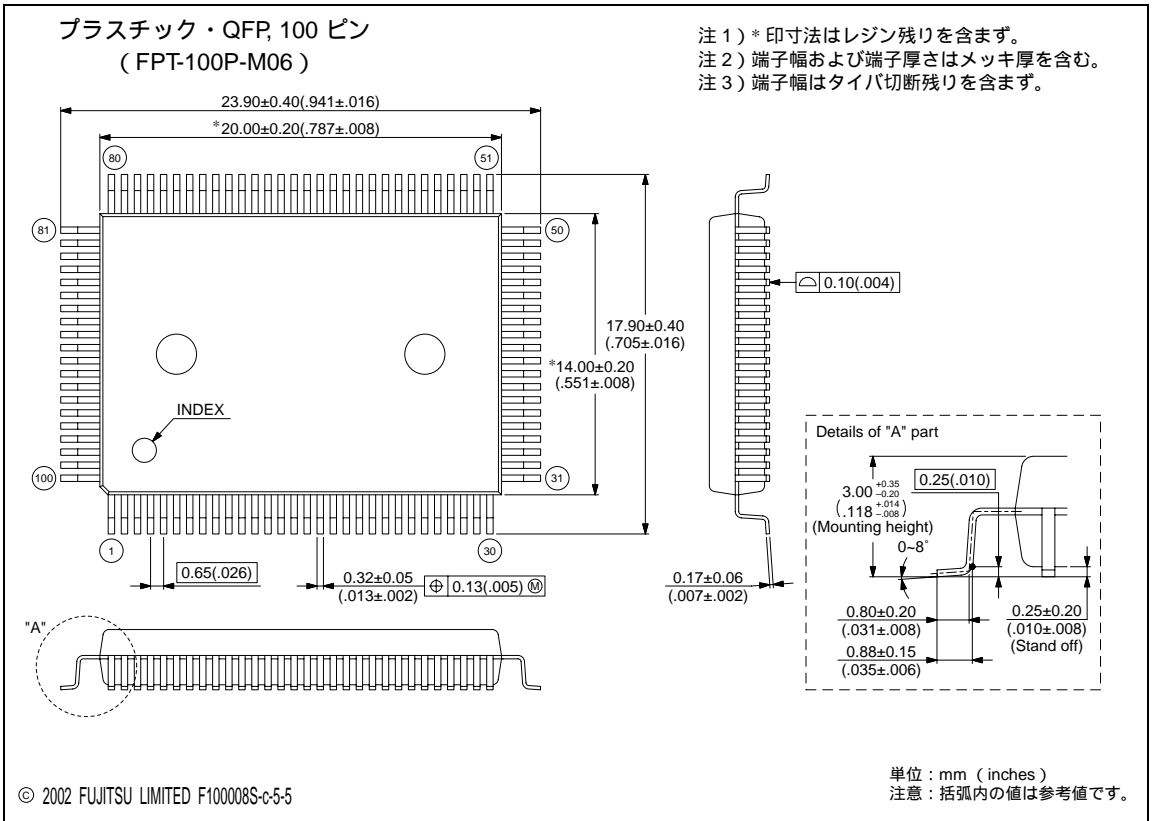
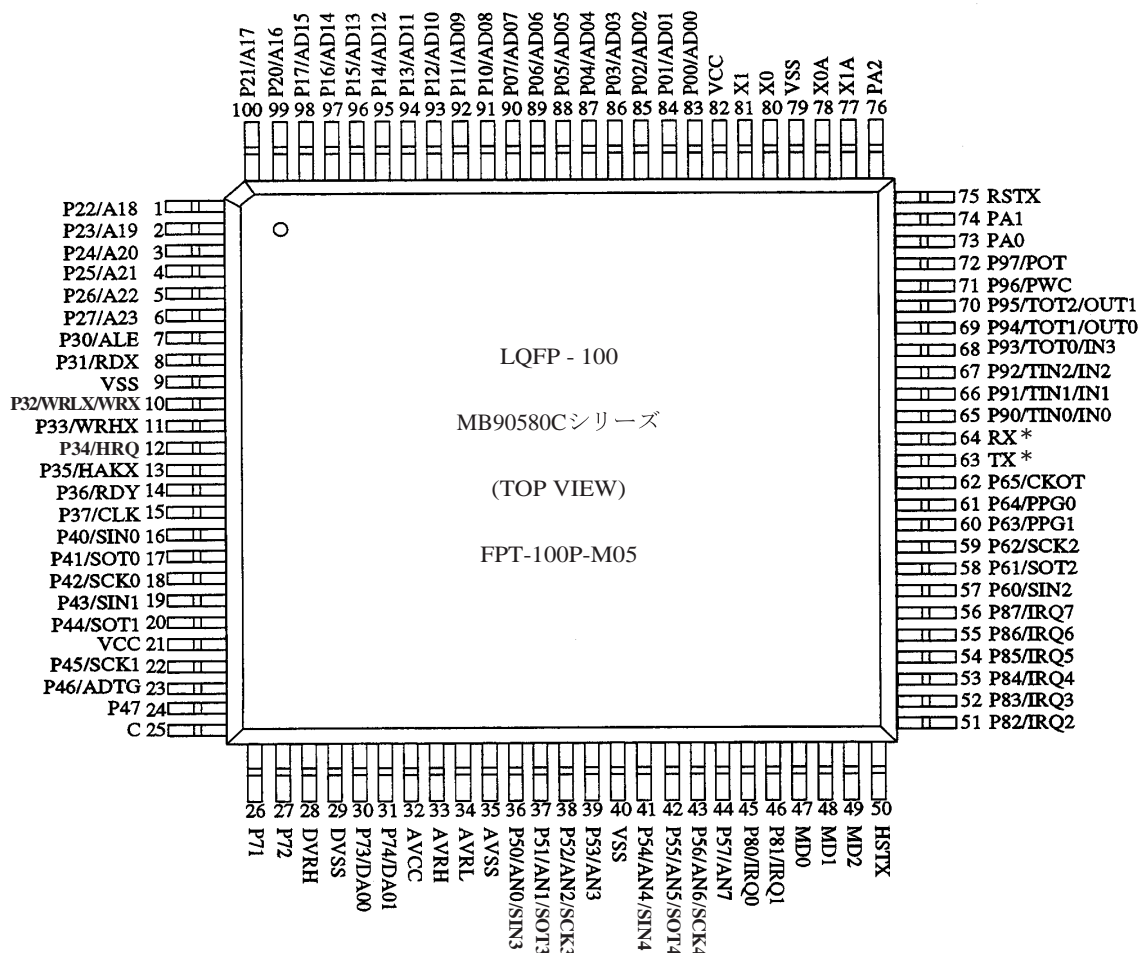


図 1.4-2 FPT-100P-M06(QFP-100)のパッケージ外形寸法図

## 1.5 端子配列図

図 1.5-1に ,FTP-100P-M05の端子配列図を図 1.5-2に ,FTP-100P-M06の端子配列図を示します。

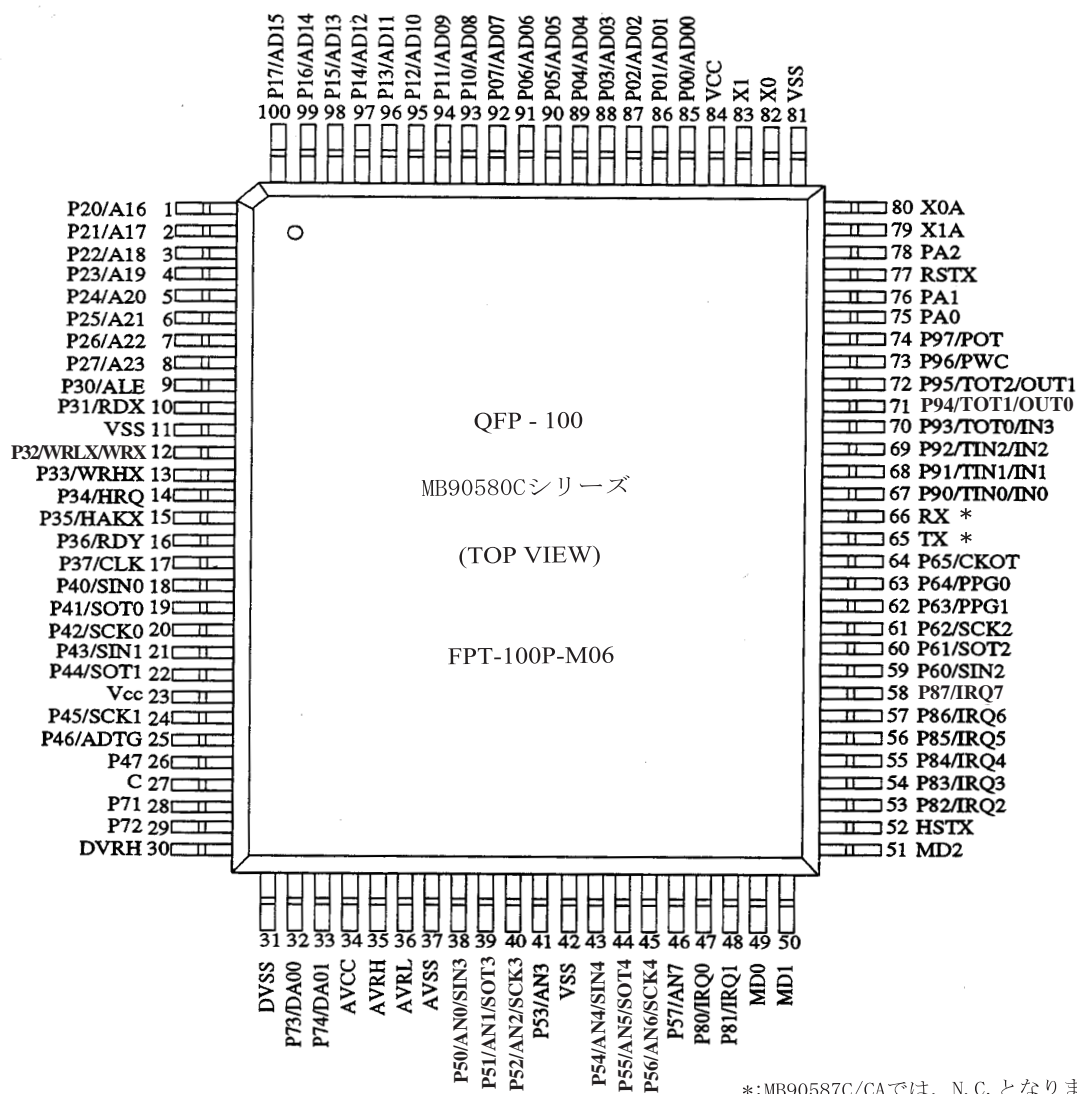
FTP-100P-M05端子配列図



\*:MB90587C/CAでは、N.C.となります。

図 1.5-1 FTP-100P-M05端子配列図

FPT-100P-M06端子配列図



\*:MB90587C/CAでは、N.C.となります。

図 1.5-2 FPT-100P-M06の端子配列図

## 1.6 端子機能説明

表 1.6-1に、MB90580Cシリーズの端子機能を示します。

表 1.6-1の回路形式欄のアルファベットは、図 1.7-1の分類欄のアルファベットに対応しています。

### 端子機能説明

表 1.6-1 端子機能説明 (続く)

QFP	LQFP	端子名	回路形式	機能説明
82	80	X0	A	発振用端子。
83	81	X1	A	発振用端子。
52	50	HSTX	C	ハードウェアスタンバイ入力端子。
77	75	RSTX	B	リセット入力端子。
85 ~ 92	83 ~ 90	P00 ~ P07	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR0)により、プルアップ抵抗の割り当て(RD07 ~ RD00="1")ができます。[出力設定時(DDR0レジスタ: D07 ~ D00="1")は無効]
		AD00 ~ AD07		外バスモード時のデータ下位入出力/アドレス下位出力(AD00 ~ 07)として機能。
93 ~ 100	91 ~ 98	P10 ~ P17	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR1)により、プルアップ抵抗の割り当て(RD17 ~ RD10="1")ができます。[出力設定時(DDR1レジスタ: D17 ~ D10="1")は無効]
		AD08 ~ AD15		バス幅が16bit外バスモード時に、データ上位入出力/アドレス中位出力(AD08 ~ 15)として機能。
1 ~ 8	99 ~ 6	P20 ~ P27	F (CMOS/H)	汎用入出力ポート。 外バスモード時、HACRレジスタの対応するビットが"0"のときにA16 ~ A23端子として機能。
		A16 ~ A23		外バスモード時、HACRレジスタの対応するビットが"0"のときにアドレス上位出力(A16 ~ A23)として機能。
9	7	P30	F (CMOS/H)	汎用入出力ポート。 外バスモード時、ALE端子として機能。
		ALE		外バスモード時、アドレス取り込み許可信号(ALE)端子として機能。
10	8	P31	F (CMOS/H)	汎用入出力ポート。 外バスモード時、RDX端子として機能。
		RDX		外バスモード時、リードストローブ出力(RDX)端子として機能。
12	10	P32	F (CMOS/H)	汎用入出力ポート。 外バスモード時、WREビットが"1"のときにWRLX端子として機能。
		WRLX		外バスモード時、下位側データライトストローブ出力(WRLX/WRX)端子として機能。
		WRX		
13	11	P33	F (CMOS/H)	汎用入出力ポート。 バス幅が16bit外バスモード時、EPCRレジスタのWREビットが"1"のときにWRHX端子として機能。
		WRHX		外バスモード時、上位側データライトストローブ出力(WRHX/WRX)端子として機能。
14	12	P34	F (CMOS/H)	汎用入出力ポート。 外バスモード時、EPCRレジスタのHDEビットが"1"のときにHRQ端子として機能。
		HRQ		外バスモード時、ホールド要求入力(HRQ)端子として機能。

表 1.6-1 端子機能説明 (続く)

QFP	LQFP	端子名	回路形式	機能説明
15	13	P35	F (CMOS/H)	汎用入出力ポート。 外バスモード時, EPCRレジスタのHDEビットが"1"のときにHAKX端子として機能。
		HAKX		外バスモード時, ホールドアクノリッジ出力(HAKX)端子として機能。
16	14	P36	F (CMOS/H)	汎用入出力ポート。 外バスモード時, EPCRレジスタのRYEビットが"1"のときに外部レディ入力(RDY)端子として機能。
		RDY		外バスモード時, 外部レディ入力(RDY)端子として機能。
17	15	P37	F (CMOS/H)	汎用入出力ポート。 外バスモード時, EPCRレジスタのCKEビットが"1"のときにCLK端子として機能。
		CLK		外バスモード時, マシンサイクルクロック出力(CLK)端子として機能。
18	16	P40	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ(ODR4)のOD40が"1"に設定されているとき, オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D40="0")は無効]。
		SINO		UART0のシリアルデータ入力(SINO)端子。 UART0が入力動作をしている間はこの入力を随時使用していますので, 意図的に行う以外は他の機能による出力を停止させておく必要があります。
19	17	P41	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ(ODR4)のOD41が"1"に設定されているとき, オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D41="0")は無効]。
		SOTO		UART0のシリアルデータ出力(SOTO)端子。 この機能は, UART0のシリアルデータ出力指定が許可のときに有効になります。
20	18	P42	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ(ODR4)のOD42が"1"に設定されているとき, オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D42="0")は無効]。
		SCK0		UART0のシリアルクロック入出力(SCK0)端子。 この機能は, UART0のクロック出力指定が許可のときに有効になります。
21	19	P43	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ(ODR4)のOD43が"1"に設定されているとき, オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D43="0")は無効]。
		SIN1		UART1のシリアルデータ入力(SIN1)端子。 UART1が入力動作をしている間はこの入力を随時使用していますので, 意図的に行う以外は他の機能による出力を停止させておく必要があります。
22	20	P44	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ(ODR4)のOD44が"1"に設定されているとき, オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D44="0")は無効]。
		SOT1		UART1のシリアルデータ出力(SOT1)端子。 この機能は, UART1のシリアルデータ出力指定が許可のときに有効になります。

表 1.6-1 端子機能説明 (続く)

QFP	LQFP	端子名	回路形式	機能説明
24	22	P45	E (CMOS/H)	汎用入出力ポートです。 オープンドレイン制御設定レジスタ (ODR4) のOD45が"1" に設定されているとき、オープンドレイン出力ポートとして機能[入力設定時(DDR4レジスタ: D45="0")は無効]。
		SCK1		UART1のシリアルクロック入出力(SCK1)端子。 この機能は、UART1のクロック出力指定が許可のときに有効になります。
25	23	P46	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ (ODR4) のOD46が"1" に設定されているとき、オープンドレイン出力ポートとして機能。[入力設定時(DDR4レジスタ: D46="0")は無効]
		ADTG		A/Dコンバータの外部トリガ入力(ADTG)端子。
26	24	P47	E (CMOS/H)	汎用入出力ポート。 オープンドレイン制御設定レジスタ (ODR4) のOD47が"1" に設定されているとき、オープンドレイン出力ポートとして機能。[入力設定時(DDR4レジスタ: D47="0")は無効]
38	36	P50	G (CMOS/H)	汎用入出力ポート。
		AN0		A/Dコンバータ動作中のアナログ入力(AN0)端子。
		SIN3		UART3のシリアルデータ入力(SIN3)端子です。 UART3が入力動作をしている間はこの入力を随時使用していますので、意図的に行う以外は他の機能による出力を停止させておく必要があります。
39	37	P51	G (CMOS/H)	汎用入出力ポート。
		AN1		A/Dコンバータ動作中のアナログ入力(AN1)端子。
		SOT3		UART3のシリアルデータ出力(SOT3)端子。 この機能は、UART3のシリアルデータ出力指定が許可のときに有効になります。
40	38	P52	G (CMOS/H)	汎用入出力ポート
		AN2		A/Dコンバータ動作中のアナログ入力(AN2)端子。
		SCK3		UART3のシリアルクロック入出力(SCK3)端子。 この機能は、UART3のクロック出力指定が許可のときに有効になります。
41	39	P53	G (CMOS/H)	汎用入出力ポート。
		AN3		A/Dコンバータ動作中のアナログ入力(AN3)端子。
43	41	P54	G (CMOS/H)	汎用入出力ポート。
		AN4		A/Dコンバータ動作中のアナログ入力(AN4)端子。
		SIN4		UART4のシリアルデータ入力(SIN4)端子。 UART4が入力動作をしている間はこの入力を随時使用していますので、意図的に行う以外は他の機能による出力を停止させておく必要があります。
44	42	P55	G (CMOS/H)	汎用入出力ポート。
		AN5		A/Dコンバータ動作中のアナログ入力(AN5)端子。
		SOT4		UART4のシリアルデータ出力(SOT4)端子。 この機能は、UART4のシリアルデータ出力指定が許可のときに有効になります。
45	43	P56	G (CMOS/H)	汎用入出力ポート。
		AN6		A/Dコンバータ動作中のアナログ入力(AN6)端子。
		SCK4		UART4のシリアルクロック出力(SCK4)端子。 この機能は、UART4のクロック出力指定が許可のときに有効になります。
46	44	P57	G (CMOS/H)	汎用入出力ポート。
		AN7		A/Dコンバータ動作中のアナログ入力(AN7)端子。
27	25	C	—	電圧供給安定化のための、0.1μFコンデンサ結合端子

表 1.6-1 端子機能説明 (続く)

QFP	LQFP	端子名	回路形式	機能説明
28	26	P71	F (CMOS/H)	汎用入出力ポート。
29	27	P72	F (CMOS/H)	汎用入出力ポート。
32	30	P73	H (CMOS/H)	汎用入出力ポート。 D/Aコントロールレジスタ(DACR)のDAE0ビットが"1"のとき、D/A 出力端子(DA00)として機能。
		DA00		D/Aコンバータ出力0(DA00)端子。
33	31	P74	H (CMOS/H)	汎用入出力ポート。 D/Aコントロールレジスタ(DACR)のDAE1ビットが"1"のとき、D/A 出力端子(DA01)として機能。
		DA01		D/Aコンバータ出力1(DA01)端子。
47	45	P80	F (CMOS/H)	汎用入出力ポート。
		IRQ0		外部割込み要求入力0(IRQ0)端子。
48	46	P81	F (CMOS/H)	汎用入出力ポート。
		IRQ1		外部割込み要求入力1(IRQ1)端子。
53	51	P82	F (CMOS/H)	汎用入出力ポート。
		IRQ2		外部割込み要求入力2(IRQ2)端子。
54	52	P83	F (CMOS/H)	汎用入出力ポート。
		IRQ3		外部割込み要求入力3(IRQ3)端子。
55	53	P84	F (CMOS/H)	汎用入出力ポート。
		IRQ4		外部割込み要求入力4(IRQ4)端子。
56	54	P85	F (CMOS/H)	汎用入出力ポート。
		IRQ5		外部割込み要求入力5(IRQ5)端子。
57	55	P86	F (CMOS/H)	汎用入出力ポート。
		IRQ6		外部割込み要求入力6(IRQ6)端子。
58	56	P87	F (CMOS/H)	汎用入出力ポート。
		IRQ7		外部割込み要求入力7(IRQ7)端子。
59	57	P60	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割 り当て(RD60="1")ができます[出力設定時(DDR6レジスタ： D60="1")は無効]。
		SIN2		UART2のシリアルデータ入力(SIN2)端子。 UART2が入力動作をしている間はこの入力を随時使用しています ので、意図的に行う以外は他の機能による出力を停止させておく 必要があります。
60	58	P61	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割 り当て(RD61="1")ができます[出力設定時(DDR6レジスタ： D61="1")は無効]。
		SOT2		UART2のシリアルデータ出力(SOT2)端子。 この機能は、UART2のシリアルデータ出力指定が許可のときに有 効になります。
61	59	P62	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割 り当て(RD62="1")ができます[出力設定時(DDR6レジスタ： D62="1")は無効]。
		SCK2		UART2のシリアルクロック出力(SCK2)端子。 この機能は、UART2のクロック出力指定が許可のときに有効にな ります。

表 1.6-1 端子機能説明 (続き)

QFP	LQFP	端子名	回路形式	機能説明
62	60	P63	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割り当て (RD63="1") ができます [出力設定時 (DDR6 レジスタ : D63="1") は無効]。
		PPG1		PPGの有効時、PPG1出力として機能。
63	61	P64	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割り当て (RD64="1") ができます [出力設定時 (DDR6 レジスタ : D64="1") は無効]。
		PPG0		PPGの有効時、PPG0出力として機能。
64	62	P65	D (CMOS/H)	汎用入出力ポート。 プルアップ抵抗設定レジスタ(RDR6)により、プルアップ抵抗の割り当て (RD65="1") ができます [出力設定時 (DDR6 レジスタ : D65="1") は無効]。
		CKOT		CKOT動作時、CKOT出力として機能。
65	63	TX *	I	IEBus™出力として機能。
66	64	RX *	J (CMOS)	IEBus™入力として機能。
67 ~ 69	65 ~ 67	P90 ~ P92	F (CMOS/H)	汎用入出力ポート。
		TIN0 ~ TIN2		リロードタイマ 0, 1, 2 用のイベント入力端子。リロードタイマ入力動作時、これらの入力は継続して使用されるため、これらの端子に対する他の機能からの出力は、意図して行わない限り避ける必要があります。
		IN0 ~ IN2		インプットキャプチャ ch0 ~ 2 トリガ入力。
70	68	P93	F (CMOS/H)	汎用入出力ポート。
		TOT0		リロードタイマ用の出力端子。リロードタイマ 0 用出力の有効時に、この機能は適用されます。
		IN3		インプットキャプチャ ch3 トリガ入力。
71, 72	69, 70	P94, P95	F (CMOS/H)	汎用入出力ポート。
		TOT1, TOT2		リロードタイマ用の出力端子。リロードタイマ 1, 2 用出力の有効時に、この機能は適用されます。
		OUT0, OUT1		アウトプットコンペア ch0, ch1 イベント出力。
73	71	P96	F (CMOS/H)	汎用入出力ポート。
		PWC		PWCタイマ有効時、PWC入力として機能。
74	72	P97	F (CMOS/H)	汎用入出力ポート。
		POT		PWCタイマ有効時、PWC出力として機能。
75, 76	73, 74	PA0, PA1	F (CMOS/H)	汎用入出力ポート。
78	76	PA2	F (CMOS/H)	汎用入出力ポート。
79	77	X1A	A	発振用入力。1系統品はオープンにしてください。
80	78	X0A	A	発振用入力。1系統品は外部でプルダウンの処理をしてください。
34	32	AVcc	——	A/Dコンバータの電源端子。
37	35	AVss	——	A/Dコンバータの電源端子。
35	33	AVRH	——	A/Dコンバータの外部基準電源端子。
36	34	AVRL	——	A/Dコンバータの外部基準電源端子。
30	28	DVRH	——	D/Aコンバータの外部基準電源端子。
31	29	DVSS	——	D/Aコンバータの電源端子。
49 ~ 51	47 ~ 49	MD0 ~ MD2	C	動作モード指定用入力端子。 VccまたはVssに直結します。
23, 84	21, 82	Vcc	——	電源(5V)入力端子。
11, 42, 81	9, 40, 79	Vss	——	電源(0V)入力端子。

\* : MB90587C/CAでは、N.C. となります。



1.7 入出力回路形式

図 1.7-1に，入出力回路形式を示します。図 1.7-1の分類欄のアルファベットは，表 1.6-1の回路形式欄のアルファベットに対応しています。

入出力回路形式

分類	回路	備考
A		<ul style="list-style-type: none"><li>・高速用発振帰還抵抗：約1MΩ</li><li>・低速用発振帰還抵抗：約10MΩ</li></ul>
B		<ul style="list-style-type: none"><li>・プルアップ付きヒステリシス入力</li><li>・プルアップ抵抗：約50kΩ</li></ul>
C		<ul style="list-style-type: none"><li>・ヒステリシス入力</li></ul>
D		<ul style="list-style-type: none"><li>・入力プルアップ抵抗制御付き</li><li>・CMOSレベル出力</li><li>・スタンバイ制御付きヒステリシス入力</li><li>・プルアップ抵抗：約50kΩ</li></ul>

図 1.7-1 入出力回路形式 (続く)

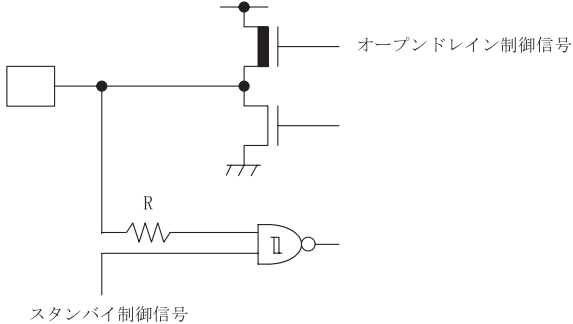
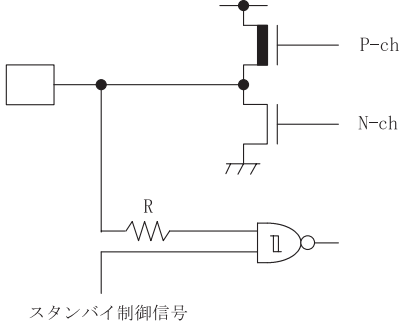
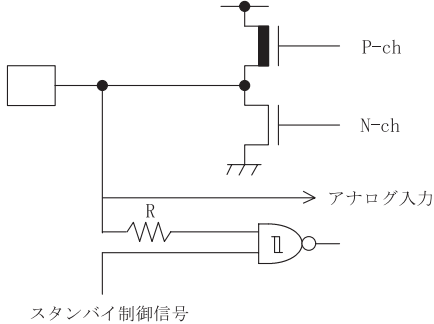
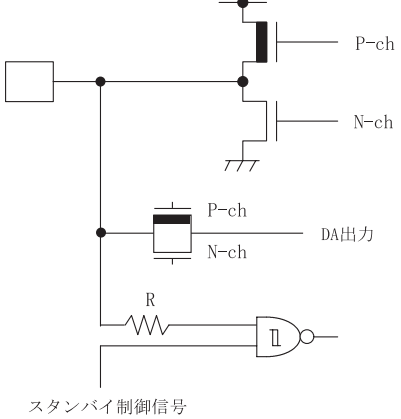
分類	回路	備考
E		<ul style="list-style-type: none"><li>• CMOSレベル出力</li><li>• スタンバイ制御付きヒステリシス入力</li><li>• オープンドレイン制御付き</li></ul>
F		<ul style="list-style-type: none"><li>• CMOSレベル出力</li><li>• スタンバイ制御付きヒステリシス入力</li></ul>
G		<ul style="list-style-type: none"><li>• CMOSレベル出力</li><li>• スタンバイ制御付きヒステリシス入力</li><li>• アナログ入力</li></ul>
H		<ul style="list-style-type: none"><li>• CMOSレベル出力</li><li>• スタンバイ制御付きヒステリシス入力</li><li>• DA出力</li></ul>

図 1.7-1 入出力回路形式 (続く)

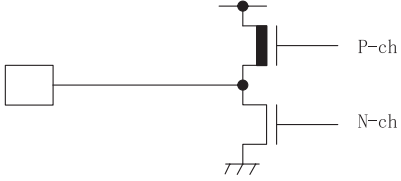
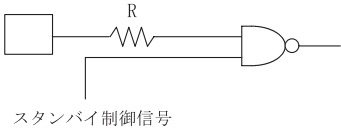
分類	回路	備考
I		・ CMOSレベル出力
J		・ スタンバイ制御付きCMOS入力

図 1.7-1 入出力回路形式 (続き)

## 1.8 デバイスの取扱いに関する注意事項

デバイスを取り扱う際には、以下の点について特に注意が必要です。

- ・ ラッチアップの防止
- ・ 未使用入力端子の処理
- ・ 外部クロック使用
- ・ 電源端子(Vcc/Vss)の取扱い

### デバイスの取扱いに関する注意事項

#### ラッチアップの防止

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

- 入力端子や出力端子に、Vccより高い電圧やVssより低い電圧が印加された場合。
- Vcc～Vss間に定格電圧を超える電圧が印加された場合。
- Vccの電圧より、AVccの電源が先に供給された場合。

ラッチアップが起きると電源電流が激増し、デバイスの熱破壊を引き起こすことがあります。

#### 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ておくと誤動作およびラッチアップによる永久破壊の原因になることがありますので、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については出力状態にして開放とするか、入力状態の場合は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の処理

A/Dコンバータを使用しない場合は、AVcc=Vss、AVss=AVRH=AVRL=Vssとなるように接続してください。

#### D/Aコンバータの電源端子の処理

D/Aコンバータを使用しない場合は、DVRH=Vss、DVss=Vssとなるように接続してください。

#### IEBus™未使用時のTX,RX端子の処理

IEBusを使用しない場合は、TX端子にはプルダウン抵抗を接続するか、開放としてください。RX端子にはプルダウン/プルアップ抵抗を接続してください。

#### 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、デバイス内部で同電位にすべきもの同士を接続してありますが、不要輻射の低減、グラウンドレベルの上昇によるストローブ信号の誤動作防止、総出力電源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に接続してください。

#### REAROSを使用する場合

REAROSを使用する場合は、拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)は使用できません。

### 電源立ち上げ

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げ時間は、 $50\mu\text{s}$  ( $0.2\text{V} \sim 2.7\text{V}$ の間) 以上を確保してください。

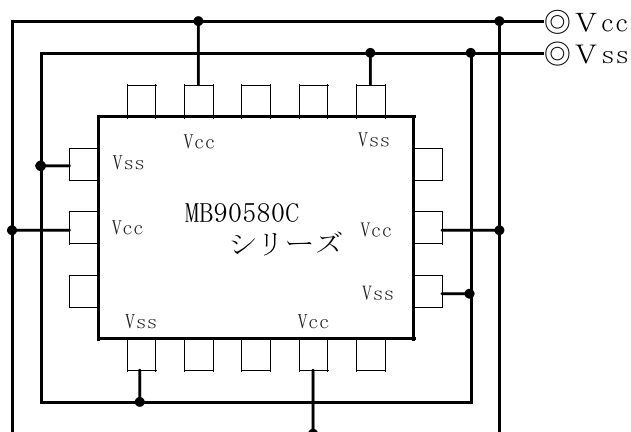


図 1.8-1 電源端子(Vcc/Vss)の取扱い

## サブクロックモードと外部クロックの使用

サブクロックを使用しない場合は1系統品をご使用ください。2系統品は、必ずX0A, X1A端子に32kHz以下の発振器を接続してください。

また、1系統品は、X0A端子にプルダウンの処理をし、X1A端子はオープンにしてください。

外部クロックを使用する際には、X0端子のみ駆動し、X1端子はオープンとしてください(図 1.8-2参照)。

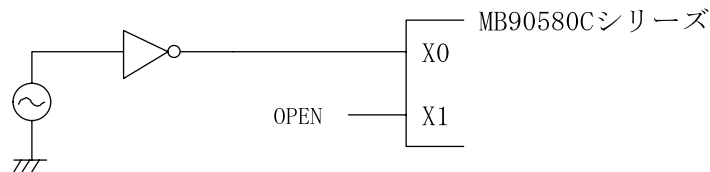


図 1.8-2 外部クロック使用方法

## ポート0, 1からの出力が不定になる場合

電源立ち上げ後、パワーオンリセット中にポート0, 1からの出力が不定になります。タイミングは、以下のようになりますので、ご注意ください。

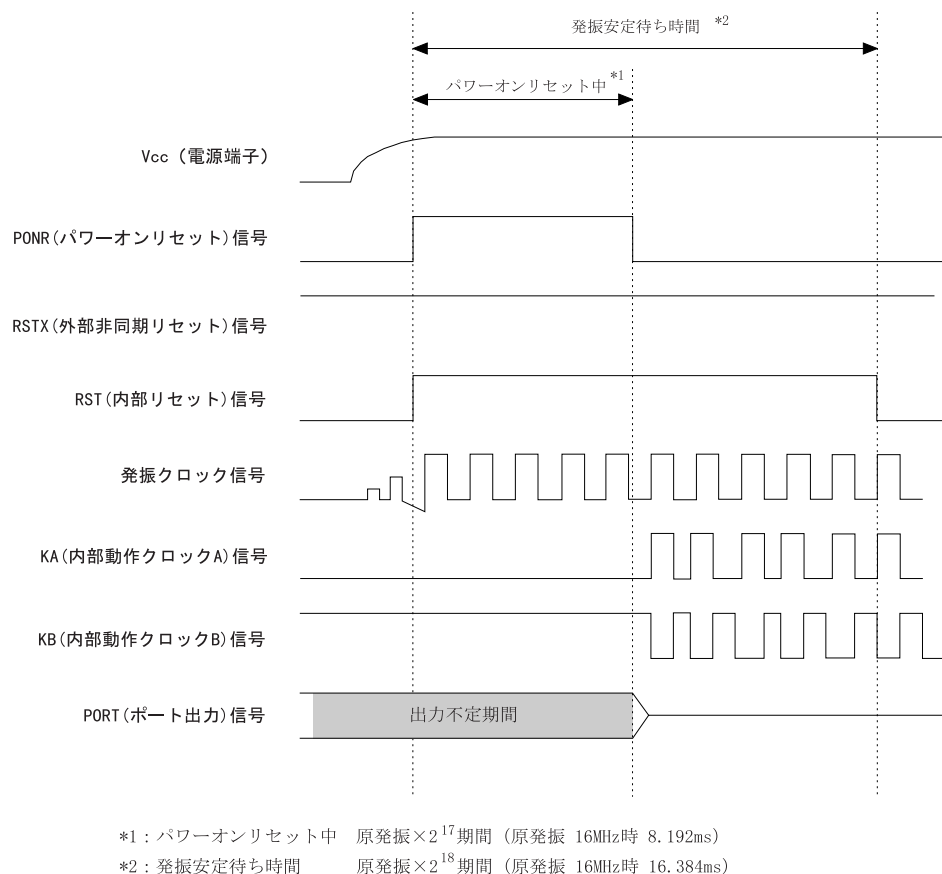


図 1.8-3 ポート0, 1からの不定出力

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

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

詳細は「2.7「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意」を参照してください。また、バンクレジスタについては「2.3.6バンクレジスタ」を参照してください。

#### PLLクロックモード動作中の注意について

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

#### 供給電圧の安定化

Vcc電源電圧の動作保証内においても、電源電圧の急激な変化があると誤動作を起こす場合がありますので、Vcc電源電圧を安定させてください。

安定化の基準としては、商用周波数(50 Hz ~ 60 Hz)でのVccリップル変動(peak to peak 値)は、標準Vcc電源電圧値の10%以下に、また電源の切り替え時などの瞬時変化においては、過渡変動率が0.1 V/ms以下になるように電源電圧を抑えてください。

## 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 専用レジスタ
- 2.4 汎用レジスタ
- 2.5 プリフィックスコード
- 2.6 割込み抑止命令とプリフィックスコード
- 2.7 「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意



## 2.1 メモリ空間

F<sup>2</sup>MC-16LX CPUコアは、民生用・車載用機器等の高速リアルタイム処理が要求される用途向けに設計された16bit CPUです。F<sup>2</sup>MC-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。

F<sup>2</sup>MC-16LXは、16ビットデータ処理のほか、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F<sup>2</sup>MC-8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・ビット処理の充実により命令が強化されています。

### メモリ空間

F<sup>2</sup>MC-16LX CPUが管理するデータ・プログラムI/Oは、すべてF<sup>2</sup>MC-16LX CPUが持つ16Mbyteのメモリ空間に配置されます。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、各リソースにアクセスすることができます。

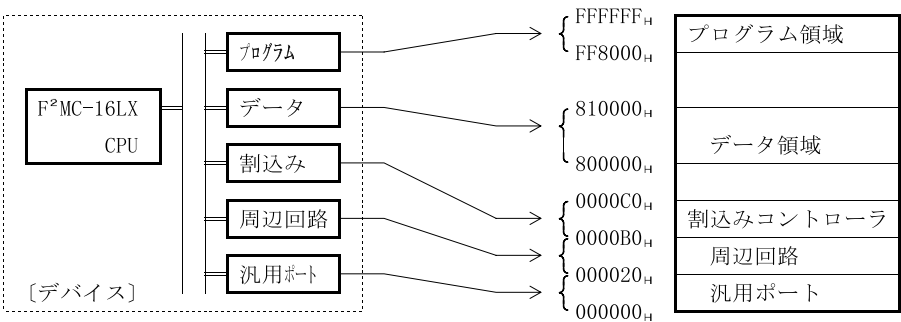


図 2.1-1 F<sup>2</sup>MC-16LX システムとメモリマップの関係例

## 2.2 アドレス指定

F<sup>2</sup>MC-16LXのアドレス指定には、以下に示す2つの方式があります。

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

### リニア方式によるアドレス指定

リニア方式には以下に示す2つの方式があります。

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

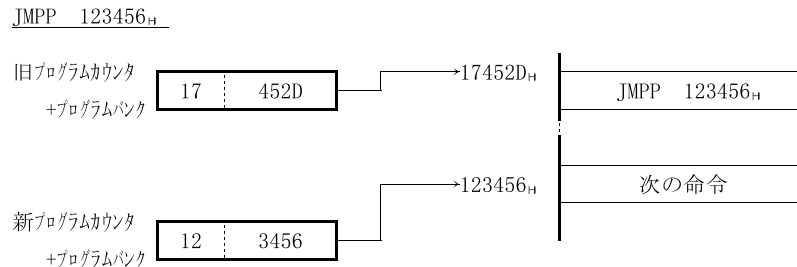


図 2.2-1 リニア方式の24ビットオペランド指定の例

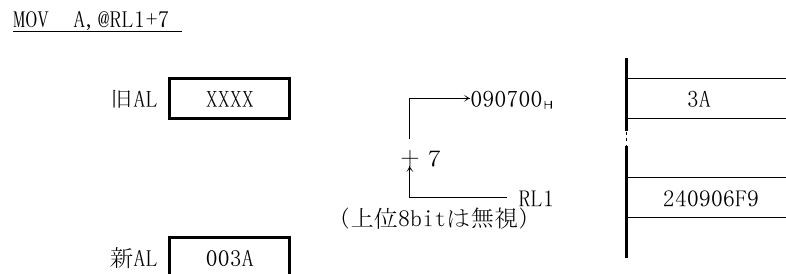


図 2.2-2 リニア方式の32ビットレジスタ間接指定の例

### バンク方式によるアドレス指定

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

プログラムバンクレジスタ(PCB) [ リセット時初期値："FF<sub>H</sub>" ]

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："00<sub>H</sub>" ]

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: "00<sub>H</sub>" ] / システムスタックバンクレジスタ (SSB) [ リセット時初期値: "00<sub>H</sub>" ]

USBあるいはSSBによって指定される64Kbyteのバンクをスタック(SP)空間とよび、プッシュ/ポップ命令や割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: "00<sub>H</sub>" ]

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。

命令のコード効率を向上するために、各命令は表 2.2-1に示すようなアドレッシングごとのデフォルト空間が決められています。あるアドレッシングモードで、デフォルト以外の空間を使用するには、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します。これによって、そのプリフィックスコードに対応した任意のバンク空間をアクセスすることができます。

DTB, USB, SSB, ADBはリセットにより"00<sub>H</sub>"に初期化され、PCBはリセットベクタにより指定された値に初期化されます。リセット後、DT, SP, ADの各空間はバンク"00<sub>H</sub>" ("000000<sub>H</sub> ~ 00FFFF<sub>H</sub>")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 デフォルト空間とアドレッシングモード

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

図 2.2-3に、レジスタバンクに分割された、メモリ空間の例を示します。

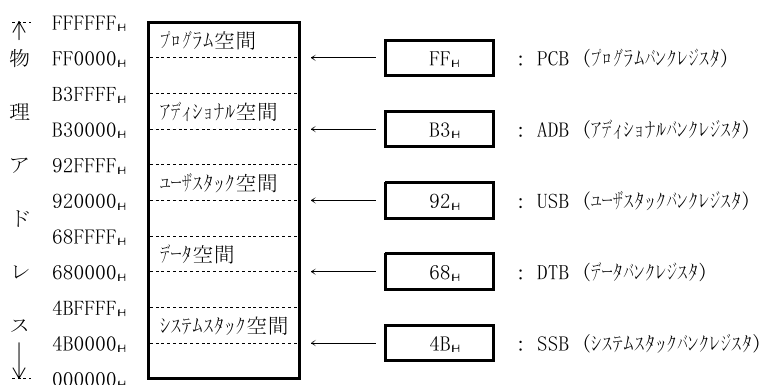


図 2.2-3 各空間の物理アドレスの例

### 2.2.1 メモリ空間における多バイト長データの配置

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

#### メモリ空間における多バイト長データの配置

図 2.2-4に示すように、データのメモリへ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。

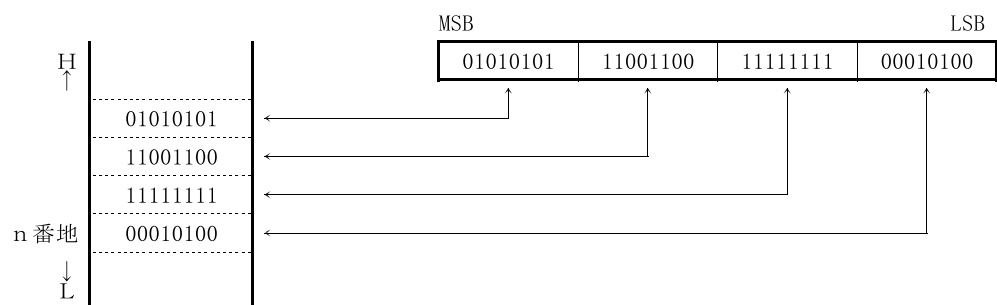


図 2.2-4 メモリ空間における多バイト長データの配置例

#### 多バイト長データのアクセス

図 2.2-5に示すように、アクセスはすべてバンク内を基本として行われますので、多バイト長のデータをアクセスする命令では、"FFFF<sub>H</sub>"番地の次のアドレスは同じバンクの"0000<sub>H</sub>"番地になります。

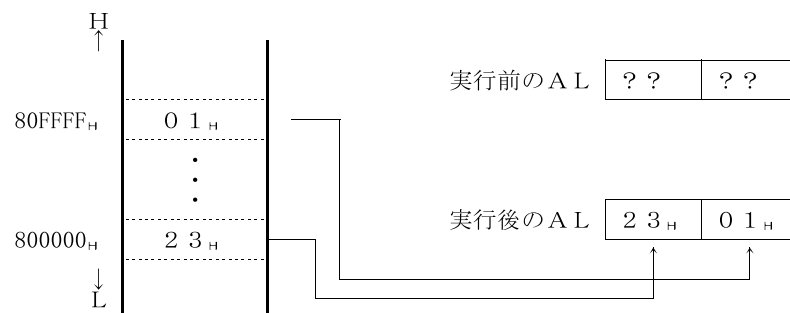


図 2.2-5 多バイト長データアクセスの例 (MOVWA, 080FFFF<sub>H</sub>の実行)

## 2.3 専用レジスタ

---

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

---

### 専用レジスタ

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

F<sup>2</sup>MC-16LXの専用レジスタは、以下に示す13本があります。

アキュムレータ (A=AH:AL)

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

ユーザスタックポインタ (USP)

ユーザスタック領域を示す16bitのポインタです。

システムスタックポインタ (SSP)

システムスタック領域を示す16bitのポインタです。

プロセッサステータス (PS)

システムの状態を示す16bitのレジスタです。

プログラムカウンタ (PC)

プログラムが格納されているアドレスを持つ16bitのレジスタです。

プログラムバンクレジスタ (PCB)

PC空間を示す8bitのレジスタです。

データバンクレジスタ (DTB)

DT空間を示す8bitのレジスタです。

ユーザスタックバンクレジスタ (USB)

ユーザスタック空間を示す8bitのレジスタです。

システムスタックバンクレジスタ (SSB)

システムスタック空間を示す8bitのレジスタです。

アディショナルバンクレジスタ (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。

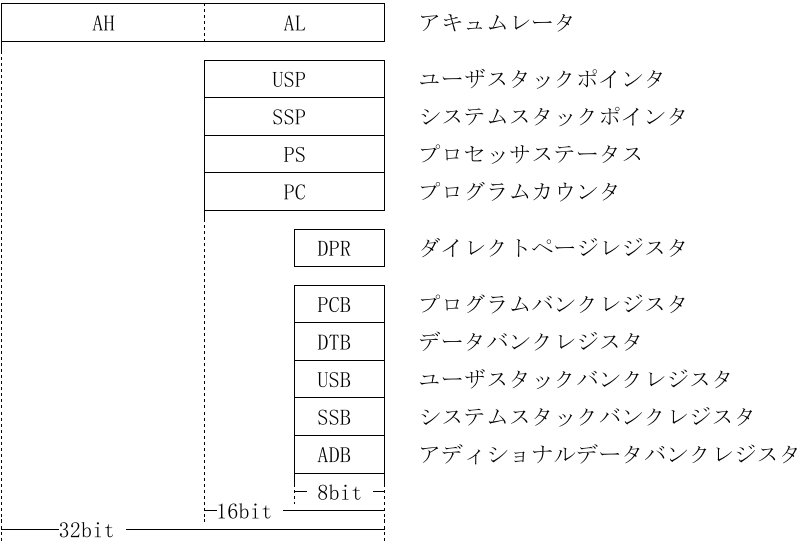


図 2.3-1 専用レジスタ

### 2.3.1 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と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データ処理モードのワード処理や8bitデータ処理モードのバイト処理の時はALのみが使用されます。

#### アキュムレータ(A)

アキュムレータ(A)中のデータは、メモリ/レジスタ(Ri, RWi, RLi)中のデータと各種演算ができます。F<sup>2</sup>MC-8Lの時と同様に、F<sup>2</sup>MC-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。

Aレジスタは、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。

MOVW A, @RW1+6 (RW1の内容+8ビット長オフセットの結果を番地としてロングワード長リードを行い、その内容をAに格納する命令)

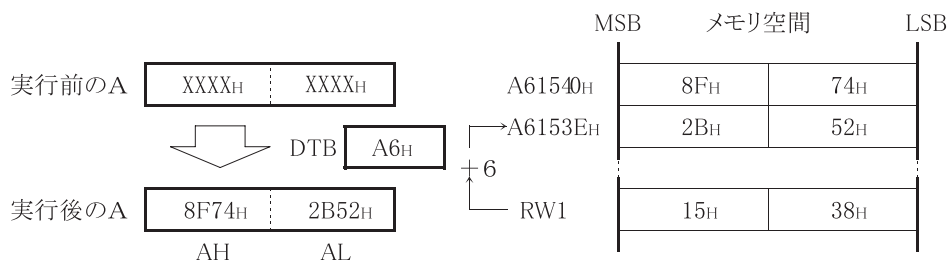


図 2.3-2 32bitデータ転送例

MOVW A, @RW1+6 (RW1の内容+8ビット長オフセットの結果を番地としてワード長リードを行い、その内容をAに格納する命令)

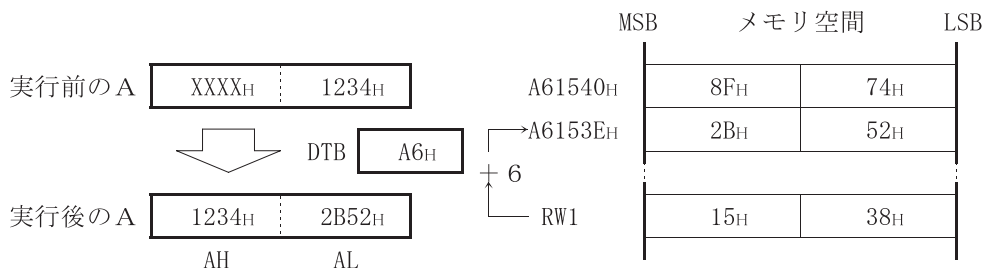


図 2.3-3 AL - AH転送例

図 2.3-4に示すように、ALへバイト長以下のデータを転送する時、データは符号拡張またはゼロ拡張され16bit長となってALへ格納されます。AL中のデータは、ワード長としてもバイト長としても扱えます。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。

MOVX A, 3000H （ 3000番地の内容をゼロ拡張し、 ALへ格納する命令）

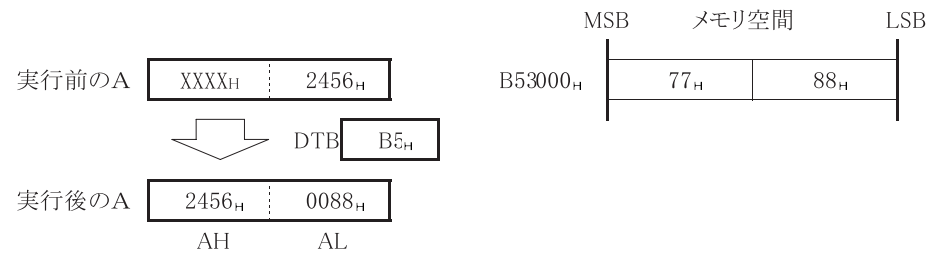


図 2.3-4 ゼロ拡張実行例

MOVX A, 3000H （3000H 番地の内容を符号拡張し、ALへ格納する命令）

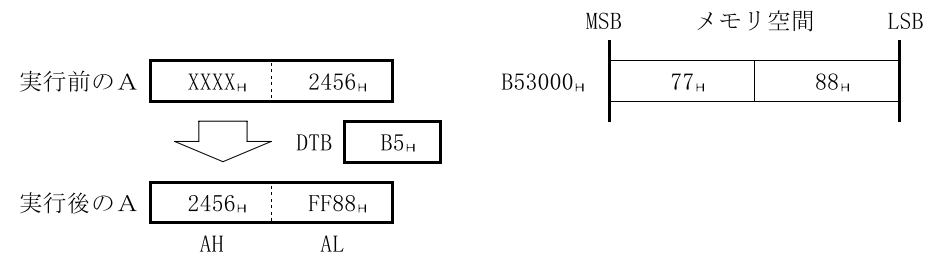


図 2.3-5 符号拡張実行例



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

ユーザスタックポインタ(USP)とシステムスタックポインタ(SSP)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"の時は、USPレジスタが有効になり、Sフラグが"1"の時はSSPレジスタが有効になります(図2.3-6と図2.3-7を参照)。また、割込みが受け付けられるとSフラグがセットされるため、割込み時のレジスタ退避は必ずSSPの示すメモリ中に行われます。割込みルーチンでのスタック処理はSSPが用いられ、割込みルーチン以外のスタック処理はUSPが用いられます。スタック空間を分ける必要のない場合はSSPだけをお使いください。スタック時のアドレスの上位8bitは、SSP、SSB、USP、USBにより示されます。また、USPおよびSSPは、リセットでは初期化されず、不定値になります。

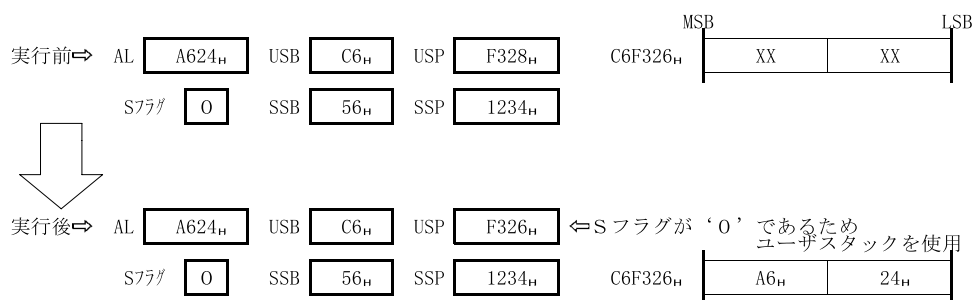


図 2.3-6 スタック操作命令とスタックポインタ(Sフラグが"0"のときの PUSHW A例)

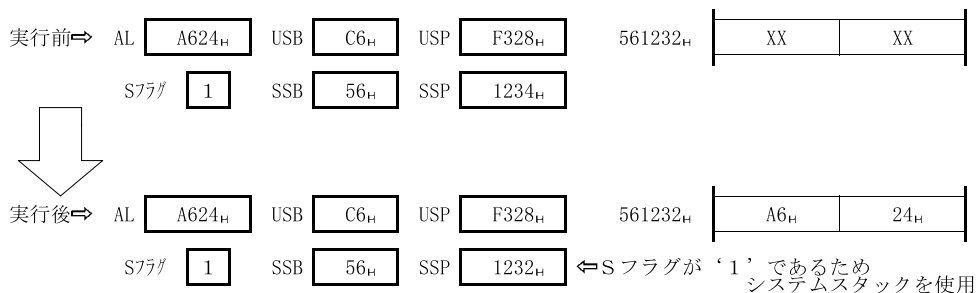


図 2.3-7 スタック操作命令とスタックポインタ(Sフラグが"1"のときの PUSHW A例)

#### < 注意事項 >

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

### 2.3.3 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。

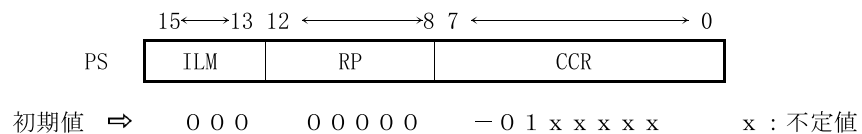


図 2.3-8 プロセッサステータス(PS)の構成

#### コンディションコードレジスタ(CCR)

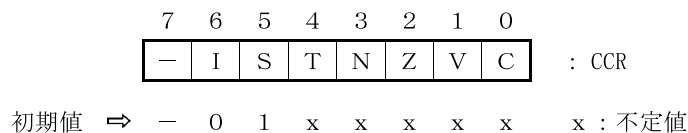


図 2.3-9 コンディションコードレジスタ(CCR)の構成

#### 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"の時は割込みが許可され、"0"の時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"の時は、スタック操作作用ポインタとしてUSPが有効になり、"1"の時はSSPが有効になります。また、割込み受け付け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"があったら"1"、それ以外は"0"になります。また、シフト量がゼロの場合も"0"になります。

#### ネガティブフラグ(N)

演算結果のMSBが"1"ならセットされ、"0"ならクリアされます。

#### ゼロフラグ(Z)

演算結果がすべて"0"ならセットされ、それ以外はクリアされます。

#### オーバフローフラグ(V)

演算の実行により符号付き数値としてオーバ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、F<sup>2</sup>MC-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す。現在使用中のレジスタバンクの先頭メモリアドレスを $[000180_H + (RP) * 10_H]$ という変換式で示します。RPは5bitで構成され、" $00_H \sim 1F_H$ "までの値をとることができ、また、" $000180_H \sim 00037F_H$ "のメモリ中にレジスタバンクを配置できます。

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

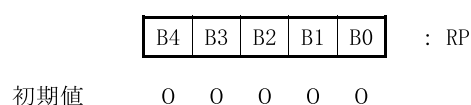


図 2.3-10 レジスタバンクポインタ(RP)の構成

## 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の割り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り込み要求のみが受付られます。表 2.3-1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。従って割り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。割り込みが受付られるとその割り込みのレベル値が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り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。

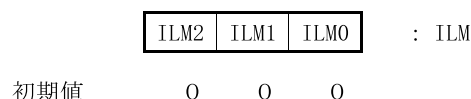


図 2.3-11 インタラプトレベルマスクレジスタ(ILM)の構成

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

ILM2	ILM1	ILM0	レベル値	許可される割り込みレベル
0	0	0	0	割り込み禁止
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

### 2.3.4 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。

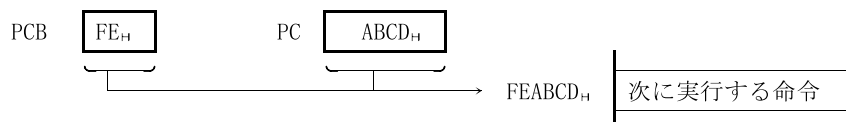


図 2.3-12 プログラムカウンタの構成

### 2.3.5 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"01<sub>H</sub>"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 2.3-13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。

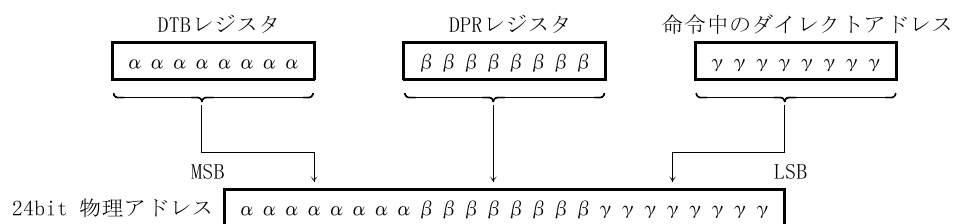


図 2.3-13 ダイレクトアドレッシングによる物理アドレスの生成

## 2.3.6 バンクレジスタ

---

バンクレジスタには、次の5つがあります。

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

### 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、リセットベクタにより、PCBは"00<sub>H</sub>"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、書込むことはできません。

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

## 2.4 汎用レジスタ

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

### 汎用レジスタ

F<sup>2</sup>MC-16LXの汎用レジスタは、RAM上の"000180<sub>H</sub> ~ 00037F<sub>H</sub>"（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レジスタバンクとして使用されているかを指定します。各バンクには以下に示す3種のレジスタが存在します。これらは独立ではなく、図 2.4-1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

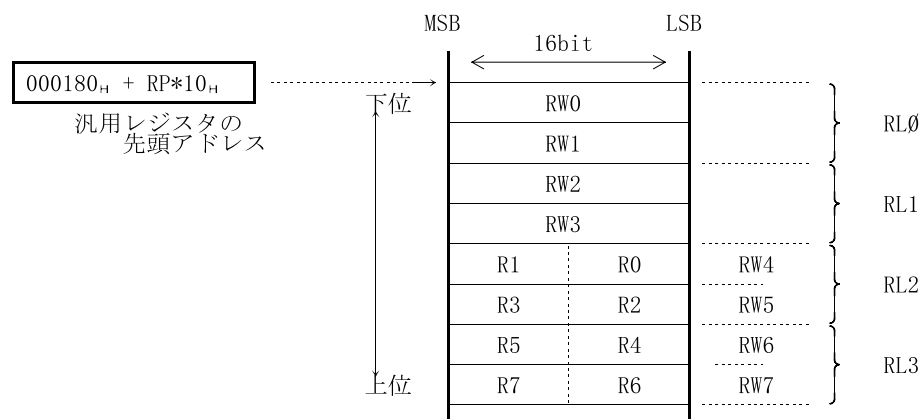


図 2.4-1 汎用レジスタ

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。

$$PW(i+4)=R(i \times 2+1) \times 256 + R(i \times 2) \quad [i=0 \sim 3]$$

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。

$$RL(i)=RW(i \times 2+1) \times 65536 + RW(i \times 2) \quad [i=0 \sim 3]$$

### レジスタバンク

レジスタバンクは、8wordで構成されています。レジスタバンクの内容は、通常のRAMと同様に、リセットでは初期化されず、リセット前の状態が保持されます。ただし、パワーオン時は、不定値になります。

表 2.4-1に示すように、レジスタバンクは、バイトレジスタ(R0 ~ R7)、ワードレジスタ(RW0 ~ RW7)、ロングワードレジスタ(RL0 ~ RL3)の汎用レジスタとして使用でき、また、各種演算、ポインタとして各種命令に使用できます。

表 2.4-2に、レジスタバンクの各レジスタの関係を示す。

表 2.4-1 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 <注記> R0はパレルシフトのカウンタおよびノーマライゼーション命令のカウンタとしても使用
RW0 ~ RW7	ポインタ，各種命令のオペランドとして使用 <注記> RW0は，ストリング命令のカウンタとして使用
RL0 ~ RL3	ロングポインタ，各種命令のオペランドとして使用

表 2.4-2 レジスタバンクの各レジスタの関係

	RW0	RL0
	RW1	
	RW2	RL1
	RW3	
R0	RW4	RL2
R1		
R2		
R3		
R4	RW6	RL3
R5		
R6		
R7		



## 2.5 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、命令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、任意に選択する事が可能です。

表 2.5-1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 2.5-1 バンクセレクトプリフィックス

バンクセレクトプリフィックス	選択される空間
PCB	PC空間
DTB	データ空間
ADB	AD空間
SPB	その時のスタックフラグの内容によりSSP空間、USP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〔MOV<sub>S</sub>, MOV<sub>SW</sub>, 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<sub>B</sub> A, io:bp / MOV<sub>B</sub> io:bp, A / SET<sub>B</sub> io:bp / CLR<sub>B</sub> 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およびます。

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およびます。

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およびます。

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、その時のRPがどのような値でも、比較的簡単に定められた同一のレジスタバンクをアクセスする手段が必要です。コモンレジスタバンクプリフィックス(CMR)を、レジスタバンクをアクセスする命令の前に置くことで、現在のRPの値に関係なくその命令のレジスタアクセスをすべて"000180<sub>H</sub> ~ 00018F<sub>H</sub>"にあるコモンバンク (RP=0の時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 [ MOV<sub>S</sub>, MOV<sub>SW</sub>, 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 [ MOV<sub>S</sub>, MOV<sub>SW</sub>, 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およびます。

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およびます。

### 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 [ MOV<sub>S</sub>, MOV<sub>SW</sub>, 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 [ MOV<sub>S</sub>, MOV<sub>SW</sub>, 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およびます。

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およびます。

## 2.6 割込み抑止命令とプリフィックスコード

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

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

### 割込み抑止命令

図 2.6-1に示すように、割込み抑止命令を実行中に、有効なハードウェア割込み要求が発生しても、割込み処理が行われるのは、割込み抑止命令以降に、割込み抑止命令以外の命令が実行された後になります。

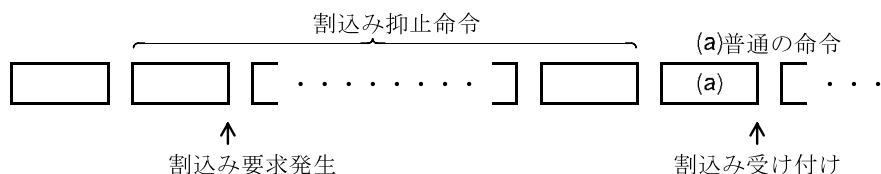


図 2.6-1 割込み抑止命令

### 割込み抑止命令とプリフィックス命令に関する制約

図 2.6-2に示すように、割込み抑止命令の前にプリフィックスコードを付加した場合、プリフィックスコードの効果は、プリフィックスコード後の最初の〔割込み抑止命令以外の命令〕までおよびます。

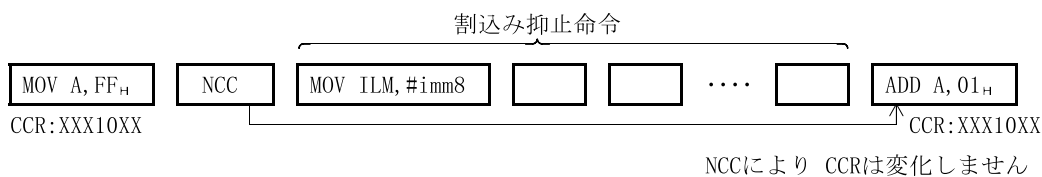


図 2.6-2 割込み抑止命令とプリフィックスコード

## プリフィックスコードが連続している場合

図 2.6-3に示すように、競合するプリフィックスコードが連続していた場合、後方のものが有効になります。競合するプリフィックスコードとはここでいう PCB, ADB, DTB, SPB のことを意味します。

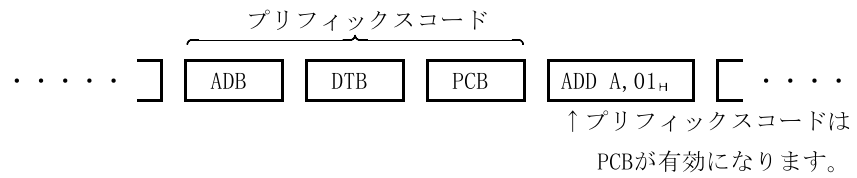


図 2.6-3 プリフィックスコードの連続

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

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

「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意

表 2.7-1 「DIV A,Ri」,「DIVW A,RWi」命令の使用上の注意 (i=0~7)

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

\*1: CCRレジスタのSビットによる

\*2: CCRレジスタのSビットが0の場合

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

【例】

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

ここで,「DIV A,R0」で指定されるバンクレジスタはデータバンクレジスタ (DTB) ですので,バンクアドレス"053<sub>H</sub>"を付加したアドレス"05301B8<sub>H</sub>"に余りが格納されます (RiおよびRWiのレジスタについては,「2.4 汎用レジスタ」を参照してください)。

## 注意事項の回避について

「DIV A,Ri」, 「DIVW A,RWi」命令の使用上の注意事項を回避してプログラムを開発していただくために,コンパイラでは表 2.7-1の命令を生成しないように変更し,アセンブラでは表 2.7-1の命令を同等の命令列に置き換える機能を追加したものが用意されています。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 (EI<sup>2</sup>OS)
- 3.7 未定義命令の実行による例外発生



## 3.1 割込みの概要

---

F<sup>2</sup>MC-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 (EI<sup>2</sup>OS)：内蔵リソースのイベント発生による転送処理
- 例外：動作例外事項の発生による中断処理

## 3.2 割り込み要因

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

割り込み要因

表 3.2-1 割り込み要因と割り込みベクタ・割り込み制御レジスタ

割り込み要因	EI <sup>2</sup> OSクリア	割り込みベクタ		割り込み制御レジスタ	
		番号	アドレス	番号	アドレス
リセット	×	# 08	FFFFDC <sub>H</sub>	-	-
INT9命令	×	# 09	FFFFD8 <sub>H</sub>	-	-
例外	×	# 10	FFFFD4 <sub>H</sub>	-	-
A/Dコンバータ		# 11	FFFFD0 <sub>H</sub>	ICR00	0000B0 <sub>H</sub>
タイムベースタイマ	×	# 12	FFFFCC <sub>H</sub>		
DTP0（外部割り込み#0）/UART3受信完了		# 13	FFFFC8 <sub>H</sub>	ICR01	0000B1 <sub>H</sub>
DTP1（外部割り込み#1）/UART4受信完了		# 14	FFFFC4 <sub>H</sub>		
DTP2（外部割り込み#2）/UART3送信完了		# 15	FFFFC0 <sub>H</sub>	ICR02	0000B2 <sub>H</sub>
DTP3（外部割り込み#3）/UART4送信完了		# 16	FFFFBC <sub>H</sub>		
DTP4-7（外部割り込み#4-#7）		# 17	FFFFB8 <sub>H</sub>	ICR03	0000B3 <sub>H</sub>
アウトプットコンペア(ch.1)一致 （入出力タイマ）		# 18	FFFFB4 <sub>H</sub>		
UART2受信完了		# 19	FFFFB0 <sub>H</sub>	ICR04	0000B4 <sub>H</sub>
UART1受信完了		# 20	FFFFAC <sub>H</sub>		
インプットキャプチャ(Ch.3)取込み （入出力タイマ）		# 21	FFFFA8 <sub>H</sub>	ICR05	0000B5 <sub>H</sub>
インプットキャプチャ(Ch.2)取込み （入出力タイマ）		# 22	FFFFA4 <sub>H</sub>		
インプットキャプチャ(Ch.1)取込み （入出力タイマ）		# 23	FFFFA0 <sub>H</sub>	ICR06	0000B6 <sub>H</sub>
インプットキャプチャ(Ch.0)取込み （入出力タイマ）		# 24	FFFF9C <sub>H</sub>		
8/16bit PPG0カウンタ・ボロー	×	# 25	FFFF98 <sub>H</sub>	ICR07	0000B7 <sub>H</sub>
16bit リロードタイマ2-0		# 26	FFFF94 <sub>H</sub>		
時計プリスケラ	×	# 27	FFFF90 <sub>H</sub>	ICR08	0000B8 <sub>H</sub>
アウトプットコンペア(ch.0)一致(入出力タイマ)		# 28	FFFF8C <sub>H</sub>		
UART2送信完了		# 29	FFFF88 <sub>H</sub>	ICR09	0000B9 <sub>H</sub>
PWCタイマ測定完了/オーバフロー		# 30	FFFF84 <sub>H</sub>		
UART1送信完了		# 31	FFFF80 <sub>H</sub>	ICR10	0000BA <sub>H</sub>
16bitフリーランタイマ（入出力タイマ） オーバフロー		# 32	FFFF7C <sub>H</sub>		
UART0送信完了		# 33	FFFF78 <sub>H</sub>	ICR11	0000BB <sub>H</sub>
8/16bit PPG1カウンタ・ボロー	×	# 34	FFFF74 <sub>H</sub>		
IEBus受信完了		# 35	FFFF70 <sub>H</sub>	ICR12	0000BC <sub>H</sub>
IEBus送信開始		# 37	FFFF68 <sub>H</sub>	ICR13	0000BD <sub>H</sub>
UART0 受信完了		# 39	FFFF60 <sub>H</sub>	ICR14	0000BE <sub>H</sub>
フラッシュメモリステータス	×	# 41	FFFF58 <sub>H</sub>	ICR15	0000BF <sub>H</sub>
遅延割り込み	×	# 42	FFFF54 <sub>H</sub>		

：EI<sup>2</sup>OS割り込みクリア信号にて割り込み要求フラグはクリアされます。ストップ要求あり。

：EI<sup>2</sup>OS割り込みクリア信号にて割り込み要求フラグはクリアされます。

×：EI<sup>2</sup>OS割り込みクリア信号にて割り込み要求フラグはクリアされません。

< 注意事項 >

---

同一割込み番号に2つの割込み要因があるリソースは、両方の割込み要求フラグがEI<sup>2</sup>OS割込みクリア信号でクリアされます。したがって、2つの要因の中で、どちらか1つにEI<sup>2</sup>OS機能を使用すると、もう1つの割込み機能は使用できません。該当リソースの割込み要求許可ビットを"0"にして、ソフトウェアポーリング処理にて対処してください

---

## 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（続く）

ソフトウェア 割込命令	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No	ハードウェア割込み
INT 0	FFFFFC <sub>H</sub>	FFFFFD <sub>H</sub>	FFFFFE <sub>H</sub>	未使用	#0	なし
:	:	:	:	:	:	:
INT 7	FFFFE0 <sub>H</sub>	FFFFE1 <sub>H</sub>	FFFFE2 <sub>H</sub>	未使用	#7	なし
INT 8	FFFFDC <sub>H</sub>	FFFFDD <sub>H</sub>	FFFFDE <sub>H</sub>	FFFFDF	#8	(RESETベクトル)
INT 9	FFFFD8 <sub>H</sub>	FFFFD9 <sub>H</sub>	FFFFDA <sub>H</sub>	未使用	#9	なし
INT 10	FFFFD4 <sub>H</sub>	FFFFD5 <sub>H</sub>	FFFFD6 <sub>H</sub>	未使用	#10	<例外>
INT 11	FFFFD0 <sub>H</sub>	FFFFD1 <sub>H</sub>	FFFFD2 <sub>H</sub>	未使用	#11	A/Dコンバータ
INT 12	FFFFC8 <sub>H</sub>	FFFFC9 <sub>H</sub>	FFFFCA <sub>H</sub>	未使用	#12	タイムベースタイマ
INT 13	FFFFC4 <sub>H</sub>	FFFFC5 <sub>H</sub>	FFFFC6 <sub>H</sub>	未使用	#13	DTP0(外部割込み#0) / UART3受信完了
INT 14	FFFFC0 <sub>H</sub>	FFFFC1 <sub>H</sub>	FFFFC2 <sub>H</sub>	未使用	#14	DTP1(外部割込み#1) / UART4受信完了
INT 15	FFFFBC <sub>H</sub>	FFFFBD <sub>H</sub>	FFFFBE <sub>H</sub>	未使用	#15	DTP2(外部割込み#2) / UART3送信完了
INT 16	FFFFB8 <sub>H</sub>	FFFFB9 <sub>H</sub>	FFFFBA <sub>H</sub>	未使用	#16	DTP3(外部割込み#3) / UART4送信完了
INT 17	FFFFB4 <sub>H</sub>	FFFFB5 <sub>H</sub>	FFFFB6 <sub>H</sub>	未使用	#17	DTP4-7 (外部割込み#4-7)
INT 18	FFFFB0 <sub>H</sub>	FFFFB1 <sub>H</sub>	FFFFB2 <sub>H</sub>	未使用	#18	アウトプットコンペア (ch.1) 一致 (入出力タイマ)
INT 19	FFFFAC <sub>H</sub>	FFFFAD <sub>H</sub>	FFFFAE <sub>H</sub>	未使用	#19	UART2受信完了
INT 20	FFFFA8 <sub>H</sub>	FFFFA9 <sub>H</sub>	FFFFAA <sub>H</sub>	未使用	#20	UART1受信完了
INT 21	FFFFA4 <sub>H</sub>	FFFFA5 <sub>H</sub>	FFFFA6 <sub>H</sub>	未使用	#21	インプットキャプチャ (ch.3) 取込み (入出力タイマ)
INT 22	FFFFA0 <sub>H</sub>	FFFFA1 <sub>H</sub>	FFFFA2 <sub>H</sub>	未使用	#22	インプットキャプチャ (ch.2) 取込み (入出力タイマ)
INT 23	FFFF9C <sub>H</sub>	FFFF9D <sub>H</sub>	FFFF9E <sub>H</sub>	未使用	#23	インプットキャプチャ (ch.1) 取込み (入出力タイマ)
INT 24	FFFF98 <sub>H</sub>	FFFF99 <sub>H</sub>	FFFF9A <sub>H</sub>	未使用	#24	インプットキャプチャ (ch.0) 取込み (入出力タイマ)
INT 25	FFFF94 <sub>H</sub>	FFFF95 <sub>H</sub>	FFFF96 <sub>H</sub>	未使用	#25	8/16bit PPG0カウンタポロー
INT 26	FFFF90 <sub>H</sub>	FFFF91 <sub>H</sub>	FFFF92 <sub>H</sub>	未使用	#26	16bitリロードタイマ2-0
INT 27	FFFF8C <sub>H</sub>	FFFF8D <sub>H</sub>	FFFF8E <sub>H</sub>	未使用	#27	時間プリスケラ
INT 28	FFFF88 <sub>H</sub>	FFFF89 <sub>H</sub>	FFFF8A <sub>H</sub>	未使用	#28	アウトプットコンペア (ch.0) 一致 (入出力タイマ)
INT 29	FFFF84 <sub>H</sub>	FFFF85 <sub>H</sub>	FFFF86 <sub>H</sub>	未使用	#29	UART2送信完了
INT 30	FFFF80 <sub>H</sub>	FFFF81 <sub>H</sub>	FFFF82 <sub>H</sub>	未使用	#30	PWCタイマ測定完了 / オーバフロー
INT 31	FFFF7C <sub>H</sub>	FFFF7D <sub>H</sub>	FFFF7E <sub>H</sub>	未使用	#31	UART1送信完了
INT 32	FFFF78 <sub>H</sub>	FFFF79 <sub>H</sub>	FFFF7A <sub>H</sub>	未使用	#32	16ビットフリーランタイム (入出力タイマ)オーバフロー
INT 33	FFFF74 <sub>H</sub>	FFFF75 <sub>H</sub>	FFFF76 <sub>H</sub>	未使用	#33	UART0送信完了
INT 34	FFFF70 <sub>H</sub>	FFFF71 <sub>H</sub>	FFFF72 <sub>H</sub>	未使用	#34	8/16ビットPPG1カウンタポロー
INT 35	FFFF68 <sub>H</sub>	FFFF69 <sub>H</sub>	FFFF6A <sub>H</sub>	未使用	#35	IEBus受信完了
INT 37				未使用	#37	IEBus送信開始

表 3.3-1 割り込みベクター一覧表 ( 続き )

ソフトウェア 割り込み命令	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	割り込みNo	ハードウェア割り込み
INT 38	FFFF64 <sub>H</sub>	FFFF65 <sub>H</sub>	FFFF66 <sub>H</sub>	未使用	#38	なし
INT 39	FFFF60 <sub>H</sub>	FFFF61 <sub>H</sub>	FFFF62 <sub>H</sub>	未使用	#39	UART0受信完了
INT 41	FFFF58 <sub>H</sub>	FFFF59 <sub>H</sub>	FFFF5A <sub>H</sub>	未使用	#41	フラッシュメモリステータス
INT 42	FFFF54 <sub>H</sub>	FFFF55 <sub>H</sub>	FFFF56 <sub>H</sub>	未使用	#42	遅延割り込み
:	:	:	:	:	:	:
INT254	FFFC04 <sub>H</sub>	FFFC05 <sub>H</sub>	FFFC06 <sub>H</sub>	未使用	#254	なし
INT255	FFFC00 <sub>H</sub>	FFFC01 <sub>H</sub>	FFFC02 <sub>H</sub>	未使用	#255	なし

## 3.4 ハードウェア割込み

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

### ハードウェア割込みの概要

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

ハードウェア割込みが発生するときにCPUが行う処理は以下のものがあります。

- CPU内部のPC, PS, AH, AL, PCB, DTB, ADB, DPRレジスタのシステムスタックへの退避
- PSレジスタ内のILMを設定。自動的に現在要求している割込みレベルと同じになる
- 対応する割込みベクトルの値を取り込み、その値が指示する処理へ分岐

### ハードウェア割込みの構造

ハードウェア割込みに関連する機構は、以下の3つの部分に分かれて存在します。

#### 内蔵リソース

割込み許可ビット、割込み要求ビット：リソースからの割込み要求の制御

#### 割込みコントローラ

ICR：割込みのレベル付け、同時要求割込みの優先度判定

#### CPU

I, ILM：要求割込みレベルと現レベルの比較、割込み許可状態の識別

マイクロコード：割込み処理用ステップ

上記3つの区分の状態は、内蔵リソースではリソース制御レジスタに、割込みコントローラではICRに、CPUではCCRの値によって指示されます。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つの区分を事前に設定します。ICRについては、「拡張インテリジェントI/Oサービス割込み制御レジスタ(ICR)」を参照してください。

割込み処理の際に参照する割込みベクトルのテーブルはメモリ領域の"FFFC00<sub>H</sub> ~ FFFFFF<sub>H</sub>"に割り当ててあり、ソフトウェア割込みと共通で使用しています。

### 内蔵リソース領域への書き込み中のハードウェア割込み要求

内蔵リソース領域への書き込み中はハードウェア割込み要求を受け付けません。これは各リソースの割込み制御レジスタ関係の書換えを行っている最中の割込み要求に対して、CPUが割込み関係で誤動作を起こすことを避けるためです。内蔵リソース領域とは、"000000<sub>H</sub> ~ 0000FF<sub>H</sub>"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り当てられている領域のことを示します。

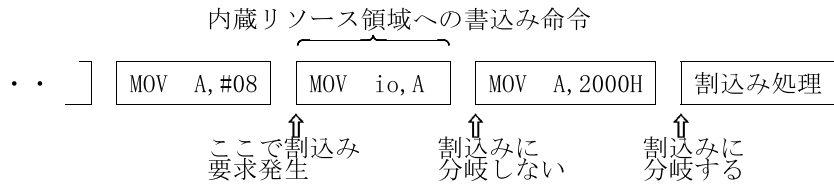


図 3.4-1 内蔵リソース領域への書き込み中のハードウェア割り込み要求

割り込み抑止命令

F<sup>2</sup>MC-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 割り込み抑止命令とプリフィックスコード」を参照してください)。

多重割り込み

F<sup>2</sup>MC-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高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、制御は高いレベルの割り込みへ転送されます。高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低いレベルの割り込み処理が発生した場合は、ILMの内容やIフラグの命令によって変更しないかぎり、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になります。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、他の割り込み要求および拡張インテリジェントI/Oサービス要求は全て保留になります。

スタック内へのレジスタの退避

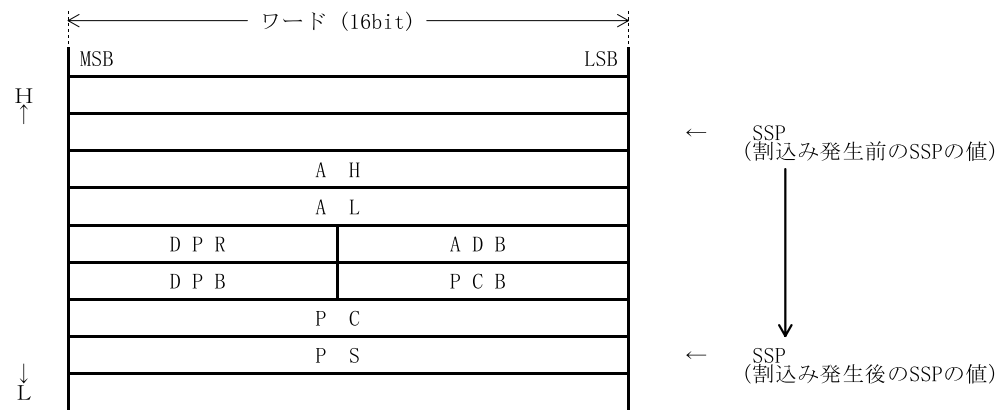


図 3.4-2 スタック内に退避されたレジスタ

ハードウェア割り込みの使用上の注意

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

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

### 3.4.1 ハードウェア割込みの動作

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

#### ハードウェア割込みの動作

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

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

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

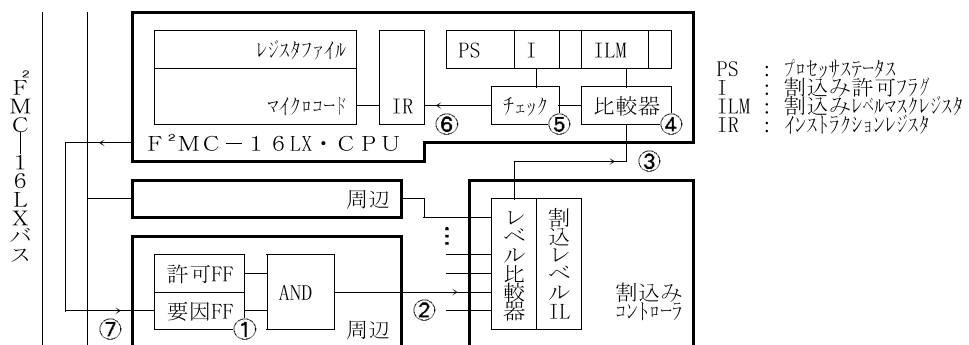


図 3.4-3 ハードウェア割込みの発生と解除まで

図 3.4-3の図中における、～の内容を以下に説明します。

周辺の内部で割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に対応する割込みレベルをCPUへ転送します。



CPUは割り込みコントローラから要求のあった割り込みレベルをプロセッサステータスレジスタ内のILMビットと比較します。

比較の結果が現在の割り込み処理レベルより優先順位が高い場合のみ、同じプロセッサステータスレジスタ内のIフラグの内容をチェックします。

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

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

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

CPUが行う割り込み処理の実行時間

CPUの割り込みシーケンスに遷移するまでの待ち時間（命令等の実行途中では割り込みシーケンスに遷移しない。）

割り込みシーケンスの実行時間

割り込み起動：24 + 6 × 表 3.4-1 マシンサイクル

割り込み復帰：15 + 6 × 表 3.4-1 マシンサイクル（RETI命令）

表 3.4-1 割り込み処理時のサイクル数の補正值

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

### 3.4.2 ハードウェア割込みの動作フロー

図 3.4-4に、ハードウェア割込みの動作フローを示します。

ハードウェア割込みの動作フロー

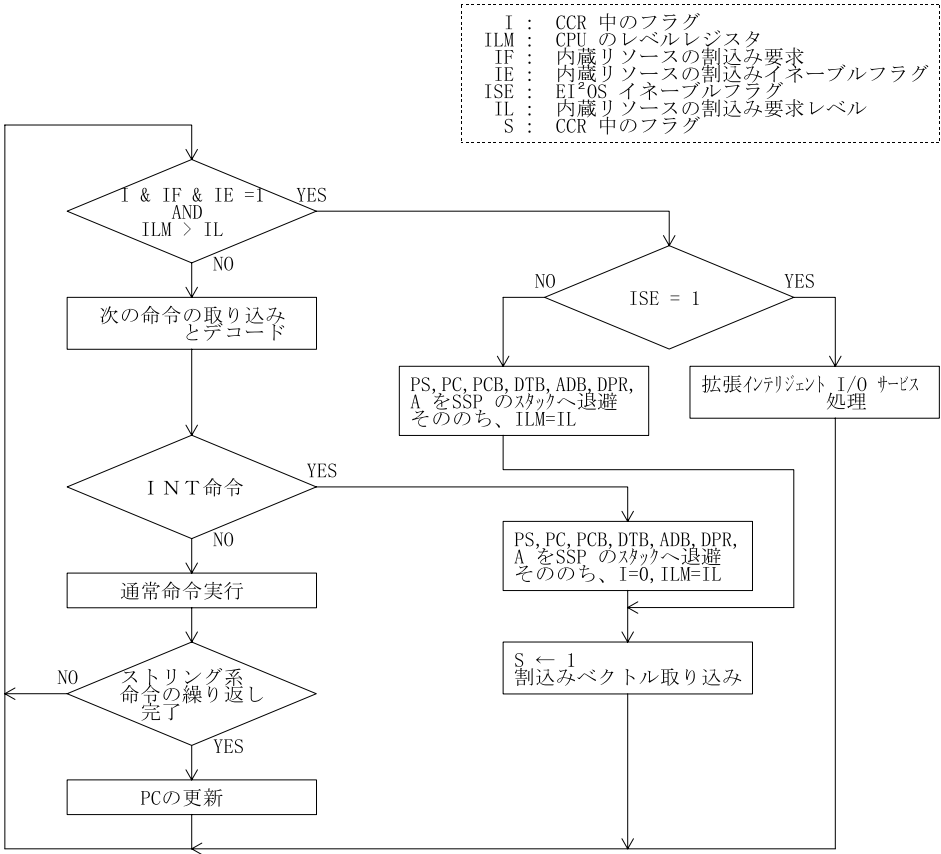


図 3.4-4 ハードウェア割込み動作フロー

### 3.4.3 ハードウェア割り込み使用手順例

図 3.4-5に，ハードウェア割り込み使用手順例を示します。

ハードウェア割り込み使用手順例

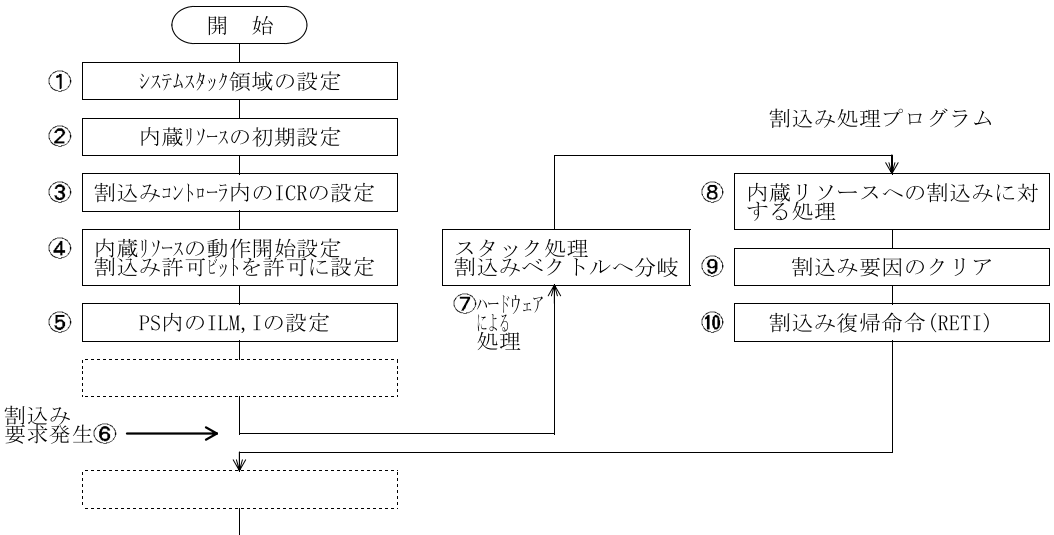


図 3.4-5 ハードウェア割り込み使用手順例

- 図 3.4-5の図中における， ～ の使用手順を以下に説明します。
- システムスタック領域を設定する。
  - 割り込み要求を発生可能な内蔵リソースの初期設定を行う。
  - 割り込みコントローラ内のICRを設定する。
  - 内蔵リソースを動作開始状態にし，割り込み許可ビットを許可に設定する。
  - CPU内のILM，Iフラグを割り込み受け付け可能に設定する。
  - 内蔵リソースの割り込み発生で，ハードウェア割り込み要求が発生する。
  - 割り込み処理ハードウェアにより，レジスタの退避が行われ，割り込み処理プログラムに分岐する。
  - 割り込み処理プログラムにて，割り込み発生に対する内蔵リソースへの処理を行う。
  - 内蔵リソース回路の割り込み要求を解除する。
  - 割り込み復帰命令を実行し，分岐前のプログラムに復帰する。

## 3.5 ソフトウェア割込み

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

### ソフトウェア割込みの概要

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

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

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

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

### ソフトウェア割込みの構造

ソフトウェア割込みは、すべてCPUの内部で取り扱われます。

#### CPU

- マイクロコード：割込み処理用ステップ

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

表 3.3-1に示すように、ソフトウェア割込みは、ハードウェア割込みと同じ割込みベクトル張き岐を共用します。たとえば、割込み要求番号のINT 13は、ハードウェア割込みの外部割込み#0/UART3受信完了に使用されると同時に、ソフトウェア割込みのINT #13に使用されます。したがって、外部割込み#0/UART3受信完了とINT #13は、同じ割込み処理ルーチンを呼び出します。

### ソフトウェア割込みの動作

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

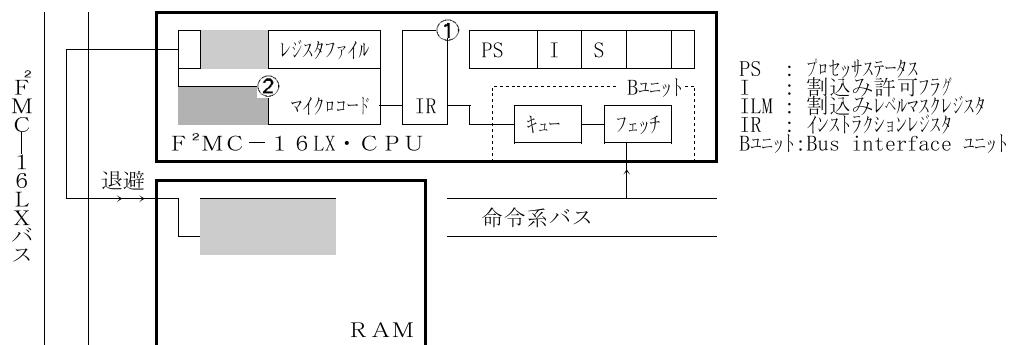


図 3.5-1の図中における、～ のソフトウェア割り込みの発生と解除までの流れを以下に示します。

ソフトウェア割り込み命令を実行します

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

ユーザの割り込み処理ルーチン内のRETI命令で割り込み処理が終了します。

#### ソフトウェア割り込みに関する注意事項

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

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

拡張インテリジェントI/Oサービス (EI<sup>2</sup>OS) とは、ハードウェア割込み動作の一種で、I/Oとメモリとの間でデータを自動的に転送します。従来は、割込み処理プログラムにより、I/Oとメモリの間でデータを転送していましたが、EI<sup>2</sup>OSでは、DMAモードのようなデータの転送が可能になります。

なお、REAROSを使用する場合は、拡張インテリジェントI/Oサービス (EI<sup>2</sup>OS) は使用できません。

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

拡張インテリジェントI/Oサービス (EI<sup>2</sup>OS) は、従来の割込み処理で行っていた方式と比べると以下に示すような利点があります。

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

EI<sup>2</sup>OSが終了時には、終了条件を設定した後に、CPUが自動的に割込み処理ルーチンへ分岐します。したがって、ユーザは終了条件の種類を判別することができます。

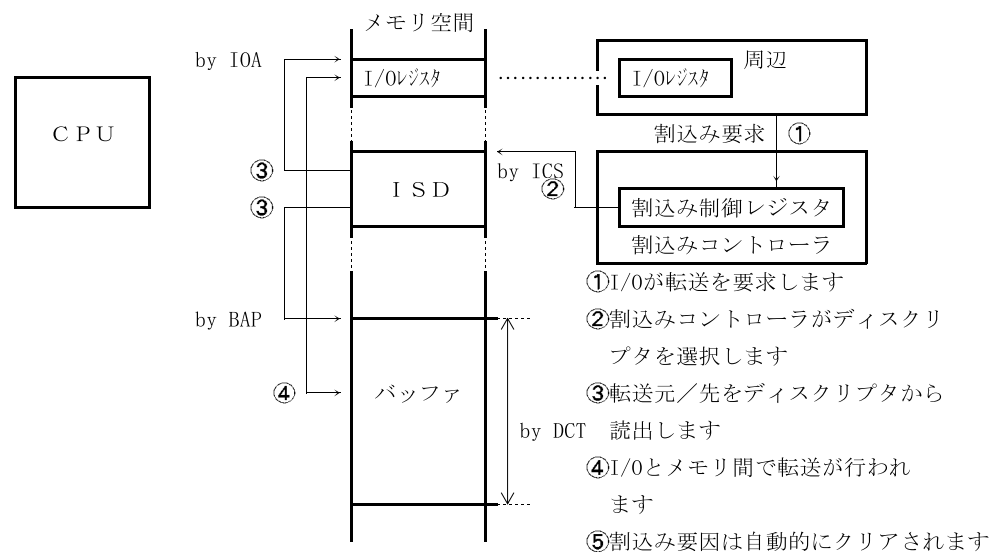
EI<sup>2</sup>OSを実現するために2ヵ所の部分にハードウェアが分散して存在し、それぞれのブロックの中に以下に示すようなレジスタやディスクリプタがあります。

#### 割込み制御レジスタ

割込みコントローラ内にありISDのアドレスを示します。

#### 拡張インテリジェントI/Oサービスディスクリプタ

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



(注)

- IOAで指定できる領域は、"000000<sub>H</sub>~00FFFF<sub>H</sub>"です。
- BAPで指定できる領域は、"000000<sub>H</sub>~FFFFFF<sub>H</sub>"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サービスの概要

#### 拡張インテリジェントI/Oサービス(EI²OS)の構造

EI²OSに関連する機構は、以下の4つの部分に分かれて存在します。

##### 内蔵リソース

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

##### 割り込みコントローラ

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

##### CPU

- I, ILM：要求割り込みレベルと現レベルの比較，割り込み許可状態の識別
- マイクロコード：EI²OS処理用ステップ

##### RAM

- ディスクリプタ：EI²OSの転送情報を記述する

### 3.6.1 割込み制御レジスタ(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3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サービスにするかの選択
- ・拡張インテリジェントI/Oサービスのチャネルの選択

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)		15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	↔ビットNo.
アドレス: B0 <sub>H</sub> ~ BF <sub>H</sub>		ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	書き込み時
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)	
		15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	↔ビットNo.
アドレス: B0 <sub>H</sub> ~ BF <sub>H</sub>		—	—	S1	S0	ISE	IL2	IL1	IL0	読出し時
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は，EI<sup>2</sup>OSを起動する場合のみ有効となります。EI<sup>2</sup>OSを起動する場合はISEを"1"に設定し，起動しない場合は"0"に設定してください。EI<sup>2</sup>OSを起動しない場合，ICS3 ~ 0は何を設定してもかまいません。

ICS1とICS0は書き込みのみ有効で，S1とS0は読出しのみ有効です。

#### 【ビット15 ~ 12，7 ~ 4】ICS3 ~ 0 (EI<sup>2</sup>OSチャネルセレクトビット)

ICS3 ~ 0ビットは，EI<sup>2</sup>OSチャネル選択ビットです。

書き込み専用のビットで，このビットによりEI<sup>2</sup>OSのチャネルを指定します。ここで設定された値によりメモリ上の拡張インテリジェントI/Oサービスディスクリプタのアドレスが決定します。ICSはリセットにより"0000"に初期化されます。



表 3.6-1 ICS3～0 (EI<sup>2</sup>OSチャンネルセレクトビット)

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 <sub>H</sub>
0	0	0	1	1	000108 <sub>H</sub>
0	0	1	0	2	000110 <sub>H</sub>
0	0	1	1	3	000118 <sub>H</sub>
0	1	0	0	4	000120 <sub>H</sub>
0	1	0	1	5	000128 <sub>H</sub>
0	1	1	0	6	000130 <sub>H</sub>
0	1	1	1	7	000138 <sub>H</sub>
1	0	0	0	8	000140 <sub>H</sub>
1	0	0	1	9	000148 <sub>H</sub>
1	0	1	0	10	000150 <sub>H</sub>
1	0	1	1	11	000158 <sub>H</sub>
1	1	0	0	12	000160 <sub>H</sub>
1	1	0	1	13	000168 <sub>H</sub>
1	1	1	0	14	000170 <sub>H</sub>
1	1	1	1	15	000178 <sub>H</sub>

【ビット13～12, 5～4】S0, S1 (EI<sup>2</sup>OSステータスビット)

S0, S1は, EI<sup>2</sup>OS終了ステータスビットです。

読出し専用のビットで, EI<sup>2</sup>OS終了時にこのビットの値を調べることにより, 終了条件が何であったかを判別することができます。リセットにより"00"に初期化されます。

表 3.6-2 拡張インテリジェントI/Oサービスステータスの終了条件

S1	S0	終了条件
0	0	EI <sup>2</sup> 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	予約
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 (EI<sup>2</sup>OSイネーブルビット)

ISEビットは, EI<sup>2</sup>OSを許可します。

割込み要求の発生時にこのビットが"1"であるとEI<sup>2</sup>OSが起動され, "0"のときは割込みシーケンスが起動されます。また, EI<sup>2</sup>OSが終了(カウント終了による場合と内蔵リソースからの要求による場合)したとき, ISEビットは"0"になります。対応する内蔵リソースにEI<sup>2</sup>OSの機能がないときには, ソフトウェアでISEを"0"にしておく必要があります。読み書き可能なビットです。リセットにより"0"に初期化されます。

## 【ビット10～8, 2～0】IL0, IL1, IL2 (割込みレベル設定ビット)

IL0, IL1, IL2ビットは, 割込みレベルを設定します。

対応する内蔵リソースの割込みレベルを指定します。読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レベル値

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

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

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

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

拡張インテリジェントI/Oサービスディスクリプタ(ISD)

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

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

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

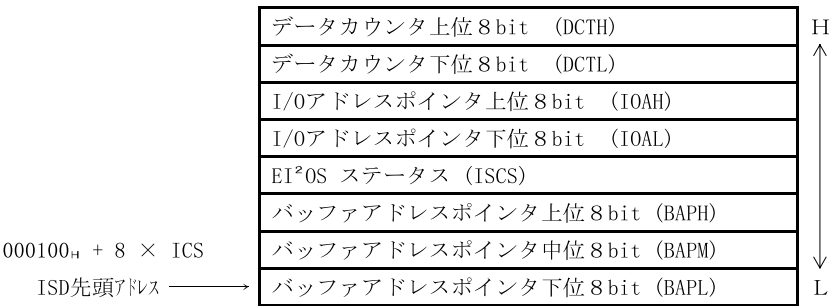


図 3.6-3 拡張インテリジェントI/Oサービスディスクリプタの構成

データカウンタ (DCT)

データカウンタ (DCT)は、16ビット長のレジスタで、転送されたデータ要素の数に対応したカウンタとして機能します。データ転送の前に、このカウンタは1個デクリメントされます。このカウンタがゼロになるとEI<sup>2</sup>OSは終了します。

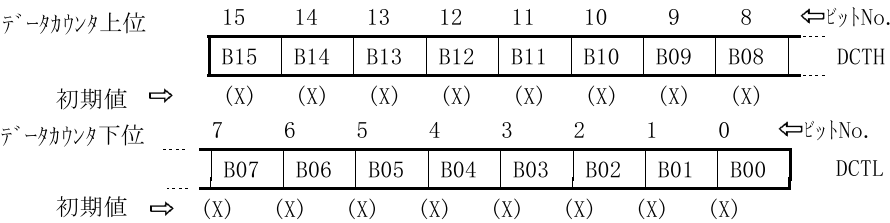


図 3.6-4 データカウンタ (DTC) の構成

## 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する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<sub>H</sub>～00FFFF<sub>H</sub>"番地までの任意のI/Oを指定できます。

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

EI<sup>2</sup>OSステータスレジスタ (ISCS)

EI<sup>2</sup>OSステータスレジスタ (ISCS) は、8ビット長のレジスタで、更新の向き (インクリメント/デクリメント)、転送データ形式 (バイト/ワード)、バッファアドレスポインタおよび、I/Oレジスタアドレスポインタの転送方向を指示します。また、このレジスタは、バッファアドレスポインタまたは、I/Oレジスタアドレスポインタが更新または固定されているかどうかも指示します。

EI <sup>2</sup> OSステータスレジスタ (ISCS)	7	6	5	4	3	2	1	0	⇐ビットNo.
	予約	予約	予約	IF	BW	BF	DIR	SE	
リード/ライト 初期値 ⇒	(-) (X)	(-) (X)	(-) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	

図 3.6-6 EI<sup>2</sup>OSステータスレジスタ (ISCS) の構成

## 【ビット7～5】

予約ビットです。ISCSを設定する時は、必ず"0"に設定してください。

## 【ビット4】IF

IFビットは、I/Oレジスタアドレスポインタの更新/固定を指定します。

表 3.6-4 I/Oレジスタアドレスポインタの更新/固定ビット(IF)

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

## 【ビット3】BW

BWビットは、転送データ長を指定します。

表 3.6-5 転送データ長を指定ビット(BW)

BW	機能
0	バイト
1	ワード

## 【ビット2】BF

BFビットは、バッファアドレスポインタの更新／固定を指定します。

表 3.6-6 バッファアドレスポインタの更新／固定指定ビット(BF)

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

## &lt; 注意事項 &gt;

---

更新する場合、バッファアドレスポインタは下位16bitのみ変化します。

---

## 【ビット1】: DIR

DIRビットは、データの転送方向を指定します。

表 3.6-7 データの転送方向指定ビット(DIR)の設定

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

## 【ビット0】SE

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

表 3.6-8 EI<sup>2</sup>OS終了制御ビット

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

## バッファアドレスポインタ (BAP)

24bitのレジスタで次にEI<sup>2</sup>OSによる転送で使用するアドレスを保持します。BAPはEI<sup>2</sup>OSの各チャネルに対してそれぞれ独立に存在しますので、EI<sup>2</sup>OSの各チャネルは16Mbyteの任意の空間を転送することができます。

## &lt; 注意事項 &gt;

---

ISCSのBFビットを"0"（更新あり）にした場合、BAPは下位16ビットのみ変化し、BAPHは変化しません。

---

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

図 3.6-7に拡張インテリジェントI/Oサービス (EI<sup>2</sup>OS) の動作フローを，図 3.6-8に，拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)の使用手順フローを示します。

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

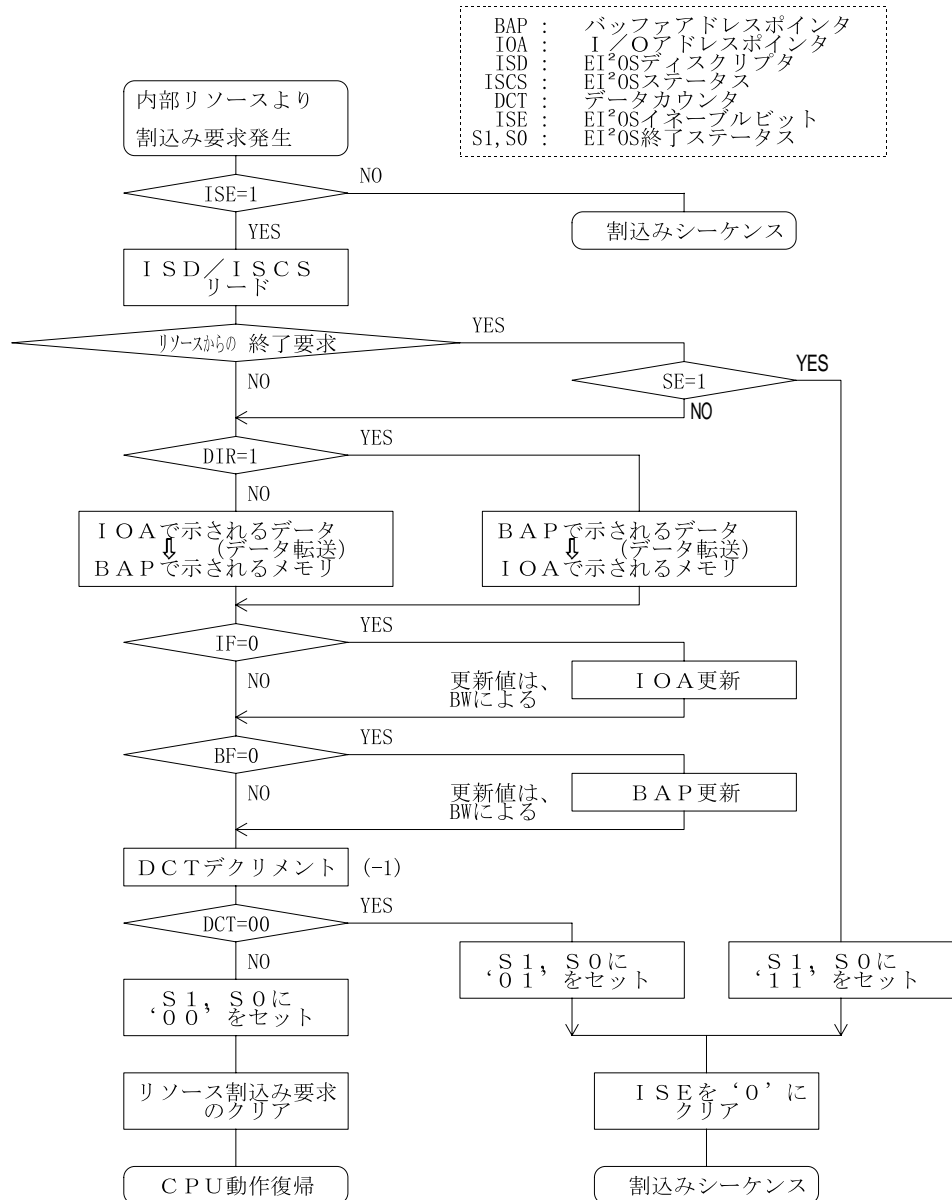


図 3.6-7 拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)動作フロー

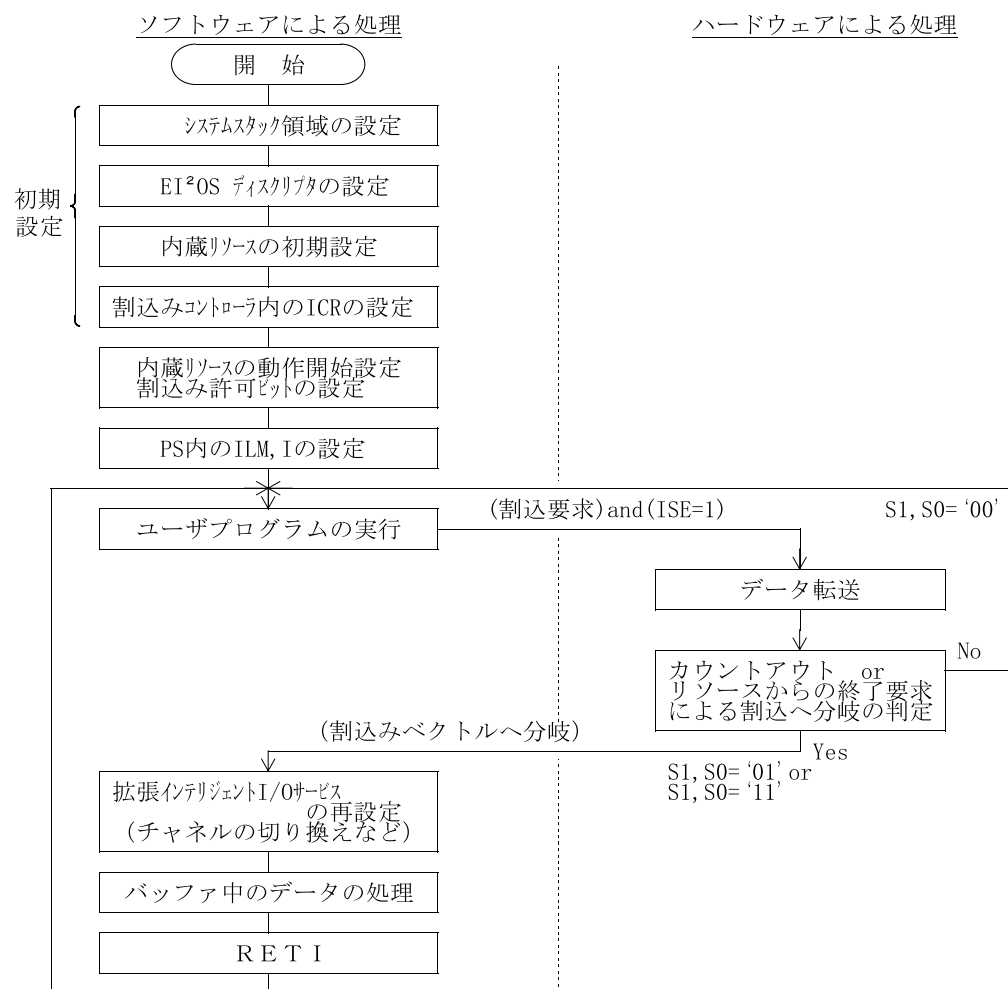


図 3.6-8 拡張インテリジェントI/Oサービス(EI²OS)の使用手順フロー

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

拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)の実行時間は次の3つの場合があります。

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

拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)の実行時間

データ転送継続時（停止条件が成立しない場合）

（表 3.6-9 + 表 3.6-10） マシンサイクル

表 3.6-9 EI<sup>2</sup>OSの継続時の実行時間

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

リソースからの停止要求時

（36 + 6 × 表 3.4-1）マシンサイクル

カウント終了時

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

表 3.6-10 EI<sup>2</sup>OS実行時間のデータ転送の補正值

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

**B** : バイトデータ転送

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



### 3.7 未定義命令の実行による例外発生

---

F<sup>2</sup>MC-16LX では、未定義命令の実行により例外が発生し、例外処理が行われます。

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

---

#### 未定義命令の実行による例外発生

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

## 第4章 クロックとリセット

---

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 リセット要因
- 4.3 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クロック通倍機能など内部クロックの動作制御を行います。この内部クロックのことをマシンのクロックとよび，その1周期をマシンサイクルとしています。また，原発振によるクロックをメインクロックとよび，内部のVCO発振によるクロックをPLLクロックとしています。

### クロック発生部に関する注意事項

動作電圧が5Vの時，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越える通倍率を指定した場合には，正常に動作しません。例えば，原発振が16MHzで発振している場合には，1通倍のみ指定が可能です。

また，VCO発振の最低動作周波数は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。

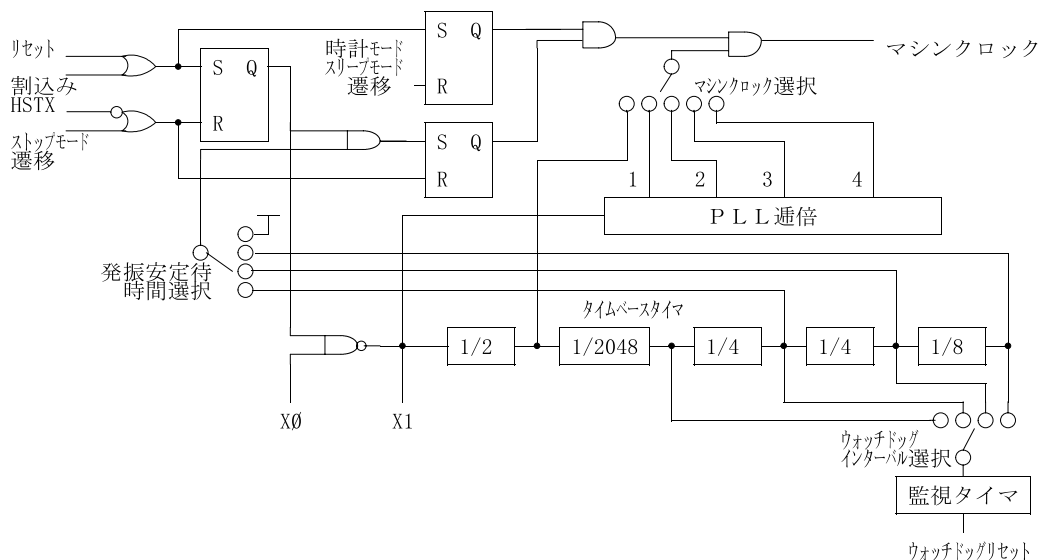


図 4.1-1 クロック発生回路ブロックダイアグラム

## 4.2 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、F<sup>2</sup>MC-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 4.2-1 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
		サブクロック時	PLLクロック時		
パワーオン	電源立ち上げ時	メインクロック*	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*	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*	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	メインクロック* またはPLLクロック	PLLクロック	前の状態を保持	なし
ソフトウェア	LPMCR中のRSTビットに"0"ライト	メインクロック* またはPLLクロック	PLLクロック	前の状態を保持	なし

\*:  $f_{osc}/2$  ( $f_{osc}$ : 原発振)

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

それぞれのリセット要因には対応したフリップフロップがあります。これらの内容はウォッチドッグタイマ制御レジスタを読み出すことで得ることができます。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。

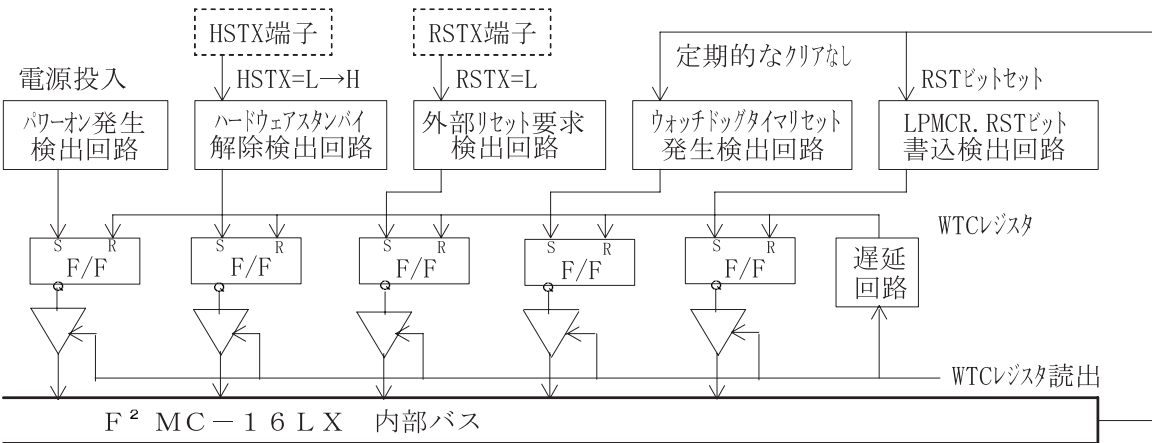


図 4.2-1 リセット要因ビットブロックダイアグラム

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

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

表 4.2-2 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9章ウォッチドッグタイマ」を参照してください。

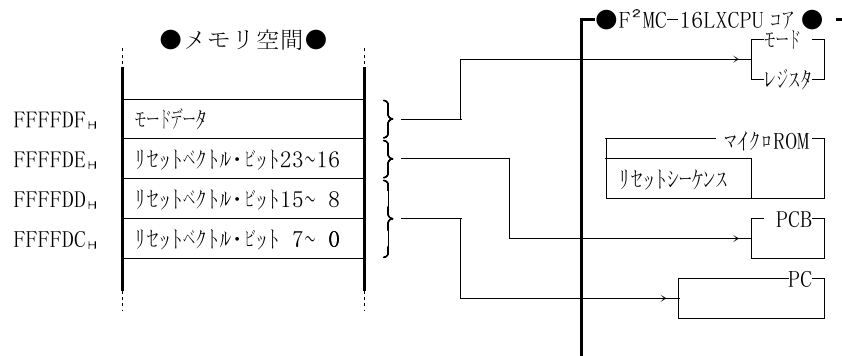
## 4.3 リセット解除後の動作

リセット要因が取り除かれると、F<sup>2</sup>MC-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<sub>H</sub> ~ FFFFDF<sub>H</sub>"の4 バイトに割り当てられており、それぞれリセット解除後にレジスタにハードウェアで転送されます。

### リセット解除後の動作

リセットベク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外バスモードで使用される場合には、モード端子で内部ベクタモードを指定することを推奨します。

リセットベク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。



(注) 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。

図 4.3-1 リセットベクトルとモードデータの格納場所と格納先

### リセット入力によって初期化されないレジスタ

本製品には、パワーオンリセットでしか初期化されないレジスタがあります。表 4.3-1に各リセット要因によって初期化されないレジスタを示します。

表 4.3-1 リセット入力によって初期化されないレジスタ

リセットの種類		CKSCR					WTC	LPMCR	
		WS1	WS0	MCS	CS1	CS0	WDCS	CG1	CG0
RSTXのみ使用 ソフトウェアリセット		×	×	×	×	×	×	×	×
ウォッチドッグリセット		×	×		×	×	×		
パワーオンリセット									
ハードウェア スタンバイ	メインモード	×	×		×	×	×		
	サブモード								

WS1,WS0 :メインクロック発振安定待ち時間設定

MCS :マシクロック設定(“0”=PLLクロック,“1”=メインクロック)

CS1,CS0 :PLLクロック逡倍設定

WDCS :ウォッチドッグ入力クロック“0”=時計タイマ,“1”タイムベースタイマ)

: 初期化する

× : 初期化しない

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

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

上記の様なケースが考えられる場合には、HSTX+RSTX HSTXとRSTXをショート)として使用することを推奨します。

表 4.3-2に、HSTX+RSTX を行った場合のリセット入力による初期化されないレジスタについて示します。表 4.3-2に示すように、リセット解除時の動作状態が、HSTX+RSTXのリセット入力と、RSTXのみの入力とで異なるので、注意が必要です。

表 4.3-2 リセット入力によって初期化されないレジスタ

リセットの種類		CKSCR					WTC	LPMCR	
		WS1	WS0	MCS	CS1	CS0	WDCS	CG1	CG0
HSTX+RSTX	メインモード	×	×		×	×	×		
	サブモード*								

\*: サブモード移行中期間も含む。

: 初期化する

× : 初期化しない

## [リセット入力による動作遷移]

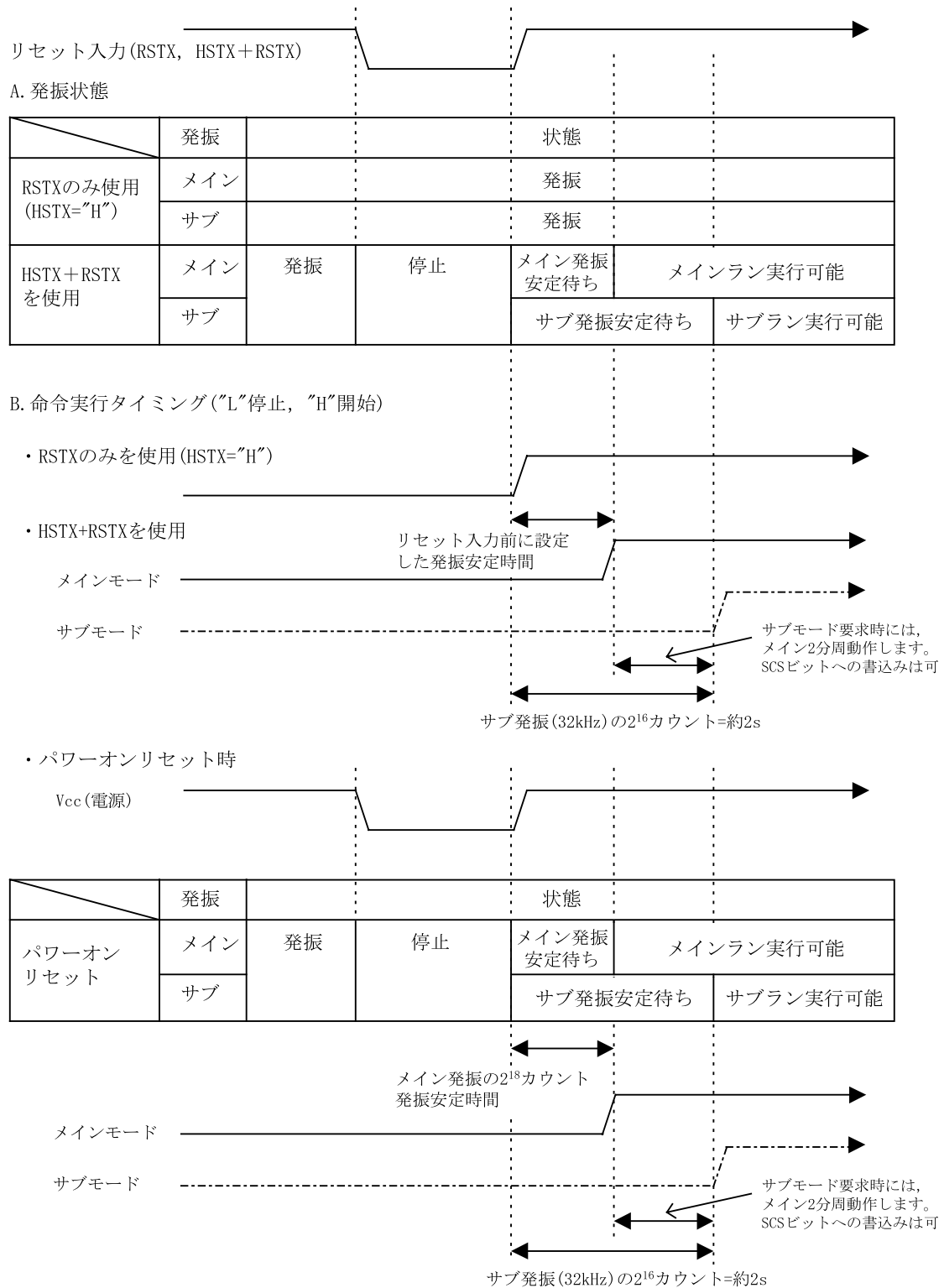


図 4.3-2 リセット入力による動作遷移





## 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 低消費電力モード制御レジスタ(LPMCR)
- 5.3 クロック選択レジスタ(CKSCR)
- 5.4 低消費電力制御回路の動作
- 5.5 CPU間欠動作機能
- 5.6 メインクロック発振安定待ち時間設定
- 5.7 マシンクロックの切り替え
- 5.8 状態遷移
- 5.9 低消費電力モードの状態遷移図

# 5.1 低消費電力制御回路の概要

動作モードとして、PLLクロックモード・PLLスリープモード・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・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・時計モード・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となります。

## 低消費電力回路の概要

### メインクロックモード / メインスリープモード

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

### PLLスリープモード / メインスリープモード

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

### 時計モード

タイムベースタイマのみを動作させるモードです。

### ストップモード / ハードウェアスタンバイモード

発振を停止させるモードであり、もっとも低消費電力でデータを保持できます。CPU間欠動作機能は、レジスタ、内蔵メモリ、内蔵リソースおよび外部バスアクセスを行うときに、CPUに供給するクロックを間欠動作させる機能であり、内蔵リソースに高速クロックを供給したままCPU実行速度を下げることで、低消費電力で処理が行えます。

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

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

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	—	LPSCR
リード/ライト ⇒	(W)	(W)	(R/W)	(W)	(—)	(R/W)	(R/W)	(—)	
初期値 ⇒	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(—)	

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 5.1-1 低消費電力回路のレジスタ

## < 注意事項 >

クロックモードを切り替えた場合、切り替えが完了するまでは、他のクロックモードおよび低消費電力モードへの切り替えを行わないでください。切り替えの完了は、クロック選択レジスタ（CKSCR）のMCMビットおよびSCMビットを参照して確認してください。切り替えが完了する前に、他のクロックモードおよび低消費電力モードへの切り替えを行った場合、切り替わらない場合があります。

## 低消費電力回路のブロックダイアグラム

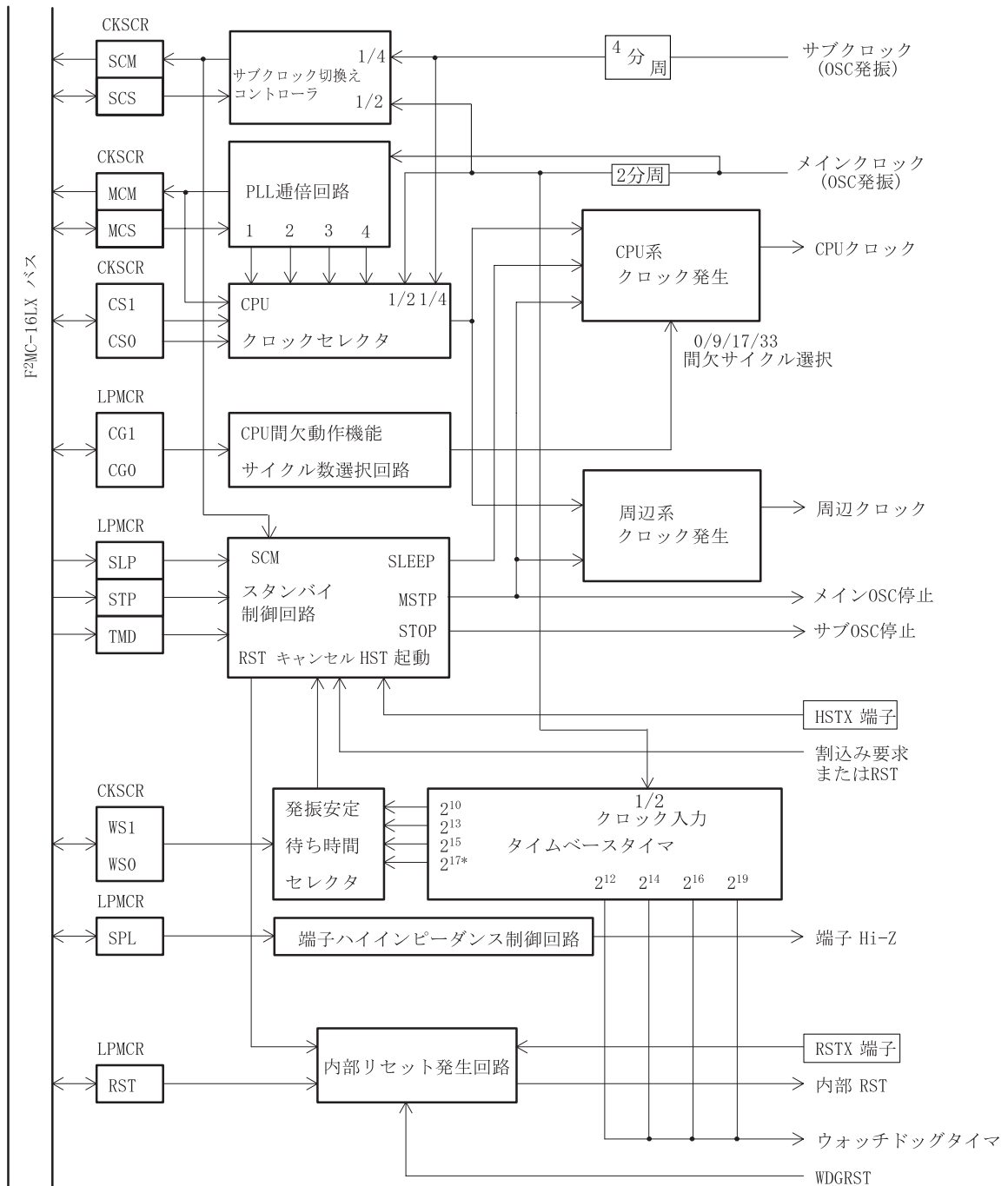
\*電源投入時は、 $2^{18}$ となります。

図 5.1-2 低消費電力制御回路とクロック生成部のブロックダイアグラム

## 5.2 低消費電力モード制御レジスタ(LPMCR)

低消費電力モード制御レジスタ(LPMCR)は、クロック選択レジスタと共に、消費電力の削減に関する各種動作モードを設定します。

低消費電力モード制御レジスタ(LPMCR)

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	予約	LPMCR
リード/ライト	⇒	(W)	(W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(-)	
初期値	⇒	(0)	(0)	(0)	(1)	(1)	(0)	(0)	(0)	

図 5.2-1 低消費電力制御回路のレジスタ

### 低消費電力モード制御レジスタアクセスに関する注意事項

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

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

表 5.2-1 低消費電力モードへ遷移する場合に使用する命令一覧

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

#### 【ビット7】STP

STPビットは、"1"を書き込むことにより疑似時計モード(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"の書き込みでは何の操作もしません。リセット、時計解除またはストップ解除で"0"にクリアされます。

書込みのみ可能なビットです。読み出し値は、常に"0"です。

#### 【ビット6】SLP

SLPビットは、"1"を書き込むことによりスリープモードに遷移します。"0"の書き込みでは何の操作もしません。リセット、スリープモード解除またはストップモード解除で"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、疑似時計モードまたはストップモードに遷移します。

書込みのみ可能なビットです。読み出し値は、常に"0"です。

## 【ビット5】SPL

SPLビットは，"0"の時，時計モードまたはストップモード時の外部端子のレベルを保持します。"1"の時，時計モードまたは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

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

リード時は"1"が，読み出されます。

## 【ビット3】TMD

2系統品：

TMDビットは，"0"を書き込むことにより時計モードに遷移します。"1"の書き込みでは何の操作もしません。このビットは，リセット，時計モード解除またはストップモード解除で，"0"にクリアされます。

書き込みのみ可能なビットです。読み出し値は，常に"1"です。

1系統品：

常に"1"を書き込んでください。

## 【ビット2，1】CG1,CG0

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

読み書き可能なビットです。

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

表 5.2-2 低消費電力モード制御レジスタ（CG1,CG0）の設定

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

## 【ビット0】予約

必ず"0"を設定してください。

### 5.3 クロック選択レジスタ(CKSCR)

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

クロック選択レジスタ(CKSCR)

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 クロック選択レジスタ(CKSCR)

#### 【ビット15】SCM

2系統品：

SCMは、マシナクロックとして、メインクロックまたはサブクロックのどちらを選択しているかを示すビットです。このビットが"0"の時は、サブクロックが選択されていることを示しています。"1"の時は、メインクロックが選択されていることを示しています。

SCS=0で、かつSCM=1ならば、メインクロックからサブクロックへの移行期間中であることを表します。また、SCS=1で、かつSCM=0ならば、サブクロックからメインクロックへの移行期間中であることを表します。

1系統品：

リード値は常に"1"になります。

#### 【ビット14】MCM

MCMは、マシナクロックとして、メインクロックまたはPLLクロックのどちらが選択されているかを示すビットです。このビットが"0"の時は、PLLクロックが選択されていることを示しています。"1"の時は、メインクロックが選択されていることを示しています。

MCS=0で、かつMCM=1ならば、PLLクロック発振安定待ち期間中であることを示します。なお、PLLクロックの発振安定待ち期間は、 $2^{13}$ メインクロックサイクル固定です。

#### 【ビット13, 12】WS1, WS0

WS1, WS0ビットでは、ストップモード、ハードウェアスタンバイモード解除時のメインクロックの発振安定待ち時間を設定します。パワーオンリセットにより"11"に初期化され、その他のリセット要因によるリセットでは初期化されません。

読み書き可能なビットです。

表 5.3-1 クロック選択レジスタ (WS1, WS0ビット) の設定

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

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

## 【ビット11】SCS

## 2系統品：

SCSは、マシナクロックとして、メインクロックまたはサブクロックのどちらかを選択するビットです。本ビットに"0"を書き込むことによりサブクロックを選択します。"1"のときに"0"を書き込むとサブクロックの同期をとって(約130 $\mu$ s)サブクロックモードに切り替わります。"1"を書き込むことにより、メインクロックを選択します。"0"のときに"1"を書き込むと、メインクロックの発振安定待ち時間を発生し、自動的にタイムベースタイマがクリアされます。SCSおよびMCSがともに"0"のときには、SCSが優先され、サブクロックが選択されます。

## 1系統品：

常に"1"を書き込んでください。

## 【ビット10】MCS

MCSは、マシナクロックとして、メインクロックまたはPLLクロックのどちらか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"の時に、"0"を書き込むと、PLLクロックの発振安定待ち期間を発生するために、自動的にタイムベースタイマがクリアされ、さらにタイムベースタイマ制御レジスタのTBOFビットもクリアされます。なお、PLLクロックの発振安定待ち期間は、 $2^{13}$ メインクロックサイクル固定です。(原発振4MHz時、発振待ち時間は約2msになります。)

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

## &lt; 注意事項 &gt;

電源投入時、またはストップモードからの解除時はサブクロックの発振安定待ち時間(約2s)が発生します。したがって、そ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替えた場合は発振安定待ち時間が発生します。

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



## 【ビット9,8】CS1,CS0

CS1,CS0は、PLLクロックの通倍率を選択するビットです。外部端子、RSTビットおよびウォッチドッグによるリセットとハードウェアスタンバイ解除では初期化されません。パワーオンによるリセットにより"00"に初期化されます。

MCSビットが"0"の時には書き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 5.3-2 クロック選択レジスタ（CS1,CS0ビット）の設定

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

## &lt; 注意事項 &gt;

動作電圧が5Vの時、OSC原発振は3MHz～16MHzの発振が可能ですが、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越える通倍率を指定した場合には、正常に動作しません。例えば、OSC原発振が16MHzで発振している場合には、1通倍のみ指定が可能です。

また、VCO発振の最低動作周波数は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。

内部動作クロック周波数 [MHz]

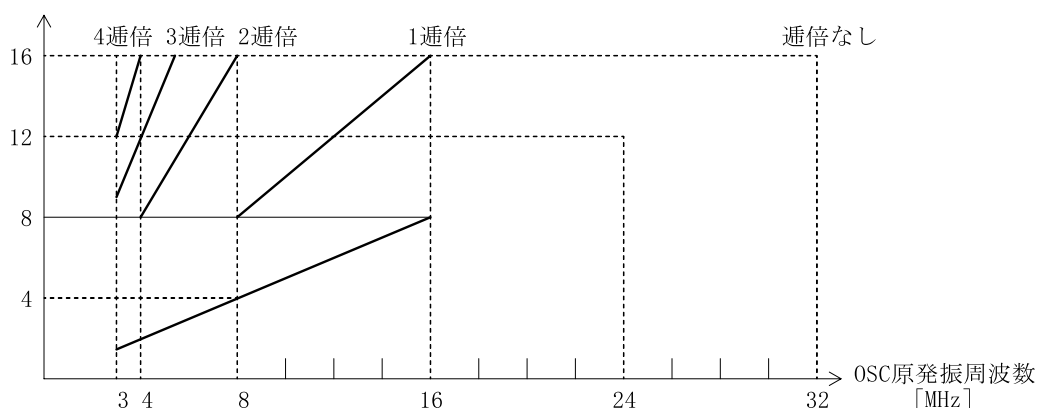


図 5.3-2 OSC原発振周波数と内部動作クロック周波数の関係

## 5.4 低消費電力制御回路の動作

表 5.4-1に、低消費電力モードの動作状態を示します。

低消費電力制御回路の動作

表 5.4-1 低消費電力モード動作状態（2系統品）

	遷移条件	サブ発振	メイン発振	マシンクロック	CPU	周辺	端子	解除方法
サブクロック <sup>*1</sup>	SCS=0 MCS=x	動作	停止	動作	動作	動作	動作	外部リセット
サブスリープ <sup>*1</sup>	SCS=0 MCS=x SLP=1	動作	停止	動作	停止	動作	動作	外部リセット 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	外部リセット 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	外部リセット 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	外部リセット 割込み <sup>*2</sup>
擬似時計 (SPL=1)	SCS=1 MCS=0 STP=1	動作	動作	停止	停止	停止	Hi-z	外部リセット 割込み <sup>*2</sup>
時計 (SPL=0)	SCS=x MCS=x TMD=0	動作	停止	停止	停止	停止	保持	外部リセット 割込み <sup>*3</sup>
時計 (SPL=1)	SCS=x MCS=x TMD=0	動作	停止	停止	停止	停止	Hi-z	外部リセット 割込み <sup>*3</sup>
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	外部リセット 割込み <sup>*4</sup>
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	外部リセット 割込み <sup>*4</sup>
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

\*1：1系統品では設定しないでください。

\*2：時計プリスケアラ，タイムベースタイマ，および外部割込み

\*3：時計プリスケアラ，および外部割込み

\*4：外部割込み

表 5.4-2 低消費電力モード動作状態（1系統品）

	遷移条件	サブ発振	メイン 発振	マシン クロック	CPU	周辺	端子	解除方法
メイン スリープ	SCS=1 MCS=1 SLP=1	-	動作	動作	停止	動作	動作	外部 リセット 割込み
POLL スリープ	SCS=1 MCS=0 SLP=1	-	動作	動作	停止	動作	動作	外部 リセット 割込み
擬似時計 （SPL=0）	SCS=1 MCS=0 STP=1	-	動作	停止	停止	停止	保持	外部 リセット 割込み <sup>*5</sup>
擬似時計 （SPL=1）	SCS=1 MCS=0 STP=1	-	動作	停止	停止	停止	Hi-z	外部 リセット 割込み <sup>*5</sup>
ストップ （SPL=0）	SCS=1 MCS=1 STP=1	-	停止	停止	停止	停止	保持	外部 リセット 割込み <sup>*6</sup>
ストップ （SPL=1）	SCS=1 MCS=1 STP=1	-	停止	停止	停止	停止	Hi-z	外部 リセット 割込み <sup>*6</sup>
ハードウェア スタンバイ	HSTX=L	-	停止	停止	停止	停止	Hi-z	HSTX=H

\*5：タイムベースタイマ，および外部割込み

\*6：外部割込み

### 5.4.1 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中の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。

SLPビットに"1"を書き込んだ時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は割り込みを受け付けない状態では次の命令の実行、受け付ける状態であれば即座に割り込み処理ルーチンへの分岐を行います。

スリープモードでは、アキュムレータなどの専用レジスタと内部RAMの内容を保持します。また、スリープモード中でも、外部バスホールド機能は動作し、ホールド要求があると、ホールド状態になります。

#### スリープモードの解除

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

##### リセットによる復帰

リセットによりメインクロックモードに初期化されます。

##### 割込みによる復帰

スリープモード中に、周辺回路などから割込みレベルが7より強い割込み要求が発生すると、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になります。コンディションコードレジスタ(CCR)のエフラグ、インタラプトレベルマスクレジスタ(ILM)、および割込み制御レジスタ(ICR)の設定によって、割込みが受け付けられる場合は、CPUは割込み処理を実行します。割込みが受け付けられない場合は、スリープモードを指定した命令の次の命令から処理を続行します。

## 5.4.2 擬似時計モード

擬似時計モードとは、ソース発振(メインおよびサブ)、時計タイマ、およびタイムベースタイマ以外のすべてのクロックを止めるモードであり、チップのほぼ全機能が停止します。

また、低消費電力モード制御レジスタ(LPMCRR)中のSLPビットを使用して、擬似時計モード中のI/O端子を、直前の状態に保持するか、ハイインピーダンス状態にするかを制御できます。

### 擬似時計モードへの遷移

クロック選択レジスタ(CKSCRR)のSCSビットに"1"、MCSビットに"0"、および低消費電力モード制御レジスタ(LPMCRR)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。

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と、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、RAMの内容は勿論、アキュムレータとその他の専用レジスタの内容を保持します。

### 擬似時計モードの解除

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

外部リセットによる復帰

外部リセットによりメインクロックモードに初期化されます。

割込みによる復帰

擬似時計モード中に時計プリスケラ、タイムベースタイマ、および外部割込みから割込みレベルが7より強い割込み要求が発生すると、低消費電力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、通常の割込み処理同じ扱いとなります。コンディションコードレジスタ(CCR)のエフラグ、インタラプトレベルマスクレジスタ(ILM)、および割込み制御レジスタ(ICR)の設定により割込みが受け付けられる場合は、割込み処理を実行します。割込みが受け付けられない場合は、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。

### 5.4.3 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すべてのクロックを止めるモードであり、チップのほぼ全機能が停止します。

また、低消費電力モード制御レジスタ(LPMCR)中のSPLビットを使用して、時計モード中のI/O端子を、直前の状態に保持するか、ハイインピーダンス状態にするかを制御できます。

なお、時計モードは1系統品では使用できません。

#### 時計モードへの遷移

低消費電力モード制御レジスタ中の 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設定します。

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と、スタンバイ制御回路は時計モードに遷移しません。

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

なお、時計モード中は、外部バスホールド機能は停止し、ホールド要求が入力されても、その要求を受け付けません。時計モードへの遷移中に、ホールド要求が入力された場合、バスをHi-Zにした状態で、HAKX信号が"L"にならないことがあります。

#### 時計モードの解除

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

##### リセットによる復帰

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

##### 割込みによる復帰

時計モード中に時間プリスケアラ、および外部割込みから割込みレベル7より強い割込み要求が発生すると、低消費電力制御回路は時計モードを解除し、即座にサブクロックモードに移行します。サブクロックモードに移行後は、通常の割込み処理と同じ扱いとなります。コンディションコードレジスタ(CCR)のエフラグ、インタラプトレベルマスクレジスタ(ILM)、および割込み制御レジスタ(ICR)を設定して割込みが受け付けられる場合は、割込み処理を実行します。割込みが受け付けられない場合は、時計モードに入る前の次の命令から処理を続行します。

#### < 注意事項 >

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

### 5.4.4 ストップモード

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

また、低消費電力モード制御レジスタ(LPMCR)中のSPLビットを使用して、ストップモード中のI/O端子を、直前の状態に保持するか、ハイインピーダンス状態にするかを制御できます。

#### ストップモードへの遷移

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

STP ビットに"1"を書き込んだときに割込み要求が発生していると、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

低消費電力制御回路は、リセット入力または割込みの発生によりストップモードを解除します。ストップモードからの復帰の場合、動作クロックの発振が停止しているため、低消費電力制御回路ははまず発振安定待ち時間に移行し、その後ストップモードを解除します。

##### リセットによる復帰

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

##### 割込みによる復帰

ストップモード中に外部割込みからレベルが7より強い割込み要求が発生すると、低消費電力制御回路はストップモードを解除します。ストップモードの解除後は、クロック選択レジスタ(CKSCR)の発振安定待ち時間選択ビット(WS1, WS0)で指定されたメインクロックの発振安定待ち時間を経過した後で、通常の割込み処理と同じ扱いとなります。コンディションコードレジスタ(CCR)のエフラグ、インタラプトレベルマスクレジスタ(ILM)、および割込み制御レジスタ(ICR)を設定して割込みが受け付けられる場合は、割込み処理を実行します。割込みが受け付けられない場合は、ストップモードに入る前の次の命令から処理を続行します。

#### < 注意事項 >

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

ただし、ストップモードへの移行と外部バスホールド要求の受け付けが同時であった場合、次の命令を実行する前に割込み処理に移行することがあります。

### 5.4.5 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、リセットを含む他の状態に関係なくHSTX端子が"L"レベルの間、発振を停止させI/O端子を全てハイインピーダンス状態にするモードです。

---

#### ハードウェアスタンバイモードへの遷移

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設定することができます。

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

#### ハードウェアスタンバイモードの解除

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

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



## 5.5 CPU間欠動作機能

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

### CPU間欠動作機能

CPU間欠動作機能では、内蔵リソースに高速クロックを供給したままCPUの実行速度を下げることにより、低消費電力で処理が行えます。CG1,0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、内蔵リソースおよび外部バスアクセスを行う回数に、一時停止サイクル数を掛けた補正値を通常の実行時間に加えることで、求めることができます。



図 5.5-1 CPU間欠動作

## 5.6 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)中の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発振安定待ち時間を選択します。X0,X1端子に接続する発振回路、発振素子の種類や特性に応じて発振安定待ち時間を選択してください。

---

### メインクロック発振安定待ち時間設定

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

## 5.7 マシニングロックの切り替え

ここでは、マシニングロックの切り替え方法と初期化について説明します。  
なお、サブクロックは1系統品では使用できません。

### メインクロック/PLLクロックの切り替え

CKSCRレジスタ中のMCSビットへの書き込みにより、メインクロックとPLLクロックの切り替えを行います。

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( $2^{13}$ マシニングロック)経過後に、マシニングロックはメインクロックからPLLクロックに切り替わります。

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時点(1~8 PLLクロック後)で、マシニングロックはPLLクロックからメインクロックに切り替わります。

MCSビットを書き換えても、即座にマシニングロックの切り替えが行われないので、マシニングロックに依存するリソースを操作する場合には、MCMビットを常に参照してマシニングロックの切り替えが行われたことを確認した後、リソースの操作を行ってください。

### メインクロック/サブクロックの切り替え

2系統品は、CKSCRレジスタ中のSCSビットへの書き込みにより、メインクロックとサブクロックの切り替えを行います。

SCSビットを "1" から "0" に書き換えた場合には、サブクロックに同期をとって(約130  $\mu$ s)、マシニングロックはメインクロックからサブクロックに切り替わります。

SCSビットを "0" から "1" に書き換えた場合には、メインクロックの発振安定待ち時間経過後に、マシニングロックはサブクロックからメインクロックに切り替わります。

SCSビットを書き換えても、即座にマシニングロックの切り替えが行われないので、マシニングロックに依存するリソースを操作する場合には、SCMビットを常に参照してマシニングロックの切り替えが行われたことを確認した後、リソースの操作を行ってください。

### < 注意事項 >

クロックモードを切り替えた場合、切り替えが完了するまでは、他のクロックモードおよび低消費電力モードへの切り替えを行わないでください。切り替えの完了は、クロック選択レジスタ(CKSCR)のMCMビットおよびSCMビットを参照して確認してください。切り替えが完了する前に、他のクロックモードおよび低消費電力モードへの切り替えを行った場合、切り替わらない場合があります。

### マシニングロックの初期化

MCSビット、SCSビットは、外部端子、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。

図 5.7-1、図 5.7-2にクロック選択の状態図を示します。

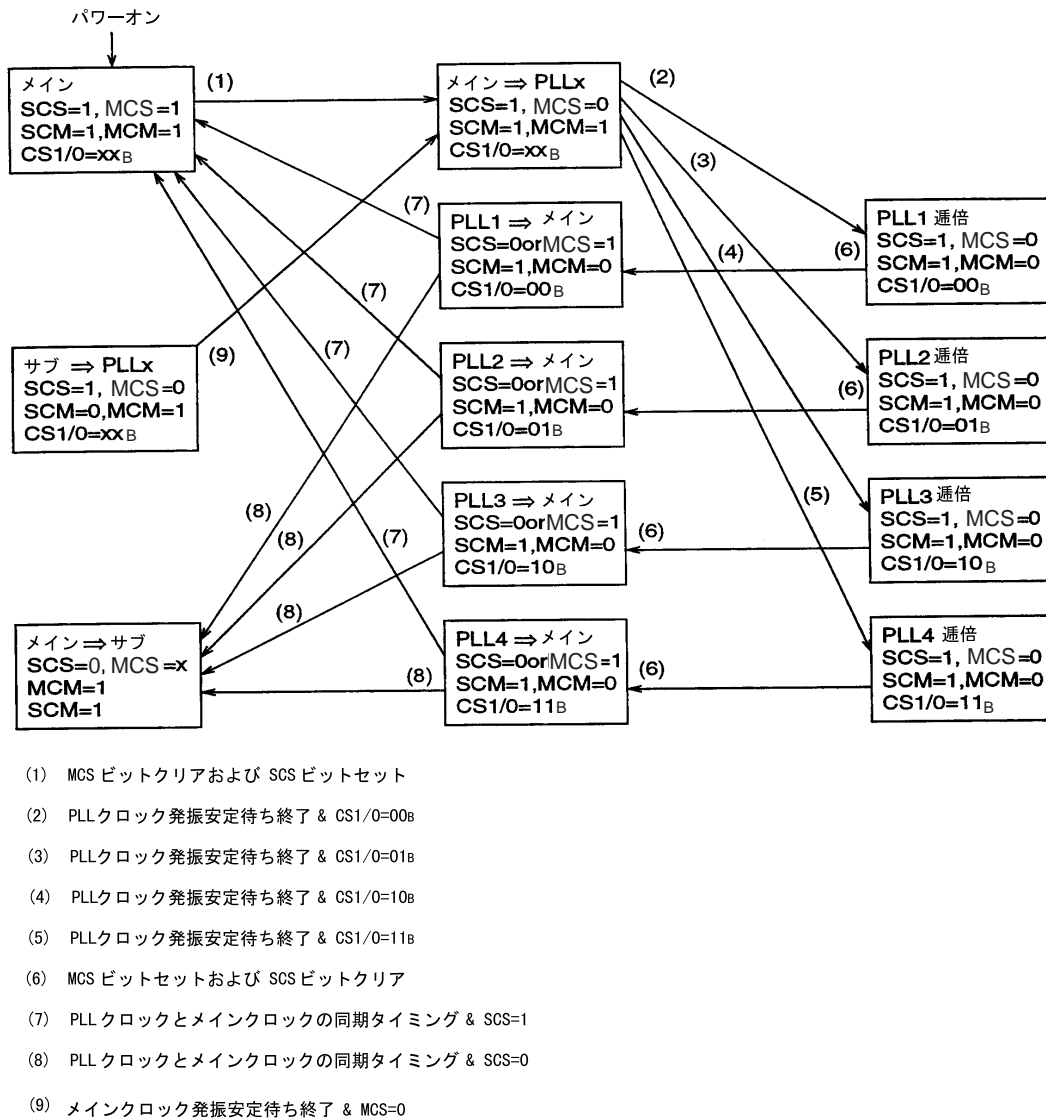


図 5.7-1 クロック選択の状態遷移図1 (2系統品 その1)

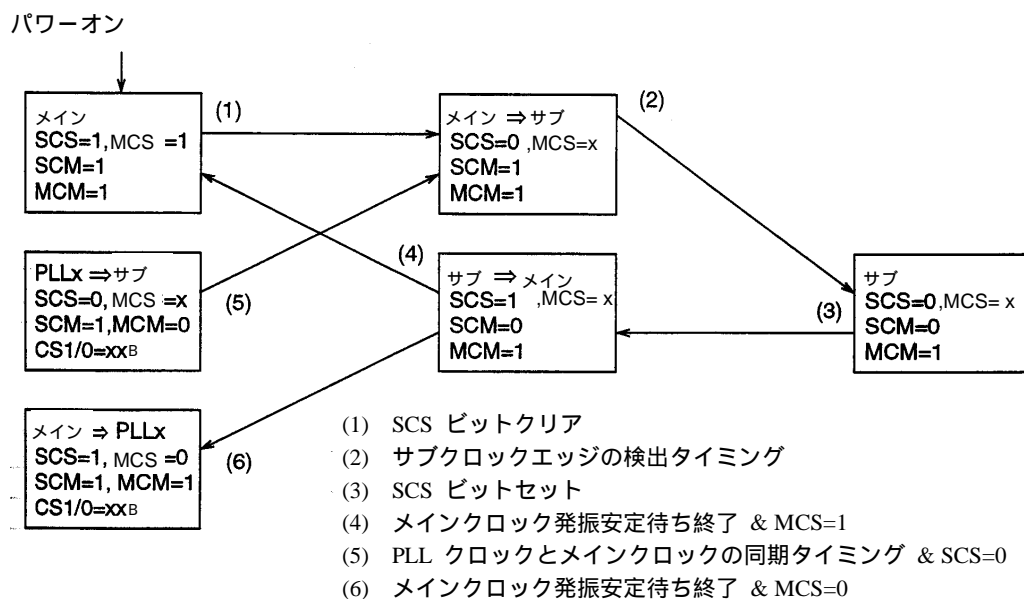


図 5.7-2 クロック選択の状態遷移図2 (2系統品 その2)

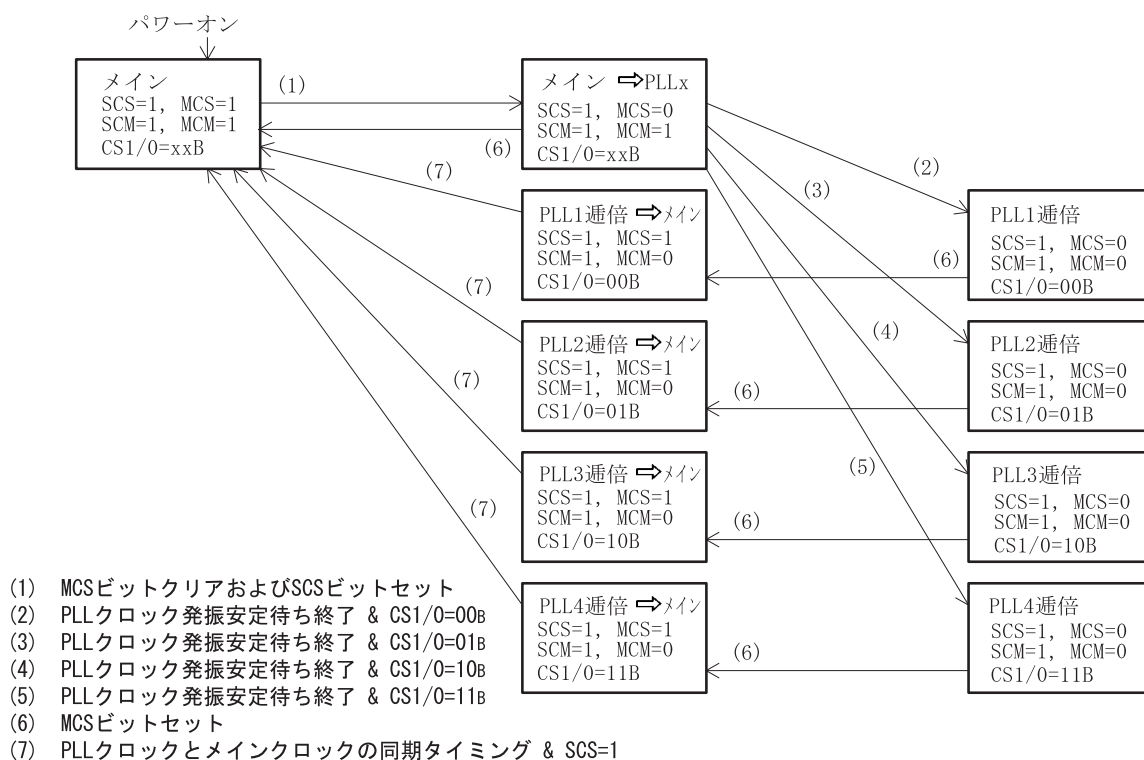


図 5.7-3 クロック選択の状態遷移図3 (1系統品)

## 5.8 状態遷移

表 5.8-1に、状態遷移条件の一覧を示します。

状態遷移

表 5.8-1 2系統品の状態遷移条件一覧表（続く）

遷移前の状態	遷移条件		遷移後の状態
パワーオン	01	メイン発振安定待ち時間終了	メインモード
メイン発振安定	05	メイン発振安定待ち時間終了	メインモード
メインモード	06	SCS=0 書込	MS 遷移モード
	07	SCS=1・MCS=0 書込	MP 遷移モード
	31	TMD=1・STP=0・SLP=1 書込	メインスリープ
	32	TMD=0 書込	メイン時計遷移
	33	TMD=1・STP=1 書込	メインストップ
PLL モード	21	SCS=0 書込	PS 遷移モード
	20	SCS=1・MCS=1 書込	PM 遷移モード
	59	TMD=1・STP=0・SLP=1 書込	PLL スリープ
	58	TMD=0 書込	PLL 時計遷移 P
	57	TMD=1・STP=1 書込	擬似時計遷移
サブモード	10	SCS=1・MCS=1 書込	SM 遷移モード
	12	SCS=1・MCS=1 書込	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	サブスリープ
	43	TMD=0 書込	サブ時計
	44	TMD=1・STP=1 書込	サブストップ
PM 遷移モード	13	PLL メイン切り替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	PM 遷移スリープ
	39	TMD=0 書込 & PLL メイン切り替えタイミング待ち終了	メイン時計遷移
	40	TMD=1・STP=1 書込 & PLL メイン切り替えタイミング待ち終了	メインストップ
SM 遷移モード	02	メイン発振安定待ち時間終了	メインモード
	03	リセット起動 or 割込	メイン発振安定
	04	SCS=0 書込	サブモード
	27	TMD=1・STP=0・SLP=1 書込	SM 遷移スリープ
	28	TMD=0 & メイン発振安定待ち時間終了	メイン時計
	29	TMD=1・STP=1 書込 & メイン発振安定待ち時間終了	メインストップ
MP 遷移モード	16	PLL 発振安定待ち時間終了	PLL モード
	14	SCS=1・MCS=1 書込	メインモード
	15	SCS=0 書込	MS 遷移モード
	68	TMD=1・STP=0・SLP=1 書込	MP 遷移スリープ
	70	TMD=0 書込	PLL 時計遷移 M
	69	TMD=1・STP=1 書込	擬似時計モード
SP 遷移モード	17	メイン発振安定待ち時間終了	MP 遷移モード
	18	MCS=1 書込	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	SP 遷移スリープ
	76	TMD=0 書込	PLL 時計
	78	TMD=1・STP=1 書込 & メイン発振安定待ち時間終了	擬似時計モード

表 5.8-1 2系統品の状態遷移条件一覧表 ( 続き )

遷移前の状態	遷移条件		遷移後の状態
MS 遷移モード	09	メイン サブクロック切り替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	MS 遷移スリープ
	52	TMD=0 書込 & メイン サブ切り替え待ち終了	サブ時計
	53	TMD=1・STP=1 書込 & メイン サブ切り替え待ち終了	サブモード
PS 遷移モード	23	PLL メインクロック切り替えタイミング待ち終了	MS 遷移モード
	22	SCS=1 書込	PM 遷移モード
	56	TMD=1・STP=0・SLP=1 書込	PS 遷移スリープ
メインスリープ	26	割込 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時間終了	メインスリープ
	25	割込 or リセット起動	SM 遷移モード
PM 遷移スリープ	34	PLL メインクロック切り替えタイミング待ち終了	メインスリープ
	35	割込 or リセット起動	PM 遷移モード
PLL スリープ	63	割込 or リセット起動	PLL モード
MP 遷移スリープ	66	PLL 発振安定待ち時間終了	PLL スリープ
	67	割込 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時間終了	MP 遷移スリープ
	74	割込 or リセット起動	SP 遷移モード
サブスリープ	46	割込 or リセット起動	サブモード
MS 遷移スリープ	49	メイン サブクロック切り替えタイミング待ち終了	サブスリープ
	50	割込 or リセット起動	MS 遷移モード
PS遷移スリープ	54	PLL メインクロック切り替えタイミング待ち終了	MS 遷移スリープ
	55	割込 or リセット起動	PS 遷移モード
メイン時計	30	割込 or リセット起動	SM 遷移モード
メイン時計遷移	36	メイン サブクロック切り替えタイミング待ち終了	メイン時計
	37	割込 or リセット起動	メインモード
PLL 時計	77	割込 or リセット起動	SP 遷移モード
PLL時計遷移 M	72	メイン サブクロック切り替えタイミング待ち終了	PLL 時計
	71	割込 or リセット起動	MP 遷移モード
PLL 時計遷移 P	65	PLL メインクロック切り替えタイミング待ち終了	PLL時計遷移 M
	64	割込 or リセット起動	PLL モード
サブ時計	47	割込 or リセット起動	サブモード
メインストップ	41	割込 or リセット起動	メイン発振安定
擬似時計	62	割込 or リセット起動	MP 遷移モード
擬似時計遷移	61	PLL メインクロック切り替えタイミング待ち終了	擬似時計モード
	60	割込 or リセット起動	PLL モード
サブストップ	48	割込	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時間終了	サブモード
	80	リセット起動	メイン発振安定

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 5.8-2 1系統品の状態遷移条件一覧表

遷移前の状態	遷移条件		遷移後の状態
パワーオン	01	メイン発振安定待ち時間終了	メインモード
メイン発振安定	05	メイン発振安定待ち時間終了	メインモード
メインモード	07	SCS=1・MCS=0 書込	MP 遷移モード
	31	TMD=1・STP=0・SLP=1 書込	メインスリープ
	33	TMD=1・STP=1 書込	メインストップ
PLL モード	20	SCS=1・MCS=1 書込	PM 遷移モード
	59	TMD=1・STP=0・SLP=1 書込	PLL スリープ
	57	TMD=1・STP=1 書込	擬似時計遷移
PM 遷移モード	13	PLL メイン切り替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	PM 遷移スリープ
	40	TMD=1・STP=1 書込 & PLL メイン切り替えタイミング待ち終了	メインストップ
MP 遷移モード	16	PLL 発振安定待ち時間終了	PLL モード
	14	SCS=1・MCS=1書込	メインモード
	68	TMD=1・STP=0・SLP=1 書込	MP 遷移スリープ
	69	TMD=1・STP=1 書込	擬似時計モード
メインスリープ	26	割込 or リセット起動	メインモード
PM 遷移スリープ	34	PLL メインクロック切り替えタイミング待ち終了	メインスリープ
	35	割込 or リセット起動	PM 遷移モード
PLL スリープ	63	割込 or リセット起動	PLL モード
MP 遷移スリープ	66	PLL 発振安定待ち時間終了	PLL スリープ
	67	割込 or リセット起動	MP 遷移モード
メインストップ	41	割込 or リセット起動	メイン発振安定
擬似時計	62	割込 or リセット起動	MP 遷移モード
擬似時計遷移	61	PLL メインクロック切り替えタイミング待ち終了	擬似時計モード
	60	割込 or リセット起動	PLL モード

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)

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)

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)

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)



## 5.9 低消費電力モードの状態遷移図

---

図 5.9-1 ~ 図 5.9-4に低消費電力モードの状態遷移を示します。

---

低消費電力モードの状態遷移図（2系統品）

図 5.9-1 ~ 図 5.9-4の状態遷移図では、簡単にするために、同時に発生した事象を段階的に発生したように描写しています。しかし、実際には即座に状態の遷移は行われます。例えば、PLLクロックモードでMCS に"1"、SLPに"1"を同時に設定した場合、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、リセットが発生した場合、いったんサブモードに遷移した後に、メイン発振安定時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時間に遷移します。

### < 注意事項 >

---

クロックモードを切り替えた場合、切り替えが完了するまでは、他のクロックモードおよび低消費電力モードへの切り替えを行わないでください。切り替えの完了は、クロック選択レジスタ (CKSCR) のMCMビットおよびSCMビットを参照して確認してください。切り替えが完了する前に、他のクロックモードおよび低消費電力モードへの切り替えを行った場合、切り替わらない場合があります。

---

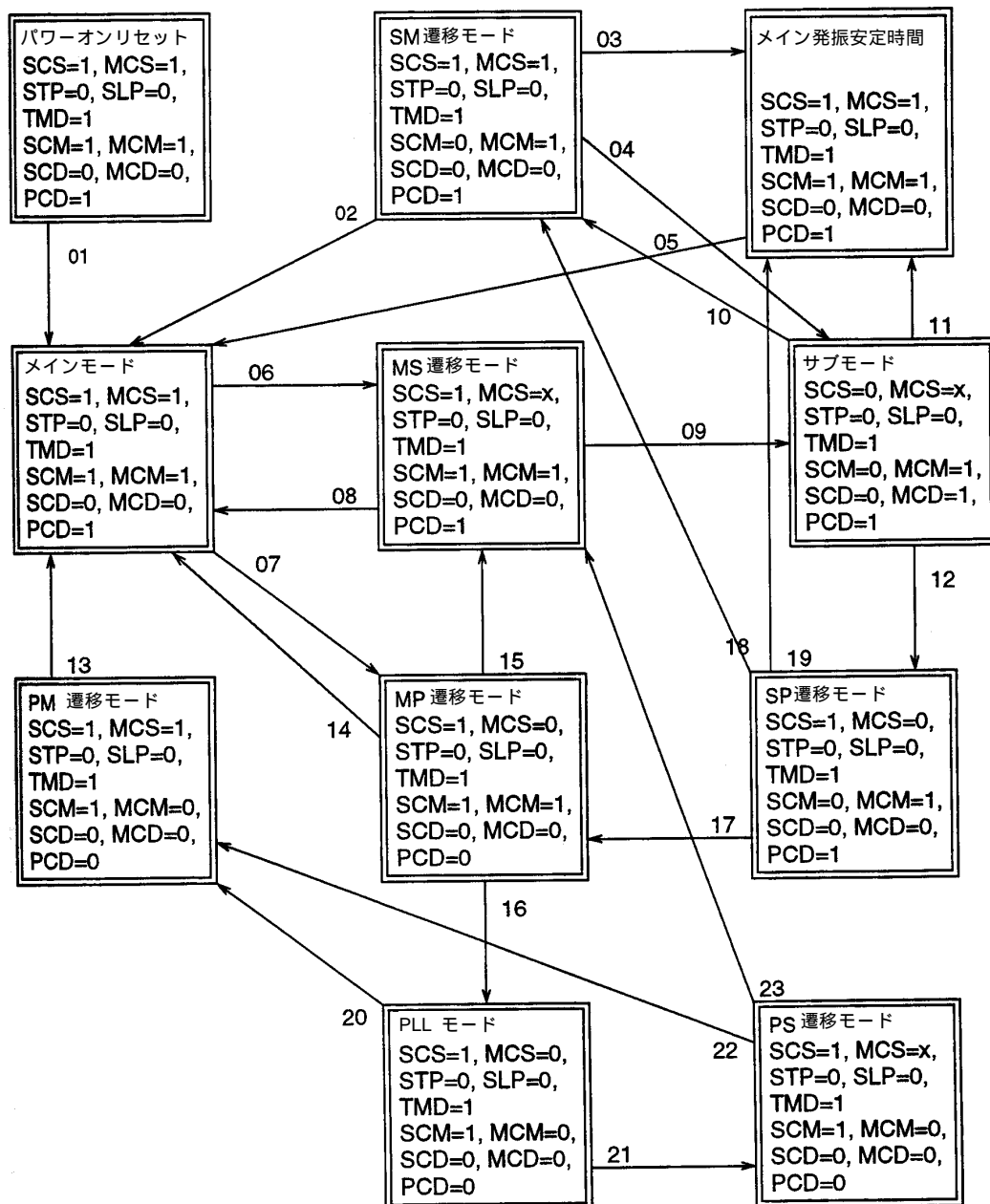


図 5.9-1 低消費電力モードの状態遷移図 (2系統品) 1

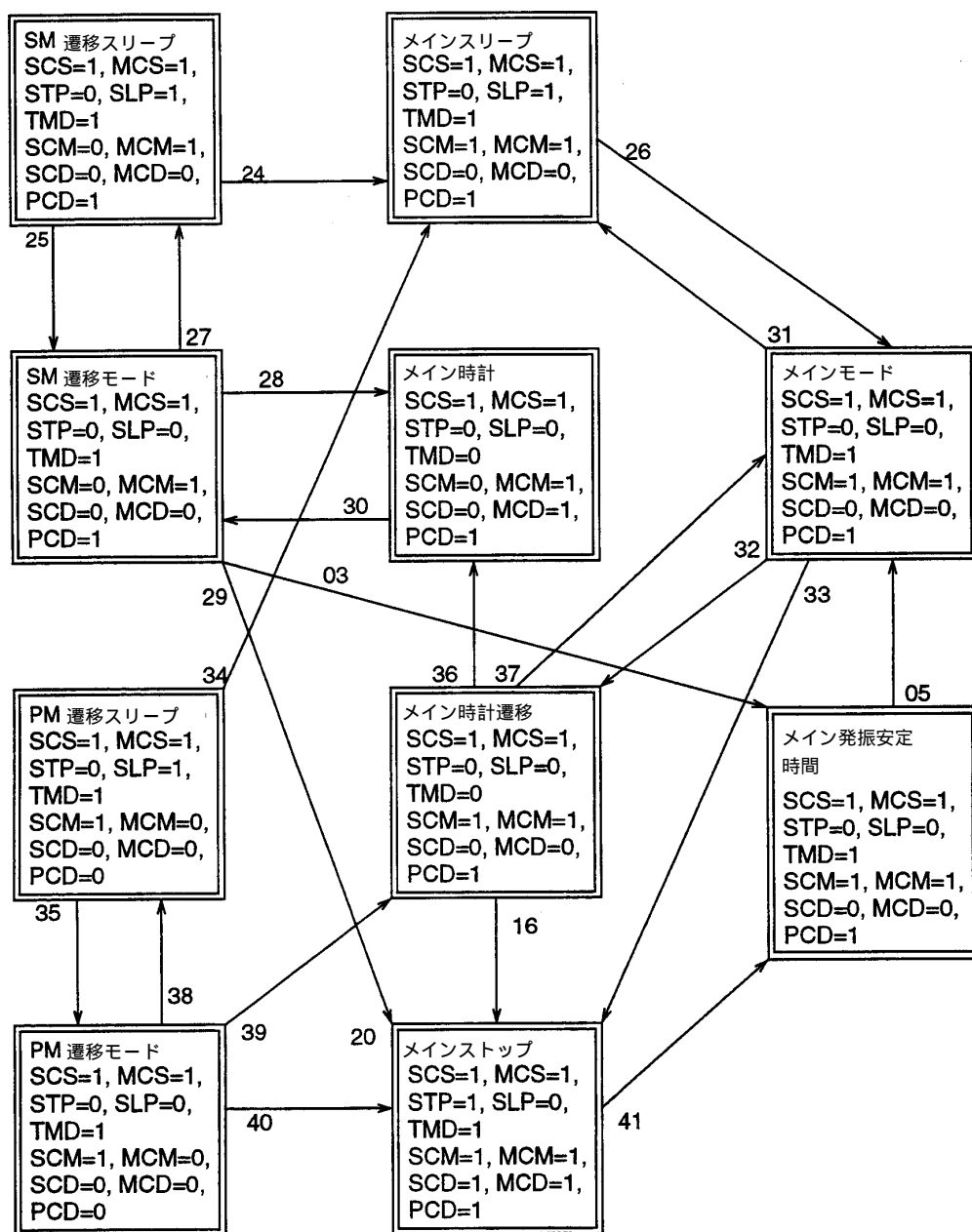


図 5.9-2 低消費電力モードの状態遷移図 (2系統品) 2

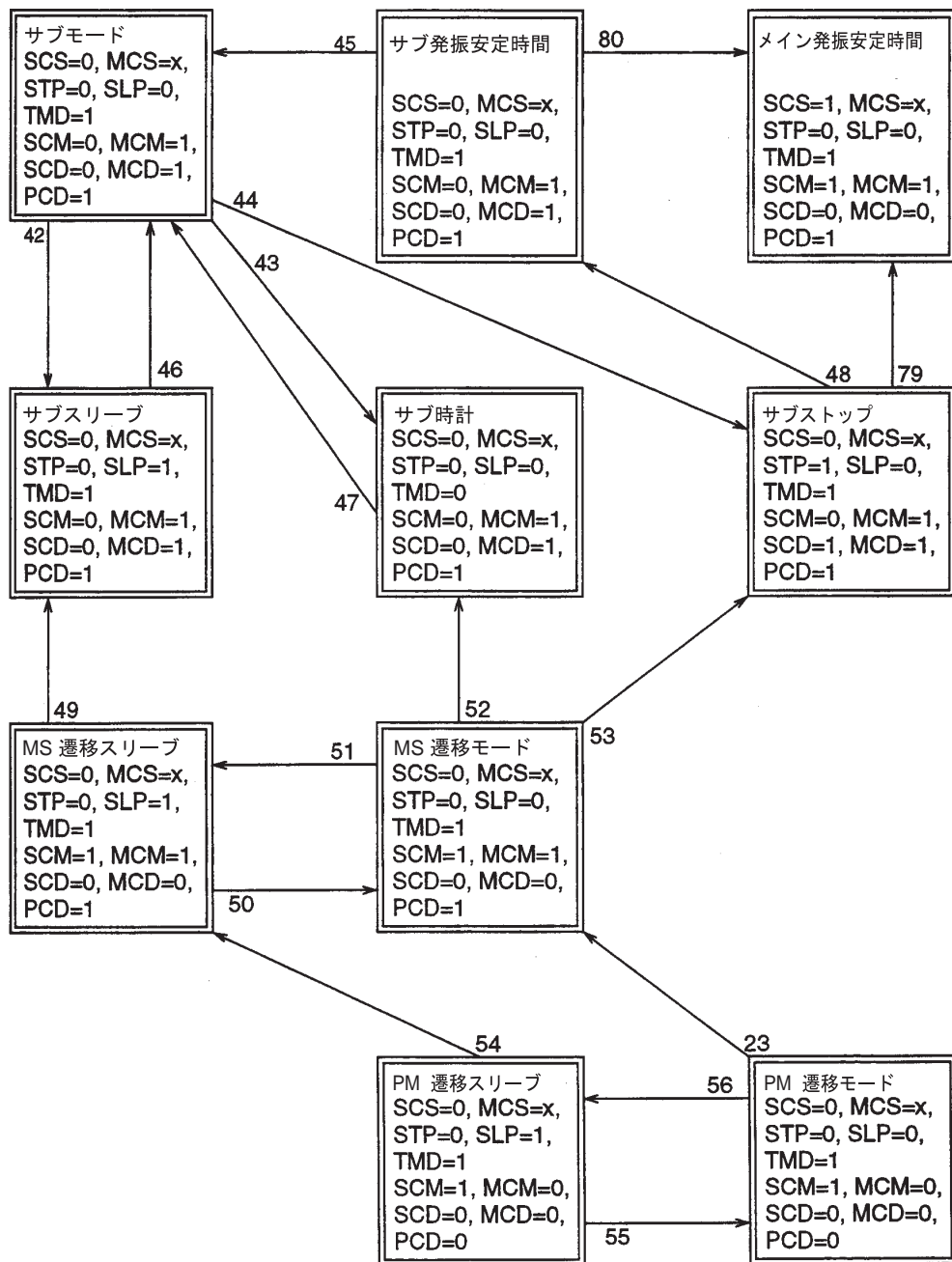


図 5.9-3 低消費電力モードの状態遷移図 (2系統品) 3

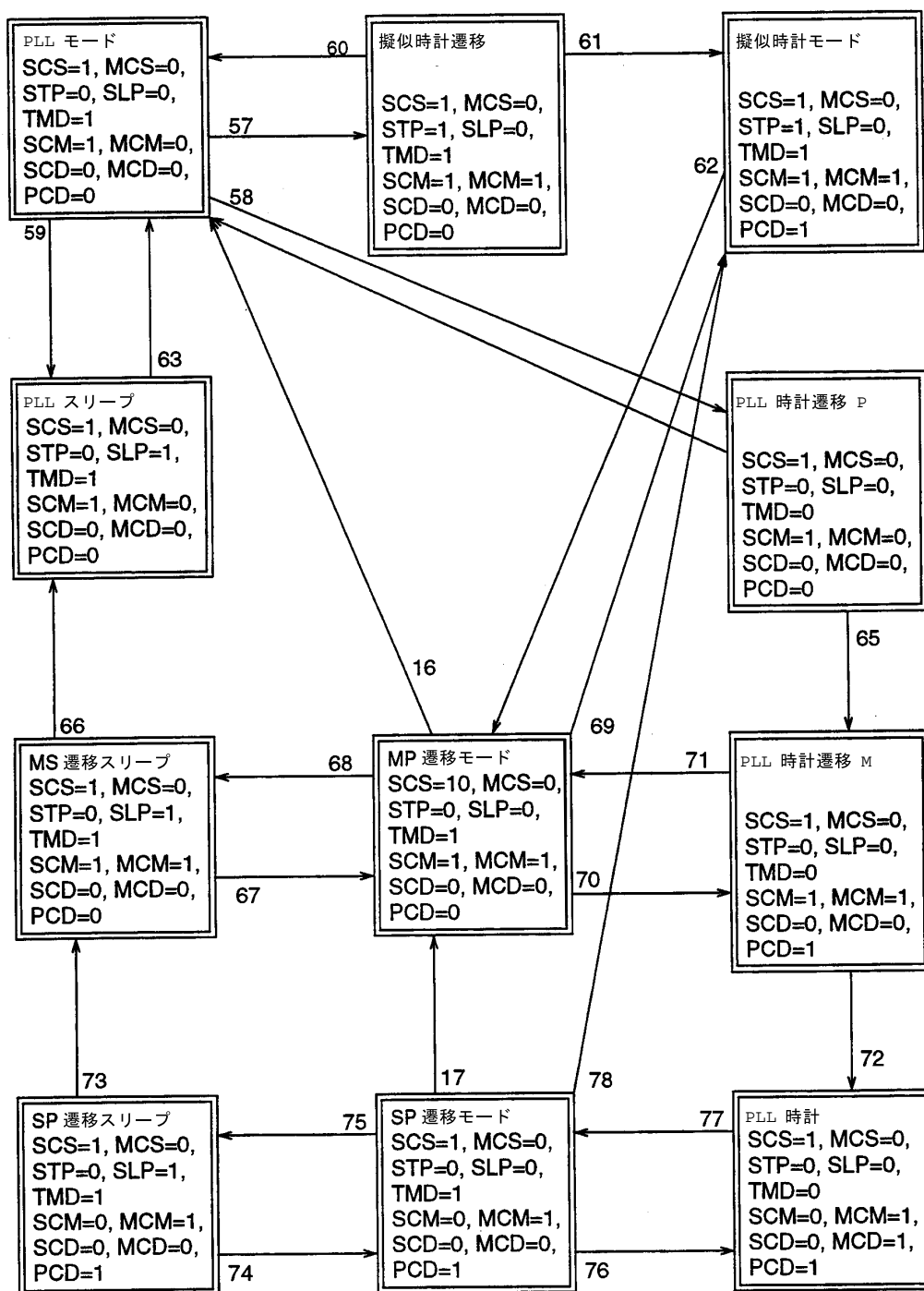


図 5.9-4 低消費電力モードの状態遷移図 (2系統品) 4

低消費電力モードの状態遷移図（1系統品）

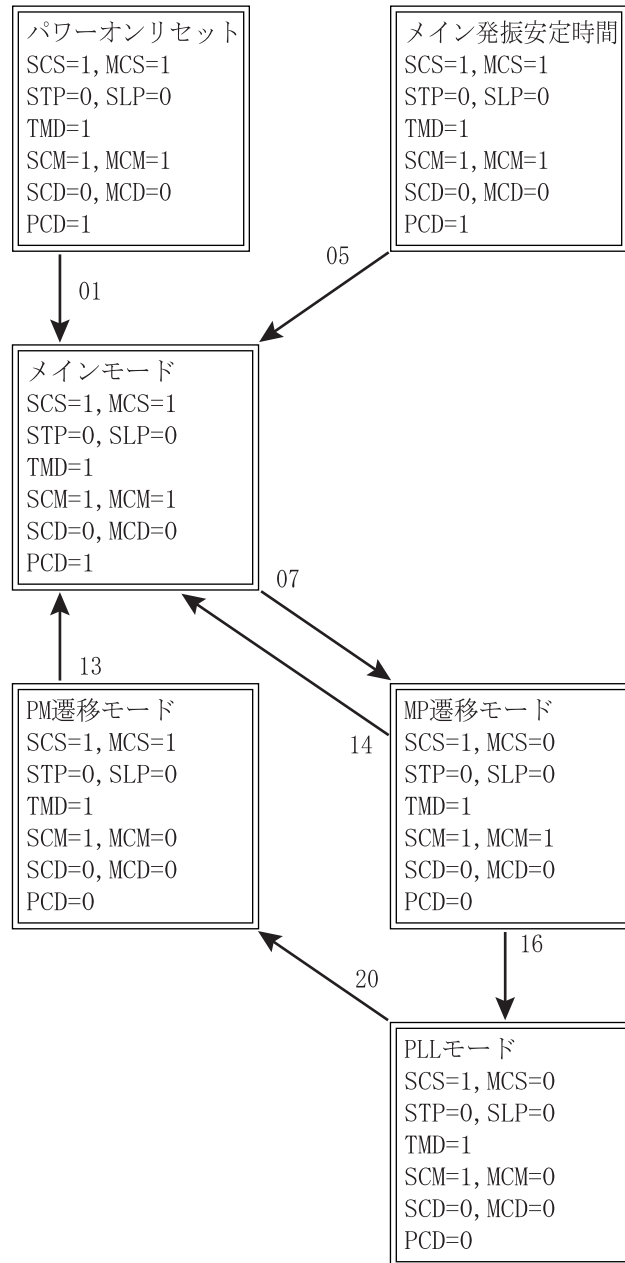


図 5.9-5 低消費電力モードの状態遷移図（1系統品）1

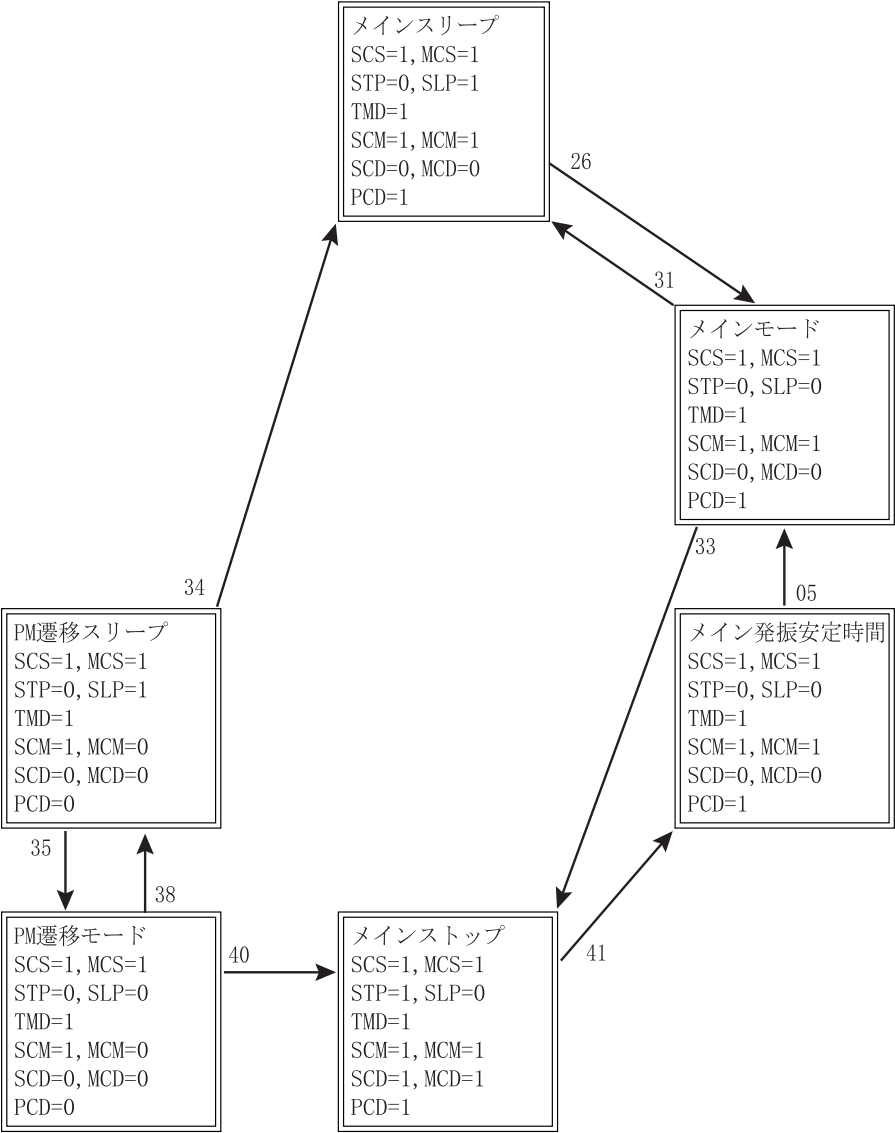


図 5.9-6 低消費電力モードの状態遷移図 (1系統品) 2

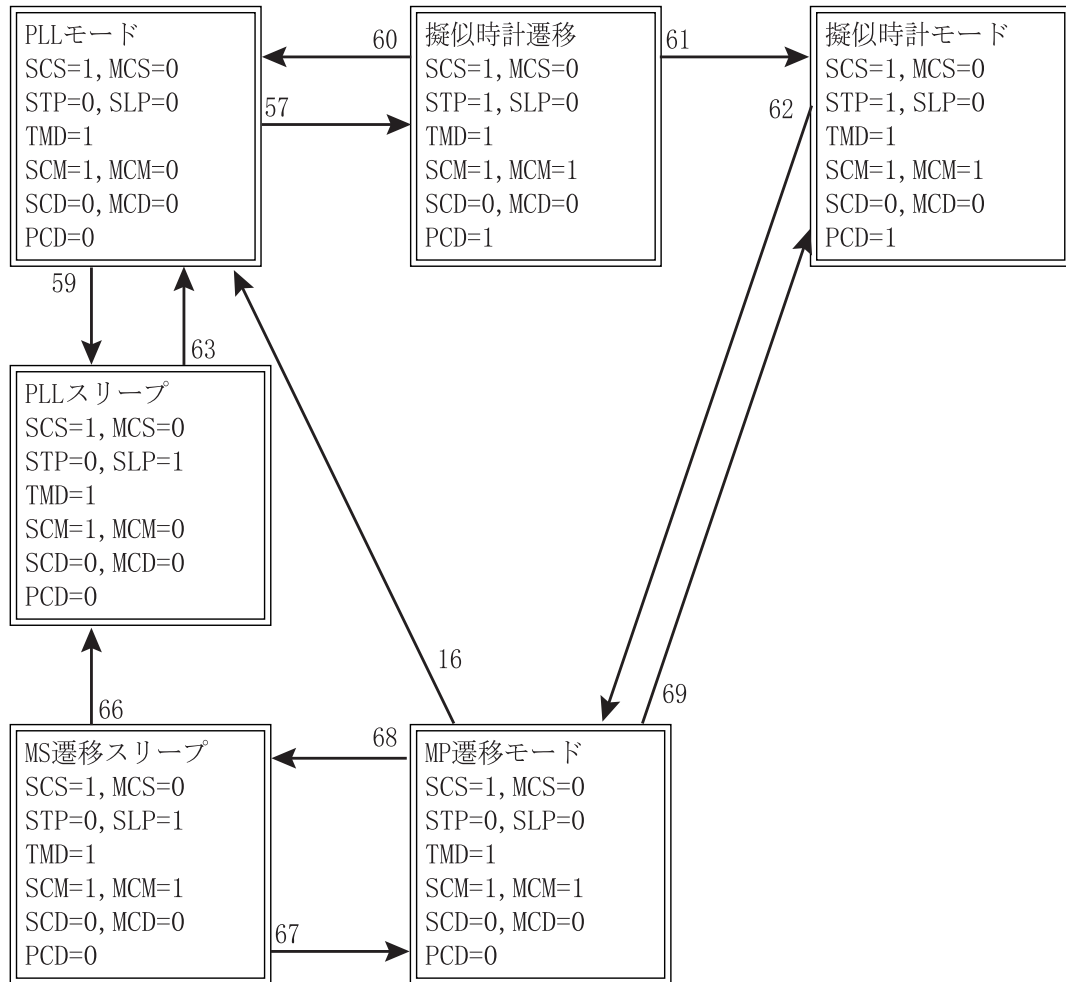


図 5.9-7 低消費電力モードの状態遷移図 (1系統品) 3

## &lt; 注意事項 &gt;

クロックモードを切り替えた場合、切り替えが完了するまでは、他のクロックモードおよび低消費電力モードへの切り替えを行わないでください。切り替えの完了は、クロック選択レジスタ (CKSCR) のMCMビットおよびSCMビットを参照して確認してください。切り替えが完了する前に、他のクロックモードおよび低消費電力モードへの切り替えを行った場合、切り替わらない場合があります。





## 第6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。

---

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

# 6.1 メモリアクセスモードの概要

F<sup>2</sup>MC-16LXでは、アクセス方式、アクセス領域について、各種のモードがあります。

メモリアクセスモードの概要

表 6.1-1 メモリアクセスモード

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

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、MDxモード設定用端子とモードデータ内のM1，M0ビットで指定します。動作モードを選択することで、通常動作の起動や、フラッシュメモリを書込むことができます。

バスモード

バスモードとは、内部ROMの動作と外部アクセス機能の動作を制御するモードを示すもので、モード設定用端子(MDx)とモードデータ内のMxビットの内容で指定します。モード設定用端子(MDx)は、リセットペクタおよびモードデータを読み出す時のバスモードを指定するもので、モードデータ内のMxビットは、通常動作時のバスモードを指定するものです。

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、MDxモード設定用端子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ビット長を指定します。

### 6.1.1 モード端子

MD2～0の3本の外部端子を組み合わせ、表 6.1-2の動作を指定ができます。

モード端子

表 6.1-2 モード端子とモード

モード端子設定			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
MD2	MD1	MD0				
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き込み	-	-	
1	1	1	フラッシュメモリモード	-	-	パラレルライタ使用時のモード

(注)・外部ベクトルモード0を選択した場合でも、バス制御信号選択レジスタのIOBS, LMBSビットの初期値は"0"にセットされていますので、0000C0H～0000FFH, 002000H～7FFFFFFH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信号選択レジスタのIOBS, LMBSビットに"1"を書き込んでください。外部ベクトルモード1の時には、HMBSビットが"0"にセットされ、16ビットバス幅でアクセスします。

・フラッシュメモリのシリアル書き込みは、モード端子の設定だけでは書き込みができません。他の端子設定も必要です。詳細は、FLASHシリアル書き込み接続例を参照してください。

### 6.1.2 モードデータ

モードデータは、主記憶FFFFDF<sub>H</sub>に置かれ、CPUの動作を制御するために使用します。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。モードレジスタの値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 <sub>H</sub>	M1	M0	予約	予約	S0	予約	予約	予約	

図 6.1-1 モードデータ構成

【ビット7,6】M1,M0(バスモード設定ビット)

M1,M0は、リセットシーケンス終了後の動作モードを指定するビットです。M1,M0ビットと、機能の関係は、表 6.1-3のようになっています。

表 6.1-3 M1,M0(バスモード設定ビット)の機能

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

【ビット3】S0(アクセスモード設定ビット)

S0は、リセットシーケンス終了後のバスモード、アクセスモードを指定するビットです。S0ビットと機能の関係は、表 6.1-4のようになっています。

表 6.1-4 S0(アクセスモード設定ビット)の機能

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

### 6.1.3 バスモード別メモリ空間

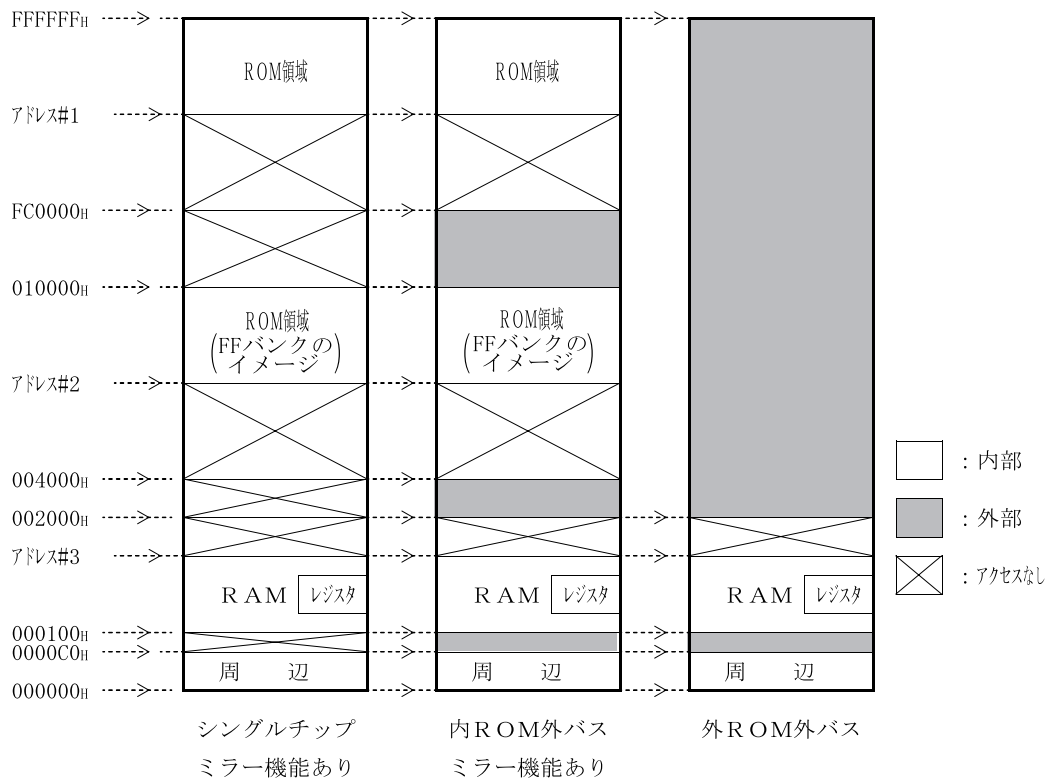
バスモードの指定による、アクセス領域と物理アドレスの対応を図 6.1-2に示します。

#### バスモード別メモリ空間

表 6.1-2に示すように、00バンクの上位にFFバンクのROMがイメージで見えるようになっていますが、これはCコンパイラのスモールモデルを有効に生かすためです。下位16ビットは同じになるようにしてありますので、ポインタの宣言においてfar指定をしなくともROM内のテーブルを参照することができます。

例えば、00C000<sub>H</sub>をアクセスした場合に、実際には、FFC000<sub>H</sub>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、48 K バイトを越しますので、00バンクのイメージにすべての領域を見せることができません。したがって、FF4000<sub>H</sub>～FFFFFF<sub>H</sub>のROMデータは004000<sub>H</sub>～00FFFF<sub>H</sub>のイメージに見えますので、ROMデータテーブルはFF4000<sub>H</sub>～FFFFFF<sub>H</sub>の領域に格納することを推奨いたします。

ROMのミラー機能なしを選択した場合は、「第23章 ROMミラー機能選択モジュール」を参照してください。



品種	アドレス #1	アドレス #2	アドレス #3
MB90583C/CA	FE0000 <sub>H</sub>	004000 <sub>H</sub>	001900 <sub>H</sub>
MB90F583C/CA	FE0000 <sub>H</sub>	004000 <sub>H</sub>	001900 <sub>H</sub>
MB90587C/CA	FF0000 <sub>H</sub>	004000 <sub>H</sub>	001100 <sub>H</sub>
MB90V580B	(FE0000 <sub>H</sub> )	004000 <sub>H</sub>	001900 <sub>H</sub>

図 6.1-2 各バスモードにおけるアクセス領域と物理アドレスの関係

バスモード別メモリ空間推奨設定例

表 6.1-5に，モード端子とモードデータの推奨設定例を示します。

表 6.1-5 モード端子とモードデータの推奨設定例

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

メモリアクセスモードに接続した外部端子は，各種モードにより入出力信号が変化します。

表 6.1-6 各種モードと関係する外部端子の動作

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

\*：アドレス上位ビットと，WRX，WRLX，WRHX，HRQ，HAKX，RDY，CLKは，機能選択にリポートとして使用することが可能です。

詳細は，「6.2 外部メモリアクセス（外部バス端子制御回路）」を参照ください。

## 6.2 外部メモリアクセス（外部バス端子制御回路）

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

### 外部メモリアクセス（外部バス端子制御回路）

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

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

### 外部メモリアクセス（外部バス端子制御回路）のブロックダイアグラム

図 6.2-1に，外部メモリアクセス（外部バス端子制御回路）のブロックダイアグラムを示します。

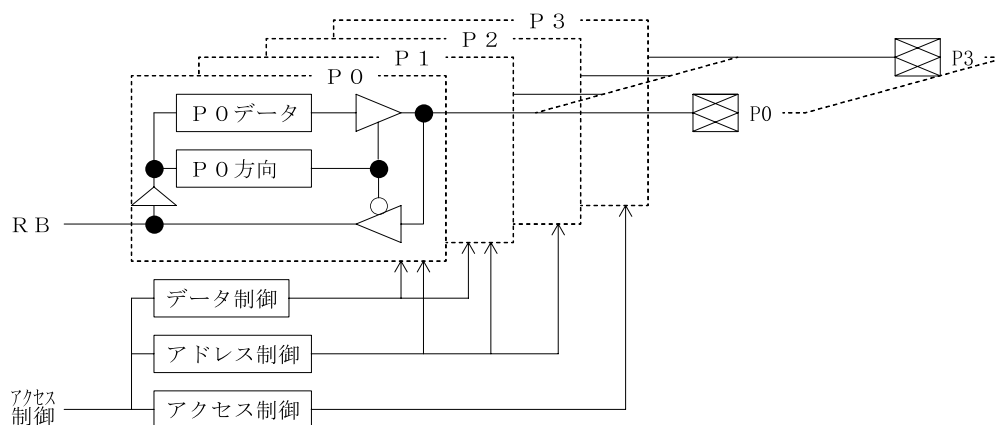


図 6.2-1 外部メモリアクセス（外部バス端子制御回路）のブロックダイアグラム



# 6.2.1 外部メモリアクセス（外部バス端子制御回路）のレジスタ

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

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

## 外部メモリアクセス（外部バス端子制御回路）のレジスタ

自動レディ機能選択レジスタ	15	14	13	12	11	10	9	8	↔ビットNo.
アドレス:0000A5 <sub>H</sub>	IOR1	IOR0	HMR1	HMR0	—	—	LMR1	LMR0	ARSR
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	↔ビットNo.
アドレス:0000A6 <sub>H</sub>	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)	
バス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:0000A7 <sub>H</sub>	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 6.2-2 外部メモリアクセス（外部バス端子制御回路）のレジスタ一覧

## 6.2.2 自動レディ機能選択レジスタ(ARSR)

自動レディ機能選択レジスタ(ARSR)は、外部アクセス時、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)

自動レディ機能選択レジスタ	15	14	13	12	11	10	9	8	←ビットNo.
アドレス:0000A5 <sub>H</sub>	IOR1	IOR0	HMR1	HMR0	—	—	LMR1	LMR0	ARSR
リード/ライト	(W)	(W)	(W)	(W)	(—)	(—)	(W)	(W)	
初期値	(0)	(0)	(1)	(1)	(—)	(—)	(0)	(0)	

図 6.2-3 自動レディ機能選択レジスタの構成

### 【ビット15，14】IOR1，IOR0

IOR1，IOR0ビットでは、0000C0<sub>H</sub>～0000FF<sub>H</sub>の領域に対する外部アクセスを行った時の自動ウェイト機能を指定します。IOR1，IOR0ビットの組合せで、表 6.2-1 に示すような設定になります。

表 6.2-1 IOR1，IOR0(自動ウェイト機能指定ビット)の機能

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

\*:初期値は"00B"です。

### 【ビット13，12】HMR1，HMR0

HMR1，HMR0では、800000<sub>H</sub>～FFFFFF<sub>H</sub>の領域に対する外部アクセスを行った時の自動ウェイト機能を指定します。HMR1，HMR0ビットの組合せで、表 6.2-2 に示すような設定になります。

表 6.2-2 HMR1，HMR0(自動ウェイト機能指定ビット)の機能

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

\*: 初期値は"11B"です。

### 【ビット9，8】LMR1，LMR0

LMR1，LMR0では、002000<sub>H</sub>～7FFFFFF<sub>H</sub>の領域に対する外部アクセスを行った時の自動ウェイト機能を指定します。LMR1，LMR0ビットの組合せで、表 6.2-3 に示すような設定になります。

表 6.2-3 LMR1,LMR0(自動ウェイト機能指定ビット)の機能

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

\*: 初期値は"00<sub>B</sub>"です。

### 6.2.3 外部アドレス出力制御レジスタ(HACR)

外部アドレス出力制御レジスタ(HACR)は、アドレス（A23～16）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図 6.2-4に示すように制御します。

外部アドレス出力制御レジスタ(HACR)

外部アドレス出力制御レジスタ	7	6	5	4	3	2	1	0	←ビットNo.
アドレス:0000A6 <sub>H</sub>	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 6.2-4 外部アドレス出力制御レジスタの構成

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

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

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

表 6.2-4 外部アドレス出力制御レジスタ(E16～23ビット)の機能

E16～23	機能
0	対応する端子はアドレス出力（AXX）になります。[初期値]
1	対応する端子はI/Oポート（PXX）になります。

### 6.2.4 バス制御信号選択レジスタ(ECSR)

バス制御信号選択レジスタ(ECSR)は、外部バスモード時のバス動作の制御機能を設定するレジスタです。デバイスがシングルチップモード時にはアクセスすることができません。その場合、本レジスタの値にかかわらず、全端子がI/Oポートとして機能します。バス制御信号選択レジスタのビットは、すべて書き込み専用で、読み出しでは"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:0000A7 <sub>H</sub>	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 6.2-5 バス制御信号選択レジスタの構成

【ビット15】CKE

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

表 6.2-5 CKE(外部クロック(CLK)出力制御ビット)の機能

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

【ビット14】RYE

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

表 6.2-6 RYE(外部レディ(RDY)入力制御ビット)の機能

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

【ビット13】HDE

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

表 6.2-7 RYE(ホールド関連端子の入出力許可指定ビット)の機能

HDE	機能
0	I/Oポート(P35, P34)動作(ホールド機能入出力禁止)[初期値]
1	ホールド要求(HRQ)入力/ホールドアクノリッジ(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<sub>H</sub>～0000FF<sub>H</sub>の領域に対する外部アクセスを行った時のバスサイズを指定します。本ビットの設定により，表 6.2-8に示すように制御します。

表 6.2-8 IOBS（バスサイズ指定ビット）

IOBS	機能
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

IOBSビットは，リセットで"0"になります。

## 【ビット11】HMBS

HMBSビットは，外部データバス16ビットモード時に，800000<sub>H</sub>～FFFFFF<sub>H</sub>の領域に対する外部アクセスを行った時のバスサイズを指定します。本ビットの設定により，表 6.2-9に示すように制御します。

表 6.2-9 HMBS(バスサイズ指定ビット)の機能

HMBS	機能
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10】WRE

WREビットは，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WRX端子）の出力を，表 6.2-10に示すように制御します。

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

表 6.2-10 WRE(外部ライト信号出力制御ビット)の機能

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

\*: WREビットは，リセットで"0"になります。

## 【ビット9】LMBS

LMBSビットは，外部データバス16ビットモード時に，002000<sub>H</sub>～7FFFFFF<sub>H</sub>の領域に対する外部アクセスを行った時のバスサイズを指定します。本ビットの設定により，表 6.2-11のように制御します。

表 6.2-11 LMBS(バスサイズ指定ビット)の機能

LMBS	機能
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

\*: LMBSビットは，リセットで"0"になります。

< 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，そのポートに対応するDDR3に，必ず"0"（入力モード）を書き込みます。
-

## 6.3 外部メモリアクセス制御信号の動作

外部メモリへのアクセスは、レディ機能を使用しない場合は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。

### 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。

なお、アドレス出力とALEのアサート出力のみを行い、RDX/WRX/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 6.3-1に、外部メモリアクセスのタイミングチャート（外部8ビットバスモード時）を、図 6.3-2に、外部メモリアクセスタイミングチャート（外部16ビットバスモード時）を示します。

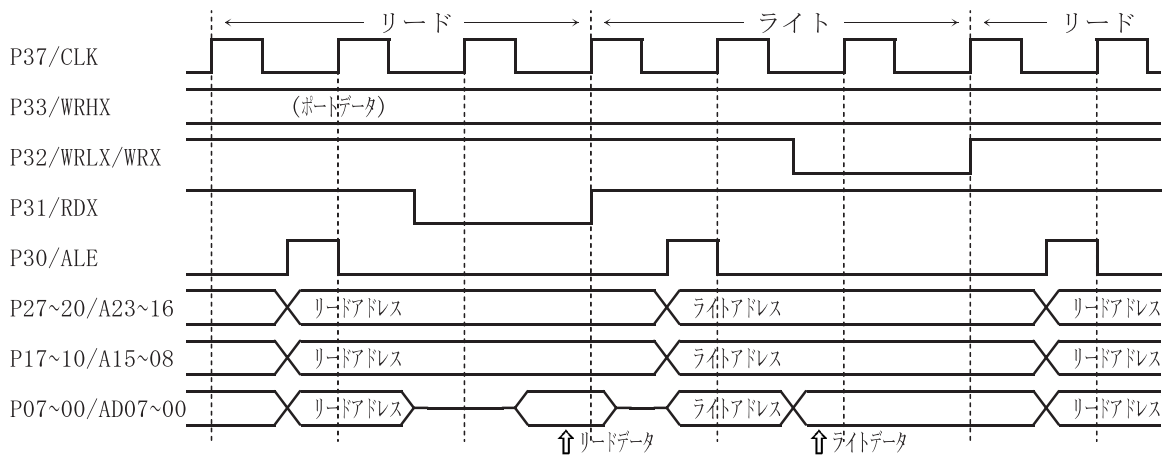
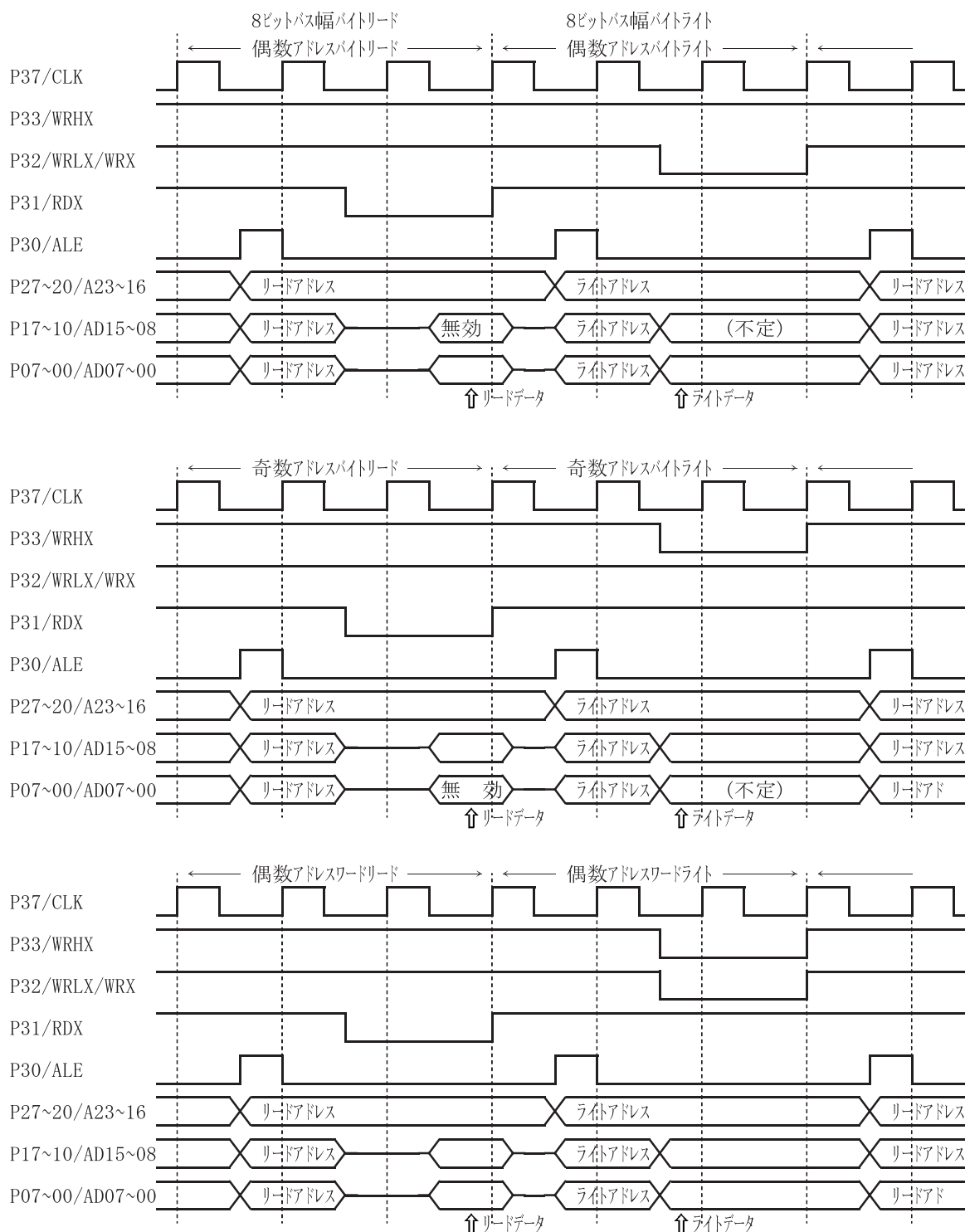


図 6.3-1 外部メモリアクセスのタイミングチャート（外部8ビットバスモード時）



## 第6章 メモリアクセスモード



常にワードで読み出すように外部回路を設定してください。  
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 6.3-2 外部メモリアクセスタイミングチャート (外部16ビットバスモード時)

### 6.3.1 レディ機能

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

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

#### レディ機能

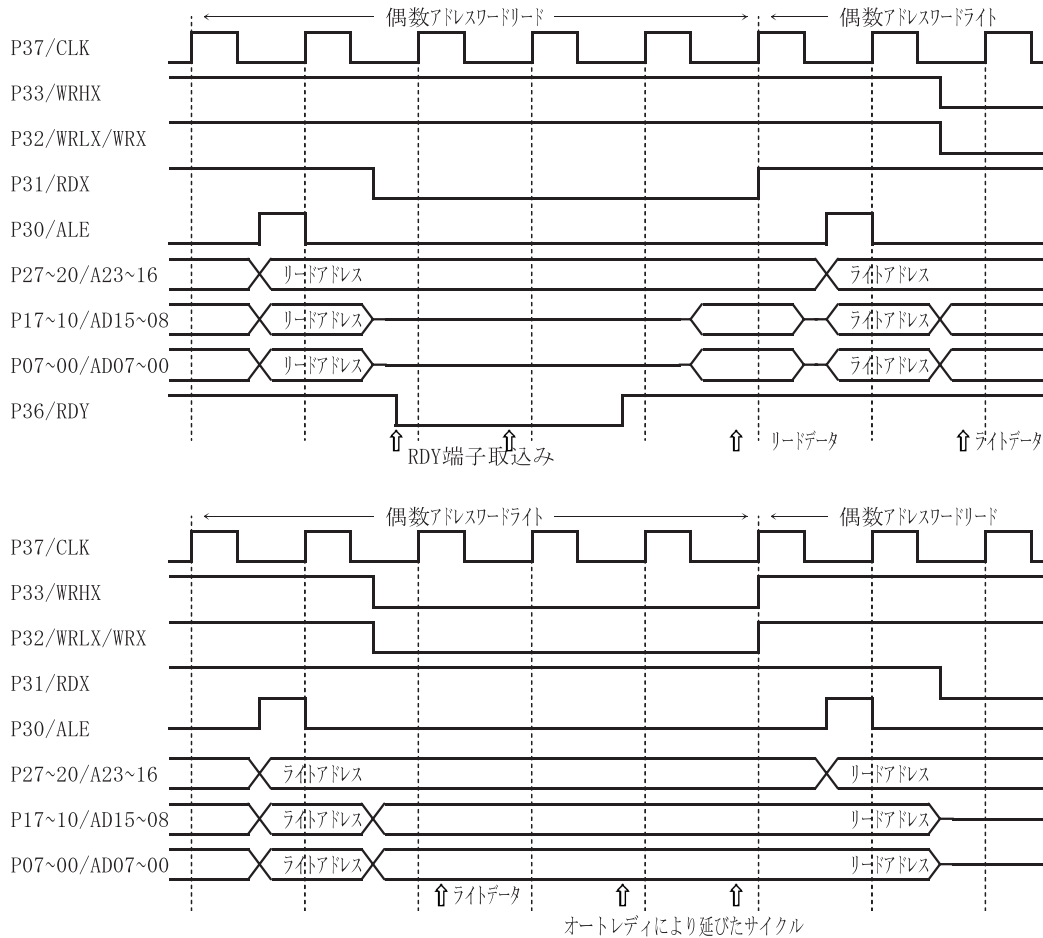


図 6.3-3 レディタイミングチャート

また, F<sup>2</sup>MC-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<sub>h</sub> ~ 7FFFFF<sub>h</sub>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時と, アドレス800000<sub>h</sub> ~ FFFFFFFF<sub>h</sub>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, F<sup>2</sup>MC-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<sub>h</sub> ~ 0000FF<sub>h</sub>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/IO0ビットの設定により起動します。

外部メモリ用，外部I/O用のオートレディどちらでもEPCR中のRYEビットが"1"に設定されている場合，上記オートレディ機能によるウェイトサイクル終了後にP36/RDY端子にLレベルが入力されているとそのままウェイトサイクルを継続します。

### 6.3.2 ホールド機能

バス制御信号選択レジスタ(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：P27/A23～P20/A16
- アドレス／データ入出力：P17/AD15～P00/AD00
- バス制御信号：P30/ALE、P31/RDX、P32/WRLX/WRX、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、ホールド要求入力を受け付けません。

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

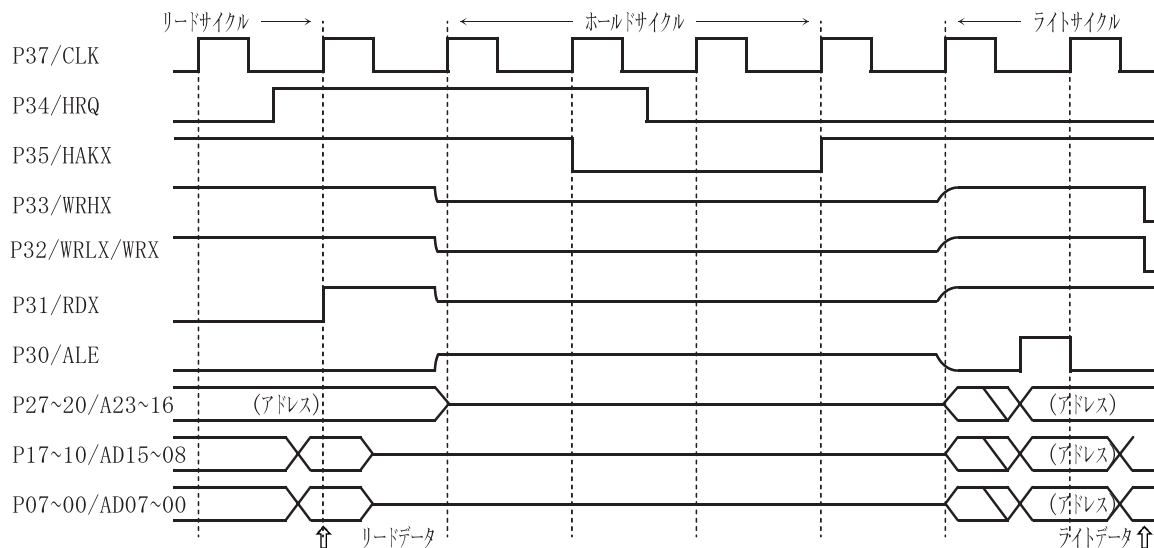


図 6.3-4 ホールドタイミング（外部バス16ビットモード時）



## 第7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。

---

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

## 7.1 I/Oポートの概要

それぞれのポートの各端子は、対応する周辺が端子を用いない設定となっている時、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、入力時にデータレジスタを読出した場合は、常に、端子のレベルによる値が読出され、出力時にデータレジスタを読出した場合は、データレジスタのラッチの値が読出されます。これは、リードモディファイライトのリードでも同じです。

### I/Oポートの概要

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

ポート0～Aは、入出力ポートで方向レジスタの値が"0"のとき入力、"1"のとき出力になります。

MB90580Cシリーズでは、ポート0～3は外部バス端子と共用になっています。したがって、外部バスモードで使用する際はこれらのポートは使用が制限されます。

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

## 7.2 I/Oポートブロックダイアグラム

図 7.2-1 ~ 図 7.2-4に, I/Oポートのブロックダイアグラムを示します。

I/Oポートブロックダイアグラム

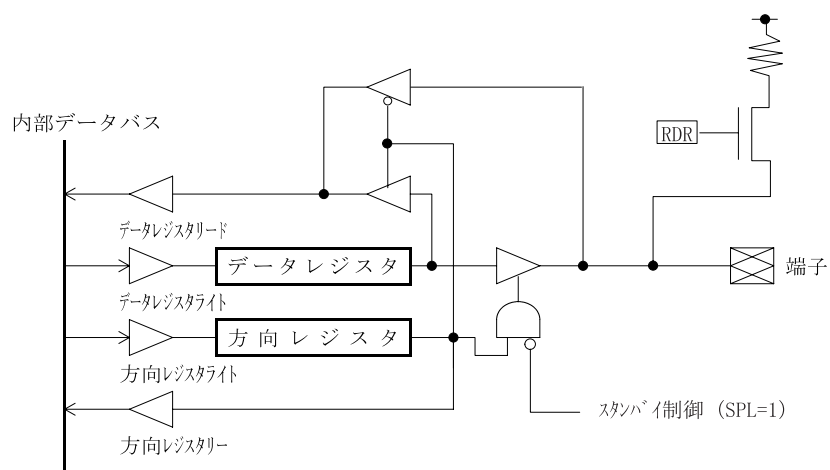


図 7.2-1 I/Oポートのブロックダイアグラム(ポート0.1.6)

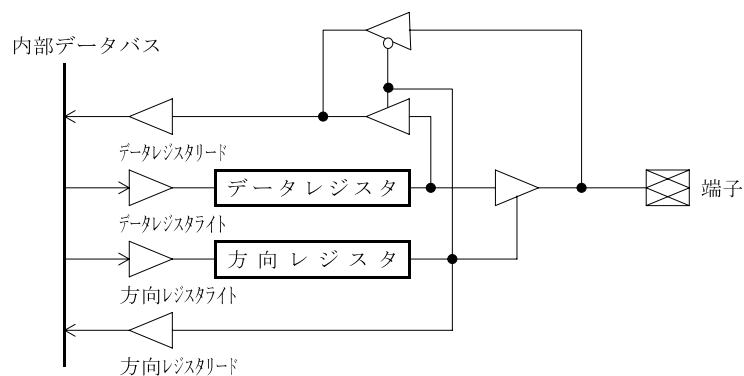


図 7.2-2 I/Oポートのブロックダイアグラム(ポート2.3.7.8.9.A)



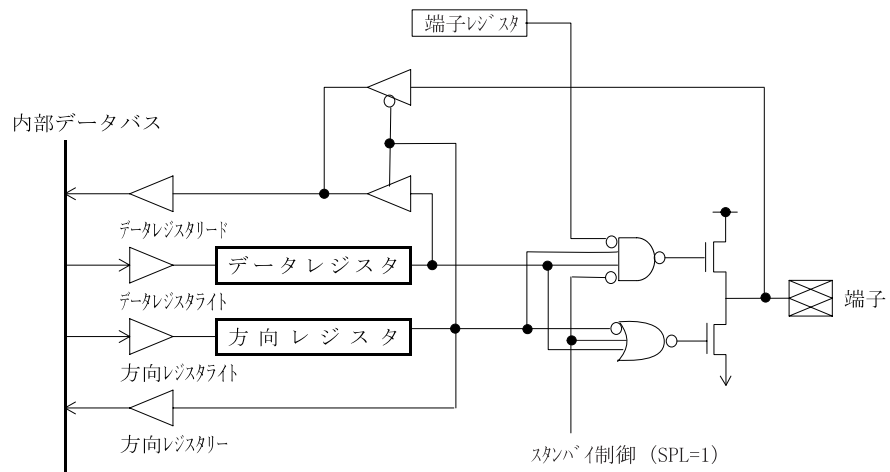


図 7.2-3 I/Oポートのブロックダイヤグラム(ポート4)

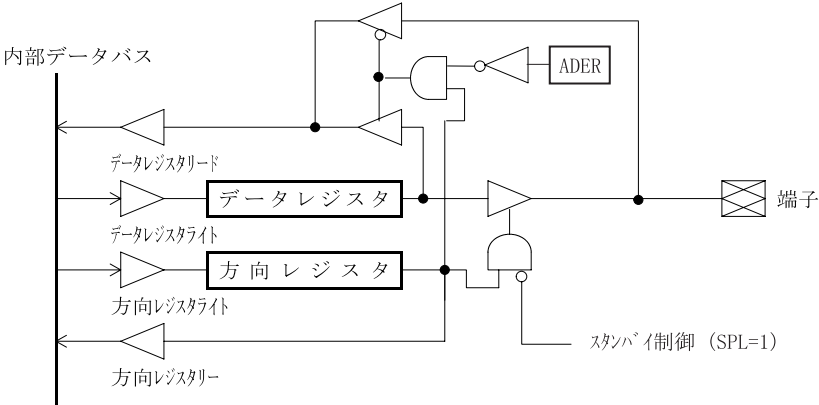


図 7.2-4 I/Oポートのブロックダイヤグラム(ポート5)

## 7.3 I/Oポートのレジスタ

I/Oポートのレジスタには、次の5種類があります。

- ・ポートデータレジスタ(PDRx)
- ・ポート方向レジスタ(DDRx)
- ・出力端子レジスタ(ODR4)
- ・入力抵抗レジスタ(RDR0,1)
- ・アナログ入力許可レジスタ(ADER)

### I/Oポートのレジスタ

ビット	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス : 000000 <sub>H</sub>	P07	P06	P05	P04	P03	P02	P01	P00	ポート 0 データレジスタ (PDR0)
アドレス : 000001 <sub>H</sub>	P17	P16	P15	P14	P13	P12	P11	P10	ポート 1 データレジスタ (PDR1)
アドレス : 000002 <sub>H</sub>	P27	P26	P25	P24	P23	P22	P21	P20	ポート 2 データレジスタ (PDR2)
アドレス : 000003 <sub>H</sub>	P37	P36	P35	P34	P33	P32	P31	P30	ポート 3 データレジスタ (PDR3)
アドレス : 000004 <sub>H</sub>	P47	P46	P45	P44	P43	P42	P41	P40	ポート 4 データレジスタ (PDR4)
アドレス : 000005 <sub>H</sub>	P57	P56	P55	P54	P53	P52	P51	P50	ポート 5 データレジスタ (PDR5)
アドレス : 000006 <sub>H</sub>	-	-	P65	P64	P63	P62	P61	P60	ポート 6 データレジスタ (PDR6)
アドレス : 000007 <sub>H</sub>	-	-	-	P74	P73	P72	P71	-	ポート 7 データレジスタ (PDR7)
アドレス : 000008 <sub>H</sub>	P87	P86	P85	P84	P83	P82	P81	P80	ポート 8 データレジスタ (PDR8)
アドレス : 000009 <sub>H</sub>	P97	P96	P95	P94	P93	P92	P91	P90	ポート 9 データレジスタ (PDR9)
アドレス : 00000A <sub>H</sub>	-	-	-	-	-	PA2	PA1	PA0	ポート A データレジスタ (PDRA)

ビット	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス : 000010 <sub>H</sub>	D07	D06	D05	D04	D03	D02	D01	D00	ポート 0 方向レジスタ (DDR0)
アドレス : 000011 <sub>H</sub>	D17	D16	D15	D14	D13	D12	D11	D10	ポート 1 方向レジスタ (DDR1)
アドレス : 000012 <sub>H</sub>	D27	D26	D25	D24	D23	D22	D21	D20	ポート 2 方向レジスタ (DDR2)
アドレス : 000013 <sub>H</sub>	D37	D36	D35	D34	D33	D32	D31	D30	ポート 3 方向レジスタ (DDR3)
アドレス : 000014 <sub>H</sub>	D47	D46	D45	D44	D43	D42	D41	D40	ポート 4 方向レジスタ (DDR4)
アドレス : 000015 <sub>H</sub>	D57	D56	D55	D54	D53	D52	D51	D50	ポート 5 方向レジスタ (DDR5)
アドレス : 000016 <sub>H</sub>	-	-	D65	D64	D63	D62	D61	D60	ポート 6 方向レジスタ (DDR6)
アドレス : 000017 <sub>H</sub>	-	-	-	D74	D73	D72	D71	-	ポート 7 方向レジスタ (DDR7)
アドレス : 000018 <sub>H</sub>	D87	D86	D85	D84	D83	D82	D81	P80	ポート 8 方向レジスタ (DDR8)
アドレス : 000019 <sub>H</sub>	D97	D96	D95	D94	D93	D92	D91	D90	ポート 9 方向レジスタ (DDR9)
アドレス : 00001A <sub>H</sub>	-	-	-	-	-	DA2	DA1	PA0	ポート A 方向レジスタ (DDRA)

図 7.3-1 I/Oポートのレジスタ (続く)

ビット	15	14	13	12	11	10	9	8	
アドレス : 00001B <sub>H</sub>	OD47	OD46	OD45	OD44	OD43	OD42	OD41	OD40	ポート 4 出力端子レジスタ (ODR4)
ビット	7	6	5	4	3	2	1	0	
アドレス : 00001C <sub>H</sub>	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	ポート 5 アナログ入力許可レジスタ (ADER)
ビット	15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0	
アドレス : 00008C <sub>H</sub>	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	ポート 0 入力プルアップ抵抗設定レジスタ (RDR0)
アドレス : 00008D <sub>H</sub>	RD17	RD16	RD15	RD14	RD13	RD12	RD11	RD10	ポート 1 入力プルアップ抵抗設定レジスタ (RDR1)
アドレス : 00008E <sub>H</sub>	-	-	RD65	RD64	RD63	RD62	RD61	RD60	ポート 6 入力プルアップ抵抗設定レジスタ (RDR6)

図 7.3-1 I/Oポートのレジスタ ( 続き )

### 7.3.1 ポートデータレジスタ(PDRx)

ポートデータレジスタ(PDRx)は、端子に状態を示します。

ポートデータレジスタ(PDRx)

	7	6	5	4	3	2	1	0	初期値	アクセス
PDR0 アドレス : 000000 <sub>H</sub>	P07	P06	P05	P04	P03	P02	P01	P00	XXXXXXXX <sub>B</sub>	R/W
PDR1 アドレス : 000001 <sub>H</sub>	15	14	13	12	11	10	9	8	XXXXXXXX <sub>B</sub>	R/W
PDR2 アドレス : 000002 <sub>H</sub>	7	6	5	4	3	2	1	0	XXXXXXXX <sub>B</sub>	R/W
PDR3 アドレス : 000003 <sub>H</sub>	15	14	13	12	11	10	9	8	XXXXXXXX <sub>B</sub>	R/W
PDR4 アドレス : 000004 <sub>H</sub>	7	6	5	4	3	2	1	0	XXXXXXXX <sub>B</sub>	R/W
PDR5 アドレス : 000005 <sub>H</sub>	15	14	13	12	11	10	9	8	11111111 <sub>B</sub>	R/W
PDR6 アドレス : 000006 <sub>H</sub>	7	6	5	4	3	2	1	0	--XXXXXX <sub>B</sub>	R/W
PDR7 アドレス : 000007 <sub>H</sub>	15	14	13	12	11	10	9	8	---XXXX- <sub>B</sub>	R/W
PDR8 アドレス : 000008 <sub>H</sub>	7	6	5	4	3	2	1	0	XXXXXXXX <sub>B</sub>	R/W
PDR9 アドレス : 000009 <sub>H</sub>	15	14	13	12	11	10	9	8	XXXXXXXX <sub>B</sub>	R/W
PDRA アドレス : 00000A <sub>H</sub>	7	6	5	4	3	2	1	0	-----XXX <sub>B</sub>	R/W

図 7.3-2 ポートデータレジスタ

< 注意事項 >

---

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

- ・入力モード
    - 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。
-

## 7.3.2 ポート方向レジスタ(DDR<sub>x</sub>)

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

ポート方向レジスタ(DDR<sub>x</sub>)

	7	6	5	4	3	2	1	0	初期値	アクセス
DDR0 アドレス : 000010 <sub>H</sub>	D07	D06	D05	D04	D03	D02	D01	D00	00000000 <sub>B</sub>	R/W
	15	14	13	12	11	10	9	8		
DDR1 アドレス : 000011 <sub>H</sub>	D17	D16	D15	D14	D13	D12	D11	D10	00000000 <sub>B</sub>	R/W
	7	6	5	4	3	2	1	0		
DDR2 アドレス : 000012 <sub>H</sub>	D27	D26	D25	D24	D23	D22	D21	D20	00000000 <sub>B</sub>	R/W
	15	14	13	12	11	10	9	8		
DDR3 アドレス : 000013 <sub>H</sub>	D37	D36	D35	D34	D33	D32	D31	D30	00000000 <sub>B</sub>	R/W
	7	6	5	4	3	2	1	0		
DDR4 アドレス : 000014 <sub>H</sub>	D47	D46	D45	D44	D43	D42	D41	D40	00000000 <sub>B</sub>	R/W
	15	14	13	12	11	10	9	8		
DDR5 アドレス : 000015 <sub>H</sub>	D57	D56	D55	D54	D53	D52	D51	D50	00000000 <sub>B</sub>	R/W
	7	6	5	4	3	2	1	0		
DDR6 アドレス : 000016 <sub>H</sub>	-	-	D65	D64	D63	D62	D61	D60	--000000 <sub>B</sub>	R/W
	15	14	13	12	11	10	9	8		
DDR7 アドレス : 000017 <sub>H</sub>	-	-	-	D74	D73	D72	D71	-	---0000- <sub>B</sub>	R/W
	7	6	5	4	3	2	1	0		
DDR8 アドレス : 000018 <sub>H</sub>	D87	D86	D85	D84	D83	D82	D81	D80	00000000 <sub>B</sub>	R/W
	15	14	13	12	11	10	9	8		
DDR9 アドレス : 000019 <sub>H</sub>	D97	D96	D95	D94	D93	D92	D91	D90	00000000 <sub>B</sub>	R/W
	7	6	5	4	3	2	1	0		
DDRA アドレス : 00001A <sub>H</sub>	-	-	-	-	-	DA2	DA1	DA0	-----000 <sub>B</sub>	R/W

図 7.3-3 ポート方向レジスタ(DDR<sub>x</sub>)

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

表 7.3-1 ポート方向レジスタ(DDR<sub>x</sub>)の機能

DDR <sub>x</sub>	機能
0	入力モード [ 初期値 ]
1	出力モード

(注) リソースの入力として使用する場合は“0”(入力モード)を設定してください。

### 7.3.3 ポート4出力端子レジスタ (ODR4)

ポート4出力端子レジスタ (ODR4) では、出力モード時のオープンドレインを制御します。

ポート4出力端子レジスタ (ODR4)

ビット	15	14	13	12	11	10	9	8	
アドレス : 00001B <sub>H</sub>	OD47	OD46	OD45	OD44	OD43	OD42	OD41	OD40	ポート 4 端子レジスタ (ODR4) 初期値 : 00000000 <sub>H</sub>

図 7.3-4 ポート4出力端子レジスタ (ODR4) の機能

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

#### < 注意事項 >

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



### 7.3.4 入力プルアップ抵抗設定レジスタ(RDR0,1,6)

入力プルアップ抵抗設定レジスタ(RDR0,1,6)は、入力モード時のプルアップ抵抗を制御します。

入力プルアップ抵抗設定レジスタ(RDR0,1,6)

ビット	7	6	5	4	3	2	1	0	
アドレス：00008C <sub>H</sub>	RD07	RD06	RD05	RD04	RD03	RD02	RD01	RD00	ポート 0 入力プルアップ抵抗設定レジスタ(RDR0) 初期値：00000000 <sub>8</sub>
ビット	15	16	15	14	13	12	11	10	
アドレス：00008D <sub>H</sub>	RD17	RD16	RD15	RD14	RD13	RD12	RD11	RD10	ポート 1 入力プルアップ抵抗設定レジスタ(RDR1) 初期値：00000000 <sub>8</sub>
ビット	7	6	5	4	3	2	1	0	
アドレス：00008E <sub>H</sub>	RD67	RD66	RD65	RD64	RD63	RD62	RD61	RD60	ポート 6 入力プルアップ抵抗設定レジスタ(RDR6) 初期値：00000000 <sub>8</sub>

図 7.3-5 入力プルアップ抵抗設定レジスタ(RDR0,1,6)

表 7.3-2 入力プルアップ抵抗設定レジスタ(RDR0,1,6)の機能

RDR0,1,6	機能
0	入力モード時プルアップ抵抗なし。[ 初期値 ]
1	入力モード時プルアップ抵抗あり。

< 注意事項 >

- ・出力モード時は意味を持ちません。(プルアップ抵抗なし)
- ・入出力モードは方向レジスタ(DDR)で決まります。
- ・ハードウェアスタンバイ時とストップ時 (SPL=1)のときはプルアップ抵抗なしになります。(ハイインピーダンス)
- ・外バスで使用するときは、本機能は禁止です。本レジスタへは書き込まないようにしてください。

7.3.5           ポート5アナログ入力許可レジスタ(ADER)

ポート5アナログ入力許可レジスタ(ADER)は ,ポート5の各端子を表 7.3-3に示すように制御します。

ポート5アナログ入力許可レジスタ(ADER)

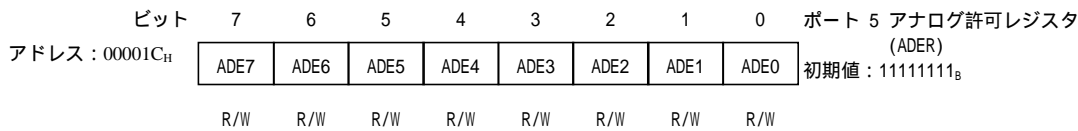


図 7.3-6   ポート5アナログ入力許可レジスタ(ADER)

表 7.3-3   ポート5アナログ入力許可レジスタ(ADER)

ADER	設定
0	ポート入力モード
1	アナログ入力モード [ 初期値 ]

< 注意事項 >

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



## 第8章 タイムベースタイマ

---

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。

---

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

# 8.1 タイムベースタイマの概要

タイムベースタイマは、18bitのタイマの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、CKSCR中のMCSビットと関係なく、発振クロックを使用します。

タイムベースタイマのレジスター一覧

タイムベースタイマ 制御レジスタ	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス: 0000A9 <sub>H</sub>	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 8.1-1 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のブロックダイアグラム

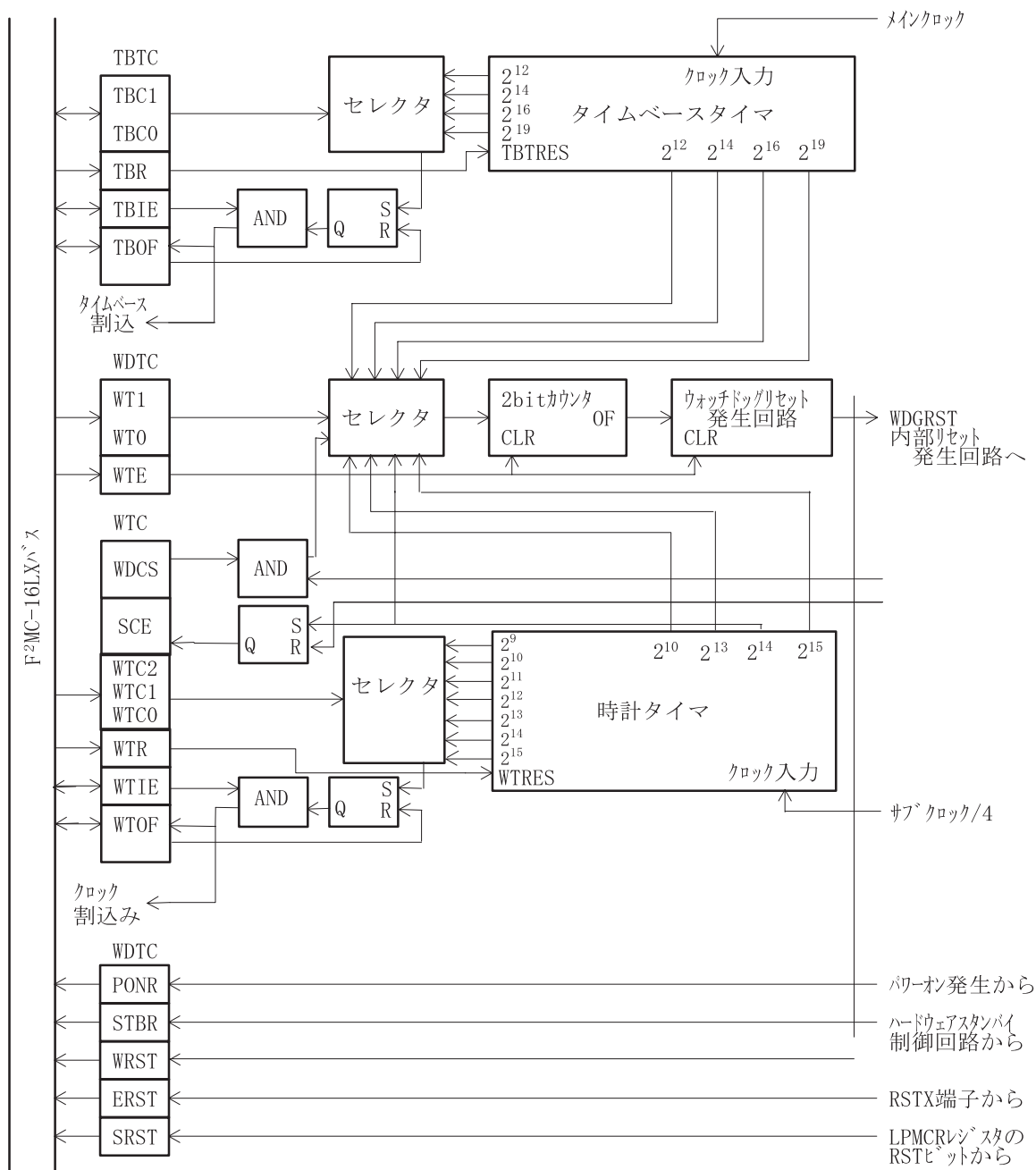


図 8.1-2 タイムベースタイマのブロックダイアグラム

## 8.2 タイムベースタイマ制御レジスタ (TBTC)

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

タイムベースタイマ制御レジスタ (TBTC)

タイムベースタイマ 制御レジスタ	15	14	13	12	11	10	9	8	⇒ビットNo.
アドレス: 0000A9 <sub>H</sub>	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 8.2-1 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】予約(Reserved)

ビット15は、予約ビットです。TBTCを設定する時は、必ず"1"に設定してください。

【ビット12】TBIE

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

【ビット11】TBOF

TBOFは、タイムベースタイマの割込み要求フラグです。TBIEビットが"1"のときTBOFが、"1"になると割込み要求を発生します。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"が読み出されません。

【ビット10】TBR

TBRは、タイムベースタイマのカウンタを全ビット"0"にクリアするビットです。"0"を書き込むことによりタイムベースカウンタをクリアします。"1"の書き込みは意味がありません。リード時は"1"がリードされます。

< 注意事項 >

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

## 【ビット9, 8】TBC1,0

TBC1,0は、タイムベースタイマのインターバルを設定するビットです。  
リセットにより"00"に初期化されます。読み書き可能なビットです。

表 8.2-1 TBC1,0のインターバル時間とサイクル数

TBC1	TBC0	原振4MHz時のインターバル時間	発振クロック（原発振）サイクル数
0	0	1.024 ms	2 <sup>8</sup> サイクル
0	1	4.096 ms	2 <sup>9</sup> サイクル
1	0	16.384 ms	2 <sup>10</sup> サイクル
1	1	131.072 ms	2 <sup>11</sup> サイクル



## 8.3 タイムベースタイマの動作

---

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

---

### タイムベースタイマの動作

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

タイムベースカウンタのクリアは、パワーオンリセット、ストップモード、ハードウェアスタンバイモードへの遷移、CKSCRレジスタ中のMCSビットによるメインクロックからPLLクロックへの遷移、TBTCレジスタ中のTBRビットへの"0"書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り込みはタイムベースタイマのクリアの影響があります。

### インターバル割り込み機能

タイムベースカウンタの桁上がり信号により一定周期で割り込みを発生します。TBTCレジスタ中のTBC1,0ビットで設定されるインターバル時間ごとにTB0F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。

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TB0Fフラグは、モード遷移と同時にクリアされます。

## 第9章 ウォッチドッグタイマ

---

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。

---

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

## 9.1 ウォッチドッグタイマの概要

ウォッチドッグタイマは、18bitのタイムベースタイマの桁上がり信号をクロックソースとする2bitのウォッチドッグカウンタ、コントロールレジスタ、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 制御レジスタ	7	6	5	4	3	2	1	0	←ビットNo.
アドレス: 0000A8 <sub>H</sub>	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)	(1)	(1)	(1)	

図 9.1-1 ウォッチドッグタイマのレジスタ一覧

ウォッチドッグタイマのブロックダイアグラム

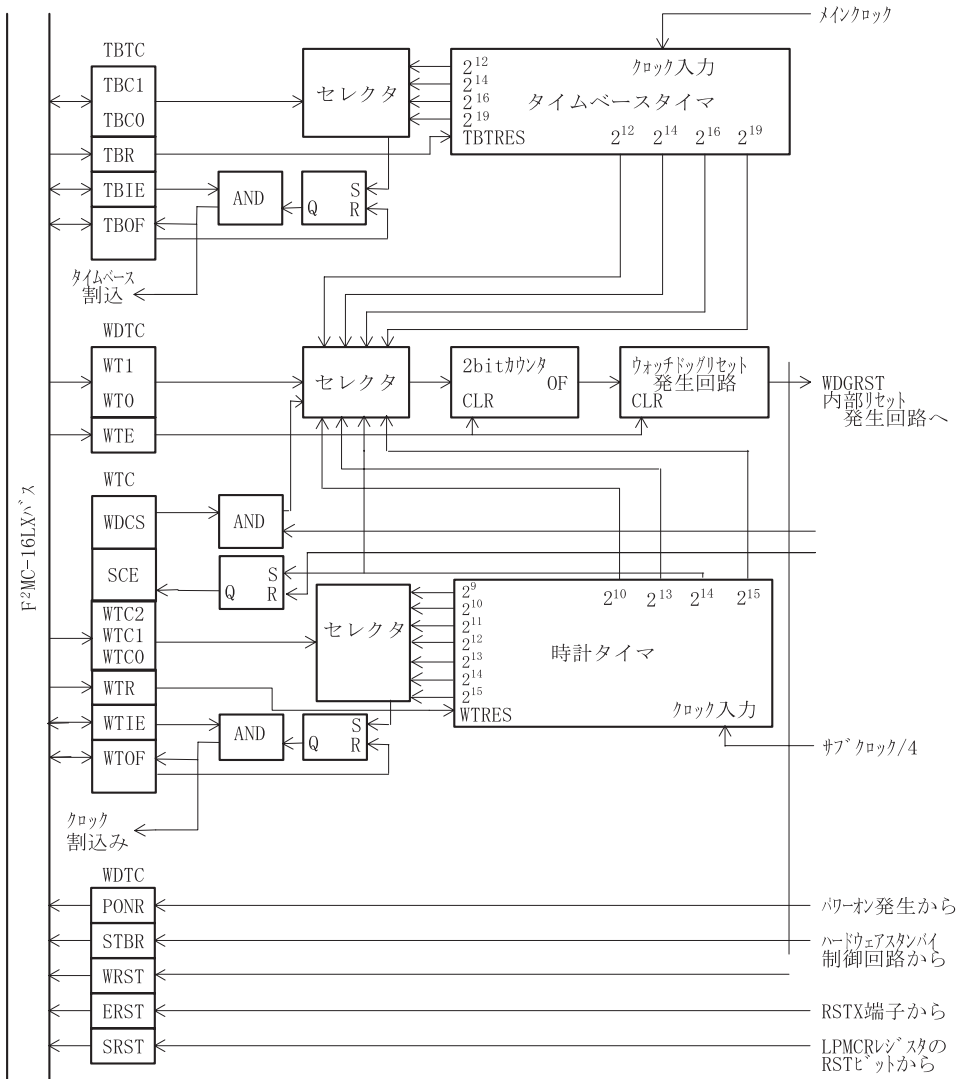


図 9.1-2 ウォッチドッグタイマのブロックダイアグラム

## 9.2 ウォッチドッグタイマ制御レジスタ(WDTC)

ウォッチドッグタイマ制御レジスタ(WDTC)は、ウォッチドッグタイマ起動、クリア、およびリセット要因の表示を行うレジスタです。

ウォッチドッグタイマ制御レジスタ(WDTC)

ウォッチドッグ 制御レジスタ	7	6	5	4	3	2	1	0	←ビットNo.
アドレス: 0000A8 <sub>H</sub>	PONR	STBR	WRST	ERST	SRST	WTE	WT1	WT0	WDTC
リード/ライト 初期値	⇒ ⇒	(R) (X)	(R) (X)	(R) (X)	(R) (X)	(W) (1)	(W) (1)	(W) (1)	

図 9.2-1 ウォッチドッグタイマ制御レジスタ(WDTC)

### < 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7～3】PONR, STBR, WRST, ERST, SRST

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

WDTCレジスタは、読出し専用のレジスタです。なお、パワーオン以外のソースを示すビットの内容は、保証されないことに注意してください。したがって、PONRビットが"1"のときは、その他のビットを無視するようにソフトウェアを設計する必要があります。

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

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

\*：前の値を保持します。

#### [ ビット2 ] WTE

WTEは、ウォッチドッグタイマが停止状態の時"0"を書き込むことによりウォッチドッグタイマを動作状態にします。2回目以降の"0"の書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"の書き込みは何の操作もしません。

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

## 【ビット1, 0】WT1,0

## 2系統品：

WT1,0は、ウォッチドッグタイマのインターバル時間を選択するビットです。ウォッチドッグタイマ起動時に書き込まれたデータのみが有効です。ウォッチドッグ起動時以外の書込みデータは無視されます。書込みのみが可能なビットです。なお、WTCのWDCSビットによりウォッチドッグタイマに入力するクロックが選択されますので注意してください。

WT1,0ビットによるインターバル時間の設定を、表 9.2-2に示します。

表 9.2-2 2系統品のWT1,0(インターバル時間選択ビット)

WDCS	WT1	WTO	インターバル時間（原振4MHz）	
			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	約0.475s	約0.576s
0	0	1	約3.584s	約4.608s
0	1	0	約7.168s	約9.216s
0	1	1	約14.336s	約18.432s

\*：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がリセットされない場合の値です。

## 1系統品：

WT1,0は、ウォッチドッグタイマのインターバル時間を選択するビットです。ウォッチドッグタイマ起動時に書き込まれたデータのみが有効です。ウォッチドッグ起動時以外の書込みデータは無視されます。書込みのみが可能なビットです。

WT1,0ビットによるインターバル時間の設定を、表 9.2-3に示します。

表 9.2-3 1系統品のWT1,0(インターバル時間選択ビット)

WDCS	WT1	WTO	インターバル時間（原振4MHz）	
			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

\*：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がリセットされない場合の値です。

## 9.3 ウォッチドッグタイマの動作

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

### ウォッチドッグタイマの起動方法

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

### ウォッチドッグタイマリセットの阻止

ウォッチドッグタイマが起動されると，プログラム中で定期的に2ビットのウォッチドッグカウンタをクリアする必要があります。具体的には，WDTCレジスタのWTEビットに定期的に"0"を書き込む必要があります。ウォッチドッグカウンタは，タイムベースタイマの桁上あが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短くなることがあります。

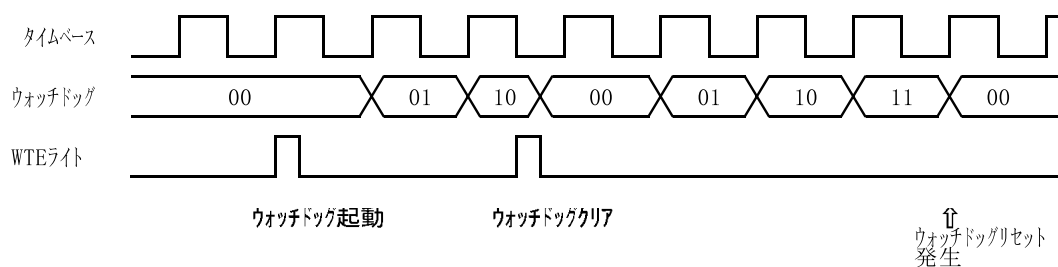


図 9.3-1 ウォッチドッグタイマ動作

### ウォッチドッグ停止

ウォッチドッグタイマは，一度起動すると，パワーオン，ハードウェアスタンバイ，ウォッチドッグによるリセットでのみ初期化され停止状態となります。外部端子，ソフトウェアによるリセットではウォッチドッグのカウンタはクリアしますがウォッチドッグ機能は，停止されません。

### ウォッチドッグタイマのクリア

ウォッチドッグタイマは，WTEビットへの書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信号によってクリアされます。



## 第10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。  
時計タイマは1系統品では使用できません。

---

- 10.1 時計タイマの概要
- 10.2 時計タイマ制御レジスタ(WTC)
- 10.3 時計タイマの動作



## 10.1 時計タイマの概要

時計タイマには、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時間待ちのためのタイマ、一定周期で割り込みを発生するインターバルタイマの機能があります。

## 時計タイマのレジスタ

時計タイマ制御レジスタ制御

	7	6	5	4	3	2	1	0	←ビットNo.
アドレッシング: 0000AAH	WDSCS	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 10.1-1 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

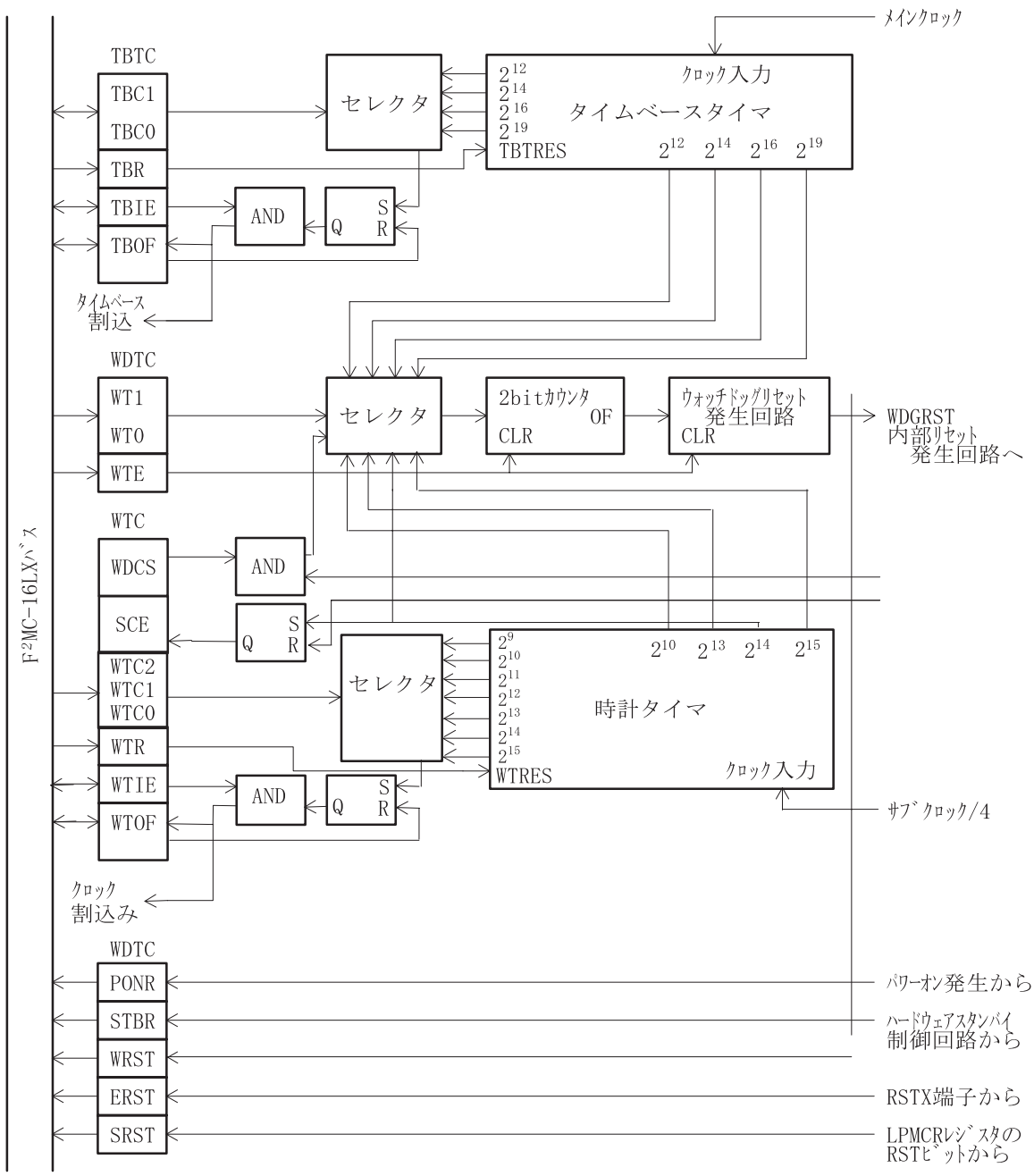


図 10.1-2 時計タイマのブロックダイアグラム

## 10.2 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、クロック信号の選択、割り込みやインターバルの制御、カウンタのクリアを行います。

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 10.2-1 時計タイマ制御レジスタ

### 【ビット7】WDCS

WDCSビットは、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、時計タイマからのクロック信号を使用するか、タイムベースタイマからのクロック信号を使用するかを選択するビットです。"0"のとき、時計タイマのクロック信号が、"1"のときタイムベースタイマのクロック信号を選択します。サブクロックモードの場合、必ず"0"に設定して時計タイマの出力を選択してください。"1"に設定した状態でサブクロックモードへ移行した場合、ウォッチドッグタイマは停止します。

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、ハードウェアスタンバイ時およびストップ時に、本ビットは"0"に初期化されます。

### 【ビット5】WTIE

WTIEビットは、時計タイマによるインターバル割り込みを許可するビットです。本ビットが "1"のとき割り込み許可、"0"のとき割り込みを禁止します。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

WTOFビットは、時計タイマの割り込み要求フラグです。WTIEビットが"1"のとき、WTOFを"1"に設定すると割り込み要求が発生します。本ビットは、WTC2～WTC0ビットで設定したインターバルごとに"1"に設定されます。本ビットは、"0" の書込み、ストップモードかハードウェアスタンバイモードへの遷移、またはリセットによりクリアされます。"1" の書込みは何の意味も持ちません。

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

WTRビットは、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"を書き込むと、クロックカウンタをクリアします。"1"の書込みは何の意味も持ちません。常に"1"が読み出されます。

## 【ビット2,1,0】WTC2, WTC1, WTC0

WTC2, WTC1, WTC0は、時計タイマのインターバルを設定するビットです。インターバルの設定を表 10.2-1に示します。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。

本ビットを書き込むときは、ビット4(WT0F)も同時にクリアします。

表 10.2-1 時計タイマインターバルの選択

WTC2	WTC1	WTC0	インターバル時間*
0	0	0	62.5 ms
0	0	1	125.0 ms
0	1	0	250 ms
0	1	1	500 ms
1	0	0	1.0 s
1	0	1	2.000 s
1	1	0	4.000 s
1	1	1	-

\*：インターバル時間は、サブクロック 32kHz時の値です。

## 10.3 時計タイマの動作

---

時計タイマには、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時間待ちのためのタイマ、一定周期で割込みを発生するインターバル割込み機能があります。

---

### 時計タイマの動作

時計タイマは、サブクロックを発生させるために使用されるソース発振入力のカウントする 15ビットのカウンタにより構成され、ソース発振が入力されている間時計タイマは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、ストップモードかハードウェアスタンバイモードへの遷移、およびWTCレジスタ中の 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、時計タイマのクリアの影響があります。

### 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、クロックカウンタの桁上がり信号により所定の時間割込みを発生します。WTCレジスタ中のWTC2～WTC0ビットで設定される通常のインターバルごとに 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ます。

ストップモードか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サブクロック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。

### ウォッチドッグタイマの動作クロックの設定

時計タイマ制御レジスタ(WTC)のWDCSビットによって、ウォッチドッグタイマのクロックソースを設定できます。マシクロックとしてサブクロックを使用する場合、必ずWDCSビットを"0"に設定して時計タイマの出力を選択してください。WDCSビットを"1"に設定した状態でサブクロックモードへ移行した場合、ウォッチドッグタイマは停止します。

## 第11章 PWCタイマ

---

この章では，PWCタイマの機能と動作について説明します。

---

- 11.1 PWCタイマの概要
- 11.2 PWCタイマのブロックダイアグラム
- 11.3 PWCタイマのレジスタ
- 11.4 PWCタイマの動作
- 11.5 タイマモードの動作の詳細
- 11.6 タイマモード動作のフローチャート
- 11.7 パルス幅測定モード動作の詳細
- 11.8 PWCタイマ取扱い上の注意事項

## 11.1 PWCタイマの概要

PWCタイマ(パルス幅測定)は、リロード機能を備えた多機能16ビットアップカウンタで、入力信号のパルス幅を算出する機能も備えています。

PWCタイマは、16ビットカウンタ、入力パルス分周器、分周比コントロールレジスタ、1つのカウント入力端子、1つのパルス出力端子、および16ビットコントロールレジスタにより構成されます。

### PWCタイマの特長

PWCタイマには、次のような特長があります。

#### タイマ機能

- 指定された時間間隔で割込み要求を発生します。
- タイマの周期と同期したパルス信号を出力します。
- 3つの内部クロックからカウンタのクロックを選択します。

#### パルス幅測定機能

- 外部パルス入力のイベント間の時間を測定します。
- 3つの内部クロックからカウンタのクロックを選択します。
- カウントモード
  - Hパルス幅 (立ち上がり～立ち下がり)/Lパルス幅(立ち下がり～立ち上がり)
  - 立上りエッジ周期(立ち上がり～立ち上がり)/立下りエッジ周期(立ち下がり～立ち下がり)
  - 中間エッジカウント(立ち上がりまたは立ち下がり～立ち下がりまたは立ち上がり)
- 8ビット入力分周器で、入力パルスを $2^2$ 、 $2^4$ 、 $2^6$ 、 $2^8$ 分周して、周期測定を行うことが可能です。
- カウント完了時に割込み要求を発生します。
- 単発カウントまたは連続カウントを選択します。

MB90580Cシリーズでは、1つのPWCタイマチャンネルを内蔵しています。

### PWCタイマの動作

このブロックは、16ビットアップカウントタイマを基本にした多機能タイマで、カウント入力端子と8ビット入力分周器を組み込んでいます。ブロックには、タイマ機能およびパルス幅測定機能の2つの主要機能があります。どちらの機能でも2種類のカウントクロックを選択することができます。





## 11.3 PWCタイマのレジスタ

PWCタイマに関連するレジスタを示します。

PWCタイマのレジスタ一覧

PWCコントロールステータスレジスタ(上位バイト)

アドレス : 000055 <sub>H</sub>	15	14	13	12	11	10	9	8	↔ ビット番号
	STRT	STOP	EDIR	EDIE	OVIR	OVIE	ERR	POUT	PWCSR 上位
	リード/ライト ⇔ 初期値 ⇔	(R/W) (0)	(R/W) (0)	(R) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R) (0)	(R/W) (0)

PWCコントロールステータスレジスタ(下位バイト)

アドレス : 000054 <sub>H</sub>	7	6	5	4	3	2	1	0	↔ ビット番号
	CKS1	CKS0	予約	予約	S/C	MOD2	MOD1	MOD0	PWCSR 下位
	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)	

PWCデータバッファレジスタ(上位バイト)

アドレス : 000057 <sub>H</sub>	15	14	13	12	11	10	9	8	↔ ビット番号
									PWCR 上位
	リード/ライト ⇔ 初期値 ⇔	(R/W) (X)	(R/W) (X)	(R) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R) (X)	(R/W) (X)

PWCデータバッファレジスタ(下位バイト)

アドレス : 000056 <sub>H</sub>	7	6	5	4	3	2	1	0	↔ ビット番号
									PWCR 下位
	リード/ライト ⇔ 初期値 ⇔	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	(R/W) (X)	

分周比コントロールレジスタ

アドレス : 000058 <sub>H</sub>	7	6	5	4	3	2	1	0	↔ ビット番号
	—	—	—	—	—	—	DIV1	DIV0	DIVR
	リード/ライト ⇔ 初期値 ⇔	(-) (-)	(-) (-)	(-) (-)	(-) (-)	(-) (-)	(R/W) (0)	(R/W) (0)	

PWCノイズフィルタレジスタ

アドレス : 000086 <sub>H</sub>	7	6	5	4	3	2	1	0	↔ ビット番号
	—	—	—	—	—	SW1	SW0	EN	RNCR
	リード/ライト ⇔ 初期値 ⇔	(-) (-)	(-) (-)	(-) (-)	(-) (-)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

### 11.3.1 PWCコントロールステータスレジスタ(PWCSR)

PWCコントロールステータスレジスタ(PWCSR)は、PWCタイマの動作を制御し、PWCタイマの状態を読み出します。

PWCコントロールステータスレジスタ(PWCSR)

PWCコントロールステータスレジスタ(PWCSR)のレジスタ構成を、図 11.3-1に示します。

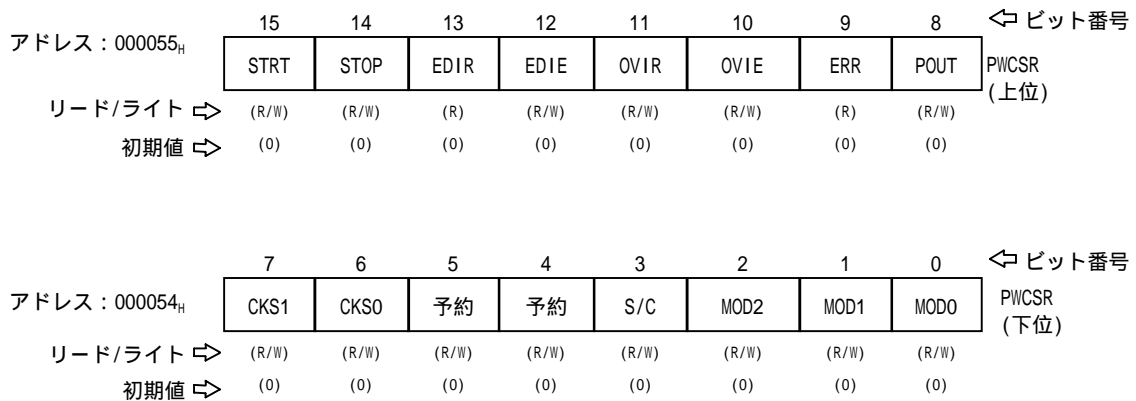


図 11.3-1 PWCコントロールステータスレジスタ(PWCSR)

#### 【ビット15,14】STRT (スタート), STOP (ストップ)

STRT (スタート), STOP (ストップ)により、16ビットアップカウンタタイマの起動、再起動、停止が行われます。ビットを読み出すと、タイマの動作状態が返されます。表 11.3-1に書込み時の、表 11.3-2に読出し時のビット機能を示します。

表 11.3-1 STRTビットとSTOPビットの動作制御機能(書込み時)

STRT	STOP	動作制御機能
0	0	機能なし。動作に影響しません。
0	1	タイマの起動または再起動させます (カウント許可)。注意：クリアビット命令を使用することができます。
1	0	タイマの動作を強制的に中断させます (カウント禁止)。注意：クリアビット命令を使用することができます。
1	1	機能なし。動作に影響しません。

表 11.3-2 STRTビットとSTOPビットの動作状態表示(読出し時)

STRT	STOP	動作ステータス表示
0	0	タイマの停止 (起動しない、またはカウントの終了)。(初期値)
1	1	タイマによりカウント動作中 (カウントの進行)。

リセット後は00<sub>h</sub>に初期化されます。

読出しと書込みが可能です。読出しと書込みではビットの意味が違うことに注意してください。

リードモディファイライト系統の命令は常にビットを11<sub>B</sub>として読み出します。  
 タイマの起動や停止を行うために、STRTビットとSTOPビットへの書込みの際には、(ビットクリア命令などの)ビット操作命令を使用することができますが、(例えば、タイマが動作中であることを常に表している)動作状態を読み出す際にはビット操作命令を使用することはできません。

#### 【ビット13】 EDIR (EnD Interrupt Request)

EDIRフラグは、パルス幅測定モード時、測定が終了したことを示すフラグです。このビットをセットして割込みが許可されると(bit 12:EDIE="1"), 測定終了の割込み要求が発生します。

セット要因	パルス幅測定が終了するとセット(PWCRに測定結果が収納されている)。
クリア要因	PWCR(測定結果)をリードすることによりクリア。

#### <注意事項>

タイマモードでは、このビットに意味はありません。

リセット後は"0"に初期化されます。

読出し専用。このビットへの書込みは意味がありません。

#### 【ビット12】 EDIE (EnD Interrupt Enable)

パルス幅カウントモード時の測定終了の割込み要求を以下のように制御します。

0	測定終了割込み要求の出力禁止 (EDIRのセット時に割込みは発生しない)。(初期値)
1	測定終了割込み要求の出力許可 (EDIRのセット時に割込みを発生させる)。

#### <注意事項>

タイマモードの最中は必ず"0"にセットしてください。

リセット後は"0"に初期化します。

読出しと書込みが可能です。

#### 【ビット11】 OVIR (OVerflow Interrupt Request)

このフラグは、いつ16ビットアップカウントタイマがFFFF<sub>H</sub> ~ 0000<sub>H</sub>の範囲でオーバーフローするかを指示します。動作はすべてのモードに及んでいます。このビットをセットして割込みが許可されると(bit 10:OVIE ="1"), タイマオーバーフロー割込み要求が発生します。

セット要因	タイマオーバーフローの発生時(FFFF <sub>H</sub> ~ 0000 <sub>H</sub> )にセット
クリア要因	"0"の書込みまたは拡張インテリジェントI/Oサービスによりクリア。

リセット後は"0"に初期化されます。

読出しと書込みが可能です。ただし、"0"の書込みだけが有効です。"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統の命令は常にビットを"1"として読み出します。

#### <注意事項>

H/Lパルス幅カウントモードでは、本ビットを用いたパルス幅時間測定は行わないでください。

## 【ビット10】 OVIE (OVerflow Interrupt Enable)

OVIEは、タイマオーバフロー割込み要求制御ビットです。タイマオーバフローの割込み要求を、表 11.3-3で示すように制御します。

リセット後は"0"に初期化します。読出しと書込みが可能です。

表 11.3-3 OVIE(タイマオーバフロー割込み要求制御ビット)

OVIE	機能
0	オーバフロー割込み要求の出力禁止 (OVIRのセット時に割込みは発生しない)。(初期値)
1	オーバフロー割込み要求の出力許可 (OVIRのセット時に割込みを発生する)。

## &lt; 注意事項 &gt;

H/Lパルス幅カウントモードでは、本ビットは"0"でご使用ください。

## 【ビット9】 ERR (ERRor)

ERRフラグを使用するのは、パルス幅カウントモードで連続カウントを実行する場合です。このフラグは、前のカウント結果をPWCRから読み出す前に、次のカウントが完了したことを表しています。このようなことが生じると、新規のカウント結果によりPWCRが上書きされ、前の結果が失われます。このビットの値とは無関係にカウント動作は継続します。

リセット後は"0"に初期化されます。読出し専用。このビットへの書込みは値を変更しません。

表 11.3-4 ERR (ERRor)

セット要因	読み出していないカウント結果が次の結果によって上書きされるとセット。
クリア要因	PWCR (測定結果) リードによりクリア。

## 【ビット8】 POUT (Pulse OUTput)

タイマモード時、16ビットアップカウントタイマがFFFF<sub>H</sub>～0000<sub>H</sub>の範囲でオーバフローするたびに、このビットを反転します。

パルス幅カウントモードでは、このビットに意味はありません。

リセット後は"0"に初期化されます。

読出しと書込みが可能です。ただし、ビットへの書込みが可能となるのは、タイマが停止する場合(bit 15およびbit 14:STRTおよびSTOPが両方ともに"0"となる場合)に限られます。タイマ動作の最中に書き込まれた場合(bit 15およびbit 14:STRTおよびSTOPが両方ともに"1"となる場合) には、ビットの値は変化しません。

表 11.3-5 POUT (Pulse OUTput)

セット要因	POUTの値が"0"でFFFF <sub>H</sub> ～0000 <sub>H</sub> の範囲でタイマがオーバフローしたとき。または、タイマの停止時に"1"を書き込みでセット。
クリア要因	POUTの値が"1"でFFFF <sub>H</sub> ～0000 <sub>H</sub> の範囲でタイマがオーバフローしたとき。または、タイマの停止時に"0"を書き込んでクリア。リセットでもクリア。

## 【ビット7,6】CKS1,CKS0 (Clock Select 1,0)

CKS1,CKS0は、内部カウントクロック選択ビットです。これらのビットにより、内部カウントクロックを表 11.3-6で示すように選択します。

リセット後は"00<sub>B</sub>"に初期化します。読出しと書込みが可能です。ただし,"11<sub>B</sub>"をセットすることはできません。

表 11.3-6 CKS1,CKS0(内部カウントクロック選択ビット)

CKS1	CKS0	カウントクロックの選択
0	0	マシンサイクルの4分周クロック(16MHzのマシンサイクルの場合は0.25 $\mu$ s) (初期値)
0	1	マシンサイクルの16分周クロック(16MHzのマシンサイクルの場合は1.0 $\mu$ s)
1	0	マシンサイクルの32分周クロック(16MHzのマシンサイクルの場合は2.0 $\mu$ s)
1	1	設定禁止(未定義)

## &lt; 注意事項 &gt;

タイマの起動後に設定を変更することは禁止します。これらのビットへの書込みは、タイマの起動前または停止後に行ってください。

## 【ビット5,4】予約ビット(reserved)

ビット5,4は、予約ビットです。必ず"00<sub>B</sub>"を書込んでください。

## 【ビット3】S/C (Single/Continuous)

S/Cビットは、カウントモード選択ビットです。カウントモードは、表 11.3-7で示すように選択します。リセット後は"0"に初期化されます。読出しと書込みが可能です。

表 11.3-7 S/C (カウントモード選択ビット)

S/C	カウントモードの選択	タイマモード	パルス幅カウントモード
0	単発測定モード (初期値)	リロードなし(ワンショット)	1回測定後に停止
1	連続測定モード	リロードあり(リロードタイマ)	連続測定： バッファレジスタ有効

## &lt; 注意事項 &gt;

タイマの起動後に設定を変更することは禁止します。これらのビットへの書込みは、タイマの起動前または停止後に行ってください。

## 【ビット2,1,0】MOD2,MOD1,MOD0 (MOD2,1,0)

これらのビットを設定することにより、動作モードと、パルス幅のカウントに適したパルスエッジを表 11.3-8で示すように選択できます。リセット後は"000<sub>B</sub>"に初期化されます。読出しと書込みが可能です。

表 11.3-8 MOD2,MOD1,MOD0(動作モード/カウントエッジ選択ビット)

MOD2	MOD1	MOD0	動作モード/カウントエッジの選択
0	0	0	タイマモード,パルス出力なし(初期値)
0	0	1	タイマモード,パルス出力あり(POT端子有効):リロードモードのみ
0	1	0	全エッジ間パルス幅測定モード(立上りまたは立下り~立下りまたは立上り)*
0	1	1	分周同期測定モード(入力分周器の使用時)*
1	0	0	立上りエッジ間周期測定モード(立ち上がり~立ち上がり)*
1	0	1	"H"パルス幅測定モード(立ち上がり~立ち下がり)*
1	1	0	"L"パルス幅測定モード(立ち下がり~立ち上がり)*
1	1	1	立下りエッジ間周期測定モード(立ち下がり~立ち下がり)*

## &lt; 注意事項 &gt;

タイマの起動後に設定を変更することは禁止します。これらのビットへの書込みは、タイマの起動前または停止後に行ってください。

\*の設定に対して連続測定モードをセットすると、エッジの数を累算するために、カウントの終了時に内部カウントクロック用の分周回路はクリアされません。その他のすべてのモードでは、カウントの終了時に内部カウントクロック用の分周回路はクリアされます。

### 11.3.2 PWCデータバッファレジスタ(PWCR)

PWCデータバッファレジスタ(PWCR)は、PWCタイマの動作モードによって、その役割が変わるレジスタです。

PWCデータバッファレジスタ(PWCR)

PWCデータバッファレジスタ(PWCR)のレジスタ構成を、図 11.3-2に示します。

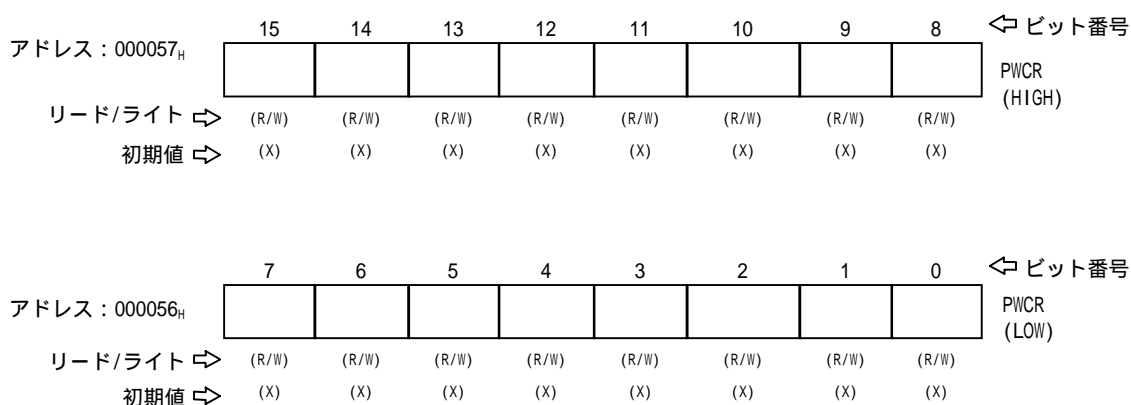


図 11.3-2 PWCデータバッファレジスタ(PWCR)

#### タイマモードの場合

リロードタイマ動作モード(PWCSRの[bit 3] S/C ="1")では、このレジスタがリロード値を格納します。この場合、レジスタの読出しと書込みが可能です。

単発タイマ動作モード(PWCSRの[bit 3] S/C ="0")では、このレジスタへの直接アクセスによりアップカウンタイマにアクセスします。このモードでは読出しと書込みの両方が許されていますが、レジスタへの書込みはタイマの停止時に限られます。レジスタの読出しはいつでも可能で、現時点のタイマの値を読み出します。

#### パルス幅測定モードの場合(読出し専用)

連続測定モード(PWCSRの[bit 3] S/C ="1")では、このレジスタがバッファレジスタとして機能し、前のカウント結果を格納します。この場合、レジスタは読出し専用で、書込みによりレジスタの値は変化しません。

単発測定モード(PWCSRの[bit 3] S/C ="0")では、このレジスタへの直接アクセスによりアップカウンタイマにアクセスします。このモードでもレジスタは読出し専用で、書込みによりレジスタの値は変化しません。レジスタの読出しはいつでも可能で、現時点のタイマの値を読み出します。カウントの終了後に、レジスタはカウント結果を格納します。

#### < 注意事項 >

このレジスタにアクセスするには、必ずワード転送命令を使用してください。

リセット後は"0000<sub>H</sub>"に初期化されます。

### 11.3.3 分周比コントロールレジスタ(DIVR)

分周比コントロールレジスタ(DIVR)は、分周周期測定モード(PWCSRの[bit 2,1,0] MOD2,1,0="011")のときに使用するレジスタです。ほかのモードでは意味を持ちません。

分周比コントロールレジスタ(DIVR)

分周比コントロールレジスタ(DIVR)のレジスタ構成を、図 11.3-3に示します。

	7	6	5	4	3	2	1	0	↔ ビット番号
アドレス : 000058 <sub>H</sub>	—	—	—	—	—	—	DIV1	DIV0	DIVR
リード/ライト ⇔	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
初期値 ⇔	(-)	(-)	(-)	(-)	(-)	(-)	(0)	(0)	

図 11.3-3 分周比コントロールレジスタ(DIVR)

#### 【ビット1,0】DIV1,DIV0

分周周期測定モードでは、測定端子からのパルス入力を、このレジスタで分周し、分周後の1周期幅を測定します。分周比は、表 11.3-9で示すように選択します。

リセット後は"00<sub>h</sub>"に初期化されます。読出しと書込みが可能です。

表 11.3-9 DIV1,DIV0ビットによる分周比の選択

DIV1	DIV0	分周比の選択
0	0	$2^2 = 4$ 分周 (初期値)
0	1	$2^4 = 16$ 分周
1	0	$2^6 = 64$ 分周
1	1	$2^8 = 256$ 分周

#### < 注意事項 >

タイマの起動後に設定を変更することは禁止します。これらのビットへの書込みは、タイマの起動前または停止後に行ってください。



### 11.3.4 PWCノイズフィルタレジスタ(RNCR)

PWCノイズフィルタレジスタ(RNCR)は、PWCノイズ除去回路を使用して入力信号からノイズを除去します。HレベルとLレベルの検出が行われるのは、ノイズフィルタ通過後です。

PWCノイズフィルタレジスタ(RNCR)

PWCノイズフィルタレジスタ(RNCR)のレジスタ構成を、図 11.3-4に示します。

	7	6	5	4	3	2	1	0	ビット番号
アドレス : 000086 <sub>H</sub>	—	—	—	—	—	SW1	SW0	EN	RNCR
リード/ライト ⇨	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	
初期値 ⇨	(-)	(-)	(-)	(-)	(-)	(0)	(0)	(0)	

図 11.3-4 PWCノイズフィルタレジスタ(RNCR)

ノイズ除去回路はデジタルローパスフィルタで、このフィルタにより入力信号の高周波成分を除去します。PWCノイズフィルタレジスタのSWビットが指定するノイズパルス幅は、PWCノイズ除去回路によって除去することができます。

このPWCノイズフィルタレジスタは8ビットレジスタで、リセット時には、すべてのビットが"0"に初期化されます。

【ビット2,1】SW1,SW0

SW1とSW0はクロックモード選択ビットで、除去すべきノイズパルス幅を指定します。表 11.3-10に記載したタイミングは、メインクロックが16MHz時のものです。

表 11.3-10 SW1,SW0(クロックモード選択ビット)

SW1	SW0	ノイズパルス幅(最小)
0	0	$2^8/\text{メインクロック}$ (メインクロック=16MHzの場合, 16.0μs)
0	1	$2^{12}/\text{メインクロック}$ (メインクロック=16MHzの場合, 256.0μs)
1	0	$2^{13}/\text{メインクロック}$ (メインクロック=16MHzの場合, 512.0μs)
1	1	$2^{14}/\text{メインクロック}$ (メインクロック=16MHzの場合, 1.024ms)

\*:原発振

【ビット0】EN

ENビットを使用してこのノイズ解消機能を有効にします。

表 11.3-11 ENビットの機能

EN	機能
0	ノイズフィルタ機能の禁止(初期値)
1	ノイズフィルタ機能の許可

## 11.4 PWCタイマの動作

PWCタイマは、16ビットアップカウントタイマを基本にした多機能タイマで、カウント入力端子と8ビット入力分周器を組み込んでいます。ブロックには次の2つの主要機能があります。タイマ機能およびパルス幅カウント機能のどちらの機能でも2種類のカウンタクロックを選択することができます。

### タイマ機能

タイマ機能は、単発モードまたはリロードモードで動作するように選択することができます。アップカウントタイマです。

いったん起動すると、各カウンタクロックのたびにタイマのカウントが行われます。

FFFF<sub>H</sub> ~ 0000<sub>H</sub>の範囲でオーバーフローが生じると、割込み要求が発生します。

オーバーフローが生じた場合は以下ようになります。

- 単発モード：カウントが中断します（図 11.4-1参照）
- リロードモード：タイマにリロードレジスタの内容をリロードし、カウントを再び開始します。（図 11.4-2参照）

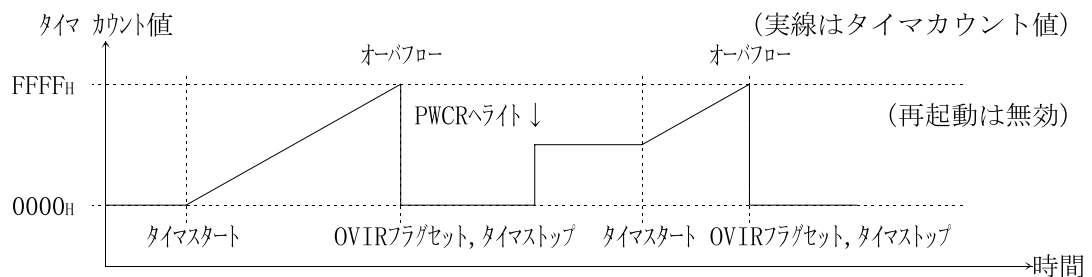


図 11.4-1 タイマの動作（単発モード）

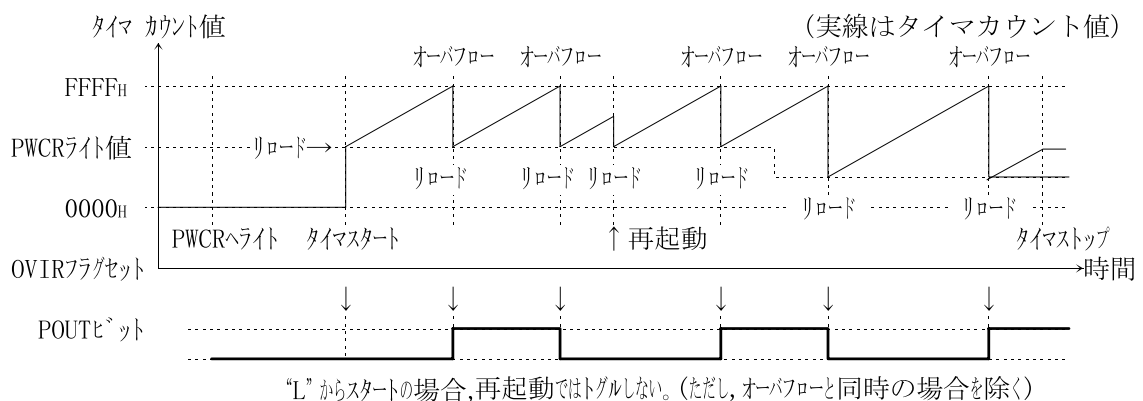


図 11.4-2 タイマの動作（リロードモード）

## パルス幅測定機能

パルス幅測定機能は、入力パルスにかかわる指定されたイベント間の時間を算出します。

この機能をアクティブにした後にカウントが開始されるのは、指定されたカウント開始エッジの入力以後です。カウンタが"0000<sub>h</sub>"にクリアされ、開始エッジの検出時にカウントを開始し停止エッジを検出すると、この間のカウント値がパルス幅としてレジスタに保持されます。

測定の終了時またはオーバフローの発生時に割り込み要求を発生させることができます。測定が完了した後は、以下のようになります。

単発測定モード時

動作が中断します（図 11.4-3参照）。

連続測定モード時

タイマの値はバッファレジスタに転送され、次の開始エッジが入力されるまでタイマはフリーラン状態です（図 11.4-4参照）。

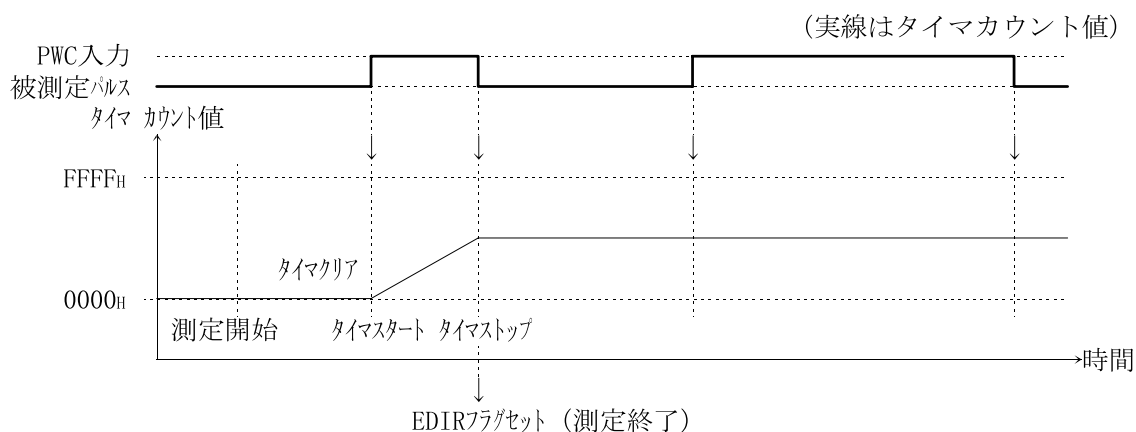
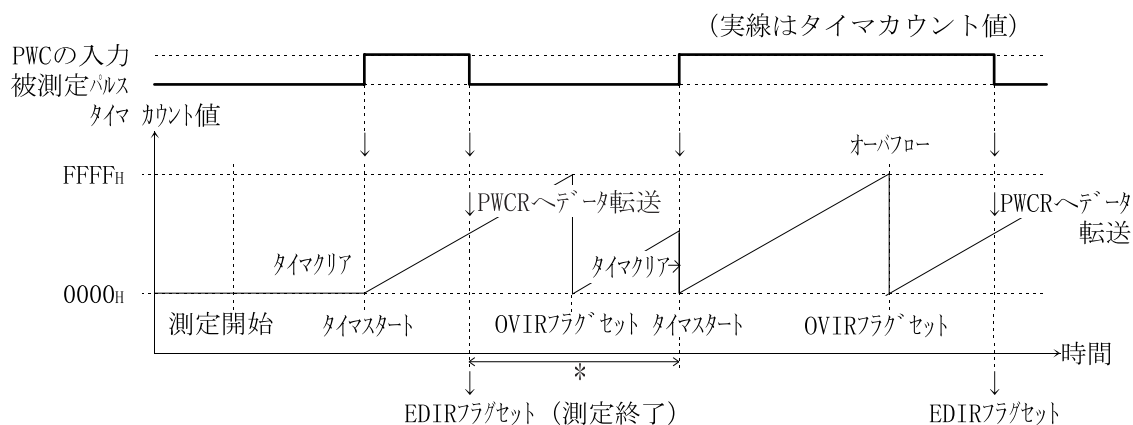


図 11.4-3 パルス幅測定動作（単発測定モード，“H”幅測定モード）



\*: この期間のタイ値は保証されません(タイのオーバーフローによりOVRがセットされる場合があります)。

図 11.4-4 パルス幅測定動作（連続測定モード，“H”幅測定モード）

### 11.4.1 カウントクロックの選択

タイマのカウントクロックは、内部クロックソースを以下の3種類から選択することができます。

- ・マシンサイクル / 4
- ・マシンサイクル / 16
- ・マシンサイクル / 32

#### カウントクロックの選択

使用可能なクロックソースの一覧を表 11.4-1に示します。

表 11.4-1 カウントクロックの選択

PWCSR/bit 7, bit 6: CKS1,0	内部カウントクロックの選択
00 <sub>2</sub>	マシンサイクル / 4 (マシンサイクル=16MHzは0.25μs) (初期値)
01 <sub>2</sub>	マシンサイクル / 16 (マシンサイクル=16MHzは1.0μs)
10 <sub>2</sub>	マシンサイクル / 32 (マシンサイクル=16MHzは2.0μs)

セット後は、マシンサイクル / 4が選択されています。

#### < 注意事項 >

タイマの起動前に、カウントクロックを必ず選択してください。

## 11.4.2 動作モードの選択

各動作モードとカウントモードの選択は、PWCSRの設定により行われます。

### 動作モードの選択

各動作モード、カウントモードの選択は、以下のレジスタで設定します。

動作モードの設定：PWCSR[bit 2,1,0] MOD2,MOD1,MOD0ビット

タイマモードまたはパルス幅測定モードを選択し、カウント動作を制御します。

カウントモードの設定：PWCSR[bit 3] S/Cビット

単発測定または連続測定、またはリロード動作またはワンショット動作を選択します。

モード設定ビットを用いて選択する動作モードは、表 11.4-2に示すとおりです。

表 11.4-2 動作モードの選択

動作モード			S/C	MOD2	MOD1	MOD0
タイマ	ワンショットタイマ		0	0	0	0
	リロードタイマ		1	0	0	0
	設定禁止		1	0	0	1
パルス幅測定	立ち上がりまたは立ち下がり ～ 立ち上がりまたは 立ち下がり 全エッジ間測定	単発測定：バッファ無効	0	0	1	0
		連続測定：バッファ有効	1	0	1	0
	分周カウント 1 ～ 256分周	単発測定：バッファ無効	0	0	1	1
		連続測定：バッファ有効	1	0	1	1
	立ち上がり～立ち上がり 立上り間周期測定	単発測定：バッファ無効	0	1	0	0
		連続測定：バッファ有効	1	1	0	0
	立ち上がり～立ち下がり "H"パルス幅測定	単発測定：バッファ無効	0	1	0	1
		連続測定：バッファ有効	1	1	0	1
	立ち下がり ～ 立ち上がり "L"パルス幅測定	単発測定：バッファ無効	0	1	1	0
		連続測定：バッファ有効	1	1	1	0
	立ち下がり ～ 立ち下がり 立下り間周期測定	単発測定：バッファ無効	0	1	1	1
		連続測定：バッファ有効	1	1	1	1

リセット後の初期値によりワンショットタイマが選択されています。

### < 注意事項 >

タイマの起動前に動作モードを必ず選択してください。

### 11.4.3 タイマ / パルス幅測定 of 開始と停止 , タイマ of クリア

タイマ / パルス幅測定 of 開始 , 再起動 , 強制停止を実行するには , PWCSR of bit 15 と bit 14 (STRT と STOP) を使用します。

また , 16 ビットアップカウントタイマは , リセット時およびパルス幅測定モードで測定開始エッジを検出し , カウントを開始するときに 0000<sub>H</sub> にクリアされます。

#### タイマ / パルス幅測定 of 開始と停止

STRT ビットに "0" を書き込むと動作を開始または再起動し , STOP ビットに "0" を書き込むと動作を強制的に停止させます。ただし , 2 つ of ビットに書き込まれる値が同じ値でなければ , どちら of ビットもその動作を実行しません。ビット操作命令以外の命令 (バイト以上の命令) を使用中の場合 , 書き込みは次のビット of 組み合わせに限られます。

表 11.4-3 スタートビットとストップビット of 機能

機能	STRT	STOP
タイマまたはパルス幅測定 of 起動 / 再起動	0	1
タイマまたはパルス幅測定 of 強制停止	1	0

ビット操作命令 (クリアビット命令) を使用中の場合 , 上記 of 組み合わせ of 書き込みがハードウェアによって自動的に書き込まれるので , 特に心配する必要はありません。

#### 開始後の動作

- タイマモード : カウント of 動作を直ちに開始します。
- パルス幅測定モード : 測定を開始するのは , 測定開始エッジ of 入力以後です。  
測定開始エッジを検出した後に , 16 ビットアップカウントタイマが 0000<sub>H</sub> にクリアされ , カウントを開始します。

#### タイマ of 再起動

タイマモードまたはパルス幅測定モードでタイマを起動した後に , タイマ of 動作が継続している最中に , 起動をかける (STRT ビットに "0" を書き込む) ことをタイマ of 再起動と呼びます。再起動時に実行する動作は次のモードに応じて変化します。

- ワンショットタイマモード : 動作には影響がおよびません。
- リロードタイマモード : リロードを実行し , 動作を続行します。  
オーバフロー発生と同時にタイマを再起動すると , オーバフローフラグ (OVIR) がセットされ , POUT ビットが反転します。
- パルス幅測定モード : 測定開始エッジ待ち of 状態の場合 , 動作に影響ありません。  
測定中 of 場合 , カウントが停止して , タイマ of 状態が「測定開始エッジ待ち」状態に戻ります。測定終了と同時にタイマを再起動すると , 測定終了フラグ (EDIR) がセットされ , 連続測定モードでは , 測定結果が PWCR の方へ転送されます。

### タイマの停止

ワンショットタイマモードまたは単発測定モードでは、測定が自動的に中断するのがタイマのオーバフロー時またはカウントの終了時であるため、タイマを意識して停止させる必要はありません。ただし、その他のモードでは、タイマを強制的に停止させる必要があります。

### 動作状態の確認

すでに解説したSTRTビットとSTOPビットは、読出し時にタイマの動作状態を表示するビットとして機能します。表示される値は表 11.4-4の内容を示しています。

表 11.4-4 動作状態表示ビットの機能

STRT	STOP	動作状態
0	0	タイマの停止中（測定開始エッジの待ち機時を除く）。 タイマが起動されていない、または測定が終了したことを表しています。
1	1	測定開始エッジ待ち状態、またはタイマカウント動作中

読出し時はSTRTビットとSTOPビットの両方は同じ値となります。ただし、（ビット操作命令などの）リードモディファイライト系統の命令による読出し時には、ビットの値が常に"1<sub>B</sub>"となるため、ビットの値を読み出すためにこれらの命令は使用しないでください。

### タイマのクリア

次の場合には、16ビットアップカウントタイマが0000<sub>H</sub>にクリアされます。

- リセット時
- パルス幅測定モードでカウント開始エッジの検出後にカウントを開始する場合

## 11.5 タイマモードの動作の詳細

---

タイマモードには、ワンショット動作モードと、リロード動作モードがあります。

---

### ワンショット動作モード

このモードでタイマを起動すると、カウントクロックのたびにカウントアップします。FFFF<sub>H</sub> ~ 0000<sub>H</sub>へカウントする際のオーバーフロー発生によって、タイマは自動的に停止します。

タイマの起動前にPWCRがセットされると、このセットした値からカウントを開始します。この場合、セットした値は消去され、PWCRの中には現行のカウント値が残ります。

PWCSRのbit 8(POUT)が反転するのは、オーバーフローが生じる場合ですが、このモードでは端子からその値が出力されません。パルス出力ありモードを指定した場合でも同様です。

### リロード動作モード

このモードでタイマを起動すると、PWCRのリロード値がタイマにセットされ、カウントクロックのたびにカウントアップされます。FFFF<sub>H</sub> ~ 0000<sub>H</sub>へカウントする際のオーバーフロー発生によって、PWCRのリロード値がタイマに再度セットされ、PWCSRのPOUTビット(bit 8)が反転し、カウント操作をくりかえします。PWCSRのSTOPビットに書き込みが行われて強制的に停止するか、またはリセットが行われるまで、タイマは停止しません。

タイマの起動前にPWCRにセットされたリロード値は、カウントの最中に格納され、タイマの起動時または再起動時、およびオーバーフローが生じる際には必ず、タイマにセットされます。カウントの最中にセットした値を変更する場合は、次のオーバーフローの発生、またはタイマの再起動の際に新規のリロード値が有効となります。

### タイマ値とリロード値

ワンショット動作モードでは、PWCRへの直接アクセスによりアップカウントタイマにアクセスします。PWCRに値を書き込むと、タイマに値が直接書き込まれ、カウント動作の最中にPWCRを読み出すと、現行のタイマの値が読み出されます。タイマの起動前にPWCRに値をセットすると、指定された値からタイマによりカウントが開始されます。

リロード動作モードでは、アップカウンタにアクセスすることが不可能となり、PWCRはリロードレジスタとして機能します(リロード値を格納する)。タイマの起動時または再起動時、およびオーバーフローが生じる際には必ず、PWCRに書き込まれた値がタイマにセットされます。PWCRを読み出すと、格納されたリロード値が読み出されます。

PWCRの値とタイマの値が定まらないのは、リロードモードで動作を強制的に中断させた後にタイマをワンショットモードに設定した場合です。したがって、タイマの使用前に必ず値をセットしてください。

PWCRの値が定まらないのは、ワンショットモードで動作を強制的に中断させた後にタイマをリロードモードに設定した場合です。したがって、タイマの使用前に必ず値をセットしてください。



### 割込み要求の発生

タイマモードで動作中の場合は、オーバフローによって割込み要求を発生させることができます。タイマのカウンタアップよりオーバフローが生じると、オーバフローフラグがセットされ、オーバフロー割込み要求が許可されると、割込み要求が発生します。

### タイマの周期

PWCRに0000<sub>H</sub>をセットした後にワンショットモードでタイマを起動する場合は、カウンタが65536を超えるとタイマのオーバフローが生じてカウンタが中断します。以下の式によりタイマの起動から停止までの時間を計算します。

$$T_1 = (65536 - n_1) \times t \quad \left\{ \begin{array}{l} T_1 \dots \text{起動から停止までの時間 (}\mu\text{s)} \\ n_1 \dots \text{タイマの起動時に PWCR にセットされたタイマの値} \\ t \dots \text{カウンタクロック周期 (}\mu\text{s)} \end{array} \right.$$

PWCRに0000<sub>H</sub>をセットした後にリロードモードでタイマを起動すると、カウンタが65536を超えるたびにタイマのオーバフローが生じます。以下の式により、リロードの周期とPOT端子出力パルスの周期を計算します。

$$\begin{array}{l} T_R = (65536 - n_R) \times t \\ T_{\text{pout}} = T_R \times 2 \end{array} \quad \left\{ \begin{array}{l} T_R \dots \text{リロード周期 (オーバフロー周期) (}\mu\text{s)} \\ T_{\text{pout}} \dots \text{POT 端子出力パルスの周期 (}\mu\text{s)} \\ n_R \dots \text{PWCR に格納されたリロード値} \\ t \dots \text{カウンタクロック周期 (}\mu\text{s)} \end{array} \right.$$

### カウンタクロックと最大周期

タイマモードでは、PWCRに0000<sub>H</sub>がセットされると最大周期になります。

表 11.5-1に、16MHzのマシンサイクル(表の中ではφで示される)に対応するカウンタクロック周期と最大タイマ周期を示します。

表 11.5-1 カウンタクロックと周期

カウンタクロックの選択	CKS1,0=00 <sub>B</sub> (φ/4)時	CKS1,0=01 <sub>B</sub> (φ/16)時	CKS1,0=10 <sub>B</sub> (φ/32)時
カウンタクロック周期	0.25μs	1μs	2μs
最大タイマ周期	16.38ms	65.5ms	131.1ms

## 11.6 タイマモード動作のフローチャート

図 11.6-1に、タイマモード動作のフローチャートを示します。

タイマモード動作のフローチャート

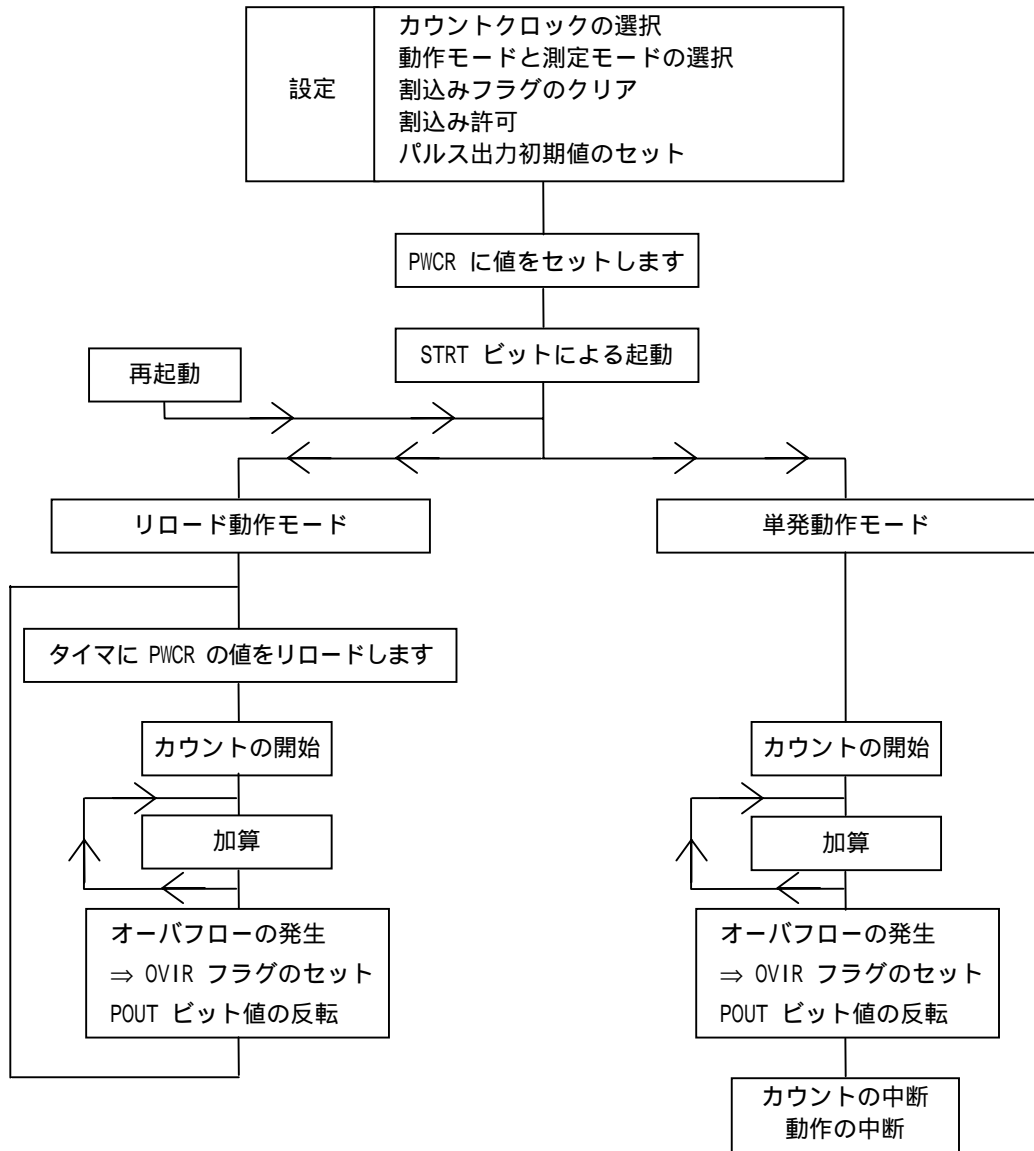


図 11.6-1 タイマモード動作のフローチャート

## 11.7 パルス幅測定モード動作の詳細

---

パルス幅測定用の信号は、PWC端子より入力します。

パルス幅測定モードには、1回のみカウントを実行する単発測定モードと、連続的にパルス幅を測定する連続測定モードがあります。

---

### 単発測定モードと連続測定モード

以下に、単発測定モードと連続測定モードの相違点を示します。

#### 単発測定モード

最初のカウント終了エッジが入力されると、タイマがカウントを中断し、PWCSR のカウント終了フラグ (EDIR) がセットされ、以降は測定しません。ただし、同時にタイマの再起動が指定されると、タイマの状態は「測定開始エッジを待機する」状態に移行します。

#### 連続測定モード

#### 【"H"/"L"パルス幅測定モードの場合】

カウント終了エッジが入力されると、PWCSR のカウント終了フラグ (EDIR) をセットし、タイマのカウント結果がPWCRに転送され、タイマはフリーラン状態でカウントアップし続ける場合があります。次のカウント開始エッジを入力すると、タイマを0000<sub>H</sub>にクリアし、パルス幅カウントを開始します。

### < 注意事項 >

---

カウント終了エッジが入力され、タイマがフリーラン状態になることにより、タイマがオーバーフローすることがあり、OVIRフラグがセットされることがありますので、「H"/"L"パルス幅測定モードでは、OVIRフラグを用いたパルス幅時間測定は行わないでください。

---

#### 【全エッジ間パルス幅測定モード/分周周期測定モード/立ち上がりエッジ間測定モード/立ち下がりエッジ間測定モードの場合】

カウントの終了エッジ（カウント開始エッジ）が入力されると、PWCSRのカウント終了フラグ(EDIR)をセットしタイマのカウント結果がPWCRに転送され、タイマを0000<sub>H</sub>にクリアし、カウントを再開します。

## 測定結果データ

測定結果，タイマの値，PWCRの機能は，単発測定モードと連続測定モードではその取り扱い方が異なっています。その違いは次のとおりです。

## 単発測定モード

タイマ動作中にPWCRを読み出すと，現行のタイマの値が読み出されます。  
測定の終了後にPWCRを読み出すと，測定結果が読み出されます。

## 連続測定モード

測定の終了時には，タイマの測定結果がPWCRの方へ転送されます。  
PWCRを読み出すと，前の測定結果が読み出されます。測定が進行中の間は，引き続きPWCRに前の測定結果が格納されます。測定中に，タイマの値を読み出すことはできません。

連続測定モードでは，次の測定が完了する前に，前の測定結果を読み出していないと，新規の測定結果が既存の値を上書きします。このとき，PWCSRのエラーフラグ(ERR)がセットされます。PWCRの読出し時にはエラーフラグ(ERR)が自動的にクリアされます。

## 最小入力パルス幅

パルス幅カウント入力端子(PWC)へのパルス入力は，次に示す最小入力パルス幅よりも長くする必要があります。

パルス幅：2マシンサイクル(16MHzのマシンクロックの場合は，0.125μs以上となる)

ただし，上記の仕様よりも短い入力パルスが有効なパルスとして認識される場合もあります。

## パルス幅/周期の計算

PWCRから読み出されたカウント結果にもとづいて，測定対象のパルス幅またはパルス周期をカウントの終了後に次のように計算します。

$$T_w = n \times t \div D_{IV} (\mu s)$$

{

$T_w$  ... 測定パルス幅またはパルス周期(μs)

$n$  ... PWC に格納された測定結果

$t$  ... カウントクロック周期(μs)

$D_{IV}$  ... 分周比レジスタ(DIVR)にセットした分周比  
(分周カウントモード以外のモードでは値1を使用する)。

## パルス幅/周期測定の範囲

測定可能なパルス幅/周期の範囲は，カウントクロックと入力分周器の分周比に応じて変化します。

表 11.7-1に，16MHz のマシンサイクル( $\phi$ で表している)に対する測定範囲を示します。

表 11.7-1 パルス幅測定範囲

分周比	DIV1,0	CKS1, 0 = 00 <sub>B</sub> ( $\phi/4$ )	CKS1, 0 = 01 <sub>B</sub> ( $\phi/16$ )	CKS1, 0 = 10 <sub>B</sub> ( $\phi/32$ )
分周なし	-	0.125 $\mu$ s ~ 16.38ms [0.25 $\mu$ s]	0.125 $\mu$ s ~ 65.5ms [1.0 $\mu$ s]	0.125 $\mu$ s ~ 131ms [2.0 $\mu$ s]
4分周	00 <sub>B</sub>	0.125 $\mu$ s ~ 4.10ms [62.5ns]	0.125 $\mu$ s ~ 16.38ms [0.25 $\mu$ s]	0.125 $\mu$ s ~ 32.75ms [500ns]
16分周	01 <sub>B</sub>	0.125 $\mu$ s ~ 1024 $\mu$ s [15.6ns]	0.125 $\mu$ s ~ 4.10ms [62.5ns]	0.125 $\mu$ s ~ 8.19ms [125ns]
64分周	10 <sub>B</sub>	0.125 $\mu$ s ~ 256 $\mu$ s [3.91ns]	0.125 $\mu$ s ~ 1024 $\mu$ s [15.6ns]	0.125 $\mu$ s ~ 2.048ms [32.25ns]
256分周	11 <sub>B</sub>	0.125 $\mu$ s ~ 64 $\mu$ s [0.98ns]	0.125 $\mu$ s ~ 256 $\mu$ s [3.9ns]	0.125 $\mu$ s ~ 512ms [7.81ns]

(注) [ ]の中の数字は、ビット当たりの分解能を表します。

#### 割込み要求の発生

パルス幅測定モードでは、次の2つの割込み要求を発生させることができます。

##### タイマオーバフロー割込み要求

カウンティングの最中にオーバフローが生じると、オーバフローフラグがセットされます。オーバフロー割込み要求が許可された場合は、割込み要求が発生します。

##### 測定終了割込み要求

測定終了エッジの検出時には、PWCSRのカウント終了フラグ(EDIR)がセットされます。測定終了割込みが許可された場合は、割込み要求が発生します。

測定終了フラグ (EDIR)はPWCRの読出しによって自動的にクリアされます。

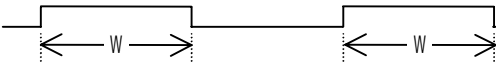

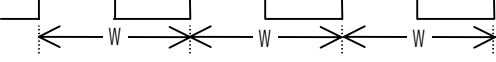

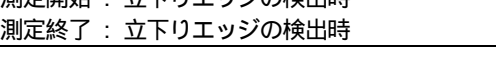
## 11.7.1 測定モードと測定動作

5つの異なるモードから、測定モードを選択することができます。このモードにより、測定すべき入力パルスの部分を決定します。高周波パルスの幅を精密に測定するために、入力パルスを指定された分周比で分周し、その結果得られた周期を測定するモードを使用することができます。

### 測定モードと測定動作

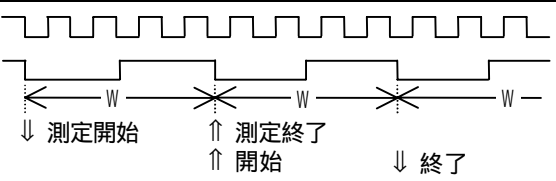
表 11.7-2に、測定モードの動作について説明します。

表 11.7-2 測定モードの動作（続く）

測定モード	MOD2	MOD1	MOD0	測定動作
Hパルス幅測定	1	0	1	 <p>↑ 測定開始 ↓ 測定終了    ↑ 開始    ↓ 測定終了</p> <p>"H" 周期の幅を測定します。 測定開始: 立上りエッジの検出時 測定終了: 立下りエッジの検出時</p>
Lパルス幅測定	1	1	0	 <p>↓ 測定開始 ↑ 測定終了    ↓ 開始    ↑ 測定停止</p> <p>"L" 周期の幅を測定します。 カウント (測定) 開始: 立下りエッジの検出時 カウント (測定) 終了: 立上りエッジの検出時</p>
立上りエッジ間 周期測定	1	0	0	 <p>↑ 測定開始    ↑ 測定終了    ↑ 開始    ↑ 終了    ↑ 開始    ↑ 終了</p> <p>立上りエッジ間の時間を測定します。 測定開始: 立上りエッジの検出時</p>
立下りエッジ間 周期の測定	1	1	1	 <p>↓ 測定開始    ↓ 測定終了    ↓ 開始    ↓ 終了    ↓ 開始    ↓ 終了</p> <p>立下りエッジ間の時間を測定します。 測定開始: 立下りエッジの検出時 測定終了: 立下りエッジの検出時</p>
全エッジパルス 幅測定	0	1	0	 <p>↑ 測定開始    ↓ 測定終了    ↓ 開始    ↑ 終了    ↑ 開始    ↓ 終了</p> <p>連続した入力エッジ間の幅を測定します。 測定開始: エッジの検出時 測定終了: エッジの検出時</p>

W: 測定中のパルス幅

表 11.7-2 測定モードの動作 ( 続き )

測定モード	MOD2	MOD1	MOD0	測定動作
分周測定	0	1	1	 <p>(下記の例では4で割る) 分周比レジスタ (DIVR) にセットした分周比で入力パルスを割り、測定周期を結果として得ます。 測定開始：動作を開始した後の立下りエッジの検出 測定終了：分割信号の 1 周期の終了</p>

W：測定中のパルス幅

すべてのモードでは、測定開始から測定開始エッジの入力までの間に、タイマによるカウントは行われません。測定開始エッジの入力後に、タイマは 0000<sub>H</sub> にクリアされ、測定終了エッジが入力されるまでカウントクロックのたびにカウントアップが行われます。

測定終了エッジの入力時には次の動作を実行します。

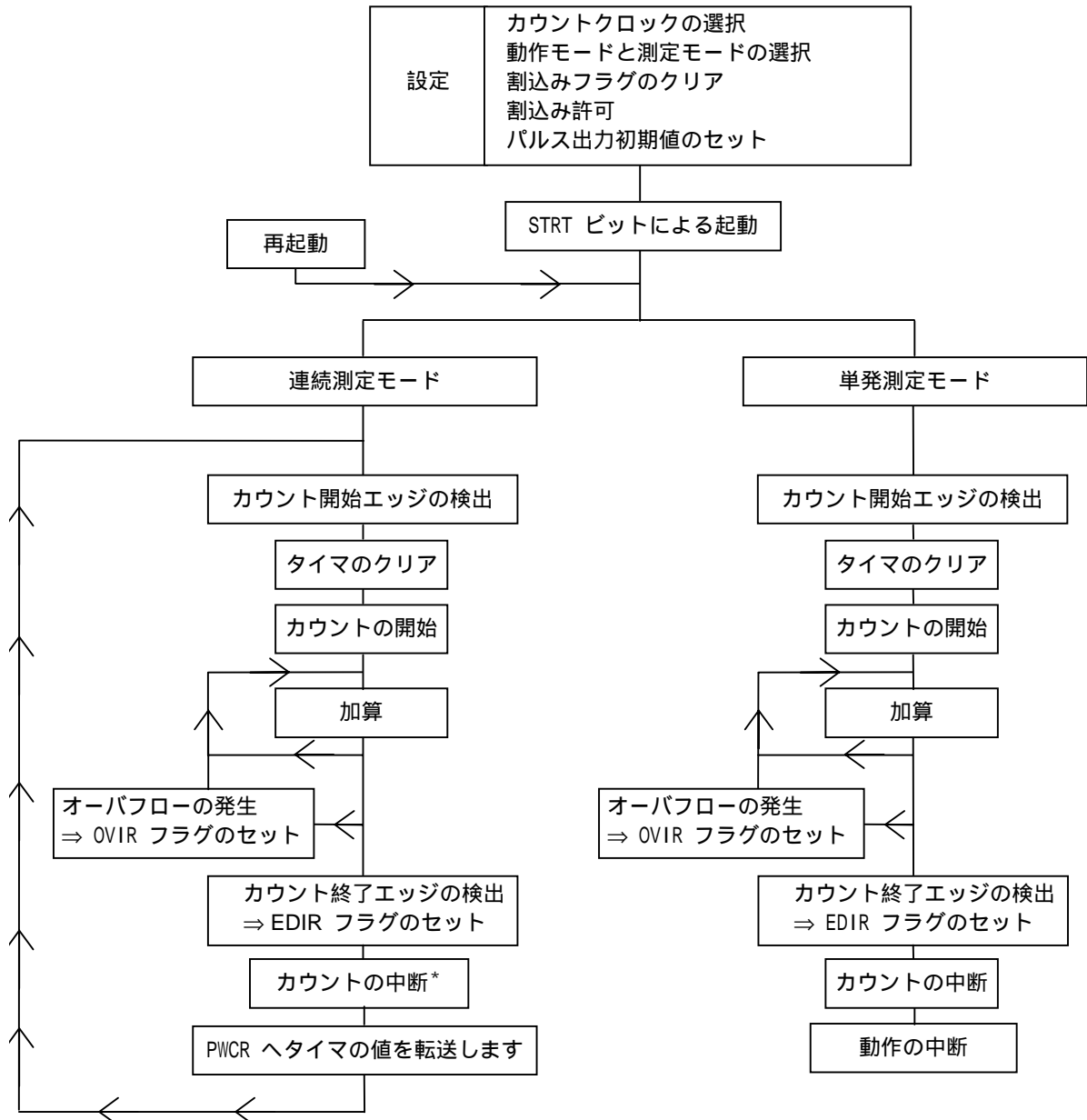
- 1) PWCSRのカウント終了フラグ(EDIR)がセットされます。
- 2) タイマのカウント動作が停止します(再起動と同時であった場合と"H"/"L"パルス幅測定連続測定モード時を除く)。
- 3) 連続測定モード時：タイマの値(測定結果)は、PWCRの方へ転送されます。
- 4) 単発測定モード時：測定を終了します(再起動と同時であった場合を除く)。

連続測定モードの場合で、全エッジ間パルス幅測定や、周期測定、立ち下がりエッジ間周期測定、立ち上がりエッジ間周期測定を行った場合、終了エッジが次の測定開始エッジとなります。

## 11.7.2 パルス幅測定動作のフローチャート

図 11.7-1に、パルス幅測定モード時の動作フローチャートを示します。

パルス幅測定動作のフローチャート



\* : "H"/"L"パルス幅測定連続測定モード時を除く。

図 11.7-1 パルス幅測定モード時の動作フローチャート



## 11.8 PWCタイマ取扱い上の注意事項

PWCタイマを取り扱う際には、次の内容に関して特に注意が必要です。

- ・レジスタの値の変更
- ・タイマモードでの測定終了フラグ
- ・PWCSRのSTRTビットとSTOPビット
- ・タイマのクリア
- ・クロック選択ビット
- ・モード変更時のPWCRとタイマの値
- ・最短入力パルス幅
- ・分周周期測定モード
- ・動作中の再起動
- ・連続測定モードを用いたパルス幅測定モード

### レジスタの値の変更

タイマの動作中に次の PWCSR ビットの値を変更することは禁止されます。ビットの値の変更はタイマの起動前または動作の中断後に限られます。

[bit 7, 6] CKS1, CKS0: クロック選択ビット

[bit 3] S/C: 測定モード (単発または連続) 選択ビット

[bit 2, 1, 0] MOD2, MOD1, MOD0: 動作モード・測定エッジ選択ビット

タイマの動作中にビットの書込みが行われても、パルス出力レベル表示ビット (POUT: bit 8) の値は変化しないことに注意してください。

タイマの動作中に DIVR の値を変更することは禁止されます。DIVR の値の変更はタイマの起動前または動作の停止後に行ってください。

### タイマモードでの測定終了フラグ

タイマモードでは、PWCSR の測定終了割込み要求フラグ (EDIR) の値に意味は持ちません。したがって PWCSR のカウント終了割込み要求 (EDIE) に対する許可ビットを必ず "0" に設定してください。

### PWCSRのSTRTビットとSTOPビット

これらの 2 つのビットは、ビットの読出しまたは、書込みに応じて違いが現れることに注意してください (レジスタの詳細での解説を参照)。

また、リードモディファイライト系統の命令により、ビットは必ず "11<sub>B</sub>" として読み出されることに注意してください。したがって、動作状態を読み出すためにビット操作命令を使用することができません。

ただし、STRT ビットまたは STOP ビットへの書込みによりタイマの起動または停止を行うために、(ビットクリア命令などの) ビット操作命令を使用することが可能です。

### タイマのクリア

パルス幅測定モードでは、測定開始エッジによってタイマをクリアしているので、タイマの以前のデータには意味がありません。

### クロック選択ビット

PWCSR のクロック選択ビット (CKS 1, CKS0: bit 7, 6) に "11<sub>B</sub>" をセットすることは禁止されています。

### モード変更時のPWCRとタイマの値

PWCRの値とタイマの値が決定するのは、リロードタイマモードで動作を強制的に終了した後にタイマをワンショットモードに設定した場合です。したがって、タイマの使用後に値を必ずセットしてください。

PWCR の値が定まらないのは、ワンショットモードで動作を強制的に中断させた後にタイマをリロードタイマモードに設定した場合です。したがって、タイマの使用前に値を必ずセットしてください。

パルス幅測定モードからタイマモードへモードを変更する場合、タイマの起動前に値を PWCR に必ずセットしてください。

### 最短入力パルス幅

パルス幅測定入力端子へのパルス入力には、以下の制御があります。

#### 最小パルス幅

マシンサイクルの2分周(16MHzのマシンサイクルの場合は、0.125 $\mu$ s以上)

#### 最大入力周波数

マシンサイクルの4分周(16MHzのマシンサイクルの場合は、4MHz以上)

これ以上小さいパルス幅や大きい周波数が入力された場合には、タイマの動作が保証されていません。上記の制約を守らないノイズが入力信号に現れる恐れがある場合は、ノイズを抑制してください。

### 分周周期測定モード

パルス幅測定モードで分周周期測定モードの使用時に入力パルスの分周が行われているため、カウント結果から計算されたパルス幅は平均値となることに注意してください。

### 動作中の再起動

カウント動作の開始後にタイマを再起動すると、タイミングに応じて以下のような動作が生じる場合があります。

リロードタイマモードでオーバフローと同時にタイマを再起動する場合。

タイマは再起動しますが、オーバフローフラグ(OVIR)がセットされ、POUTビットが反転します(すなわち、通常のオーバフローと同じ動作が実行されます)。

ワンショットパルス幅測定モードで測定終了エッジと同時にタイマを再起動する場合。

タイマが再起動されて、測定開始エッジ待ち状態となりますが、測定終了フラグ(EDIR)もセットされます。

連続パルス幅測定モードで測定終了エッジと同時にタイマを再起動する場合。

タイマが再起動されて測定開始エッジ待ち状態となり、カウント終了フラグ(EDIR)がセットされ、その時点での測定結果がPWCRの方へ転送されます。

動作中にタイマを再起動する場合には、上記のようにフラグの動作に注意して、割込みや、その他の制御を実行してください。

### 連続測定モードを用いたパルス幅測定モード

このモードの連続測定中は、内部カウントクロック用の分周回路がクリアされないで、カウントクロックを下回るエッジの数が、カウント結果に加算されます。



## 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の動作
- 12.5 16ビットアウトプットコンペアの動作
- 12.6 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2本、インプットキャプチャ4本のモジュールから構成されています。

本機能を使用すると、16ビットフリーランタイマをベースに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： $\phi/4$ 、 $\phi/16$ 、 $\phi/64$ 、 $\phi/256$

$\phi$ ：マシニングクロック

割込み

割込みは、16ビットフリーランタイマのカウンタ値のオーバーフロー、コンペアレジスタ0とのコンペアマッチにより発生することができます。（コンペアマッチはモード設定が必要です）

カウンタ値

16ビットフリーランタイマのカウンタ値がオーバーフローした場合、または、コンペアレジスタ0との一致が生じた場合に、割り込みを発生させることができます（モードの設定により、コンペアマッチを使用することができます）。

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"0000<sub>H</sub>"に初期化することができます。

### アウトプットコンペア(×2)

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。

- 2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
- 2本のコンペアレジスタをペアにして出力端子を制御することができます。
  - コンペアレジスタ2本を使用して出力端子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 (x4)

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、それぞれが4本の独立した外部入力端子と対応します。キャプチャレジスタには、16ビットフリーランタイムの値を格納することができます。さらに、外部端子からの信号入力のエッジ検出と同時に割り込みが発生します。

- 外部入力信号の検出エッジを選択可能です。
  - 立上がりエッジ、立下がりエッジ、両エッジから選択可能
- 4本のインプットキャプチャは独立して動作可能です。
- 割り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り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。

## 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

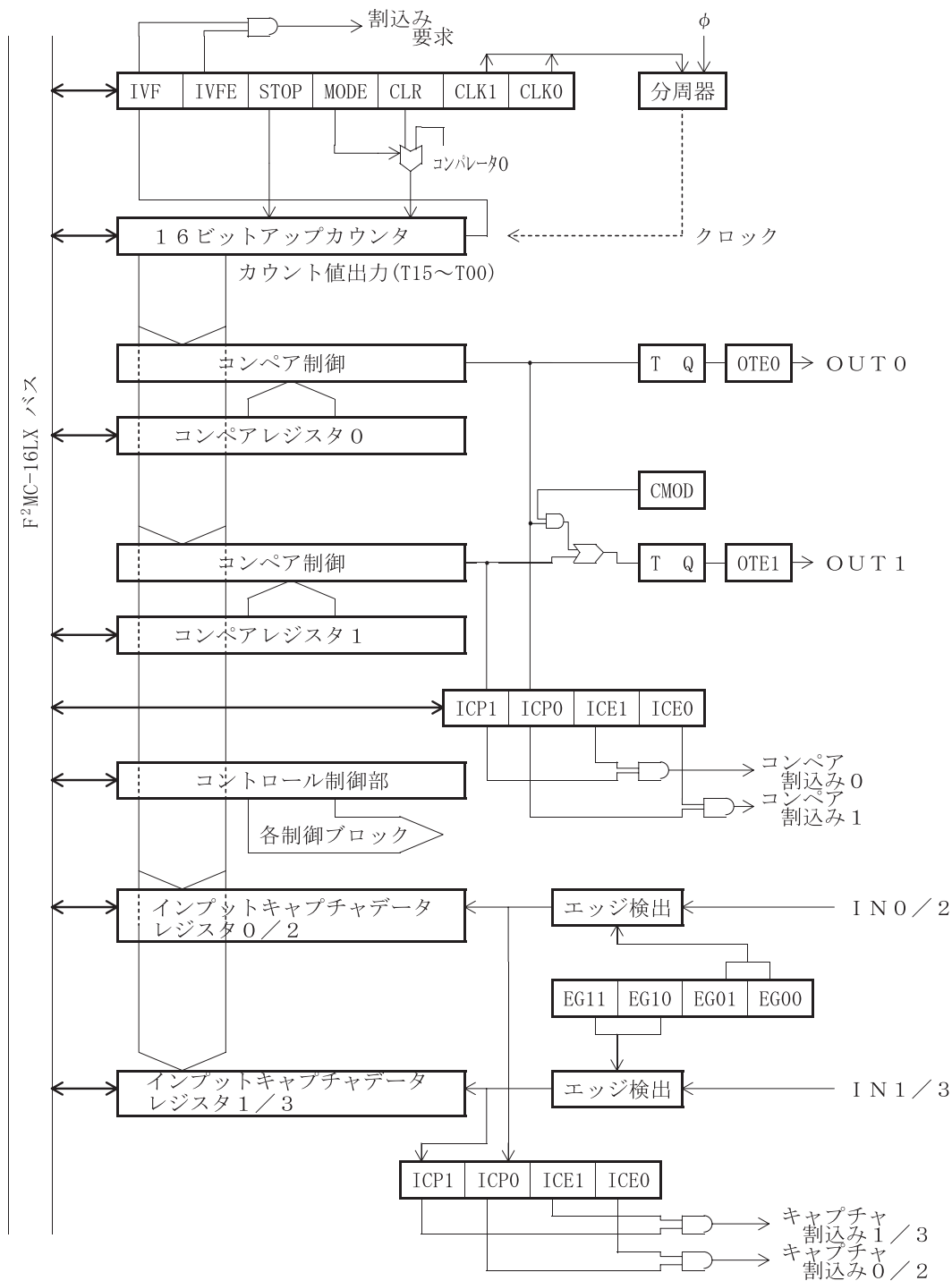


図 12.2-1 16ビット入出力タイマのブロックダイアグラム

## 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つがあります。

- ・タイマデータレジスタ(TCDTH,TCDTL)
- ・タイマコントロールステータスレジスタ(TCCS)
- ・コンペアレジスタ(OCCP0,1)
- ・コンペアコントロールステータスレジスタ(OCS0,1)
- ・インプットキャプチャレジスタ(IPCP0~3)
- ・コントロールステータスレジスタ(ICS01,ICS23)

### 16ビット入出力タイマのレジスタ

タイマ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: 00006D <sub>H</sub>	T15	T14	T13	T12	T11	T10	T09	T08	TCDTH
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
タイマデータレジスタ下位	7	6	5	4	3	2	1	0	⇐ビットNo.
アドレス: 00006C <sub>H</sub>	T07	T06	T05	T04	T03	T02	T01	T00	TCDTL
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
タイマ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: 00006E <sub>H</sub>	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)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
コンペアレジスタ上位	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス: ch0 00005B <sub>H</sub> ch1 00005D <sub>H</sub>	C15	C14	C13	C12	C11	C10	C09	C08	OCCP0 OCCP1
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
コンペアレジスタ下位	7	6	5	4	3	2	1	0	⇐ビットNo.
アドレス: ch0 00005A <sub>H</sub> ch1 00005C <sub>H</sub>	C07	C06	C05	C04	C03	C02	C01	C00	OCCP0 OCCP1
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
コンペアコントロールステータスレジスタ 1	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス: ch1 00005F <sub>H</sub>	—	—	—	CMOD	OTE1	OTE0	OTD1	OTD0	OCS1
リード/ライト ⇒	(-)	(-)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(-)	(-)	(-)	(0)	(0)	(0)	(0)	(0)	
コンペアコントロールステータスレジスタ 0	7	6	5	4	3	2	1	0	⇐ビットNo.
アドレス: ch0 00005E <sub>H</sub>	ICP1	ICP0	ICE1	ICE0	—	—	CST1	CST0	OCS0
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)



## 第12章 16ビット入出力タイマ

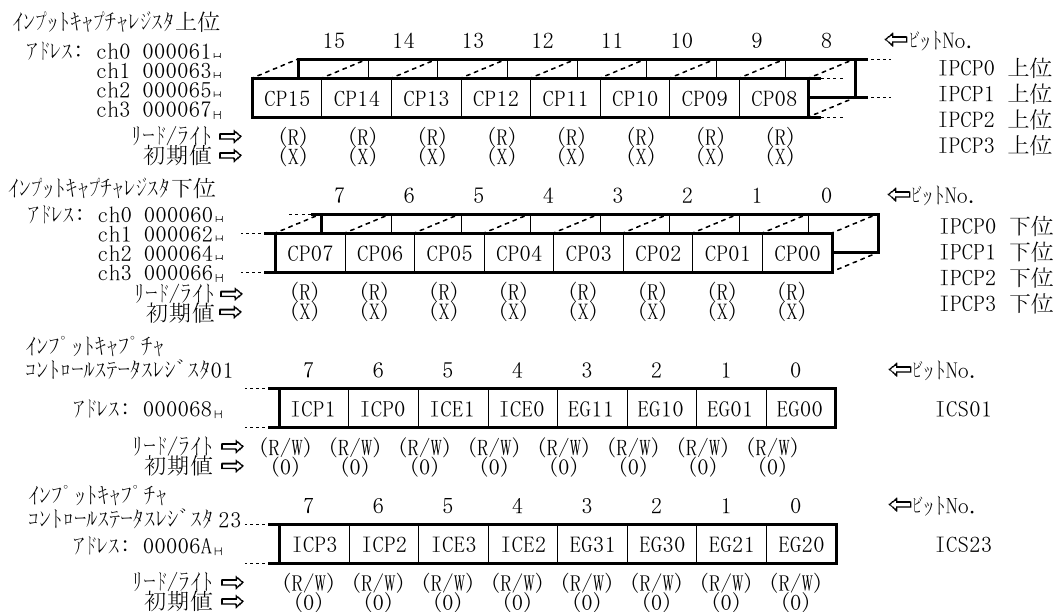


図 12.3-1 16ビット入出力タイマのレジスタ (続き)

## 12.3.1 16ビットフリーランタイム

16ビットフリーランタイムのレジスタには、次の2つがあります。

- ・タイマデータレジスタ(TCDTH, TCDTL)
- ・タイマコントロールステータスレジスタ(TCCS)

タイマデータレジスタ(TCDTH, TCDTL)

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

16ビットフリーランタイムの初期化は次の要因で行われます。

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

タイマ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: 00006D <sub>H</sub>	T15	T14	T13	T12	T11	T10	T09	T08	TCDT
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)	
タイマ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: 00006C <sub>H</sub>	T07	T06	T05	T04	T03	T02	T01	T00	TCDT
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 12.3-2 タイマデータレジスタ

### <注意事項>

データレジスタは、ワードアクセスしてください。

タイマコントロールステータスレジスタ(TCCS)

タイマ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: 00006E <sub>H</sub>	予約	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0	TCCS
リード/ライト ⇒ 初期値 ⇒	(-) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

図 12.3-3 コントロール・ステータスレジスタ

### 【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マの割込み要求フラグです。

16ビットフリーランタイマ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 12.3-1 IVF( 割込み要求フラグ)の機能

IVF	機能
0	割込み要求なし（初期値）
1	割込み要求あり

## 【ビット5】IVFE

IVFEは、16ビットフリーランタイマの割込み許可ビットです。

本ビットが"1"の時、IVFビット（ビット6）が"1"にセットされると割込みが発生します。

表 12.3-2 IVFE(割込み許可ビット)の機能

IVFE	機能
0	割込み禁止（初期値）
1	割込み許可

## 【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"書込み時にタイマのカウント停止。"0"書込み時にタイマのカウントを開始。

表 12.3-3 STOP(カウントを停止ビット)の機能

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

（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"の時はリセットとCLR ビット（ビット2）でカウンタ値を初期化可能です。"1"の時はリセットと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 12.3-4 MODE(初期化条件設定ビット)の機能

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

（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ム値を"0000<sub>H</sub>"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<sub>H</sub>"に初期化します。"0"を書込んだりも意味を持ちません。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タ値の変化点で行われます。

表 12.3-5 CLR(初期化ビット)の機能

CLR	機能
0	意味を持ちません(初期値)
1	カウンタ値を"0000 <sub>H</sub> "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<sub>H</sub>"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ム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 12.3-6 CLK1, CLK0(カウントクロック選択ビット)

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

$\phi$  = マシンクロック

## 12.3.2 アウトプットコンペア

アウトプットコンペアには、次の2つのレジスタがあります。

- ・コンペアレジスタ(OCCP0,1)
- ・コントロールステータスレジスタ(OCS0,1)

コンペアレジスタ(OCCP0,1)

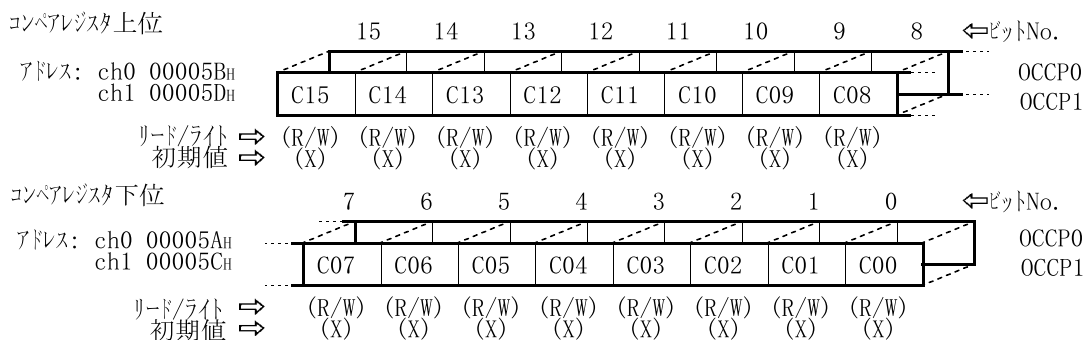


図 12.3-4 コンペアレジスタ

### <注意事項>

コンペアレジスタは、ワードアクセスしてください。

コンペアレジスタは、16ビットフリーランタイムと比較する16ビット長のコンペアレジスタです。レジスタ値は初期値不定ですので設定してから起動許可してください。本レジスタ値と16ビットフリーランタイム値が一致した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。

コントロールステータスレジスタ(OCS0,1)

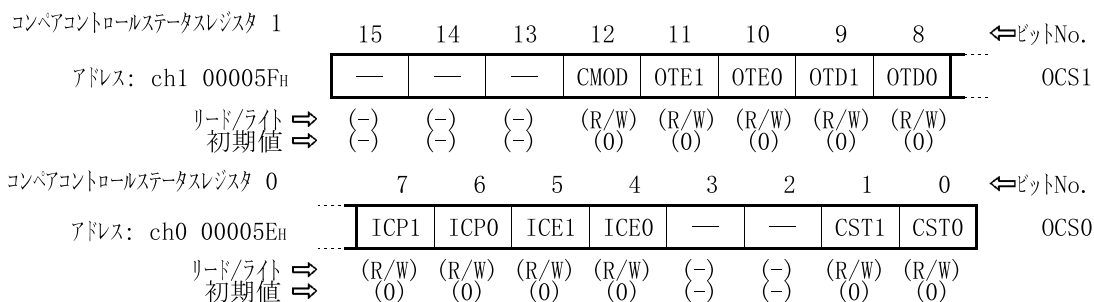


図 12.3-5 コントロールステータスレジスタ

## 【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替えます。

CMOD=0のとき（初期値）

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

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

CMOD=1のとき

CMOD=1のときは、コンペアレジスタ0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が同値の時はコンペアレジスタ1本のときと同じ動作をします。

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

## 【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 12.3-7 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。

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

## 【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 12.3-8 OTD1，OTD0(端子出力レベル変更ビット)の機能

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

OTD1：アウトプットコンペア1に対応，OTD0：アウトプットコンペア0に対応

## 【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 12.3-9 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

ICP1 : アウトプットコンペア1に対応, ICP0 : アウトプットコンペア0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"の時割込みフラグ(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 12.3-10 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

ICE1 : アウトプットコンペア1に対応

ICE0 : アウトプットコンペア0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。

表 12.3-11 CST1, CST0(16ビットフリーランタイムとの一致動作許可ビット)

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

・コンペア動作許可をする前にコンペアレジスタ値を設定してください。

・CST1 : アウトプットコンペア1に対応

・CST0 : アウトプットコンペア0に対応

## &lt; 注意事項 &gt;

アウトプットコンペアは、16ビットフリーランタイムのクロックと同期させているため、16ビットフリーランタイムを停止させるとコンペア動作も停止します。

### 12.3.3 インพุットキャプチャ

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。

- ・インพุットキャプチャデータレジスタ(IPCP0~3)
- ・コントロール・ステータスレジスタ(ICS23/01)

インพุットキャプチャデータレジスタ(IPCP0~3)

インพุットキャプチャデータレジスタ(IPCP0~3)は、対応した外部端子入力波形の有効エッジを検出したとき、16ビットフリーランタイム値を保持するレジスタです。

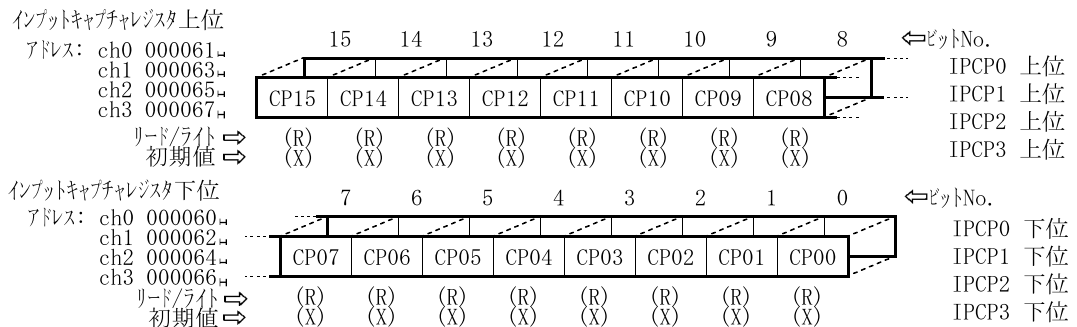


図 12.3-6 インพุットキャプチャデータレジスタ(IPCP0~3)

#### < 注意事項 >

インพุットキャプチャデータレジスタ(IPCP0~3)は、ワードアクセスしてください。書き込みはできません。

コントロールステータスレジスタ(ICS23/01)

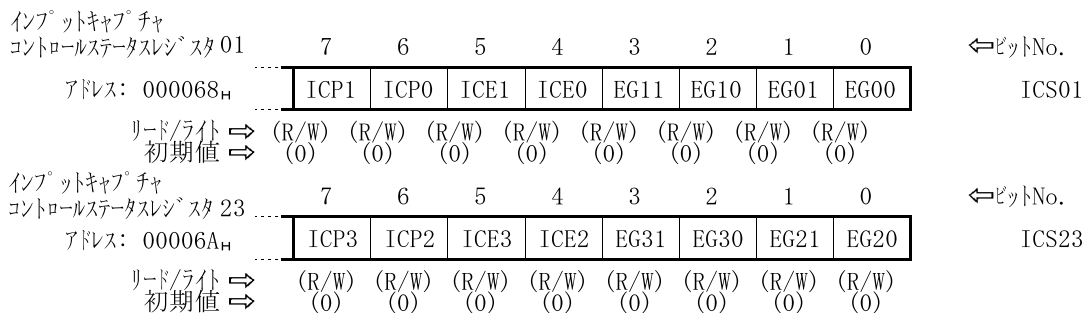


図 12.3-7 コントロール・ステータスレジスタ(ICS23/01)

#### < 注意事項 >

コントロール・ステータスレジスタ(ICS23/01)は、バイトアクセスしてください。



## 【ビット7, 6】ICPx(x: ch番号)

ICPxは、インプットキャプチャ割込みフラグです。

外部入力端子の有効エッジを検出すると本ビットを"1"にセットします。割込み許可ビット(ICEx) 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 12.3-12 ICPx(インプットキャプチャ割込みフラグ)の機能

ICPx	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

## 【ビット5, 4】ICEx(xはch番号です。)

ICExは、インプットキャプチャ割込み許可ビットです。本ビットが"1"のとき割込みフラグ(ICPx) 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 12.3-13 ICEx(インプットキャプチャ割込み許可ビット)の機能

ICEx	機能
0	割込み禁止(初期値)
1	割込み許可

## 【ビット3, 2, 1, 0】EGx1, EGx0(x: ch番号)

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

表 12.3-14 EGx1, EGx0(外部入力の有効エッジ極性指定ビット)の機能

EGx1	EGx0	エッジ検出極性
0	0	エッジ検出なし(停止状態)(初期値)
0	1	立上りエッジ検出
1	0	立下りエッジ検出
1	1	両エッジ検出

## 12.4 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"0000<sub>H</sub>"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とき。
- アウトプットコンペアレジスタ0値とコンペアマッチしたとき（モード設定が必要）
- 動作中にTCCSレジスタのCLRビットに"1"を書き込んだとき。
- 停止中にTCDCレジスタに"0000<sub>H</sub>"を書き込んだとき。
- リセット時

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

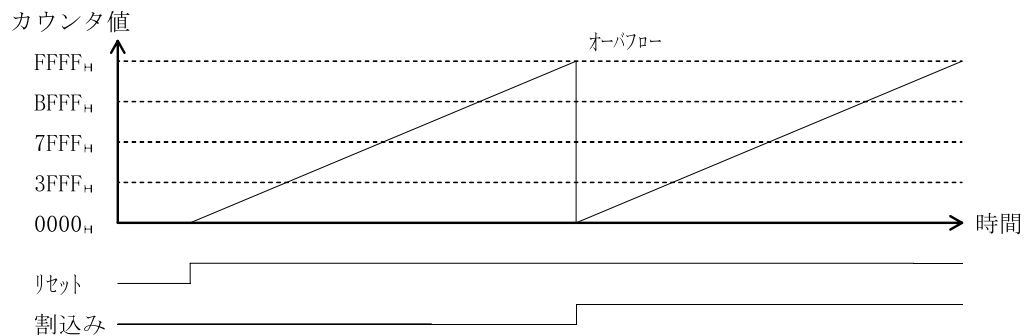


図 12.4-1 オーバフローによるカウンタクリア

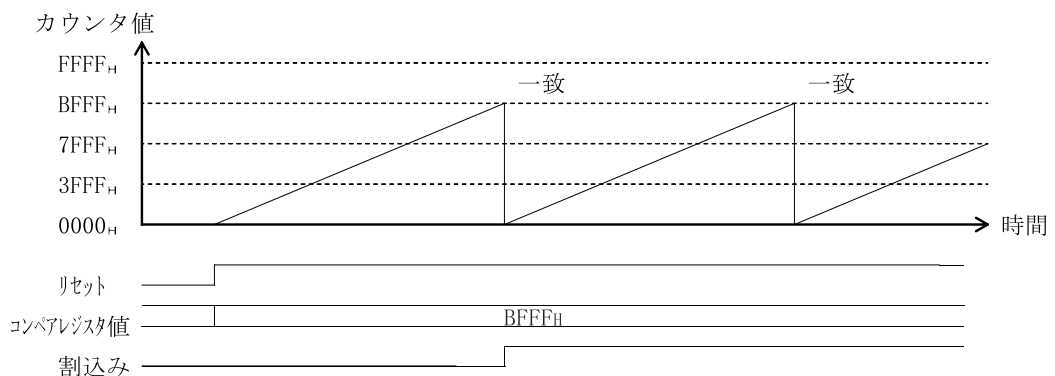


図 12.4-2 アウトプットコンペアレジスタ0値とコンペアマッチしたときのカウンタクリア

16ビットフリーランタイマのカウントタイミング

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との一致によって，カウンタがクリアされます。

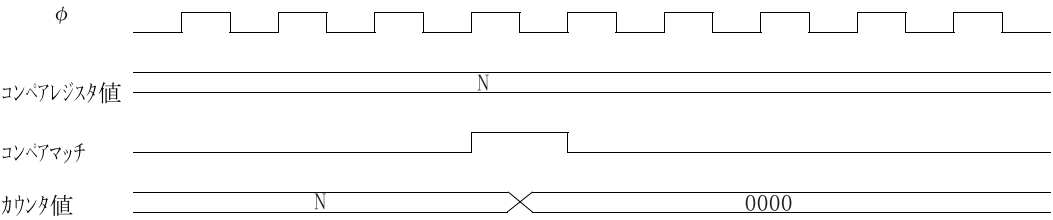


図 12.4-3 フリーランタイマのクリアタイミング (コンペアレジスタ0との一致)

## 12.5 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

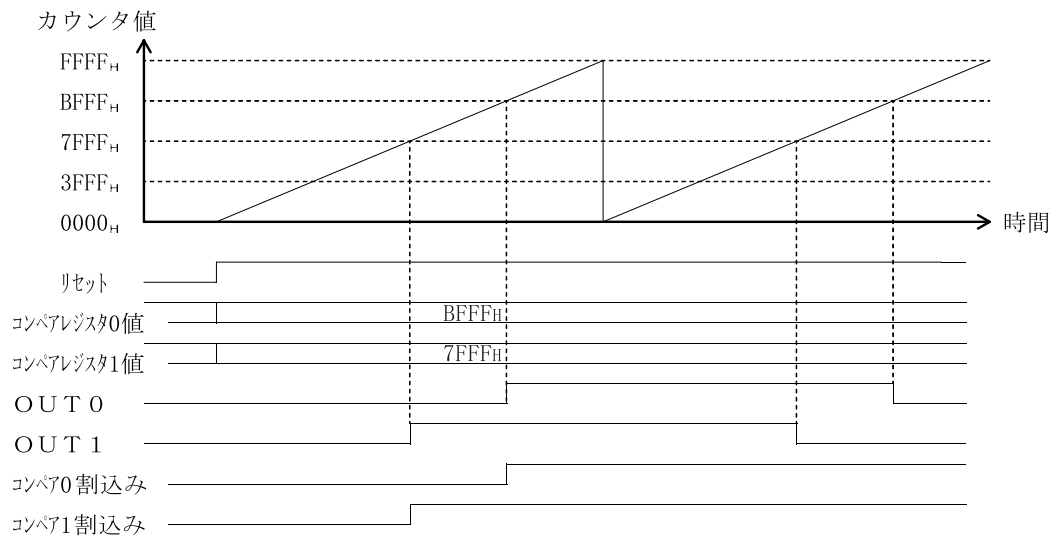


図 12.5-1 コンペアレジスタ0,1 を使用したときの出力波形例（出力の初期値は0とする）

図 12.5-2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)

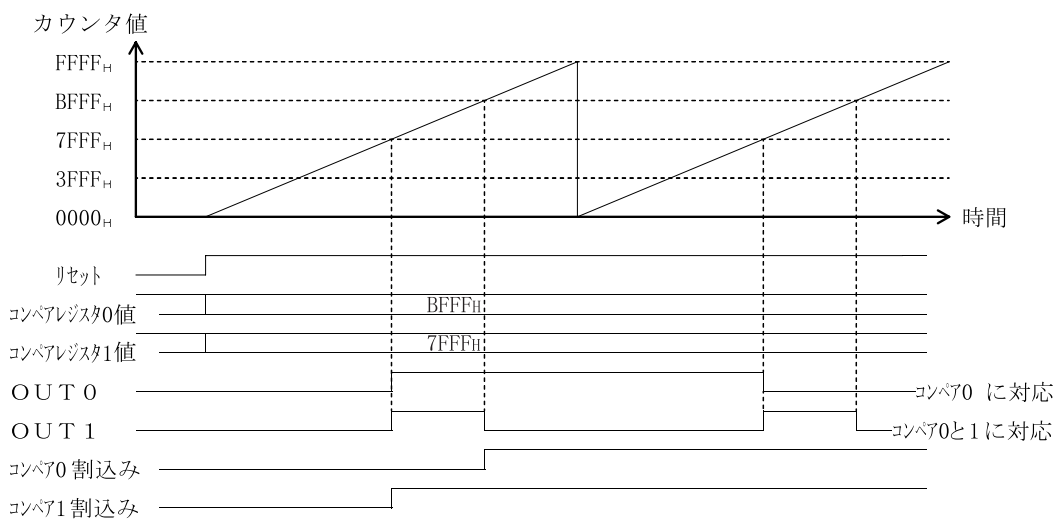


図 12.5-2 2組のコンペアレジスタによる出力波形例（出力の初期値は0とする）

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が発生することができます。

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

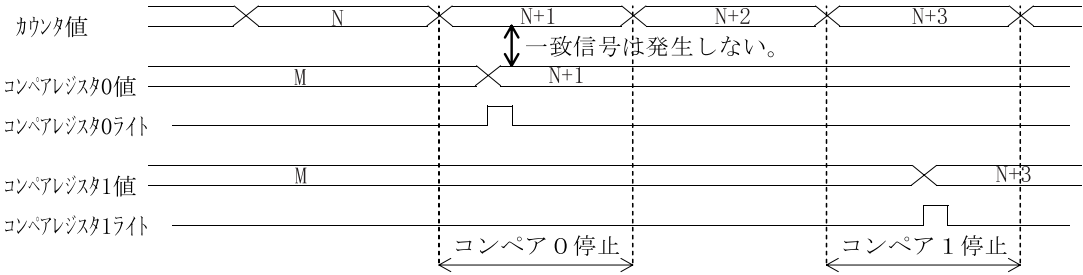


図 12.5-3 コンペアレジスタ書換え時のコンペア動作

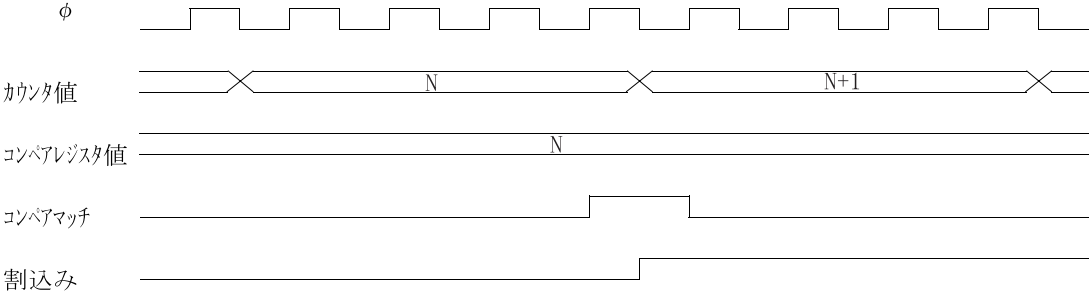


図 12.5-4 割込みタイミング

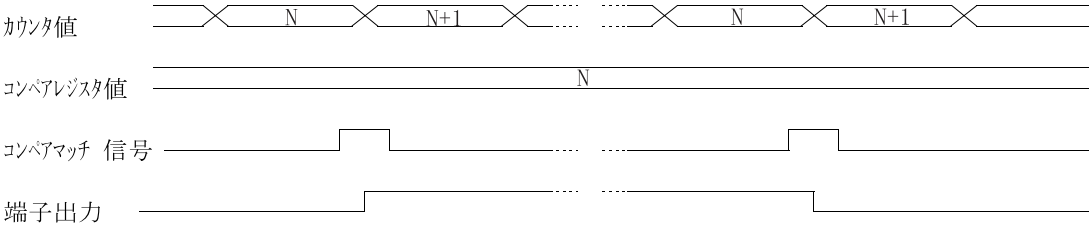


図 12.5-5 出力端子変化タイミング

# 12.6 16ビットインプットキャプチャの動作

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

16ビットインプットキャプチャの動作

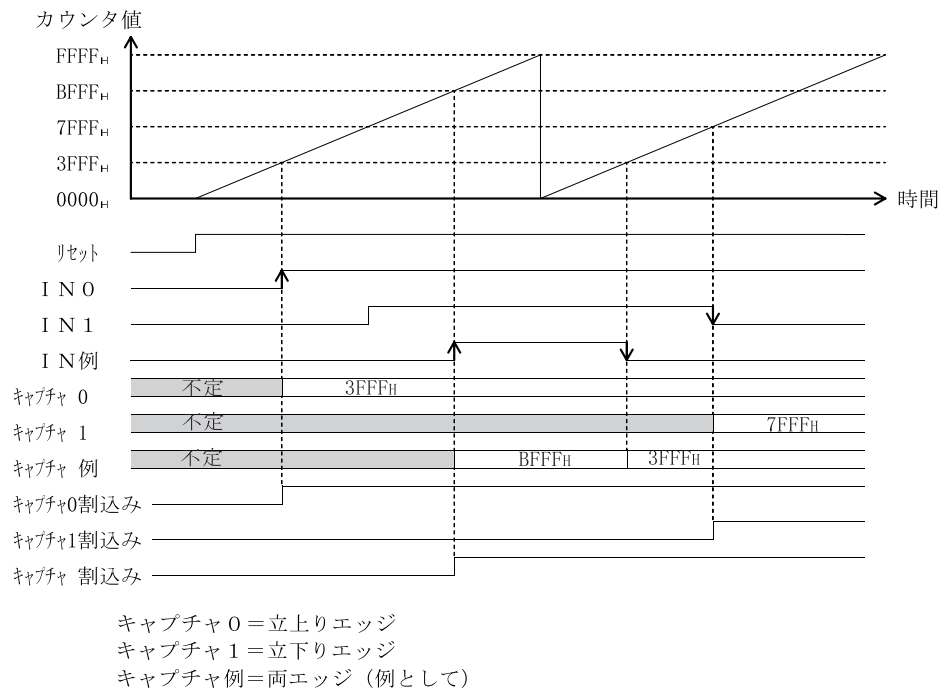


図 12.6-1 インプットキャプチャの取り込みタイミング例

## インプットキャプチャ入力タイミング

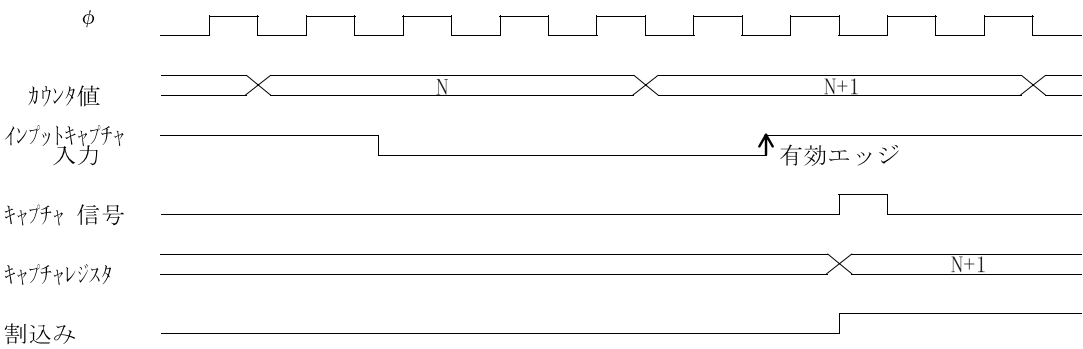


図 12.6-2 入力信号に対するキャプチャタイミング



## 第13章 16ビットリロードタイマ( イベントカウント機能付 )

---

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

---

- 13.1 16ビットリロードタイマ( イベントカウント機能付 ) の概要
- 13.2 16ビットリロードタイマ( イベントカウント機能付 ) のレジスタ
- 13.3 クロック動作
- 13.4 アンダフロー動作
- 13.5 入出力端子機能
- 13.6 カウンタの動作状態



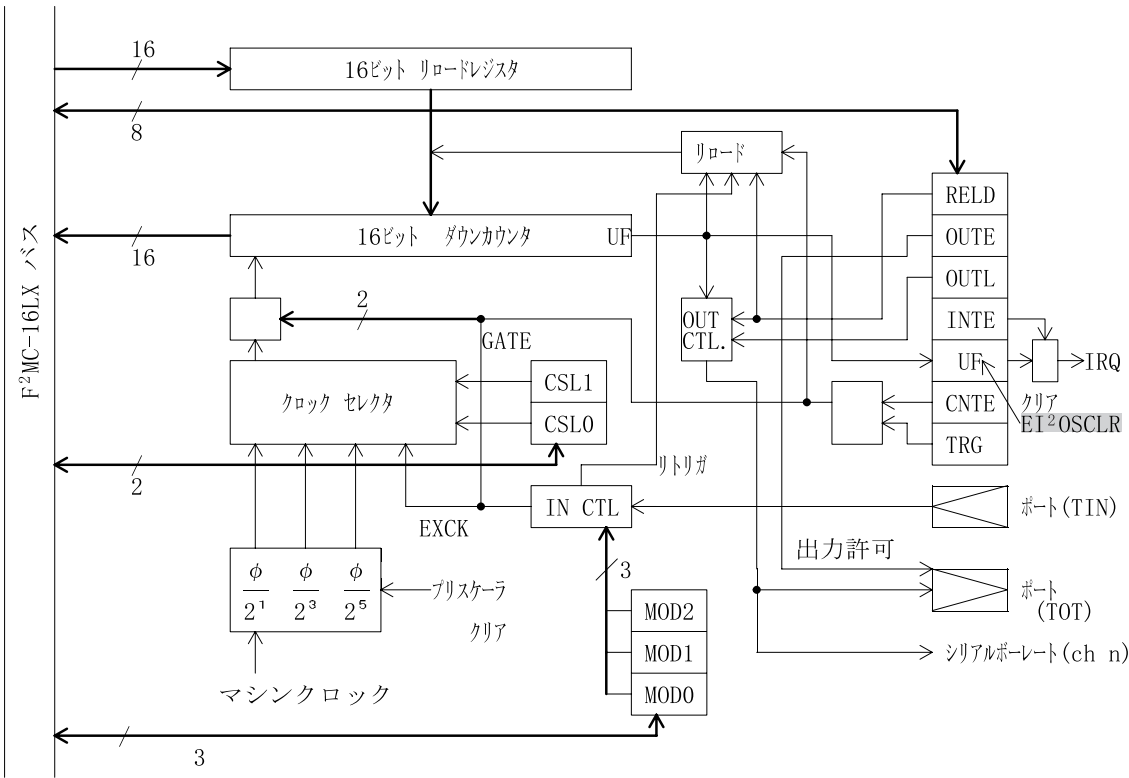
# 13.1 16ビットリロードタイマ（イベントカウント機能付）の概要

16ビットリロードタイマは3チャンネルあり，各々16bitのダウンカウンタ1個，16bitのリロードレジスタ1個，入力端子1本(TIN)，出力端子1本(TOT)，コントロールレジスタ1個から構成されています。入力クロックは，1つの外部クロックと，3種類の内部クロックから選択できます。

## 16ビットリロードタイマ（イベントカウント機能付）の概要

出力端子(TOT)は，リロードモードではトグル出力波形を出力し，ワンショットモードのカウンタ中は矩形波を出力します。入力端子(TIN)は，イベントカウントモードでのイベント入力，内部クロックモードでのトリガ入力またはゲート入力として機能します。

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



(注) リロードタイマの各チャンネルとUARTの各チャンネルは，以下のように接続されています。

- ・リロードタイマch0 : UART0, UART3
- ・リロードタイマch1 : UART1, UART4
- ・リロードタイマch2 : UART2

図 13.1-1 16ビットリロードタイマ（イベントカウント機能付）のブロックダイアグラム

## 13.2 16ビットリロードタイマ（イベントカウント機能付）のレジスタ

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

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

### 16ビットリロードタイマ（イベントカウント機能付）のレジスタ

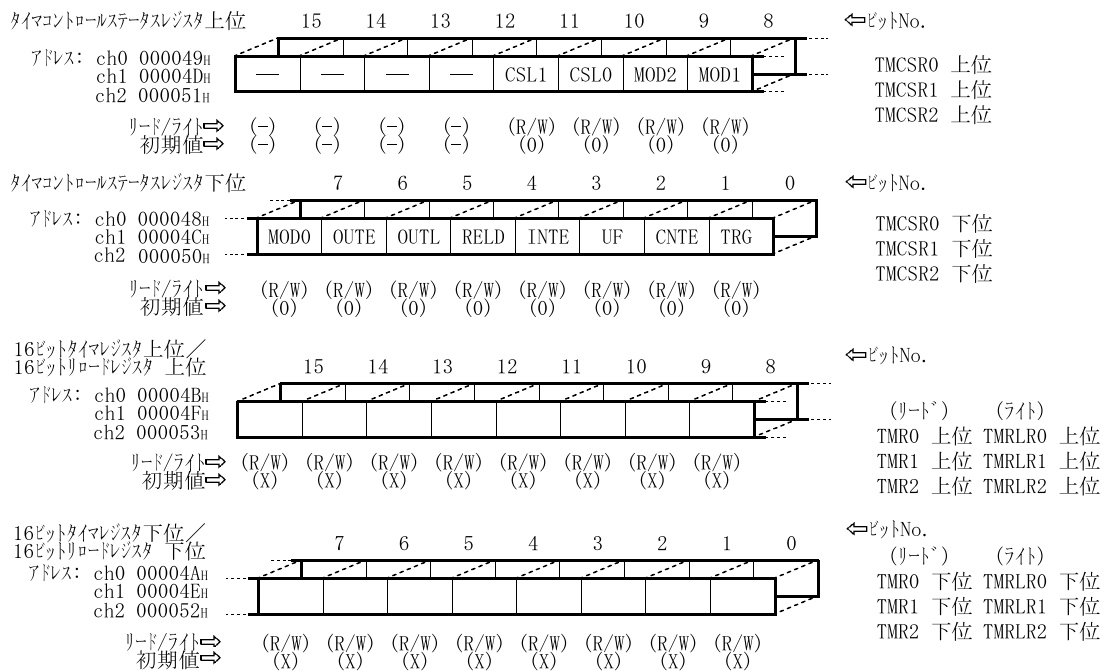


図 13.2-1 16ビットリロードタイマ（イベントカウント機能付）のレジスタ

### 13.2.1 タイマコントロールステータスレジスタ(TMCSR)

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

タイマコントロールステータスレジスタ(TMCSR)

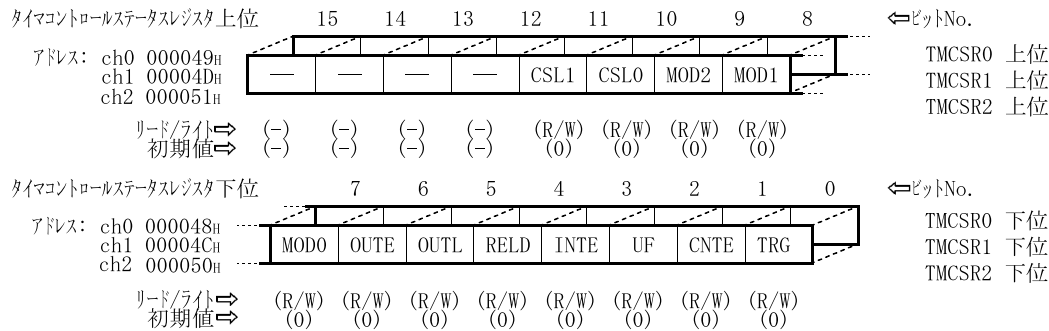


図 13.2-2 タイマコントロールステータスレジスタ(TMCSR)

< 注意事項 >

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

【ビット11, 10】CSL1, CSL0

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

表 13.2-1 CSL1, CSL0(カウントクロックセレクトビット)の機能

CSL1	CSL0	クロックソース(マシンサイクル $\phi=16\text{MHz}$ )
0	0	$\phi/2^4$ (0.125 $\mu\text{s}$ ) [初期値]
0	1	$\phi/2^3$ (0.5 $\mu\text{s}$ )
1	0	$\phi/2^2$ (2.0 $\mu\text{s}$ )
1	1	外部イベントカウントモード

【ビット9, 8, 7】MOD2, MOD1, MOD0

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

MOD2=0のとき,入力端子はトリガ入力として機能し,有効エッジが入力されるとリロードレジスタの内容をカウンタへロードし,カウント動作を継続します。MOD2=1の場合,ゲートカウンタモードになり,入力端子(TIN) はゲート入力となり,有効レベルが入力されている間のみカウントをします。

MOD1,0ビットは,各モードにおける端子の機能を設定します。

表 13.2-2 MOD2, MOD1, MOD0(動作モード/入出力端子機能設定ビット)の機能

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

x: 任意の値。

## 【ビット6】OUTE

OUTEビットは, 出力許可ビットです。

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

## 【ビット5】OUTL

OUTLビットは, TOT端子の出力レベルを設定するビットです。

## 【ビット4】RELD(RELoad)

RELDビットは, リロード動作を許可するビットです。

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

表 13.2-3 RELD(リロード動作許可ビット)の機能

OUTE	RELD	OUTL	出力波形
0	x	x	汎用ポート
1	0	0	カウント中に"H"レベル矩形波を出力
1	0	1	カウント中に"L"レベル矩形波を出力
1	1	0	トグル出力。カウント開始時"L"レベル
1	1	1	トグル出力。カウント開始時"H"レベル

x: 任意の値

## 【ビット3】INTE(INTerrupt Enable)

INTEビットは, タイマ割込み要求許可ビットです。INTEビットが"1"のときは, UFビットが, "1"に変わっても割込み要求を生成します。INTEが"0"の時は, UFビットが"1"に変わっても割込み要求を生成しません。

表 13.2-4 INTE(タイマ割込み要求許可ビット)の機能

INTE	機能
0	割込み禁止
1	割込み許可

【ビット2】UF(Under Flow)

UFビットは、タイマ割込み要求フラグです。

カウンタの値が、0000<sub>H</sub>～FFFF<sub>H</sub>へのアンダフローにより"1"にセットされます。

"0"の書込みまたは拡張インテリジェントI/Oサービスによってクリアされます。このビットへの"1"書込みは、意味がありません。リードモディファイライト系命令における読出し時には、"1"が読出されます。

【ビット1】CNTE(CouNT Enable)

CNTEビットは、タイマのカウントイネーブルビットです。

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

【ビット0】TRG(TRiG)

TRGビットは、ソフトウェアトリガビットです。

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

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

### 13.2.2 16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR)

16ビットタイマレジスタ(TMR) ( リード時 ) は , 16ビットタイマのカウンタ値を  
出す事ができるレジスタです。初期値は不定です。

16ビットリロードレジスタ(TMRLR)( ライト時 ) は , 16ビットリロードレジスタは ,  
カウンタの初期値を保持しておくレジスタです。初期値は不定です。

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

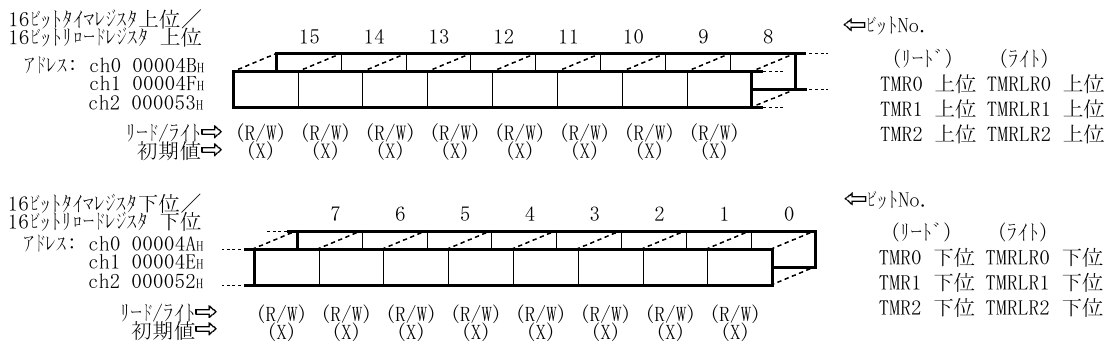


図 13.2-3 16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR)

#### < 注意事項 >

16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR)は , ワードアクセスしてください。

### 13.3 クロック動作

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

#### 内部クロック動作

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

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

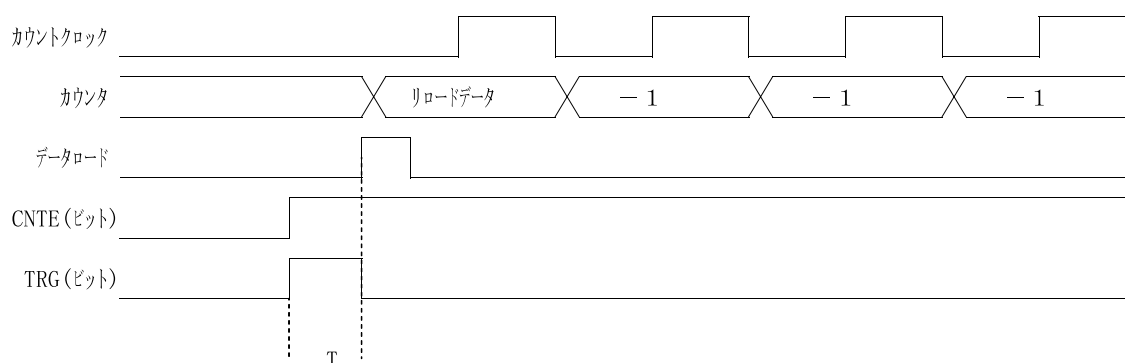


図 13.3-1 カウンタの起動および動作

#### 外部イベントカウント

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

## 13.4 アンダフロー動作

16ビットリロードタイマ(イベントカウント機能付)は、カウンタの値が0000<sub>H</sub>からFFFF<sub>H</sub>になる時をアンダフローと定義しています。したがって、[リロードレジスタの設定値 + 1] カウントでアンダフローが発生することになります。

### アンダフロー動作

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

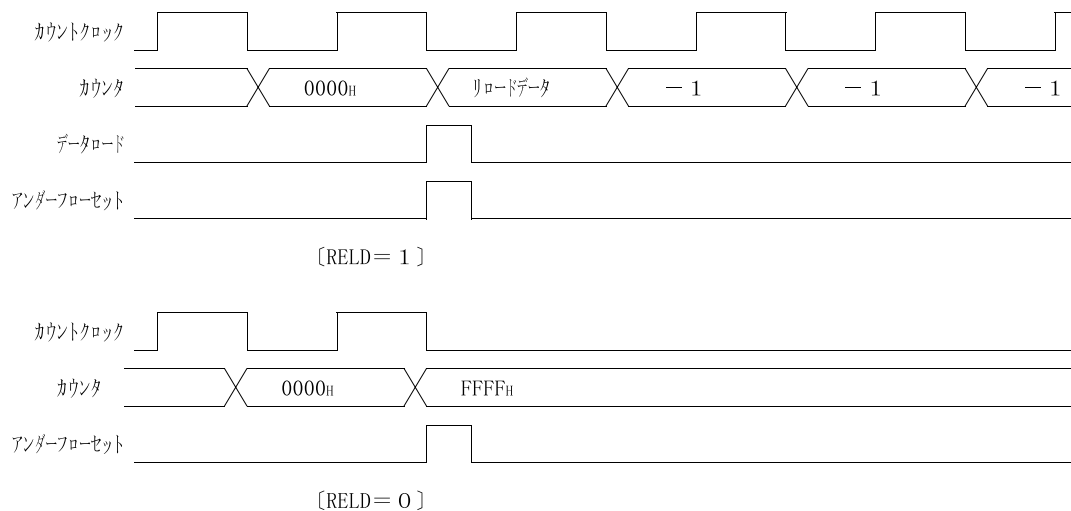


図 13.4-1 アンダフロー動作

### 拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)機能と割り込み

本タイマはEI<sup>2</sup>OSをサポートするタイマがあり、アンダフローが発生した時にEI<sup>2</sup>OSを起動することができます。本製品では、両方のタイマでEI<sup>2</sup>OSを使用できますが、3チャンネルのタイマ(ch0-2)は割り込みコントローラ中の同じ割り込み制御レジスタ(ICRx)に接続されているため、ch0-2を異なるEI<sup>2</sup>OSサービスに割り当てることはできません。また、両タイマは異なる割り込みベクトルを持つため、2つの異なる割り込みサービスに割り当てることができます。しかし、ch0-2は前述のように割り込み制御レジスタを共有するため、同一の割り込みレベルが3チャンネルに適用されます。



## 13.5 入出力端子機能（内部クロックモード時）

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

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

### 入力端子機能（内部クロックモード時）

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

トリガ入力として使用した場合、図 13.5-1に示すように、有効エッジが入力されるとリロードレジスタの内容をカウンタにロードして、内部プリスケアラをクリアした後、カウント動作を開始します。

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

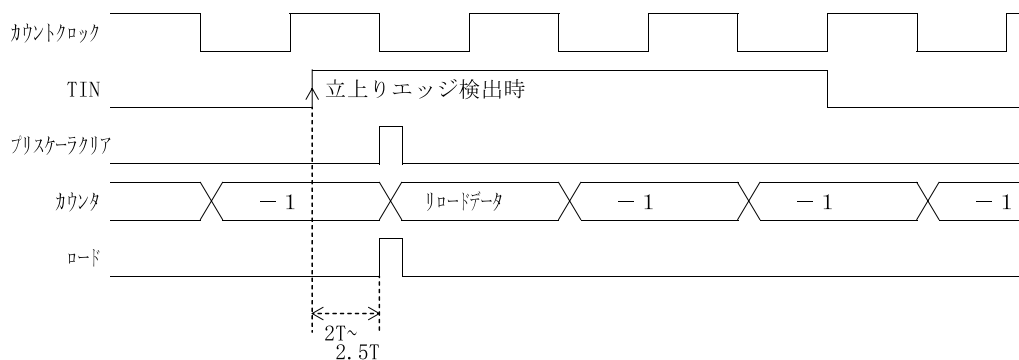


図 13.5-1 トリガ入力の動作

ゲート入力として使用する場合、図 13.5-2に示すように、コントロールレジスタのMOD0ビットにより設定される有効レベルが、TIN端子から入力されている間のみカウントをします。この時カウントクロックは、止まらずに動き続けます。ゲートモード時のソフトウェアトリガは、ゲートレベルに関わらず可能です。TIN端子のパルス幅は、 $2 \times T$ （Tはマシンサイクル）以上にしてください。

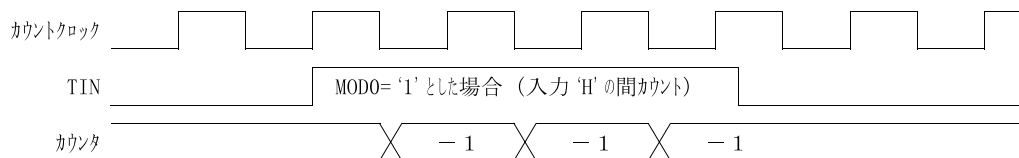


図 13.5-2 ゲート入力の動作

出力端子機能

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

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

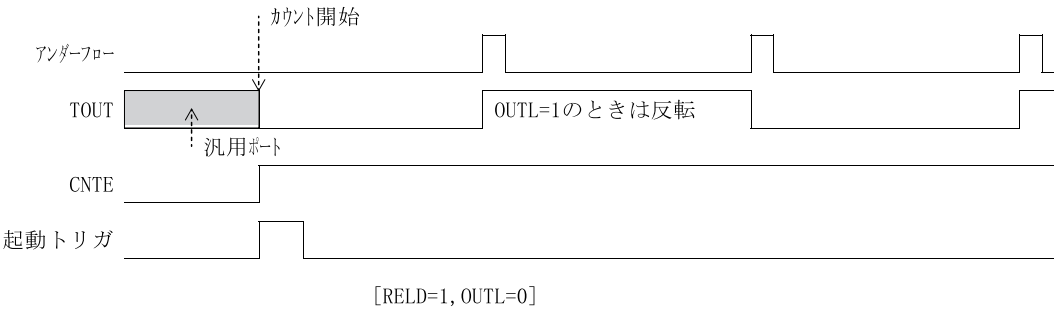


図 13.5-3 出力端子機能(1)

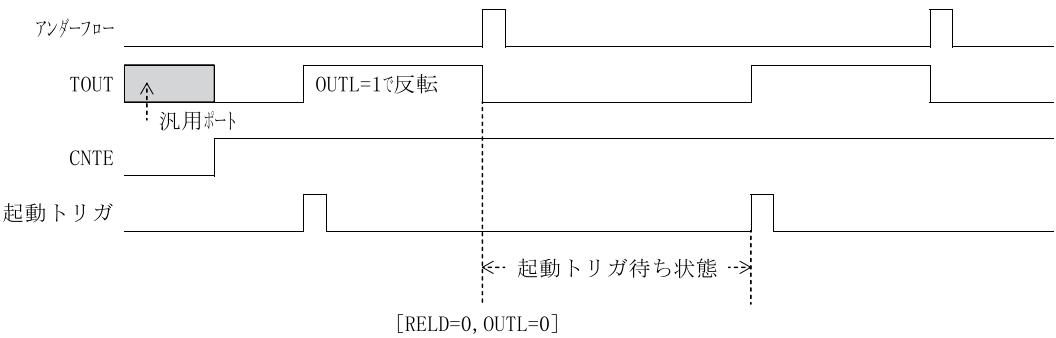


図 13.5-4 出力端子機能(2)

# 13.6 カウンタの動作状態

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

- ・ CNTE = "0", WAIT = "1" の停止状態（STOP状態）
- ・ CNTE = "1", WAIT = "1" の起動トリガ待ち状態（WAIT状態）
- ・ CNTE = "1", WAIT = "0" の動作状態（RUN状態）があります。

## カウンタの動作状態

図 13.6-1に、カウンタの動作状態を示します。

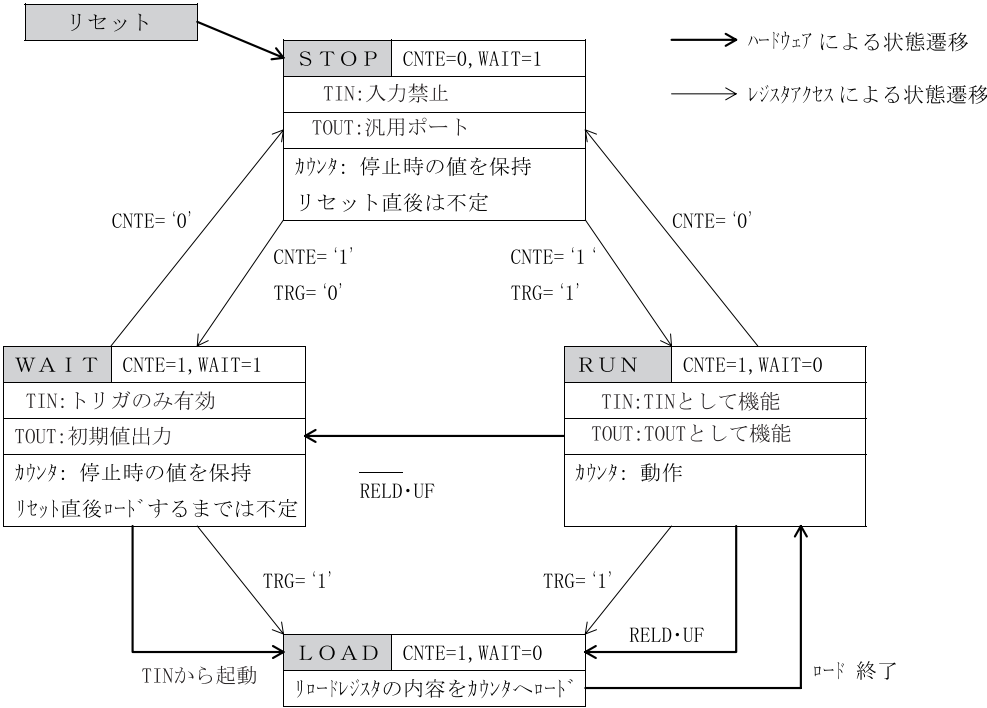


図 13.6-1 カウンタ状態遷

## 第14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。

---

- 14.1 8/16ビットPPGの概要
- 14.2 8/16ビットPPGのブロックダイヤグラム
- 14.3 8/16ビットPPGのレジスタ
- 14.4 8/16ビットPPGの動作

## 14.1 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

ハードウェアとして、8bitダウンカウンタ2個、8bitリロードレジスタ4個、16bit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本を有し、これらにより次の機能を実現します。

8ビットPPG出力2ch独立動作モード

2ch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

1chの16bitのPPG出力動作が可能です。

8 + 8ビットPPG出力動作モード

ch0の出力を、ch1のクロック入力とすることにより、指定間隔の8bit PPG出力動作が可能です。

PPG出力動作

指定間隔・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。

## 14.2 8/16ビットPPGのブロックダイアグラム

図 14.2-1に、8/16ビットPPG(ch0)のブロックダイアグラムを、図 14.2-2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

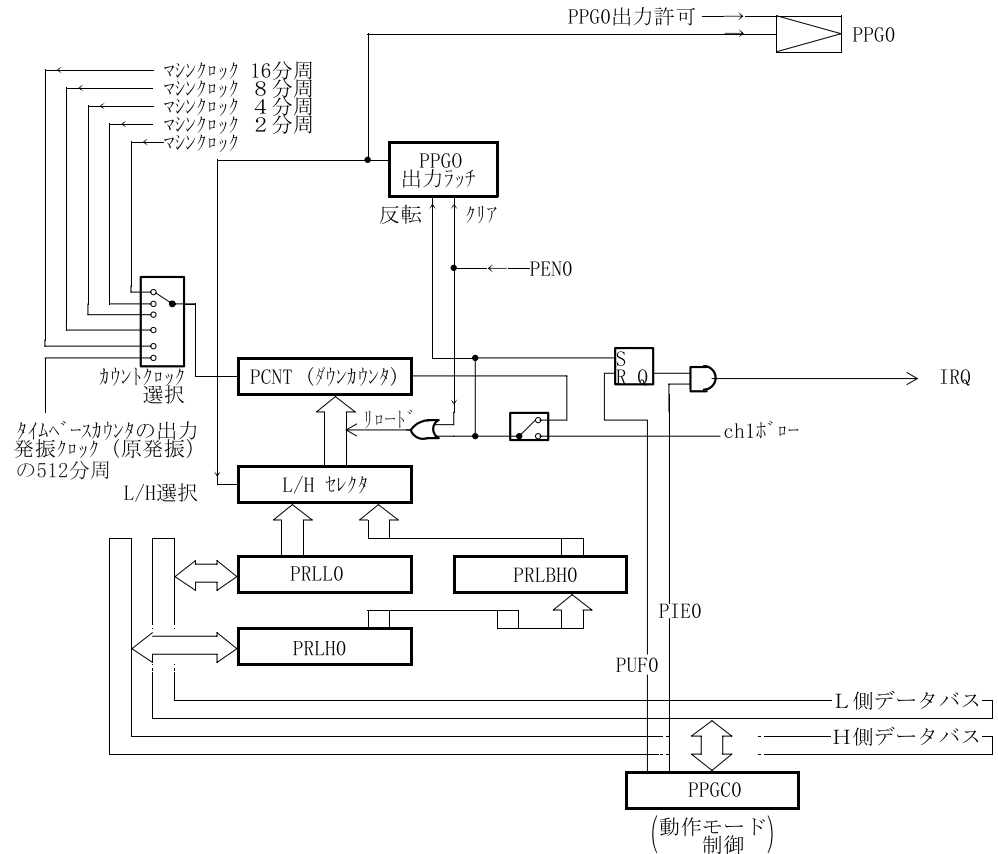


図 14.2-1 8/16ビットPPG(ch0)のブロックダイアグラム

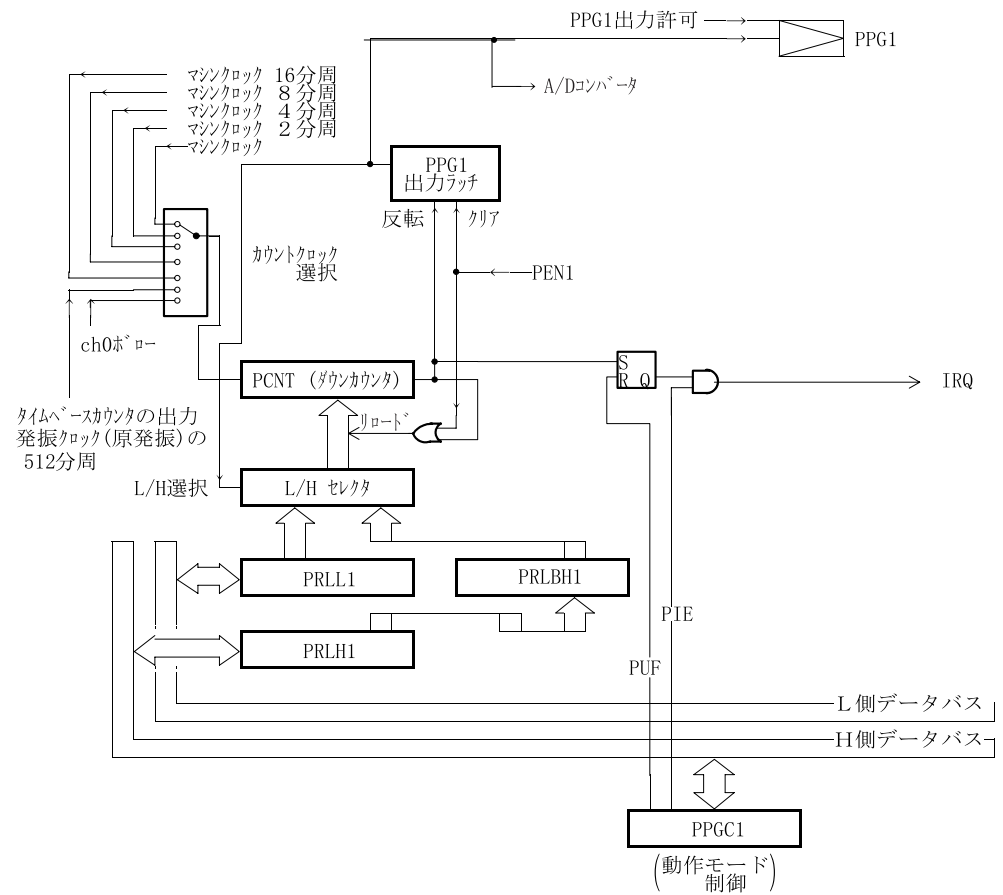


図 14.2-2 8/16ビットPPG(ch1)のブロックダイヤグラム

## 14.3 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。

- PPG0/1動作モード制御レジスタ
- PPG0/1出力制御レジスタ
- リロードレジスタH/L

### 8/16ビットPPGのレジスタ

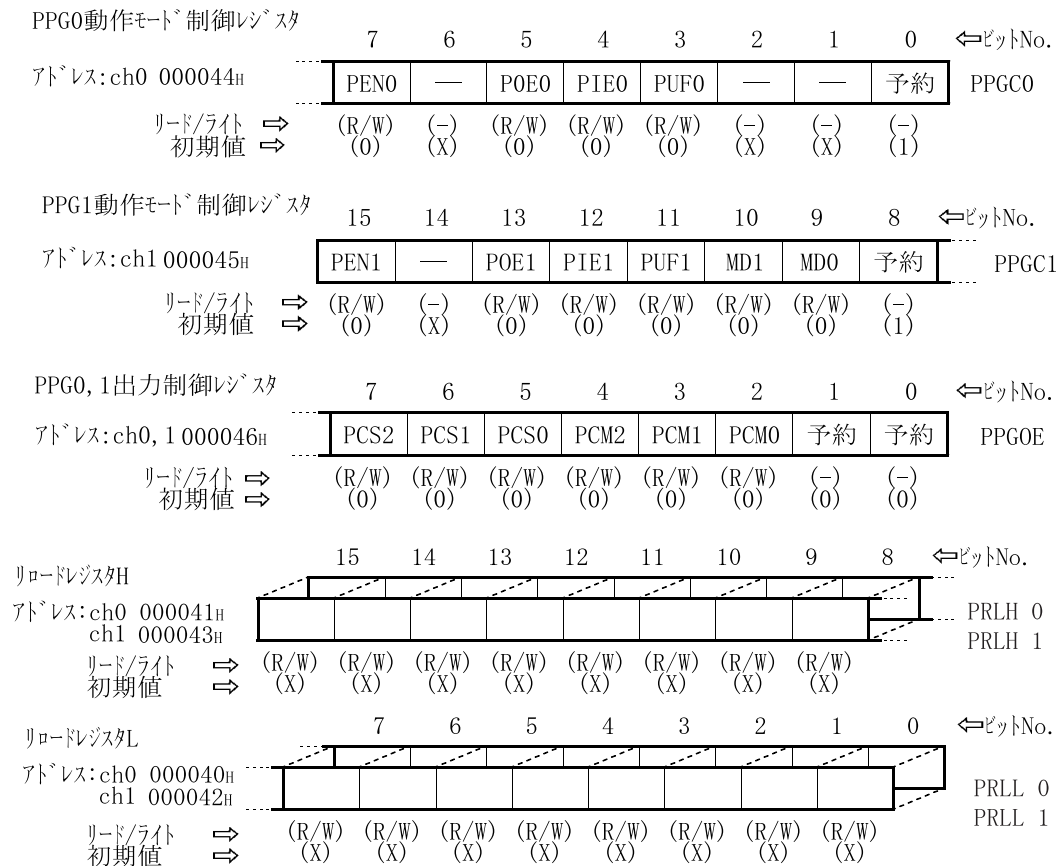


図 14.3-1 8/16ビットPPGのレジスタ



### 14.3.1 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、トリガの制御を行います。

PPG0動作モード制御レジスタ(PPGC0)

PPG0動作モード <sup>※</sup> 制御レジスタ								ビットNo. ←	
アドレス: ch0 000044H								PPGC0	
	7	6	5	4	3	2	1		0
	PEN0	—	POE0	PIE0	PUF0	—	—		予約
リード/ライト ⇒ 初期値 ⇒	(R/W) (0)	(—) (X)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(—) (X)	(—) (X)		(—) (1)

図 14.3-2 PPG0動作モード制御レジスタ(PPGC0)

【ビット7】PEN0(PPG ENable)

PEN0ビットは、PPGの動作開始および動作モードを、表 14.3-1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。リード/ライトが可能です。

表 14.3-1 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】POE0(PPG 0output Enable0)

POE0ビットは、パルス出力外部端子PPG0を、表 14.3-2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。リード/ライトが可能です。

表 14.3-2 POE0(PPG0端子出力許可ビット)の機能

PE0	機能
0	汎用ポート端子 (パルス出力禁止) [ 初期値 ]
1	PPG0 = パルス出力端子 (パルス出力許可)

【ビット4】PIE0(PPG Interrupt Enable)

PIE0ビットは、PPGの割込み許可を、表 14.3-3で示すように制御します。本ビットが"1"の時、PUF0が"1"にセットされると割込み要求を発生します。本ビットが"0"の時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。リード/ライトが可能です。

表 14.3-3 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

## &lt; 注意事項 &gt;

8/16ビットPPGタイマ0(PPG0)と16ビットリロードタイマは、割込み制御レジスタ(ICR07)を共有しています。したがって、16ビットリロードタイマでEI<sup>2</sup>OSを使用するときは、PPG0の割込みは禁止(PIEビット="0")してください。

## 【ビット3】PUF0(PPG Underflow Flag)

PUF0ビットは、PPGのカウンタアンダフロービットを表 14.3-4で示すように制御します。

表 14.3-4 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

8ビットPPG2chモードまたは8ビットプリスケラ+8ビットPPGモードでは、ch0のカウンタの値が00H~FFHへなった結果アンダフローが発生すると本ビットに"1"が書き込まれます。16ビットPPG1chモードの時には、ch1/ch0のカウンタの値が0000H~FFFFHになった結果アンダフローが発生すると、本ビットに"1"が書き込まれます。"0"の書き込みにより、本ビットは"0"に設定されます。

このビットへの"1"の書込は、無効です。

リード、モディファイ、ライトのリード動作では"1"が読みだされます。

リセットにより、本ビットは、"0"に初期化されます。リード/ライトが可能です。

## 【ビット0】予約ビット

ビット0は、予約ビットです。PPGC0を設定するときは必ず"1"に設定してください

# 14.3.2 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	15	14	13	12	11	10	9	8	↔ビットNo.
アドレス: ch1 000045H	PEN1	—	POE1	PIE1	PUF1	MD1	MD0	予約	PPGC1
リード/ライト	⇒ (R/W)	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(-)	
初期値	⇒ (0)	(X)	(0)	(0)	(0)	(0)	(0)	(1)	

図 14.3-3 PPG1動作モード制御レジスタ(PPGC1)

## 【ビット15】PEN1(PPG ENable)

PEN1ビットは, PPGの動作開始, および動作モードを, 表 14.3-5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。リード/ライトが可能です。

表 14.3-5 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】POE1(PPG Output ENable)

POE1は, パルス出力外部端子PPG1を, 表 14.3-6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。リード/ライトが可能です。

表 14.3-6 POE1(PPG1端子出力許可ビット)の機能

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

## 【ビット12】PIE1(PPG Interrupt Enable)

PIE1ビットは, PPGの割込み許可を, 表 14.3-7で示すように制御します。本ビットが"1"の時, PUF1が"1"にセットされると割込み要求を発生します。本ビットが"0"の時には, 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。リード/ライトが可能です。

表 14.3-7 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

## &lt; 注意事項 &gt;

8/16ビットPPGタイマ1(PPG1)とUART0の送信完了は、割込み制御レジスタ(ICR11)を共有しています。したがって、UART0の送信完了でEI<sup>2</sup>OSを使用するときは、PPG1の割込みは禁止(PIEビット="0")してください。

## 【ビット11】PUF1(PPG Underflow Flag)

PUF1は、PPGのカウンタアンダフロービットを、表 14.3-8のように制御します。

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モードの時には、ch1のカウンタの値が00H~FFH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モードの時には、ch1/ch0のカウンタの値が0000H~FFFFHへなった時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"書込は、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"が読まれます。

リセットにより、本ビットは"0"に初期化されます。本ビットは、リード/ライトが可能です。

表 14.3-8 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】MD1, 0(PPG count MoDe)

MD1, 0ビットは、PPGタイマの動作モードを、表 14.3-9のように選択します。

リセットにより、本ビットは"00<sub>B</sub>"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 14.3-9 MD1, 0(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
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

## &lt; 注意事項 &gt;

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

## 【ビット8】予約ビット

ビット8は、予約ビットです。PPGC1を設定するときは必ず"1"に設定してください。

### 14.3.3 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8ビットの制御レジスタです。

PPG0,1出力端子制御レジスタ(PPGOE)

PPG0,1出力制御レジスタ	7	6	5	4	3	2	1	0	←ビットNo.
アドレス: ch0, 1 000046 <sub>H</sub>	PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	予約	予約	PPGOE
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)	(0)	(0)	

図 14.3-4 PPG0,1出力端子制御レジスタ(PPGOE)

#### 【ビット7～5】PCS2～0(PPG Count Select)

PCS2～0は、ch1のダウンカウンタの動作クロックを表 14.3-10に示すように選択します。リセットにより、本ビットは"000<sub>b</sub>"に初期化されます。リード/ライトが可能です。

表 14.3-10 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	マシニングロック (62.5nsマシニングロック 16MHz時)
0	0	1	マシニングロック / 2 (125nsマシニングロック 16MHz時)
0	1	0	マシニングロック / 4 (250nsマシニングロック 16MHz時)
0	1	1	マシニングロック / 8 (500nsマシニングロック 16MHz時)
1	0	0	マシニングロック / 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 原発振 4MHz時)

#### < 注意事項 >

8ビットプリスケラ+8ビットPPGモードまたは16ビットPPGモードの時には、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ビットは無効です。

#### 【ビット4～2】PCM2～0(PPG Count Mode)

PCM2～0ビットは、ch0のダウンカウンタの動作クロックを表 14.3-11のように選択します。リセットにより、本ビットは"000<sub>b</sub>"に初期化されます。リード/ライトが可能です。

表 14.3-11 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	マシニングロック (62.5nsマシニングロック 16MHz時)
0	0	1	マシニングロック / 2 (125nsマシニングロック 16MHz時)
0	1	0	マシニングロック / 4 (250nsマシニングロック 16MHz時)
0	1	1	マシニングロック / 8 (500nsマシニングロック 16MHz時)
1	0	0	マシニングロック / 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 原発振 4MHz時)

#### 【ビット1～0】予約ビット

ビット1～0は、予約ビットです。PPGEを設定するときは、必ず"0"に設定してください。

14.3.4 リロードレジスタ (PRLL/PRLH)

リロードレジスタ (PRLL/PRLH) は ,ダウンカウンタPCNTへのリロード値を保持する , 各8ビットのレジスタです。

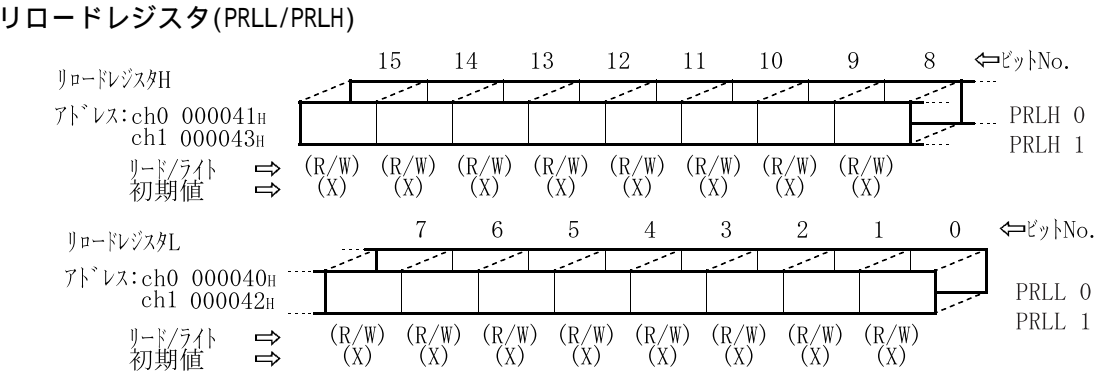


図 14.3-5 リロードレジスタ (PRLL/PRLH)

リロードレジスタ (PRLL/PRLH) は ,それぞれ表 14.3-12に示す機能をもっています。どちらのレジスタも , リード / ライトが可能です。

表 14.3-12 リロードレジスタ (PRLL/PRLH)

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

< 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 , ch1のPRLL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PRLLとPRLHは , 同じ値を書き込んでください。

## 14.4 8/16ビットPPGの動作



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

### 8/16ビットPPGの動作

8bit長のPPGユニットそれぞれは、8bit長のリロードレジスタがL側(PRL)とH側(PRH)の2本あります。これら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。この動作の結果、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。

動作開始と再開は、対応するレジスタのビットへのデータ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 14.4-1に示します。

表 14.4-1 リロード動作とパルス出力の関係

リロード動作	端子出力変化
PRLH PCNT	PPG0/1 [0 1]  立ち上がり
PRLH PCNT	PPG0/1 [1 0]  立ち下がり

また、PPGC0レジスタのbit4(PIE0)または、PPGC1レジスタのbit12(PIE1)が"1"に設定されている時、それぞれのカウンタの00<sub>H</sub>からFF<sub>H</sub>へのボロー（16ビットPPGモードの場合には、0000<sub>H</sub>からFFFF<sub>H</sub>へのボロー）によって割込み要求が出力されます。

### 8/16ビットPPGの割込み

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

8ビットPPG2chモードおよび8ビットプリスケアラ + 8ビットPPGモードの時には、それぞれのカウンタのボローにより、それぞれの割込みの要求が行われますが、16ビットPPGモードでは、16ビットのカウンタのボローにより、PUF0ビットとPUF1ビットが同時にセットされます。このため、割込み要因を一本化するために、PIE0ビットまたはPIE1ビットのどちらか一方のみを許可にすることを推奨します。また、割込み要因のクリアもPUF0ビットとPUF1ビットを同時に行うことを推奨します。

### 各ハードウェアの初期値

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

#### レジスタ

- PPGC0 0X000XX1B
- PPGC1 0X000001B
- PPGOE 00000000B

#### パルス出力

- PPG0 "L"
- PPG1 "L"
- PE0 PPG0出力禁止

- PE1 PPG1出力禁止

割込み要求

- IRQ0 "L"
- IRQ1 "L"

上記以外のハードウェアは初期化されません。



## 14.4.1 8/16ビットPPGの動作モード

---

8/16ビットPPGには、以下の3種類の動作モードがあります。

- ・ 2ch独立モード
  - ・ 8ビットプリスケラ + 8ビットPPGモード
  - ・ 16ビットPPGモード
- 

### 8/16ビットPPGの動作モード

#### 2ch独立モード

2ch独立モードでは、8ビットPPGユニットの2chが独立して動作します。

PPG0端子には、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ビットプリスケラ + 8ビットPPGモード

8ビットプリスケラ + 8ビットPPGモードでは、ch0を8ビットプリスケラとして使用し、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します。

PPG0端子には、ch0のプリスケラ出力が接続され、PPG1端子には、ch1のPPG出力が接続されます。

#### 16ビットPPG 1chモード

16ビットPPG 1chモードでは、ch0とch1を連結させて、16ビットのPPGとして使用します。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。

## 14.4.2 PPG出力動作

8/16ビットPPGは、ch0のPPGについては、PPGC0(PWM動作モード制御)レジスタのbit7(PEN0)を"1"に書き込むと起動され、カウントを開始します。同様に、ch1のPPGは、PPGC1レジスタのbit15(PEN1)に"1"を書き込むと、起動されカウントを開始します。動作を開始した後は、PPG0のbit7または、PPGC1のbit15に"0"を書き込むことによってカウント動作を停止します。一度停止すると、パルス出力は" L " レベルを保持します。

### PPG出力動作

PPGの出力動作時には、次の2つの点について注意してください。

- 8ビットプリスケラ + 8ビットPPGモードでは、ch0の動作を停止して、ch1を動作状態に設定しないでください。
- 16ビットPPGモードでは、PPGC0レジスタのbit7(PEN0)とPPGC1レジスタのbit15(PEN1)は、必ず同時に開始・停止を行ってください。

図 14.4-1に、PPG出力動作を説明します。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。

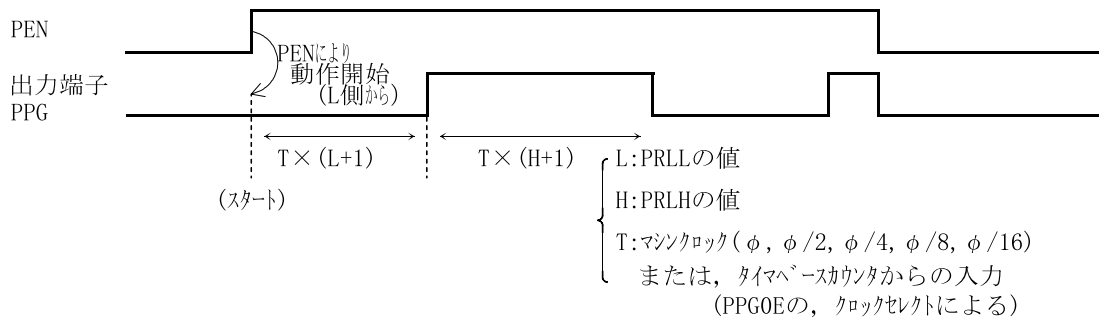


図 14.4-1 PPG出力動作の出力波形

### 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<sub>H</sub>の時、または、16ビットPPG動作時のリロードレジスタ値が 0000<sub>H</sub>の時、カウントクロック1周期分のパルス幅とらる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<sub>H</sub>の時、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<sub>H</sub>の時、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。

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

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

L : PRLの値

H : PRLHの値

T : 入力クロック周期

Ph: ハイパルスの幅

PI: ローパルスの幅

### 14.4.3 カウントクロックの選択

8/16ビットPPGの動作に使用するカウントクロックは、マシニングロックおよびタイムベースカウンタから供給され、6種類のカウントクロック入力を選択できます。

PPG0Eレジスタのbit4～2(PCM2～0)でch0のクロックを、PPG0Eレジスタのbit7～S(PCS2～0)でch1のクロックを選択します。クロックは、マシニングロックの1/16～1倍のマシニングロックとタイムベースカウンタのからの入力クロックを選択します。

ただし、8ビットプリスケラ+8ビットPPGモードまたは、16ビットPPGモードの時には、PPGC1レジスタのbit14(PCS1)の値は無効です。ch1のPPGは、ch0からカウントクロックを受とるため、レジスタが無効となります。

#### カウントクロックの選択に関する注意事項

タイムベースタイマの入力を使用した場合、トリガにより起動がかかる最初のカウント周期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。

8ビットプリスケラ+8ビットPPGモード・ch0が動作状態で、ch1が停止状態のときにch1を起動すると、最初のカウント周期がずれる可能性があります。

#### 14.4.4 パルスの端子出力の制御

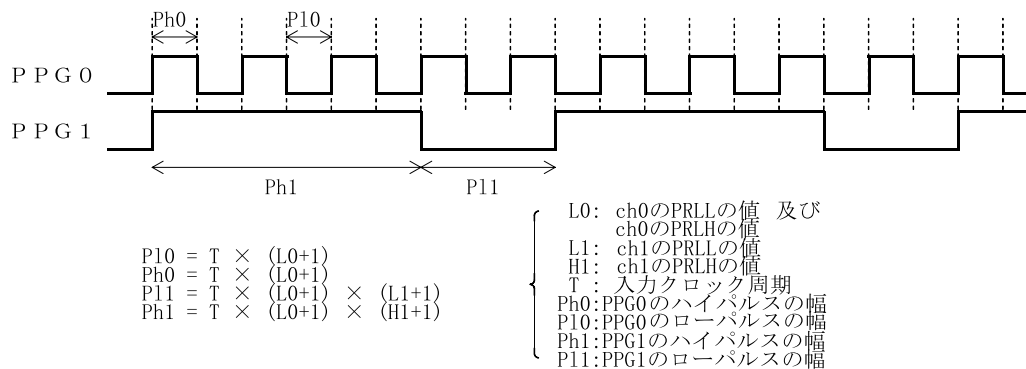
本モジュールによって生成されたパルス出力は，外部端子PPG0/PPG1より出力させることができます。

外部端子からパルスを出力するには，各端子に対応するビットに"1"を書き込みます。PPG0端子に対してはPPGC0レジスタのbit5(P0E0)を，PPG1端子に対してはPPGC1レジスタのbit13(P0E1)を使用します。本ビットがに"0"（初期値）が書き込まれている時は，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。

##### パルスの端子出力の制御

16ビットPPGモードでは，PPG0，PPG1から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からは8ビットプリスケラのトグル波形が出力され，PPG1からは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 14.4-2に示します。



(注意)ch0のPRLHとch1のPRLHは，同じ値を設定します。

図 14.4-2 8 + 8PPG動作の出力波形

### 14.4.5 リロードレジスタへのライトタイミング

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

リロードレジスタへのライトタイミング

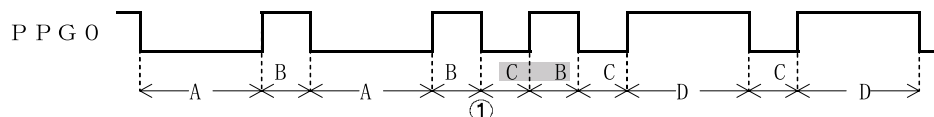


図 14.4-3 ライトタイミングタイムチャート図

図 14.4-3のタイムチャートにおいて、①のタイミングの前でPRLをAからCに書き換えて、②の後PRLHの値をBからDに書き換えた場合、③のタイミングでの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。このモードの時には、ch0のPRL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に書き込んだ時点で、実際にch0のPRLへ書き込まれます。

なお、16ビットPPGモード以外では、ch0とch1のPRLの書き込みは独立して行えます。

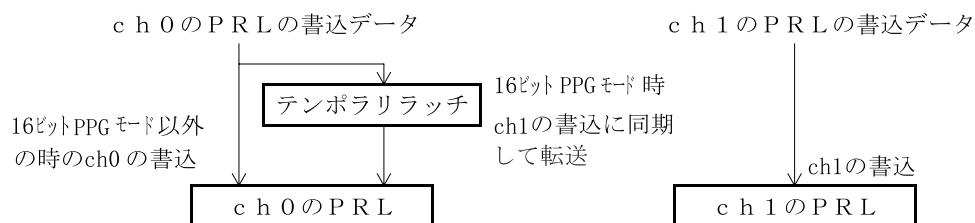


図 14.4-4 PRL書き込み部分のブロックダイヤグラム

## 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)は、デバイス外部に存在するペリフェラルとF<sup>2</sup>MC-16LX CPUとの間に置かれています。DTPは、外部ペリフェラルからDMA要求あるいは割込み要求を受け取り、この要求をF<sup>2</sup>MC-16LX CPUに転送して、拡張インテリジェントI/Oサービスまたは、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サービスの場合は"H"レベル、"L"レベルの2種が、外部割込み要求の場合は、"H"レベル、"L"レベルのほか、立上がりエッジと立下がりエッジの計4種が選択可能です。

## DTP/外部割込みのレジスタ

割込み／DTP許可レジスタ	7	6	5	4	3	2	1	0	⇐ビットNo.
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)	
割込み／DTP要因レジスタ	15	14	13	12	11	10	9	8	⇐ビットNo.
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下位	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 下位
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
要求レベル設定レジスタ上位	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	ELVR 上位
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 15.1-1 DTP/外部割込みのレジスタ

## DTP/外部割込みのブロックダイアグラム

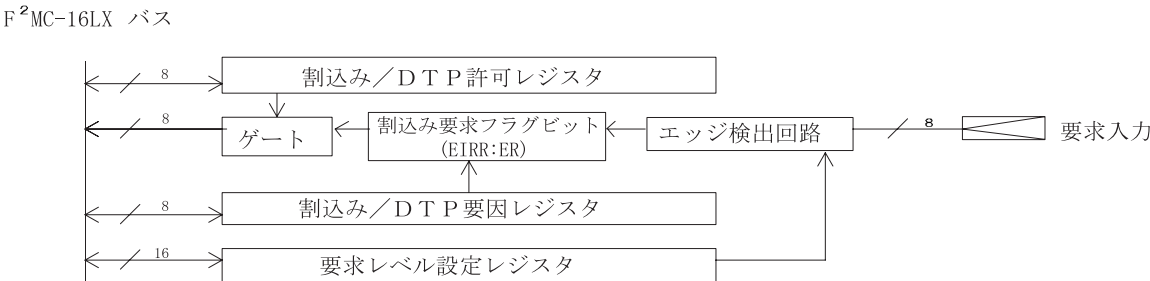


図 15.1-2 DTP/外部割込みのブロックダイアグラム

# 15.2 DTP/外部割込みのレジスタ

ENIRレジスタは、デバイス端子を外部割込み/DTP要求入力として使用し、割込みコントローラに対して要求を発生する機能があります。

## 割込み/DTP許可レジスタ(ENIR)

割込み/DTP許可レジスタ	7	6	5	4	3	2	1	0	←ビットNo.
アドレス:000030h	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	ENIR
リード/ライト 初期値	⇒ (0)	⇒ (0)	⇒ (0)	⇒ (0)	⇒ (0)	⇒ (0)	⇒ (0)	⇒ (0)	

図 15.2-1 割込み/DTP許可レジスタ(ENIR)

割込み/DTP許可レジスタ(ENIR)の"1"を書き込んだビットに対応する端子は、外部割込み/DTP要求入力として使用されます。"0"を書き込んだビットに対応する端子は外部割込み/DTP要求入力要因を保持しますが、割込みコントローラに対しては要求を発生しません。

## 割込み/DTP要因レジスタ(EIRR)

割込み/DTP要因レジスタ	15	14	13	12	11	10	9	8	←ビットNo.
アドレス:000031h	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	EIRR
リード/ライト 初期値	⇒ (X)	⇒ (X)	⇒ (X)	⇒ (X)	⇒ (X)	⇒ (X)	⇒ (X)	⇒ (X)	

図 15.2-2 割込み/DTP要因レジスタ(EIRR)

EIRRレジスタは、読出し時に対応する外部割込み/DTP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の値をクリアします。このレジスタを読出した時に"1"であったとき、このビットに対応する端子に外部割込み/DTP要求があることを示します。

また、このレジスタに"0"を書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"が読まれます。



要求レベル設定レジスタ(ELVR)

要求レベル設定レジスタ下位	7	6	5	4	3	2	1	0	⇐ビットNo.
アドレス:000032 <sub>H</sub>	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	ELVR 下位
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)	
要求レベル 設定レジスタ上位	15	14	13	12	11	10	9	8	⇐ビットNo.
アドレス:000033 <sub>H</sub>	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	ELVR 上位
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 15.2-3 要求レベル設定レジスタ(ELVR)

ELVRレジスタは要求検出の選択を行うレジスタです。1つの端子あたり2bitが割り当てられていて、表 15.2-1に示すような対応となっています。レベルに応じて要求を検出する必要がある場合、入力がアクティブになっている間は、レジスタをクリアしてもそのレジスタの値は維持されます。

表 15.2-1 要求レベル設定レジスタ(ELVR)の動作

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	立ち上がりエッジで要求あり
1	1	立ち下がりエッジで要求あり

< 注意事項 >

複数の外部割込み要求出力が許可(ENIR:EN7～EN0=1)されている場合、CPUが割込みを受け付けたビット(ER7～ER0の”1”にセットされているビット)だけを”0”にクリアするようにしてください。それ以外のビットを無条件にクリアすることは避けてください。

## 15.3 DTP/外部割込みの動作

外部割込み要求が設定された後、対応する端子にELVRレジスタで設定された要求が入力された場合、外部割込み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。

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。

### 外部割込みの動作

外部割込み要求が設定された後、対応する端子にELVRレジスタで設定された要求が入力された場合、外部割込み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、同時に発生した割込みの優先順位を識別した結果、このリソースからの割込みが最も優先順位が高かったときに、割込みコントローラはF<sup>2</sup>MC-16LX CPU に対して割込み要求を発生します。

F<sup>2</sup>MC-16LX CPUは、自分の内部にあるCCRレジスタ中のILMビットと割込み要求を比較し、要求レベルがILMビットより高かったときに、現在実行中の命令が終了し次第、ハードウェア割込み処理マイクロプログラムを起動します。

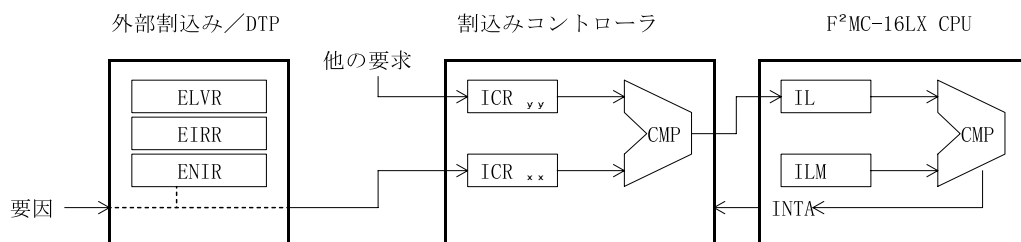


図 15.3-1 外部割込みの動作

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は割込みコントローラからISEビットの情報を読み出し、これにより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。割込み処理用マイクロプログラムでは割込みベクトル領域の読み出しと割込みコントローラへの割込みアクリッジを発生し、ベク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

拡張インテリジェントI/Oサービスを起動するには、最初にユーザプログラムが、拡張インテリジェントI/OサービスディスクリプタのI/Oアドレスポイントに000000hから0000FFhの範囲に割りつけられているレジスタのアドレスを設定します。次に、ユーザプログラムは、バッファアドレスポイント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ISEビットの内容がDTPを示しているので、拡張インテリジェントI/Oサービス処理マイクロプログラムへ制御を移します。拡張インテリジェントI/Oサービスが起動されると、アドレッシングされている外部ペリフェラルにリードまたは、ライト信号が送られ本チップとの転送が行われます。外部ペリフェラルは、その転送が行われてから3マシンサイクル以内に本チップに対する割込み要求を取り下げてください。転送が終了するとディスクリプタの更新などが行われ、その後転送要因をクリアする信号を割込みコ

ントローラに発生させます。転送要因をクリアする信号を受けとった本リソースは、要因を保持しているフリップフロップをクリアして端子からの次の要求に備えます。

拡張インテリジェントI/Oサービス処理の詳細については、MB90500プログラミングマニュアルを参照してください。

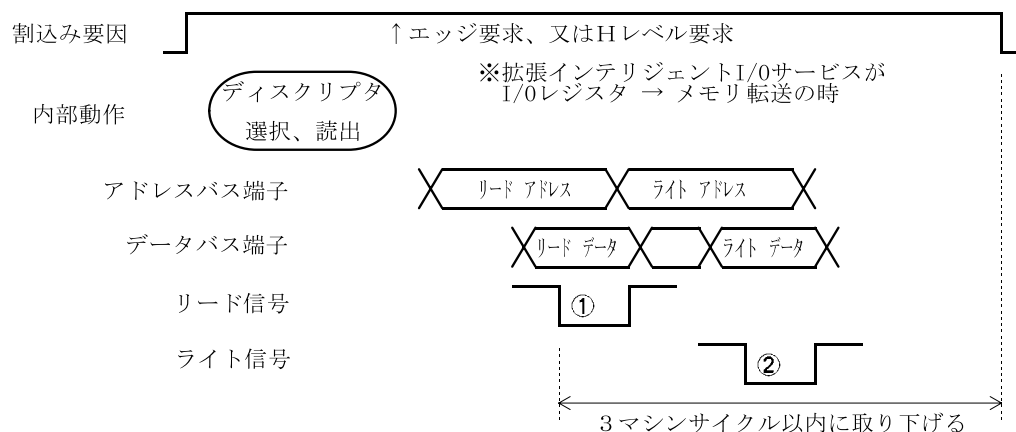


図 15.3-2 DTP動作終了時の外部割込み取下げタイミング

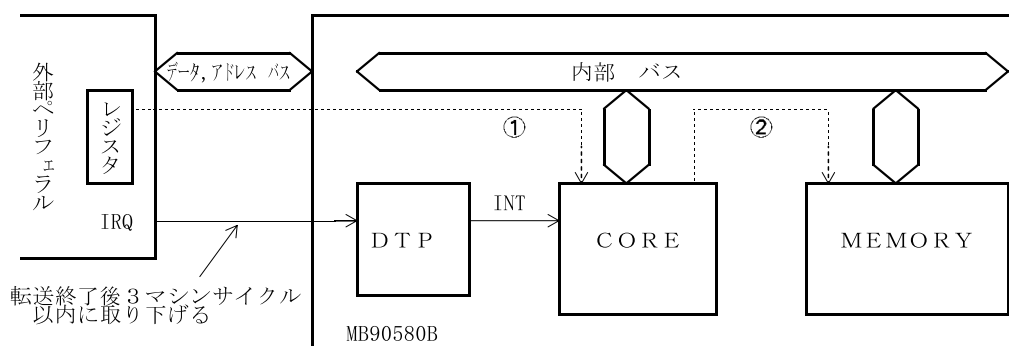


図 15.3-3 外部ペリフェラルとのインタフェースの例

#### 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ICRレジスタ中のISEビットの設定によって行います。

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

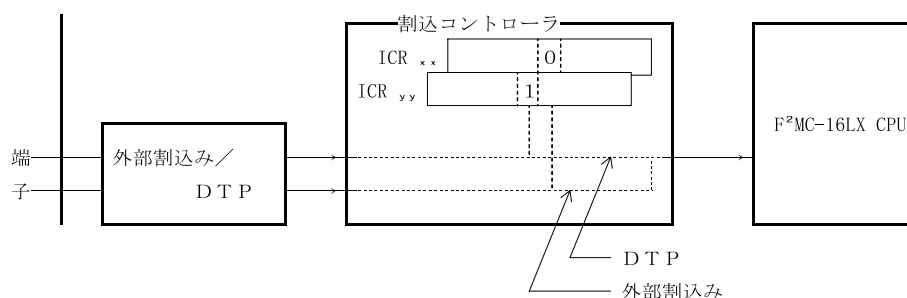


図 15.3-4 外部割込み要求とDTP要求の切替え

## 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つの点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
- ・ 外部割込み/DTPの動作手順
- ・ 外部割込み要求レベル

### DTPを用いた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、転送が完了すると、自動的に要求をクリアするものでなければなりません。また、転送動作が開始してから3マシンサイクル以内で転送要求を取り下げようになっていないと、本リソースは次の転送要求が発生したものとして扱ってしまいます。

### スタンバイからの復帰

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

### 外部割込み/DTPの動作手順

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

- 1) 許可レジスタの対象となるビットをデセーブル状態にする。
- 2) 要求レベル設定レジスタの対象となるビットを設定する。
- 3) 要因レジスタの対象となるビットをクリアする。
- 4) 許可レジスタの対象となるビットを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許可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、許可レジスタをイネーブル状態にする前に必ず要因レジスタをクリアしておく必要があります。これは、レジスタ設定時や割込み許可状態時に、誤って割込み要因が起こってしまうことを避けるためです。

### 外部割込み要求レベル

- 要求レベルがエッジ要求の時、エッジがあったことを検出するためには、パルス幅は最小3マシンサイクル必要とします。
- 要求入力レベルがレベル設定の時、外部より要求入力が入ってその後取り下げられても、割込み要求許可状態(ENIR:EN=1)であれば、割込みコントローラへの要求はアクティブのままです。割込みコントローラへの要求を取り下げるには割込み要求フラグビット(EIRR:ER)をクリアする必要があります。

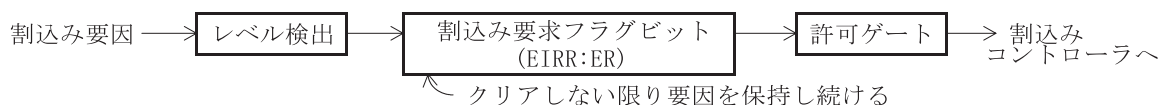


図 15.4-1 レベル設定時の割込み要求フラグビット(EIRR:ER)のクリア

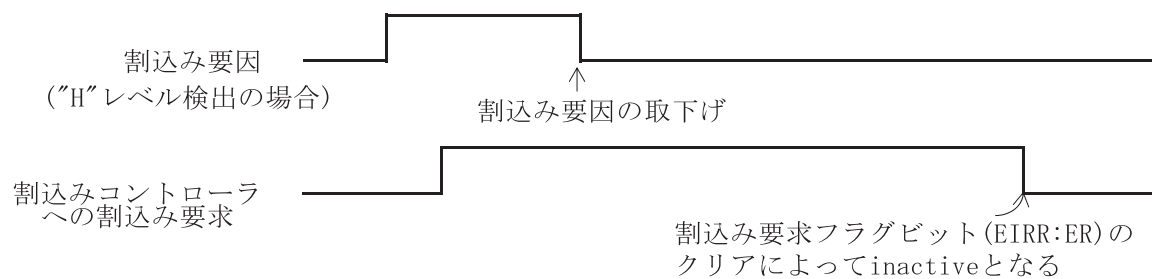


図 15.4-2 割り込み許可時の割り込み要因と割り込みコントローラへの割り込み要求

## 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、タスク切り替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F<sup>2</sup>MC-16LX CPU に対して割込み要求の発生 / 取消を行うことができます。

## 遅延割込み発生モジュールのレジスタ(DIRR)

遅延割込み要因発生 / 取消レジスタ(DIRR) は、遅延割込み要求の発生や取消を制御するレジスタです。このレジスタに"1"を書込むと、遅延割込みの要求を発生させ、"0"を書込むと、遅延割込みの要求を取消します。

リセット時には要因解除状態になります。  
未使用ビット領域には、"0"、"1"どちらの書込みも可能ですが、将来の拡張を考慮してこのレジスタをアクセスする際にはセットビット、クリアビット命令を使用することをお勧めします。

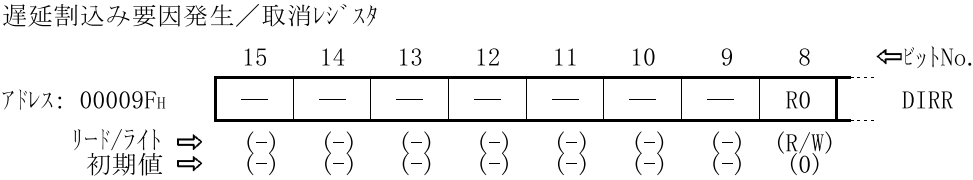


図 16.1-1 遅延割込み要因発生 / 取消レジスタ(DIRR)

## 遅延割込み発生モジュールのブロックダイアグラム

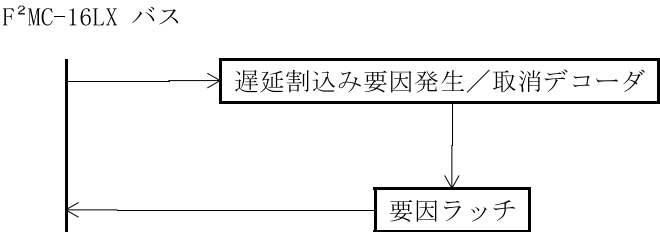


図 16.1-2 遅延割込み発生モジュールのブロックダイアグラム

## 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が発生します。

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F<sup>2</sup>MC-16LX CPUに対して割込み要求が発生します。

### 遅延割込み発生モジュールの動作

F<sup>2</sup>MC-16LX CPUは、自分の内部にあるCCRレジスタのILMビットと割込み要求を比較し、要求レベルがILMビットより高かったときに現在実行中の命令が終了し次第、ハードウェア割込み処理マイクロプログラムを起動します。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。

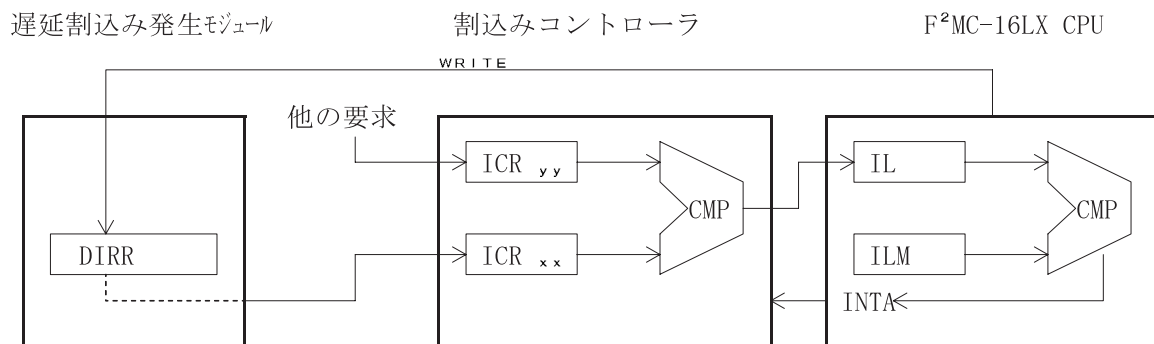


図 16.2-1 遅延割込み発生動作説明

割込み処理ルーチン内で、DIRRレジスタの該当ビットに"0"を書込むことで本割込み要因をクリアし、タスクが切替わります。

### 遅延割込み要求ラッチの使用上の注意

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割込み処理を再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概要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コンバータのブロックダイアグラム
- 17.3 A/Dコンバータのレジスタ
- 17.4 A/Dコンバータの動作
- 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

1チャンネル当たり最小34.7μs（マシンサイクル 12MHz時）

サンプル，ホールド回路付，RC逐次比較変換フォーマットを採用

### 8/10ビット分解能

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の電圧を変換。最大8chプログラム可能
- 連続変換モード：指定チャンネルの電圧を繰り返し変換
- 停止変換モード：1チャンネルの電圧を変換すると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変換終了時の割込み要求

A/D変換終了時には、CPUに対してA/D変換終了の割込み要求を発生させることができます。この割込みの発生によって、EIOSを起動することができ、A/D変換結果データをメモリに転送できますので連続処理に適しています。

起動要因は、ソフト、外部トリガ（立下りエッジ）、タイマ（立上りエッジ）の選択。

## A/Dコンバータ使用上の注意

A/Dコンバータを外部トリガまたは内部タイマを使って起動する場合、ADCS2レジスタのA/D起動要因ビットSTS1,0で設定しますが、このときに外部トリガまたは、内部タイマの入力値はインアクティブであることを確認します。値がアクティブの場合、A/Dコンバータはただちに起動することがあります。

STS1,0設定時は、ADTG = "1"入力、内部タイマ（タイマ2）= "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。

ビットNo.	15	14	13	12	11	10	9	8	初期値
アドレス:00001C	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 17.1-1 アナログ入力に使用する端子の設定

ポート5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。

## 17.2 A/Dコンバータのブロックダイアグラム

図 17.2-1に、A/Dコンバータのブロックダイアグラムを示します。

A/Dコンバータのブロックダイアグラム

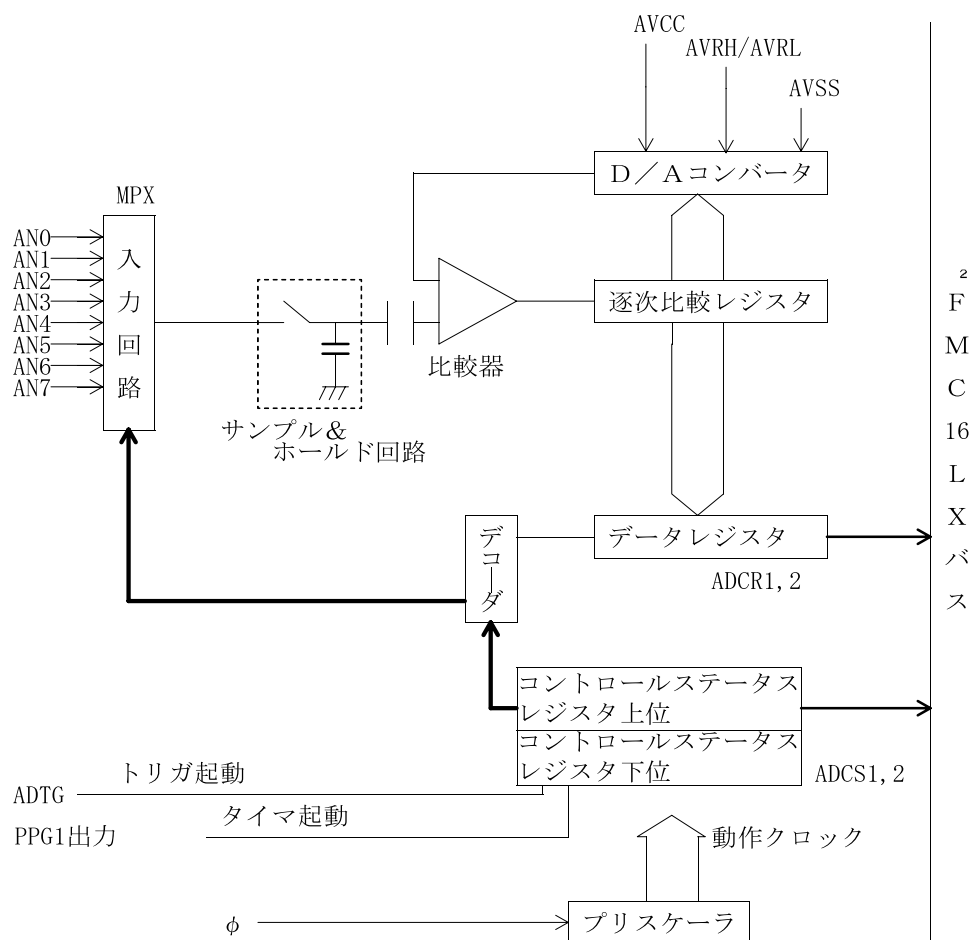


図 17.2-1 A/Dコンバータのブロックダイアグラム

# 17.3 A/Dコンバータのレジスタ

図 17.3-1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: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)	(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)	
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:000039H	SELB	ST1	ST0	CT1	CT0	—	D 9	D 8	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:000038H	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	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 17.3-1 A/Dコンバータのレジスタ

## 17.3.1 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コンバータの制御と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)

ADCS1,2は、A/D変換動作中に書換えないようにしてください。

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)	(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 17.3-2 コントロール・ステータスレジスタ(ADCS0,1)

### < 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)

リード時：

A/Dコンバータ動作表示用のビットです。A/D変換起動でセットされ終了でクリアされます。

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。リード/モディファイ/ライト(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。リセットにより"0"に初期化されます。

### < 注意事項 >

強制停止とソフト起動を同時に行わないでください (BUSYビットに"0"を書き込む時、STRTビットに"1"を書き込まないでください)。

#### 【ビット14】INT(Interrupt)

INTは、データ表示ビットです。変換データがADCRに書き込まれるとセットされます。

Bit5(INTE)が"1"の時に本ビットがセットされると割り込み要求が発生します、またEI<sup>2</sup>OS起動を許可している場合EI<sup>2</sup>OSが起動されます。"1"書き込みは意味を持ちません。

クリアは"0"書き込みとEI<sup>2</sup>OS割り込みクリア信号で行われます。リセットにより"0"に初期化されます。

## &lt; 注意事項 &gt;

---

本ビットの"0"書込みクリアはA/D停止中に行ってください。

---

## 【ビット13】INTE(INTerrupt Enable)

INTEビットは、変換終了による割込みの許可・不許可を指定します。

EI<sup>2</sup>OSを使用するときは本ビットをセットしてください。EI<sup>2</sup>OSは割込み要求発生で起動するようになっています。リセットにより"0"に初期化されます。

表 17.3-1 INTE(割込みの許可・禁止指定ビット)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EI<sup>2</sup>OSで転送しなければ結果データが連続して書き換えられ、前データが書き換えられてしまいます。

これを避けるため、データレジスタの値をEI<sup>2</sup>OSで転送しなければ、次の変換データが格納されないようになっています。この間A/D変換動作は停止します。EI<sup>2</sup>OSで転送を終了するとA/D変換は再開します。リセットにより"0"に初期化されます。

## &lt; 注意事項 &gt;

---

このビットはEI<sup>2</sup>OSを使用したときのみに有効です。動作説明の変換データ保護機能を参照してください。

---

## 【ビット11, 10】STS1, STS0(Start Source select)

STS1, STS0ビットの設定により、A/D起動要因を選択します。

表 17.3-2 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

複数の起動要因を許可するモードでは、最初に入力された要因でA/Dを起動します。起動要因は書換えと同時に変わるので、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替えてください。

外部端子トリガは立下がりエッジを検出します。

外部トリガ入力レベルが"L"の時に、本ビットを書き換えて外部端子トリガ起動に設定すると、A/Dが起動する場合があります。

タイマ選択時は、16ビットリロードタイマ1(PPG1)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。リセットにより、"0"に初期化されます。

バイト/ワード命令では"1"が読み出されます。

リードモディファイライト系命令では"0"が読み出されます。

## &lt; 注意事項 &gt;

---

強制停止とソフト起動を同時に行わないでください( STRTビットに"1"を書き込む時、BUSYビットに"0"を書き込まないでください)。

---

## 【ビット8】予約ビット

ビット8は、予約ビットです。ADCS1を設定する時は、必ず"0"を書き込んでください。

## 【ビット7,6】MD1,MD0(a/d converter MoDe set)

MD1,MD0ビットは、A/Dコンバータの動作モードを設定するためのビットです。

表 17.3-3 MD1,MD0の動作モード

MD1	MD0	動作モード
0	0	単発モード、動作中の再起動可能 [ 初期値 ]
0	1	単発モード、動作中の再起動禁止
1	0	連続モード、動作中の再起動禁止
1	1	停止モード、動作中の再起動禁止

## 単発モード

ANS2～ANS0の設定チャンネルからANE2～ANE0の設定チャンネルまでA/D変換を連続して行い1回変換が終了したら停止する。

## 連続モード

ANS2～ANS0の設定チャンネルからANE2～ANE0の設定チャンネルまでA/D変換を繰り返し行います。

## 停止モード

ANS2～ANS0の設定チャンネルからANE2～ANE0の設定チャンネルまで1chごとにA/D変換を行い一時停止します。変換再開は起動要因発生によって行われます。リセットにより、"00"に初期化されます。

## &lt; 注意事項 &gt;

- 
- ・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"を書き込む事により行われます。
  - ・単発、連続、停止の各モードの再起動は、起動要因(タイマ、外部トリガ、ソフト)に関わらず禁止されます。
- 

## 【ビット5,4,3】ANS2,ANS1,ANS0(Avalog Start channel set)

ANS2,ANS1,ANS0ビットは、A/D変換の開始チャンネルを指定するために使用します。A/Dコンバータを起動すると、このビットで選択されたチャンネルからA/D変換を開始します。



表 17.3-4 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

## &lt; 注意事項 &gt;

本ビット群は、リード時において、A/D変換中は変換チャンネル番号が読めますが、A/D変換停止中は、前にA/D変換したチャンネル番号が読めます。本レジスタに値を設定した場合でも、A/D変換が始まるまでは、設定した値ではなく、前にA/D変換した値が読めます。本ビットの読み出し値は、A/D変換が始まるまでは、前回の変換チャンネル番号が読めます。リセット時は、"000<sub>B</sub>"に初期化されます。

【ビット2, 1, 0】ANE2, ANE1, ANE0(ANalog End channel set)

ANE2, ANE1, ANE0ビットは、A/D変換の終了チャンネルを指定するために使用します。

表 17.3-5 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

## &lt; 注意事項 &gt;

- ANE2 ~ ANE0とANS2 ~ ANS0に同じチャンネルを書き込むと、1chのみ変換します。(シングル変換)
- 連続モード、または停止モードでは、ANE2 ~ ANE0で指定したチャンネルの変換が終わるとANS2 ~ ANS0で指定した開始チャンネルに戻ります。
- ANSの値がANEの値より小さい場合は、ANSチャンネルから変換が始まり、7chまで変換したら0chに戻りANEチャンネルまで変換します。
- リセットにより、"000<sub>B</sub>"に初期化されます。

【例】チャンネル設定 ANS = 6ch, ANE = 3chで単発モードの時

次のシーケンスで変換実行 6ch 7ch 0ch 1ch 2ch 3ch

17.3.2 データレジスタ(ADCR1,2)

データレジスタ(ADCR1,2)では，分解能の選択やマシンサイクルを設定します。

データレジスタ(ADCR1,2)

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:000039H	SELB	ST1	ST0	CT1	CT0	—	D 9	D 8	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:000038H	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	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 17.3-3 データレジスタ(ADCR1,2)

< 注意事項 >

ADCR2の読み出し値は不定です。

【ビット15】SELB

SELBビットでは，8/10bitの分解能を選択します。

表 17.3-6 SELBビットの分解能

SELB	分解能
0	10bit
1	8bit

【ビット14，13】ST1，ST0(Sampling Time)

ST1，ST0は，サンプリング時のマシンサイクル数を設定するために使用するビットです。

表 17.3-7 ST1，ST0(サンプリング時のマシンサイクル数設定ビット)

ST1	ST0	サンプリング時マシンサイクル	サンプリング時間
0	0	64マシンサイクル	4μs / マシンサイクル 16MHz
0	1		予約
1	0		予約
1	1	4096マシンサイクル	256μs / マシンサイクル 16MHz

【ビット12, 11】CT1, CT0(Compare Time)

CT1, CT0ビットは, コンペア時のマシンサイクルを設定するために使用するビットです。

表 17.3-8 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	$22\mu\text{s}$ / マシンクロック 8MHz
0	1	352マシンサイクル	$22\mu\text{s}$ / マシンクロック 16MHz
1	0	予約	
1	1	予約	

< 注意事項 >

本ビットを"00<sub>8</sub>"に設定する時は, マシンクロックを8MHz以上にしないでください。

【ビット9~0】D9~D0

D9~D0ビットには, A/D変換の結果が格納され, レジスタ値は1回の変換終了時ごとに更新されます。通常は最終変換値が格納されます。変換データ保護機能が有ります。「17.4 A/Dコンバータの動作」を参照してください。

このレジスタは, リセットにより不定となります。

< 注意事項 >

A/D動作中に, このレジスタにデータを書き込まないようにしてください。

## 17.4 A/Dコンバータの動作

A/Dコンバータは、逐次比較方式で動作し、8/10ビットの分解能をもっています。このA/Dコンバータは、変換結果格納用のレジスタが1つ(8/10ビット)しかないため、1回の変換終了とともに変換データレジスタ(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EI<sup>2</sup>OS機能を使って変換データをメモリに転送しながら変換することを推奨いたします。

### 単発モード

単発モードは、ANSビットとANEビットで設定されたアナログ入力を順に変換していき、ANEビットで設定された終了チャンネルまで変換が終わるとA/Dは動作を停止します。

開始チャンネルと終了チャンネルが同じとき(ANS = ANE)はANSで指定したチャンネルのみの変換動作になります。

#### 【例】

```
ANS = 000B, ANE = 011B
開始 AN0 AN1 AN2 AN3 終了
ANS = 010B, ANE = 010B
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変換を繰り返します。

開始チャンネルと終了チャンネルが同じとき(ANS = ANE)は、ANSで指定した1ch変換を繰り返します。

#### 【例】

```
ANS = 000B, ANE = 011B
開始 AN0 AN1 AN2 AN3 AN0 繰り返し
ANS = 010B, ANE = 010B
開始 AN2 AN2 AN2 繰り返し
```

連続モードでは、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"を書き込むと動作を強制停止)。

強制停止されると、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、変換レジスタは変換完了した前データが格納されています)。

### 停止モード

停止モードは、ANSビットとANEビットで設定されたアナログ入力を順に変換していきますが、1ch変換する度に一時停止します。一時停止を解除するには、A/Dコンバータをもう一度起動します。

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変換を繰り返します。

開始チャンネルと終了チャンネルが同じとき(ANS = ANE)は、ANSで指定した1chを変換をします。

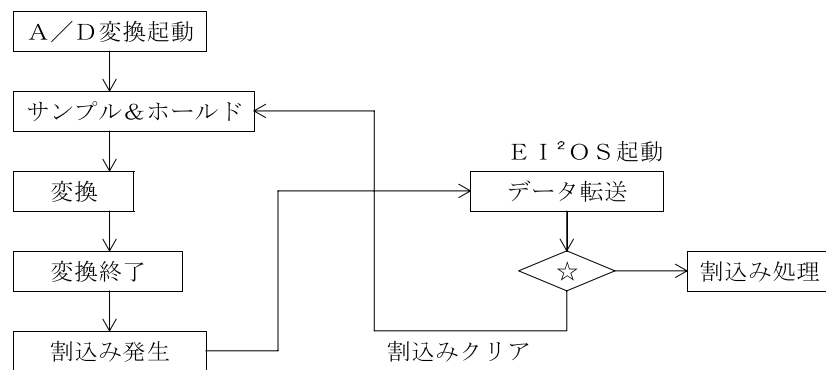
【例】

ANS = 000<sub>B</sub>, ANE = 011<sub>B</sub>  
開始 AN0 終了 再起動 AN1 終了 再起動 AN2 終了 再起動 AN3  
終了 再起動 AN0 繰り返し  
ANS = 010<sub>B</sub>, ANE = 010<sub>B</sub>  
開始 AN2 終了 再起動 AN2 終了 再起動 AN2 繰り返し

この時の起動要因は、STS1,0ビットで設定されたものだけが使用されます。  
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。

EI<sup>2</sup>OSを使った変換動作

図 17.4-1に、A/D変換起動から変換データ転送までのフロー例（連続モード）を示します。



で示す部分は、EI<sup>2</sup>OSの設定で決まります。

図 17.4-1 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）

## 17.4.1 単発モード時のEI<sup>2</sup>OSの起動例

単発モード時のEI<sup>2</sup>OSは、次の手順で起動します。

- ・アナログ入力(AN1～AN3)まで変換して終了する。
- ・変換データは、アドレス200<sub>H</sub>～205<sub>H</sub>番地に順に転送する。
- ・変換開始はソフトで行う。
- ・割込みレベル最強を使用する。

単発モード時のEI<sup>2</sup>OSの起動例

表 17.4-1 単発モード時のEI<sup>2</sup>OSの起動例

設定項目	プログラム例	動作説明
EI <sup>2</sup> OSの設定	MOV ICR00, #08H	最強割込み設定, 割込み時EI <sup>2</sup> 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アドレスの指定。
	MOV BAPM, #02H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送の指定。転送後, 転送先アドレスをインクリメントします。データは, I/Oからメモリへ転送, リソースからの要求で転送を終了しない。
	MOV IOA, #38H	転送元アドレス(A/Dコンバータの結果レジスタ)の設定
	MOV DCT, #03H	EI <sup>2</sup> OS転送を3回行う。このカウントは変換回数と同じにする。
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の指定。
	MOV ADCS2, #A2H	ソフト起動, A/D変換開始の指定。
割込みシーケンス	RETI	割込みからの復帰の指定。

ICR00 : 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 : EI<sup>2</sup>OSステータスレジスタ

IOA : I/Oアドレスレジスタ

DCT : データカウンタ

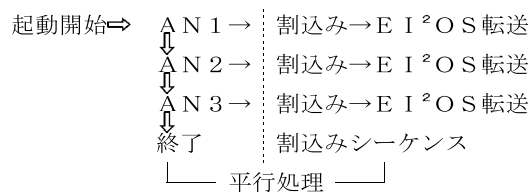


図 17.4-2 単発モード時のEI<sup>2</sup>OSの起動例

## 17.4.2 連続モード時のEI<sup>2</sup>OSの起動例

連続モード時のEI<sup>2</sup>OSは、次の手順で起動します。

- ・アナログ入力( AN3 ~ AN5)の変換をして各チャンネルの変換データを2つ取得する。
- ・変換データはアドレス600<sub>H</sub> ~ 60B<sub>H</sub>番地に順に転送する。
- ・変換開始は外部エッジ入力で行う。
- ・割込みレベル最強を使用する。

連続モード時のEI<sup>2</sup>OSの起動例

表 17.4-2 連続モード時のEI<sup>2</sup>OSの起動例

設定項目	プログラム例	動作説明
EI <sup>2</sup> OSの設定	MOV ICR00, #08H	最強割込み, 割込み時EI <sup>2</sup> OS起動, ディスクリプタアドレスの指定。
	MOV BAPL, #00H	変換データの転送先アドレスの指定。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #18H	ワードデータ転送の指定。転送後転送先アドレスをインクリメントする。データはI/Oからメモリへ転送。リソースからの要求で転送を終了しない。
	MOV I/OA, #38H	転送元アドレス(A/Dコンバータの結果レジスタの設定)
A/Dコンバータ設定	MOV DCT, #06H	EI <sup>2</sup> OS転送を6回行います。3ch×2のデータ転送を行います。
	MOV ADCS1, #9DH	連続モード, 開始チャンネルAN3, 終了チャンネルAN5の指定
EI <sup>2</sup> OS終了割込みシーケンス	MOV ADCS2, #A4H	外部エッジ起動, A/D変換開始の指定
	MOV ADCS2, #00H	割込みからの復帰の指定
	RETI	

ICR00 : 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 : EI<sup>2</sup>OSステータスレジスタ  
IOA : I/Oアドレスレジスタ  
DCT : データカウンタ



図 17.4-3 連続モード時のEI<sup>2</sup>OSの起動例

### 17.4.3 停止モード時のEI<sup>2</sup>OS起動例

停止モード時のEI<sup>2</sup>OS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。
- ・ 変換データはアドレス600<sub>H</sub>～617<sub>H</sub>番地に順に転送する。
- ・ 変換開始は外部エッジ入力で行う。
- ・ 割込みレベル最強を使用。

停止モード時のEI<sup>2</sup>OS起動例

表 17.4-3 停止モード時のEI<sup>2</sup>OS起動例

設定項目	プログラム例	動作説明
EI <sup>2</sup> OSの設定	MOV ICR00, #08H	最強割込み設定, 割込み時EI <sup>2</sup> 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アドレスの指定。
	MOV BAPM, #06H	
	MOV BAPH, #00H	
	MOV ISCS, #08H	ワードデータ転送の指定。転送後転送先アドレスをインクリメントする。データはI/Oからメモリへ転送。リソースからの要求で終了しない。
	MOV IOA, #38H	転送元アドレス
A/Dコンバータ設定	MOV DCT, #0CH	EI <sup>2</sup> OS転送を12回行う。
	MOV ADCS1, #0BH	停止モード, 開始チャンネルAN3, 終了チャンネルAN3 (1ch変換)
	MOV ADCS2, #A4H	外部エッジ起動, A/D変換開始の指定
割込みシーケンス	MOV ADCS2, #00H	割込みからの復帰
	RETI	

ICR00 : 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 : EI<sup>2</sup>OSステータスレジスタ  
 IOA : I/Oアドレスレジスタ  
 DCT : データカウンタ

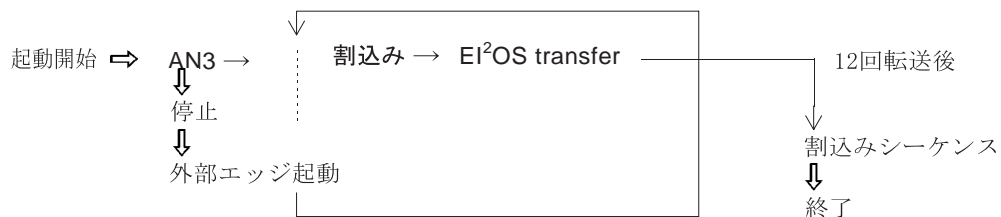


図 17.4-4 停止モード時のEI<sup>2</sup>OS起動例



## 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EI<sup>2</sup>OSを使って連続変換と複数のデータを確保できます。

### 変換データ保護機能

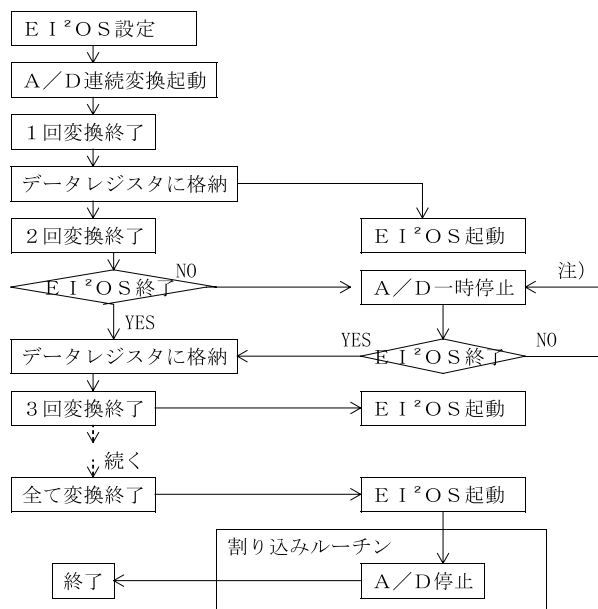
変換データレジスタは1つですので、連続でA/D変換をすると1回の変換終了と共に変換データが格納されて前データが失われます。これを保護するために、A/Dコンバータは前のデータがEI<sup>2</sup>OSによりメモリに転送されていない場合、変換後に一時停止します。変換データは、前データがメモリに転送されるまで格納されません。

一時停止の解除は、EI<sup>2</sup>OSでメモリへ転送された後、行われます。

前データが転送されていれば一時停止することなくA/Dコンバータは動作を連続します。

### < 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込み許可（INTE = "1"）状態でしか動作しないようになっています。  
割込み禁止（INTE = "0"）の場合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。
- ・割込み許可（INTE = "1"）状態でEI<sup>2</sup>OS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コンバータは変換を一時停止します。この場合は割込みシーケンスでINTビットをクリアすると停止状態が解除されます。
- ・EI<sup>2</sup>OS動作中で、A/Dコンバータが一時停止をしているときに割込みを禁止にすると、A/Dコンバータが再起動し転送する前に変換データレジスタの内容が変わることがあります。  
また、一時停止中に再起動をかけると、待機データが壊れます。



(注意)

- ・一時停止中に再起動をかけると待機中の変換データは壊れてしまいます。
- ・A/Dコンバータ動作停止時のフローは省略します。

図 17.5-1 データ保護機能フロー（EI<sup>2</sup>OSを使用時）

## 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コンバータには、2つのチャンネルが存在します。

D/Aコントロールレジスタを用いた2つのチャンネルに対して、個別に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一覧

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ビット									
	15	14	13	12	11	10	9	8	
アドレス：00003BH	DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	DAT1
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)	

D/Aコンバータデータレジスタ 0ビット									
	7	6	5	4	3	2	1	0	
アドレス：00003AH	DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00	DAT0
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)	

D/Aコントロールレジスタ1 ビット									
	15	14	13	12	11	10	9	8	
アドレス：00003DH	-	-	-	-	-	-	-	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								
ビット								
アドレス: 00003CH	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DAE0
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一覧

## 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。

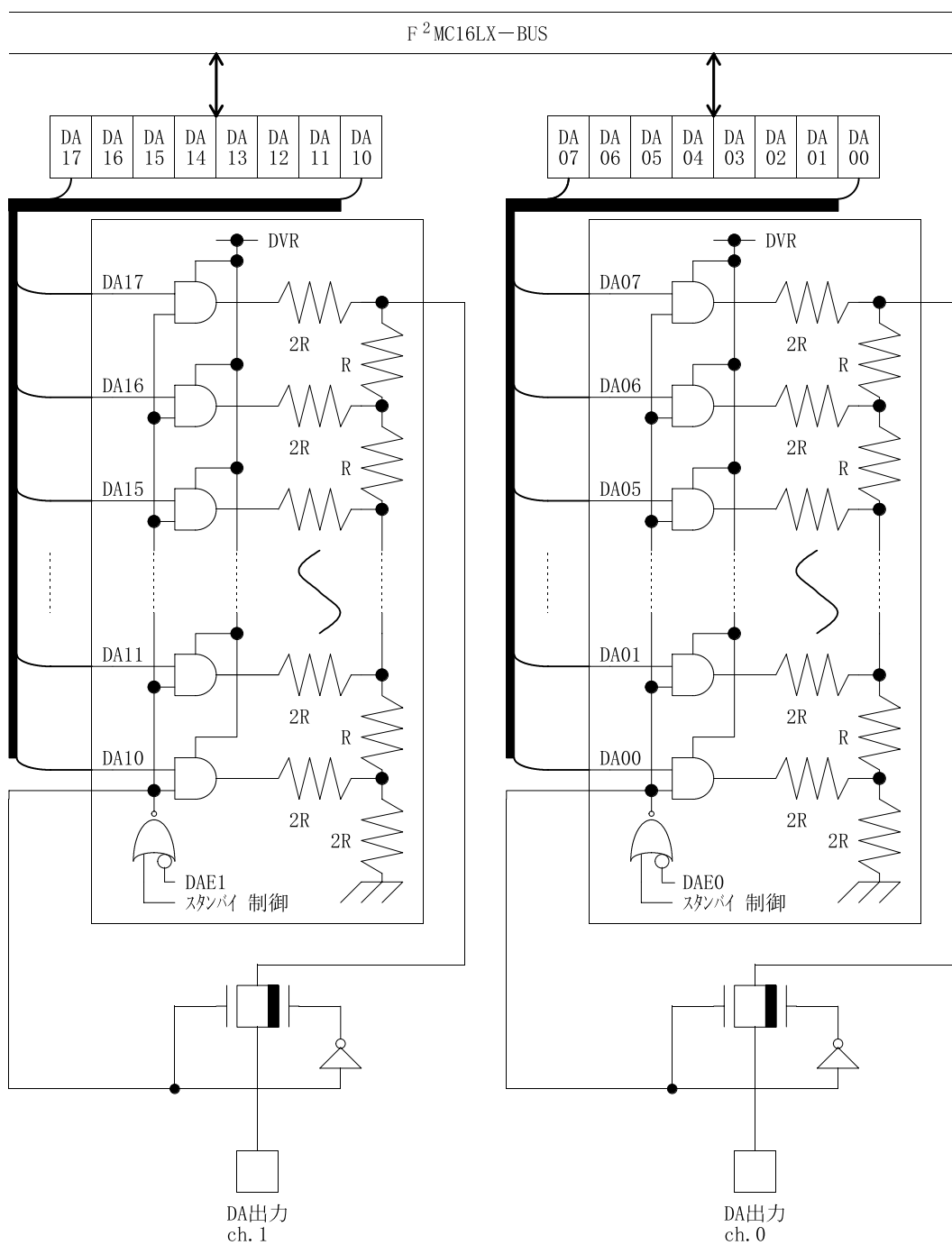


図 18.1-2 D/Aコンバータのブロックダイアグラム

# 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(DAT0/1)
  - ・ D/Aコントロールレジスタ(DACR0/1)

D/Aコンバータレジスタ(DAT0/1)

D/Aコンバータレジスタ(DAT0/1)のレジスタ構成は、図 18.2-1のとおりです。

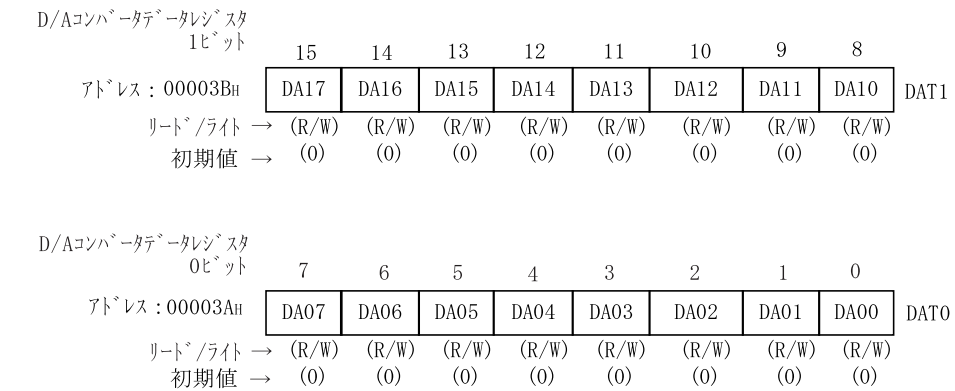


図 18.2-1 D/Aコンバータデータレジスタ(DAT0/1)

[bit 15~8] DA17-DA10

これらのビットを使用して、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。これらのビットの読出しと書込みが可能です。

[bit 7~0] DA07-DA00

これらのビットを使用して、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。これらのビットの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 ビット	15	14	13	12	11	10	9	8	
アドレス : 00003DH	—	—	—	—	—	—	—	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 ビット	7	6	5	4	3	2	1	0	
アドレス : 00003CH	—	—	—	—	—	—	—	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)

[bit 8,0] DAE1,DAE0

これらのビットを使用して、D/Aコンバータの出力の許可または禁止を設定します。DAE1はチャンネル1を制御し、他方の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。これらのビットの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。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 および DA17 ~ DA10 に書き込まれた値	出力電圧の理論値
00 <sub>H</sub>	$0/256 \times \text{DVR} (=0 \text{ V})$
01 <sub>H</sub>	$1/256 \times \text{DVR}$
02 <sub>H</sub>	$2/256 \times \text{DVR}$
▪	▪
FD <sub>H</sub>	$253/256 \times \text{DVR}$
FE <sub>H</sub>	$254/256 \times \text{DVR}$
FF <sub>H</sub>	$255/256 \times \text{DVR}$

## 第19章 通信プリスケータ

---

この章では、通信プリスケータの機能と概要について説明します。  
なお、通信プリスケータの出力は、UARTで使用されます。

---

19.1 通信プリスケータの概要

19.2 通信プリスケータの動作



# 19.1 通信プリスケアラの概要

通信プリスケアラのレジスタ(クロック分周コントロールレジスタ)は,マシ  
ンクロックの分周を制御します。ユーザの設定により,種々のマシ  
ンクロックに対して一定のボーレートが得られるように設計されてい  
ます。  
通信プリスケアラの出力は,UARTで使用されます。

## クロック分周コントロールレジスタ(CDCR)

クロック分周コントロールレジスタ0~4

アドレス: 00002C <sub>H</sub>	15	14	13	12	11	10	9	8	←ビットNo.
00002E <sub>H</sub>	MD	—	—	—	DIV3	DIV2	DIV1	DIV0	CDCR0
000034 <sub>H</sub>									CDCR1
000087 <sub>H</sub>									CDCR2
00008F <sub>H</sub>									CDCR3
リード/ライト ⇒	(R/W)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)	CDCR4
初期値 ⇒	(0)	(—)	(—)	(—)	(1)	(1)	(1)	(1)	

図 19.1-1 クロック分周コントロールレジスタ [ 0~4 ]

【ビット15】MD(Machine clock device moDe select)

MDは,通信プリスケアラの動作許可ビットです。

表 19.1-1 MD(Machine clock device moDe select)ビットの機能

MD	機能
0	通信プリスケアラは停止します。[ 初期値 ]
1	通信プリスケアラは動作します。

【ビット11, 10, 9, 8】DIV3~0(DIVide 3~0)

DIV3~0は,マシ  
ンクロックの分周比を決定します。

表 19.1-2 DIV3~0(DIVide 3~0) ビットの機能

DIV3	DIV2	DIV1	DIV0	分周比
1	1	1	1	設定禁止 [ 初期値 ]
1	1	1	0	2分周
1	1	0	1	3分周
1	1	0	0	4分周
1	0	1	1	5分周
1	0	1	0	6分周
1	0	0	1	7分周
1	0	0	0	8分周

(注)

- ・実際の使用時は,"1111<sub>2</sub>"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分周期分の時間を置いてから通信してください。

表 19.1-3 通信プリスケラとUARTの各チャネルの対応

通信プリスケラの設定	通信プリスケラの出力
CDCR 0	UART 0
CDCR 1	UART 1
CDCR 2	UART 2
CDCR 3	UART 3
CDCR 4	UART 4

## 19.2 通信プリスケアラの動作

使用するマシクロック $\Phi$ に応じて、クロック分周コントロールレジスタを表 19.2-1のように設定してください。詳細は、「第20章 UART」を参照してください。

通信プリスケアラの動作

表 19.2-1 通信プリスケアラの動作

マシクロック $\Phi$	div	DIV3	DIV2	DIV1	DIV0	$\Phi \div \text{div}$
4MHz	4	1	1	0	0	1MHz
6MHz	6	1	0	1	0	
8MHz	8	1	0	0	0	
6MHz	3	1	1	0	1	2MHz
8MHz	4	1	1	0	0	
10MHz	5	1	0	1	1	
12MHz	6	1	0	1	0	
14MHz	7	1	0	0	1	
16MHz	8	1	0	0	0	
8MHz	2	1	1	1	0	4MHz
12MHz	3	1	1	0	1	
16MHz	4	1	1	0	0	

上記以外のマシクロックとdivの設定で使用する場合、 $\Phi \div \text{div}$ は、最大4.25 MHzを越えないようにしてください。

## 第20章 UART

---

この章では，UARTの機能と動作について説明します。

---

- 20.1 UARTの概要
- 20.2 UARTのブロックダイアグラム
- 20.3 UARTのレジスタ
- 20.4 UARTの動作
- 20.5 UART(モード1使用時)の応用例

# 20.1 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）通信，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 20.1-1 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

\*：内部マシナクロックが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）
- HRz符号転送信号

## 20.2 UARTのブロックダイアグラム

図 20.2-1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

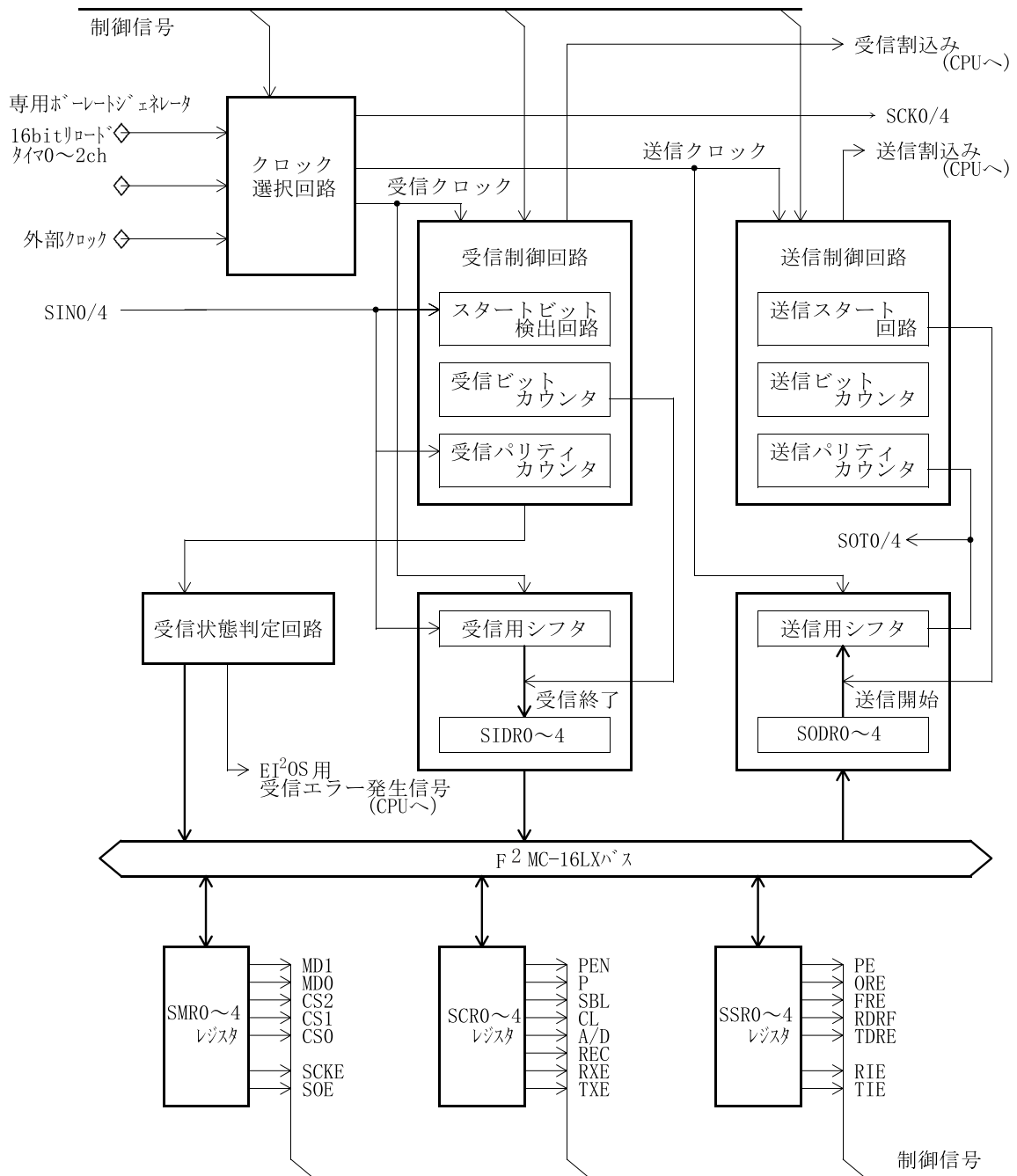


図 20.2-1 UARTのブロックダイアグラム

## 20.3 UARTのレジスタ

UARTのレジスタには、次の4つの種類があります。

- ・シリアルモードレジスタ
- ・シリアルコントロールレジスタ
- ・シリアルインプットレジスタ/シリアルアウトプットレジスタ
- ・シリアルステータスレジスタ

### UARTのレジスタ

#### シリアルモードレジスタ

アドレス: 000020H 000024H 000028H 000082H 000088H	7	6	5	4	3	2	1	0	⇐ビットNo.
	MD1	MD0	CS2	CS1	CS0	予約	SCKE	SOE	SMR0~4
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

アドレス: 000021H 000025H 000029H 000083H 000089H	15	14	13	12	11	10	9	8	⇐ビットNo.
	PEN	P	SBL	CL	A/D	REC	RXE	TXE	SCR0~4
リード/ライト ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値 ⇒	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	

#### シリアルインプットレジスタ/シリアルアウトプットレジスタ

アドレス: 000022H 000026H 00002AH 000084H 000088H	7	6	5	4	3	2	1	0	⇐ビットNo.
	D7	D6	D5	D4	D3	D2	D1	D0	SIDR0~4(read) SODR0~4(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

アドレス: 000023H 000027H 00002BH 000085H 00008BH	15	14	13	12	11	10	9	8	⇐ビットNo.
	PE	OPE	FRE	RDRF	TDRE	—	RIE	TIE	SSR0~4
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 20.3-1 UARTのレジスタ

20.3.1 シリアルモードレジスタ(SMR0～4)

SMR0～4レジスタは、UARTの動作モードを指定します。  
動作モードの設定は動作停止中に行い、動作中には、このレジスタへ書込まないでください。

シリアルモードレジスタ(SMR0～4)

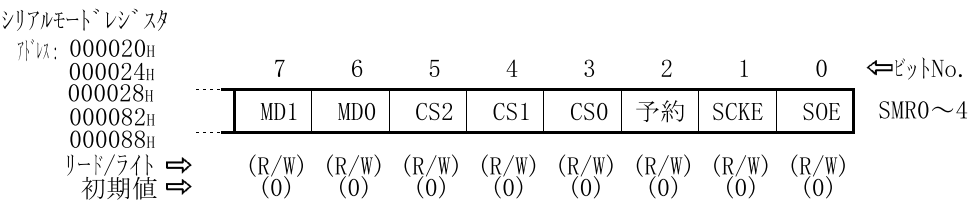


図 20.3-2 シリアルモードレジスタ(SMR)の構成

【ビット7，6】MD1，MD0(MoDe select)

MD1，MD0ビットは、UARTの動作モードを選択します。

表 20.3-1 MD0,1(動作モードを選択ビット)

MD1	MD0	モード	動作モード
0	0	0	非同期ノーマルモード [ 初期値 ]
0	1	1	非同期マルチプロセッサモード
1	0	2	CLK同期モード
1	1	-	設定禁止

< 注意事項 >

CLK非同期マルチプロセッサモード（モード1）では、2つ以上のスレーブCPUを1台のホストCPUに接続します。このリソースでは、受信データのフォーマットを識別することができません。したがって、このリソースがマルチプロセッサモードでサポートするのは、マスタだけに限られます。また、パリティチェック機能は使用できませんので、SCRレジスタのPENは"0"に設定してください。

【ビット5～3】CS2，CS1，CS0(Clock Select)

CS2～0ビットを使用して、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する場合には、同時にボーレートが決定されます。

表 20.3-2 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 <sub>B</sub> ～100 <sub>B</sub>			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
1	1	1	外部クロック



## &lt; 注意事項 &gt;

内部タイマを選択した場合、MB90580Cシリーズでは、3つの16ビットリロードタイマを選択します。

UARTのチャンネルとリロードタイマのチャンネルは以下のようになります。

- ・ UART 0ch : リロードタイマ0ch
- ・ UART 1ch : リロードタイマ1ch
- ・ UART 2ch : リロードタイマ2ch
- ・ UART 3ch : リロードタイマ0ch
- ・ UART 4ch : リロードタイマ1ch

## 【ビット2】予約

ビット2は、予約ビットです。必ず"0"に設定してください。

## 【ビット1】SCKE(SCLK Enable)

SCKEビットは、CLK同期モード（モード2）で通信する場合、このビットを使用し、シリアルクロック入出力端子(SCK)をクロック入力端子として使用するか、クロック出力端子として指定するかを指定します。

表 20.3-3 SCKE (SCLK Enable) ビットの機能

SCKE	機能
0	クロック入力端子としてSCK端子を使用します。[ 初期値 ]
1	クロック出力端子としてSCK端子を使用します。

## &lt; 注意事項 &gt;

クロック入力端子としてSCK端子を使用するには、外部クロックソースを使用している必要があります。

UARTのチャンネルとシリアルクロック入出力端子は以下のように対応しています。

- ・ UART 0ch : SCK0端子
- ・ UART 1ch : SCK1端子
- ・ UART 2ch : SCK2端子
- ・ UART 3ch : SCK3端子
- ・ UART 4ch : SCK4端子

## 【ビット0】SOE(Serial Output Enable)

SOEビットは、外部端子を、シリアル出力端子(SOT)として使用するか、I/Oポート端子として指定するかを指定します。

表 20.3-4 SOE (Serial Output Enable) ビットの機能

SOE	機能
0	汎用I/Oポート端子として外部端子を使用します。[ 初期値 ]
1	シリアルデータ出力端子(SOT)として外部端子を使用します。

## &lt; 注意事項 &gt;

UARTのチャンネルとシリアルデータ出力端子は以下のように対応しています。

- ・ UART 0ch : SOT0端子
- ・ UART 1ch : SOT1端子
- ・ UART 2ch : SOT2端子
- ・ UART 3ch : SOT3端子
- ・ UART 4ch : SCK4端子

20.3.2 シリアルコントロールレジスタ(SCR0～4)

シリアルコントロールレジスタ(SCR0～4)は、シリアル通信する場合の転送プロトコルを制御します。

シリアルコントロールレジスタ(SCR0～4)

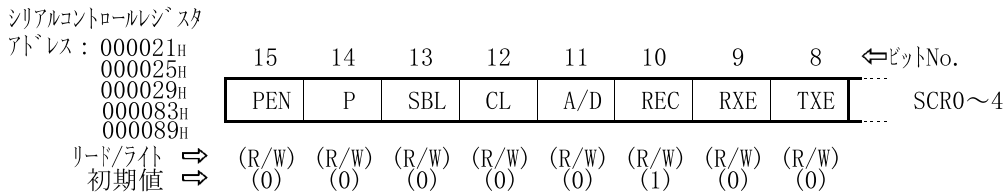


図 20.3-3 シリアルコントロールレジスタ(SCR0～4)の構成

【ビット15】PEN(Parity Enable)

PENビットは、パリティビットを用いてシリアルデータ通信を実行するかどうかを指定します。

表 20.3-5 PEN (Parity Enable)ビットの機能

PEN	機能
0	パリティなし [ 初期値 ]
1	パリティあり

< 注意事項 >

パリティを付加できるのは、通常の非同期通信モード(モード0)に限られます。マルチプロセッサモード(モード1)または、CLK 同期通信モード(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)

P ビットは、パリティを付加してデータ通信を行うとき、偶数パリティまたは、奇数パリティを指定します。

表 20.3-6 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ビットを使用し、ストップビット長を指定します。非同期通信では、フレームエンドマークとしてストップビットを指定します。

表 20.3-7 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)

CLビットでは、送受信の対象となる各フレームのデータ長を指定します。

表 20.3-8 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

## &lt; 注意事項 &gt;

7ビットデータを扱えるのは、通常の同期通信モード（モード0）に限られます。マルチプロセッサモード（モード1）または、CLK 同期通信（モード2）では、8ビットデータを指定します。

## 【ビット11】A/D（Address/Data）

A/Dビットを使用し、マルチプロセッサ非同期通信モード（モード1）で送信の対象となるフレームのデータ形式を指定します。

表 20.3-9 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）

"0"書込みで、SSRレジスタのエラーフラグ（PE,ORE,FRE）をクリアします。  
"1"の書込みは無効で、読出し値は常に"1"になります。

## 【ビット9】RXE（Receiver Enable）

RXEビットを使用して、UARTの受信動作を制御します。

表 20.3-10 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。

## &lt; 注意事項 &gt;

データの受信中（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は、当該フレームの受信が完了し、受信データが、受信データバッファ(SIDRレジスタ)に格納された時点で、送信動作を停止します。

## 【ビット8】TXE（Transmitter Enable）

TXEビットは、UARTの送信動作を制御します。

表 20.3-11 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。

## &lt; 注意事項 &gt;

データの送信中（送信シフトレジスタからデータが出力されているとき）に送信動作を禁止した場合は、送信データバッファ(SODRレジスタ)に全データが出力された後に、送信動作を停止します。

### 20.3.3 シリアルインプットデータレジスタ(SIDR0～4)/ シリアルアウトプットデータレジスタ(SODR0～4)

シリアルインプットデータレジスタ(SIDR0～4)/シリアルアウトプットデータレジスタ(SODR0～4)は、受信/送信用のデータバッファレジスタです。

シリアルインプットデータレジスタ(SIDR0～4)/シリアルアウトプットデータレジスタ(SODR0～4)の構成

SIDR、SODRのデータ長が7bitの場合は、上位1bit(D7)は無効データとなります。  
SODRレジスタへの書込みは、SSRレジスタのTDREが必ず"1"を書込んだ状態で行ってください。

シリアルインプットレジスタ/シリアルアウトプットレジスタ

アドレス : 000022H										
000026H										
00002AH	.....	7	6	5	4	3	2	1	0	⇐ビットNo.
000084H	.....	D7	D6	D5	D4	D3	D2	D1	D0	
00008AH	.....									SIDR0～4(read) SODR0～4(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)	

図 20.3-4 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

< 注意事項 >

このアドレスへのデータの書込みは、SODRレジスタへの書き込みを、読み出しはSIDRレジスタの読み出しを意味します。

### 20.3.4 シリアルステータスレジスタ(SSR0～4)

シリアルステータスレジスタ(SSR0～4)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(SSR0～4)

シリアルステータスレジスタ  
アドレス：000023H  
000027H  
00002BH  
000085H  
00008BH

	15	14	13	12	11	10	9	8	↩ ビットNo.
	PE	ORE	FRE	RDRF	TDRE	—	RIE	TIE	SSR0～4
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 20.3-5 シリアルステータスレジスタ(SSR0～4)の構成

【ビット15】PE (Parity Error)

PEビットは，受信時にパリティエラーが発生したときにセットされる，割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，SCRレジスタの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，SIDRレジスタのデータは無効となります。

表 20.3-12 PE (Parity Error)ビットの機能

PE	機能
0	パリティエラーなし [ 初期値 ]
1	パリティエラーが発生

【ビット14】ORE (Over Run Error)

OREビットは，受信時にオーバーラン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，SCRレジスタの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，SIDRレジスタのデータは無効となります。

表 20.3-13 ORE (Over Run Error) の機能

ORE	機能
0	オーバーランエラーなし [ 初期値 ]
1	オーバーランエラー発生

【ビット13】FRE (FRaming Error)

FREは，受信時にフレーミングエラーが発生したときにセットされる，割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，SCRレジスタの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，SIDRレジスタのデータは無効となります。

表 20.3-14 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)

RDRFビットは，SIDRレジスタに受信データがあることを示す，割込み要求フラグです。SIDRレジスタに受信データがロードされるとセットされ，SIDRレジスタを読み出すと自動的にクリアされます。

表 20.3-15 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)

TDREビットは，SODRレジスタに送信データを書き込めることを示す，割込み要求フラグです。SODRレジスタに送信データを書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，次の送信データを書き込めることを表します。

表 20.3-16 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10】空きビット

【ビット9】RIE (Receiver Interrupt Enable)

RIEビットは，受信割込みを制御します。

表 20.3-17 RIE (Receiver Interrupt Enable) の機能

RIE	機能
0	割込みを禁止します。 [ 初期値 ]
1	割込みを許可します。

< 注意事項 >

受信割込み要因は，PE，ORE，FREによるエラーのほか，RDRFによる正常受信があります。

【ビット8】TIE (Transmitter Interrupt Enable)

TIEビットは，送信割込みを制御します。

表 20.3-18 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による送信要求があります。

# 20.4 UARTの動作

UARTは、表 20.4-1に示す動作モードを持ち、SMRレジスタ、SCRレジスタに値を設定するとモードを切り替えることができます。

UARTの動作

表 20.4-1 UARTの動作モード

モード	パリティ	データ長	動作モード	ストップビット長
0	有/無	7	非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）マルチプロセッサモード	
2	無	8	CLK同期モード	無

< 注意事項 >

- ・非同期（調歩同期）モードは、送信動作に限り、ストップビット長を指定することが可能です。受信動作では、ストップビットは常に1ビット長となります。表 20.4-1に示すモード以外では動作しませんので、設定しないでください。
- ・CLK同期モードを使用中の場合、データバイトに、スタートとストップビットは付加されません。
- ・通信モードの設定は、動作停止中に行ってください。モード設定時の送受信したデータは保証されません。

拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)

EI<sup>2</sup>OSにつきましては、「3.6 拡張インテリジェントI/Oサービス(EI<sup>2</sup>OS)」を参照してください。

## 20.4.1 UARTのクロック選択

UARTのクロックは、次の3種類から選択できます。

- ・専用ボーレートジェネレータ
- ・内部タイマ
- ・外部クロック

### 通信プリスケアラ

専用ボーレートジェネレータを選択したときのボーレートは、表 20.4-2と表 20.4-3のようになります。表中の計算式については、「第19章 通信プリスケアラ」を参照してください。

表 20.4-2 ボーレート(非同期時(調歩同期))

CS2	CS1	CS0	$\phi \div \text{div}=2\text{MHz}$	$\phi \div \text{div}=4\text{MHz}$	算出式
0	0	0	9615 bps	19230 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2)$
0	0	1	4808 bps	9615 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^2)$
0	1	0	2404 bps	4808 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^3)$
0	1	1	1202 bps	2404 bps	$(\phi \div \text{div}) / (8 \times 13 \times 2^4)$
1	0	0	31250 bps	62500 bps	$(\phi \div \text{div}) / 2^6$

$\phi$ : マシクロック       $\text{div}$ : 通信プリスケアラの設定

表 20.4-3 ボーレート(CLK同期時)

CS2	CS1	CS0	$\phi \div \text{div}=2\text{MHz}$	$\phi \div \text{div}=4\text{MHz}$	算出式
0	0	0	1M bps	2M bps	$(\phi \div \text{div}) / 2$
0	0	1	500K bps	1M bps	$(\phi \div \text{div}) / 2^2$
0	1	0	250K bps	500K bps	$(\phi \div \text{div}) / 2^3$
0	1	1	125K bps	250K bps	$(\phi \div \text{div}) / 2^4$
1	0	0	62.5K bps	125K bps	$(\phi \div \text{div}) / 2^5$

$\phi$ : マシクロック       $\text{div}$ : 通信プリスケアラの設定

### 内部タイマ

SMRレジスタのCS2～0ビットを"110<sub>2</sub>"に設定して内部タイマを選択した場合は、16ビットリロードタイマ(タイマ0～2)をリロードモードで動作させます。このときのボーレートの算出式は、次の通りです。

非同期(調歩同期)       $(\phi \div N) / (16 \times 2 \times (n + 1))$

CLK同期       $(\phi \div N) / (2 \times (n + 1))$

$\phi$ : マシクロック

$\phi$ は、マイコンの内部動作周波数を示します。詳細は、「5.7 マシクロックの切り替え」を参照してください。

N

Nは、内蔵の16bitリロードタイマのカウントクロックソース(TMCSR:CSL1, CSL0)の分周比を示します。

例: CSL1,0="01<sub>2</sub>"の時のN=8となります。

詳細は、「13.2.1 タイマコントロールステータスレジスタ(TMCSR)」を参照願います。



n

n は、内蔵の16bitリロードタイマのリロード値です。

マシクロックを 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 20.4-4に示します。

表 20.4-4 ボーレートとリロード値

ボーレート	リロード値	
	N = 2 (マシクロックの2分周)	N = 2 <sup>3</sup> (マシクロックの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0~2)を選択すると、16ビットリロードタイマ0~2の出力(TOT0~2)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ビットリロードタイマ0~2の外部端子(TOT0~2)をUARTの外部クロック入力端子(SCK0~4)に外部で接続する必要はありません。また、16ビットリロードタイマ0~2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<sub>B</sub>"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、次のようになります。外部クロックの周波数は、f とします。

非同期(調歩同期):  $f/16$

CLK同期 : f

ただし、fは最大2MHzです。

## 20.4.2 非同期（調歩同期）モード

非同期（調歩同期）モードでは、転送データは必ずスタートビット（"L"レベルデータ）から始まり、ストップビット（"H"レベルデータ）で終了します。

受信動作は、SCRレジスタで制御され、送信動作は、SODRレジスタで制御されます。

### 転送データフォーマット

UARTは、NRZ（Non Return to Zero）形式のデータのみを扱います。図 20.4-1 に、転送データフォーマットを示します。

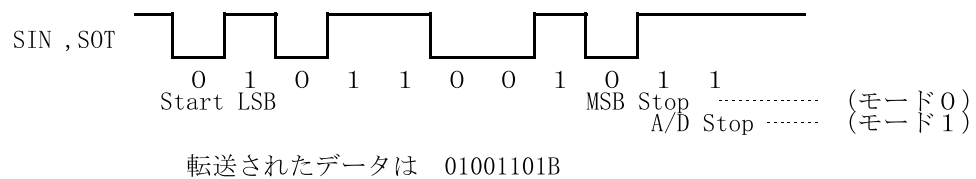


図 20.4-1 転送データフォーマット（モード0, 1）

転送データは必ずスタートビット（"L"レベルデータ）から始まり、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(bit9)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あと、シリアルステータスレジスタ(SSR)内のRDRFフラグ(bit12)がセットされます。この場合、同じSSRレジスタのRIEビット(bit9)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。

## スタートビット検出方法

スタートビットを検出するには、以下のように設定してください。

- 通信期間直前は通信線を必ず“H”(マークレベルを付加)にしてください。
- 通信線が“H”(マークレベル)の期間に、受信許可(RXE=H)にしてください。
- 非通信期間中(マークレベルを除く)は、受信許可(RXE=H)にしないでください。正しいデータが受信できません。
- ストップビット検出後(RDRFフラグが“1”にセットされた後)、通信線が“H”(マークレベル)の間に受信禁止(RXE=L)にしてください。

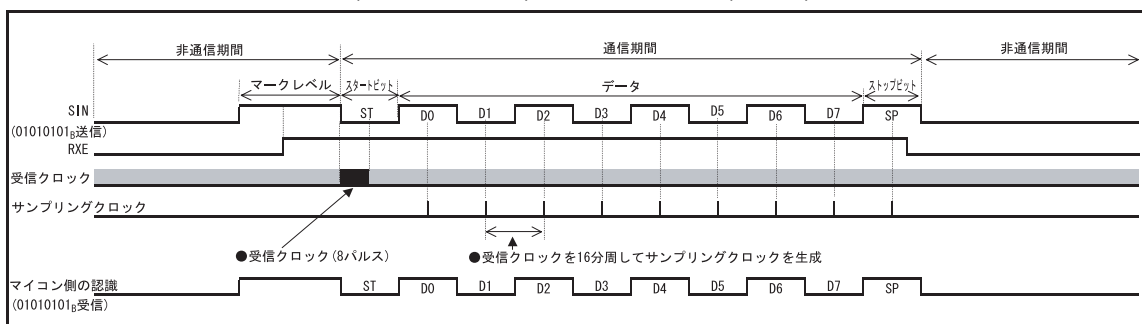


図 20.4-2 正常動作

以下の例のようなタイミングで受信許可に設定しますとマイコン側で入力データ(SIN)が正しく認識されませんので注意してください。

- 通信線が“L”の期間に、受信許可(RXE=H)に設定した場合の動作例

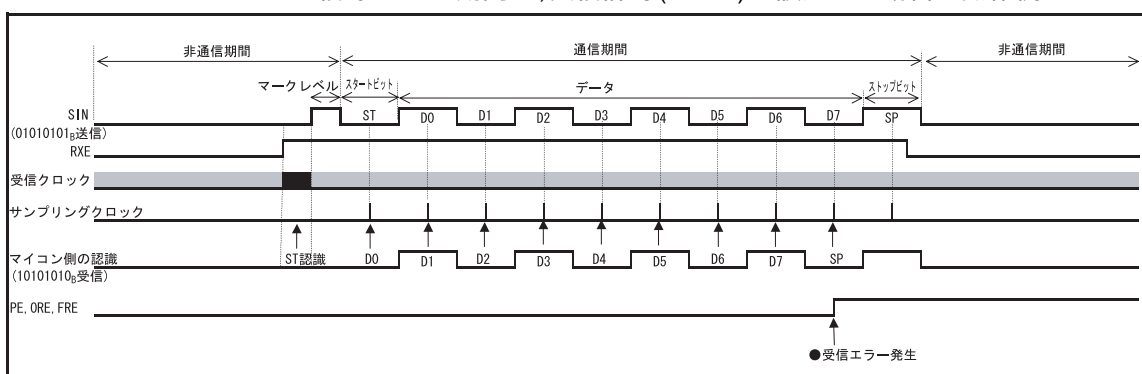


図 20.4-3 異常動作

## 送信動作

シリアルコントロールレジスタ(SCR)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、SCRレジスタ内の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クリアされます。

### 20.4.3 CLK同期モード

CLK同期モードでは、内部クロックを選択した場合は、データ受信用同期クロックが自動的に生成され、外部クロックを選択した場合は、1バイト分のクロックを供給する必要があります。

通信の開始は、SODRレジスタで制御され、通信の終了は、SSRレジスタで制御されます。

#### 転送データフォーマット

UARTは、NRZ (Non Return to Zero) 形式のデータのみを扱います。図 20.4-4 に、送受信クロックとデータとの関係を示します。

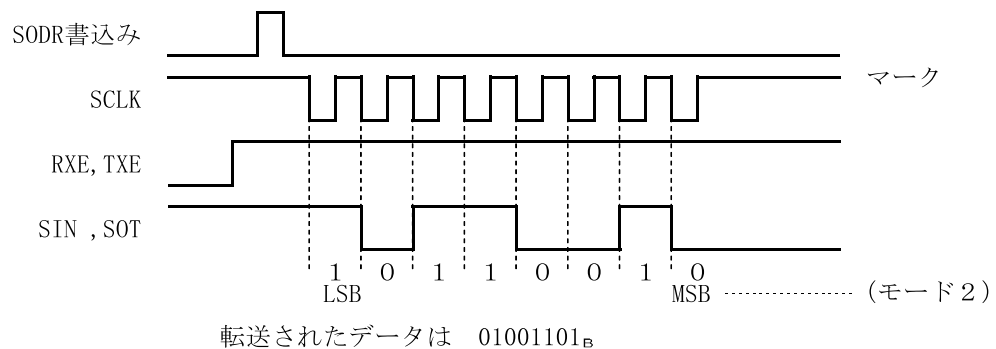


図 20.4-4 転送データフォーマット (モード2)

内部クロック (専用ボーレートジェネレータまたは内部リロードタイマ) を選択した場合は、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た場合は、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 (TDREフラグが"0") ことを確かめた後、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、必ずマークレベルにしてください。

データは8ビットデータのみ取り扱うことができ、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同期モード使用時のレジスタの設定値

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 20.4-5 CLK同期モードを使用する場合の、各制御レジスタの設定値

レジスタ名	ビット名	設定
SMR レジスタ	MD1,MD0	"10"
	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	無効
	CL	"1"
	REC	"0" (初期化)
	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"

## 通信開始

SODRレジスタへの書き込みによって通信を開始します。受信だけを実行しなければならぬ場合でも、SODRレジスタに仮の送信データを書き込む必要があります。

## 通信終了

SSRレジスタのRDRFフラグに"1"が書き込まれているかどうかにより、通信の終了を調べることができます。SSRレジスタのOREビットを使用して、通信が正常に行われたかを判断します。

## 20.4.4 割り込み発生およびフラグのセットタイミング

UARTには、5つのフラグと2つの割り込み要因を持ちます。  
5つのフラグとは、PE/ORE/FRE/RDRF/TDREで、2つの割り込み要因は、受信用のものと送信用のものです。

5つのフラグ（PE/ORE/FRE/RDRF/TDRE）と2つの割り込み要因

PE（パリティエラー）/ORE（オーバランエラー）/FRE（フレーミングエラー）  
PE/ORE/FREフラグがセットされるのは、送信中に該当するエラーが生じた場合で、クリアされるのは、SCRレジスタのRECビットに"0"を書込んだ場合です。

RDRF  
RDRFフラグは、受信データがSIDRレジスタ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を使用できません。また、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を使用できません。

TDRE  
TDREフラグは、SODRレジスタが空になり書き込み可能な状態になるとセットされ、SODRレジスタへ書き込むとクリアされます。

2つの割り込み要因は、受信用のものと送信用のものがあります。受信時は、PE/ORE/FRE/RDRFにより割り込みを要求します。送信時は、TDREにより割り込みを要求します。

各動作モードによる割り込みフラグのセットタイミング

モード0の受信動作時  
PE,ORE,FRE,RDRFは受信転送が終了し、最後のストップビットを検出する時にフラグがセットされ、CPUへの割り込み要求が発生します。PE,ORE,FREがアクティブ時は、SIDRのデータは無効データとなります。

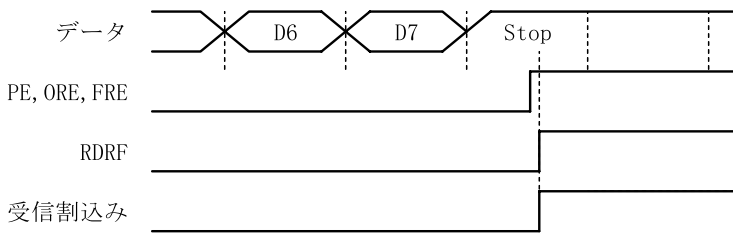


図 20.4-5 PE,ORE,FRE,RDRF のセットタイミング（モード0）

モード1の受信動作時  
受信動作が完了して最後のストップビットが検出されると、ORE,FRE,RDRFの各フラグがセットされます。次に、CPUに対して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なので、アドレスとデータを表す9番目のビットは無効となります。OREフラグとFREフラグがアクティブであれば、SIDRレジスタのデータは無効となります。

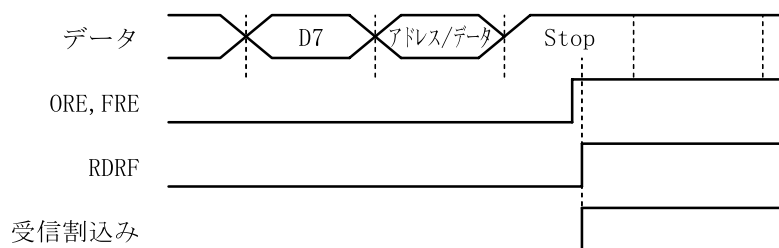


図 20.4-6 ORE, FRE, RDRF のセットタイミング (モード1)

## モード2の受信動作時

受信動作が完了して最後のデータ(D7)が検出されると、OREフラグと、RDRFフラグがセットされます。次に、CPUへの割込み要求が発生します。OREフラグがアクティブであれば、SIDRレジスタのデータは無効となります。

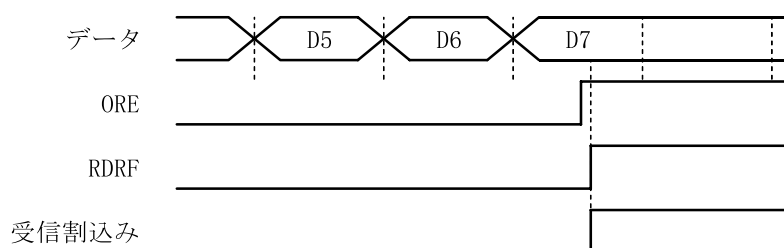


図 20.4-7 ORE, RDRF のセットタイミング (モード2)

## モード0, モード1, モード2の送信動作時

TDREがクリアされるのは、SODRレジスタの中にデータが書込まれた場合です。データが、内部のシフトレジスタに転送され、次のデータを書込むことができる場合、TDREがセットされ、CPUに対して割込み要求が発生します。送信動作中にSCRレジスタのTXE (モード2の時はRXEも含む) に"0"を書込むと、SSRレジスタのTDREに"1"がセットされます。次に、送信用のシフトレジスタが停止してからUARTによる送信動作が禁止されます。送信動作中にSCRレジスタのTXEに (モード2の時はRXEも含む) に"0"を書込んだ後、送信動作が停止する前にSODRレジスタへ書込まれたデータは送信されます。

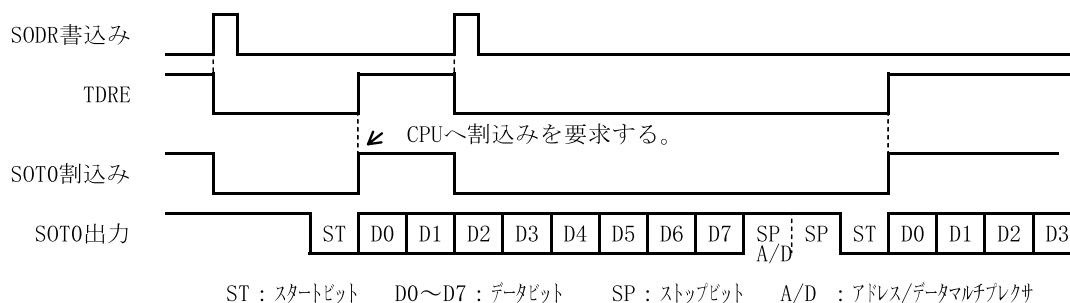


図 20.4-8 TDREをセットするタイミング (モード0, 1)

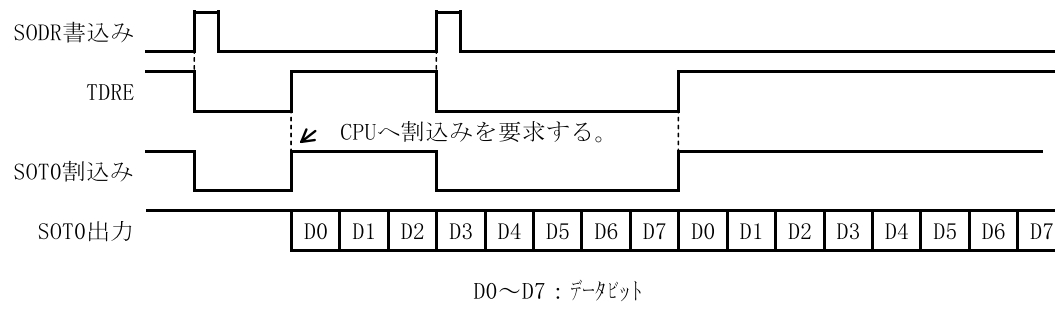


図 20.4-9 TDRE のセットタイミング（モード2）



## 20.5 UART(モード1使用時)の応用例

モード1は、1台のホストCPUに数台のスレーブCPUが接続された場合に使用されます(図 20.5-1を参照)。UART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。

UART(モード1使用時)の応用例

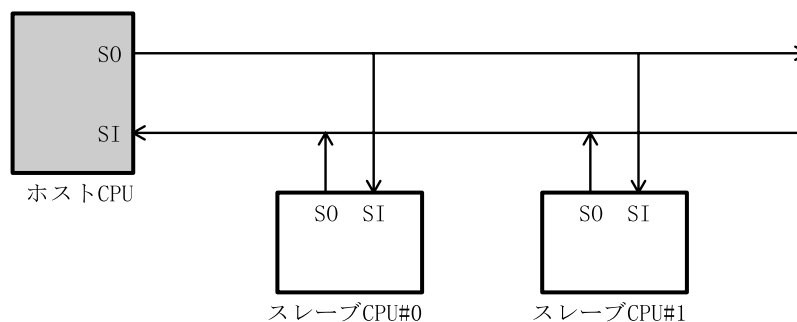


図 20.5-1 モード1を使用時のシステム構築例

図 20.5-2に示すように、通信は、ホストCPUがアドレスデータを転送することによって開始されます。アドレスデータとは、SCRレジスタのA/Dに"1"がセットされた時のデータです。アドレスデータは、通信先となるスレーブCPUが選択され、ホストCPUとの通信が可能になります。通常データは、SCRレジスタのA/Dに"0"がセットされた時のデータです。

モード1では、パリティチェック機能は使用できませんので、SCRレジスタのPENビットへは"0"をセットしてください。

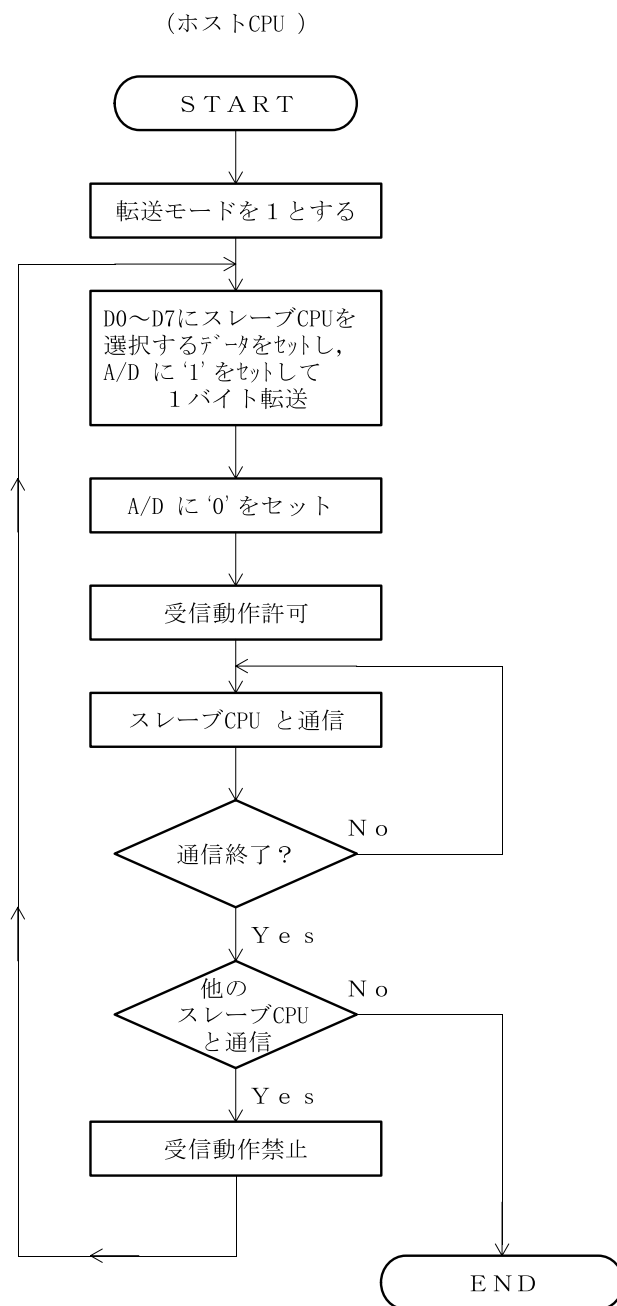


図 20.5-2 モード1使用時の通信フローチャート



## 第21章 IEBus™コントローラ

---

この章では、IEBus™コントローラの機能と動作について、説明します。  
なお、MB90587C/CAには、IEBus™コントローラの機能はありません。

---

- 21.1 IEBus™コントローラの概要
- 21.2 IEBus™コントローラのブロックダイアグラム
- 21.3 IEBus™コントローラのレジスタ
- 21.4 IEBus™の送信制御
- 21.5 IEBus™の受信制御
- 21.6 通信制御のステータス
- 21.7 IEBus™コントローラのメイン/割り込み処理フロー例
- 21.8 IEBus™コントローラの送信時の動作
- 21.9 IEBus™プロトコル動作
- 21.10 伝送プロトコル
- 21.11 伝送データ
- 21.12 ビットフォーマット

## 21.1 IEBus™コントローラの概要

IEBus™ (Inter Equipment Bus)は、装置と装置との間のデータ伝送を行うことを目的とした、小規模の2線式シリアル・バス・インタフェースです。

応用分野として、車載機器コントロール用バス・インタフェースなどがあります。

### IEBus™コントローラの特長

マルチ・マスタ方式

IEBus™に接続している全てのユニットが他のユニットへデータの伝送が実現可能。

同報通信機能(1ユニット対複数ユニットの通信)

グループ同報通信 : グループ・ユニットに対しての同報通信

一斉同報通信 : 全てのユニットに対しての同報通信

伝送速度の異なる3種類のモードが選択可能。

	IEBus™内部周波数	
	6MHz時	6.29MHz時
モード0	約3.9Kbps	約4.1Kbps
モード1	約17Kbps	約18Kbps
モード2	約26Kbps	約27Kbps

送信用データバッファ

8バイトFIFO

受信用データバッファ

8バイトFIFO

CPU内部動作周波数(12MHz, 12.58MHz)

周波数精度

モード0,1使用時 :  $\pm 1.5\%$

モード2使用時 :  $\pm 0.5\%$

## 21.2 IEBus™コントローラのブロックダイアグラム

図 21.2-1 IEBus™コントローラのブロックダイアグラムを示します。

IEBus™コントローラのブロックダイアグラム

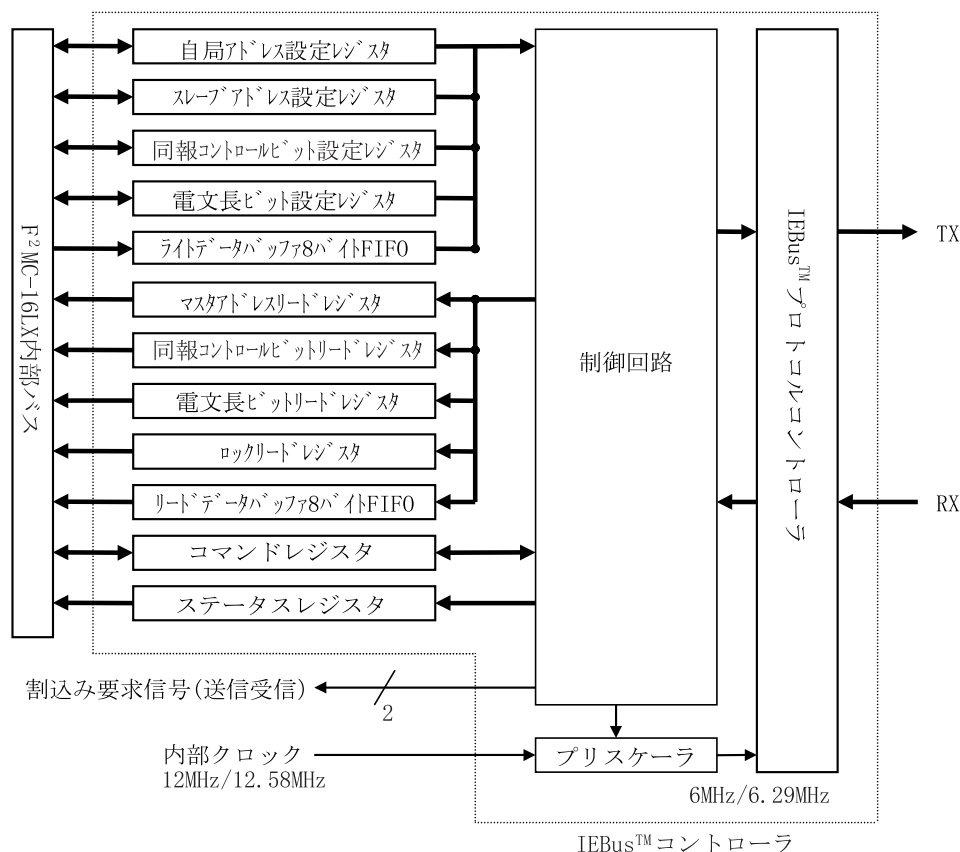


図 21.2-2 IEBus™コントローラのブロックダイアグラム

IEBus™コントローラの制御回路では、つぎのような制御を行います。

- 送信，受信データ・バイト数の制御
- 最大伝送バイト数の制御
- アービトレーション結果の検出
- 各フィールドのアクノリッジ返送判断
- 割込み信号の生成

## 21.3 IEBus™コントローラのレジスタ

IEBus™コントローラには、次の18のレジスタがあります。

- ・ 自局アドレス設定レジスタH/L
- ・ スレーブアドレス設定レジスタH/L
- ・ 同胞コントロールビット設定レジスタ
- ・ 同報コントロールビットリードレジスタ
- ・ 電文長ビット設定レジスタ
- ・ 電文長ビットリードレジスタ
- ・ コマンドレジスタH/L
- ・ ステータスレジスタH/L
- ・ ロックリードレジスタH/L
- ・ マスタアドレスリードレジスタH/L
- ・ リードデータバッファ
- ・ ライトデータバッファ

### IEBus™コントローラのレジスタ

BIT	15	14	13	12	11	10	9	8	
アドレス：000071H	予約	予約	予約	予約	MA11	MA10	MA09	MA08	自局アドレス設定レジスタH
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	MAWH
初期値	X	X	X	X	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：000070H	MA07	MA06	MA05	MA04	MA03	MA02	MA01	MA00	自局アドレス設定レジスタL
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	MAWL
初期値	X	X	X	X	X	X	X	X	
BIT	15	14	13	12	11	10	9	8	
アドレス：000073H	予約	予約	予約	予約	SA11	SA10	SA09	SA08	スレーブアドレス設定レジスタH
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	SAWH
初期値	X	X	X	X	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：000072H	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	スレーブアドレス設定レジスタL
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	SAWL
初期値	X	X	X	X	X	X	X	X	
BIT	15	14	13	12	11	10	9	8	
アドレス：000075H	D03	D02	D01	D00	C3	C2	C1	C0	同報コントロールビット設定レジスタ
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	DCWR
初期値	0	0	0	0	0	0	0	0	
BIT	15	14	13	12	11	10	9	8	
アドレス：00007FH	D03	D02	D01	D00	C3	C2	C1	C0	同報コントロールビットリードレジスタ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	DCRR
初期値	0	0	0	X	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：000074H	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0	電文長ビット設定レジスタ
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	DEWR
初期値	0	0	0	0	0	0	0	0	
BIT	7	6	5	4	3	2	1	0	
アドレス：00007EH	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0	電文長ビットリードレジスタ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	DERR
初期値	X	X	X	X	X	X	X	X	

BIT	15	14	13	12	11	10	9	8	
アドレス：000077H	MD1	MD0	PCOM	RIE	TIE	GOTM	GOTS	予約	コメントレジスタH
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	CMRH
初期値	0	0	0	0	0	0	0	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：000076H	RXS	TXS	TIT1	TIT0	CS1	CS0	RDBC	WDBC	コメントレジスタL
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	CMRL
初期値	1	1	0	0	0	0	0	0	
BIT	15	14	13	12	11	10	9	8	
アドレス：000079H	COM	TE	PEF	ACK	RIF	TIF	TSL	EOD	ステータスレジスタH
リード/ライト	(R)	(R/W)	(R)	(R)	(R/W)	(R/W)	(R)	(R)	STRH
初期値	0	0	X	X	0	0	0	0	
BIT	7	6	5	4	3	2	1	0	
アドレス：000078H	WDBF	RDBF	WDBE	RDBE	ST3	ST2	ST1	ST0	ステータスレジスタL
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	STRL
初期値	0	0	1	1	X	X	X	X	
BIT	15	14	13	12	11	10	9	8	
アドレス：00007BH	予約	予約	予約	LOC	LD11	LD10	LD09	LD08	ロックリードレジスタH
リード/ライト	(R)	(R)	(R)	(R/W)	(R)	(R)	(R)	(R)	LRRH
初期値	1	1	1	0	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：00007AH	LD07	LD06	LD05	LD04	LD03	LD02	LD01	LD00	ロックリードレジスタL
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	LRRL
初期値	X	X	X	X	X	X	X	X	
BIT	15	14	13	12	11	10	9	8	
アドレス：00007DH	予約	予約	予約	予約	MA11	MA10	MA09	MA08	マスタアドレスリードレジスタH
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	MARH
初期値	1	1	1	1	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：00007CH	MA07	MA06	MA05	MA04	MA03	MA02	MA01	MA00	マスタアドレスリードレジスタL
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	MARL
初期値	X	X	X	X	X	X	X	X	
BIT	15	14	13	12	11	10	9	8	
アドレス：000081H	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RDO	リードデータバッファ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	RDB
初期値	X	X	X	X	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：000080H	WD7	WD6	WD5	WD4	WD3	WD2	WD1	WDO	ライトデータバッファ
リード/ライト	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	WDB
初期値	X	X	X	X	X	X	X	X	



### 21.3.1 自局アドレス設定レジスタ(MAWH,MAWL)

自局のアドレス(12ビット)を設定するレジスタです。本レジスタは通信禁止解除前に設定してください。

自局アドレス設定レジスタ(MAWH,MAWL)

図 21.3-1に、自局アドレス設定レジスタ(MAWH,MAWL)の構成を示します。  
MAWH,MAWLに設定したアドレスは、マスタ時はマスタ・アドレスとして、スレーブ時は受信したスレーブ・アドレスとの比較に使用されます。  
ビット15-12は予約ビットです。必ず "1" をライトしてください。また、読み出し値は不定です。

BIT	15	14	13	12	11	10	9	8	
アドレス : 000071H	予約	予約	予約	予約	MA11	MA10	MA09	MA08	自局アドレス設定レジスタH
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	MAWH
初期値	X	X	X	X	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス : 000070H	MA07	MA06	MA05	MA04	MA03	MA02	MA01	MA00	自局アドレス設定レジスタL
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	MAWL
初期値	X	X	X	X	X	X	X	X	

図 21.3-1 自局アドレス設定レジスタ(MAWH,MAWL)

21.3.2      スレーブアドレス設定レジスタ(SAWH,SAWL)

スレーブアドレス設定レジスタ(SAWH,SAWL)は、マスタ時、スレーブアドレス(12ビット)を設定するレジスタです。本レジスタは通信禁止解除前に設定してください。

スレーブアドレス設定レジスタ(SAWH,SAWL)

図 21.3-2に、スレーブアドレス設定レジスタ(SAWH,SAWL)の構成を示します。  
ビット15-12は予約ビットです。必ず"1"をライトして下さい。また、読み出し値は不定です。

	BIT	15	14	13	12	11	10	9	8	
アドレス:	000073H	予約	予約	予約	予約	SA11	SA10	SA09	SA08	スレーブアドレス設定レジスタH
リード/ライト		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	SAWH
初期値		X	X	X	X	X	X	X	X	
	BIT	7	6	5	4	3	2	1	0	
アドレス:	000072H	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	スレーブアドレス設定レジスタL
リード/ライト		(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	SAWL
初期値		X	X	X	X	X	X	X	X	

図 21.3-2    スレーブアドレス設定レジスタ(SAWH,SAWL)

### 21.3.3 同報コントロールビット設定レジスタ(DCWR)

同報コントロールビット設定レジスタ(DCWR)では，同報通信，通常通信の選択とデータの読み込みやロックなどの制御を行います。

同報コントロールビット設定レジスタ(DCWR)

BIT	15	14	13	12	11	10	9	8	
アドレス： 000075H	D03	D02	D01	D00	C3	C2	C1	C0	同報コントロールビット設定レジスタ
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	DCWR
初期値	0	0	0	0	0	0	0	0	

図 21.3-3 同報コントロールビット設定レジスタ(DCWR)

【ビット15,14,13,12】D03,D02,D01,D00

D03,D02,D01,D00ビットは，同報通信，通常通信を選択します。

- 同報通信時：0H(0000B)
- 通常通信時：8H(1000B)

ビット14,13,12は，必ず "0" をライトして下さい。また，リード時は必ず"0" が読み出されます。

【ビット11,10,9,8】C3,C2,C1,C0

C3,C2,C1,C0ビットは，コントロールビットの4ビットを設定します。

表 21.3-1 コントロール・ビットの設定

	ビット3 <sup>*1</sup>	ビット2	ビット1	ビット0	機能 <sup>*2</sup>
0H	0	0	0	0	スレーブ・ステータスの読み込み
1H	0	0	0	1	未定義
2H	0	0	1	0	未定義
3H	0	0	1	1	データの読み込みとロック
4H	0	1	0	0	ロック・アドレスの読み込み（下位8ビット）
5H	0	1	0	1	ロック・アドレスの読み込み（上位4ビット）
6H	0	1	1	0	スレーブ・ステータスの読み込みとロック解除
7H	0	1	1	1	データ読み込み
8H	1	0	0	0	未定義
9H	1	0	0	1	未定義
AH	1	0	1	0	コマンドの書き込みとロック
BH	1	0	1	1	データ書き込みとロック
CH	1	1	0	0	未定義
DH	1	1	0	1	未定義
EH	1	1	1	0	コマンド書き込み
FH	1	1	1	1	データ書き込み

\*1：ビット3(MSB)の値により，以後の電文長フィールドの電文長ビットおよびデータ・フィールドのデータの転送方向が変わります。

ビット3が "1" の場合：マスタユニットからスレーブユニットへ転送

ビット3が "0" の場合：スレーブユニットからマスタユニットへ転送

\*2：3H,6H,AH,BHは，ロックの設定，および解除を設定するコントロール・ビットです。1H,2H,8H,9H,CH,DHの未定義値が送信された場合はアクノリッジは返しません。

21.3.4 電文長ビット設定レジスタ(DEWR)

電文長ビット設定レジスタ(DEWR)は、送信データバイト数(8ビット)を設定します。  
本レジスタの設定は送信時のみ有効です。

電文長ビット設定レジスタ (DEWR)

BIT	7	6	5	4	3	2	1	0	
アドレス : 000074H	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0	電文長ビット設定レジスタ
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	DEWR
初期値	0	0	0	0	0	0	0	0	

図 21.3-4 電文長ビット設定レジスタ (DEWR)

表 21.3-2に、送信データバイト数の設定値を示します。

表 21.3-2 送信データバイト数の設定値

送信データバイト数	コマンドパラメータ
1バイト	1H
2バイト	2H
：	：
255バイト	FFH
256バイト	00H

スレープ時のコマンドパラメータ受信

スレープ時にコントロールフィールドで、次のコマンドパラメータを受信した場合は、送信データ・バイト数は必ず1バイトに設定してください。

- 0H(スレープ・ステータスの読み込み)
- 4H(下位8ビットのロック・アドレス読み込み)
- 5H(上位4ビットのロック・アドレスの読み込み)
- 6H(スレープ・ステータスの読み込みとロックの解除)

最大伝送バイト数以上の設定をした場合

最大伝送バイト数以上の設定をすると複数フレームにわたる通信となります。  
このとき2回目以降は残りの通信データのバイト数を設定してください。

残りのデータバイト数は、電文長ビットリードレジスタ(DERR)の値を設定します。

### 21.3.5 コマンドレジスタ上位8ビット(CMRH)

コマンドレジスタ上位8ビット(CMRH)では，通信モードの設定や，送受信および割り込みを制御します。

コマンドレジスタ上位8ビット(CMRH)

BIT	15	14	13	12	11	10	9	8	
アドレス：000077H	MD1	MD0	PCOM	RIE	TIE	GOTM	GOTS	予約	コマンドレジスタH
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	CMRH
初期値	0	0	0	0	0	0	0	X	

図 21.3-5 コマンドレジスタ上位8ビット(CMRH)

【ビット15，14】MD1,MD0

MD1,MD0ビットは，通信モードを設定します。

表 21.3-3 通信モードの設定

MD1	MD0	動作モード
0	0	モード0
0	1	モード1
1	0	モード2
1	1	設定禁止

【ビット13】PCOM

PCOMビットは，通信許可信号です。"1"をライトすると，ステータスレジスタのCOMフラグが"1"になり，通信が可能になります。"0"をライトすると，通信を終了します。本ビットはステータス・レジスタのCOMフラグが"0"のとき"1"をライトしてください。

【ビット12】RIE

RIEビットは，受信割り込みを許可します。

- 0：受信割り込み禁止
- 1：受信割り込み許可

受信割り込みが発生するのは以下の通りです。

- データ受信でデータがRDBの8バイトいっぱいになった時
- データ受信が正常に終了した時
- 1通信フレームが，電文長ビットで指定されたデータ・バイト数分の受信を完了せずに終了した時
- アービトレーションに負け，さらにマスタ・ユニットからスレーブ・ユニットに選ばれなかった時。

【ビット11】TIE

TIEビットは，送信割り込みを許可します。

- 0：送信割り込み禁止
- 1：送信割り込み許可

送信割り込みが発生するのは以下の通りです。

- マスタ送信時，マスタ・アドレス・フィールドを終了し，マスタ・ユニットに勝ち残った時。
- スレーブ送信時，マスタ・ユニットよりデータの送信を要求するコントロール・ビットを受信した時。
- WDBの書き込み可能領域がコマンド・レジスタのTIT1,TIT0の2ビットで設定したバイト数になった時。
- 電文長で指定したデータ・バイト数分の通信を1通信フレーム内に終了した時。
- 1通信フレームで，電文長ビットで指定されたデータ・バイト数分の送信を完了せずに終了した時。

#### 【ビット10】GOTM

GOTMは，マスタの送信ビットです。通信禁止解除後，本ビットに "1" をセットすると，送信を開始します。

本ビットは "1" ライトのみ有効です。"0" ライトは意味を持ちません。本ビットをリードすると必ず"0"が読み出せます。

#### 【ビット9】GOTS

GOTSは，スレーブの送信ビットです。スレーブとなった時，データを送信する必要がある場合，通信禁止解除時に本ビットに"1"をライトしてください。

本ビットは"1"ライトのみ有効です。"0"ライトは意味を持ちません。本ビットをリードすると必ず"0"が読み出せます。

GOTMとGOTSの設定は以下のような関係があります。

表 21.3-4 GOTMとGOTSの設定

GOTM	GOTS	アビトレーション	スレーブ送信	動作
0	0	なし	不可	スレーブ受信のみ可能
0	1	なし	可	スレーブ送信可能。マスタにはなれない。
1	0	あり	不可	アビトレーションに負けてスレーブになったら受信のみ可能。
1	1	あり	可	アビトレーションに負けてスレーブになっても送信可能。

#### 【ビット8】予約

ビット8は，予約ビットです。必ず"1"を書き込んでください。また，読み出し値は不定です。

## 21.3.6 コマンドレジスタ下位8ビット(CMRL)

コマンドレジスタ下位8ビット(CMRL)では，正論理，負論理極性を設定し，また，割り込みや，内部周波数を設定します。

コマンドレジスタ下位8ビット(CMRL)

BIT	7	6	5	4	3	2	1	0	
アドレス: 000076H	RXS	TXS	TIT1	TIT0	CS1	CS0	RDBC	WDBC	コマンドレジスタ CMRL
リード/ライト	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
初期値	1	1	0	0	0	0	0	0	

図 21.3-6 コマンドレジスタ下位8ビット(CMRL)

### 【ビット7】RXS

RXSビットは，RX1端子入力の極性を設定します。外部に接続するドライバ/レシーバの仕様により選択してください。

表 21.3-5 RX1端子の極性の選択

RXS	RX1の出力
0	RX1端子正論理入力。 ・ロジック"1":ハイ・レベル ・ロジック"0":ロー・レベル
1	RX1端子負論理入力。 ・ロジック"1":ロー・レベル ・ロジック"0":ハイ・レベル

### 【ビット6】TXS

TX1端子出力の極性を設定します。外部に接続するドライバ/レシーバの仕様により選択してください。

表 21.3-6 TX1端子の極性の選択

TXS	RX1 の 出 力
0	TX1端子正論理出力。 ・ロジック"1":ハイレベル ・ロジック"0":ローレベル
1	TX1端子負論理出力 ・ロジック"1":ローレベル ・ロジック"0":ハイレベル

(注) MB90580Cシリーズではリセット時，TX1端子は"L"を出力します。使用するドライバ/レシーバが正論理("L"でイネーブルのドライバ/レシーバ)の場合は，そのまま接続するとリセット直後から本ビット設定まで"L"を出力するため他の通信ユニット間で通信が行われていた場合，それらに通信エラーが発生してしまいます。したがって，その場合はリセット直後から本ビット設定までドライバ/レシーバに"H"を入力するように外付け回路に工夫が必要になります。

### 【ビット5，4】TIT1,TIT0

TIT1,TIT0ビットは，送信データをライト・データ・バッファにライトしたときに発生する割り込みのタイミングを設定します。

表 21.3-7 データ送信割込みタイミングの設定

TIT1	TIT0	割込み発生タイミング
0	0	1バイト以上の書き込み可能領域がある場合
0	1	2バイト以上の書き込み可能領域がある場合
1	0	4バイト以上の書き込み可能領域がある場合
1	1	8バイトの書き込み可能領域がある場合

## 【ビット3,2】CS1,CS0

CPUの内部動作周波数とIEBus™の内部周波数の設定は、CS1,CS0の2ビットにより表4-8のようになります。必ずCS1,CS0ともに"0"を設定してください。

表 21.3-8 CS1,CS0の設定

CS1	CS0	CPU内部周波数 $\phi$	IEBus™内部周波数 $\phi_{IE}$	計算式
0	0	12MHz (12.58MHz)	6MHz (6.29MHz)	$\phi_{IE} = \phi / 2$
0	1	設定禁止	—————	—————
1	0	設定禁止	—————	—————
1	1	設定禁止	—————	—————

注)内部動作周波数と、IEBus™内部周波数の設定は上表の計算式より計算してください。

ただし、CPU内部周波数は動作保証範囲を越えないようにしてください。

周波数精度は、モード0および1使用時は±1.5%、モード2使用時は±0.5%以内としてください。

## 【ビット1】RDBC

"1"をライトすると、RDBの8バイトをクリアし、RDBを空の状態(RDBE=1)にします。本ビットをリードすると必ず "0" が読み出せます。

## 【ビット0】WDBC

"1"をライトすると、WDBの8バイトをクリアし、WDBを空の状態(WDBE=1)にします。本ビットをリードすると必ず"0"が読み出せます。

最大伝送バイト数を越える通信バイト数の通信では、本ビットに"1"をライトした場合、前のフレームでライトしたデータは無効となり、現在のフレームでWDBにライトしたデータから送信します。また、"0"をライトした場合は、前のフレームでライトしたデータが全て送信していなければ、その続きのデータから送信します。

送信中タイミング・エラーなどで途中終了した場合、本ビットに"0"をライトしてもエラーの発生したとき送信していたデータから送信せず、次のデータから送信します。



### 21.3.7 ステータスレジスタ上位8ビット(STRH)

ステータスレジスタ上位8ビット(STRH)では、通信状態、エラー発生の有無、割り込み要求の有無を示します。

ステータスレジスタ上位8ビット(STRH)

BIT	15	14	13	12	11	10	9	8	
アドレス: 000079H	COM	TE	PEF	ACK	RIF	TIF	TSL	EOD	ステータスレジスタH
リード/ライト	(R)	(R/W)	(R)	(R)	(R/W)	(R/W)	(R)	(R)	STRH
初期値	0	0	X	X	0	0	0	0	

図 21.3-7 ステータスレジスタ上位8ビット(STRH)

#### 【ビット15】COM

通信状態であることを表すフラグです。

- 0: 通信禁止
- 1: 通信許可

このフラグが "0" のときに、通信許可ビットのPCOMに"1"をライトするとセットされます。通信が終了するとクリアされます。

#### 【ビット14】TE

通信中にタイミングエラーが発生するとセットされ、"0"ライトによってクリアされます。このビットは"0"書き込みのみ有効です。

#### 【ビット13】PEF

パリティエラーを検出するとセットされます。受信側でこのビットがセットされると、アクノリッジを返送しません。通信禁止解除でクリアされます。

- 0: パリティ・エラーなし
- 1: パリティ・エラーあり

#### 【ビット12】ACK

個別通信時、アクノリッジ・ビットで送信、または受信したデータがセットされます。通信禁止解除でクリアされます。

- 0: アクノリッジ・ビットが"0"
- 1: アクノリッジ・ビットが"1"

同報通信では意味がありません。リード値は不定となります。

#### 【ビット11】RIF

受信割込みが発生したとき"1"にセットされます。

- 0: 受信割込み要求なし
- 1: 受信割込み要求あり

本ビットは、"0"ライト、またはインテリジェントI/Oサービスによってクリアされます。このビットは"0"書き込みのみ有効です。

#### 【ビット10】TIF

送信割込みが発生したとき"1"にセットされます。

- 0: 送信割込み要求なし
- 1: 送信割込み要求あり

本ビットは, "0"ライト, またはインテリジェントI/Oサービスによってクリアされます。このビットは"0"書き込みのみ有効です。

**【ビット9】TSL**

データフィールドにて, データの送受信が最大伝送バイト数となって通信フレームが終了したとき, "1"がセットされます。次の通信フレームが開始されると, クリアされます。

**【ビット8】EOD**

データ・フィールドにて, データの送受信が電文長ビットで設定したバイト数となって通信フレームが終了したとき, "1" がセットされます。本ビットが "1" であればその通信フレームは正常に終了したことを表します。

次の通信フレームが開始されると, クリアされます。

## 21.3.8 ステータスレジスタ下位8ビット(STRL)

ステータスレジスタ下位8ビット(STRL)は、ライトデータバッファ(WDB)、リードデータバッファ(RDB)の状態を表します。

また、次の4つの通信状態を読み出すことができます。

- ・マスタ、スレーブ送信時
- ・マスタ受信時
- ・スレーブ受信時
- ・同報受信時

ステータスレジスタ下位8ビット(STRL)

BIT	7	6	5	4	3	2	1	0	
アドレス: 000078H	WDBF	RDBF	WDBE	RDBE	ST3	ST2	ST1	ST0	ステータスレジスタ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	STRL
初期値	0	0	1	1	X	X	X	X	

図 21.3-8 ステータスレジスタ下位8ビット(STRL)

### 【ビット7】WDBF

ライトデータバッファ(WDB)の状態を表すフラグです。

WDBがいっぱいになったときセットされ、データが送信され1バイトでも書き込み可能となればクリアされます。

- 0: WDBがいっぱいでない
- 1: WDBがいっぱい

### 【ビット6】RDBF

リードデータバッファ(RDB)の状態を表すフラグです。

RDBが受信データでいっぱいになったときにセットされ、データを読み出し、1バイトでも受信可能となったときクリアされます。

- 0: RDBがいっぱいでない
- 1: RDBがいっぱい

### 【ビット5】WDBE :

ライト・データ・バッファ(WDB)の状態を表すフラグです。

WDBが空になったときにセットされ、WDBにデータを1バイトでもライトすればクリアされます。また、コマンド・レジスタのWDBCビットに"1"をライトするとセットされます。

- 0: WDBにデータが入っている。
- 1: WDBが空

### 【ビット4】RDBE

リード・データ・バッファ(RDB)の状態を表すフラグです。

RDBを読み出し、空になったときにセットされます。データを受信し、RDBにデータが1バイトでも入ればクリアされます。また、コマンド・レジスタのRDBCビットに"1"をライトするとセットされます。

- 0: RDBにデータが入っている
- 1: RDBが空

## 【ビット3～0】ST3, ST2, ST1, ST0

送受信中，本4ビットに通信の状態を設定し，割込みが発生します。本ビットを読み出すことにより，通信状態を知ることができます。

表 21.3-9 ステータスフラグ

ST3	ST2	ST1	ST0	状態	ステート
0	0	0	0	マスタ，スレーブ送信時	送信開始
0	0	0	1		送信中
0	0	1	0		送信正常終了
0	0	1	1		送信途中終了
0	1	0	0	マスタ受信時	マスタ受信開始
0	1	0	1		マスタ受信データ・フル
0	1	1	0		マスタ受信途中終了
0	1	1	1		マスタ受信途中終了
1	0	0	0	スレーブ受信時	スレーブ受信開始
1	0	0	1		スレーブ受信データ・フル
1	0	1	0		スレーブ受信正常終了
1	0	1	1		スレーブ受信途中終了
1	1	0	0	同報受信時	同報受信開始
1	1	0	1		同報受信データ・フル
1	1	1	0		同報受信正常終了
1	1	1	1		同報受信途中終了

本ビットのセットされるタイミングなどは，「21.6 通信制御のステータス」を参照してください。

### 21.3.9 ロックリードレジスタ(LRRH, LRRL)

ロックリードレジスタ(LRRH, LRRL)では、ユニットからのロック状態を読み出します。

ロックリードレジスタ(LRRH, LRRL)

BIT	15	14	13	12	11	10	9	8	
アドレス：00007BH	予約	予約	予約	LOC	LD11	LD10	LD09	LD08	ロックリードレジスタH
リード/ライト	(R)	(R)	(R)	(R/W)	(R)	(R)	(R)	(R)	LRRH
初期値	1	1	1	0	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス：00007AH	LD07	LD06	LD05	LD04	LD03	LD02	LD01	LD00	ロックリードレジスタL
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	LRRL
初期値	X	X	X	X	X	X	X	X	

図 21.3-9 ロックリードレジスタ(LRRH, LRRL)

【ビット15～13】予約

ビット15～13は、予約ビットです。必ず"1"が読み出されます。

【ビット12】LOC

LOCビットは、自局が他のユニットからロックされているかいないか(ロック状態)を読み出します。

- 0：ロックされていない
- 1：ロックされている

"0"を書き込むと自局のロックを解除します。"1"書き込みは意味を持ちません。

【ビット11～0】LD11～LD00

ロックしているユニットのアドレスが読み出せます。自局がロックされていないときは本ビットの意味はありません。

< 注意事項 >

IEBus™では、複数のフレームにわたって通信ができるように、ロック機能が設けられています。ただし、ロックをかけたユニットがロックを解除せずにダウンした場合には、ロックをかけられたユニットはなにもデータを受信することができなくなってしまいます。このような状態を防止するため、ロック機能を使用するシステムにおいては定期的に本レジスタをリードし、ロック状態を監視する必要があります。  
自局のロックを解除するにはLOCビットに"0"をライトします。

21.3.10 マスタアドレスリードレジスタ(MARH,MARL)

マスタアドレスリードレジスタ(MARH,MARL)では , マスタユニットからのアドレスを読み出します。

マスタアドレスリードレジスタ(MARH,MARL)

BIT	15	14	13	12	11	10	9	8	
アドレス : 00007DH	予約	予約	予約	予約	MA11	MA10	MA09	MA08	マスタアドレスリードレジスタH
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	MARH
初期値	1	1	1	1	X	X	X	X	
BIT	7	6	5	4	3	2	1	0	
アドレス : 00007CH	MA07	MA06	MA05	MA04	MA03	MA02	MA01	MA00	マスタアドレスリードレジスタL
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	MARL
初期値	X	X	X	X	X	X	X	X	

図 21.3-10 マスタアドレスリードレジスタ(MARH,MARL)

【ビット15～12】予約

予約ビットです。必ず"1"が読み出されます。

【ビット11～0】MA11～00

スレーブ時 , マスタユニットのアドレスが読み出せます。自局がマスタのときは自局アドレス設定レジスタに設定したアドレスが設定され読み出されます。マスタアドレスフィールドでアービトレーションが終了した時点で本レジスタにデータが設定されます。

### 21.3.11 同報コントロールビットリードレジスタ(DCRR)

同報コントロールビットリードレジスタ(DCRR)は、マスタユニットから送られてきた同報ビットやコントロールビットを読み出せます。

同報コントロールビットリードレジスタ(DCRR)

BIT	15	14	13	12	11	10	9	8	
アドレス：00007FH	D03	D02	D01	D00	C3	C2	C1	C0	同報コントロールビットリードレジスタ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	DCRR
初期値	0	0	0	X	X	X	X	X	

図 21.3-11 同報コントロールビットリードレジスタ(DCRR)

【ビット15～12】D03～0

スレーブ時、マスタユニットから送られてきた同報ビットを読み出せます。自局がマスタのときは同報、コントロールビット設定レジスタの同報ビットに設定したデータを読み出せます。同報ビットを送信、または受信すると設定されます。

- 個別通信時：1H(0001B)
  - 同報通信時：0H(0000B)
- D03-1は必ず"0"が読み出されます。

【ビット11～8】C3～0

スレーブ時、マスタユニットから送られてきたコントロールビットを読み出せます。自局がマスタのときは同報、コントロールビット設定レジスタのコントロールビットに設定したデータを読み出せます。コントロールフィールドが終了し、アクノリッジを検出すると設定されます。

表4-1 コントロールビットの設定値参照。

21.3.12 電文長ビットリードレジスタ下位8ビット(DERR)

電文長ビットリードレジスタ下位8ビット(DERR)は、通信するデータのバイト数を表わします。自局が送信する時は電文長ビット設定レジスタに設定した値が、また、自局が受信の時は電文長フィールドで受信した値が読み出せます。

電文長ビットリードレジスタ下位8ビット(DERR)

BIT	7	6	5	4	3	2	1	0	
アドレス: 00007EH	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0	電文長ビットリードレジスタ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	DERR
初期値	X	X	X	X	X	X	X	X	

図 21.3-12 電文長ビットリードレジスタ下位8ビット(DERR)

電文長ビットリードレジスタ下位8ビット(DERR)に設定されるタイミングと値は、以下のようになります。

マスタ時

- DEWRに送信データバイト数を書き込んだとき。
- マスタ受信で電文長ビットを受信したとき
- 通信が終了したとき。

スレーブ時

- DEWRIに送信データバイト数を書き込んだとき。
- スレーブ受信で電文長ビットを受信したとき。
- 通信が終了したとき。

表 21.3-2 送信データバイト数の設定値を参照ください。



### 21.3.13 リードデータバッファ(RDB)

リードデータバッファ(RDB)は、データフィールドで受信したデータを格納する8バイトのFIFOバッファです。

リードデータバッファ(RDB)

図 21.3-13に、リードデータバッファ(RDB)のビット構成を示します。

BIT	15	14	13	12	11	10	9	8	
アドレス: 000081H	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	リードデータバッファ
リード/ライト	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	RDB
初期値	X	X	X	X	X	X	X	X	

図 21.3-13 リードデータバッファ(RDB)

リードデータバッファ(RDB)では、受信データで8バイトがいっぱいになると受信割込みが発生します。受信割込みが発生したら、データを読み出してください。次の受信データを受信するまでに読み出さない場合は、エラーになります。エラーになった場合は、同報通信のときは通信を終了し、個別通信のときにはアクノリッジを返送せず、最大伝送バイト数を越えるまで受信動作を続行します。RDBがいっぱいになって次のデータを受信するまでの時間は、表 21.3-5のようになります。

受信データがいっぱいにならなくても電文長ビットで指定したデータ数または、最大伝送バイト数データを受信すると受信割込みが発生します。受信割込みが発生したらデータを読み出してください。

CMRLレジスタのRDBCビットに "1" をライトすると、空の状態になります。必ずRDBが空でないことを確認してから本レジスタをリードしてください。

表 21.3-10 割込み発生から次のデータ受信するまでの時間

	最大時間[μs]	サイクル数
モード0	1580	19000
モード1	400	4800
モード2	290	3400

21.3.14 ライトデータバッファ(WDB)

ライトデータバッファ(WDB)は、データフィールドで送信するデータを設定する8バイトのFIFOバッファです。コマンドレジスタのTIT1, TIT0の2ビットで、データ送信割込みの発生タイミングを設定できます。

ライトデータバッファ(WDB)

図 21.3-14に、ライトデータバッファ(WDB)のレジスタ構成を示します。

BIT	7	6	5	4	3	2	1	0	
アドレス: 000080H	WD7	WD6	WD5	WD4	WD3	WD2	WD1	WD0	ライトデータバッファ
リード/ライト	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)	WDB
初期値	X	X	X	X	X	X	X	X	

図 21.3-14 ライトデータバッファ(WDB)

ライトデータバッファ(WDB)で、割込みが発生したら次に送信するデータをライトしてください。データが送信され、8バイトのFIFOが空になり送信するデータがなくなります。このとき、図 21.3-14の時間以内に、次に送信するデータを設定しないと、エラーとなり、通信を終了します。

CMRLレジスタのWDBCビットに "1" をライトすると、空の状態になります。また、いっぱいの状態では本レジスタにデータを書き込まないでください。必ずWDBがいっぱいでないことを確認してから本レジスタにライトしてください。

表 21.3-11 割込み発生から次のデータライトまでの時間

	最大時間[μs]	サイクル数
モード0	1580	19000
モード1	400	4800
モード2	290	3400

## 21.4 IEBus™の送信制御

IEBus™の送信には、次の3つがあります。

- ・ マスタ送信
- ・ スレーブ送信
- ・ スレーブステータスとロックアドレスの送信

### マスタ送信

スレーブユニットに対してデータ、コマンドを送信する場合の通信データのやりとりのことをマスタ送信といいます。

マスタユニットからスレーブユニットにデータを送信するためには、コントロールビット設定レジスタにAH, BH, EH, FHを設定します。

- 1) 自局アドレス設定レジスタにマスタアドレスを、スレーブアドレス設定レジスタにスレーブアドレスを、同報コントロールビット設定レジスタに同報ビットおよび、コントロールビットを設定し、さらに、コマンドレジスタを設定して通信禁止を解除します。
- 2) マスタユニットとしてアービトレーションに勝ち残ったところ(マスタ・アドレス・フィールドの終わり)で、送信開始のステート・コード(0H)がステータス・レジスタのST3-0の4ビットに設定され、送信割込みが発生します。このとき、電文長ビット設定レジスタに送信データ・バイト数を、WDBに送信データをWDBがいっぱいでない事を確認しながら設定します。
- 3) WDBのデータは、1バイト送信するごとに1つずつ減っていきます。書き込み可能領域がコマンドレジスタのTIT1, TIT0の2ビットで設定した割込み発生タイミングになると、送信割込みが発生します。このとき、データ送信のステートコード(1H)かつWDBがいっぱいでなければWDBに送信データを設定してください。
- 4) 送信データバイト数分のデータ、コマンドを正しく送信できた場合には、送信終了のステートコード(2H)がステータスレジスタのST3-0の4ビットに設定され、EODビットが"1"に設定されて送信割込みが発生します。
- 5) 最大伝送バイト数以上の送信あるいは送信の途中でエラーが発生して、電文長ビット設定レジスタで設定したバイト数送信できず途中で終了する場合には、送信途中終了のステートコード(3H)がステータスレジスタのST3-0の4ビットに設定され、送信割込みが発生します。このとき、ステータスレジスタのTSL, PEF, TEの各ビットの状態で通信エラーの内容を知ることができます。

送信中にタイミングエラーが発生してすべてのデータを送信できなかった場合も、TEビットをクリアし再度送信処理を実行することにより、WDBに残っているデータを送信することができます。エラーが発生した時は、送信していたデータの次のデータから送信します。

通信をあらたにやり直す場合は、コマンドレジスタのWDBCビットに"1"を書き込んで、WDBの内容をクリアします。

アビトレーションに負けたときは、スレーブ受信状態になり、スレーブアドレスフィールドを受信してスレーブ受信でなければ、スレーブもしくは同報受信途中終了(ステートコードB<sub>H</sub>もしくはF<sub>H</sub>)となり受信割り込みが発生します。

送信のリトライはハードウェアではサポートしていないため、プログラムでリトライの回数とその処理をおこなう必要があります。

## スレーブ送信

スレーブ送信は、スレーブユニットがマスタユニットからコントロールビットとして、3Hあるいは7Hを受信してデータを送信する場合の通信データのやりとりです。

- 1) マスタユニットより、コントロールビットを3H、または7Hを受信したときに、送信開始のステートコード(0H)がステータスレジスタのST3-0の4ビットに設定され、送信割込みが発生します。このとき、電文長ビット設定レジスタに送信データバイト数を、WDBがいっぱいでないことを確認しながら、WDBに送信データを設定します(以下の注意事項を参照してください)。
- 2) 電文長の送信開始時に、ステータスレジスタのST3-0に送信中を示すコード(1H)が設定され、送信割り込みが発生します。このとき、WDBがいっぱいでないことを確認しながら、WDBに送信データをライトします。
- 3) WDBのデータは、1バイト送信するごとに、1つつ減っていきます。書き込み可能領域がコマンドレジスタのTIT1、TIT0の2ビットで設定した割込み発生タイミングになると、送信割込みが発生します。このとき、データ送信のステートコード(1H)で、なおかつWDBがいっぱいでないことを確認しながら、WDBに送信データを設定してください。
- 4) 送信データバイト数分のデータ、コマンドの送信を正しく送信できた場合には、送信終了のステートコード(2H)がステータスレジスタのST3-0の4ビットに設定され、EODビットが "1" に設定されて送信割込みが発生します。
- 5) 最大伝送バイト数以上の送信あるいは送信の途中でエラーが発生すると、電文長ビット設定レジスタで設定したバイト数が送信できず、途中で終了する場合があります。この場合は、送信途中終了のステートコード(3H)がステータスレジスタのST3-0の4ビットに設定され、送信割込みが発生します。このとき、ステータスレジスタのTSL、PEF、TEの各ビットの状態で通信エラーの内容を知ることができます。

## マスタユニットによるコントロールビット受信に関する注意事項

送信バイト数、および送信データは、コントロールビットを受信後の割り込みで設定することもできます。

WDBが空の状態では、この割り込みによって初めて送信データを設定する場合は、次の電文長ビット送信の割り込みまでの間隔が短いので、以下の点に注意する必要があります。

- 必ずWDBがいっぱいでないことを確認して、データをライトしてください。
- 送信データバイト数をこの割り込みで設定する場合は、割り込み発生から表 21.4-1に示す時間内で設定してください。もし間に合わなければ、エラーは発生せず、先に設定してあった送信バイト数を送信します。もし初期値のままであれば、256バイト送信します。
- 送信データをこの割り込みまで設定していない場合、この割り込み発生から表 21.4-1に示す時間内で最低1バイトは設定してください。もし、設定しないと、WDBが空と判断し、電文長ビット送信後エラーとなって通信は終了します。

表 21.4-1 割込み発生から送信バイト数設定までの最大時間

	最大時間[μs]	サイクル数[サイクル]
モード0	約158	約1900
モード1	約40	約480
モード2	約29	約350

## スレーブステータスとロックアドレスの送信

マスタユニットよりコントロールビットとして0H, 4H, 5H, 6Hを受信した場合, 自動的にスレーブステータス, ロックアドレスを生成し, マスタユニットに送信します。そのため, スレーブステータス, ロックアドレスの送信はWDBに送信データを設定する必要はありません。ただし, 電文長ビット設定レジスタに設定する送信バイト数は1バイトに設定してください。

## 21.5 IEBus™の受信制御

---

IEBus™の受信には、次の3つがあります。

- ・ マスタ受信
  - ・ スレーブ受信
  - ・ 同報受信
- 

### マスタ受信

スレーブユニットよりデータ、スレーブステータス、ロックアドレスを受信する場合の通信データのやりとりのことを、マスタ受信といいます。通信ユニットをマスタユニットにするためには、コントロールビットとして0H, 3H, 4H, 5H, 6H, 7Hを設定します。

- 1) スレーブユニットは、コントロールビット受信後、電文長ビットを送信してきます。マスタユニットは電文長ビットを受信し、アクノリッジを返した時に電文長ビットリードレジスタに受信データバイト数が設定されます。
- 2) 電文長ビット受信後、アクノリッジを返すと、次にデータフィールドとなり、マスタユニットはデータを送信してきます。受信データは、1バイト受信するごとにRDBに設定されます。
- 3) 受信データを8バイト(RDBの容量)受信するごとに、マスタ受信バッファフルのステートコード(5H)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。割込みが発生したら、RDBが空でないことを確認しながらRDBを読み出します。
- 3) 1フレームの最終データをRDBに受信後、マスタ受信正常終了のステートコード(5H)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。この割込みはRDBがいっぱいにならなくても発生します。
- 4) 最大伝送バイト数受信、あるいは受信途中で通信エラーが発生し、スレーブ・ユニットより電文長フィールドで受信した伝送バイト数分のデータを受信できずに通信を終了する場合には、マスタ受信途中終了のステートコード(7H)がステータス・レジスタのST3-0の4ビットに設定され、割込みが発生します。

## スレーブ受信

スレーブユニットは、マスタユニットよりコントロールビットとしてAH, BH, EH, FHのいずれかを受信します。データ、コマンドの受信を行う場合、この通信データのやりとりのことをスレーブ受信といいます。

- 1) スレーブユニットが、電文長フィールドでアクノリッジ返送したときには、電文長ビットリードレジスタに受信データバイト数が設定されます。
- 2) 電文長ビット受信後、アクノリッジを返すとデータフィールドとなり、マスタユニットはデータを送信してきます。受信データは1バイト正常に受信するごとにRDBに設定されます。
- 3) 受信データを8バイト(RDBの容量)受信するごとに、スレーブ受信バッファフルのステートコード(9H)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。この割り込みが発生したら、RDBが空でないことを確認しながら、RDBを読み出します。
- 4) 1フレームの最終データをRDBに受信後、スレーブ受信正常終了のステートコード(AH)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。この割り込みは、RDBがいっぱいでなくても発生します。
- 5) 最大伝送バイト数受信、あるいは受信途中で通信エラーが発生し、スレーブユニットより電文長フィールドで受信した伝送バイト数分のデータを受信できずに通信を終了する場合があります。この場合は、マスタ受信途中終了のステートコード(BH)がステータスレジスタのST3-0の4ビットに設定され、割込みが発生します。

## 同報受信

- 1) スレーブユニットが電文長フィールドでアクノリッジ返送したときは、電文長ビットリードレジスタに受信データバイト数が設定されます。
- 2) 電文長ビット受信後、アクノリッジを返すとデータフィールドとなり、マスタユニットはデータを送信してきます。受信データは1バイト正常に受信するごとにRDBに設定されます。
- 3) 受信データを8バイト(RDBの容量)受信するごとに、スレーブ受信バッファフルのステートコード(EH)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。この割り込みが発生したらRDBが空でないことを確認しながらRDBを読み出します。
- 4) 1フレームの最終データをRDBに受信後、スレーブ受信正常終了のステートコード(DH)がステータスレジスタのST3-0の4ビットに設定され、受信割込みが発生します。この割り込みは、RDBがいっぱいでなくても発生します。
- 5) 最大伝送バイト数受信、あるいは受信途中で通信エラーが発生し、スレーブユニットより電文長フィールドで受信した伝送バイト数分のデータを受信できずに通信を終了する場合があります。この場合は、マスタ受信途中終了のステートコード(FH)がステータスレジスタのST3-0の4ビットに設定され、割込みが発生します。ステータスレジスタのST3-0については、「21.6 通信制御のステータス」を参照してください。

## 21.6 通信制御のステータス

ステータスレジスタには、ステータスコードを読めるST3-0の4ビットが存在します。ステータスコードが設定されると、割り込み要求を行います。割り込み期間中に本ビットを読み出し、通信状態を知ることができます。

マスタスレーブデータ送信時(送信割り込みが発生)

同報，またはマスタ・アドレス・フィールドでアービトレーションに勝ち残り，マスタユニットとなり，スレーブユニットにデータ，コマンドを送信する場合(同報通信も含む)や，スレーブユニットからマスタユニットにデータを送信する場合にST3-0に設定されるコードの内容を，表 21.6-1に示します。

表 21.6-1 マスタ・スレーブ送信時のST3-0のコード

コード名	コード	内 容
送信開始	0000 <sub>B</sub>	マスタ・スレーブ送信を開始することを表します。マスタ送信とスレーブ送信では，発生個所が異なります。 ・マスタ送信 マスタアドレスフィールドを終了し，マスタユニットに勝ち残った場合に設定されます。 ・スレーブ送信 マスタ・ユニットより，データの送信を要求するコントロールビット(0H, 3H, 4H, 5H, 6H, 7H)を受信した時に設定されます。
データ送信	0001 <sub>B</sub>	マスタユニットあるいはスレーブユニットが，データフィールドにてデータ送信中であることを表します。電文長ビットの送信開始時から設定されます。
送信正常終了	0010 <sub>B</sub>	電文長ビットで指定したデータ・バイト数分の送信を，1通信フレーム内に終了したことを表します。
送信途中終了	0011 <sub>B</sub>	1フレーム内で，電文長ビットで指定したデータ・バイト分送信を，完了せずに通信を終了したことを表します。

マスタ受信時(受信割り込みが発生)

同報，またはマスタアドレスフィールドでアービトレーションに勝ち残り，マスタユニットとなり，スレーブユニットより，データスレーブユニットより，データ，ステータス，ロック・アドレスを受信する場合に，ST3-0に設定されるコードの内容を，表 21.6-2に示します。

表 21.6-2 マスタ受信時のST3-0のコード

コード名	コード	内 容
マスタ受信開始	0100 <sub>B</sub>	マスタユニットが，スレーブユニットより電文長コードを正しく受信したときに，このコードが発生し，マスタ受信の開始を知らせます。
マスタ受信データ・フル	0101 <sub>B</sub>	マスタ受信データを8バイト(RDBの容量)受信することにRDBがいっぱいであれば，ホストコントローラに対し，RDBより受信データの読み出しを要求します。
マスタ受信正常終了	0110 <sub>B</sub>	電文長ビットで指定されたデータ・バイト数分の受信を1通信フレーム内で終了したことを表します。
マスタ受信途中終了	0111 <sub>B</sub>	1通信フレーム内で，電文長ビットで指定されたデータ・バイト数分の受信を完了せずに通信を終了したことを表します。



## スレーブ受信時(受信割込みが発生)

マスタユニットよりデータ,コマンドを受信する場合に,ST3-0に設定されるコードの内容を表 21.6-3に示します。

表 21.6-3 スレーブ受信時のST3-0のコード

コード名	コード	内 容
スレーブ受信 開始	1000 <sub>B</sub>	スレーブユニットが,マスタユニットより電文長コードを正しく受信したときに,このコードが発生し,スレーブ受信の開始を知らせます。
スレーブ受信 データ・フル	1001 <sub>B</sub>	マスタ受信データを8バイト(RDBの容量)受信するごとに,RDBがいっぱいであれば,ホストコントローラに対し,RDBより受信データの読み出しを要求します。
スレーブ受信 正常終了	1010 <sub>B</sub>	電文長ビットで指定されたデータ・バイト数分の受信を1通信フレーム内で終了したことを表します。
スレーブ受信 途中終了	1011 <sub>B</sub>	1通信フレーム内で,電文長ビットで指定されたデータ・バイト数分の受信を完了せずに通信を終了したことを表します。

## 同報受信時(受信割込みが発生)

マスタユニットより,同報通信でデータ,コマンドを受信する場合に,ST3-0に設定されるコードの内容を表 21.6-4に示します。

表 21.6-4 同報受信時のST3-0のコード

コード名	コード	内 容
同報受信開始	1100 <sub>B</sub>	スレーブユニットが,マスタユニットより電文長コードを正しく受信したときに,このコードが発生し,同報受信の開始を知らせます。
同報受信 データ・フル	1101 <sub>B</sub>	マスタ受信データを,8バイト(RDBの容量)受信するごとにRDBがいっぱいであれば,ホスト・コントローラに対し,RDBより受信データの読み出しを要求します。
同報受信 正常終了	1110 <sub>B</sub>	電文長ビットで指定されたデータ・バイト数分の受信を1通信フレーム内で終了したことを表します。
同報受信 途中終了	1111 <sub>B</sub>	1通信フレーム内で,電文長ビットで指定されたデータ・バイト数分の受信を完了せずに通信を終了したことを表します。

## 21.7 IEBus™コントローラのメイン/割り込み処理フロー例

MB90580Cシリーズにおける，IEBus™の処理フローの一例について説明します。

### メインルーチン

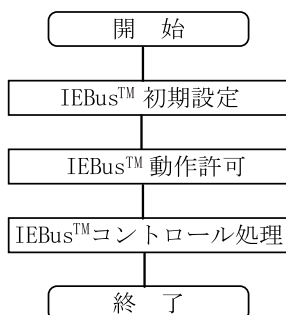
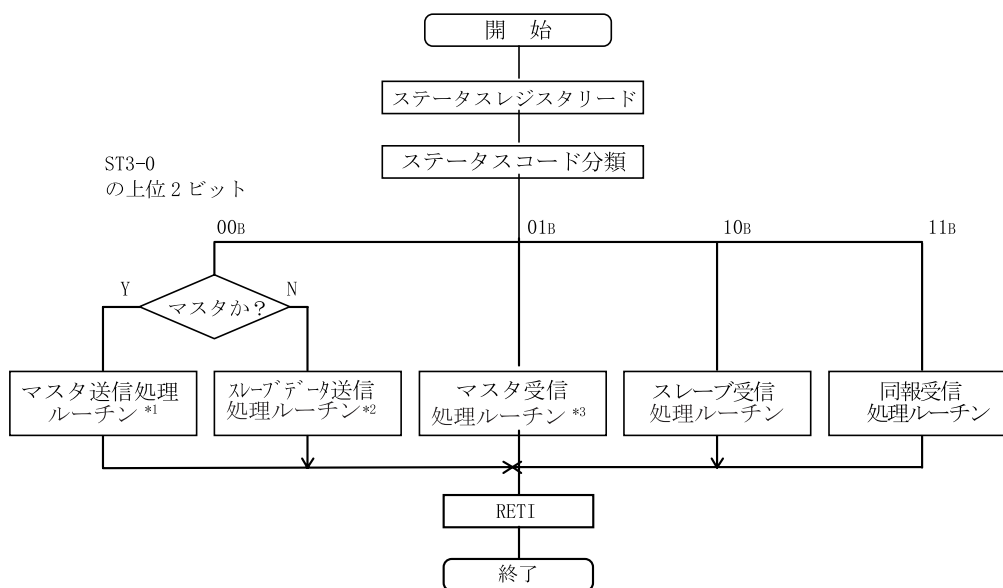


図 21.7-1 メインルーチン

### 割り込み処理ルーチン

図 21.7-2に示す例は，IEBus™にて，送信開始，あるいは受信完了割り込みが発生したときのルーチンです。このルーチンでは，ステータスレジスタのステータスコード(ST3-0)の内容を読み出して，送信データの書き込みおよび受信データの読み出しを行います。



\*1：「20.8 IEBus™ コントローラの各種処理フロー例」の「■マスタ送信処理ルーチン」参照

\*2：「20.8 IEBus™ コントローラの各種処理フロー例」の「■スレーブデータ送信処理ルーチン」参照

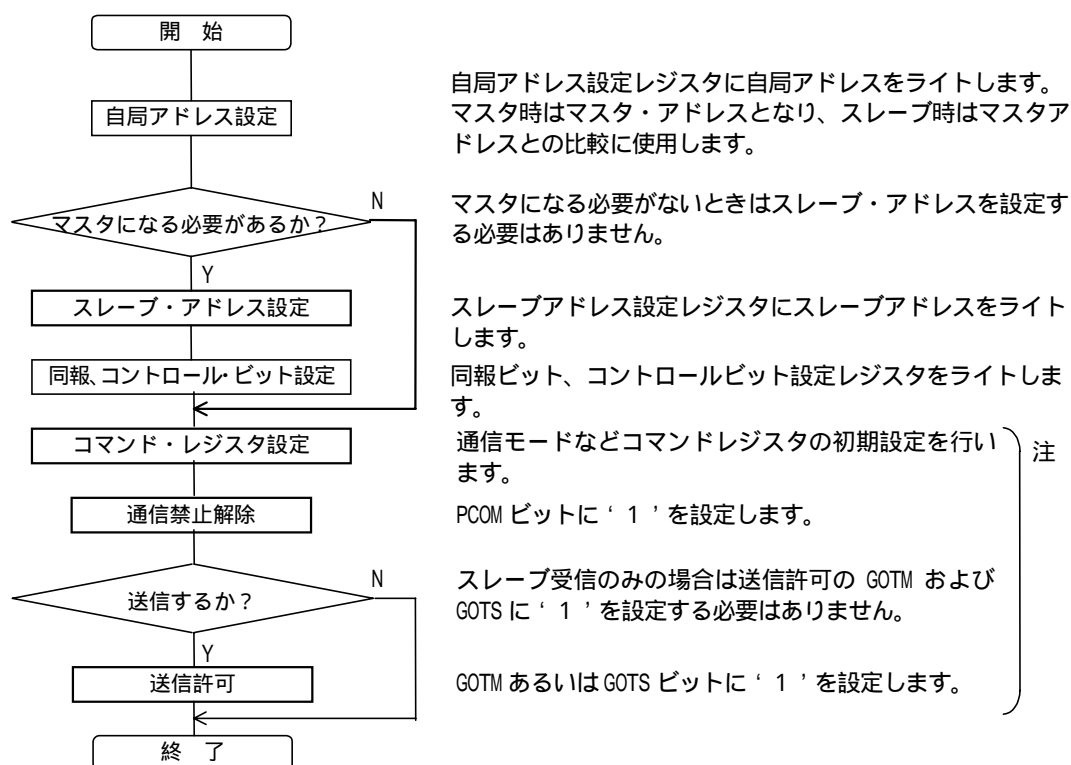
\*3：「20.8 IEBus™ コントローラの各種処理フロー例」の「■マスタ受信処理ルーチン」参照

図 21.7-2 割り込み処理ルーチン

## 21.7.1 初期設定処理ルーチン

図 21.7-3では、IEBus™の初期設定の処理ルーチンを示します。自局のアドレスとスレーブアドレスを設定し、コマンドレジスタを設定し、通信禁止状態を解除します。スレーブアドレス、同報ビットおよびコントロールビットは、マスタになる必要がないときは設定する必要はありません。

初期設定処理ルーチン



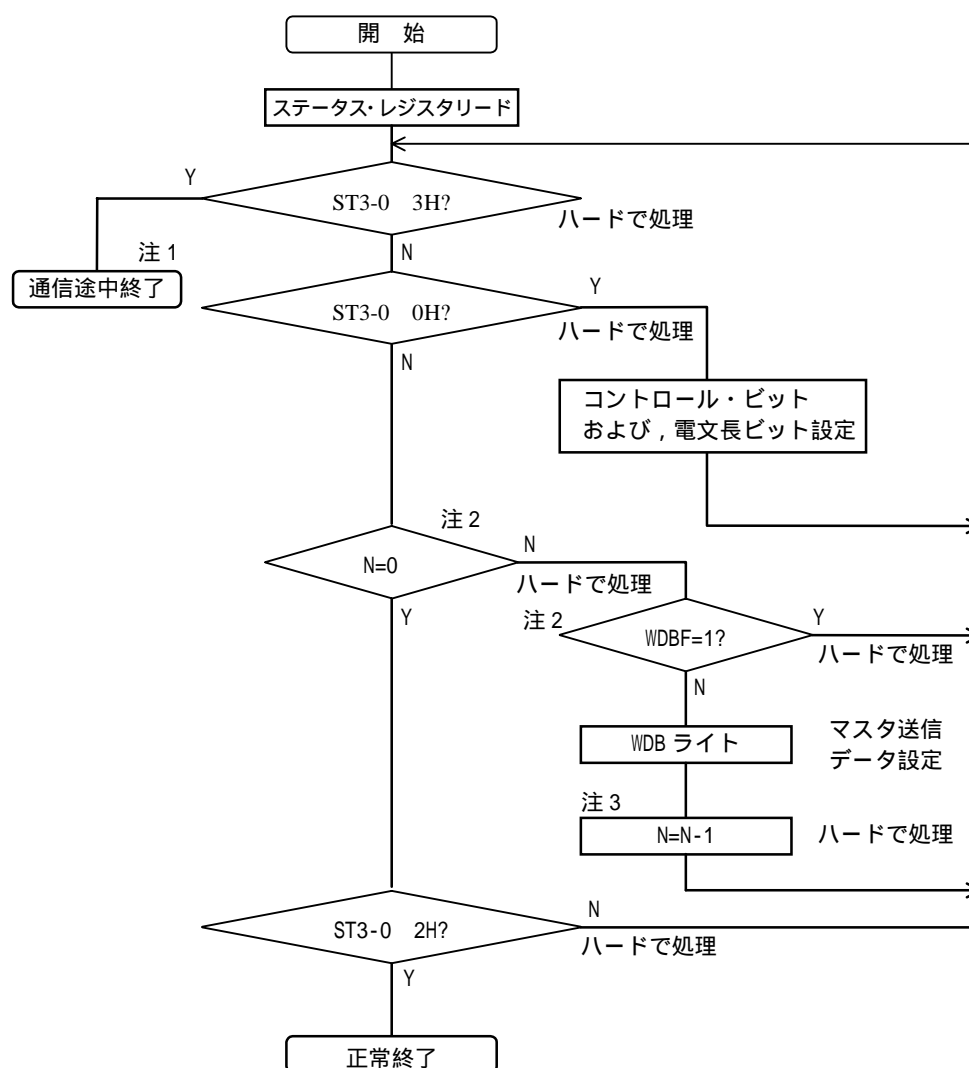
注：コマンド・レジスタのライト動作を行うことで実行されます。

図 21.7-3 初期設定処理ルーチン

## 21.7.2 マスタ送信処理ルーチン

図 21.7-4に、マスタ送信処理ルーチンを示します。マスタ送信処理ルーチンは、通信禁止解除後、アービトレーションに勝ち残りマスタユニットとなり、スレーブユニットにデータを送信する場合の処理ルーチンです。この処理ルーチンでは、割り込み処理ルーチンでステータスレジスタのST3-0の4ビットがマスタ送信のステート(上位2ビットが00<sub>h</sub>)となっていたとき実行されます。

マスタ送信処理ルーチン



注1：ステータス・レジスタのTE, PEF, ACKビットを読み出すことによって途中終了した原因を知ることができます。このとき、WDBにはデータが残っているため、再度送信処理を行うと、WDBの残りのデータを送信します。WDBをクリアする場合は、コマンド・レジスタのWDBCビットに"1"をライトします。

注2：WDBF=1でWDBにライトしないで下さい。

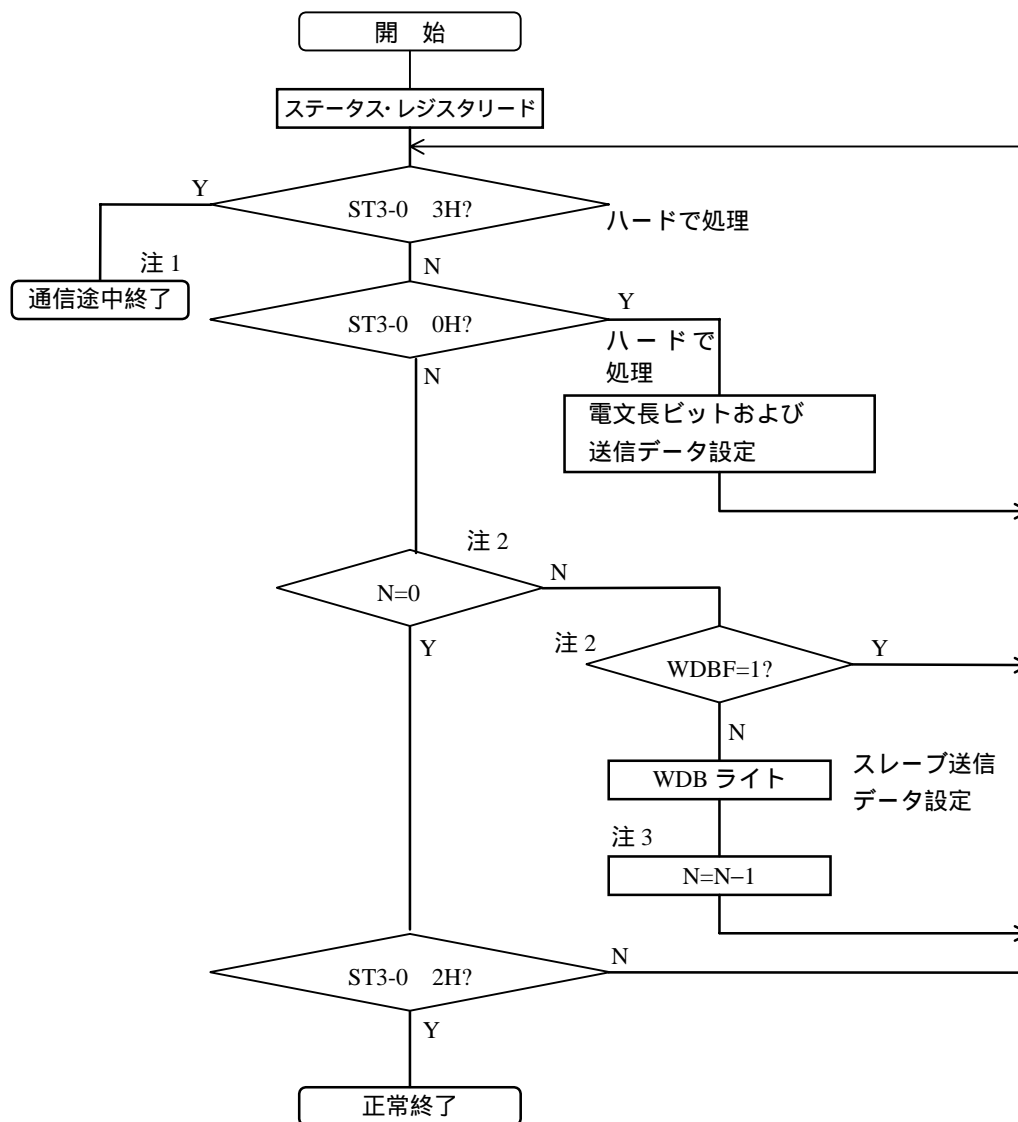
注3：N：マスタ送信データ・バイト数

図 21.7-4 マスタ送信処理ルーチン

### 21.7.3 スレーブデータ送信処理ルーチン

図 21.7-5に示すスレーブデータ送信処理ルーチンは、スレーブユニットがコントロールビット受信後、スレーブ送信となった時、マスタユニットにデータを送信する場合の処理ルーチンです。この処理ルーチンでは、割込み処理ルーチンでステータスレジスタのST3-0の4ビットがスレーブデータ送信のステート(上位2ビットが00<sub>B</sub>)となっていた時実行されます。

スレーブデータ送信処理ルーチン



注1：ステータス・レジスタのTE, PEF, ACKビットを読み出すことによって途中終了した原因を知ることができます。このとき、WDBにはデータが残っているため、再度送信処理を行うと、WDBの残りのデータを送信します。WDBをクリアする場合は、コマンド・レジスタのWDBCビットに "1" をライトします。

注2：WDBF=1でWDBにライトしないで下さい。

注3：N:スレーブ送信データ・バイト数

図 21.7-5 スレーブデータ送信処理ルーチン

## 21.7.4 マスタ受信処理ルーチン

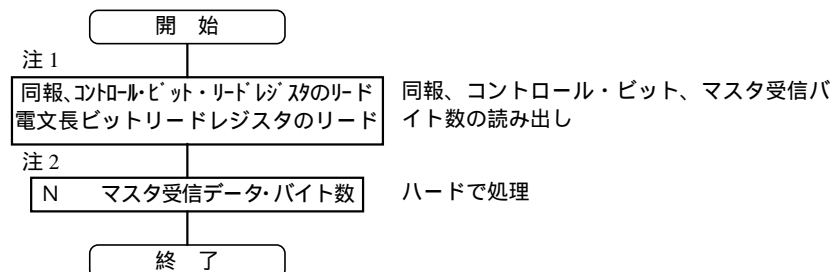
マスタ受信処理ルーチンは、マスタユニットがデータ送信開始後、スレーブユニットよりデータ、スレーブステータス、ロックアドレスを受信する場合の処理ルーチンです。

### マスタ受信処理ルーチン

マスタ受信処理ルーチンは、ステータスレジスタのST3-0の4ビットの内容によって、以下に示す4種類のルーチンより構成されています。

#### マスタ受信開始のステータスコード(4H)のときの処理

図 21.7-6に、マスタ受信開始のステータスコード(4H)のときの処理フローを示します。マスタユニットが、スレーブユニットより電文長ビットを正しく受信したときにこのステータスコードが発生し、マスタ受信の開始を知らせます(ただし、この時割り込みは発生しません)。

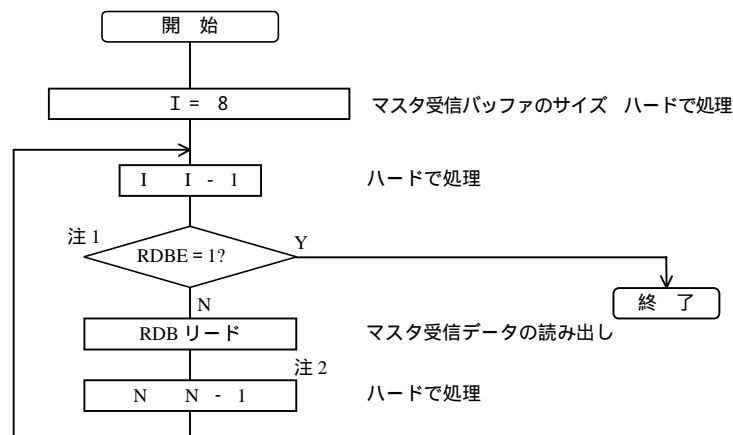


注1：上記内容は各レジスタに設定されるため、必ずしも本ステータスで読む必要はありません。ただし、各レジスタの設定タイミングに注意し、レジスタが更新される前に読む必要があります。  
注2：N：マスタ受信データ・バイト数

図 21.7-6 マスタ受信開始処理フロー

#### マスタ受信データ読み出し要求のステータスコード(5H)のときの処理

図 21.7-7に、マスタ受信データ読み出し要求のステータスコード(5H)のときの処理フローを示します。

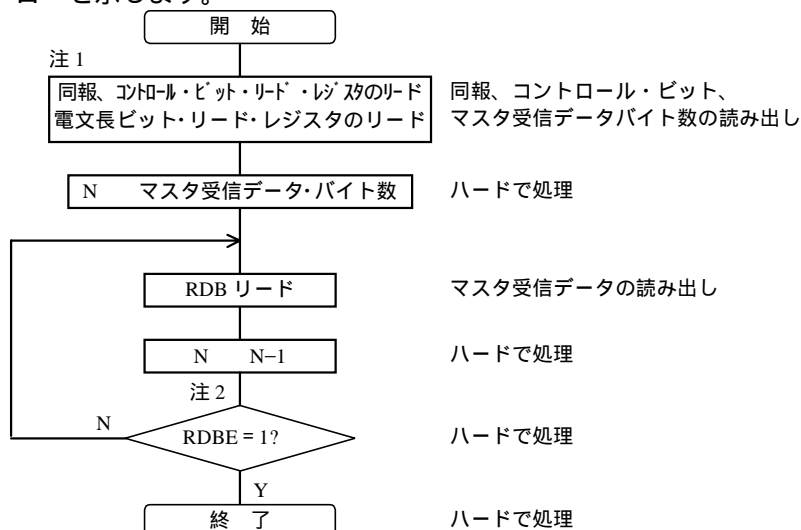


注1：RDBE=1のとき、RDBをリードしないでください。  
注2：N：マスタ受信データ・バイト数

図 21.7-7 マスタ受信データ読み出し要求処理ルーチン

# マスタ受信正常終了のステータス・コード(6H)のときの処理

図 21.7-8にマスタ受信正常終了のステータス・コード(6H)のときの処理フローを示します。



注1: これらのデータは各個別のレジスタに格納されるため、必ずしも本ステータスで読む必要はありません。次の通信フレームが開始され、データが更新されるまでに読み出してください。

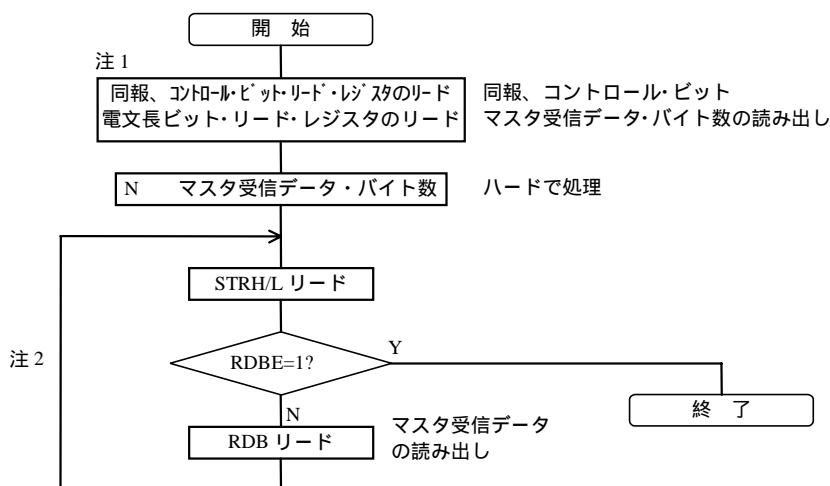
注2: RDBE=1でRDBをリードしないでください。

図 21.7-8 マスタ受信正常終了処理ルーチン

# マスタ受信途中終了のステータスコード(7H)のときの処理

図 21.7-9に、マスタ受信途中終了のステータスコード(7H)のときの処理ルーチンを示します。

なお、スレーブ受信処理および同報受信処理については、マスタ受信処理と同様です。ただし、ステータス・コードは異なります。「21.6 通信制御のステータス」を参照してください。



注1: これらのデータは各個別のレジスタに格納されるため、必ずしも本ステータスで読む必要はありません。次の通信フレームが開始され、データが更新されるまでに読み出してください。

注2: RDBE=1でRDBをリードしないで下さい。

図 21.7-9 マスタ受信途中終了

## 21.8 IEBus™コントローラの送信時の動作

図 21.8-1と図 21.8-2に、複数フレームにわたる送信時の動作を、図 21.8-3と図 21.8-4に、エラー発生時の送信データを示します。

複数フレームにわたる送信時の動作

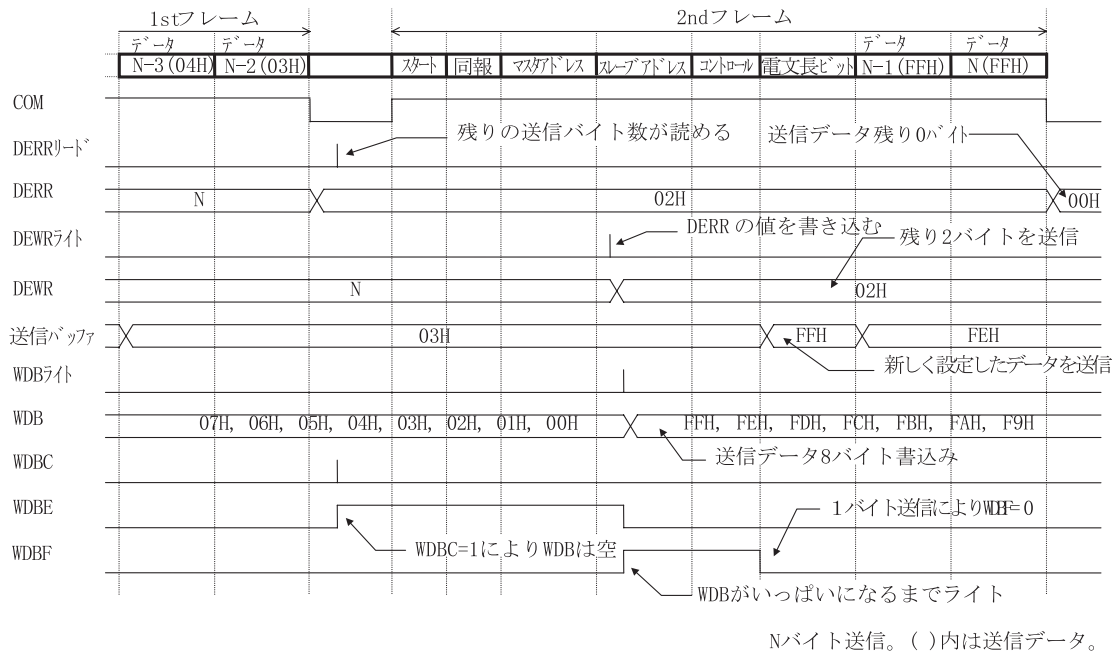
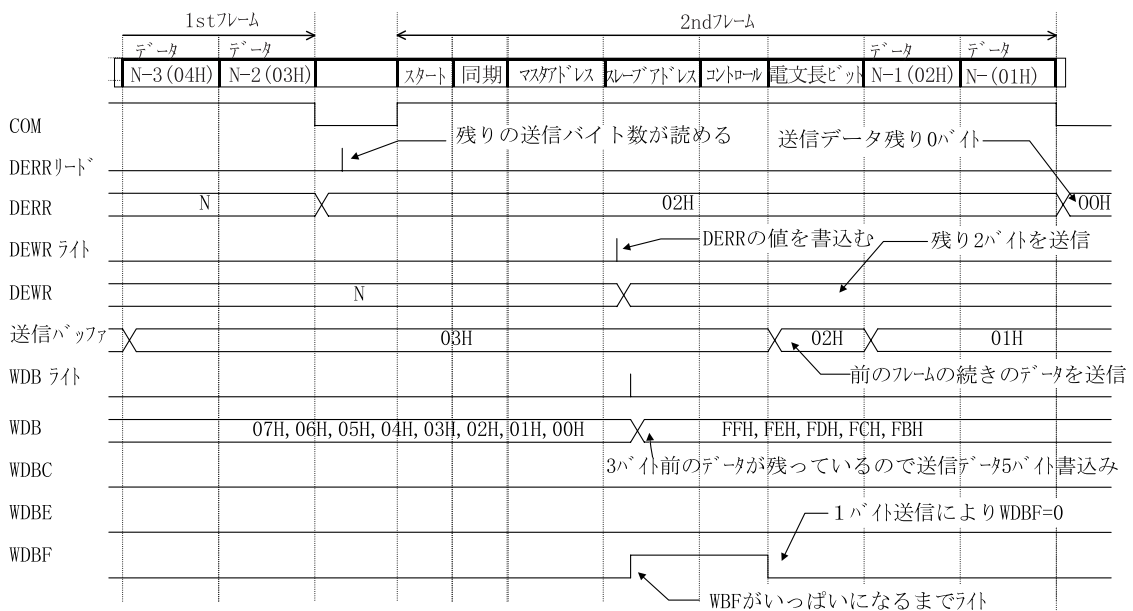


図 21.8-1 WDBCに"1"を設定した場合（マスタ送信のマスタ側）



\*: Nバイト送信。( )内は送信データ。

\*: 送信バッファは次に送るデータを設定するバッファでハードでのみアクセス可能なバッファです。

\*: マスタ受信の場合もDEWRやWDBにデータを設定するタイミングは異なりますが(「21.5 IEBus™の受信制御」「21.6 通信制御のステータス」参照) DERRに値が設定されるタイミングは同じです。

図 21.8-2 WDBCに"0"を設定した場合（マスタ送信のマスタ側）



## エラー発生時の送信データ

図 21.8-3に、マスタ送信でスレーブ側に2バイト目のデータでエラーが発生し、マスタがNAKを受信、第2フレームで続きのデータを送信する場合の動作例を示します。

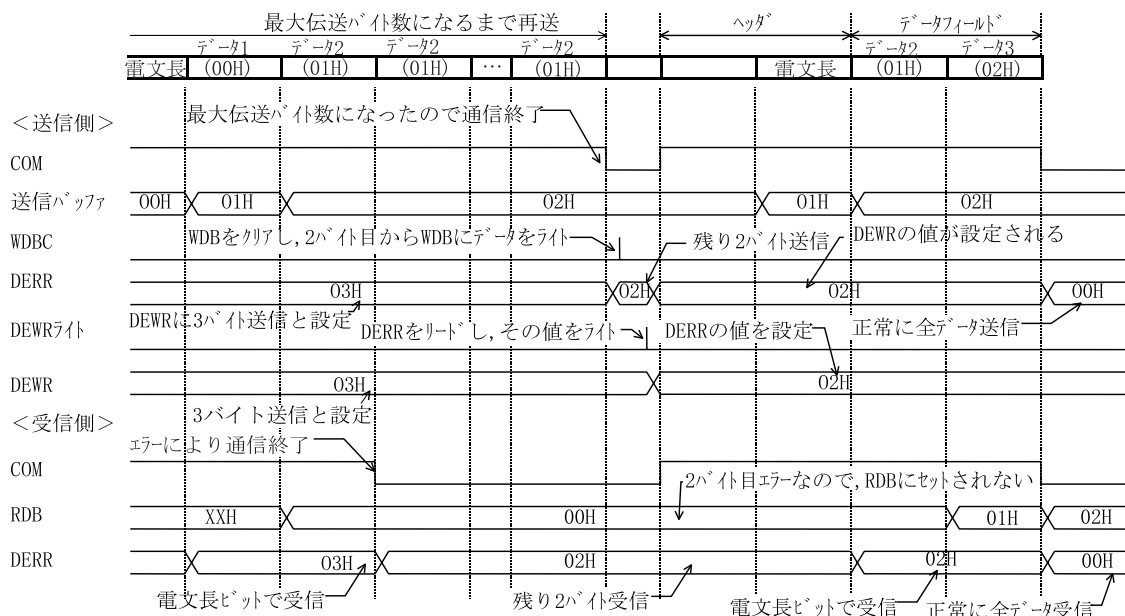
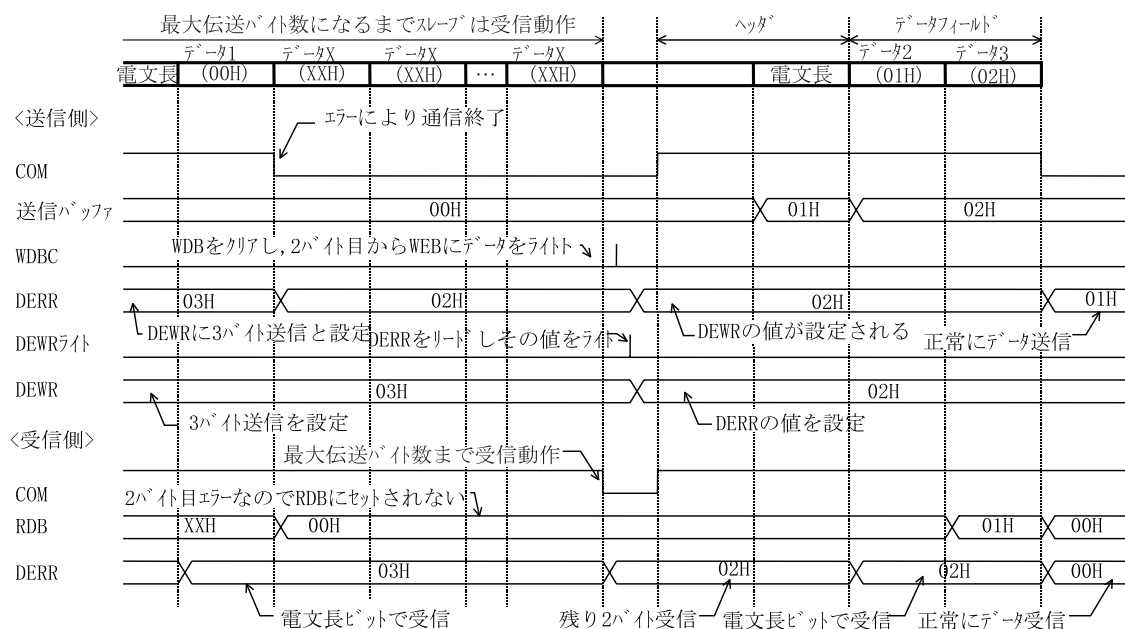


図 21.8-3 スレーブ側でエラーが発生した場合の動作例

図 21.8-4に、マスタ送信でマスタ側に2バイト目のデータでエラーが発生し、第2フレームで続きのデータを送信する場合の動作例を示します。



- \* 送信バッファは次に送るデータを設定するバッファでハードでのみアクセス可能なバッファです。
- \* RDBは8バイトのうち一番最初に設定されるところのデータを示します。
- \* スレーブ送信の場合もDEWRやWDBCにデータを設定するタイミングは異なりますが(「21.5 IEBus™の受信制御」「21.6 通信制御のステータス」参照) DERRの設定されるタイミングは同じです。

図 21.8-4 マスタ側でエラーが発生した場合の動作

## 21.9 IEBus™プロトコル動作

IEBus™ (Inter Equipment Bus)は、装置と装置との間のデータ伝送を行うことを目的とした、小規模のデジタルデータ伝送システム用のバスです。

### IEBus™プロトコル動作の概要

IEBus™プロトコル動作の概要は、次の通りです。

通信方式

半2重非同期通信

マルチ・マスタ方式

IEBus™に接続している全てのユニットが他のユニットへデータの伝送が実現できます。

同報通信機能(1ユニット対複数ユニットの通信)

- グループ同報通信：グループユニットに対しての同報通信
- 一斉同報通信：全てのユニットに対しての同報通信

伝送速度の異なる3種類のモードが選択可能

表 21.9-1 IEBus™プロトコル動作のモード別転送速度

	IEBus™内部周波数 (6MHz)	IEBus™内部周波数 (6.29MHz)	最大伝送バイト数 (バイト/フレーム)
モード0	約3.9K(bps)	約4.1 K(bps)	16
モード1	約17 K(bps)	約18 K(bps)	32
モード2	約26 K(bps)	約27 K(bps)	128

アクセス制御

CSMA/CD(Carrier Sense Multiple Access with Collision Detection)

バス占有の優先順位

- 1) 同報通信が通常の通信(1ユニット対1ユニットの通信)より優先
- 2) マスタ・アドレスが小さい方が優先

通信規模

- ユニット数 : 最大50
- ケーブル長 : 最大150m (ツイストペアケーブルを使用時<抵抗値 0.1Ω/m以下>)
- 負荷容量：最大8000pF <BUS- とBUS+間> IEBus™内部周波数= 6MHz  
: 最大7100pF <BUS- とBUS+間> IEBus™内部周波数= 6.29MHz
- 終端抵抗：120Ω

なお、通信規模は、IEBus™用ドライバ/レシーバを含めたシステム上での規模です。

### バス占有権の決定(アービトレーション)

IEBus™に接続された装置は他の装置を制御するときに、バスを占有するための動作を行います。この動作のことを、アービトレーションと呼びます。

アービトレーションでは、複数のユニットが同時に送信を開始した場合にそれらの複数のユニット中から1つのユニットに対しバスを占有する許可を与える処理が行われます。

アービトレーションにより1装置のみがバス占有権を得るため、次のようなバス占有の優先条件が決められています。

#### 通信の種類による優先

同報通信(1ユニット対複数ユニットの通信)が通常通信(1ユニット対1ユニットの通信)より優先されます。

#### マスタアドレスによる優先

通信種類が同じ場合には、マスタ・アドレスが最も小さいものが優先されます。

例えば、マスタアドレスは12ビットで構成され、000Hのユニットが最上位の優先順位を持ち、FFFHのユニットが最下位の優先順位を持ちます。

### 通信モード

IEBus™には、伝送速度の異なる3種類の通信モードが用意されています。各通信モードにおける、伝送速度および1通信フレーム中の最大伝送バイト数は下表に示す通りです。

表 21.9-2 各通信モードにおける伝送速度と最大伝送バイト数

通信モード	最大伝送バイト数 (バイト/フレーム)	実効伝送速度(bps) *1	
		IEBus™内部周波数 *2	IEBus™内部周波数 *2
0	16	約3.9K	約4.1K
1	32	約17K	約18K
2	128	約26K	約27K

\*1: 最大伝送バイト数を伝送したときの実効伝送速度

\*2: IEBus™の内部周波数とCPUの内部周波数の関係は、表 21.3-4の計算式を参照してください。

### < 注意事項 >

- IEBus™に接続した各装置は、通信を行う前に予め通信モードを選択しておきます。また、マスタユニットとその通信相手ユニット(スレーブユニット)の通信モードが同一でないと、通信は正しく行われません。
- IEBus™の内部周波数がユニットによって違っていると、通信モードが同一でも通信は正しく行われません。必ず全ユニット同じ周波数で通信してください。

### 通信アドレス

IEBus™では、各装置に12ビットの固有な通信アドレスが割り当てられます。通信アドレスは、次の様に構成されます。

- 上位4ビット：グループ番号(各装置の所属するグループを識別する番号)
- 下位8ビット：ユニット番号(グループ内の各装置を識別する番号)

## 同報通信

通常の通信では、マスタ・ユニットがその通信相手局となるスレーブユニットは1ユニットで、1対1の送信または、受信が行われます。それに対し、同報通信では、スレーブ・ユニットが複数存在し、マスタユニットは複数のスレーブユニットに対して送信を行います。スレーブユニットは複数存在するため、通信中スレーブユニットからは、アクノリッジ信号は返されません。

また、同報通信を行うか、通常の通信を行うかは同報ビットによって選択することができます。

同報通信には、次の2種類があります。

### グループ同報通信

グループ同報通信は、通信アドレスの上位4ビットのグループ番号が等しいグループ内の装置に対して同報通信を行います。

### 一斉同報通信

一斉同報通信は、グループ番号の値にかかわらず全ての装置に対して同報通信を行います。グループ同報と一斉同報は、スレーブアドレスの値により識別されます。

21.10 伝送プロトコル

IEBus™の伝送信号フォーマットを、表 21.10-1に示します。  
通信データは、通信フレームと呼ぶ1連の信号として伝送されます。1通信フレームで伝送可能なデータ数および伝送速度は、通信モードによって異なります。

伝送プロトコル

表 21.10-1 伝送信号フォーマット

フィールド名	ヘッダ		マスタアドレス フィールド		スレーバアドレス フィールド			コントロール フィールド			電文長 フィールド			データ・フィールド					
	1 1		12 1		12 1 1			4 1 1			8 1 1			8 1 1		8 1 1			
	スタート ビット	同報 ビット	マスタ アドレス	P	スレーバ アドレス	P	A	コントロ ール ビット	P	A	電文長 ビット	P	A	データ 1バイト	P	A	...	データ 1バイト	P
モード0	約7330μs													約1590×Nμs					
モード1	約2090μs													約410×Nμs					
モード2	約1590μs													約300×Nμs					

P：パリティ・ビット(1ビット)    N：データ・バイト数  
A：アクノリッジ・ビット(1ビット)    0：ACK    1：NAK  
(注) 同報通信時は、アクノリッジ・ビットの値は無視されます。

## 21.10.1 伝送プロトコルのヘッダ

---

伝送プロトコルのヘッダは、スタート・ビット、および、同報ビットより構成されています。

---

### スタートビット

スタートビットは、データ伝送の始まりを他のユニットに知らせるための信号です。データ伝送を開始しようとするユニットは、決められた時間、ローレベルの信号(スタートビット)を出力し、同報ビットの出力へ移行します。

スタートビットを出力しようとしたとき、すでに他のユニットがスタートビットを出力している場合には、スタートビットを出力しないでそのユニットのスタートビット出力終了を待ち、その終了タイミングに同期して同報ビット出力へ移行します。送信を始めたユニット以外は、このスタートビットを検出し受信状態へ移行します。

### 同報ビット

同報ビットは、同報通信、または通常の通信の識別を行うビットです。同報ビットが、"0"の場合には同報通信を表し、"1"の場合には通常の通信を表します。また、同報通信には、グループ同報と一斉同報があり、これらの識別はスレーブアドレスの値によって行われます。

同報通信の場合には通信相手局となるスレーブユニットが複数存在するため、以後の各フィールドでのアクノリッジビットは返されません。

2つ以上のユニットが同じタイミングで通信フレームの送出を開始した場合には、同報通信が通常の通信より優先し、アービトレーションに勝ち残ります。

## 21.10.2 伝送プロトコルのマスタアドレスフィールド

---

マスタアドレスフィールドは、マスタ局を決定するフィールドです。

---

### マスタアドレスフィールド

マスタアドレスフィールドは、12ビットのマスタアドレスビットとパリティビットで構成されています。

2つ以上のユニットが、同じタイミングで同じ値の同報ビットの送信を開始した場合、アービトレーションの判定はマスタアドレスフィールドへ持ち越されます。マスタアドレスフィールドでは、1ビット送信するごとに自分が出力しているデータとバス上のデータとの比較を行います。比較の結果、自分の出力しているマスタアドレスとバス上のデータが異なった場合、アービトレーションに負けたと判断し、送信を中止し受信状態にまわります。

IEBus™のバスは、ワイヤードANDで構成されているため、アービトレーションに参加しているユニット(アービトレーションマスタ)の中で、最小のマスタアドレスを持つユニットがアービトレーションに勝ち残ります。

12ビットのマスタアドレス出力後、1つのユニットのみがマスタユニットに残ります。次に、このマスタユニットはパリティビット\*を出力し、他のユニットに対してマスタアドレスを確定させ、スレーブアドレスフィールド出力へ移行します。

マスタアドレスは、12ビットで構成されておりMSBより出力されます。

\*： パリティは偶数パリティを使用しており、マスタアドレスビット中の "1" のビットの数が奇数のときパリティビットが"1"となります。

### 21.10.3 伝送プロトコルのスレーブアドレスフィールド

---

スレーブアドレスフィールドは、スレーブ局を決定するフィールドです。

---

#### スレーブアドレスフィールド

スレーブアドレスフィールドは、12ビットのスレーブアドレスビット、パリティビットおよび、アクノリッジビットで構成されています。

スレーブアドレスは、12ビットで構成されMSBより出力されます。12ビットのスレーブアドレス送信後、スレーブアドレスが間違っ受受信されることを防ぐため、パリティビットを出力します。次に、スレーブユニットがバス上に存在することを確認するために、マスタユニットはスレーブユニットからのアクノリッジ信号の検出を行います。アクノリッジ信号を検出した場合、コントロールフィールド出力へ移行します。ただし、同報通信時には、アクノリッジビットを検出せずに、コントロールフィールド出力へ移行します。

スレーブユニットは、スレーブアドレスが一致し、マスタアドレスとスレーブアドレスの両方のパリティが、偶数であることを検出した場合、アクノリッジ信号を出力します。スレーブユニットはパリティが奇数の場合、マスタアドレスまたはスレーブアドレスが正しく受信されなかったと判断し、アクノリッジ信号を出力しません。このとき、マスタユニットは待機(モニタ)状態になり、通信が終了します。

また、同報通信の場合には、スレーブアドレスは次のようにグループ同報か、一斉同報かの識別に使用されます。

- スレーブアドレスが FFFH の時           ：一斉同報通信
- スレーブアドレスが FFFH 以外の時   ：グループ同報通信

なお、グループ同報通信時のグループ番号は、スレーブアドレスの上位4ビットの値になります



## 21.10.4 伝送プロトコルのコントロールフィールド

---

コントロールフィールドは、データフィールドの種類や方向を決定するフィールドです。

---

### 伝送プロトコルのコントロールフィールド

コントロールフィールドは、4ビットのコントロールビット、パリティビットおよび、アクノリッジビットで構成されています。

コントロールビットは、4ビットで構成されMSBより出力されます。コントロールビットに続いて、パリティビットが出力されます。パリティが偶数でかつ、マスタユニットの要求機能をスレーブが実行可能な場合は、スレーブユニットはアクノリッジ信号を出力し、次の電文長フィールドへ移行します。ただし、パリティが偶数でもスレーブユニットがマスタユニットの要求を実行できない場合や、パリティが奇数の場合は、スレーブユニットはアクノリッジ信号を出力せず、待機状態に戻ります。

マスタユニットは、アクノリッジ信号を確認後、次の電文長フィールドへ移行します。アクノリッジ信号の確認ができない場合は、マスタユニットは待機(モニタ)状態になり、通信が終了します。ただし、同報通信時には、マスタ・ユニットはアクノリッジ信号を確認せずに、次の電文長フィールドへ移行します。

## 21.10.5 電文長フィールド

電文長フィールドは、通信データのバイト数を指定するためのフィールドです。

### 電文長フィールド

電文長フィールドは、8ビットの電文長ビットと、パリティビット、および、アクノリッジビットより構成されます。電文長ビットは、8ビットで構成され、MSBから出力されます。表 21.10-2のように通信データのバイト数を表します。

表 21.10-2 電文長ビットの内容

電文長ビット(16進)	送信データ・バイト数
01H	1 $\overline{\text{A}}$ 1 $\overline{\text{B}}$
02H	2 $\overline{\text{A}}$ 1 $\overline{\text{B}}$
:	:
FFH	255 $\overline{\text{A}}$ 1 $\overline{\text{B}}$
00H	256 $\overline{\text{A}}$ 1 $\overline{\text{B}}$

備考：通信モードにより、1フレームの最大伝送バイト数以上を設定すると、複数フレームでの通信となります。その際、2回目以降は、電文長ビットは残りの通信データバイト数となります。

このフィールドの動作は、マスタ送信時(コントロールビットのビット3が"1")とマスタ受信時(コントロールビットのビット3が"0")で異なります。

#### マスタ送信時

電文長ビットおよびパリティビットは、マスタユニットによって出力されます。スレーブユニットは、偶数であることを検出した場合、アクノリッジ信号を出力し、次のデータフィールドへ移行します。

ただし、同報通信時では、スレーブユニットは、パリティが奇数の場合、電文長ビットが正しく受信されなかったと判断し、アクノリッジ信号を出力せず、待機状態に戻ります。このとき、マスタ・ユニットも、待機(モニタ)状態に戻り、通信が終了します。

#### マスタ受信時

電文長ビットおよびパリティビットは、スレーブユニットによって出力されます。マスタユニットは、パリティビットが偶数であることを検出した場合、アクノリッジ信号を出力します。

マスタユニットはパリティが奇数の場合、電文長ビットが正しく受信されなかったと判断し、アクノリッジ信号を出力せず、待機状態に戻ります。このとき、スレーブユニットも、待機状態に戻り、通信が終了します。

## 21.10.6 データフィールド

データフィールドは、コントロールビットにしたがいデータを送受信するフィールドです。

### データフィールド

データフィールドは、8ビットのデータビット、パリティビットおよびアクノリッジビットの計10ビットで構成されています。

データビットは、8ビットで構成されMSBより出力されます。データビットに続きパリティビットとアクノリッジビットが、それぞれマスタユニットおよびスレーブユニットより出力されます。

同報通信は、マスタユニットの送信動作のみに行われます。また、この時はアクノリッジ信号は無視されます。

マスタ送信時とマスタ受信時の動作は次のようになります。

#### マスタ送信時

マスタユニットからスレーブユニットへ書き込みを行う場合、マスタユニットは、スレーブユニットに対してデータビット、パリティビットを送信します。スレーブユニットは、データビット、パリティビットを受信し、パリティが偶数で、かつ、受信バッファが空いていれば、アクノリッジ信号を出力します。パリティが奇数、または、受信バッファが空いていない場合には、スレーブユニットは、対応するデータの受け付けを拒否し、アクノリッジ信号出力を行いません。

スレーブユニットからアクノリッジ信号が出力されなかった場合、マスタユニットは再び同じデータを送信します。この動作はスレーブユニットからのアクノリッジ信号を検出するか、データが最大伝送バイト数を超えるまで続けられます。パリティが、偶数でスレーブユニットよりアクノリッジ信号が出力された場合は、データに続きがあり、かつ最大伝送バイト数を越えていなければ、マスタユニットは次のデータを送信します。

また、同報通信時では、スレーブユニットからはアクノリッジ信号は出力されず、マスタユニットはデータを1バイトごとに転送します。

#### マスタ受信時

マスタユニットがスレーブユニットから読み込みを行う場合、マスタユニットはすべての読み込みビットに対応する同期信号を出力します。

スレーブユニットは、データ、パリティビットの内容をマスタユニットからの同期信号に応じてバス上に出力します。

マスタユニットは、スレーブユニットの出力したデータ、パリティビットを読み込み、パリティを確認します。

パリティが奇数の場合、または、受信バッファが空いていない場合、マスタユニットはそのデータの受け付けを拒否し、アクノリッジ信号を出力しません。1通信フレームで送信可能な最大伝送バイト数以内であれば、マスタユニットは同じデータの読み込み動作を繰り返します。

また、パリティが偶数でかつ、受信バッファが空いていれば、マスタユニットはデータを受け付け、アクノリッジ信号を返します。1フレームで送信可能な最大バイト数以内であればマスタ・ユニットは次のデータを読み込みます。

## 21.10.7 パリティビット

---

パリティビットは、伝送データに誤りがないことを確認するために使用されます。パリティビットは、マスタアドレスビット、スレーブアドレスビット、コントロールビット、電文長ビット、データビットの各データに対して付加されます。

---

### パリティビット

パリティは、偶数パリティです。データ中の"1"のビット数が奇数の場合は、パリティビットは"1"となります。データ中の"1"の数が偶数の場合は、パリティビットは"0"となります。

## 21.10.8 アクノリッジビット

通常の通信(1ユニット対1ユニット)においては、データを正しく受け付けたかを確認するために、次の個所においてアクノリッジ・ビットが付加されます。

- ・スレーブアドレスフィールドの最後
- ・コントロールフィールドの最後
- ・電文長フィールドの最後
- ・データ・フィールドの最後

### アクノリッジビット

アクノリッジ・ビットの定義は、以下の通りです。

"0"：伝送データを確認したことを表します。(ACK)

"1"：伝送データを認識しなかったことを表します。(NAK)

ただし、同報通信時には、アクノリッジ・ビットの内容は無視されます。

#### スレーブアドレスフィールドの最後のアクノリッジビット

スレーブアドレスフィールドの最後のアクノリッジビットは、次に示すいずれかの条件を満たした場合、NAKとなり、通信は中止されます。

- マスタアドレスビット、または、スレーブアドレスビットのパリティが正しくない場合。
- タイミングエラー(ビットフォーマットにエラー)が発生した場合。
- スレーブユニットが存在しなかった場合。

#### コントロールフィールドの最後のアクノリッジビット

コントロールフィールドの最後のアクノリッジビットは、次に示すいずれかの条件を満たした場合、NAKとなり、伝送は中止されます。

- コントロールビットのパリティが正しくない場合。
- スレーブ受信バッファ<sup>注)</sup>が空でないのに、コントロールビットのビット3が"1"(書き込み動作)の場合。
- スレーブ送信バッファ<sup>注)</sup>が空なのに、データの読み込みのコントロールビットが "3H", "7H" の場合。
- ロックを設定されているのに、ロックを設定したユニット以外からコントロールビットが "3H", "6H", "7H", "AH", "BH", "EH", "FH" の場合。
- ロックを設定されていないのに、ロックアドレスの読み込みのコントロールビットが"4H"の場合。
- タイミングエラーが発生した場合。
- 未定義のコントロールビットの場合。

### <注意事項>

スレーブ受信バッファについては、「21.11 伝送データ」のスレーブステータス(SSR)の読み込み(コントロール・ビット0H, 6H)参照してください。

#### 電文長フィールドのアクノリッジ・ビット

電文長フィールドの最後のアクノリッジ・ビットは、次に示すいずれかの条件を満たした場合、NAKとなります。

- 電文長ビットのパリティが正しくない場合。
- タイミング・エラーが発生した場合。

#### データフィールドの最後のアクノリッジビット

データフィールドの最後のアクノリッジビットは、次に示すいずれかの条件を満たした場合、NAKとなります。

- データビットのパリティが正しくない場合<sup>注</sup>
- タイミングエラーが前回のアクノリッジビット伝送以降で発生した場合。
- 受信バッファがいっぱいになり、それ以上のデータを受け付けることができない場合。<sup>注)</sup>

注) この場合、送信側は1フレームで伝送可能な最大伝送バイト数以内であれば、最大伝送バイト数に達するまでそのデータフィールドの送信を再実行します。

## 21.11 伝送データ

データフィールドの内容は、コントロールビットで示されるデータになります。

伝送データ

表 21.11-1 コントロールビットの内容

	ビット3 <sup>*1</sup>	ビット2	ビット1	ビット0	機能 <sup>*2</sup>
0H	0	0	0	0	スレーブ・ステータス(SSR)の読み込み
1H	0	0	0	1	未定義
2H	0	0	1	0	未定義
3H	0	0	1	1	データ読み込みとロック
4H	0	1	0	0	ロックアドレスの読み込み(下位8ビット)
5H	0	1	0	1	ロックアドレスの読み込み(上位4ビット)
6H	0	1	1	0	スレーブステータス(SSR)の読み込みとロック解除
7H	0	1	1	1	データ読み込み
8H	1	0	0	0	未定義
9H	1	0	0	1	未定義
AH	1	0	1	0	コマンド書き込みとロック
BH	1	0	1	1	データ書き込みとロック
CH	1	1	0	0	未定義
DH	1	1	0	1	未定義
EH	1	1	1	0	コマンド書き込み
FH	1	1	1	1	データ書き込み

\*1：注意ビット3(MSB)の値により、以後の電文長フィールドの電文長ビットおよびデータ・フィールドのデータの転送方向が変わります。

ビット3が "1" の場合： マスタ・ユニットからスレーブ・ユニットへ転送

ビット3が "0" の場合： スレーブ・ユニットからマスタ・ユニットへ転送

\*2：3H, 6H, AH, BHはロックの設定、および解除を指定するコントロール・ビットです。

1H, 2H, 8H, 9H, CH, DHの未定義値が送信された場合はアクノリッジを返しません。

マスタユニットによりロック指定されたユニットは、ロックを要求したマスタユニット以外から受信したコントロールビットが表6-4以外の場合、受け付けを拒否し、アクノリッジ・ビットを出力しません。

表 21.11-2 ロックされたスレーブユニットに対するコントロールフィールド

	ビット3	ビット2	ビット1	ビット0	機能
0H	0	0	0	0	スレーブ・ステータスの読み込み
4H	0	1	0	0	ロック・アドレスの読み込み(下位8ビット)
5H	0	1	0	1	ロック・アドレスの読み込み(上位4ビット)

## スレーブステータス(SSR)の読み込み(コントロールビット：0H, 6H)

マスタユニットは、スレーブステータスの読み込み(0H, 6H)を行うことにより、スレーブユニットが、アクノリッジ・ビット(ACK)を返送しなかった理由を知ることができます。

スレーブステータスは、スレーブユニットが最後に行った通信結果に対して決定されます。すべてのスレーブユニットは、スレーブステータスの情報を提供できます。スレーブステータスの意味を図 21.11-1に示します。

MSB				LSB			
ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0

図 21.11-1 スレーブステータスのビット構成

表 21.11-3 スレーブステータスの意味

ビット	値	意 味	
ビット0 <sup>*1</sup>	0	スレーブ送信バッファが空	
	1	スレーブ送信バッファが空でない	
ビット1 <sup>*2</sup>	0	スレーブ受信バッファがいっぱいでない	
	1	スレーブ受信バッファがいっぱい	
ビット2	0	ユニットがロック状態でない	
	1	ユニットがロック状態である	
ビット3	0	"0"固定	
ビット4 <sup>*3</sup>	0	スレーブ送信停止	
	1	スレーブ送信動作可能	
ビット5	0	"0"固定	
ビット6 ビット7	00	モード0	ユニットがサポートしている 最高位のモードを表します。 <sup>*4</sup>
	01	モード1	
	10	モード2	
	11	設定禁止	

<sup>\*1</sup>:スレーブ送信バッファとは、データ読み込み処理時(コントロールビット "3H", "7H")にアクセスされるバッファのことです。ライトデータバッファ(WDB)に該当します。

<sup>\*2</sup>:スレーブ受信バッファとは、データ書き込み処理時(コントロールビット "8H", "AH", "BH", "EH", "FH")にアクセスされるバッファのことです。リードデータバッファ(RDB)に該当します。

<sup>\*3</sup>:コマンド・レジスタのPCOMによって選択することができます。

<sup>\*4</sup>:MB90580Cシリーズでは"10"固定になります。



データコマンド転送(コントロールビット：読み込み(3H, 7H), 書き込み(AH, BH, EH, FH))

データ読み込み(3H, 7H)の場合，スレーブ・ユニットのデータバッファにあるデータが，マスタ・ユニットに読み込まれます。

データ書き込み(BH, FH)またはコマンド書き込み(AH, EH)の場合，スレーブユニットが受信したデータはそのスレーブ・ユニットの動作規定に従って処理されず。

#### 備考1

データとコマンドの選択は，ユーザがシステムに応じて自由に決めることができます。

#### 備考2

3H, AH, BHは通信条件，状態によりロックが設定されることがあります。

ロックアドレスの読み込み(コントロールビット：4H, 5H)

コントロールビット"4H"または，"5H"で，ロックアドレスの読み込み処理時には，ロック命令を発行したマスタユニットのアドレス(12ビット)が，図21.11-2に示すように1バイト単位に構成されて，読み出されます。

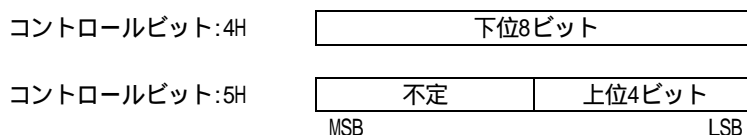


図 21.11-2 ロック・アドレスの構成

#### ロックの設定，解除

ロック機能は，メッセージを複数の通信フレームに渡って転送する場合に使用します。ロックの設定されたユニットは，ロックをかけたユニット以外からの受信は行いません。

ロックの設定，および解除は，次の様に行われます。

##### ロックの設定

ロックを指定したコントロールビット("3H", "AH", "BH")で，電文長フィールドのアクノリッジビット"0"の送受信終了後，電文長ビットにて指定されたデータバイト数分のデータを送信または，受信を成功せずに通信フレームを終了した場合に，スレーブユニットは，マスタユニットよりロックが設定されます。また，この時，スレーブステータスを表すバイト中のロックに関するビット(ビット2)が，"1"にセットされます。

##### ロックの解除

ロック指定したコントロールビット"3H", "AH", "BH"または，ロックの解除を指定したコントロールビット"6H"で，1通信フレーム内に，電文長ビットで指定したデータバイト数分のデータを送信または，受信を成功した場合に，スレーブユニットは，マスタユニットよりロックが解除されます。また，この時，スレーブステータスを表すバイト中のロックに関するビット(ビット2)が "0"にリセットされます。

なお，同報通信時にはロックの設定，および解除は行われません。

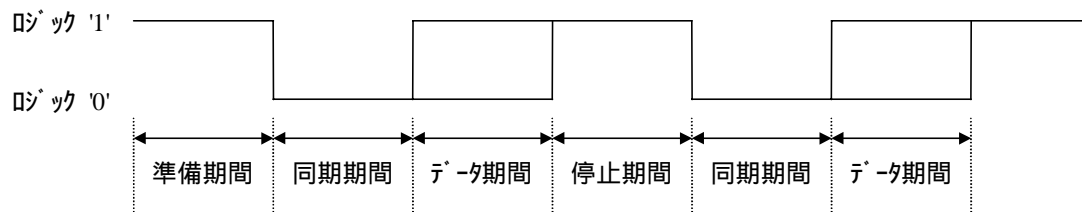
## 21.12 ビットフォーマット

IEBus™は、1ビットごとに同期が取られています。また、ビット全体の時間と、そのビット内に割り当てられている期間の時間に関する仕様は、伝送ビットの種類、マスタユニットかスレーブユニットかの違いにより異なります。

また、マスタユニット、スレーブユニットは通信中、各期間(準備期間、同期期間、データ期間)が所定の時間どおり出力されているかを検出しています。所定の時間通りに出力されていない場合には、マスタユニット、スレーブユニットはタイミングエラーとして、ただちに通信を終了し、待機状態にもどります。

### ビットフォーマット

IEBus™の通信フレームを構成するビットのフォーマットを、図 21.12-1に示します。



ロジック "1" : パスの線間(BUS+端子とBUS - 端子)の電位差が 20mV以下(ローレベル)

ロジック "0" : パスの線間(BUS+端子とBUS - 端子)の電位差が120mV以上(ハイレベル)

準備期間 : 最初以降のロウ・レベル(ロジック "1")期間

同期帰還 : 次のハイ・レベル(ロジック "0")期間

データ期間 : ビットの値を表す期間(ロジック "1" =ロウ・レベル, ロジック "0" =ハイ・レベル)

同期期間とデータ期間の長さは、ほぼ等しくなっています。

図 21.12-1 IEBus™のビットのフォーマット



## 第22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。

---

22.1 クロックモニタ機能の概要

22.2 クロック出力許可レジスタ (CLKR)

## 22.1 クロックモニタ機能の概要

クロックモニタ機能は、CK0T端子からメインクロックモード、PLLクロックモード、サブクロックモードで動作している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

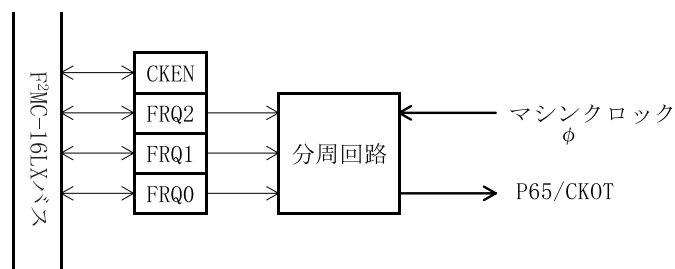


図 22.1-1 クロックモニタ機能のブロックダイアグラム

## 22.2 クロック出力許可レジスタ(CLKR)

クロック出力許可レジスタ(CLKR)の各ビットを使用して、CKOT出力許可やクロック出力周波数の選択をします。

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: 00003Eh	—	—	—	—	CKEN	FRQ2	FRQ1	FRQ0	CLKR
リード/ライト 初期値 ⇒	(-)	(-)	(-)	(-)	(R/W) (0)	(R/W) (0)	(R/W) (0)	(R/W) (0)	

図 22.2-1 クロック出力許可レジスタ(CLKR)

### 【ビット3】CKEN

CKENは、CKOT出力許可ビットです。

表 22.2-1 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

FRQ2, FRQ1, FRQ0ビットは、クロック出力周波数を選択するビットです。

表 22.2-2 FRQ2, FRQ1, FRQ0ビットの機能

FRQ2	FRQ1	FRQ0	出力クロック	$\phi = 16\text{MHz}$	$\phi = 8\text{MHz}$	$\phi = 4\text{MHz}$	$\phi = 8\text{kHz}$
0	0	0	$\phi / 2^1$	125ns	250ns	500ns	250 $\mu\text{s}$
0	0	1	$\phi / 2^2$	250ns	500ns	1 $\mu\text{s}$	500 $\mu\text{s}$
0	1	0	$\phi / 2^3$	500ns	1 $\mu\text{s}$	2 $\mu\text{s}$	1ms
0	1	1	$\phi / 2^4$	1 $\mu\text{s}$	2 $\mu\text{s}$	4 $\mu\text{s}$	2ms
1	0	0	$\phi / 2^5$	2 $\mu\text{s}$	4 $\mu\text{s}$	8 $\mu\text{s}$	4ms
1	0	1	$\phi / 2^6$	4 $\mu\text{s}$	8 $\mu\text{s}$	16 $\mu\text{s}$	8ms
1	1	0	$\phi / 2^7$	8 $\mu\text{s}$	16 $\mu\text{s}$	32 $\mu\text{s}$	16ms
1	1	1	$\phi / 2^8$	16 $\mu\text{s}$	32 $\mu\text{s}$	64 $\mu\text{s}$	32ms



## 第23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。

---

- 23.1 アドレス一致検出機能の概要
- 23.2 アドレス一致検出機能のレジスタ
- 23.3 アドレス一致検出機能の動作
- 23.4 アドレス一致検出機能の使用例



## 23.1 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。INT9割り込みルーチンで処理を行うことにより、アドレス一致検出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、各レジスタごとにコンペア許可ビットがあります。アドレス検出レジスタに設定された値とアドレスが一致して、かつ割り込み許可ビットが"1"の場合、CPUに読み込まれる命令コードを強制的にINT9命令のコードに置き換えます。

アドレス一致検出機能のブロックダイアグラム

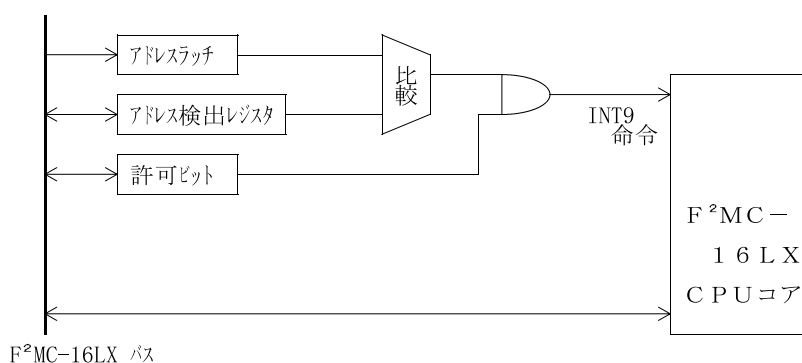


図 23.1-1 アドレス一致検出機能のブロックダイアグラム

23.2 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/PADR1)

・プログラムアドレス検出コントロール/ステータスレジスタ(PACSR)

プログラムアドレス検出レジスタ(PADRO/PADR1)

プログラムアドレス検出レジスタ(PADRO/PADR1:Program Address Detect Register 0/1)は、各レジスタに書き込まれた値と、アドレスを比較します。一致した場合は、ADCSRの対応する割り込み許可ビットが"1"の場合、CPUに対してINT9命令の発生を要求します。

対応する割り込み許可ビットが"0"の場合は、一致していても何も行いません。

プログラムアドレス検出レジスタ			byte	byte	byte	アクセス	初期値
PADRO	1FF2H/1FF1H/1FF0H					R/W	不定
PADR1	1FF5H/1FF4H/1FF3H					R/W	不定

図 23.2-1 プログラムアドレス検出レジスタ(PADRO/PADR1)

表 23.2-1に、プログラムアドレス検出レジスタ(PADRO/PADR1)とPACSRとの対応を示します。

表 23.2-1 PADRO/PADR1レジスタとPACSRとの対応

アドレス検出レジスタ	割り込み許可ビット
PADRO	AD0E
PADR1	AD1E

プログラムアドレス検出コントロール/ステータスレジスタ(PACSR)

プログラムアドレス検出コントロールレジスタ(PACSR:Program Address detect Control Register)は、アドレス検出機能の動作を制御します。

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) (-)
初期値⇒								(0) (0) (0) (0) (0) (0) (0)

図 23.2-2 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)

- 【ビット7～4】予約ビット
- ビット7～4は、予約ビットです。PACSRを設定する時は、必ず"0"を設定してください。
- 【ビット3】AD1E(Address Detect register 1 Enable)
- AD1E は、ADR1の動作許可ビットです。
- このビットが"1"のときPADR1レジスタとアドレスの比較を行い、一致した場合にCPUはINT9命令を実行します。

**【ビット2】予約ビット**

ビット2は、予約ビットです。PACSRを設定する時は、必ず"0"を設定してください。

**【ビット1】AD0E(Address Detect register 0 Enable)**

AD0Eビットは、AD0の動作許可ビットです。

このビットが"1"のときPAD0レジスタとアドレスの比較を行い、一致した場合にCPUはINT9命令を実行します。

**【ビット0】予約ビット**

ビット0は、予約ビットです。PACSRを設定する時は、必ず"0"を設定してください。

## 23.3 アドレス一致検出機能の動作

---

プログラムカウンタがアドレス一致検出レジスタと同じアドレスを指示している場合は、INT9命令を実行します。INT9命令ルーチン进行处理することによってアドレス一致検出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、各レジスタごとにコンペア許可ビットがあります。アドレス検出レジスタに設定された値とプログラムカウンタの値が一致して、かつコンペア許可ビットが"1"の場合、CPUはINT9命令を実行します。

### < 注意事項 >

---

アドレス検出レジスタとプログラムカウンタの値が一致した場合は、内部データバスの内容が強制的に"01H"に変更されるので、INT9命令を実行することになります。アドレス検出レジスタの内容を変更する前には、コンペア許可ビットを必ず"0"にしてください。コンペア許可ビットが"1"になっている間に変更すると、誤動作を引き起こす可能性があります。

---

## 23.4 アドレス一致検出機能の使用例

図 23.4-1にアドレス一致検出機能のシステム構成例を表 23.4-1にEEPROMメモリマップを示します。

アドレス一致検出機能のシステム構成例

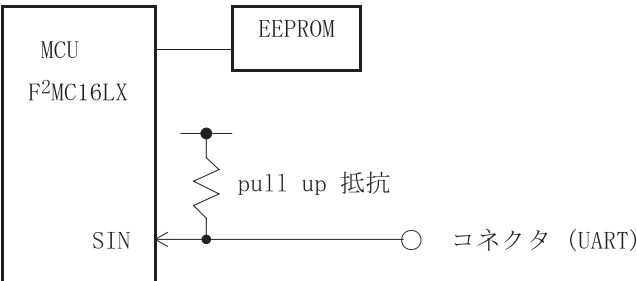


図 23.4-1 アドレス一致検出機能のシステム構成例

EEPROMメモリマップ

表 23.4-1 EEPROMメモリマップ

アドレス	意味
0000H	パッチプログラムNo.0バイト数 (0 の時プログラムミス無し)
0001H	プログラムアドレスNo.0 bit7-0
0002H	プログラムアドレスNo.0 bit15-8
0003H	プログラムアドレスNo.0 bit24-16
0004H	パッチプログラムNo.1バイト数 (0 の時プログラムミス無し)
0005H	プログラムアドレスNo.1 bit7-0
0006H	プログラムアドレスNo.1 bit15-8
0007H	プログラムアドレスNo.1 bit24-16
0010H～	パッチプログラムNo.0 / 1本体

### 初期状態

EEPROMは全て"0"とします。

### パッチが必要となる場合

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EEPROMに書き込みます。

### リセットシーケンス

MCUはリセット後EEPROM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がEEPROMから読み出されRAMに書き込まれます。次に、MCUによりPADR0、またはPADR1どちらか一方でパッチアドレスをセットし、コンペア許可ビットをセットします。再配置可能はパッチプログラムが望まれる場合は、RAM領域に対してパッチ済みプログラムの最初のアドレスを書き込むことができます。この場合、INT9ルーチンがこのユーザ定義のRAM領域を参照し、パッチ済みプログラムへ向けてジャンプします。

### INT9命令

割り込みルーチンでは、スタックプログラムのカウンタの値を調べることにより、割り込みが発生するアドレスを知ることができます。割り込みの最中にスタックへ押し込まれた情報は破棄されます。

プログラムパッチ処理例

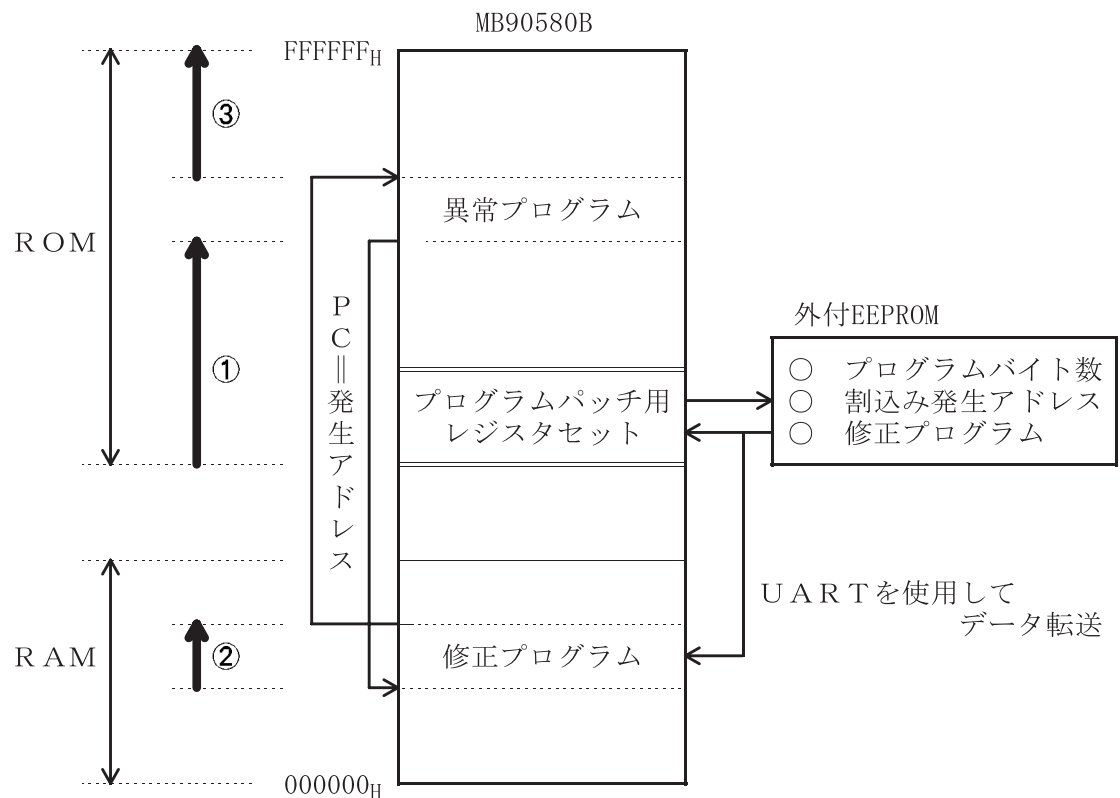


図 23.4-2 プログラムパッチ処理例

プログラムパッチ処理フロー図

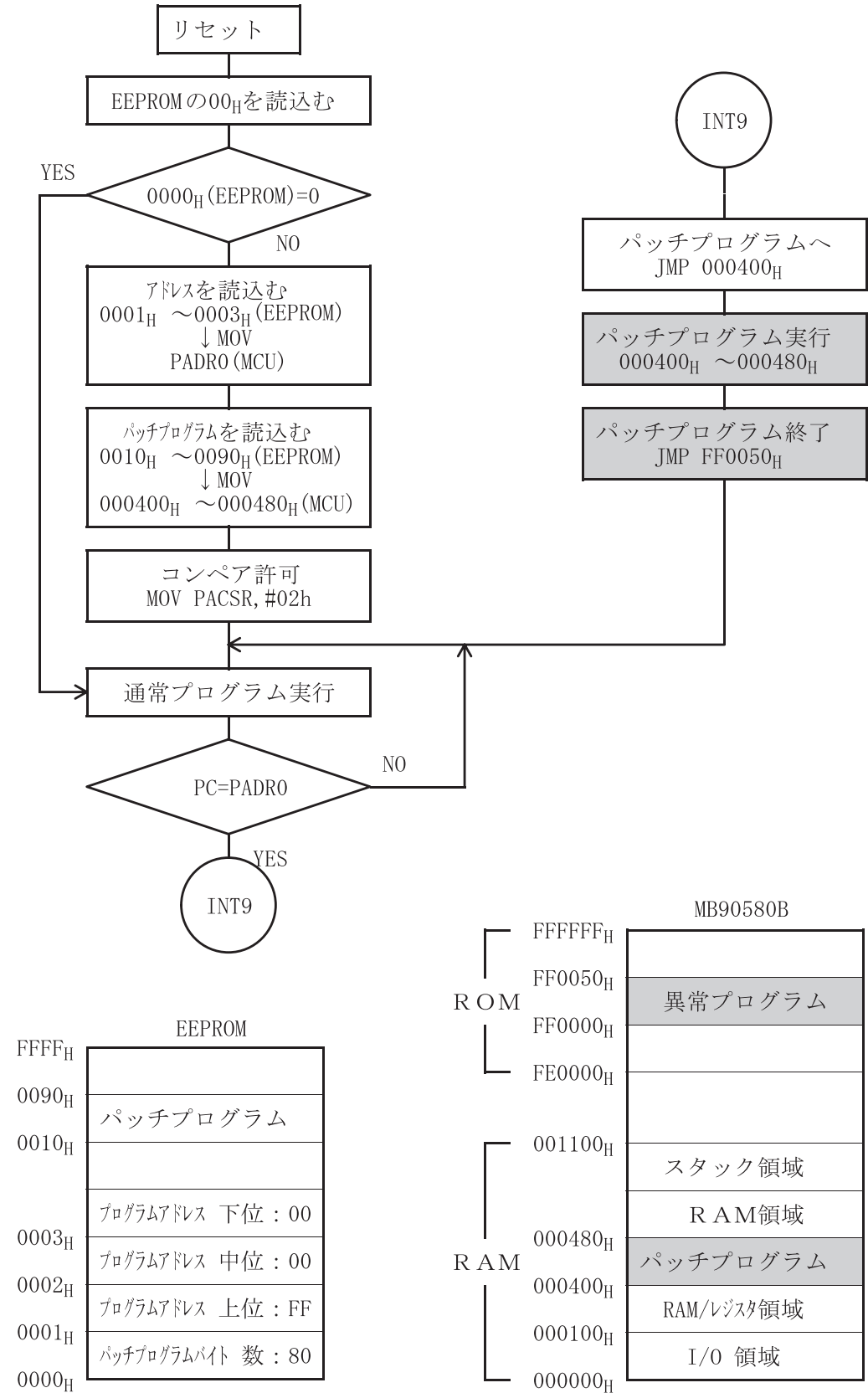


図 23.4-3 プログラムパッチ処理フロー図

## 第24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。

---

24.1 ROMミラー機能選択モジュールの概要

24.2 ROMミラー機能選択レジスタ (ROMM)



## 24.1 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バンクを00バンクでみることができます。

ROMミラー機能選択モジュールのブロックダイアグラム

F<sup>2</sup>MC-16LX バス

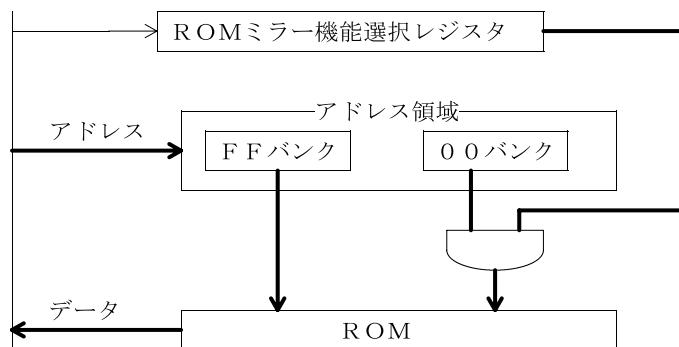


図 24.1-1 ROMミラー機能選択モジュールのブロックダイアグラム

## 24.2 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)を図 24.2-1に示します。

ROMミラー機能選択レジスタ(ROMM)

ROMミラーレジスタ	15	14	13	12	11	10	9	8	←ビットNo.
アドレス: 00006F <sub>H</sub>	—	—	—	—	—	—	—	MI	ROMM
リード/ライト 初期値	⇒ ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(W) (1)	

図 24.2-1 ROMミラー機能選択レジスタ(ROMM)

### < 注意事項 >

このレジスタをアドレス004000<sub>H</sub>～00FFFF<sub>H</sub>番地の動作中にアクセスしないでください。

#### 【ビット8】MI

MIビットでは, "1"を書込み時には00バンクでもFFバンクのROMのデータが読みだせるようになります。このビットに"0"が書込まれると, メモリマッピングのような処理は実行されません。本ビットは, 書込み専用です。

シングルチップモード時, および内ROM外バスモード時のメモリ空間を以下に示します。

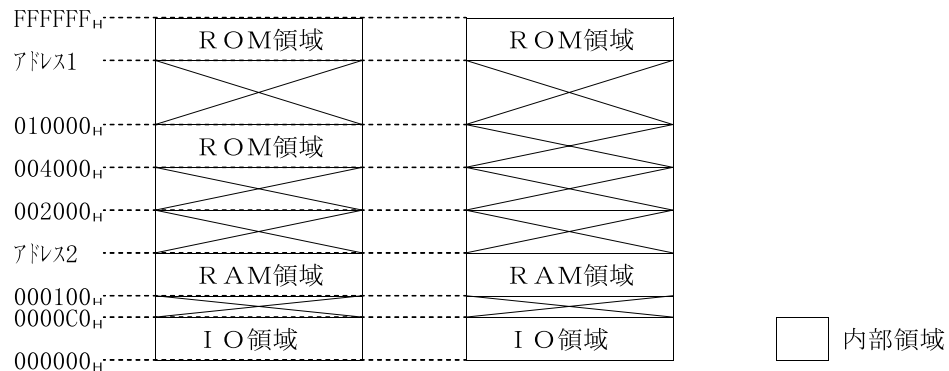
### < 注意事項 >

FF4000<sub>H</sub>～FFFFFF<sub>H</sub>から004000<sub>H</sub>～00FFFF<sub>H</sub>へミラーリングが行われるのは, ROMミラーリング機能を起動した場合に限られます。したがって, アドレスFF0000<sub>H</sub>～FF3FFF<sub>H</sub>から00バンクへのミラーリングは行われません。

表 24.2-1 メモリ空間のアドレス

	MB90583C/CA	MB90F583C/CA	MB90V580B	MB90587C/CA
アドレス1	FE0000 <sub>H</sub>	FE0000 <sub>H</sub>	-	FF0000 <sub>H</sub>
アドレス2	001900 <sub>H</sub>	001900 <sub>H</sub>	001900 <sub>H</sub>	001100 <sub>H</sub>

アドレス



MI = '1' のとき      MI = '0' のとき

図 24.2-2 シングルチップモード時メモリ空間

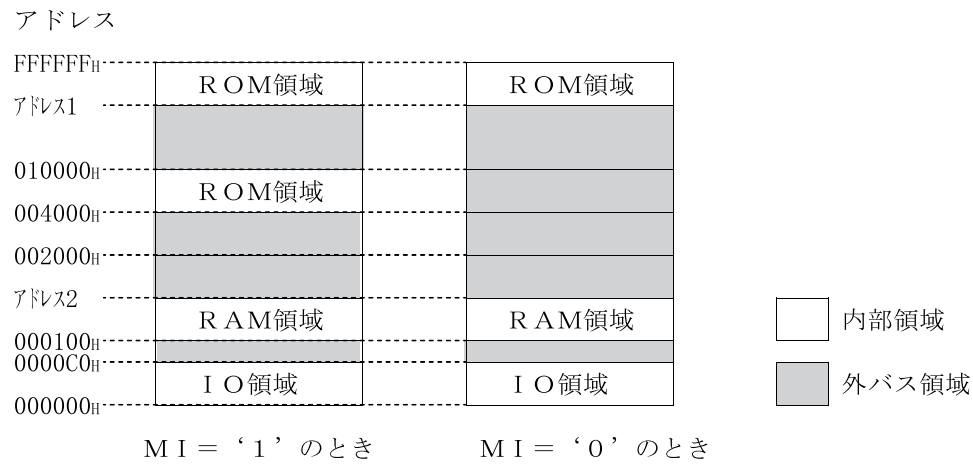


図 24.2-3 内ROM外バスモード時メモリ空間

## 第25章 1Mビットフラッシュメモリ

---

この章では、1Mビットフラッシュメモリの機能や動作について説明します。  
フラッシュメモリへのデータ書込み/消去の方法には、下記の3通りの方法があります。

1. パラレルライタ
2. シリアル専用ライタ
3. プログラム実行による書込み/消去

ここでは、"3. プログラム実行による書込み/消去"について解説します。

---

- 25.1 1Mビットフラッシュメモリの概要
- 25.2 フラッシュメモリのセクタ構成
- 25.3 フラッシュメモリコントロールステータスレジスタ(FMCS)
- 25.4 フラッシュメモリ自動アルゴリズム起動方法
- 25.5 自動アルゴリズム実行状態の確認
- 25.6 フラッシュメモリ書込み/消去の詳細説明
- 25.7 1Mビットフラッシュメモリのプログラム例

# 25.1 1Mビットフラッシュメモリの概要

1Mビットフラッシュメモリは、CPUメモリマップ上のFE<sub>H</sub>～FF<sub>H</sub>バンクに配置され、フラッシュメモリインタフェース回路の機能により、マスクROMと同様にCPUからのリードアクセスおよびプログラムアクセスが可能です。フラッシュメモリへの書き込み/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。

なお、イネーブル・セクタプロテクト等のセクタオペレーションは使用できません。

## 1Mビットフラッシュメモリの特長

- 128Kワード×8/64Kワード×16ビット(16K+512×2+7K+8K+32K+64K) セクタ構成
- 自動プログラムアルゴリズム(Embedded Alogrithm:MBM29F400TAと同様)
- 消去一時停止/消去再開機能の搭載
- データボーリング、トグルビットによる書き込み/消去完了検出
- CPU割り込みによる書き込み/消去の完了検出
- JEDEC標準型コマンドと互換
- セクタごとの消去が可能(セクタ組み合わせ自由)
- 書き込み/消去回数(最小) 10,000回

Embedded AlogrithmはAdvanced Micro Device社の商標です。

## フラッシュメモリ書き込み/消去の方法

フラッシュメモリは書き込み/消去と読み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き込み/消去動作を行う際には、フラッシュメモリ上にあるプログラムをRAMにいったんコピーし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き込み動作だけを行うことが可能となります。

## フラッシュメモリのレジスタ

### フラッシュメモリコントロールステータスレジスタ(FMCS)

ビットNo.	7	6	5	4	3	2	1	0
アドレス: 0000AEh	INTE	RDYINT	WE	RDY	予約	LPM1	予約	LPM0
リード/ライト	(R/W)	(R/W)	(R/W)	(R)	(W)	(R/W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(X)	(0)	(0)	(0)	(0)

## 25.2 フラッシュメモリのセクタ構成

図 25.2-1に，フラッシュメモリのセクタ構成を示します。

1Mビットフラッシュメモリのセクタ構成

図 25.2-1に，1Mビットフラッシュメモリのセクタ構成を示します。図中アドレスには，各セクタの上位アドレスと下位アドレスを示します。  
CPUからアクセスする場合FEバンクレジスタにSA0，FFバンクレジスタにSA1～4が配置されています。

フラッシュメモリ	CPUアドレス	ライターアドレス*
SA4 (16Kバイト)	FFFFFF <sub>H</sub>	7FFFF <sub>H</sub>
	FFC000 <sub>H</sub>	7C000 <sub>H</sub>
SA3 (8Kバイト)	FFBFFF <sub>H</sub>	7BFFF <sub>H</sub>
	FFA000 <sub>H</sub>	7A000 <sub>H</sub>
SA2 (8Kバイト)	FF9FFF <sub>H</sub>	79FFF <sub>H</sub>
	FF8000 <sub>H</sub>	78000 <sub>H</sub>
SA1 (32Kバイト)	FF7FFF <sub>H</sub>	77FFF <sub>H</sub>
	FF0000 <sub>H</sub>	70000 <sub>H</sub>
SA0 (64Kバイト)	FEFFFF <sub>H</sub>	6FFFF <sub>H</sub>
	FEC000 <sub>H</sub>	60000 <sub>H</sub>

\*：ライターアドレスとは，フラッシュメモリにパラレルライターでデータ書き込みを行う際，CPUアドレスに対して相当するアドレスです。汎用ライターを使用し書き込み/消去を行う際は，このアドレスで書き込み/消去を実行します。

図 25.2-1 1Mビットフラッシュメモリのセクタ構成

## 25.3 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き込み/消去の際に使用します。

コントロールステータスレジスタ (FMCS)

ビットNo.	7	6	5	4	3	2	1	0
アドレス: 0000AEH	INTE	RDYINT	WE	RDY	予約	LPM1	予約	LPM0
リード/ライト	(R/W)	(R/W)	(R/W)	(R)	(W)	(R/W)	(W)	(R/W)
初期値	(0)	(0)	(0)	(X)	(0)	(0)	(0)	(0)

### ビット内容

#### 【bit7】INTE(INTerrupt Enable)

フラッシュメモリの書き込み/消去の終了でCPUに割り込みを発生させるビットです。

INTEビットが"1"でかつRDYINTビットが"1"の時にCPUへ割り込みが発生します。INTEビットが"0"であれば割り込みは発生しません。

0: 書き込み/消去終了での割り込み禁止

1: 書き込み/消去終了での割り込み許可

#### 【bit6】RDYINT(ReaDY INTerrupt)

フラッシュメモリの動作状態を表すビットです。

フラッシュメモリの書き込み/消去が終わって"1"となります。フラッシュメモリ書き込み/消去後このビットが"0"の間は、フラッシュメモリへの書き込み/消去はできません。書き込み/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き込み/消去が可能になります。

"0"書き込みによって"0"にクリアされ、"1"の書き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(1.4フラッシュメモリ自動アルゴリズム参照)終了タイミングで、"1"にセットされます。リードモディファイライト(RMW)命令使用時は、必ず"1"が読めます。

0: 書き込み/消去動作実行中

1: 書き込み/消去動作終了(割り込み要求発生)

#### 【bit5】WE(Write Enable)

フラッシュメモリ領域へのライトイネーブルビットです。

このビットが"1"の時、FC～FFバンクへのコマンドシーケンス(1.4フラッシュメモリ自動アルゴリズム参照)発行後のライトは、フラッシュメモリ領域へのライトになります。本ビットが"0"の時、書き込み/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き込み/消去のコマンドを起動する際に使用します。

書き込み/消去を行わない時は、誤ってフラッシュメモリにデータを書込まないように、常に"0"に設定する事を推奨します。

0: フラッシュメモリ書き込み/消去禁止

1: フラッシュメモリ書き込み/消去許可

**【bit4】RDY(ReadDY)**

フラッシュメモリの書込み/消去許可ビットです。

このビットが"0"の間は、フラッシュメモリへの書込み/消去はできません。  
 なお、この状態でも読み出し/リセットコマンド、セクタ消去一時停止等のサスペンドコマンドは受け付けられます。

0: 書込み/消去動作実行中 (次データ書込み/消去不可)

1: 書込み/消去動作終了 (次データ書込み/消去許可)

**【bit3】予約ビット**

試験用予約ビットです。通常使用時は必ず"0"に設定してください。

**【bit1】空きビット**

通常使用時は必ず"0"に設定してください。

**【bit2,0】LPM1,LPM0(Low Power Mode)**

LPM1,LPM0ビットを使用すると、フラッシュメモリ本体の消費電力を制御できます。しかし、CPUからフラッシュメモリへのアクセスタイムが設定により大きく異なりますので、CPUの動作周波数により設定値を選択してください。

01: 低電力消費モード (内部動作周波数4MHz以下で動作)

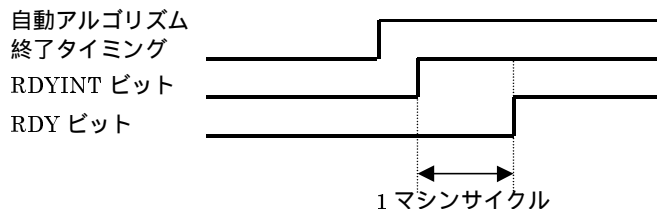
10: 低電力消費モード (内部動作周波数8MHz以下で動作)

11: 低電力消費モード (内部動作周波数12.58MHz以下で動作)

00: 通常消費電力モード (内部動作周波数16MHz以下で動作)

**< 注意事項 >**

RDYINTビットとRDYビットは同時には変化しません。どちらか片方のビットで判定するようプログラムを作成してください。





## 25.4 フラッシュメモリ自動アルゴリズム起動方法

フラッシュメモリの自動アルゴリズムを起動するコマンドには、読み出し/リセット、書き込み、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 25.4-1に、フラッシュメモリの書き込み/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下さい。この時の上位バイト分のデータは無視されます。

表 25.4-1 コマンドシーケンス表

コマンド シーケンス	バスサイクル アドレス	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
読み出し/ リセット	1	FxXXXX	XXF0	-	-	-	-	-	-	-	-	-	-
読み出し/ リセット	4	FxAAAA	XXAA	Fx5554	XX55	FxAAAA	XXF0	RA	RD	-	-	-	-
書き込み 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(xx80H)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、セクタ消去中の消去一時停止後、消去開始											

(注)

- ・表中のアドレスFxは、FF、FEを意味します。それぞれの操作時にはアクセス対象バンクの値としてください。
  - ・表中のアドレスは、CPUメモリマップ上の値です。アドレスおよびデータはすべて16進数表記しています。ただし"X"は任意の値です。
  - ・RA:読み出しアドレス
  - ・PA:書き込みアドレス、偶数アドレスのみ指定可
  - ・SA:セクタアドレス、1.2 1Mビットフラッシュメモリセクタ構成を参照願います。
  - ・RD:読み出しデータ
  - ・PD:書き込みデータ、ワードデータのみ指定可
- \*: 2種類の読み出し/リセットコマンドは、どちらもフラッシュメモリを読み出しモードにリセットすることができます。

## 25.5 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事を知らせるハードウェアを持ちます。この自動アルゴリズムは、下記のハードウェアシーケンスによって内蔵フラッシュメモリの動作状態の確認ができます。

### ハードウェア・シーケンス・フラグ

ハードウェア・シーケンス・フラグは、DQ7, DQ6, DQ5, DQ3の4ビットの出力で構成されます。それぞれがデータボーリングフラグ(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(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1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 25.5-1にハードウェア・シーケンス・フラグのビット割り当てを示します。

表 25.5-1 ハードウェア・シーケンス・フラグのビット割り当て

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ビットを確認する事で、書込みが終了しているかを知る事ができ、書込み/消去終了後は、読み出し/リセット状態に戻ります。実際にプログラムを作成する際には、いずれかのフラグで自動書込み/消去終了を確認後に、データの読み出し等の次処理を行って下さい。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も、ハードウェア・シーケンス・フラグによって確認することができます。事項に各ハードウェア・シーケンス・フラグの説明を示します。表 25.5-2に、ハードウェア・シーケンス・フラグ機能の一覧を示します。

表 25.5-2 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

### 25.5.1 データポーリングフラグ(DQ7)

データポーリングフラグ(DQ7)は、自動アルゴリズム実行が進行中もしくは終了状態であることをデータポーリング機能によって知らせるためのフラグです。

データポーリングフラグ(DQ7)

表 25.5-3と表 25.5-4に、データポーリングフラグの状態遷移を示します。

表 25.5-3 データポーリングフラグの状態遷移（正常動作時の状態変化）

動作状態	書き込み動作 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 25.5-4 データポーリングフラグの状態遷移(異常動作時の状態変化)

動作状態	書き込み動作	チップ・セクタ消去動作
DQ7	$\overline{\text{DQ7}}$	0

#### 書き込み動作時

自動書き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。自動書き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み出し値のビット7を出力します。

#### チップ/セクタ消去動作時

チップ消去/セクタ消去アルゴリズム実行中に、セクタ消去時は現在消去しているセクタから、チップ消去時はアドレスの指し示す番地に関係なく、リードアクセスするとフラッシュメモリは"0"を出力します。同様に終了時には"1"を出力します。

#### セクタ消去一時停止時

セクタ消去一時停止時にリードアクセスすると、フラッシュメモリはアドレスの指し示す番地が消去中のセクタであれば"1"を出力し、消去中のセクタでなければアドレスの指し示す番地の読み出し値のビット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。

#### < 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。データの読み出しは、データポーリングフラグ(DQ7)の終了を受けて他のビットの出力が可能となります。このため自動アルゴリズム終了後のデータの読み出しは、データポーリング終了を確認したリードアクセスの次に行うようにしてください。

25.5.2 トグルビットフラグ(DQ6)

トグルビットフラグ(DQ6)はデータポーリングフラグ(DQ7)と同様に、主に自動アルゴリズム実行が進行中、もしくは終了状態であることをトグルビット機能によって知らせるためのフラグです。

トグルビットフラグ(DQ6)

表 25.5-5と表 25.5-6に、トグルビットフラグの状態遷移を示します。

表 25.5-5 トグルビットフラグの状態遷移(正常動作時の状態変化)

動作状態	書込み動作完了	チップセクタ消去完了	セクタ消去ウェイト開始	セクタ消去一時停止 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 25.5-6 トグルビットフラグの状態遷移(異常動作時の状態変化)

動作状態	書込み動作	チップ・セクタ消去動作
DQ6	Toggle	Toggle

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。自動書き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み出し値のビット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み出し値のビット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2μsのトグル動作をした後、データを書き換えることなくトグル動作を終わります。  
消去の際、選択された全てのセクタが書き換え保護されている場合トグルビットは約100μsのトグル動作をし、その後データを書き換えしないで読み出し/リセット状態に戻ります。

### 25.5.3 タイミングリミット超過フラグ(DQ5)

タイミングリミット超過フラグ(DQ5)は、自動アルゴリズムの実行がフラッシュメモリ内部に規定された時間(内部パルス回数)を越えてしまったことを知らせるフラグです。

タイミングリミット超過フラグ(DQ5)

表 25.5-7と表 25.5-8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 25.5-7 タイミングリミット超過フラグの状態遷移（正常動作時の状態変化）

動作状態	書き込み動作完了	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 25.5-8 タイミングリミット超過フラグの状態遷移（異常動作時の状態変化）

動作状態	書き込み動作	チップ・セクタ消去動作
DQ5	1	1

#### 書き込み/チップ・セクタ消去時

書き込みまたは、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(書き込み/消去に要する時間)内であれば"0"を、規定時間を越えてしまっている場合は"1"を出力します。これは、自動アルゴリズムが実行中か終了状態にあるか無関係ですので、書き込み/消去が成功したか失敗したかの判定が可能です。すなわち、このフラグが"1"を出力したとき、データポーリング機能もしくはトグルビット機能により自動アルゴリズムがまだ実行中であれば、書き込みが失敗していると判断することができます。

例えば、"0"が書き込まれているフラッシュメモリアドレスに"1"を書き込もうとするとフェイルが発生します。この場合フラッシュメモリはロックされ、自動アルゴリズムは終了しません。したがって、データポーリングフラグ(DQ7)から有効なデータが出力されません。またトグルビットフラグ(DQ6)はトグル動作を止めず、タイムリミットを越え、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事を表しています。この状態が発生した時は、リセットコマンドを実行してください。

25.5.4 セクタ消去タイマフラグ(DQ3)

セクタ消去タイマフラグ(DQ3)は、セクタ消去コマンド起動後セクタ消去ウェイト期間中であるか否かを知らせるフラグです。

セクタ消去タイマフラグ(DQ3)

表 25.5-9と表 25.5-10に、セクタ消去タイマフラグの状態遷移を示します。

表 25.5-9 セクタ消去タイマフラグの状態遷移（正常動作時の状態変化）

動作状態	書き込み動作完了	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 25.5-10 セクタ消去タイマフラグの状態遷移（異常動作時の状態変化）

動作状態	書き込み動作	チップ・セクタ消去動作
DQ3	0	1

セクタ消去動作時

セクタ消去コマンド起動後にリードアクセスすると、フラッシュメモリはコマンドを発行したセクタのアドレス信号の指し示す番地によらず、セクタ消去ウェイト期間中であれば"0"を、セクタ消去ウェイト期間を越えてしまっている場合は"1"を出力します。

データボ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。この事を確認するために、引き続くセクタ消去コードのライトに先立ちこのフラグの状態をチェックする事を推奨します。もし2回目の状態チェックで"1"であったなら追加セクタの消去コードは受け付けられて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み出し値のビット3(DATA:3)を出力します。

## 25.6 フラッシュメモリ書込み/消去の詳細説明

自動アルゴリズムを起動するコマンドを発行し、フラッシュメモリに読み出し/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み出し/リセット、書込み、チップ消去、セクタ消去、セクタ消去一時停止、消去再開の動作がコマンドシーケンス(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)のバスへのライトサイクルを行うことで自動アルゴリズムを実行することが可能です。それぞれのバスへのライトサイクルは必ず続けて行う必要があります。また、自動アルゴリズムはデータ・ポーリング機能等で終了時を知ることができます。正常終了後は読み出し/リセット状態に戻ります。

各動作について、次項より下記の順に示します。

- 読み出し/リセット状態にする
- データを書き込む
- 全データを消去する(チップ全消去)
- 任意のデータを消去する(セクタ消去)
- セクタ消去を一時停止する
- セクタ消去を再開する

## 25.6.1 フラッシュメモリを読み出し/リセット状態にする

---

読み出し/リセットコマンドを発行し、フラッシュメモリを読み出し/リセット状態にする手順について説明します。

---

### フラッシュメモリを読み出し/リセット状態にする

フラッシュメモリを読み出し/リセット状態にするには、コマンドシーケンス表(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)の読み出し/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。

読み出し/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。

読み出し/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み出し/リセット状態になります。読み出し/リセット状態は他のコマンドの入力待ち状態です。

読み出し/リセット状態では通常のリードアクセスでデータを読み出せます。マスクROMと同様にCPUからのプログラムアクセスが可能です。通常読み出しでのデータ読み出しにこのコマンドは必要ありません、何らかの理由でコマンドが正常に終了しなかった時など、自動アルゴリズムを初期化する場合に主にこのコマンドを使用します。



## 25.6.2 フラッシュメモリヘデータを書き込む

書き込みコマンドを発行し、フラッシュメモリヘデータを書き込む手順について説明します。

### フラッシュメモリヘデータを書き込む

フラッシュメモリの、データ書き込み自動アルゴリズムを起動するには、コマンドシーケンス表(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)の書き込みコマンドを、フラッシュメモリ内の対象セクタに続けて送ることによって実行可能です。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き込みが開始します。

#### アドレス指定方法

書き込みデータサイクルの中で指定する書き込みアドレスは、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事ができません。つまり、偶数アドレスへのワードデータ単位での書き込みが必要となります。

書き込みはどのようなアドレスの順番でも、また、セクタの境界を越えても可能ですが、1回の書き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。

#### データ書き込み上の注意

書き込みによって、データ"0"をデータ"1"に戻す事はできません。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、書き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"が書き込まれた様に見えるかのどちらかとなります。しかし、読み出し/リセット状態でデータを読み出すとデータは"0"のままです。消去動作のみが"0"データを"1"にすることができます。

自動書き込み実行中は全てのコマンドが無視されます。書き込み中にハードウェアリセットが起動されると、書き込んでいるアドレスのデータは保証されませんので注意が必要です。

### フラッシュメモリ書き込み手順

図2にフラッシュメモリ書き込みの手順の例を示します。ハードウェア・シーケンス・フラグ(1.5自動アルゴリズム実行状態の確認参照)を用いる事でフラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。ここでは、書き込み終了の確認にデータポーリングフラグ(DQ7)を用いています。

フラグチェックのために読み込むデータは最後に書き込みを行ったアドレスからの読み込みとなります。

データポーリングフラグ(DQ7)は、タイミングリミット超過フラグ(DQ5)と同時に変わるので、たとえタイミングリミット超過フラグ(DQ5)が"1"であってもデータポーリングフラグビット(DQ7)は再チェックする必要があります。

トグルビットフラグ(DQ6)でも同様にタイミングリミット超過フラグビット(DQ5)が"1"に変わると同時にトグル動作を止めるので、トグルビットフラグ(DQ6)を再チェックする必要があります。

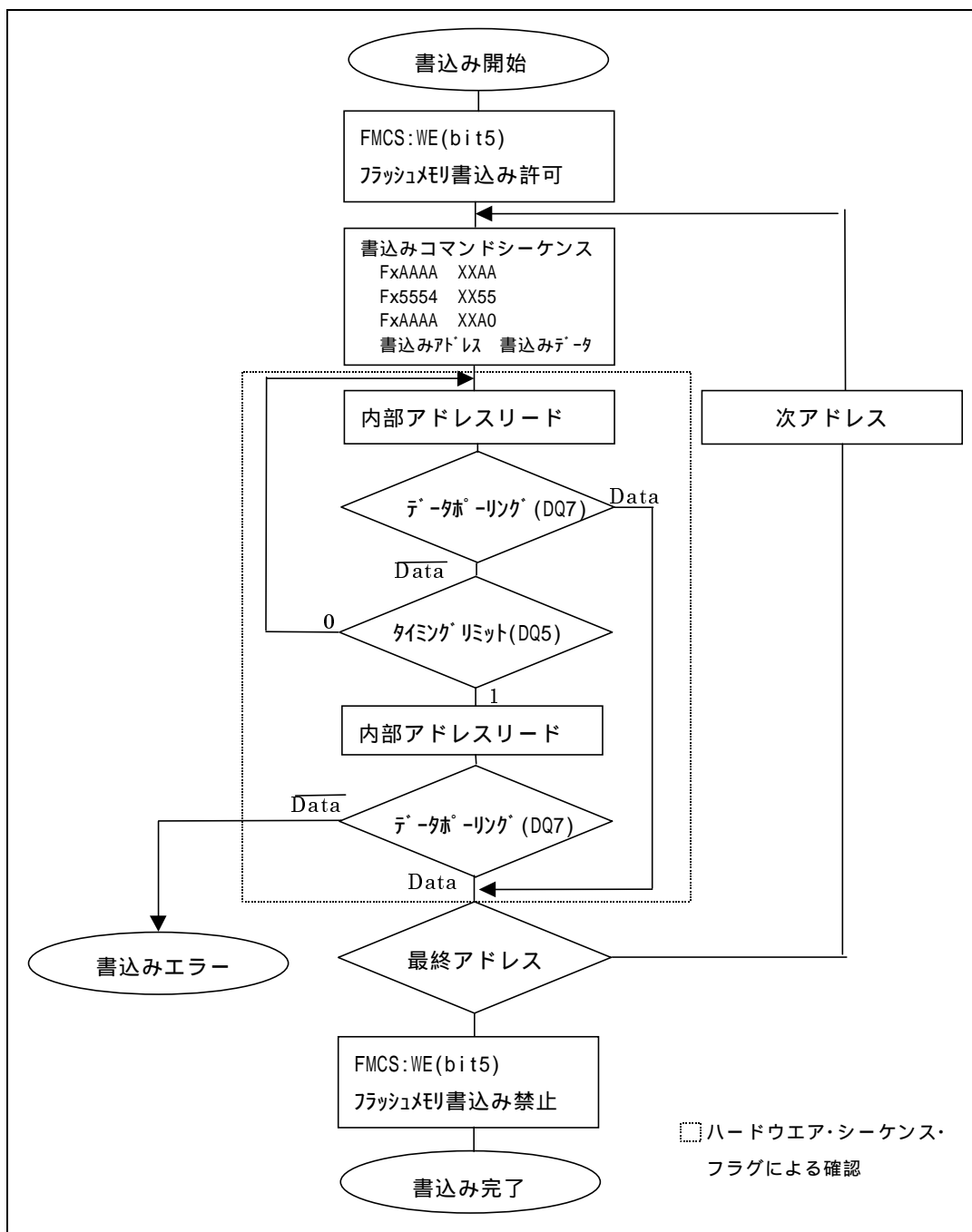


図 25.6-1 フラッシュメモリ書き込み手順の例

### 25.6.3 フラッシュメモリの全データを消去する(チップ消去)

---

チップ消去コマンドを発行し、フラッシュメモリの全データを消去する手順について説明します。

---

#### フラッシュメモリのデータを消去する(チップ消去)

フラッシュメモリから全てのデータを消去するには、コマンドシーケンス表(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)のチップ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。

チップ消去コマンドは6回のバス動作で行われます。6サイクル目のライトが完了した時点でチップ消去動作が開始します。チップ消去では、消去前にユーザがフラッシュメモリに書込みを行う必要はありません。自動消去アルゴリズム実行中には、フラッシュメモリは自動的に全てのセルを消去する前に"0"を書き込んで検証します。

## 25.6.4 フラッシュメモリの任意のデータを消去する(セクタ消去)

セクタ消去コマンドを発行し、フラッシュメモリの任意のデータを消去する(セクタ消去)手順について説明します。セクタごとの消去が可能であり、複数のセクタを同時に指定することも可能です。

### フラッシュメモリの任意のデータを消去する(セクタ消去)フラッシュメモリ

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(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)のセクタ消去コマンドを、フラッシュメモリ内の対象セクタに続けて送ることで実行可能です。

#### 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 $\mu$ sのセクタ消去ウェイトが開始します。複数のセクタ消去を行う場合は、上記の処理に引き続き消去する目的のセクタ内のアドレスに消去コード(30H)をライトします。

#### 複数のセクタを指定する時の注意

最後のセクタ消去コードのライトから50 $\mu$ sのセクタ消去ウェイト期間終了により消去が開始します。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 $\mu$ s以内に入力する必要があるため、それ以降では受け付けられないことがあります。引き続き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。なおこの時、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(「25.5 自動アルゴリズム実行状態の確認」を参照)を用いる事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3にフラッシュメモリセクタ消去の手順の例を示します。ここでは、消去終了の確認にトグルビットフラグ(DQ6)を用いています。

フラグチェックのために読み込むデータは、消去しようとしているセクタからの読み込みとなりますので、注意が必要です。

トグルビットフラグ(DQ6)は、タイミングリミット超過フラグ(DQ5)が"1"に変わるのと同時にトグル動作を止めるので、たとえタイミングリミット超過フラグ(DQ5)が"1"であってもトグルビットフラグ(DQ6)は再チェックする必要があります。

データポーリングフラグ(DQ7)でも同様に、タイミングリミット超過フラグ(DQ5)と同時に変わるので、データポーリングフラグを再チェックする必要があります。

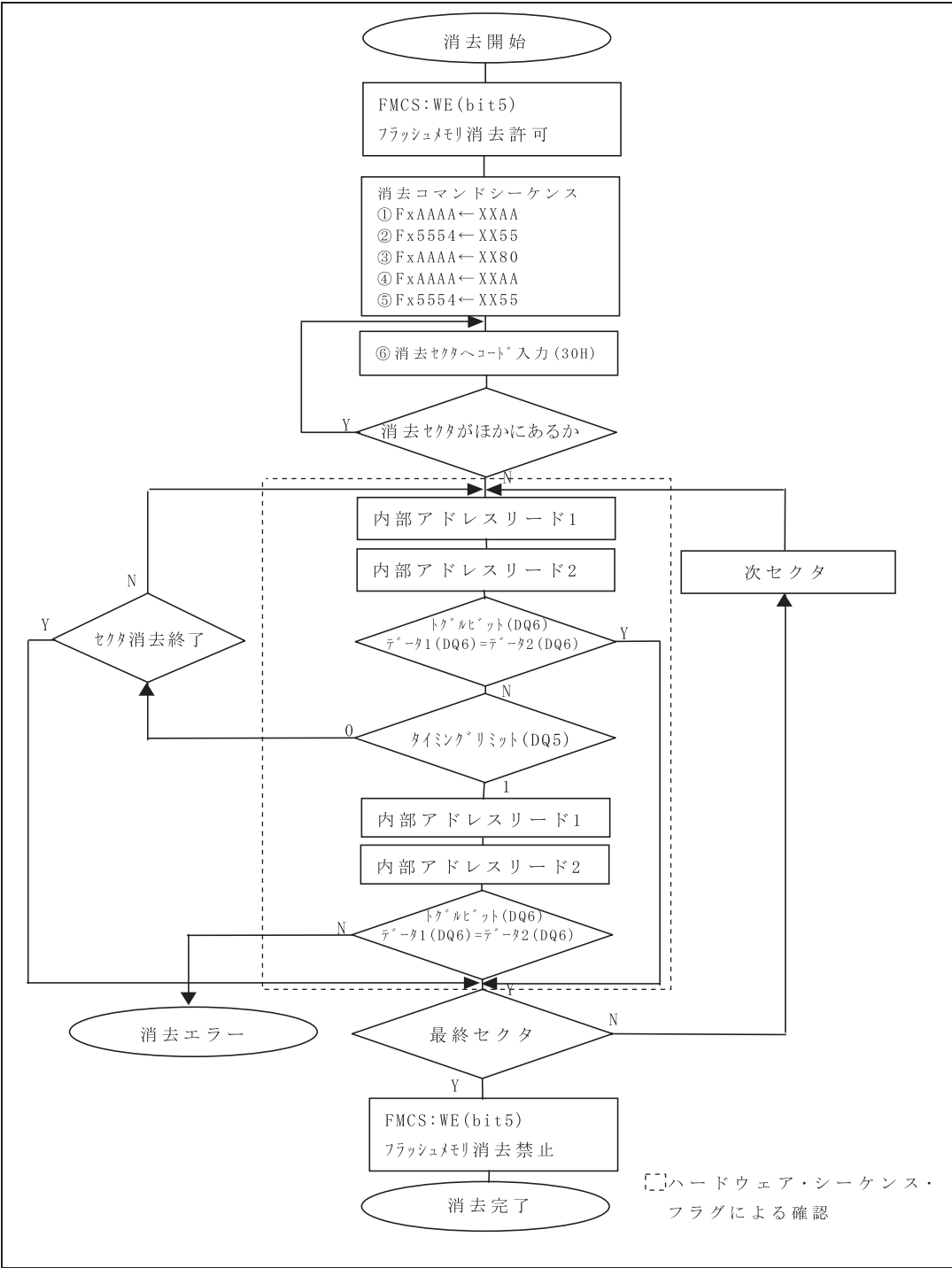


図 25.6-2 フラッシュメモリのセクタ消去手順の例

### 25.6.5 フラッシュメモリのセクタ消去を一時停止する

セクタ消去一時停止コマンドを発行し、フラッシュメモリセクタ消去の一時停止を行う手順について説明します。消去中でないセクタから、データを読みだすことが可能です。

#### フラッシュメモリのセクタ消去を一時停止する

フラッシュメモリのセクタ消去を一時停止するには、コマンドシーケンス表(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)のセクタ消去一時停止コマンドを、フラッシュメモリ内に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)のライトを行う事で実施されますが、このときアドレスはフラッシュメモリ内の任意のアドレスを指すようにします。消去一時停止での再度の消去一時停止コマンドは無視されます。

セクタ消去ウェイト期間中に、セクタ消去一時停止コマンドが入力されると、直ちにセクタ消去ウェイトを終了し、消去動作を中断して消去停止状態になります。セクタ消去ウェイト期間後のセクタ消去動作中に消去一時停止コマンドが入力されると、最大15 $\mu$ s後、消去一時停止状態に入ります。

## 25.6.6 フラッシュメモリのセクタ消去を再開する

---

セクタ消去再開コマンドを発行し、一時停止したフラッシュメモリセクタ消去を再開する手順について説明します。

---

### フラッシュメモリのセクタ消去を再開する

一時停止したセクタ消去を再開させるには、コマンドシーケンス表(「25.4 フラッシュメモリ自動アルゴリズム起動方法」表 25.4-1を参照)のセクタ消去再開コマンドを、フラッシュメモリ内に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(30H)のライトを行うことで実施されますが、このときのアドレスはフラッシュメモリ領域内の任意のアドレスを指すようにします。

なお、セクタ消去中のセクタ消去再開コマンドの発行は無視されます。

## 25.7 1Mビットフラッシュメモリのプログラム例

ここでは、1Mビットフラッシュメモリのプログラム例を掲載します。

1Mビットフラッシュメモリのプログラム例

```

NAME    FLASHWE
TITLE   FLASHWE
;-----
;1Mbit-FLASH サンプルプログラム
;
;1: FLASHにあるプログラム(アドレス FFBC00H セクタ SA4)をRAM(アドレス 000700H)に
;   転送する。
;2: RAM上でプログラムを実行する。
;3: PDR1の値をFLASH(アドレス FE0000Hセクタ SA0)に書込む。
;4: 書込んだ値(アドレス FE0000H セクタ SA0)を読み出しPDR2に出力する。
;5: 書込んだセクタ(SA0)を消去する。
;6: 消去データ確認の出力
;   条件
;       ・ RAM転送バイト数: 100H(256B)
;       ・ 書込み、消去の終了判定
;           DQ5(タイミングリミット超過フラグ)での判定
;           DQ6(トグルビットフラグ)での判定
;           RDY(FMCS)での判定
;       ・ エラー時の処理
;           P00 ~ P07にHiを出力する
;           リセットコマンド発行
;-----
;
RESOUS   IOSEG   ABS=00           ;"RESOUS" I/Oセグメントの定義
          ORG     0000H
PDR0     RB      1
PDR1     RB      1
PDR2     RB      1
PDR3     RB      1
          ORG     0010H
DDR0     RB      1
DDR1     RB      1
DDR2     RB      1
DDR3     RB      1
          ORG     00A1H
CKSCR    RB      1
          ORG     00AEH
FMCS     RB      1
          ORG     006FH
ROMM     RB      1
RESOUS   ENDS
;
SSTA     SSEG
          RW      0127H
STA_T    RW      1
SSTA     ENDS
;
DATA     DSEG    ABS=0FFH        ;FLASHコマンドアドレス
          ORG     5554H
COMADR2  RW      1
```



```

        ORG      0AAAAH
COMADR1 RW      1
DATA     ENDS
;//////////
;メインプログラム(SA1)
;//////////
CODE     CSEG
START:
;//////////
;初期化
;//////////
MOV      CKSCR,#0BAH      ;3通倍に設定
MOV      RP,#0
MOV      A,#!STA_T
MOV      SSB,A
MOVW     A,#STA_T
MOVW     SP,A
MOV      ROMM,#00H        ;ミラーOFF
MOV      PDR0,#00H        ;エラー確認用
MOV      DDR0,#0FFH
MOV      PDR1,#00H        ;データ入力用ポート
MOV      DDR1,#00H
MOV      PDR2,#00H        ;データ出力用ポート
MOV      DDR2,#0FFH
;//////////
;RAM(700H番地)に"FLASH書き込み消去プログラム(FFBC00H)"を転送する
;//////////
MOVW     A,#0700H          ;転送先RAM領域
MOVW     A,#0BC00H          ;転送元アドレス(プログラムのある位置)
MOVW     RW0,#100H          ;転送するバイト数
MOVS     ADB,PCB            ;FFBC00Hから000700Hへ100H転送
CALLP    000700H            ;転送したプログラムのあるアドレスへジャンプ
;//////////
;データ出力
;//////////
OUT      MOV      A,#0FEH
        MOV      ADB,A
        MOVW     RW2,#0000H
        MOVW     A,@RW2+00
        MOV      PDR2,A
END      JMP      *
CODE     ENDS
;//////////
;FLASH書き込み消去プログラム(SA4)
;//////////
RAMPRG   CSEG      ABS=0FFH
        ORG      0BC00H
;
; 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,#0FEH            ;ADB変更1
MOV      ADB,A              ;書き込みモード指定アドレス用バンク指定
MOV      PDR3,#00H          ;スイッチ初期化

```

```

MOV    DDR3,#00H
;
;
WAIT1  BBC    PDR3:0,WAIT1      ;PDR3:0 Hiで書込みスタート
;
;
;////////////////////////////////////
; 書込み(SA0)
;////////////////////////////////////
MOV    A,PDR1
MOVW   @RW0+00,A              ;RAMにPDR1データを確保
MOV    FMCS,#20H              ;書込みモード設定
MOVW   ADB:COMADR1,#00AAH     ;フラッシュ書込みコマンド1
MOVW   ADB:COMADR2,#0055H     ;フラッシュ書込みコマンド2
MOVW   ADB:COMADR1,#00A0H     ;フラッシュ書込みコマンド3
;
;
MOVW   A,@RW0+00              ;入力データ(RW0)をフラッシュメモリ(RW2)
;                               に書込む
MOVW   @RW2+00,A
WRITE  ;待ち時間チェック
;
;////////////////////////////////////
; タイムリミット超過チェック-フラグが立ちトグル動作中であるときERROR
;////////////////////////////////////
MOVW   A,@RW2+00
AND    A,#20H                 ;DQ5タイムリミットチェック
BZ     NTOW                   ;タイムリミットオーバー
MOVW   A,@RW2+00              ;AH
MOVW   A,@RW2+00              ;AL
XORW   A                     ;AH ALのXOR(値が違えば1)
AND    A,#40H                 ;DQ6トグルビットは違っているか
BNZ    ERROR                  ;違えばERRORへ
;
;////////////////////////////////////
; 書込み終了チェック(FMCS-RDY)
;////////////////////////////////////
NTOW   MOVW   A,FMCS
AND    A,#10H                 ;FMCS RDYビット(4bit)抽出
BZ     WRITE                  ;書込み終了か?
MOV    FMCS,#00H              ;書き込みモード解除
;////////////////////////////////////
; 書込みデータ出力
;////////////////////////////////////
MOVW   RW2,#0000H             ;書込みデータ出力
MOVW   A,@RW2+00
MOV    PDR2,A
;
;
WAIT2  BBC    PDR3:1,WAIT2      ;PDR3:1 Hiでセクタ消去スタート
;
;
;////////////////////////////////////
; セクタ消去(SA0)
;////////////////////////////////////
MOV    @RW2+00,#0000H         ;アドレス初期化
MOV    FMCS,#20H              ;消去モード設定
MOVW   ADB:COMADR1,#00AAH     ;フラッシュ消去コマンド1
MOVW   ADB:COMADR2,#0055H     ;フラッシュ消去コマンド2
MOVW   ADB:COMADR1,#0080H     ;フラッシュ消去コマンド3
MOVW   ADB:COMADR1,#00AAH     ;フラッシュ消去コマンド4
MOVW   ADB:COMADR2,#0055H     ;フラッシュ消去コマンド5
MOV    @RW2+00,#0030H         ;消すセクタに消去コマンド発行6

```

```

ELS      ; 待ち時間チェック
;
; //////////////////////////////////////
;      タイムリミット超過チェック- フラグが立ちトグル動作中であるときERROR
;      //////////////////////////////////////
;
MOVW     A, @RW2+00
AND      A, #20H           ;DQ5タイムリミットチェック
BZ       NTOE              ;タイムリミットオーバー
MOVW     A, @RW2+00        ;AH 書き込み動作中は, DQ6から
MOVW     A, @RW2+00        ;AL  リードごとHi Lowが交互出力される
XORW     A                 ;AHとALのXOR(DQ6の値が違えば1 書き込み動作
;                          ;中である)
;
AND      A, #40H           ;DQ6トグルビットはHiか
BNZ      ERROR             ;HiならERRORへ
;
; //////////////////////////////////////
;      消去終了チェック (FMCS-RDY)
;      //////////////////////////////////////
;
NTOE     MOVW     A, FMCS    ;
AND      A, #10H           ;FMCS RDYビット(4bit)抽出
BZ       ELS               ;セクタ消去終了か?
MOV      FMCS, #00H        ;FLASH消去モード解除
RETP     ;メインプログラムに戻る
; //////////////////////////////////////
; エラー
; //////////////////////////////////////
ERROR    MOV      ADB:COMADR1, #0F0H ;リセットコマンド(読み出しが可能になる)
MOV      FMCS, #00H          ;FLASHモード解除
MOV      PDRO, #0FFH        ;エラー処理の確認リセットコマンド
RETP     ;メインプログラムに戻る
RAMPRG   ENDS
; //////////////////////////////////////
VECT     CSEG      ABS=0FFH
ORG      OFFDCH
DSL      START
DB       00H
VECT     ENDS
;
;      END      START

```

## 第26章 MB90F583C/CAシリアル書き込み接続例

---

この章では、株式会社ワイ・ディ・シー製フラッシュマイコンプログラムを用いた場合の、シリアル書き込み接続例について説明しています。

---

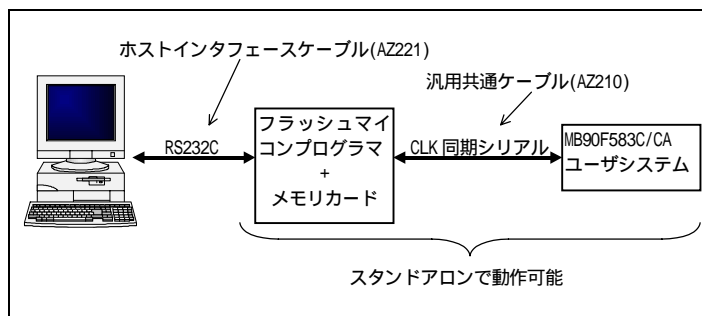
- 26.1 MB90F583C/CAシリアル書き込み接続の基本構成
- 26.2 シリアル書き込み接続例（ユーザ電源使用時）
- 26.3 シリアル書き込み接続例（ライターから電源供給時）
- 26.4 フラッシュマイコンプログラムとの最小限の接続例（ユーザ電源使用時）
- 26.5 フラッシュマイコンプログラムとの最小限の接続例（ライターから電源供給時）

## 26.1 MB90F583C/CAシリアル書き込み接続の基本構成

MB90F583C/CAでは、フラッシュROMのシリアルオンボード書き込み(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。

### MB90F583C/CAシリアル書き込み接続の基本構成

富士通標準シリアルオンボード書き込みには、株式会社ワイ・ディ・シー製フラッシュマイコンプログラマを使用します。



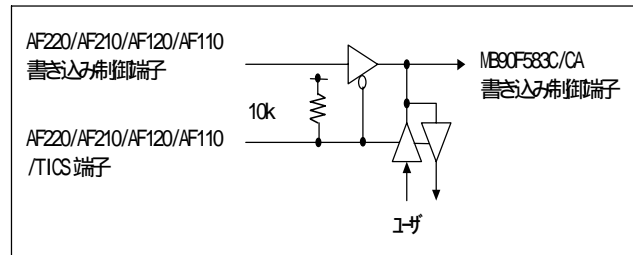
### < 注意事項 >

フラッシュマイコンプログラマ(AF220/AF210/AF120/AF110)の機能・操作方法および接続用汎用共通ケーブル(AZ210)、コネクタにつきましては、横河デジタルコンピュータ株式会社殿にお問い合わせください。

表 26.1-1 富士通標準シリアルオンボード書き込みに使用する端子

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、書き込みモードに制御します。
X0, X1	発振用端子	書き込みモード時に、CPU内部動作クロックはPLLクロック1通倍となっております。従いまして、発振クロック周波数が、内部動作クロックとなりますので、シリアル書き換え時に使用する発振子は3MHz～16MHzとなります。
P00, P01	書き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	UART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	
SCKO	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	書き込み電圧(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ようにしてください。
VSS	GND端子	フラッシュマイコンプログラマのGNDと共通にします。
HSTX	ハードウェアスタンバイ端子	シリアル書き込みモード中は"H"レベルを入力してください。

P00, SIN0, S0T0, SCK0端子をユーザシステムでも使用する場合には, 下図の制御回路が必要となります。フラッシュマイコンプログラムの /TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます。



26.1項～26.5項に, 以下のシリアル書き込み接続例を示しますのでご参照ください。

- シリアル書き込み接続例 (ユーザ電源使用時)
- シリアル書き込み接続例 (ライターから電源供給時)
- フラッシュマイコンプログラムの最小限の接続例 (ユーザ電源使用時)
- フラッシュマイコンプログラムの最小限の接続例 (ライターから電源供給時)

#### 発振クロック周波数とシリアルクロック入力周波数について

MB90F583C/CAの入力可能なシリアルクロック周波数は, 以下の計算式によって求めることができます。したがって, ご使用の発振クロック周波数に合わせて, シリアルクロック入力周波数をフラッシュマイコンプログラムの設定により変更してください。

$$\text{入力可能なシリアルクロック周波数} = 0.125 \times \text{発振クロック周波数}$$

表 26.1-2 最大シリアルクロック周波数

発振クロック周波数	マイコンの入力可能な最大シリアルクロック周波数	AF220/AF210/AF120/AF110の設定可能な最大シリアルクロック周波数	AF200の設定可能な最大シリアルクロック周波数
4MHz時	500kHz	500kHz	500kHz
8MHz時	1MHz	850kHz	500kHz
16MHz時	2MHz	1.25MHz	500kHz

フラッシュマイコンプログラムのシステム構成（株式会社ワイ・ディ・シー製）

表 26.1-3 フラッシュマイコンプログラムのシステム構成（株式会社ワイ・ディ・シー製）

型 格		機 能
本体	AF220/AC4P	イーサネットインタフェース内蔵モデル/100V～220V電源アダプタ
	AF210/AC4P	スタンダードモデル/100V～220V電源アダプタ
	AF120/AC4P	単キー イーサネットインタフェース内蔵モデル/100V～220V電源アダプタ
	AF110/AC4P	単キーモデル/100V～220V電源アダプタ
AZ221		ライター専用 PC/AT用RS232Cケーブル
AZ210		標準ターゲットプローブ（a）長さ：1m
FF201		富士通製F2MC-16LXフラッシュマイコン用コントロールモジュール
/P2		2MB PC Card（Option）FLASHメモリ容量～128KB対応
/P4		4MB PC Card（Option）FLASHメモリ容量～512KB対応

< 注意事項 >

AF200フラッシュマイコンプログラマは終息製品ですが，コントロールモジュールFF201を用いることにより使用できます。シリアル書き込み接続も，次項に示します接続例にて対応できます。

## 26.2 シリアル書き込み接続例（ユーザ電源使用時）

図 26.2-1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<sub>B</sub>

### シリアル書き込み接続例（ユーザ電源使用時）

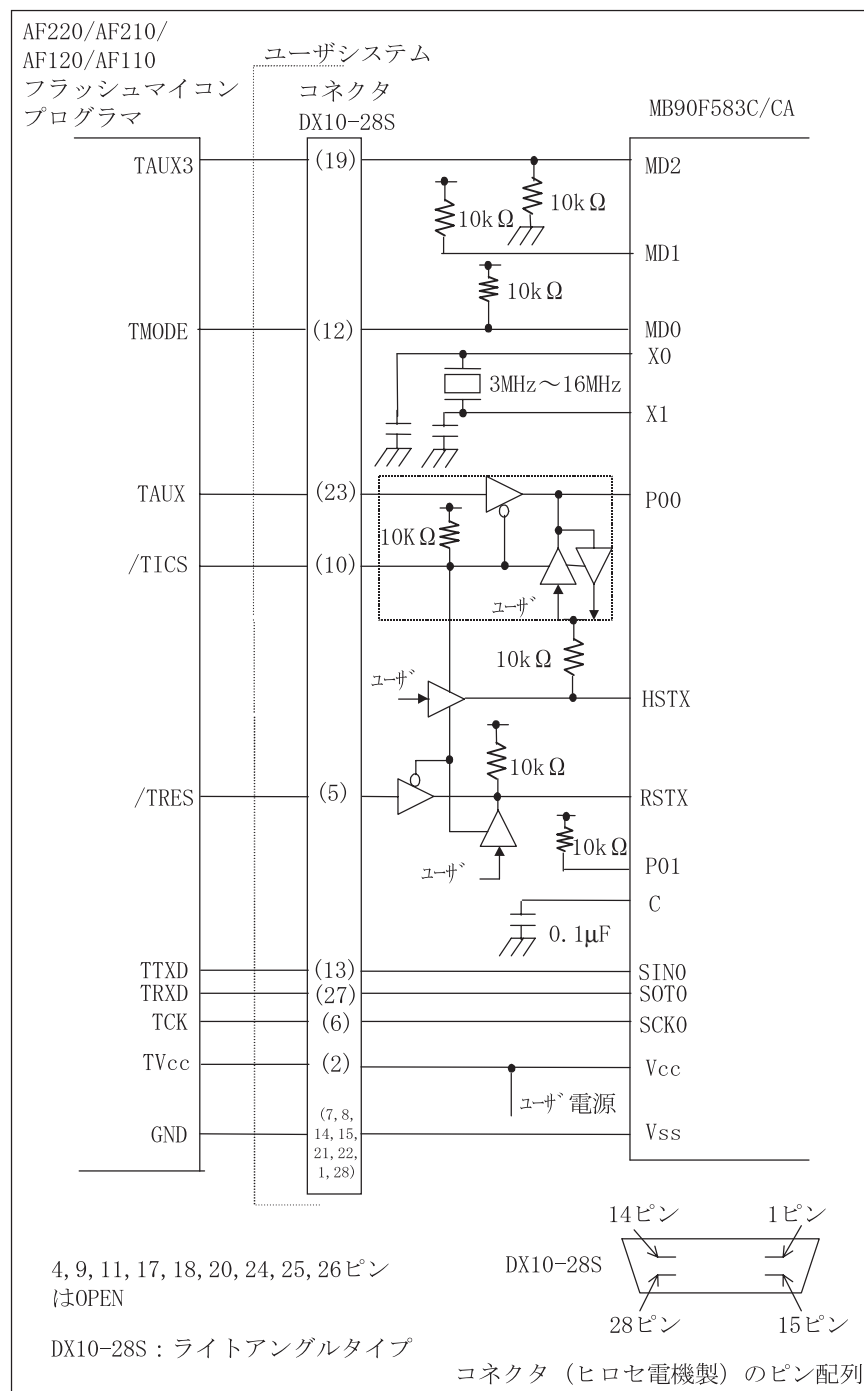
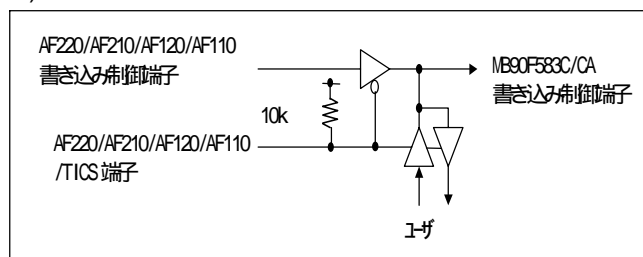


図 26.2-1 シリアル書き込み接続例（ユーザ電源使用時）



- SINO, SOTO, SCKO端子をユーザシステムでも使用する場合には, P00と同様下図の制御回路が必要となります(フラッシュマイコンプログラムの/TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。

## 26.3 シリアル書き込み接続例（ライターから電源供給時）

図 26.3-1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<sub>B</sub>

シリアル書き込み接続例（ライターから電源供給時）

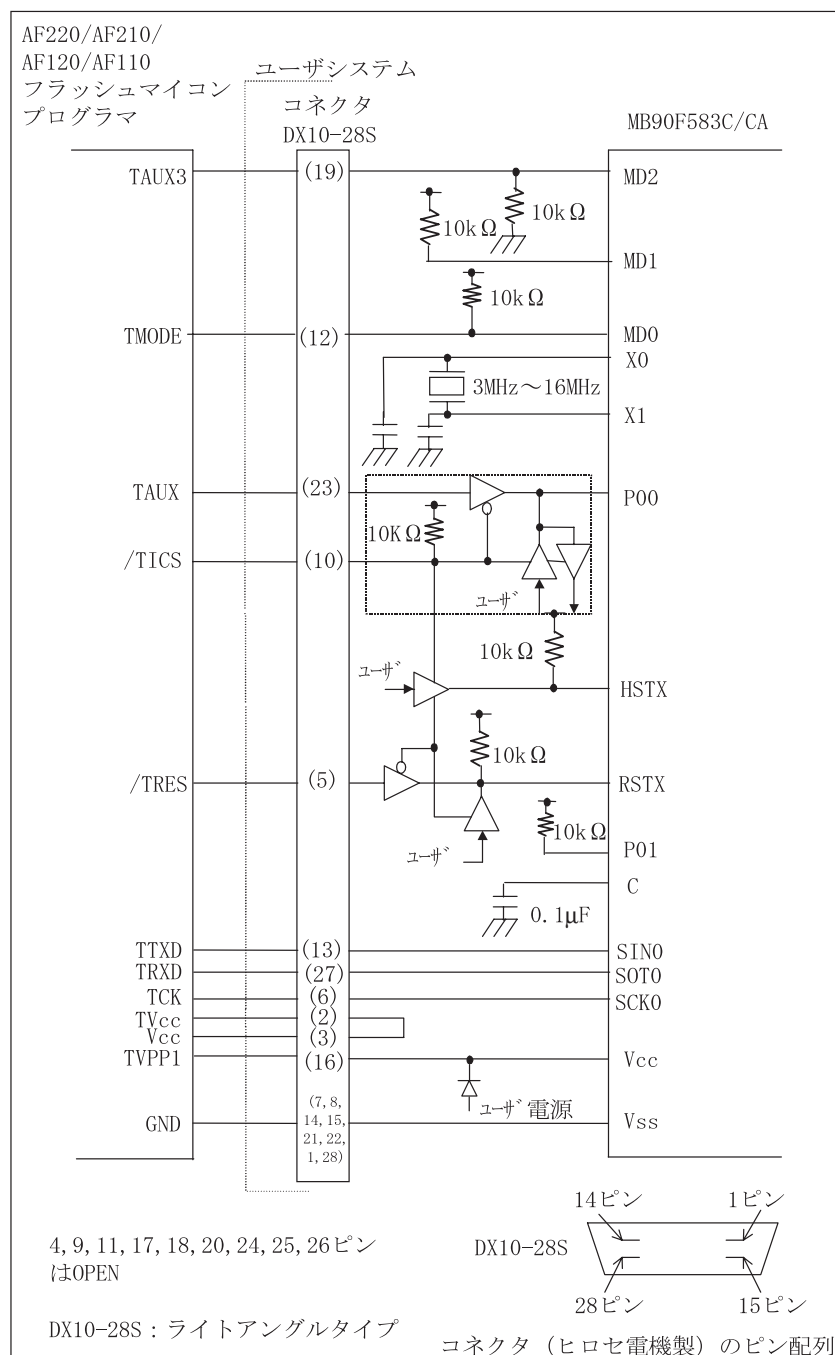
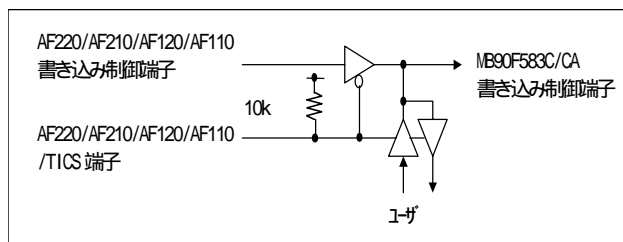


図 26.3-1 シリアル書き込み接続例（ライターから電源供給時）

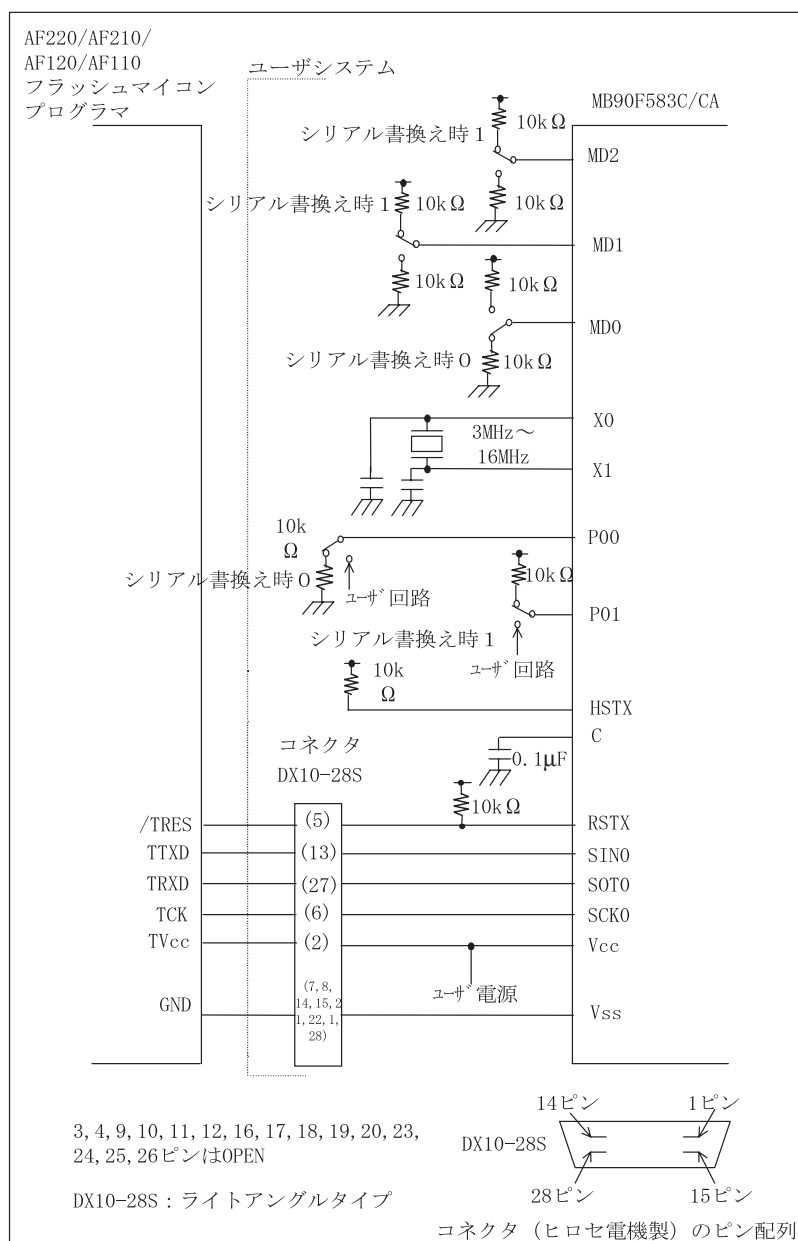
- SIN0, S0T0, SCK0端子をユーザシステムでも使用する場合には, P00と同様下図の制御回路が必要となります(フラッシュマイコンプログラムの/TICS信号により, シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続は, ユーザ電源がOFFの状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給する時は, ユーザ電源と短絡しないでください。

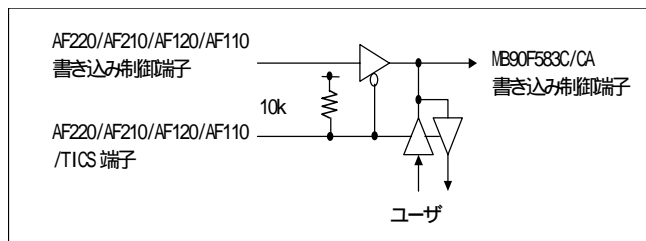
シリアル書き換えモード: MD2, MD1, MD0=110<sub>B</sub>

フラッシュメモリ書き込み時に、各端子を下記のように設定して頂ければ、MD2、MD1、MD0、P00とフラッシュマイコンプログラマとの接続は必要ありません。



411

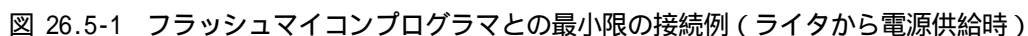
- SIN0, S0T0, SCK0端子をユーザシステムでも使用する場合には，下図の制御回路が必要となります(フラッシュマイコンプログラムの /TICS信号により，シリアル書き込み中はユーザ回路を切り離すことができます)。



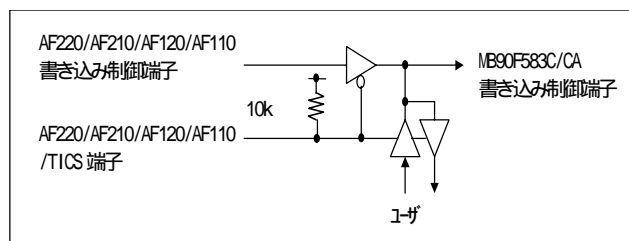
- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。

シリアル書き換えモード: MD2, MD1, MD0=110<sub>B</sub>

フラッシュメモリ書き込み時に各端子を下記のように設定して頂ければ、MD2、MD1、MD0、P00とフラッシュマイコンプログラマとの接続は必要ありません。



- SIN0, S0T0, SCK0端子をユーザシステムでも使用する場合には，下図の制御回路が必要となります(フラッシュマイコンプログラムの /TICS信号により，シリアル書き込み中はユーザ回路を切り離すことができます)。



- AF220/AF210/AF120/AF110との接続はユーザ電源がOFFの状態で行ってください。
- 書き込み電源をAF220/AF210/AF120/AF110から供給する時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

本マイクロコントローラの各リソースのレジスタは、以下のよう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 <sub>8</sub>
01 H	ポート1データレジスタ	PDR1	R/W	ポート1	XXXXXXXX <sub>8</sub>
02 H	ポート2データレジスタ	PDR2	R/W	ポート2	XXXXXXXX <sub>8</sub>
03 H	ポート3データレジスタ	PDR3	R/W	ポート3	XXXXXXXX <sub>8</sub>
04 H	ポート4データレジスタ	PDR4	R/W	ポート4	XXXXXXXX <sub>8</sub>
05 H	ポート5データレジスタ	PDR5	R/W	ポート5	11111111 <sub>8</sub>
06 H	ポート6データレジスタ	PDR6	R/W	ポート6	--XXXXX <sub>8</sub>
07 H	ポート7データレジスタ	PDR7	R/W	ポート7	---XXXX- <sub>8</sub>
08 H	ポート8データレジスタ	PDR8	R/W	ポート8	XXXXXXXX <sub>8</sub>
09 H	ポート9データレジスタ	PDR9	R/W	ポート9	XXXXXXXX <sub>8</sub>
0A H	ポートAデータレジスタ	PDRA	R/W	ポートA	-----XXX <sub>8</sub>
0B H ~ 0F H	使用禁止				
10 H	ポート0方向レジスタ	DDR0	R/W	ポート0	00000000 <sub>8</sub>
11 H	ポート1方向レジスタ	DDR1	R/W	ポート1	00000000 <sub>8</sub>
12 H	ポート2方向レジスタ	DDR2	R/W	ポート2	00000000 <sub>8</sub>
13 H	ポート3方向レジスタ	DDR3	R/W	ポート3	00000000 <sub>8</sub>
14 H	ポート4方向レジスタ	DDR4	R/W	ポート4	00000000 <sub>8</sub>
15 H	ポート5方向レジスタ	DDR5	R/W	ポート5	00000000 <sub>8</sub>
16 H	ポート6方向レジスタ	DDR6	R/W	ポート6	--000000 <sub>8</sub>
17 H	ポート7方向レジスタ	DDR7	R/W	ポート7	---0000- <sub>8</sub>
18 H	ポート8方向レジスタ	DDR8	R/W	ポート8	00000000 <sub>8</sub>
19 H	ポート9方向レジスタ	DDR9	R/W	ポート9	00000000 <sub>8</sub>
1A H	ポートA方向レジスタ	DDRA	R/W	ポートA	-----000 <sub>8</sub>
1B H	ポート4出力端子レジスタ	ODR4	R/W	ポート4	00000000 <sub>8</sub>
1C H	ポート5アナログ入力許可レジスタ	ADER	R/W	ポート5, A/D	11111111 <sub>8</sub>
1D H ~ 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 <sub>8</sub>
21 H	シリアルコントロールレジスタ0	SCR0	R/W		00000100 <sub>8</sub>
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 <sub>8</sub>
23 H	シリアルステータスレジスタ0	SSR0	R/W	UART1	00001-00 <sub>8</sub>
24 H	シリアルモードレジスタ1	SMR1	R/W		00000000 <sub>8</sub>
25 H	シリアルコントロールレジスタ1	SCR1	R/W		00000100 <sub>8</sub>
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 <sub>8</sub>
27 H	シリアルステータスレジスタ1	SSR1	R/W		00001-00 <sub>8</sub>

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
28 H	シリアルモードレジスタ2	SMR2	R/W	UART2	00000000 <sub>B</sub>
29 H	シリアルコントロールレジスタ2	SCR2	R/W		00000100 <sub>B</sub>
2A H	シリアルインプット/シリアルアウトプットレジスタ2	SIDR2/ SODR2	R/W		XXXXXXXX <sub>B</sub>
2B H	シリアルステータスレジスタ2	SSR2	R/W		00001-00 <sub>B</sub>
2C H	クロック分周コントロールレジスタ0	CDCR0	R/W	通信プロトコル0	0---1111 <sub>B</sub>
2D H	使用禁止				
2E H	クロック分周コントロールレジスタ1	CDCR1	R/W	通信プロトコル1	0---1111 <sub>B</sub>
2F H	使用禁止				
30 H	割り込み/DTP許可レジスタ	ENIR	R/W	DTP / 外部割り込み	00000000 <sub>B</sub>
31 H	割り込み/DTP要因レジスタ	EIRR	R/W		XXXXXXXX <sub>B</sub>
32 H	要求レベル設定レジスタ下位	ELVR	R/W		00000000 <sub>B</sub>
33 H	要求レベル設定レジスタ上位				00000000 <sub>B</sub>
34 H	クロック分周コントロールレジスタ2	CDCR2	R/W	通信プロトコル2	0---1111 <sub>B</sub>
35 H	使用禁止				
36 H	コントロールステータスレジスタ下位	ADCS1	R/W	A/Dコンバータ	00000000 <sub>B</sub>
37 H	コントロールステータスレジスタ上位	ADCS2	R/W		00000000 <sub>B</sub>
38 H	データレジスタ下位	ADCR1	R		XXXXXXXX <sub>B</sub>
39 H	データレジスタ上位	ADCR2	R/W		00001-XX <sub>B</sub>
3A H	D/Aコンバータデータレジスタ0	DAT0	R/W	D/Aコンバータ	00000000 <sub>B</sub>
3B H	D/Aコンバータデータレジスタ1	DAT1	R/W		00000000 <sub>B</sub>
3C H	D/Aコントロールレジスタ0	DACR0	R/W		-----0 <sub>B</sub>
3D H	D/Aコントロールレジスタ1	DACR1	R/W		-----0 <sub>B</sub>
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 <sub>B</sub>
3F H	使用禁止				
40 H	リロードレジスタL (ch.0)	PRLLO	R/W	8/16bitPPG 0/1	XXXXXXXX <sub>B</sub>
41 H	リロードレジスタH (ch.0)	PRLHO	R/W		XXXXXXXX <sub>B</sub>
42 H	リロードレジスタL (ch.1)	PRLLO	R/W		XXXXXXXX <sub>B</sub>
43 H	リロードレジスタH (ch.1)	PRLH1	R/W		XXXXXXXX <sub>B</sub>
44 H	PPG0動作モード制御レジスタ	PPGCO	R/W		0X000XX1 <sub>B</sub>
45 H	PPG1動作モード制御レジスタ	PPG01	R/W		0X000001 <sub>B</sub>
46 H	PPG0,1動作出力制御レジスタ	PPGOE	R/W		00000000 <sub>B</sub>
47 H	使用禁止				
48 H	タイマコントロールステータスレジスタ0 下位	TMCSR0	R/W	16bit リロードタイマ0	00000000 <sub>B</sub>
49 H	タイマコントロールステータスレジスタ0 上位				----0000 <sub>B</sub>
4A H	16ビットタイマレジスタ0 /	TMR0/ TMRLR0	R/W	16bit リロードタイマ1	XXXXXXXX <sub>B</sub>
4B H	16ビットリロードレジスタ0				XXXXXXXX <sub>B</sub>
4C H	タイマコントロールステータスレジスタ1 下位	TMCSR1	R/W	16bit リロードタイマ2	00000000 <sub>B</sub>
4D H	タイマコントロールステータスレジスタ1 上位				----0000 <sub>B</sub>
4E H	16ビットタイマレジスタ1 /	TMR1/ TMRLR1	R/W	16bit リロードタイマ2	XXXXXXXX <sub>B</sub>
4F H	16ビットリロードレジスタ1				XXXXXXXX <sub>B</sub>
50 H	タイマコントロールステータスレジスタ2下位	TMCSR2	R/W	16bit リロードタイマ2	00000000 <sub>B</sub>
51 H	タイマコントロールステータスレジスタ2上位				----0000 <sub>B</sub>
52 H	16ビットタイマレジスタ2 /	TMR2/ TMRLR2	R/W	16bit リロードタイマ2	XXXXXXXX <sub>B</sub>
53 H	16ビットリロードレジスタ2				XXXXXXXX <sub>B</sub>

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
54 H	PWCコントロールステータスレジスタ下位	PWCSR	R/W	16bit PWCタイマ	00000000 <sub>B</sub>
55 H	PWCコントロールステータスレジスタ上位				00000000 <sub>B</sub>
56 H	PWCデータバッファレジスタ下位	PWCR	R/W		XXXXXXXX <sub>B</sub>
57 H	PWCデータバッファレジスタ上位				XXXXXXXX <sub>B</sub>
58 H	分周比コントロールレジスタ	DIVR	R/W		-----00 <sub>B</sub>
59 H	使用禁止				
5A H	コンペアレジスタ ch.0	OCCP0	R/W	アウトプット コンペア (ch.0 ~ ch.1)	XXXXXXXX <sub>B</sub>
5B H					XXXXXXXX <sub>B</sub>
5C H	コンペアレジスタ ch.1	OCCP1	R/W		XXXXXXXX <sub>B</sub>
5D H				XXXXXXXX <sub>B</sub>	
5E 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	アウトプット コンペア (ch.0 ~ ch.1)	0000--00 <sub>B</sub>
5F 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 <sub>B</sub>
60 H	インプットキャプチャレジスタ0	IPCP0	R	インプット キャプチャ (ch.0 ~ ch.3)	XXXXXXXX <sub>B</sub>
61 H					XXXXXXXX <sub>B</sub>
62 H	インプットキャプチャレジスタ1	IPCP1	R		XXXXXXXX <sub>B</sub>
63 H					XXXXXXXX <sub>B</sub>
64 H	インプットキャプチャレジスタ2	IPCP2	R		XXXXXXXX <sub>B</sub>
65 H					XXXXXXXX <sub>B</sub>
66 H	インプットキャプチャレジスタ3	IPCP3	R		XXXXXXXX <sub>B</sub>
67 H					XXXXXXXX <sub>B</sub>
68 H	インプットキャプチャコントロール ステータスレジスタ ch.0,1	ICS01	R/W		00000000 <sub>B</sub>
69 H	使用禁止				
6A H	インプットキャプチャコントロール ステータスレジスタ ch.2,3	ICS23	R/W		00000000 <sub>B</sub>
6B H	使用禁止				
6C H	タイマデータレジスタ ( 下位 )	TCDTL	R/W	フリーラン タイマ	00000000 <sub>B</sub>
6D H	タイマデータレジスタ ( 上位 )	TCDTH	R/W		00000000 <sub>B</sub>
6E H	タイマコントロールステータスレジスタ	TCCS	R/W		00000000 <sub>B</sub>
6F H	ROMミラー機能選択レジスタ	ROMM	W	ROMミラー機能	-----1 <sub>B</sub>
70 H	自局アドレス設定レジスタ ( 下位 )	MAWL	R/W	IEBus コントローラ	XXXXXXXX <sub>B</sub>
71 H	自局アドレス設定レジスタ ( 上位 )	MAWH	R/W		XXXXXXXX <sub>B</sub>
72 H	スレーブアドレス設定レジスタ ( 下位 )	SAWL	R/W		XXXXXXXX <sub>B</sub>
73 H	スレーブアドレス設定レジスタ ( 上位 )	SAWH	R/W		XXXXXXXX <sub>B</sub>
74 H	電文長ビット設定レジスタ	DEWR	R/W		00000000 <sub>B</sub>
75 H	同報コントロールビット設定レジスタ	DCWR	R/W		00000000 <sub>B</sub>
76 H	コマンドレジスタ ( 下位 )	CMRL	R/W		11000000 <sub>B</sub>
77 H	コマンドレジスタ ( 上位 )	CMRH	R/W		00000000 <sub>B</sub>
78 H	ステータスレジスタ ( 下位 )	STRL	R		0011XXXX <sub>B</sub>
79 H	ステータスレジスタ ( 上位 )	STRH	R/W		00XX0000 <sub>B</sub>
7A H	ロックリードレジスタ ( 下位 )	LRRL	R		XXXXXXXX <sub>B</sub>
7B H	ロックリードレジスタ ( 上位 )	LRRH	R/W		1110XXXX <sub>B</sub>
7C H	マスタアドレスリードレジスタ ( 下位 )	MARL	R		XXXXXXXX <sub>B</sub>
7D H	マスタアドレスリードレジスタ ( 上位 )	MARH	R		1111XXXX <sub>B</sub>
7E H	電文長ビットリードレジスタ	DERR	R		XXXXXXXX <sub>B</sub>
7F H	同報コントロールビットリードレジスタ	DCRR	R		000XXXXX <sub>B</sub>

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
80 H	ライトデータバッファ	WDB	W	IEBus	XXXXXXXX <sub>B</sub>
81 H	リードデータバッファ	RDB	R	コントローラ	XXXXXXXX <sub>B</sub>
82 H	シリアルモードレジスタ3	SMR3	R/W	UART3	00000000 <sub>B</sub>
83 H	シリアルコントロールレジスタ3	SCR3	R/W		00000100 <sub>B</sub>
84 H	シリアルインプット/シリアルアウト プットレジスタ3	SIDR3/ SODR3	R/W		XXXXXXXX <sub>B</sub>
85 H	シリアルステータスレジスタ3	SSR3	R/W		00001-00 <sub>B</sub>
86 H	PWCノイズフィルタレジスタ	RNCR	R/W	PWC noisefilter	-----000 <sub>B</sub>
87 H	クロック分周制御レジスタ3	CDCR3	R/W	通信プロシケ-ラ3	0---1111 <sub>B</sub>
88 H	シリアルモードレジスタ4	SMR4	R/W	UART4	00000000 <sub>B</sub>
89 H	シリアルコントロールレジスタ4	SCR4	R/W		00000100 <sub>B</sub>
8A H	シリアルインプット/ シリアルアウトプットレジスタ4	SIDR4/ SODR4	R/W		XXXXXXXX <sub>B</sub>
8B H	シリアルステータスレジスタ4	SSR4	R/W		00001-00 <sub>B</sub>
8C H	ポート 0 入力プルアップ抵抗設定レジスタ	RDR0	R/W	ポート 0	00000000 <sub>B</sub>
8D H	ポート 1 入力プルアップ抵抗設定レジスタ	RDR1	R/W	ポート 1	00000000 <sub>B</sub>
8E H	ポート 6 入力プルアップ抵抗設定レジスタ	RDR6	R/W	ポート 2	00000000 <sub>B</sub>
8F H	クロック分周制御レジスタ4	CDCR4	R/W	通信プロシケ-ラ4	0---1111 <sub>B</sub>
90 ~ 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 <sub>B</sub>
9F H	遅延割込み要因発生/ 取消レジスタ	DIRR	R/W	遅延割込 発生モジュール	-----0 <sub>B</sub>
A0 H	低消費電力モード制御レジスタ	LPMCR	R/W	低消費電力制御 回路	0001100- <sub>B</sub>
A1 H	クロック選択レジスタ	CKSCR	R/W		11111100 <sub>B</sub>
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 <sub>B</sub>
A6 H	外部アドレス出力制御レジスタ	HACR	W		00000000 <sub>B</sub>
A7 H	バス制御信号選択レジスタ	ECSR	W		0000000- <sub>B</sub>
A8 H	ウォッチドッグ制御レジスタ	WDTC	R/W	ウォッチドッグ タイマ	XXXXX111 <sub>B</sub>
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 <sub>B</sub>
AA H	時計タイマ制御レジスタ	WTC	R/W	時計タイマ	1X000000 <sub>B</sub>
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	フラッシュイン タフェース	000X0000 <sub>B</sub>
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 <sub>B</sub>
B1 H	割込み制御レジスタ 01	ICR01	R/W		00000111 <sub>B</sub>
B2 H	割込み制御レジスタ 02	ICR02	R/W		00000111 <sub>B</sub>
B3 H	割込み制御レジスタ 03	ICR03	R/W		00000111 <sub>B</sub>
B4 H	割込み制御レジスタ 04	ICR04	R/W		00000111 <sub>B</sub>
B5 H	割込み制御レジスタ 05	ICR05	R/W		00000111 <sub>B</sub>
B6 H	割込み制御レジスタ 06	ICR06	R/W		00000111 <sub>B</sub>
B7 H	割込み制御レジスタ 07	ICR07	R/W		00000111 <sub>B</sub>
B8 H	割込み制御レジスタ 08	ICR08	R/W		00000111 <sub>B</sub>
B9 H	割込み制御レジスタ 09	ICR09	R/W		00000111 <sub>B</sub>
BA H	割込み制御レジスタ10	ICR10	R/W		00000111 <sub>B</sub>

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
BB <sub>H</sub>	割込み制御レジスタ11	ICR11	R/W	割込みコントローラ	00000111 <sub>B</sub>
BC <sub>H</sub>	割込み制御レジスタ12	ICR12	R/W		00000111 <sub>B</sub>
BD <sub>H</sub>	割込み制御レジスタ13	ICR13	R/W		00000111 <sub>B</sub>
BE <sub>H</sub>	割込み制御レジスタ14	ICR14	R/W		00000111 <sub>B</sub>
BF <sub>H</sub>	割込み制御レジスタ15	ICR15	R/W		00000111 <sub>B</sub>
00 ~ FF <sub>H</sub>	外部領域				
100 ~ # <sub>H</sub>	RAM領域				
# <sub>H</sub> ~ 1FE <sub>H</sub>	予約領域				
1FF0 <sub>H</sub>	プログラムアドレス検出レジスタ0 ( 下位 )	PADR0	R/W	アドレス一致検出機能	XXXXXXXX <sub>B</sub>
1FF1 <sub>H</sub>	プログラムアドレス検出レジスタ1 ( 中位 )		R/W		XXXXXXXX <sub>B</sub>
1FF2 <sub>H</sub>	プログラムアドレス検出レジスタ2 ( 上位 )		R/W		XXXXXXXX <sub>B</sub>
1FF3 <sub>H</sub>	プログラムアドレス検出レジスタ3 ( 下位 )	PADR1	R/W		XXXXXXXX <sub>B</sub>
1FF4 <sub>H</sub>	プログラムアドレス検出レジスタ4 ( 中位 )		R/W		XXXXXXXX <sub>B</sub>
1FF5 <sub>H</sub>	プログラムアドレス検出レジスタ5 ( 上位 )		R/W		XXXXXXXX <sub>B</sub>
1FF6 <sub>H</sub> ~ 1FFF <sub>H</sub>	予約領域				

- ・ 初期値について "0":初期値"0" / "1":初期値"1" / "X":初期値不定 / "-":初期値不定 ( なし )。
- ・ 00FF<sub>H</sub> 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 #<sub>H</sub> は品種ごとに変わります。

(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。読み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 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。

## 付録 B 命令

---

F<sup>2</sup>MC-16LX に使用している命令について説明します。

---

- B.1 命令の種類
- B.2 アドレッシング
- B.3 直接アドレッシング
- B.4 間接アドレッシング
- B.5 実行サイクル数
- B.6 実効アドレスフィールド
- B.7 命令一覧表の読み方
- B.8 F<sup>2</sup>MC-16LX 命令一覧表
- B.9 命令マップ

## B.1 命令の種類

---

F<sup>2</sup>MC-16LX には、以下に示す 351 種類の命令があります。

---

### ■ 命令の種類

- 転送系命令 (バイト) 41 命令
- 転送系命令 (ワード, ロングワード) 38 命令
- 加減算命令 (バイト, ワード, ロングワード) 42 命令
- 増減算命令 (バイト, ワード, ロングワード) 12 命令
- 比較命令 (バイト, ワード, ロングワード) 11 命令
- 符号なし乗除算命令 (ワード, ロングワード) 11 命令
- 符号付き乗除算命令 (ワード, ロングワード) 11 命令
- 論理演算命令 (バイト, ワード) 39 命令
- 論理演算命令 (ロングワード) 6 命令
- 符号反転命令 (バイト, ワード) 6 命令
- ノーマライズ命令 (ロングワード) 1 命令
- シフト命令 (バイト, ワード, ロングワード) 18 命令
- 分岐命令 50 命令 (分岐命令 1: 31 命令, 分岐命令 2: 19 命令)
- アキュムレータ操作命令 (バイト, ワード) 6 命令
- その他制御命令 (バイト, ワード, ロングワード) 28 命令
- ビット操作命令 21 命令
- スtring命令 10 命令

## B.2 アドレッシング

---

F<sup>2</sup>MC-16LX では、命令の実効アドレスフィールドまたは命令コード自体 (インプライド) でアドレス形式が決定されます。命令コード自体でアドレス形式が決定する場合は、使用する命令コードに合わせてアドレスを指定します。命令によっては、数種類のアドレス指定方式から設定できるものがあります。

---

### ■ アドレッシング

F<sup>2</sup>MC-16LX には、以下に示す 23 種類のアドレッシングがあります。

- 即値 (#imm)
- レジスタ直接
- 直接分岐アドレス (addr16)
- 物理直接分岐アドレス (addr24)
- I/O 直接 (io)
- 短縮直接アドレス (dir)
- 直接アドレス (addr16)
- I/O 直接ビットアドレス (io: bp)
- 短縮直接ビットアドレス (dir: bp)
- 直接ビットアドレス (addr16: bp)
- ベクタアドレス (#vct)
- レジスタ間接 (@RWj j=0 ~ 3)
- ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)
- ディスプレースメント付レジスタ間接 (@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)
- ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3)
- ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)
- ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7)
- プログラムカウンタ相対分岐アドレス (rel)
- レジスタリスト (rlst)
- アキュムレータ間接 (@A)
- アキュムレータ間接分岐アドレス (@A)
- 間接指定分岐アドレス (@ear)
- 間接指定分岐アドレス (@eam)



## ■ 実効アドレスフィールド

実効アドレスフィールドで指定される、アドレス形式を表 B.2-1 に示します。

表 B.2-1 実効アドレスフィールド

コード	表記			アドレス形式	デフォルトバンク
00	R0	RW0	RL0	レジスタ直接 ea は左から順に バイト ワード ロングワード の型に対応する。	なし
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	DTB
09	@RW1				DTB
0A	@RW2				ADB
0B	@RW3				SPB
0C	@RW0+			ポストインクリメント付 レジスタ間接	DTB
0D	@RW1+				DTB
0E	@RW2+				ADB
0F	@RW3+				SPB
10	@RW0+disp8			8 ビットディスプレースメント付 レジスタ間接	DTB
11	@RW1+disp8				DTB
12	@RW2+disp8				ADB
13	@RW3+disp8				SPB
14	@RW4+disp8				DTB
15	@RW5+disp8				DTB
16	@RW6+disp8				ADB
17	@RW7+disp8				SPB
18	@RW0+disp16			16 ビットディスプレースメント付 レジスタ間接	DTB
19	@RW1+disp16				DTB
1A	@RW2+disp16				ADB
1B	@RW3+disp16				SPB
1C	@RW0+RW7			インデックス付レジスタ間接	DTB
1D	@RW1+RW7			インデックス付レジスタ間接	DTB
1E	@PC+disp16			16 ビットディスプレースメント付 PC 間接	PCB
1F	addr16			直接アドレス	DTB

## B.3 直接アドレッシング

直接アドレッシングでは、オペランド値、レジスタおよびアドレスを直接指定します。

### ■ 直接アドレッシング

#### ● 即値 (#imm)

オペランドの値を直接指定します。(#imm4/#imm8/#imm16/#imm32)。

図 B.3-1 に例を示します。

図 B.3-1 即値 (#imm) 例

MOVW A, #01212H (A にオペランドの値を格納する命令)		
実行前	A	2 2 3 3 : 4 4 5 5
実行後	A	4 4 5 5 : 1 2 1 2 (命令によっては AL → AH に転送が行われる)

#### ● レジスタ直接

オペランドとして、直接レジスタを指定します。指定できるレジスタを表 B.3-1 に示します。

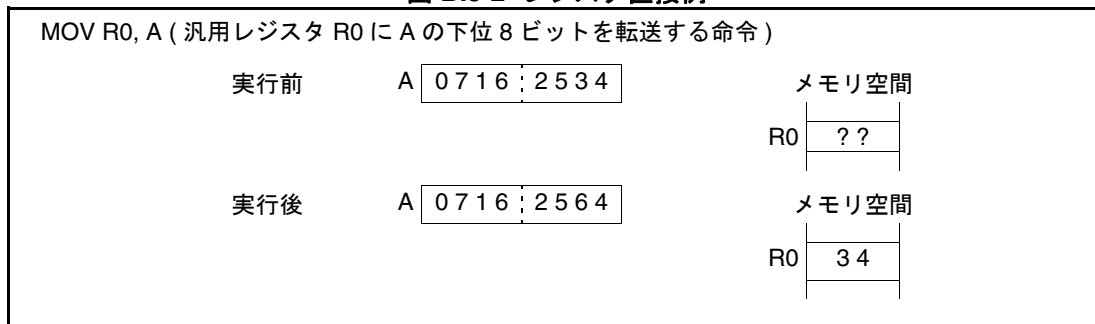
表 B.3-1 レジスタ直接

汎用レジスタ	バイト	R0, R1, R2, R3, R4, R5, R6, R7
	ワード	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
	ロングワード	RL0, RL1, RL2, RL3
専用レジスタ	アキュムレータ	A, AL
	ポインタ	SP *
	バンク	PCB, DTB, USB, SSB, ADB
	ページ	DPR
	制御	PS, CCR, RP, ILM

\*: SP は、コンディションコードレジスタ (CCR) 中の S フラグビットの値に応じてユーザスタックポインタ (USP) またはシステムスタックポインタ (SSP) のどちらか一方が選択され、使用されます。分岐系の命令ではプログラムカウンタ (PC) は命令のオペランドには記述されることなく指定されます。

図 B.3-2 に例を示します。

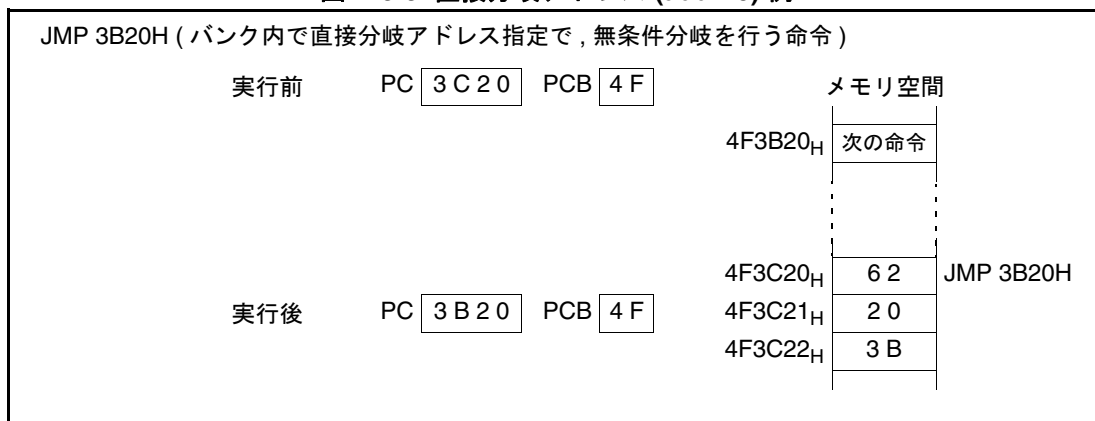
**図 B.3-2 レジスタ直接例**



● 直接分岐アドレス (addr16)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのデータ長は 16 ビットで、論理空間内での分岐先を示します。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されます。図 B.3-3 に例を示します。

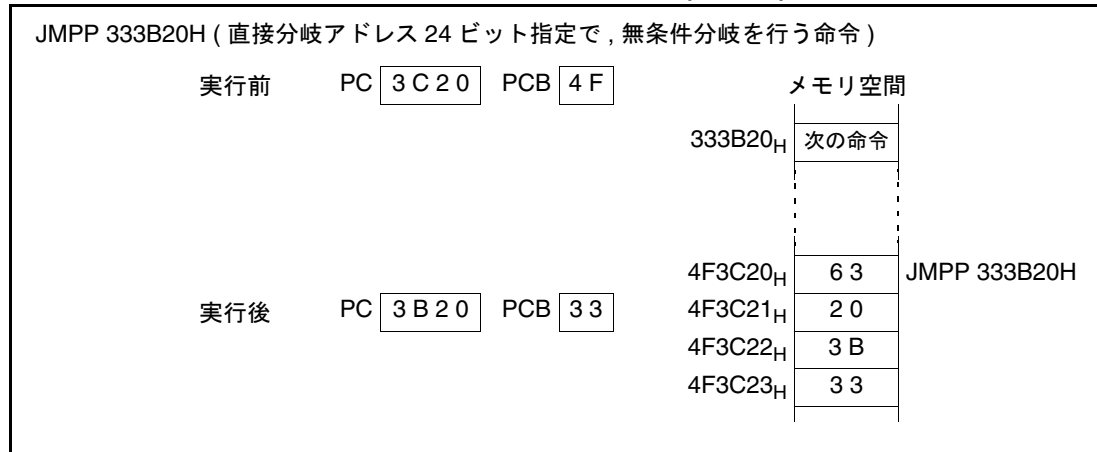
**図 B.3-3 直接分岐アドレス (addr16) 例**



● 物理直接分岐アドレス (addr24)

分岐先のアドレスをディスプレースメントで直接指定します。ディスプレースメントのデータ長は 24 ビットです。無条件分岐命令、サブルーチンコール命令およびソフトウェア割込み命令に使用します。図 B.3-4 に例を示します。

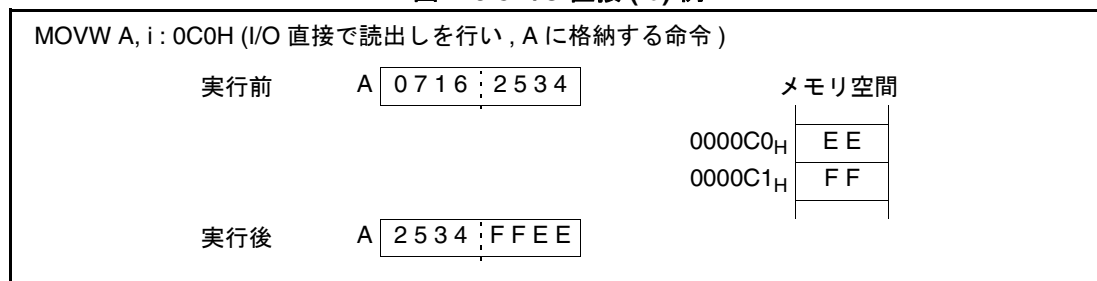
図 B.3-4 物理直接分岐アドレス (addr24) 例



● I/O 直接 (io)

オペランドのメモリアドレスを 8 ビットのディスプレ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<sub>H</sub>" ~ "0000FF<sub>H</sub>" 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。

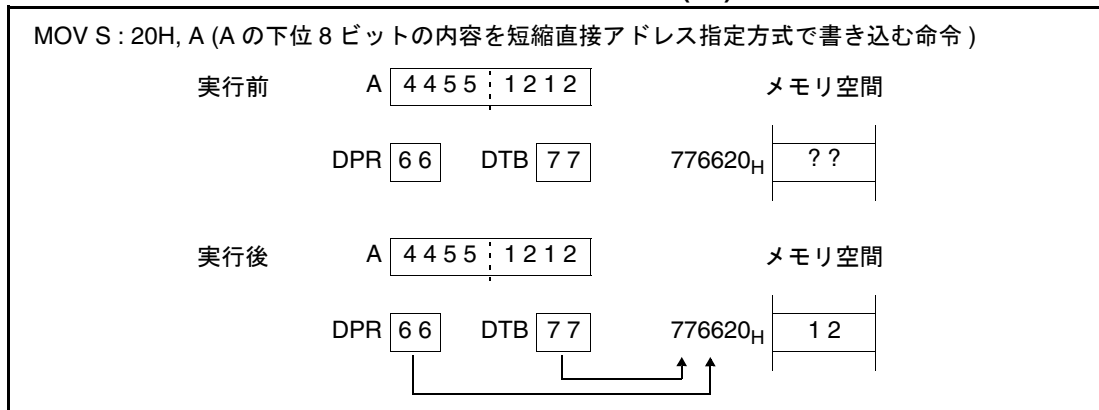
図 B.3-5 I/O 直接 (io) 例



● 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。

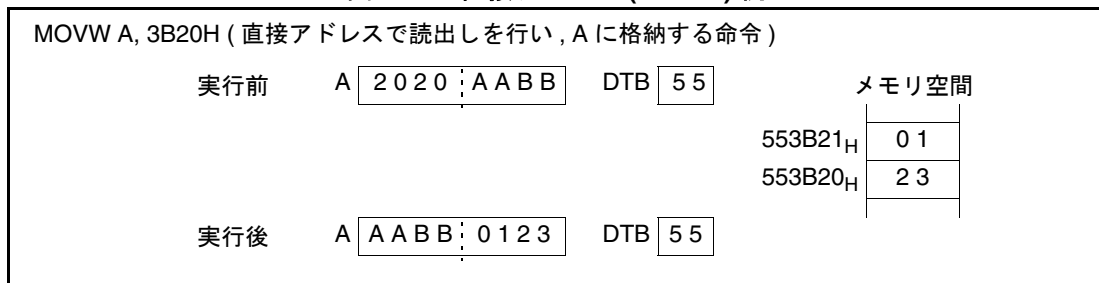
図 B.3-6 短縮直接アドレス (dir) 例



● 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。

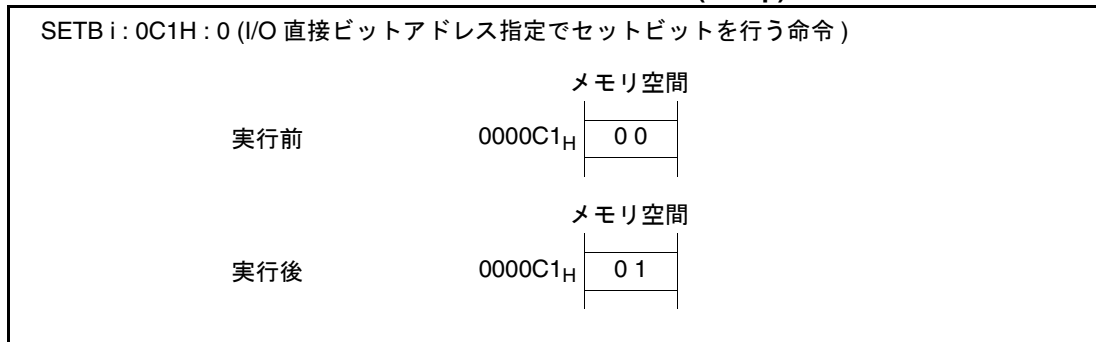
図 B.3-7 直接アドレス (addr16) 例



● I/O 直接ビットアドレス (io: bp)

物理アドレス "000000<sub>H</sub>" ~ "0000FF<sub>H</sub>" 内のビットを直接指定します。ビットの位置は ": bp" で表され, 数字の大きいほうが最上位ビット (MSB), 小さい方が最下位ビット (LSB) となります。図 B.3-8 に例を示します。

図 B.3-8 I/O 直接ビットアドレス (io: bp) 例



● 短縮直接ビットアドレス (dir: bp)

オペランドでメモリアドレスの下位 8 ビットを直接指定します。アドレスの bit15 ~ bit8 はダイレクトページレジスタ (DPR) で指定されます。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され, 数字の大きい方が最上位ビット, 小さい方が最下位ビットとなります。図 B.3-9 に例を示します。

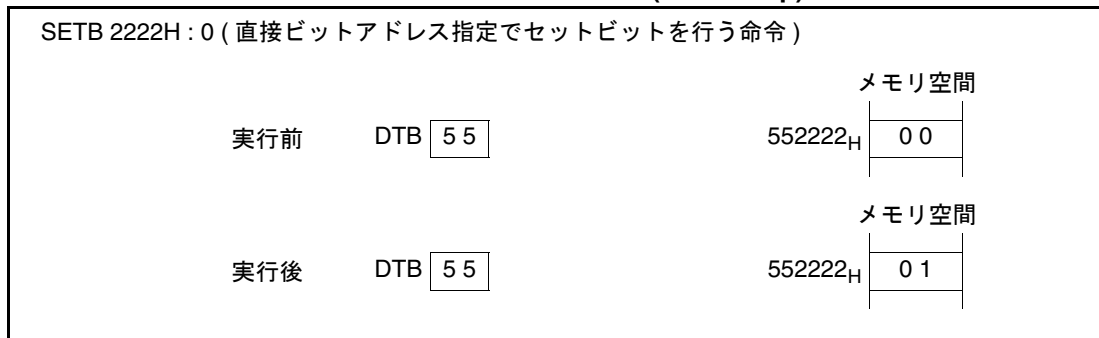
図 B.3-9 短縮直接ビットアドレス (dir: bp) 例



● 直接ビットアドレス (addr16: bp)

64K バイト内の任意のビットに対し直接指定します。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。ビットの位置は ": bp" で表され、数字の大きい方が最上位ビット、小さい方が最下位ビットとなります。図 B.3-10 に例を示します。

図 B.3-10 直接ビットアドレス (addr16: bp) 例



● ベクタアドレス (#vct)

分岐先のアドレスは指定されたベクタの内容となります。ベクタ番号のデータ長には 4 ビットと 8 ビットの 2 種類があります。サブルーチンコール命令、ソフトウェア割り込み命令に使用します。図 B.3-11 に例を示します。

図 B.3-11 ベクタアドレス (#vct) 例



表 B.3-2 CALLV ベクター一覧表

命令	ベクタアドレス L	ベクタアドレス H
CALLV #0	XXFFFE <sub>H</sub>	XXFFFF <sub>H</sub>
CALLV #1	XXFFFC <sub>H</sub>	XXFFFD <sub>H</sub>
CALLV #2	XXFFFA <sub>H</sub>	XXFFFB <sub>H</sub>
CALLV #3	XXFFF8 <sub>H</sub>	XXFFF9 <sub>H</sub>
CALLV #4	XXFFF6 <sub>H</sub>	XXFFF7 <sub>H</sub>
CALLV #5	XXFFF4 <sub>H</sub>	XXFFF5 <sub>H</sub>
CALLV #6	XXFFF2 <sub>H</sub>	XXFFF3 <sub>H</sub>
CALLV #7	XXFFF0 <sub>H</sub>	XXFFF1 <sub>H</sub>
CALLV #8	XXFFEE <sub>H</sub>	XXFFEF <sub>H</sub>
CALLV #9	XXFFEC <sub>H</sub>	XXFFED <sub>H</sub>
CALLV #10	XXFFEA <sub>H</sub>	XXFFEB <sub>H</sub>
CALLV #11	XXFFE8 <sub>H</sub>	XXFFE9 <sub>H</sub>
CALLV #12	XXFFE6 <sub>H</sub>	XXFFE7 <sub>H</sub>
CALLV #13	XXFFE4 <sub>H</sub>	XXFFE5 <sub>H</sub>
CALLV #14	XXFFE2 <sub>H</sub>	XXFFE3 <sub>H</sub>
CALLV #15	XXFFE0 <sub>H</sub>	XXFFE1 <sub>H</sub>

( 注意事項 ) XX には PCB レジスタの値が入ります。

---

#### <注意事項>

プログラムカウンタバンクレジスタ (PCB) が "FF<sub>H</sub>" の場合ベクタ領域は INT #vct8 (#0 ~ #7) のベクタ領域と共有しているため、使用する場合は注意が必要です。( 表 B.3-2 を参照してください )

---



## B.4 間接アドレッシング

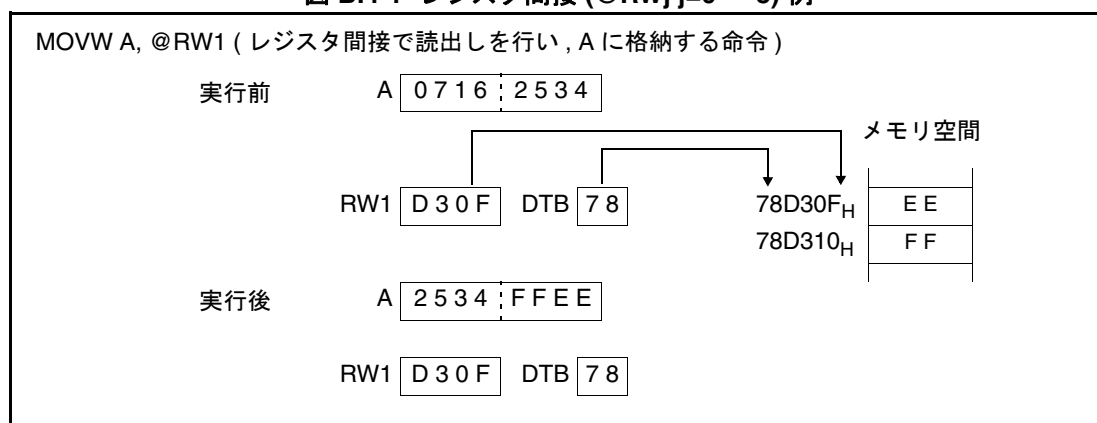
間接アドレッシングでは、記述したオペランドが示すアドレスのデータで、間接的にアドレスを指定します。

### ■ 間接アドレッシング

#### ● レジスタ間接 (@RWj j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-1 に例を示します。

図 B.4-1 レジスタ間接 (@RWj j=0 ~ 3) 例



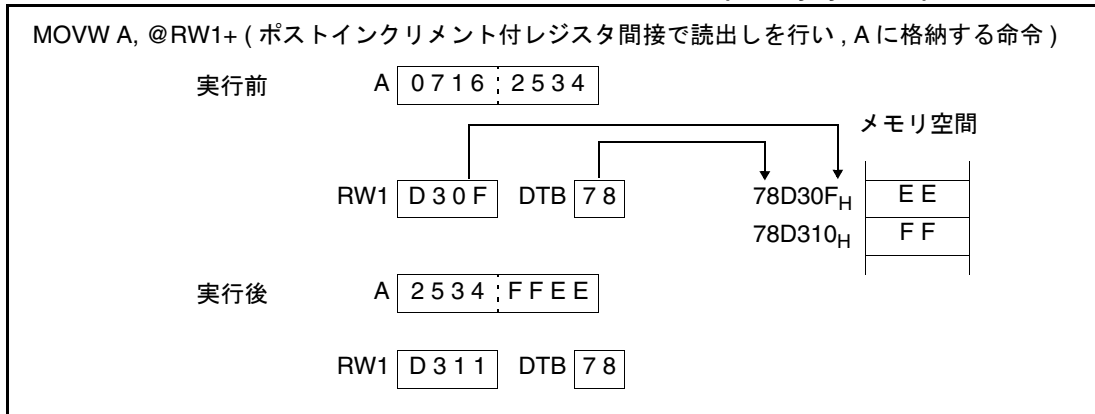
#### ● ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3)

汎用レジスタ RWj の内容をアドレスとしたメモリをアクセスするアドレッシングです。オペランド操作後RWjはオペランドデータ長 (バイトの場合は1, ワードの場合は2, ロングワードの場合は4) 分だけ加算されます。アドレスの bit23 ~ bit16 は、RW0, RW1 を用いた場合はデータバンクレジスタ (DTB) で示され、RW3 の場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され、RW2 の場合はアディショナルデータバンクレジスタ (ADB) で示されます。

ポストインクリメントした結果がインクリメント指定したレジスタ自身のアドレスだった場合は、参照される値はインクリメントした値となり、命令が書込みだった場合は命令による書込みが優先されるので、インクリメントするはずだったレジスタは書込みデータとなります。

図 B.4-2 に例を示します。

図 B.4-2 ポストインクリメント付レジスタ間接 (@RWj+ j=0 ~ 3) 例

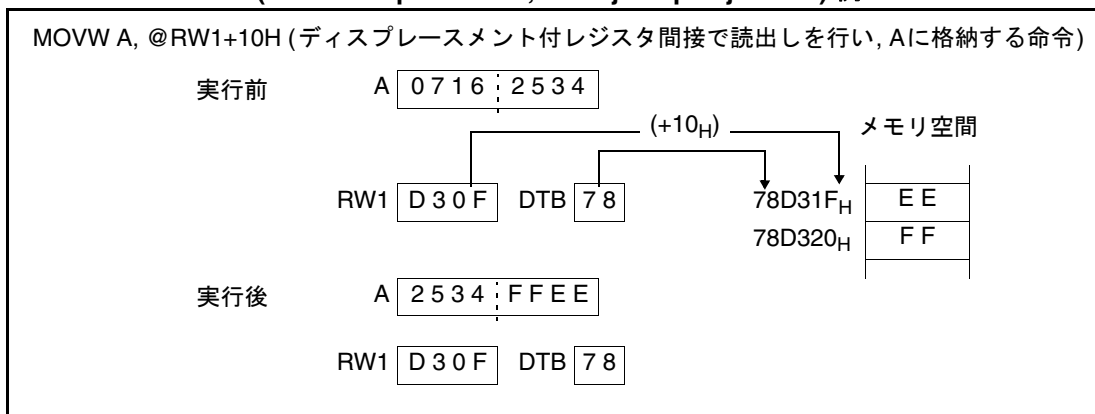


● ディスプレースメント付レジスタ間接

(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3)

汎用レジスタ RWj の内容にディスプレースメントを加算したものをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは、バイトとワードの2種類があり、符号付数値として加算されます。アドレスの bit23 ~ bit16 は, RW0, RW1, RW4, RW5 を用いた場合はデータバンクレジスタ (DTB) で示され, RW3, RW7 を用いた場合はシステムスタックバンクレジスタ (SSB) またはユーザスタックバンクレジスタ (USB) で示され, RW2, RW6 を用いた場合はアディショナルデータバンクレジスタ (ADB) で示されます。図 B.4-3 に例を示します。

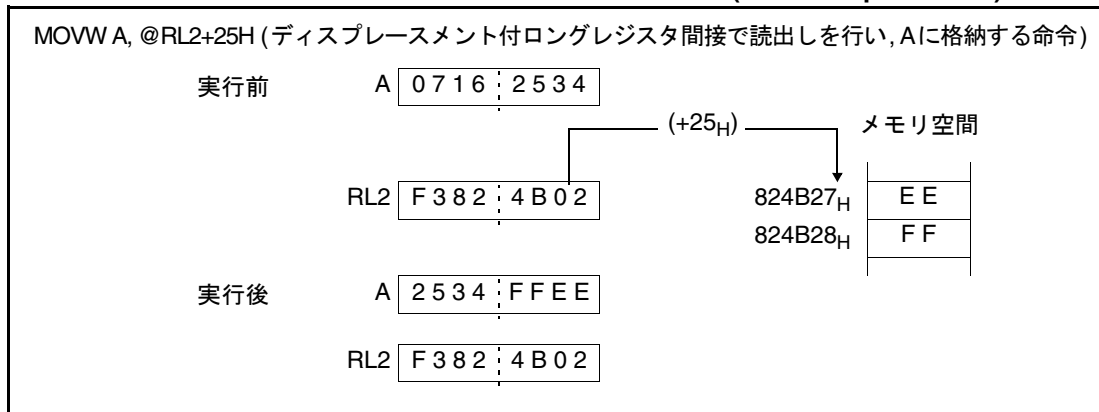
図 B.4-3 ディスプレースメント付レジスタ間接  
(@RWi+disp8 i=0 ~ 7, @RWj+disp16 j=0 ~ 3) 例



● ディスプレースメント付ロングレジスタ間接 (RLi+disp8 i=0 ~ 3)

汎用レジスタ RLi の内容にディスプレースメントを加算した結果の下位 24 ビットをアドレスとしたメモリをアクセスするアドレッシングです。ディスプレースメントは 8 ビットで符号付数値として加算されます。図 B.4-4 に例を示します。

図 B.4-4 ディスプレースメント付ロングレジスタ間接 (@RLi+disp8 i=0 ~ 3) 例



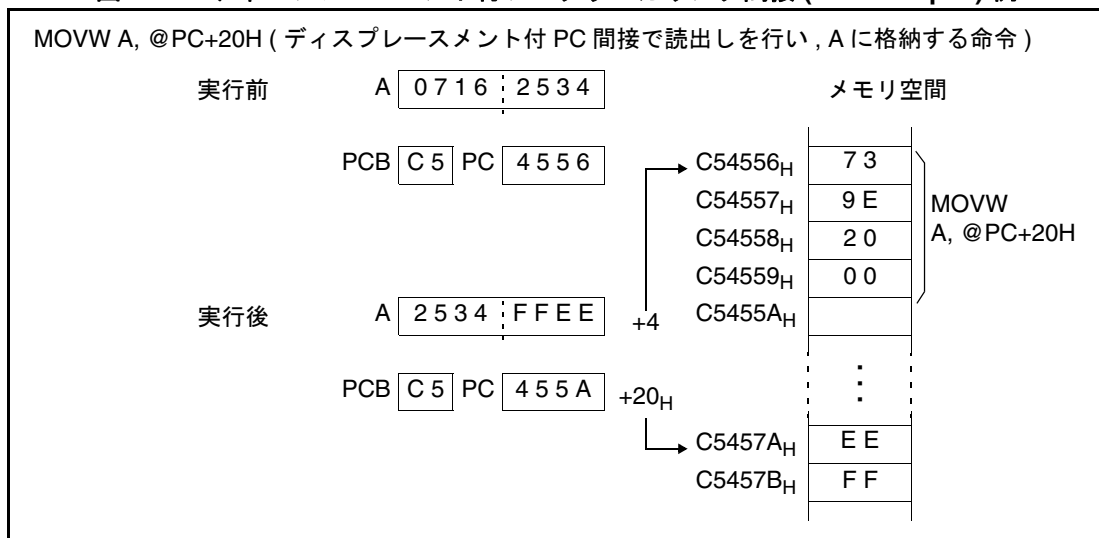
● ディスプレースメント付プログラムカウンタ間接 (@PC+disp16)

(命令のアドレス+4+disp16) で示されるアドレスのメモリをアクセスするアドレッシングです。ディスプレースメントはワード長です。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) により指定されます。次に示す命令のオペランドアドレスは, (次の命令のアドレス +disp16) とは, みなしませんでしたので注意してください。

- DBNZ eam, rel
- CBNE eam, #imm8, rel
- MOV eam, #imm8
- DWBNZ eam, rel
- CWBNE eam, #imm16, rel
- MOVW eam, #imm16

図 B.4-5 に例を示します。

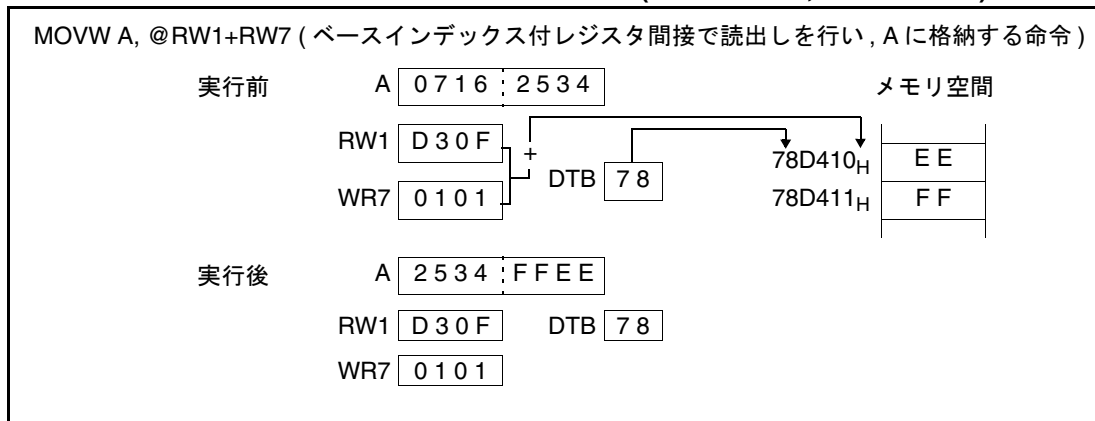
図 B.4-5 ディスプレースメント付プログラムカウンタ間接 (@PC+disp16) 例



● ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7)

汎用レジスタ RW7 の内容に, RW0 あるいは RW1 を加算したものをアドレスとしたメモリをアクセスするアドレッシングです。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で示されます。図 B.4-6 に例を示します。

図 B.4-6 ベースインデックス付レジスタ間接 (@RW0+RW7, @RW1+RW7) 例

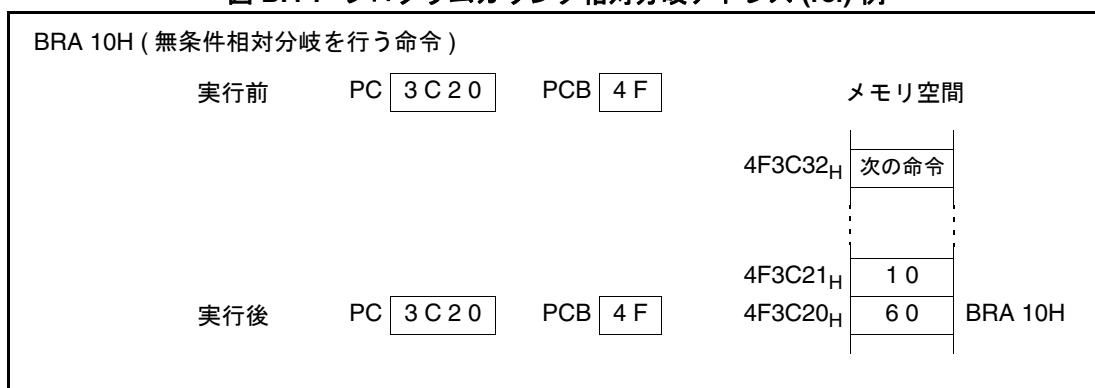


● プログラムカウンタ相対分岐アドレス (rel)

分岐先のアドレスはプログラムカウンタ (PC) の値と 8 ビットのディスプレースメントを加算した値となります。加算の結果が 16 ビットを超えた場合は, バンクレジスタのインクリメントまたはデクレメントをせずに, 超えた分は無視されますので, 64K バイトのバンク内で閉じたアドレスとなります。プログラムカウンタ相対分岐アドレスによるアドレッシングは, 無条件 / 条件分岐命令に使用します。アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で示されます。

図 B.4-7 に例を示します。

図 B.4-7 プログラムカウンタ相対分岐アドレス (rel) 例



● レジスタリスト (rlst)

スタックに対するプッシュ/ポップの対象となるレジスタを指定します。図 B.4-8 にレジスタリストの構成, 図 B.4-9 に例を示します。

図 B.4-8 レジスタリストの構成

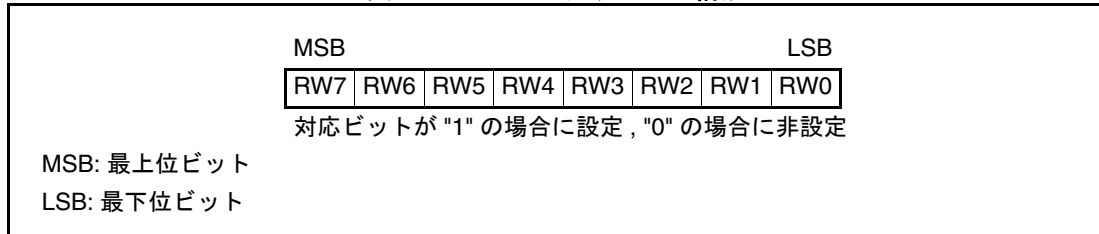
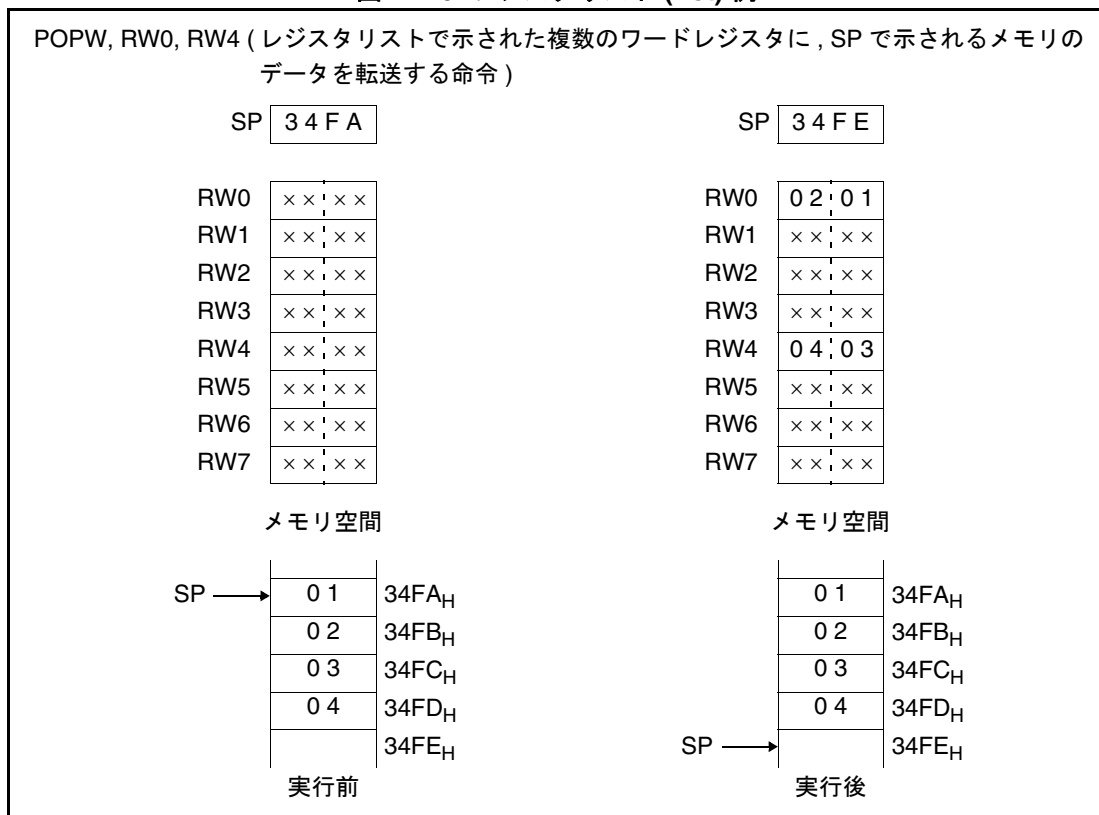


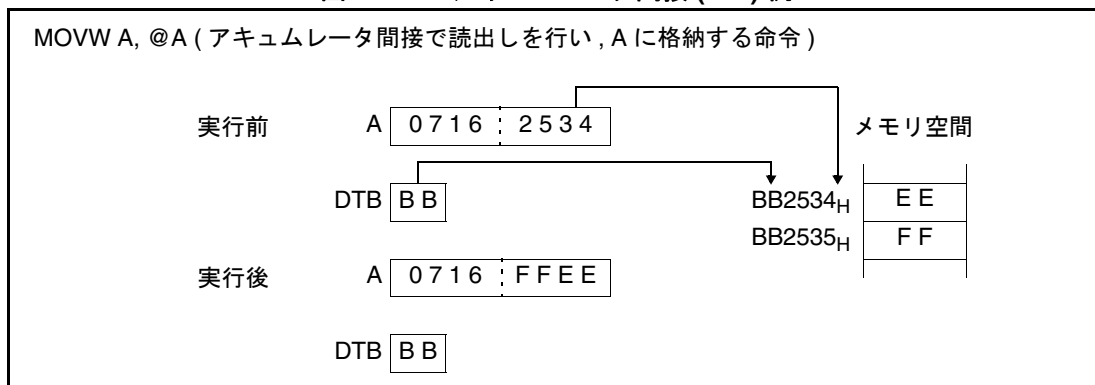
図 B.4-9 レジスタリスト (rlst) 例



### ● アキュムレータ間接 (@A)

アキュムレータの下位バイト (AL) の内容 (16 ビット) で示されるアドレスのメモリをアクセスするアドレス方式です。アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) によりニーモニックで指定されます。図 B.4-10 に例を示します。

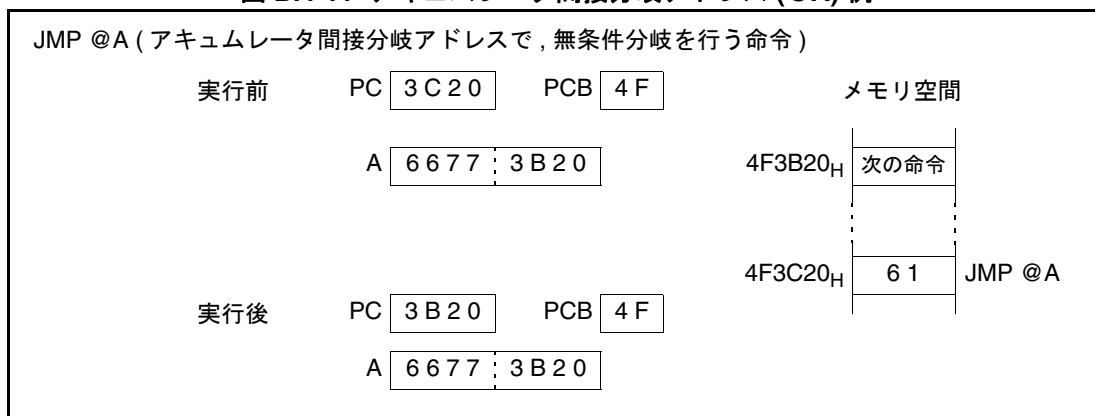
図 B.4-10 アキュムレータ間接 (@A) 例



### ● アキュムレータ間接分岐アドレス (@A)

分岐先のアドレスは, アキュムレータの下位バイト (AL) の内容 (16 ビット) となります。バンク空間内での分岐先を示し, アドレスの bit23 ~ bit16 はプログラムカウンタバンクレジスタ (PCB) で指定されますが, JCTX (Jump Context) 命令の場合は, アドレスの bit23 ~ bit16 はデータバンクレジスタ (DTB) で指定されます。アキュムレータ間接分岐アドレスによるアドレッシングは, 無条件分岐命令に使用します。図 B.4-11 に例を示します。

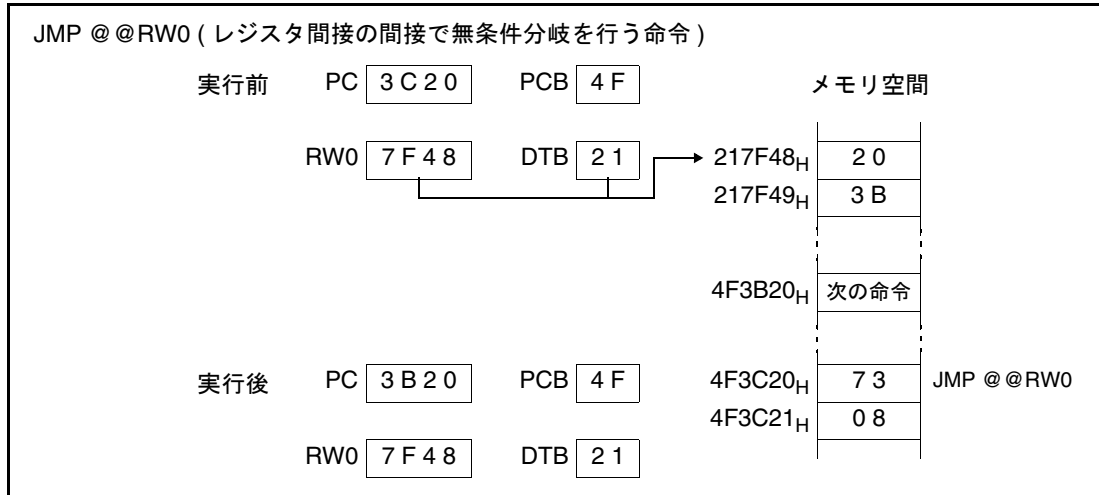
図 B.4-11 アキュムレータ間接分岐アドレス (@A) 例



● 間接指定分岐アドレス (@ear)

ear で示される番地のワードデータが分岐先アドレスとなります。図 B.4-12 に例を示します。

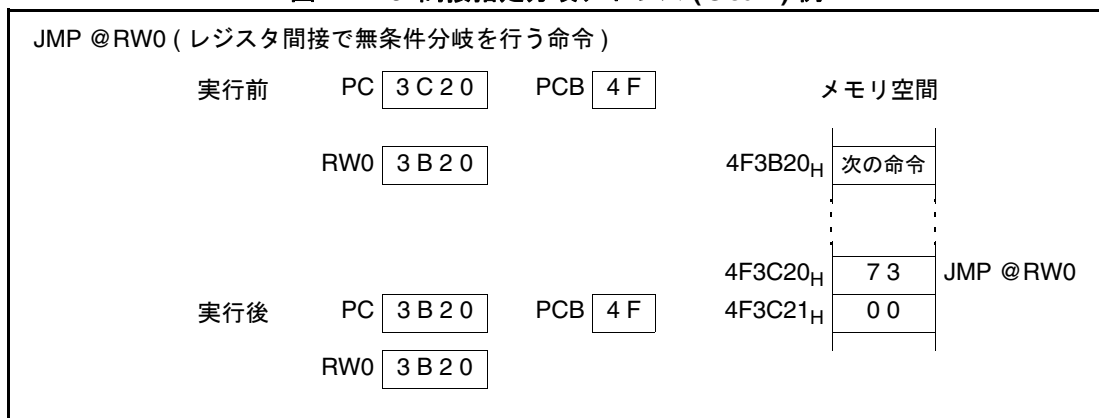
図 B.4-12 間接指定分岐アドレス (@ear) 例



● 間接指定分岐アドレス (@eam)

eam で示される番地のワードデータが分岐先アドレスとなります。図 B.4-13 に例を示します。

図 B.4-13 間接指定分岐アドレス (@eam) 例



## B.5 実行サイクル数

命令の実行に要するサイクル数 ( 実行サイクル数 ) は , 各命令の「サイクル数」の値と , 条件で決まる「補正值」の値およびプログラムフェッチの「サイクル数」の値を加算することで得られます。

### ■ 実行サイクル数

内蔵 ROM などの 16 ビットバスに接続されたメモリ上のプログラムをフェッチする場合には , 実行中の命令がワード境界を越えるごとにプログラムフェッチをするため , データのアクセスなどに干渉すると実行サイクル数が増大します。

外部データバスの 8 ビットバスに接続されたメモリ上のプログラムをフェッチする場合は , 実行中の命令の 1 バイトごとにプログラムフェッチをするため , データのアクセスなどに干渉すると実行サイクル数が増大します。

CPU 間欠動作時は , 汎用レジスタ , 内蔵 ROM, 内蔵 RAM, 内蔵 I/O, 外部データバスのアクセスをすると , 低消費電力モード制御レジスタの CG0, CG1 ビットで指定されるサイクル数分 CPU へ供給されるクロックが一時停止しますので , CPU 間欠動作の命令の実行に要するサイクル数は , 通常の実行サイクル数に , 汎用レジスタ , 内蔵 ROM, 内蔵 RAM, 内蔵 I/O, 外部データバスの「アクセス回数」×一時停止の「サイクル数」の値を「補正值」として加算してください。

### ■ 実行サイクル数計算方法

表 B.5-1 , 表 B.5-2 , 表 B.5-3 に命令実行サイクル数および補正值のデータを示します。

表 B.5-1 各種アドレッシングに対する実行サイクル数

コード	オペランド	(a) *	各種アドレッシングに対するレジスタアクセス回数
		各種アドレッシングに対する実行サイクル数	
00 ~ 07	Ri Rwi RLi	命令一覧表に記載	命令一覧表に記載
08 ~ 0B	@RWj	2	1
0C ~ 0F	@RWj+	4	2
10 ~ 17	@Rwi+disp8	2	1
18 ~ 1B	@Rwi+disp16	2	1
1C	@RW0+RW7	4	2
1D	@RW1+RW7	4	2
1E	@PC+disp16	2	0
1F	addr16	1	0

\*: (a) は「B.8 F2MC-16LX 命令一覧表」の～ ( サイクル数 ) , B ( 補正值 ) で使用されています。" ～ " と "B" の意味については「B.7 命令一覧表の読み方」を参照してください。



表 B.5-2 実サイクル数算出用サイクル数の補正值

オペランド	(b) バイト * <sup>1</sup>		(c) ワード * <sup>1</sup>		(d) ロング * <sup>1</sup>	
	サイクル数	アクセス回数	サイクル数	アクセス回数	サイクル数	アクセス回数
内部レジスタ	+0	1	+0	1	+0	2
内部メモリ 偶数アドレス	+0	1	+0	1	+0	2
内部メモリ 奇数アドレス	+0	1	+2	2	+4	4
外部データバス * <sup>2</sup> 16 ビット偶数アドレス	+1	1	+1	1	+2	2
外部データバス * <sup>2</sup> 16 ビット奇数アドレス	+1	1	+4	2	+8	4
外部データバス * <sup>2</sup> 8 ビット	+1	1	+4	2	+8	4

\*1: (b), (c), (d) は「B.8 F2MC-16LX 命令一覧表」の～ (サイクル数), B (補正值) で使用されています。

\*2: 外部データバスを使用した場合は, レディ入力および自動レディによりウェイトしたサイクル数も加算する必要があります。

表 B.5-3 プログラムフェッチサイクル数算出用サイクル数の補正值

命令	バイト境界	ワード境界
内部メモリ	—	+2
外部データバス 16 ビット	—	+3
外部データバス 8 ビット	+3	—

(注意事項) ・ 外部データバスを使用した場合は, レディ入力および自動レディでウェイトしたサイクル数も加算する必要があります。  
 ・ 実際にはすべてのプログラムフェッチで, 命令実行が遅くなるわけではないので, この補正值は最悪ケースを算出する場合に使用してください。

## B.6 実効アドレスフィールド

表 B.6-1 に実効アドレスフィールドを示します。

### ■ 実効アドレスフィールド

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記			アドレス形式	アドレス拡張部の バイト数*
00	R0	RW0	RL0	レジスタ直接 ea は左から順に バイト ワード ロングワード の型に対応する。	—
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			レジスタ間接	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0+			ポストインクリメント付 レジスタ間接	0
0D	@RW1+				
0E	@RW2+				
0F	@RW3+				
10	@RW0+disp8			8 ビットディスプレースメント付 レジスタ間接	1
11	@RW1+disp8				
12	@RW2+disp8				
13	@RW3+disp8				
14	@RW4+disp8				
15	@RW5+disp8				
16	@RW6+disp8				
17	@RW7+disp8				

表 B.6-1 実効アドレスフィールド (1 / 2)

コード	表記	アドレス形式	アドレス拡張部の バイト数*
18	@RW0+disp16	16 ビットディスプレースメント付 レジスタ間接	2
19	@RW1+disp16		
1A	@RW2+disp16		
1B	@RW3+disp16		
1C	@RW0+RW7	インデックス付レジスタ間接	0
1D	@RW1+RW7	インデックス付レジスタ間接	0
1E	@PC+disp16	16 ビットディスプレースメント付 PC 間接	2
1F	addr16	直接アドレス	2

\*: アドレス拡張部のバイト数は、「B.8 F2MC-16LX 命令一覧表」の#(バイト数)の "+" に当てはまります。  
 "#" の意味については「B.7 命令一覧表の読み方」を参照してください。

## B.7 命令一覧表の読み方

「B.8 F2MC-16LX 命令一覧表」で使用している項目の説明を表 B.7-1 に、記号の説明を表 B.7-2 に示します。

### ■ 命令の表示記号の説明

表 B.7-1 命令一覧表の項目の説明

項目	説明
ニーモニック	英大文字、記号：アセンブラ上もそのまま表記します。 英小文字：アセンブラ上では、書き替えて記述します。 英小文字の後の数：命令中のビット幅を示します。
#	バイト数を示します。
～	サイクル数を示します。
RG	命令実行時のレジスタアクセス回数を示します。 CPU 間欠動作時の補正値を算出するのに使用します。
B	命令実行時の実サイクル数の算出用補正値を示します。 命令実行時の実サイクルは～欄の数値を加算したものとなります。
オペレーション	命令の動作を示します。
LH	アキュムレータの bit15 ～ bit8 に対する特殊動作を示します。 Z: 0 を転送する。 X: 符号を拡張して転送する。 －: 転送しない。
AH	アキュムレータの上位 16 ビットに対する特殊動作を示します。 *: AL から AH へ転送する。 －: 転送しない。 Z: AH へ 00 <sub>H</sub> を転送する。 X: AL の符号拡張で AH へ 00 <sub>H</sub> または FF <sub>H</sub> を転送する。
I	I ( 割込み許可 ), S ( スタック ), T ( ステッキビット ), N ( ネガティブ ), Z ( ゼロ ), V ( オーバフロー ), C ( キャリー ) の各フラグの状態を示します。 *: 命令の実行で変化する。 －: 変化しない。 S: 命令の実行でセットされる。 R: 命令の実行でリセットされる。
S	
T	
N	
Z	
V	
C	
RMW	
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 <sub>H</sub> ~ 0000FF <sub>H</sub> )
#imm4	4 ビット即値データ
#imm8	8 ビット即値データ
#imm16	16 ビット即値データ
#imm32	32 ビット即値データ
ext (imm8)	8 ビット即値データを符号拡張した 16 ビットデータ

表 B.7-2 命令一覧表の記号の説明 (1 / 2)

表記	意味
disp8	8 ビットディスプレースメント
disp16	16 ビットディスプレースメント
bp	ビットオフセット値
vct4	ベクタ番号 (0 ～ 15)
vct8	ベクタ番号 (0 ～ 255)
( ) b	ビットアドレス
rel	PC 相対分岐指定
ear	実効アドレス指定 (コード 00 ～ 07)
eam	実効アドレス指定 (コード 08 ～ 1F)
rlst	レジスタ並び

## B.8 F<sup>2</sup>MC-16LX 命令一覧表

F<sup>2</sup>MC-16LX で使用している命令の一覧を示します。

### ■ F<sup>2</sup>MC-16LX 命令一覧表

表 B.8-1 転送系命令 ( バイト ) 41 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A,dir	2	3	0	(b)	byte (A) ← (dir)	Z	*	-	-	-	*	*	-	-	-
MOV A,addr16	3	4	0	(b)	byte (A) ← (addr16)	Z	*	-	-	-	*	*	-	-	-
MOV A,Ri	1	2	1	0	byte (A) ← (Ri)	Z	*	-	-	-	*	*	-	-	-
MOV A,ear	2	2	1	0	byte (A) ← (ear)	Z	*	-	-	-	*	*	-	-	-
MOV A,eam	2+	3 + (a)	0	(b)	byte (A) ← (eam)	Z	*	-	-	-	*	*	-	-	-
MOV A,io	2	3	0	(b)	byte (A) ← (io)	Z	*	-	-	-	*	*	-	-	-
MOV A,#imm8	2	2	0	0	byte (A) ← imm8	Z	*	-	-	-	*	*	-	-	-
MOV A,@A	2	3	0	(b)	byte (A) ← ((A))	Z	-	-	-	-	*	*	-	-	-
MOV A,@RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	Z	*	-	-	-	*	*	-	-	-
MOVN A,#imm4	1	1	0	0	byte (A) ← imm4	Z	*	-	-	-	R	*	-	-	-
MOVX A,dir	2	3	0	(b)	byte (A) ← (dir)	X	*	-	-	-	*	*	-	-	-
MOVX A,addr16	3	4	0	(b)	byte (A) ← (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX A,Ri	2	2	1	0	byte (A) ← (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX A,ear	2	2	1	0	byte (A) ← (ear)	X	*	-	-	-	*	*	-	-	-
MOVX A,eam	2+	3 + (a)	0	(b)	byte (A) ← (eam)	X	*	-	-	-	*	*	-	-	-
MOVX A,io	2	3	0	(b)	byte (A) ← (io)	X	*	-	-	-	*	*	-	-	-
MOVX A,#imm8	2	2	0	0	byte (A) ← imm8	X	*	-	-	-	*	*	-	-	-
MOVX A,@A	2	3	0	(b)	byte (A) ← ((A))	X	-	-	-	-	*	*	-	-	-
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) ← ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX A,@RLi+disp8	3	10	2	(b)	byte (A) ← ((RLi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOV dir,A	2	3	0	(b)	byte (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV addr16,A	3	4	0	(b)	byte (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,A	1	2	1	0	byte (Ri) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV ear,A	2	2	1	0	byte (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV eam,A	2+	3 + (a)	0	(b)	byte (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV io,A	2	3	0	(b)	byte (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV @RLi+disp8,A	3	10	2	(b)	byte ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,ear	2	3	2	0	byte (Ri) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOV Ri,eam	2+	4 + (a)	1	(b)	byte (Ri) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOV ear,Ri	2	4	2	0	byte (ear) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV eam,Ri	2+	5 + (a)	1	(b)	byte (eam) ← (Ri)	-	-	-	-	-	*	*	-	-	-
MOV Ri,#imm8	2	2	1	0	byte (Ri) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV io,#imm8	3	5	0	(b)	byte (io) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV dir,#imm8	3	5	0	(b)	byte (dir) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV ear,#imm8	3	2	1	0	byte (ear) ← imm8	-	-	-	-	-	*	*	-	-	-
MOV eam,#imm8	3+	4 + (a)	0	(b)	byte (eam) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV @AL,AH	2	3	0	(b)	byte ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCH A,ear	2	4	2	0	byte (A) ↔ (ear)	Z	-	-	-	-	-	-	-	-	-
XCH A,eam	2+	5 + (a)	0	2 × (b)	byte (A) ↔ (eam)	Z	-	-	-	-	-	-	-	-	-
XCH Ri,ear	2	7	4	0	byte (Ri) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCH Ri,eam	2+	9 + (a)	2	2 × (b)	byte (Ri) ↔ (eam)	-	-	-	-	-	-	-	-	-	-

( 注意事項 ) 表中の (a), (b) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-2 転送系命令 (ワード・ロングワード) 38 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A,dir	2	3	0	(c)	word (A) ← (dir)	-	*	-	-	-	*	*	-	-	-
MOVW A,addr16	3	4	0	(c)	word (A) ← (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW A,SP	1	1	0	0	word (A) ← (SP)	-	*	-	-	-	*	*	-	-	-
MOVW A,RWi	1	2	1	0	word (A) ← (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW A,ear	2	2	1	0	word (A) ← (ear)	-	*	-	-	-	*	*	-	-	-
MOVW A,eam	2+	3 + (a)	0	(c)	word (A) ← (eam)	-	*	-	-	-	*	*	-	-	-
MOVW A,io	2	3	0	(c)	word (A) ← (io)	-	*	-	-	-	*	*	-	-	-
MOVW A,@A	2	3	0	(c)	word (A) ← ((A))	-	-	-	-	-	*	*	-	-	-
MOVW A,#imm16	3	2	0	0	word (A) ← imm16	-	*	-	-	-	*	*	-	-	-
MOVW A,@RWi+disp8	2	5	1	(c)	word (A) ← ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW A,@RLi+disp8	3	10	2	(c)	word (A) ← ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW dir,A	2	3	0	(c)	word (dir) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW addr16,A	3	4	0	(c)	word (addr16) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW SP,A	1	1	0	0	word (SP) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,A	1	2	1	0	word (RWi) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW ear,A	2	2	1	0	word (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW eam,A	2+	3 + (a)	0	(c)	word (eam) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW io,A	2	3	0	(c)	word (io) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RWi+disp8,A	2	5	1	(c)	word ((RWi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RLi+disp8,A	3	10	2	(c)	word ((RLi)+disp8) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,ear	2	3	2	0	word (RWi) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,eam	2+	4 + (a)	1	(c)	word (RWi) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVW ear,RWi	2	4	2	0	word (ear) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW eam,RWi	2+	5 + (a)	1	(c)	word (eam) ← (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,#imm16	3	2	1	0	word (RWi) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW io,#imm16	4	5	0	(c)	word (io) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW ear,#imm16	4	2	1	0	word (ear) ← imm16	-	-	-	-	-	*	*	-	-	-
MOVW eam,#imm16	4+	4 + (a)	0	(c)	word (eam) ← imm16	-	-	-	-	-	-	-	-	-	-
MOVW @AL,AH	2	3	0	(c)	word ((A)) ← (AH)	-	-	-	-	-	*	*	-	-	-
XCHW A,ear	2	4	2	0	word (A) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCHW A,eam	2+	5 + (a)	0	2 × (c)	word (A) ↔ (eam)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, ear	2	7	4	0	word (RWi) ↔ (ear)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, eam	2+	9 + (a)	2	2 × (c)	word (RWi) ↔ (eam)	-	-	-	-	-	-	-	-	-	-
MOVL A,ear	2	4	2	0	long (A) ← (ear)	-	-	-	-	-	*	*	-	-	-
MOVL A,eam	2+	5 + (a)	0	(d)	long (A) ← (eam)	-	-	-	-	-	*	*	-	-	-
MOVL A,#imm32	5	3	0	0	long (A) ← imm32	-	-	-	-	-	*	*	-	-	-
MOVL ear,A	2	4	2	0	long (ear) ← (A)	-	-	-	-	-	*	*	-	-	-
MOVL eam,A	2+	5 + (a)	0	(d)	long(eam) ← (A)	-	-	-	-	-	*	*	-	-	-

( 注意事項 ) 表中の (a), (c), (d) は , 表 B.5-1 および表 B.5-2 を参照してください。



表 B.8-3 加減算命令 (バイト・ワード・ロングワード) 42 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ADD A,#imm8	2	2	0	0	byte (A) $\leftarrow$ (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) $\leftarrow$ (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	5+(a)	0	2 × (b)	byte (eam) $\leftarrow$ (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	3	2	0	byte (ear) $\leftarrow$ (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	5+(a)	0	2 × (b)	byte (eam) $\leftarrow$ (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	3	2	0	word (ear) $\leftarrow$ (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	5+(a)	0	2 × (c)	word (eam) $\leftarrow$ (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	3	2	0	word (ear) $\leftarrow$ (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	5+(a)	0	2 × (c)	word (eam) $\leftarrow$ (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	7	4	0	long (ear) $\leftarrow$ (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	9+(a)	0	2 × (d)	long (eam) $\leftarrow$ (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	7	4	0	long (ear) $\leftarrow$ (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	9+(a)	0	2 × (d)	long (eam) $\leftarrow$ (eam) - 1	-	-	-	-	-	*	*	*	-	*

( 注意事項 ) 表中の (a) ～ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-5 比較命令 (バイト・ワード・ロングワード) 11 命令

ニーモニック		#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	0	byte (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMP	A,ear	2	2	1	0	byte (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMP	A,eam	2+	3+(a)	0	(b)	byte (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMP	A,#imm8	2	2	0	0	byte (A) - imm8	-	-	-	-	-	*	*	*	*	-
CMPW	A	1	1	0	0	word (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMPW	A,ear	2	2	1	0	word (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPW	A,eam	2+	3+(a)	0	(c)	word (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPW	A,#imm16	3	2	0	0	word (A) - imm16	-	-	-	-	-	*	*	*	*	-
CMPL	A,ear	2	6	2	0	long (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPL	A,eam	2+	7+(a)	0	(d)	long (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPL	A,#imm32	5	3	0	0	long (A) - imm32	-	-	-	-	-	*	*	*	*	-

( 注意事項 ) 表中の (a) ～ (d) は、表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-6 符号なし乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU A	1	*1	0	0	word (AH) / byte (AL) 商→ byte (AL) 余り→ byte (AH)	-	-	-	-	-	-	-	*	*	-
DIVU A,ear	2	*2	1	0	word (A) / byte (ear) 商→ byte (A) 余り→ byte (ear)	-	-	-	-	-	-	-	*	*	-
DIVU A,eam	2+	*3	0	*6	word (A) / byte (eam) 商→ byte (A) 余り→ byte (eam)	-	-	-	-	-	-	-	*	*	-
DIVUW A,ear	2	*4	1	0	long (A) / word (ear) 商→ word (A) 余り→ word (ear)	-	-	-	-	-	-	-	*	*	-
DIVUW A,eam	2+	*5	0	*7	long (A) / word (eam) 商→ word (A) 余り→ word (eam)	-	-	-	-	-	-	-	*	*	-
MULU A	1	*8	0	0	byte (AH) * byte (AL) → word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,ear	2	*9	1	0	byte (A) * byte (ear) → word (A)	-	-	-	-	-	-	-	-	-	-
MULU A,eam	2+	*10	0	(b)	byte (A) * byte (eam) → word (A)	-	-	-	-	-	-	-	-	-	-
MULUW A	1	*11	0	0	word (AH) * word (AL) → Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,ear	2	*12	1	0	word (A) * word (ear) → Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW A,eam	2+	*13	0	(c)	word (A) * word (eam) → Long (A)	-	-	-	-	-	-	-	-	-	-

\*1 : ゼロディバイトの場合, 3      オーバフローの場合, 7      正常の場合, 15  
 \*2 : ゼロディバイトの場合, 4      オーバフローの場合, 8      正常の場合, 16  
 \*3 : ゼロディバイトの場合, 6+(a)      オーバフローの場合, 9+(a)      正常の場合, 19+(a)  
 \*4 : ゼロディバイトの場合, 4      オーバフローの場合, 7      正常の場合, 22  
 \*5 : ゼロディバイトの場合, 6+(a)      オーバフローの場合, 8+(a)      正常の場合, 26+(a)  
 \*6 : ゼロディバイト, またはオーバフローの場合, (b)      正常の場合, 2 × (b)  
 \*7 : ゼロディバイト, またはオーバフローの場合, (c)      正常の場合, 2 × (c)  
 \*8 : byte(AH) がゼロの場合, 3      byte(AH) がゼロでない場合, 7  
 \*9 : byte(ear) がゼロの場合, 4      byte(ear) がゼロでない場合, 8  
 \*10 : byte(eam) がゼロの場合, 5+(a)      byte(eam) がゼロでない場合, 9+(a)  
 \*11 : word(AH) がゼロの場合, 3      word(AH) がゼロでない場合, 11  
 \*12 : word(ear) がゼロの場合, 4      word(ear) がゼロでない場合, 12  
 \*13 : word(eam) がゼロの場合, 5+(a)      word(eam) がゼロでない場合, 13+(a)  
 ( 注意事項 ) 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-7 符号付き乗除算命令 (ワード・ロングワード) 11 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
DIV A	2	*1	0	0	word (AH) / byte (AL) 商 --> byte (AL) 余り → byte (AH)	Z	-	-	-	-	-	-	*	*	-
DIV A,ear	2	*2	1	0	word (A) / byte (ear) 商 --> byte (A) 余り → byte (ear)	Z	-	-	-	-	-	-	*	*	-
DIV A,eam	2+	*3	0	*6	word (A) / byte (eam) 商 → byte (A) 余り → byte (eam)	Z	-	-	-	-	-	-	*	*	-
DIVW A,ear	2	*4	1	0	long (A) / word (ear) 商 --> word (A) 余り → word (ear)	-	-	-	-	-	-	-	*	*	-
DIVW A,eam	2+	*5	0	*7	long (A) / word (eam) 商 → word (A) 余り → word (eam)	-	-	-	-	-	-	-	*	*	-
MUL A	2	*8	0	0	byte (AH) * byte (AL) → word (A)	-	-	-	-	-	-	-	-	-	-
MUL A,ear	2	*9	1	0	byte (A) * byte (ear) → word (A)	-	-	-	-	-	-	-	-	-	-
MUL A,eam	2+	*10	0	(b)	byte (A) * byte (eam) → word (A)	-	-	-	-	-	-	-	-	-	-
MULW A	2	*11	0	0	word (AH) * word (AL) → Long (A)	-	-	-	-	-	-	-	-	-	-
MULW A,ear	2	*12	1	0	word (A) * word (ear) → Long (A)	-	-	-	-	-	-	-	-	-	-
MULW A,eam	2+	*13	0	(c)	word (A) * word (eam) → Long (A)	-	-	-	-	-	-	-	-	-	-

\*1 : ゼロディバイトの場合, 3      オーバフローの場合, 8 or 18      正常の場合, 18

\*2 : ゼロディバイトの場合, 4      オーバフローの場合, 11 or 22      正常の場合, 23

\*3 : ゼロディバイトの場合, 5+(a)      オーバフローの場合 12+(a) or 23+(a)      正常の場合, 24+(a)

\*4 : 被除数が正: ゼロディバイトの場合, 4      オーバフローの場合, 12 or 30      正常の場合, 31

被除数が負: ゼロディバイトの場合, 4      オーバフローの場合, 12 or 31      正常の場合, 32

\*5 : 被除数が正: ゼロディバイトの場合, 5+(a)      オーバフローの場合, 12+(a) or 31+(a)      正常の場合, 32+(a)

被除数が負: ゼロディバイトの場合, 5+(a)      オーバフローの場合, 13+(a) or 32+(a)      正常の場合, 33+(a)

\*6 : ゼロディバイト, またはオーバフローの場合, (b)      正常の場合, 2 × (b)

\*7 : ゼロディバイト, またはオーバフローの場合, (c)      正常の場合, 2 × (c)

\*8 : byte(AH) がゼロの場合, 3      結果が正の場合, 12      結果が負の場合, 13

\*9 : byte(ear) がゼロの場合, 4      結果が正の場合, 13      結果が負の場合, 14

\*10 : byte(eam) がゼロの場合, 5+(a)      結果が正の場合, 14+(a)      結果が負の場合, 15+(a)

\*11 : word(AH) がゼロの場合, 3      結果が正の場合, 16      結果が負の場合, 19

\*12 : word(ear) がゼロの場合, 4      結果が正の場合, 17      結果が負の場合, 20

\*13 : word(eam) がゼロの場合, 5+(a)      結果が正の場合, 18+(a)      結果が負の場合, 21+(a)

(注意事項) ・ DIV, DIVW 命令でオーバフロー発生時の実行サイクル数に 2 種類ありますが, 演算前と演算後に検出する場合があります。

DIV, DIVW 命令でオーバフロー発生時には, AL の内容は破壊されます。

・ 表中の (a),(b),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-8 論理 1 命令 (バイト・ワード) 39 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
AND A,imm8	2	2	0	0	byte (A) ← (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND A,ear	2	3	1	0	byte (A) ← (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND A,eam	2+	4+(a)	0	(b)	byte (A) ← (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND ear,A	2	3	2	0	byte (ear) ← (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND eam,A	2+	5+(a)	0	2 × (b)	byte (eam) ← (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR A,imm8	2	2	0	0	byte (A) ← (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR A,ear	2	3	1	0	byte (A) ← (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR A,eam	2+	4+(a)	0	(b)	byte (A) ← (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR ear,A	2	3	2	0	byte (ear) ← (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) ← (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR A,imm8	2	2	0	0	byte (A) ← (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR A,ear	2	3	1	0	byte (A) ← (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR A,eam	2+	4+(a)	0	(b)	byte (A) ← (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR ear,A	2	3	2	0	byte (ear) ← (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR eam,A	2+	5+(a)	0	2 × (b)	byte (eam) ← (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT A	1	2	0	0	byte (A) ← not (A)	-	-	-	-	-	*	*	R	-	-
NOT ear	2	3	2	0	byte (ear) ← not (ear)	-	-	-	-	-	*	*	R	-	-
NOT eam	2+	5+(a)	0	2 × (b)	byte (eam) ← not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW A	1	2	0	0	word (A) ← (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW A,imm16	3	2	0	0	word (A) ← (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW A,ear	2	3	1	0	word (A) ← (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW A,eam	2+	4+(a)	0	(c)	word (A) ← (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW ear,A	2	3	2	0	word (ear) ← (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW eam,A	2+	5+(a)	0	2 × (c)	word (eam) ← (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW A	1	2	0	0	word (A) ← (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW A,imm16	3	2	0	0	word (A) ← (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW A,ear	2	3	1	0	word (A) ← (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW A,eam	2+	4+(a)	0	(c)	word (A) ← (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW ear,A	2	3	2	0	word (ear) ← (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) ← (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW A	1	2	0	0	word (A) ← (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW A,imm16	3	2	0	0	word (A) ← (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW A,ear	2	3	1	0	word (A) ← (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW A,eam	2+	4+(a)	0	(c)	word (A) ← (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW ear,A	2	3	2	0	word (ear) ← (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW eam,A	2+	5+(a)	0	2 × (c)	word (eam) ← (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW A	1	2	0	0	word (A) ← not (A)	-	-	-	-	-	*	*	R	-	-
NOTW ear	2	3	2	0	word (ear) ← not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW eam	2+	5+(a)	0	2 × (c)	word (eam) ← not (eam)	-	-	-	-	-	*	*	R	-	*

( 注意事項 ) 表中の (a), (b), (c) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-9 論理 2 演算命令 (ロングワード) 6 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL A,ear	2	6	2	0	long (A) $\leftarrow$ (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL A,eam	2+	7+(a)	0	(d)	long (A) $\leftarrow$ (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL A,ear	2	6	2	0	long (A) $\leftarrow$ (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL A,eam	2+	7+(a)	0	(d)	long (A) $\leftarrow$ (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL A,ear	2	6	2	0	long (A) $\leftarrow$ (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL A,eam	2+	7+(a)	0	(d)	long (A) $\leftarrow$ (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

(注意事項) 表中の (a), (d) は, 表 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) $\leftarrow$ 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG ear	2	3	2	0	byte (ear) $\leftarrow$ 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG eam	2+	5+(a)	0	2 $\times$ (b)	byte (eam) $\leftarrow$ 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW A	1	2	0	0	word (A) $\leftarrow$ 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW ear	2	3	2	0	word (ear) $\leftarrow$ 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW eam	2+	5+(a)	0	2 $\times$ (c)	word (eam) $\leftarrow$ 0 - (eam)	-	-	-	-	-	*	*	*	*	*

(注意事項) 表中の (a), (b), (c) は, 表 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) $\leftarrow$ 最初に "1" が立っていた所まで左シフトする byte (R0) $\leftarrow$ その場合のシフト回数	-	-	-	-	-	-	*	-	-	-

\*1: アキュムレータがすべて "0" であった場合は, 4 "0" 以外であった場合は, 6+(R0)

表 B.8-12 シフト命令 ( バイト・ワード・ロングワード ) 18 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
RORC A	2	2	0	0	byte (A) ← 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLc A	2	2	0	0	byte (A) ← 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC ear	2	3	2	0	byte (ear) ← 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
RORC eam	2+	5+(a)	0	2 × (b)	byte (eam) ← 右ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ROLc ear	2	3	2	0	byte (ear) ← 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	-
ROLc eam	2+	5+(a)	0	2 × (b)	byte (eam) ← 左ローテートキャリー付	-	-	-	-	-	*	*	-	*	*
ASR A,R0	2	*1	1	0	byte (A) ← 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSR A,R0	2	*1	1	0	byte (A) ← ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSL A,R0	2	*1	1	0	byte (A) ← ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRW A	1	2	0	0	word (A) ← 算術右シフト (A,1 ビット)	-	-	-	-	*	*	*	-	*	-
LSRW A/SHRW A	1	2	0	0	word (A) ← ロジカル右シフト (A,1 ビット)	-	-	-	-	*	R	*	-	*	-
LSLW A/SHLW A	1	2	0	0	word (A) ← ロジカル左シフト (A,1 ビット)	-	-	-	-	-	*	*	-	*	-
ASRW A,R0	2	*1	1	0	word (A) ← 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRW A,R0	2	*1	1	0	word (A) ← ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLW A,R0	2	*1	1	0	word (A) ← ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-
ASRL A,R0	2	*2	1	0	long (A) ← 算術右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSRL A,R0	2	*2	1	0	long (A) ← ロジカル右バレルシフト (A,R0)	-	-	-	-	*	*	*	-	*	-
LSLL A,R0	2	*2	1	0	long (A) ← ロジカル左バレルシフト (A,R0)	-	-	-	-	-	*	*	-	*	-

\*1 : R0 が "0" であった場合 , 6 "0" 以外であった場合は , 5+(R0)

\*2 : R0 が "0" であった場合 , 6 "0" 以外であった場合は , 6+(R0)

( 注意事項 ) 表中の (a),(b) は , 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-13 分岐命令 1 31 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ rel	2	*1	0	0	(Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNZ/BNE rel	2	*1	0	0	(Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BC/BLO rel	2	*1	0	0	(C) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNC/BHS rel	2	*1	0	0	(C) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BN rel	2	*1	0	0	(N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BP rel	2	*1	0	0	(N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BV rel	2	*1	0	0	(V) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNV rel	2	*1	0	0	(V) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BT rel	2	*1	0	0	(T) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BNT rel	2	*1	0	0	(T) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLT rel	2	*1	0	0	(V) xor (N) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGE rel	2	*1	0	0	(V) xor (N) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLE rel	2	*1	0	0	((V) xor (N)) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BGT rel	2	*1	0	0	((V) xor (N)) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BLS rel	2	*1	0	0	(C) or (Z) = 1 の場合分岐	-	-	-	-	-	-	-	-	-	-
BHI rel	2	*1	0	0	(C) or (Z) = 0 の場合分岐	-	-	-	-	-	-	-	-	-	-
BRA rel	2	*1	0	0	無条件分岐	-	-	-	-	-	-	-	-	-	-
JMP @A	1	2	0	0	word (PC) ← (A)	-	-	-	-	-	-	-	-	-	-
JMP addr16	3	3	0	0	word (PC) ← addr16	-	-	-	-	-	-	-	-	-	-
JMP @ear	2	3	1	0	word (PC) ← (ear)	-	-	-	-	-	-	-	-	-	-
JMP @eam	2+	4+(a)	0	(c)	word (PC) ← (eam)	-	-	-	-	-	-	-	-	-	-
JMPP @ear 注 1)	2	5	2	0	word (PC) ← (ear), (PCB) ← (ear+2)	-	-	-	-	-	-	-	-	-	-
JMPP @eam 注 1)	2+	6+(a)	0	(d)	word (PC) ← (eam), (PCB) ← (eam+2)	-	-	-	-	-	-	-	-	-	-
JMPP addr24	4	4	0	0	word (PC) ← ad24 0-15, (PCB) ← ad24 16-23	-	-	-	-	-	-	-	-	-	-
CALL @ear 注 2)	2	6	1	(c)	word (PC) ← (ear)	-	-	-	-	-	-	-	-	-	-
CALL @eam 注 2)	2+	7+(a)	0	2 × (c)	word (PC) ← (eam)	-	-	-	-	-	-	-	-	-	-
CALL addr16 注 3)	3	6	0	(c)	word (PC) ← addr16	-	-	-	-	-	-	-	-	-	-
CALLV #vct4 注 3)	1	7	0	2 × (c)	ベクタコール命令	-	-	-	-	-	-	-	-	-	-
CALLP @ear 注 4)	2	10	2	2 × (c)	word (PC) ← (ear)0-15, (PCB) ← (ear)16-23	-	-	-	-	-	-	-	-	-	-
CALLP @eam 注 4)	2+	11+(a)	0	*2	word (PC) ← (eam)0-15, (PCB) ← (eam)16-23	-	-	-	-	-	-	-	-	-	-
CALLP addr24 注 5)	4	10	0	2 × (c)	word (PC) ← addr0-15, (PCB) ← addr16-23	-	-	-	-	-	-	-	-	-	-

\*1 : 分岐が行われる場合 , 4 行われない場合 , 3

\*2 : 3 × (c)+(b)

注 1) 分岐先アドレスの読み込み (word)

注 2) W : スタックへの退避 (word), R : 分岐先アドレスの読み込み (word)

注 3) スタックへの退避 (word)

注 4) W : スタックへの退避 (long), R : 分岐先アドレスの読み込み (long)

注 5) スタックへの退避 (long)

(注意事項) 表中の (a) ~ (d) は , 表 B.5-1 および表 B.5-2 を参照してください。



表 B.8-14 分岐命令 2 19 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE A,#imm8,rel	3	*1	0	0	byte (A) ≠ imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE A,#imm16,rel	4	*1	0	0	word (A) ≠ imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE ear,#imm8,rel	4	*2	1	0	byte (ear) ≠ imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CBNE eam,#imm8,rel 注 1)	4+	*3	0	(b)	byte (eam) ≠ imm8 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE ear,#imm16,rel	5	*4	1	0	word (ear) ≠ imm16 で分岐	-	-	-	-	-	*	*	*	*	-
CWBNE eam,#imm16,rel 注 1)	5+	*3	0	(c)	word (eam) ≠ imm16 で分岐	-	-	-	-	-	*	*	*	*	-
DBNZ ear,rel	3	*5	2	0	byte (ear) = (ear) - 1, (ear) ≠ 0 で分岐	-	-	-	-	-	*	*	*	*	-
DBNZ eam,rel	3+	*6	2	2 × (b)	byte (eam) = (eam) - 1, (eam) ≠ 0 で分岐	-	-	-	-	-	*	*	*	*	-
DWBNZ ear,rel	3	*5	2	0	word (ear) = (ear) - 1, (ear) ≠ 0 で分岐	-	-	-	-	-	*	*	*	*	-
DWBNZ eam,rel	3+	*6	2	2 × (c)	word (eam) = (eam) - 1, (eam) ≠ 0 で分岐	-	-	-	-	-	*	*	*	*	-
INT #vct8	2	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT addr16	3	16	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INTP addr24	4	17	0	6 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
INT9	1	20	0	8 × (c)	ソフトウェア割込み	-	-	R	S	-	-	-	-	-	-
RETI	1	*8	0	*7	割込みからの復帰	-	-	*	*	*	*	*	*	*	-
LINK #imm8	2	6	0	(c)	関数の入口で、旧フレームポインタをスタックに保存し、新フレームポインタの設定、およびローカルポインタの領域を確保する	-	-	-	-	-	-	-	-	-	-
UNLINK	1	5	0	(c)	関数の出口で、旧フレームポインタをスタックから復帰させる	-	-	-	-	-	-	-	-	-	-
RET 注 2)	1	4	0	(c)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-
RETP 注 3)	1	6	0	(d)	サブルーチンからの復帰	-	-	-	-	-	-	-	-	-	-

\*1：分岐が行われる場合，5 行われない場合，4

\*2：分岐が行われる場合，13 行われない場合，12

\*3：分岐が行われる場合，7+(a) 行われない場合，6+(a)

\*4：分岐が行われる場合，8 行われない場合，7

\*5：分岐が行われる場合，7 行われない場合，6

\*6：分岐が行われる場合，8+(a) 行われない場合，7+(a)

\*7：次の割込み要求へ分岐する場合，3 × (b)+2 × (c) 今の割込みから復帰の場合，6 × (c)

\*8：次の割込みへ分岐する場合，15 今の割込みから復帰する場合，17

注 1) CBNE / CWBNE 命令では，RWj+ のアドレッシングモードは，使用しないでください。

注 2) スタックからの復帰 (word)

注 3) スタックからの復帰 (long)

( 注意事項 ) 表中の (a) ～ (d) は，表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-15 その他制御命令 (バイト・ワード・ロングワード) 28 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW A	1	4	0	(c)	word (SP) ← (SP) - 2, ((SP)) ← (A)	-	-	-	-	-	-	-	-	-	-
PUSHW AH	1	4	0	(c)	word (SP) ← (SP) - 2, ((SP)) ← (AH)	-	-	-	-	-	-	-	-	-	-
PUSHW PS	1	4	0	(c)	word (SP) ← (SP) - 2, ((SP)) ← (PS)	-	-	-	-	-	-	-	-	-	-
PUSHW rlst	2	*3	*5	*4	(SP) ← (SP) - 2n, ((SP)) ← (rlst)	-	-	-	-	-	-	-	-	-	-
POPW A	1	3	0	(c)	word (A) ← ((SP)), (SP) ← (SP) + 2	-	*	-	-	-	-	-	-	-	-
POPW AH	1	3	0	(c)	word (AH) ← ((SP)), (SP) ← (SP) + 2	-	-	-	-	-	-	-	-	-	-
POPW PS	1	4	0	(c)	word (PS) ← ((SP)), (SP) ← (SP) + 2	-	-	*	*	*	*	*	*	*	-
POPW rlst	2	*2	*5	*4	(rlst) ← ((SP)), (SP) ← (SP) + 2n	-	-	-	-	-	-	-	-	-	-
JCTX @A	1	14	0	6 × (c)	コンテキストスイッチ命令	-	-	*	*	*	*	*	*	*	-
AND CCR,#imm8	2	3	0	0	byte (CCR) ← (CCR) and imm8	-	-	*	*	*	*	*	*	*	-
OR CCR,#imm8	2	3	0	0	byte (CCR) ← (CCR) or imm8	-	-	*	*	*	*	*	*	*	-
MOV RP,#imm8	2	2	0	0	byte (RP) ← imm8	-	-	-	-	-	-	-	-	-	-
MOV ILM,#imm8	2	2	0	0	byte (ILM) ← imm8	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,ear	2	3	1	0	word (RWi) ← ear	-	-	-	-	-	-	-	-	-	-
MOVEA RWi,eam	2+	2+(a)	1	0	word (RWi) ← eam	-	-	-	-	-	-	-	-	-	-
MOVEA A,ear	2	1	0	0	word (A) ← ear	-	*	-	-	-	-	-	-	-	-
MOVEA A,eam	2+	1+(a)	0	0	word (A) ← eam	-	*	-	-	-	-	-	-	-	-
ADDSP #imm8	2	3	0	0	word (SP) ← (SP) + ext(imm8)	-	-	-	-	-	-	-	-	-	-
ADDSP #imm16	3	3	0	0	word (SP) ← (SP) + imm16	-	-	-	-	-	-	-	-	-	-
MOV A,brg1	2	*1	0	0	byte (A) ← (brg1)	Z	*	-	-	-	*	*	-	-	-
MOV brg2,A	2	1	0	0	byte (brg2) ← (A)	-	-	-	-	-	*	*	-	-	-
NOP	1	1	0	0	ノーオペレーション	-	-	-	-	-	-	-	-	-	-
ADB	1	1	0	0	AD 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
DTB	1	1	0	0	DT 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
PCB	1	1	0	0	PC 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
SPB	1	1	0	0	SP 空間アクセス用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
NCC	1	1	0	0	フラグ無変化用プリフィックスコード	-	-	-	-	-	-	-	-	-	-
CMR	1	1	0	0	コモンレジスタバンク用プリフィックス	-	-	-	-	-	-	-	-	-	-

\*1 : PCB,ADB,SSB,USB . . . . . 1 ステート

DTB,DPR . . . . . 2 ステート

\*2 : 7+3 × (POP する回数)+2 × (POP する最後のレジスタ番号), RLST=0(転送レジスタ無し) の場合 7

\*3 : 29+3 × (PUSH する回数)-3 × (PUSH する最後のレジスタ番号), RLST=0(転送レジスタ無し) の場合 8

\*4 : (POP する回数) × (c), または (PUSH する回数) × (c)

\*5 : (POP する回数), または (PUSH する回数)

(注意事項) 表中の (a),(c) は, 表 B.5-1 および表 B.5-2 を参照してください。

表 B.8-16 ビット操作命令 21 命令

ニーモニック	#	～	RG	B	オペレーション	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB A,dir:bp	3	5	0	(b)	byte (A) $\leftarrow$ (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。

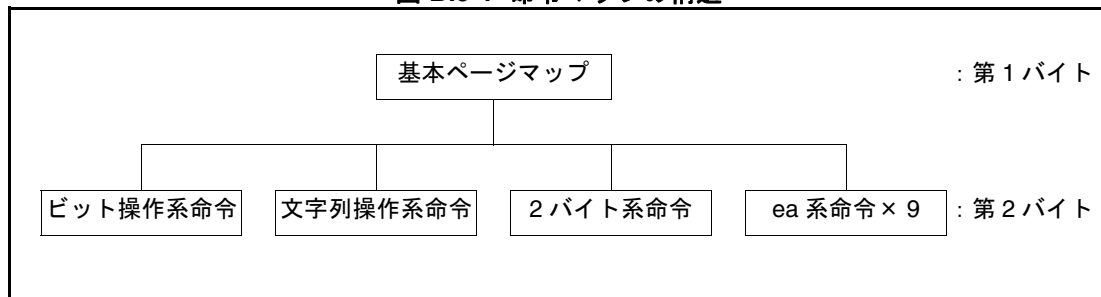
## B.9 命令マップ

F<sup>2</sup>MC-16LX の命令コードは、1 ～ 2 バイトで構成されていますので、命令マップは複数のページで構成されています。

F<sup>2</sup>MC-16LX の命令マップを示します。

### ■ 命令マップの構造

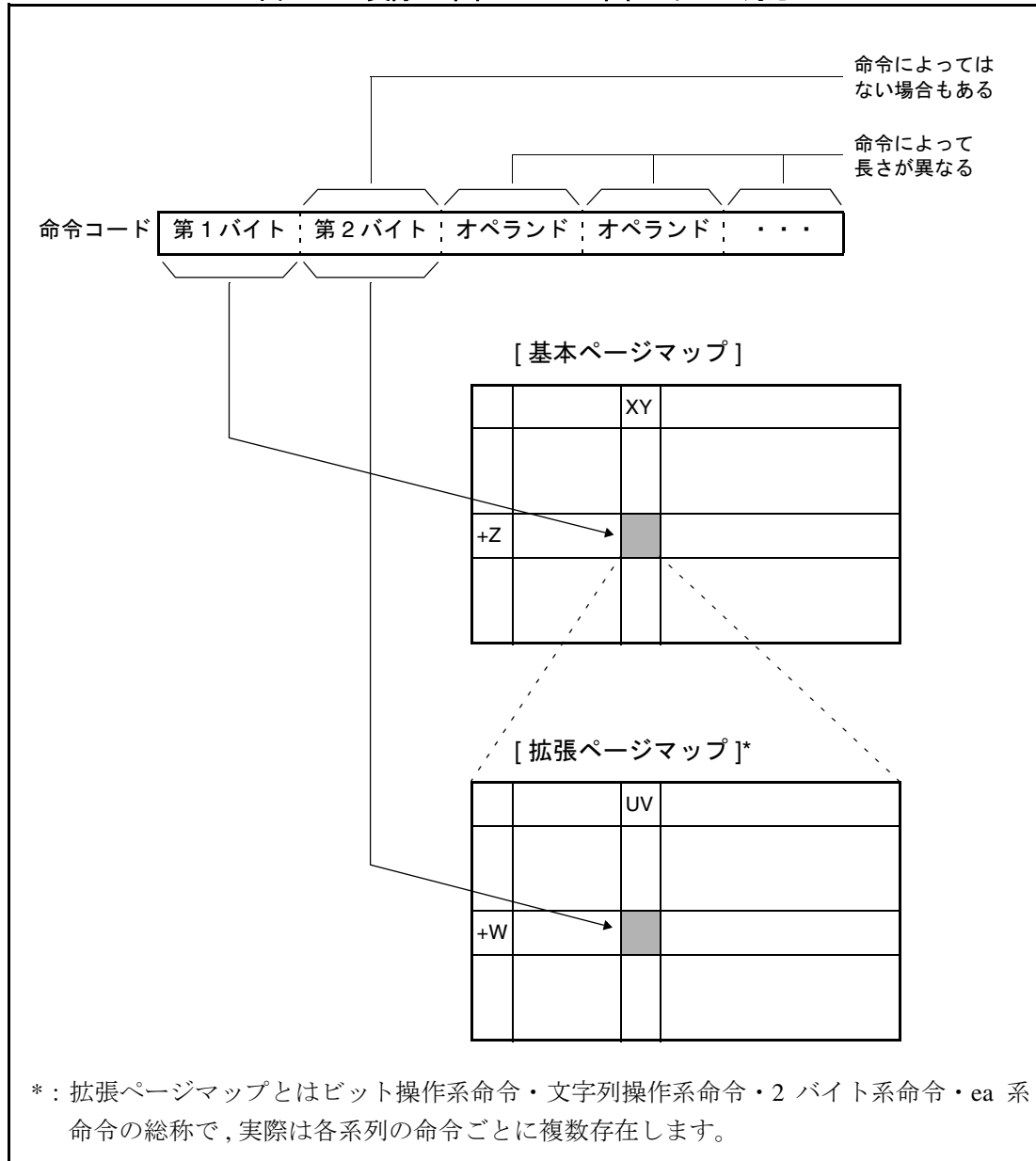
図 B.9-1 命令マップの構造



1 バイトで終わる命令 (NOP 命令など) は基本ページで完結し、2 バイト必要とする命令 (MOVS 命令など) は第 1 バイトを参照した段階で第 2 バイト用のマップの存在がわかり、第 2 バイト用のマップを参照して続く 1 バイトを調べることができます。

実際の命令コードと命令マップの対応を図 B.9-2 に示します。

図 B.9-2 実際の命令コードと命令マップの対応



命令コードの例を表 B.9-1 に示します。

表 B.9-1 命令コードの例

命令	第1バイト (基本ページマップより)	第2バイト (拡張ページマップより)
NOP	$00 + 0 = 00$	—
AND A, #8	$30 + 4 = 34$	—
MOV A, ADB	$60 + F = 6F$	$00 + 0 = 00$
@RW2+d8, #8, rel	$70 + 0 = 70$	$F0 + 2 = F2$

表 B.9-2 基本ページマップ

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	NOP	CMR	ADD A, dir	ADD A, #8	MOV A, dir	MOV A, io	BRA rel	ea 系命令 その 1	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BZ/BEQ rel
+1	INT9	NCC	SUB A, dir	SUB A, #8	MOV dir, A	MOV io, A	JMP @A	ea 系命令 その 2	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BNZ/BNL rel
+2	ADDC A	SUBDC A	ADDC A	SUBC A	MOV A, #8	MOV A, addr16	JMP addr16	ea 系命令 その 3	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BC/BLO rel
+3	NEG A	JCTX @A	CMP A	CMP A, #8	MOVX A, #8	MOV addr16, A	JMPP addr24	ea 系命令 その 4	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BNC/BHS rel
+4	PCB	EXT	AND CCR, #8	AND A, #8	MOV dir, #8	MOV io, #8	CALL addr16	ea 系命令 その 5	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BN rel
+5	DTB	ZEXT	OR CCR, #8	OR A, #8	MOVX A, dir	MOVX A, io	CALLP addr24	ea 系命令 その 6	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BP rel
+6	ADB	SWAP	DIVU A	XOR A, #8	MOVW A, SP	MOVW io, #16	RETP	ea 系命令 その 7	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BV rel
+7	SPB	ADDSP #8	MULU #8	NOT A	MOVW SP, A	MOVX A, addr16	RET	ea 系命令 その 8	MOV A, Ri	MOV Ri, A	MOV Ri, #8	MOV A, Ri	MOVX A, @RWi+d8	MOV A, #4	CALL #4	BNV rel
+8	LINK #imm8	ADDL A, #32	ADDW A	ADDW A, #16	MOVW A, dir	MOVW A, io	INT #vct8	ea 系命令 その 9	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	MOVW 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	MOVW A, #4	CALL #4	BNT rel
+A	MOV RP, #8	MOV ILM, #8	CBNE A, #8, rel	CWBNL A, #16, rel	MOVW A, #16	MOVW A, addr16	INTP	MOV Ri, ea	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	MOVW 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	MOVW 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	MOVW A, #4	CALL #4	BLE rel
+D		ZEXTW	ORW A	ORW A, #16	PUSHW AH	POPW AH		MOVW ea, Ri	MOVW A, Ri	MOVW Ri, A	MOVW Ri, #16	MOVW A, Ri	MOVW @RWi+d8 @RWi+d8,A	MOVW 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	MOVW 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	MOVW A, #4	CALL #4	BHI rel

表 B.9-3 ビット操作系命令マップ (第 1 バイト = 6C<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVB A, io:bp		MOVB io:bp, A		CLRB io:bp		SETB io:bp		BBC io:bp, rel		BBS io:bp, rel		WBTS io:bp		WBTC io:bp	
+1																
+2																
+3																
+4																
+5																
+6																
+7																
+8	MOVB A, dir:bp	MOVB A, addr16:bp	MOVB dir:bp, A	MOVB addr16:bp, A	CLRB dir:bp	CLRB addr16:bp	SETB dir:bp	SETB addr16:bp	BBC dir:bp, rel	BBC addr16:bp, rel	BBS dir:bp, rel	BBS addr16:bp, rel				SBS 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<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVSI, MOVSD, MOVSWI, MOVSWD								SCWEQI, SCEQD, PCB	SCWEQI, SCEQD, PCB	SCWEQI, SCEQD, PCB	SCWEQD, FILSI, PCB	FILSI, PCB			
+1	PCB, DTB								PCB, DTB	PCB, DTB	PCB, DTB	PCB, DTB	PCB, DTB		DTB	
+2	PCB, ADB								PCB, ADB	PCB, ADB	PCB, ADB	PCB, ADB	PCB, ADB		ADB	
+3	PCB, SPB								PCB, SPB	PCB, SPB	PCB, SPB	PCB, SPB	PCB, SPB		SPB	
+4	DTB, PCB															
+5	DTB, DTB															
+6	DTB, ADB															
+7	DTB, SPB															
+8	ADB, PCB															
+9	ADB, DTB															
+A	ADB, ADB															
+B	ADB, SPB															
+C	SPB, PCB															
+D	SPB, DTB															
+E	SPB, ADB															
+F	SPB, SPB															

表 B.9-5 2 バイト系命令マップ (第 1 バイト = 6F<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV A, DTB	MOV DTB, A	MOVX A, @RL0+d8	MOV @RL0+d8, A	MOV @RL0+d8, A											
+1	MOV A, ADB	MOV ADB, A														
+2	MOV A, SSB	MOV SSB, A	MOVX A, @RL1+d8	MOV @RL1+d8, A	MOV @RL1+d8, A											
+3	MOV A, USB	MOV USB, A														
+4	MOV A, DPR	MOV DPR, A	MOVX A, @RL2+d8	MOV @RL2+d8, A	MOV @RL2+d8, A											
+5	MOV A, @A	MOV @AL, AH														
+6	MOV A, PCB	MOV A, @A	MOVX A, @RL3+d8	MOV @RL3+d8, A	MOV @RL3+d8, A											
+7	ROL A	ROL A														
+8				MOVW @RL0+d8, A	MOVW A, @RL0+d8			MUL A								
+9								MULW A								
+A				MOVW @RL1+d8, A	MOVW A, @RL1+d8			DIVU A								
+B																
+C	LSLW A, R0	LSLL A, R0	LSL A, R0	MOVW @RL2+d8, A	MOVW A, @RL2+d8											
+D	MOVW A, @A	MOVW @AL, AH	NRML A, R0													
+E	ASRW A, R0	ASRL A, R0	ASR A, R0	MOVW @RL3+d8, A	MOVW A, @RL3+d8											
+F	LSRW A, R0	LSRL A, R0	LSR A, R0													

表 B.9-6 ea 系命令 その 1 (第 1 バイト = 70<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
					CBNE↓	CWNE↓									CBNE↓	
+0	ADDL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	SUBL A, A, RL0, @RW0+d8	RW0, @RW0+d8 #16, rel	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	CMPL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ANDL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	ORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	XORL A, A, RL0, @RW0+d8	R0, @RW0+d8 #8, rel	R0, @RW0+d8 #8, rel
+1	ADDL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	SUBL A, A, RL0, @RW1+d8	RW1, @RW1+d8 #16, rel	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	CMPL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ANDL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	ORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	XORL A, A, RL0, @RW1+d8	R1, @RW1+d8 #8, rel	R1, @RW1+d8 #8, rel
+2	ADDL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	SUBL A, A, RL1, @RW2+d8	RW2, @RW2+d8 #16, rel	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	CMPL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ANDL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	ORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	XORL A, A, RL1, @RW2+d8	R2, @RW2+d8 #8, rel	R2, @RW2+d8 #8, rel
+3	ADDL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	SUBL A, A, RL1, @RW3+d8	RW3, @RW3+d8 #16, rel	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	CMPL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ANDL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	ORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	XORL A, A, RL1, @RW3+d8	R3, @RW3+d8 #8, rel	R3, @RW3+d8 #8, rel
+4	ADDL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	SUBL A, A, RL2, @RW4+d8	RW4, @RW4+d8 #16, rel	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	CMPL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ANDL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	ORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	XORL A, A, RL2, @RW4+d8	R4, @RW4+d8 #8, rel	R4, @RW4+d8 #8, rel
+5	ADDL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	SUBL A, A, RL2, @RW5+d8	RW5, @RW5+d8 #16, rel	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	CMPL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ANDL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	ORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	XORL A, A, RL2, @RW5+d8	R5, @RW5+d8 #8, rel	R5, @RW5+d8 #8, rel
+6	ADDL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	SUBL A, A, RL3, @RW6+d8	RW6, @RW6+d8 #16, rel	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	CMPL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ANDL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	ORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	XORL A, A, RL3, @RW6+d8	R6, @RW6+d8 #8, rel	R6, @RW6+d8 #8, rel
+7	ADDL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	SUBL A, A, RL3, @RW7+d8	RW7, @RW7+d8 #16, rel	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	CMPL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ANDL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	ORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	XORL A, A, RL3, @RW7+d8	R7, @RW7+d8 #8, rel	R7, @RW7+d8 #8, rel
+8	ADDL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	SUBL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #16, rel	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	CMPL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ANDL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	ORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	XORL A, A, @RW0, @RW0+d16	@RW0, @RW0+d16 #8, rel	@RW0, @RW0+d16 #8, rel
+9	ADDL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	SUBL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #16, rel	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	CMPL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ANDL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	ORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	XORL A, A, @RW1, @RW1+d16	@RW1, @RW1+d16 #8, rel	@RW1, @RW1+d16 #8, rel
+A	ADDL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	SUBL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #16, rel	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	CMPL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ANDL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	ORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	XORL A, A, @RW2, @RW2+d16	@RW2, @RW2+d16 #8, rel	@RW2, @RW2+d16 #8, rel
+B	ADDL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	SUBL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #16, rel	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	CMPL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ANDL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	ORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	XORL A, A, @RW3, @RW3+d16	@RW3, @RW3+d16 #8, rel	@RW3, @RW3+d16 #8, rel
+C	ADDL A, A, @RW0+@RW0-RW7	SUBL A, A, @RW0+@RW0-RW7	SUBL A, A, @RW0+@RW0-RW7	SUBL A, A, @RW0+@RW0-RW7	@RW0+@RW0-RW7 #16, rel	CMPL A, A, @RW0+@RW0-RW7	CMPL A, A, @RW0+@RW0-RW7	CMPL A, A, @RW0+@RW0-RW7	ANDL A, A, @RW0+@RW0-RW7	ANDL A, A, @RW0+@RW0-RW7	ORL A, A, @RW0+@RW0-RW7	ORL A, A, @RW0+@RW0-RW7	XORL A, A, @RW0+@RW0-RW7	XORL A, A, @RW0+@RW0-RW7	@RW0+@RW0-RW7 #8, rel	@RW0+@RW0-RW7 #8, rel
+D	ADDL A, A, @RW1+@RW1-RW7	SUBL A, A, @RW1+@RW1-RW7	SUBL A, A, @RW1+@RW1-RW7	SUBL A, A, @RW1+@RW1-RW7	@RW1+@RW1-RW7 #16, rel	CMPL A, A, @RW1+@RW1-RW7	CMPL A, A, @RW1+@RW1-RW7	CMPL A, A, @RW1+@RW1-RW7	ANDL A, A, @RW1+@RW1-RW7	ANDL A, A, @RW1+@RW1-RW7	ORL A, A, @RW1+@RW1-RW7	ORL A, A, @RW1+@RW1-RW7	XORL A, A, @RW1+@RW1-RW7	XORL A, A, @RW1+@RW1-RW7	@RW1+@RW1-RW7 #8, rel	@RW1+@RW1-RW7 #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 #16, rel	CMPL A, A, @PC+d16	CMPL A, A, @PC+d16	CMPL A, A, @PC+d16	ANDL A, A, @PC+d16	ANDL A, A, @PC+d16	ORL A, A, @PC+d16	ORL A, A, @PC+d16	XORL A, A, @PC+d16	XORL A, A, @PC+d16	@PC+d16 #8, rel	@PC+d16 #8, rel
+F	ADDL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	SUBL A, A, @RW3+addr16	addr16 #16, rel	CMPL A, A, @RW3+addr16	CMPL A, A, @RW3+addr16	CMPL A, A, @RW3+addr16	ANDL A, A, @RW3+addr16	ANDL A, A, @RW3+addr16	ORL A, A, @RW3+addr16	ORL A, A, @RW3+addr16	XORL A, A, @RW3+addr16	XORL A, A, @RW3+addr16	addr16 #8, rel	addr16 #8, rel

表 B.9-7 ea 系命令 その 2 (第 1 バイト = 71<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMPP @ RL0, @ @RW0+d8	JMPP @ RL0, @ @RW0+d8	CALLP @ RL0, @ @RW0+d8	CALLP @ RL0, @ @RW0+d8	INCL RL0, @RW0+d8	INCL RL0, @RW0+d8	DECL RL0, @RW0+d8	DECL RL0, @RW0+d8	MOVL A, RL0, @RW0+d8	MOVL A, RL0, @RW0+d8	MOVL A, RL0, @RW0+d8	MOVL A, RL0, @RW0+d8	MOV R0, #8, @RW0+d8, #8	MOV R0, #8, @RW0+d8, #8	MOVEA A, RW0, @RW0+d8	MOVEA A, RW0, @RW0+d8
+1	JMPP @ RL0, @ @RW1+d8	JMPP @ RL0, @ @RW1+d8	CALLP @ RL0, @ @RW1+d8	CALLP @ RL0, @ @RW1+d8	INCL RL0, @RW1+d8	INCL RL0, @RW1+d8	DECL RL0, @RW1+d8	DECL RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOVL A, RL0, @RW1+d8	MOV R1, #8, @RW1+d8, #8	MOV R1, #8, @RW1+d8, #8	MOVEA A, RW1, @RW1+d8	MOVEA A, RW1, @RW1+d8
+2	JMPP @ RL1, @ @RW2+d8	JMPP @ RL1, @ @RW2+d8	CALLP @ RL1, @ @RW2+d8	CALLP @ RL1, @ @RW2+d8	INCL RL1, @RW2+d8	INCL RL1, @RW2+d8	DECL RL1, @RW2+d8	DECL RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOVL A, RL1, @RW2+d8	MOV R2, #8, @RW2+d8, #8	MOV R2, #8, @RW2+d8, #8	MOVEA A, RW2, @RW2+d8	MOVEA A, RW2, @RW2+d8
+3	JMPP @ RL1, @ @RW3+d8	JMPP @ RL1, @ @RW3+d8	CALLP @ RL1, @ @RW3+d8	CALLP @ RL1, @ @RW3+d8	INCL RL1, @RW3+d8	INCL RL1, @RW3+d8	DECL RL1, @RW3+d8	DECL RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOVL A, RL1, @RW3+d8	MOV R3, #8, @RW3+d8, #8	MOV R3, #8, @RW3+d8, #8	MOVEA A, RW3, @RW3+d8	MOVEA A, RW3, @RW3+d8
+4	JMPP @ RL2, @ @RW4+d8	JMPP @ RL2, @ @RW4+d8	CALLP @ RL2, @ @RW4+d8	CALLP @ RL2, @ @RW4+d8	INCL RL2, @RW4+d8	INCL RL2, @RW4+d8	DECL RL2, @RW4+d8	DECL RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOVL A, RL2, @RW4+d8	MOV R4, #8, @RW4+d8, #8	MOV R4, #8, @RW4+d8, #8	MOVEA A, RW4, @RW4+d8	MOVEA A, RW4, @RW4+d8
+5	JMPP @ RL2, @ @RW5+d8	JMPP @ RL2, @ @RW5+d8	CALLP @ RL2, @ @RW5+d8	CALLP @ RL2, @ @RW5+d8	INCL RL2, @RW5+d8	INCL RL2, @RW5+d8	DECL RL2, @RW5+d8	DECL RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOVL A, RL2, @RW5+d8	MOV R5, #8, @RW5+d8, #8	MOV R5, #8, @RW5+d8, #8	MOVEA A, RW5, @RW5+d8	MOVEA A, RW5, @RW5+d8
+6	JMPP @ RL3, @ @RW6+d8	JMPP @ RL3, @ @RW6+d8	CALLP @ RL3, @ @RW6+d8	CALLP @ RL3, @ @RW6+d8	INCL RL3, @RW6+d8	INCL RL3, @RW6+d8	DECL RL3, @RW6+d8	DECL RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOVL A, RL3, @RW6+d8	MOV R6, #8, @RW6+d8, #8	MOV R6, #8, @RW6+d8, #8	MOVEA A, RW6, @RW6+d8	MOVEA A, RW6, @RW6+d8
+7	JMPP @ RL3, @ @RW7+d8	JMPP @ RL3, @ @RW7+d8	CALLP @ RL3, @ @RW7+d8	CALLP @ RL3, @ @RW7+d8	INCL RL3, @RW7+d8	INCL RL3, @RW7+d8	DECL RL3, @RW7+d8	DECL RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOVL A, RL3, @RW7+d8	MOV R7, #8, @RW7+d8, #8	MOV R7, #8, @RW7+d8, #8	MOVEA A, RW7, @RW7+d8	MOVEA A, RW7, @RW7+d8
+8	JMPP @ @ @RW0, @ @RW0+d16	JMPP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	CALLP @ @ @RW0, @ @RW0+d16	INCL @RW0, @RW0+d16	INCL @RW0, @RW0+d16	DECL @RW0, @RW0+d16	DECL @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOVL A, @RW0, @RW0+d16	MOV @RW0, #8, @RW0+d16, #8	MOV @RW0, #8, @RW0+d16, #8	MOVEA A, @RW0, @RW0+d16	MOVEA A, @RW0, @RW0+d16
+9	JMPP @ @ @RW1, @ @RW1+d16	JMPP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	CALLP @ @ @RW1, @ @RW1+d16	INCL @RW1, @RW1+d16	INCL @RW1, @RW1+d16	DECL @RW1, @RW1+d16	DECL @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOVL A, @RW1, @RW1+d16	MOV @RW1, #8, @RW1+d16, #8	MOV @RW1, #8, @RW1+d16, #8	MOVEA A, @RW1, @RW1+d16	MOVEA A, @RW1, @RW1+d16
+A	JMPP @ @ @RW2, @ @RW2+d16	JMPP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	CALLP @ @ @RW2, @ @RW2+d16	INCL @RW2, @RW2+d16	INCL @RW2, @RW2+d16	DECL @RW2, @RW2+d16	DECL @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOVL A, @RW2, @RW2+d16	MOV @RW2, #8, @RW2+d16, #8	MOV @RW2, #8, @RW2+d16, #8	MOVEA A, @RW2, @RW2+d16	MOVEA A, @RW2, @RW2+d16
+B	JMPP @ @ @RW3, @ @RW3+d16	JMPP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	CALLP @ @ @RW3, @ @RW3+d16	INCL @RW3, @RW3+d16	INCL @RW3, @RW3+d16	DECL @RW3, @RW3+d16	DECL @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOVL A, @RW3, @RW3+d16	MOV @RW3, #8, @RW3+d16, #8	MOV @RW3, #8, @RW3+d16, #8	MOVEA A, @RW3, @RW3+d16	MOVEA A, @RW3, @RW3+d16
+C	JMPP @ @ @RW0+, @ @RW0+RW7	JMPP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	CALLP @ @ @RW0+, @ @RW0+RW7	INCL @RW0+, @RW0+RW7	INCL @RW0+, @RW0+RW7	DECL @RW0+, @RW0+RW7	DECL @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOVL A, @RW0+, @RW0+RW7	MOV @RW0+, #8, @RW0+RW7, #8	MOV @RW0+, #8, @RW0+RW7, #8	MOVEA A, @RW0+, @RW0+RW7	MOVEA A, @RW0+, @RW0+RW7
+D	JMPP @ @ @RW1+, @ @RW1+RW7	JMPP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	CALLP @ @ @RW1+, @ @RW1+RW7	INCL @RW1+, @RW1+RW7	INCL @RW1+, @RW1+RW7	DECL @RW1+, @RW1+RW7	DECL @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOVL A, @RW1+, @RW1+RW7	MOV @RW1+, #8, @RW1+RW7, #8	MOV @RW1+, #8, @RW1+RW7, #8	MOVEA A, @RW1+, @RW1+RW7	MOVEA A, @RW1+, @RW1+RW7
+E	JMPP @ @ @PC+d16	JMPP @ @ @PC+d16	CALLP @ @ @PC+d16	CALLP @ @ @PC+d16	INCL @RW2+, @PC+d16	INCL @RW2+, @PC+d16	DECL @RW2+, @PC+d16	DECL @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOVL A, @RW2+, @PC+d16	MOV @RW2+, #8, @PC+d16, #8	MOV @RW2+, #8, @PC+d16, #8	MOVEA A, @RW2+, @PC+d16	MOVEA A, @RW2+, @PC+d16
+F	JMPP @ @ @RW3+, @ addr16	JMPP @ @ @RW3+, @ addr16	CALLP @ @ @RW3+, @ addr16	CALLP @ @ @RW3+, @ addr16	INCL @RW3+, @addr16	INCL @RW3+, @addr16	DECL @RW3+, @addr16	DECL @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOVL A, @RW3+, @addr16	MOV @RW3+, #8, @addr16, #8	MOV @RW3+, #8, @addr16, #8	MOVEA A, @RW3+, @addr16	MOVEA A, @RW3+, @addr16

表 B.9-8 ea 系命令 その 3 (第 1 バイト = 72<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ROLc R0' @RW0+d8	ROLc R0' @RW0+d8	RORc R0' @RW0+d8	RORc R0' @RW0+d8	INC R0' @RW0+d8	INC R0' @RW0+d8	DEC R0' @RW0+d8	DEC R0' @RW0+d8	MOV A, R0' @RW0+d8	MOV A, R0' @RW0+d8	MOV R0, A' @RW0+d8,A	MOV A, R0' @RW0+d8,A	MOVX A, R0' @RW0+d8	MOVX A, R0' @RW0+d8	XCH A, R0' @RW0+d8	XCH A, R0' @RW0+d8
+1	ROLc R1' @RW1+d8	ROLc R1' @RW1+d8	RORc R1' @RW1+d8	RORc R1' @RW1+d8	INC R1' @RW1+d8	INC R1' @RW1+d8	DEC R1' @RW1+d8	DEC R1' @RW1+d8	MOV A, R1' @RW1+d8	MOV A, R1' @RW1+d8	MOV R1, A' @RW1+d8,A	MOV A, R1' @RW1+d8,A	MOVX A, R1' @RW1+d8	MOVX A, R1' @RW1+d8	XCH A, R1' @RW1+d8	XCH A, R1' @RW1+d8
+2	ROLc R2' @RW2+d8	ROLc R2' @RW2+d8	RORc R2' @RW2+d8	RORc R2' @RW2+d8	INC R2' @RW2+d8	INC R2' @RW2+d8	DEC R2' @RW2+d8	DEC R2' @RW2+d8	MOV A, R2' @RW2+d8	MOV A, R2' @RW2+d8	MOV R2, A' @RW2+d8,A	MOV A, R2' @RW2+d8,A	MOVX A, R2' @RW2+d8	MOVX A, R2' @RW2+d8	XCH A, R2' @RW2+d8	XCH A, R2' @RW2+d8
+3	ROLc R3' @RW3+d8	ROLc R3' @RW3+d8	RORc R3' @RW3+d8	RORc R3' @RW3+d8	INC R3' @RW3+d8	INC R3' @RW3+d8	DEC R3' @RW3+d8	DEC R3' @RW3+d8	MOV A, R3' @RW3+d8	MOV A, R3' @RW3+d8	MOV R3, A' @RW3+d8,A	MOV A, R3' @RW3+d8,A	MOVX A, R3' @RW3+d8	MOVX A, R3' @RW3+d8	XCH A, R3' @RW3+d8	XCH A, R3' @RW3+d8
+4	ROLc R4' @RW4+d8	ROLc R4' @RW4+d8	RORc R4' @RW4+d8	RORc R4' @RW4+d8	INC R4' @RW4+d8	INC R4' @RW4+d8	DEC R4' @RW4+d8	DEC R4' @RW4+d8	MOV A, R4' @RW4+d8	MOV A, R4' @RW4+d8	MOV R4, A' @RW4+d8,A	MOV A, R4' @RW4+d8,A	MOVX A, R4' @RW4+d8	MOVX A, R4' @RW4+d8	XCH A, R4' @RW4+d8	XCH A, R4' @RW4+d8
+5	ROLc R5' @RW5+d8	ROLc R5' @RW5+d8	RORc R5' @RW5+d8	RORc R5' @RW5+d8	INC R5' @RW5+d8	INC R5' @RW5+d8	DEC R5' @RW5+d8	DEC R5' @RW5+d8	MOV A, R5' @RW5+d8	MOV A, R5' @RW5+d8	MOV R5, A' @RW5+d8,A	MOV A, R5' @RW5+d8,A	MOVX A, R5' @RW5+d8	MOVX A, R5' @RW5+d8	XCH A, R5' @RW5+d8	XCH A, R5' @RW5+d8
+6	ROLc R6' @RW6+d8	ROLc R6' @RW6+d8	RORc R6' @RW6+d8	RORc R6' @RW6+d8	INC R6' @RW6+d8	INC R6' @RW6+d8	DEC R6' @RW6+d8	DEC R6' @RW6+d8	MOV A, R6' @RW6+d8	MOV A, R6' @RW6+d8	MOV R6, A' @RW6+d8,A	MOV A, R6' @RW6+d8,A	MOVX A, R6' @RW6+d8	MOVX A, R6' @RW6+d8	XCH A, R6' @RW6+d8	XCH A, R6' @RW6+d8
+7	ROLc R7' @RW7+d8	ROLc R7' @RW7+d8	RORc R7' @RW7+d8	RORc R7' @RW7+d8	INC R7' @RW7+d8	INC R7' @RW7+d8	DEC R7' @RW7+d8	DEC R7' @RW7+d8	MOV A, R7' @RW7+d8	MOV A, R7' @RW7+d8	MOV R7, A' @RW7+d8,A	MOV A, R7' @RW7+d8,A	MOVX A, R7' @RW7+d8	MOVX A, R7' @RW7+d8	XCH A, R7' @RW7+d8	XCH A, R7' @RW7+d8
+8	ROLc @RW0, @RW0+d16	ROLc @RW0, @RW0+d16	RORc @RW0, @RW0+d16	RORc @RW0, @RW0+d16	INC @RW0, @RW0+d16	INC @RW0, @RW0+d16	DEC @RW0, @RW0+d16	DEC @RW0, @RW0+d16	MOV A, @RW0, @RW0+d16	MOV A, @RW0, @RW0+d16	MOV @RW0, A' @RW0+d16,A	MOV A, @RW0+d16,A	MOVX A, @RW0, @RW0+d16	MOVX A, @RW0, @RW0+d16	XCH A, @RW0, @RW0+d16	XCH A, @RW0, @RW0+d16
+9	ROLc @RW1, @RW1+d16	ROLc @RW1, @RW1+d16	RORc @RW1, @RW1+d16	RORc @RW1, @RW1+d16	INC @RW1, @RW1+d16	INC @RW1, @RW1+d16	DEC @RW1, @RW1+d16	DEC @RW1, @RW1+d16	MOV A, @RW1, @RW1+d16	MOV A, @RW1, @RW1+d16	MOV @RW1, A' @RW1+d16,A	MOV A, @RW1+d16,A	MOVX A, @RW1, @RW1+d16	MOVX A, @RW1, @RW1+d16	XCH A, @RW1, @RW1+d16	XCH A, @RW1, @RW1+d16
+A	ROLc @RW2, @RW2+d16	ROLc @RW2, @RW2+d16	RORc @RW2, @RW2+d16	RORc @RW2, @RW2+d16	INC @RW2, @RW2+d16	INC @RW2, @RW2+d16	DEC @RW2, @RW2+d16	DEC @RW2, @RW2+d16	MOV A, @RW2, @RW2+d16	MOV A, @RW2, @RW2+d16	MOV @RW2, A' @RW2+d16,A	MOV A, @RW2+d16,A	MOVX A, @RW2, @RW2+d16	MOVX A, @RW2, @RW2+d16	XCH A, @RW2, @RW2+d16	XCH A, @RW2, @RW2+d16
+B	ROLc @RW3, @RW3+d16	ROLc @RW3, @RW3+d16	RORc @RW3, @RW3+d16	RORc @RW3, @RW3+d16	INC @RW3, @RW3+d16	INC @RW3, @RW3+d16	DEC @RW3, @RW3+d16	DEC @RW3, @RW3+d16	MOV A, @RW3, @RW3+d16	MOV A, @RW3, @RW3+d16	MOV @RW3, A' @RW3+d16,A	MOV A, @RW3+d16,A	MOVX A, @RW3, @RW3+d16	MOVX A, @RW3, @RW3+d16	XCH A, @RW3, @RW3+d16	XCH A, @RW3, @RW3+d16
+C	ROLc @RW0+, @RW0+RW7	ROLc @RW0+, @RW0+RW7	RORc @RW0+, @RW0+RW7	RORc @RW0+, @RW0+RW7	INC @RW0+, @RW0+RW7	INC @RW0+, @RW0+RW7	DEC @RW0+, @RW0+RW7	DEC @RW0+, @RW0+RW7	MOV A, @RW0+, @RW0+RW7	MOV A, @RW0+, @RW0+RW7	MOV @RW0+, A' @RW0+RW7,A	MOV A, @RW0+RW7,A	MOVX A, @RW0+, @RW0+RW7	MOVX A, @RW0+, @RW0+RW7	XCH A, @RW0+, @RW0+RW7	XCH A, @RW0+, @RW0+RW7
+D	ROLc @RW1+, @RW1+RW7	ROLc @RW1+, @RW1+RW7	RORc @RW1+, @RW1+RW7	RORc @RW1+, @RW1+RW7	INC @RW1+, @RW1+RW7	INC @RW1+, @RW1+RW7	DEC @RW1+, @RW1+RW7	DEC @RW1+, @RW1+RW7	MOV A, @RW1+, @RW1+RW7	MOV A, @RW1+, @RW1+RW7	MOV @RW1+, A' @RW1+RW7,A	MOV A, @RW1+RW7,A	MOVX A, @RW1+, @RW1+RW7	MOVX A, @RW1+, @RW1+RW7	XCH A, @RW1+, @RW1+RW7	XCH A, @RW1+, @RW1+RW7
+E	ROLc @RW2+, @PC+d16	ROLc @PC+d16	RORc @RW2+, @PC+d16	RORc @RW2+, @PC+d16	INC @RW2+, @PC+d16	INC @RW2+, @PC+d16	DEC @RW2+, @PC+d16	DEC @RW2+, @PC+d16	MOV A, @RW2+, @PC+d16	MOV A, @RW2+, @PC+d16	MOV @RW2+, A' @PC+d16, A	MOV A, @PC+d16, A	MOVX A, @RW2+, @PC+d16	MOVX A, @RW2+, @PC+d16	XCH A, @RW2+, @PC+d16	XCH A, @RW2+, @PC+d16
+F	ROLc @RW3+, addr16	ROLc addr16	RORc @RW3+, addr16	RORc @RW3+, addr16	INC @RW3+, addr16	INC @RW3+, addr16	DEC @RW3+, addr16	DEC @RW3+, addr16	MOV A, @RW3+, addr16	MOV A, @RW3+, addr16	MOV @RW3+, A' addr16, A	MOV A, addr16, A	MOVX A, @RW3+, addr16	MOVX A, @RW3+, addr16	XCH A, @RW3+, addr16	XCH A, @RW3+, addr16

表 B.9-9 ea 系命令 その 4 (第 1 バイト = 73<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	JMP @ RW0, @ RW0+d8	JMP @ RW0, @ RW0+d8	CALL RW0, @ RW0+d8	CALL RW0, @ RW0+d8	INCW RW0, @ RW0+d8	INCW RW0, @ RW0+d8	DECW RW0, @ RW0+d8	DECW RW0, @ RW0+d8	MOVW A, RW0, @ RW0+d8	MOVW A, RW0, @ RW0+d8	MOVW RW0, #16, @ RW0+d8, #16	MOVW RW0, #16, @ RW0+d8, #16	MOVW RW0, #16, @ RW0+d8, #16	MOVW RW0, #16, @ RW0+d8, #16	XCHW A, RW0, @ RW0+d8	XCHW A, RW0, @ RW0+d8
+1	JMP @ RW1, @ RW1+d8	JMP @ RW1, @ RW1+d8	CALL RW1, @ RW1+d8	CALL RW1, @ RW1+d8	INCW RW1, @ RW1+d8	INCW RW1, @ RW1+d8	DECW RW1, @ RW1+d8	DECW RW1, @ RW1+d8	MOVW A, RW1, @ RW1+d8	MOVW A, RW1, @ RW1+d8	MOVW RW1, #16, @ RW1+d8, #16	MOVW RW1, #16, @ RW1+d8, #16	MOVW RW1, #16, @ RW1+d8, #16	MOVW RW1, #16, @ RW1+d8, #16	XCHW A, RW1, @ RW1+d8	XCHW A, RW1, @ RW1+d8
+2	JMP @ RW2, @ RW2+d8	JMP @ RW2, @ RW2+d8	CALL RW2, @ RW2+d8	CALL RW2, @ RW2+d8	INCW RW2, @ RW2+d8	INCW RW2, @ RW2+d8	DECW RW2, @ RW2+d8	DECW RW2, @ RW2+d8	MOVW A, RW2, @ RW2+d8	MOVW A, RW2, @ RW2+d8	MOVW RW2, #16, @ RW2+d8, #16	MOVW RW2, #16, @ RW2+d8, #16	MOVW RW2, #16, @ RW2+d8, #16	MOVW RW2, #16, @ RW2+d8, #16	XCHW A, RW2, @ RW2+d8	XCHW A, RW2, @ RW2+d8
+3	JMP @ RW3, @ RW3+d8	JMP @ RW3, @ RW3+d8	CALL RW3, @ RW3+d8	CALL RW3, @ RW3+d8	INCW RW3, @ RW3+d8	INCW RW3, @ RW3+d8	DECW RW3, @ RW3+d8	DECW RW3, @ RW3+d8	MOVW A, RW3, @ RW3+d8	MOVW A, RW3, @ RW3+d8	MOVW RW3, #16, @ RW3+d8, #16	MOVW RW3, #16, @ RW3+d8, #16	MOVW RW3, #16, @ RW3+d8, #16	MOVW RW3, #16, @ RW3+d8, #16	XCHW A, RW3, @ RW3+d8	XCHW A, RW3, @ RW3+d8
+4	JMP @ RW4, @ RW4+d8	JMP @ RW4, @ RW4+d8	CALL RW4, @ RW4+d8	CALL RW4, @ RW4+d8	INCW RW4, @ RW4+d8	INCW RW4, @ RW4+d8	DECW RW4, @ RW4+d8	DECW RW4, @ RW4+d8	MOVW A, RW4, @ RW4+d8	MOVW A, RW4, @ RW4+d8	MOVW RW4, #16, @ RW4+d8, #16	MOVW RW4, #16, @ RW4+d8, #16	MOVW RW4, #16, @ RW4+d8, #16	MOVW RW4, #16, @ RW4+d8, #16	XCHW A, RW4, @ RW4+d8	XCHW A, RW4, @ RW4+d8
+5	JMP @ RW5, @ RW5+d8	JMP @ RW5, @ RW5+d8	CALL RW5, @ RW5+d8	CALL RW5, @ RW5+d8	INCW RW5, @ RW5+d8	INCW RW5, @ RW5+d8	DECW RW5, @ RW5+d8	DECW RW5, @ RW5+d8	MOVW A, RW5, @ RW5+d8	MOVW A, RW5, @ RW5+d8	MOVW RW5, #16, @ RW5+d8, #16	MOVW RW5, #16, @ RW5+d8, #16	MOVW RW5, #16, @ RW5+d8, #16	MOVW RW5, #16, @ RW5+d8, #16	XCHW A, RW5, @ RW5+d8	XCHW A, RW5, @ RW5+d8
+6	JMP @ RW6, @ RW6+d8	JMP @ RW6, @ RW6+d8	CALL RW6, @ RW6+d8	CALL RW6, @ RW6+d8	INCW RW6, @ RW6+d8	INCW RW6, @ RW6+d8	DECW RW6, @ RW6+d8	DECW RW6, @ RW6+d8	MOVW A, RW6, @ RW6+d8	MOVW A, RW6, @ RW6+d8	MOVW RW6, #16, @ RW6+d8, #16	MOVW RW6, #16, @ RW6+d8, #16	MOVW RW6, #16, @ RW6+d8, #16	MOVW RW6, #16, @ RW6+d8, #16	XCHW A, RW6, @ RW6+d8	XCHW A, RW6, @ RW6+d8
+7	JMP @ RW7, @ RW7+d8	JMP @ RW7, @ RW7+d8	CALL RW7, @ RW7+d8	CALL RW7, @ RW7+d8	INCW RW7, @ RW7+d8	INCW RW7, @ RW7+d8	DECW RW7, @ RW7+d8	DECW RW7, @ RW7+d8	MOVW A, RW7, @ RW7+d8	MOVW A, RW7, @ RW7+d8	MOVW RW7, #16, @ RW7+d8, #16	MOVW RW7, #16, @ RW7+d8, #16	MOVW RW7, #16, @ RW7+d8, #16	MOVW RW7, #16, @ RW7+d8, #16	XCHW A, RW7, @ RW7+d8	XCHW A, RW7, @ RW7+d8
+8	JMP @ RW0, @ RW0+d16	JMP @ RW0, @ RW0+d16	CALL @ RW0, @ RW0+d16	CALL @ RW0, @ RW0+d16	INCW @ RW0, @ RW0+d16	INCW @ RW0, @ RW0+d16	DECW @ RW0, @ RW0+d16	DECW @ RW0, @ RW0+d16	MOVW A, @ RW0, @ RW0+d16	MOVW A, @ RW0, @ RW0+d16	MOVW @ RW0, #16, @ RW0+d16, #16	MOVW @ RW0, #16, @ RW0+d16, #16	MOVW @ RW0, #16, @ RW0+d16, #16	MOVW @ RW0, #16, @ RW0+d16, #16	XCHW A, @ RW0, @ RW0+d16	XCHW A, @ RW0, @ RW0+d16
+9	JMP @ RW1, @ RW1+d16	JMP @ RW1, @ RW1+d16	CALL @ RW1, @ RW1+d16	CALL @ RW1, @ RW1+d16	INCW @ RW1, @ RW1+d16	INCW @ RW1, @ RW1+d16	DECW @ RW1, @ RW1+d16	DECW @ RW1, @ RW1+d16	MOVW A, @ RW1, @ RW1+d16	MOVW A, @ RW1, @ RW1+d16	MOVW @ RW1, #16, @ RW1+d16, #16	MOVW @ RW1, #16, @ RW1+d16, #16	MOVW @ RW1, #16, @ RW1+d16, #16	MOVW @ RW1, #16, @ RW1+d16, #16	XCHW A, @ RW1, @ RW1+d16	XCHW A, @ RW1, @ RW1+d16
+A	JMP @ RW2, @ RW2+d16	JMP @ RW2, @ RW2+d16	CALL @ RW2, @ RW2+d16	CALL @ RW2, @ RW2+d16	INCW @ RW2, @ RW2+d16	INCW @ RW2, @ RW2+d16	DECW @ RW2, @ RW2+d16	DECW @ RW2, @ RW2+d16	MOVW A, @ RW2, @ RW2+d16	MOVW A, @ RW2, @ RW2+d16	MOVW @ RW2, #16, @ RW2+d16, #16	MOVW @ RW2, #16, @ RW2+d16, #16	MOVW @ RW2, #16, @ RW2+d16, #16	MOVW @ RW2, #16, @ RW2+d16, #16	XCHW A, @ RW2, @ RW2+d16	XCHW A, @ RW2, @ RW2+d16
+B	JMP @ RW3, @ RW3+d16	JMP @ RW3, @ RW3+d16	CALL @ RW3, @ RW3+d16	CALL @ RW3, @ RW3+d16	INCW @ RW3, @ RW3+d16	INCW @ RW3, @ RW3+d16	DECW @ RW3, @ RW3+d16	DECW @ RW3, @ RW3+d16	MOVW A, @ RW3, @ RW3+d16	MOVW A, @ RW3, @ RW3+d16	MOVW @ RW3, #16, @ RW3+d16, #16	MOVW @ RW3, #16, @ RW3+d16, #16	MOVW @ RW3, #16, @ RW3+d16, #16	MOVW @ RW3, #16, @ RW3+d16, #16	XCHW A, @ RW3, @ RW3+d16	XCHW A, @ RW3, @ RW3+d16
+C	JMP @ RW0+, @ RW0+RW7	JMP @ RW0+, @ RW0+RW7	CALL @ RW0+, @ RW0+RW7	CALL @ RW0+, @ RW0+RW7	INCW @ RW0+, @ RW0+RW7	INCW @ RW0+, @ RW0+RW7	DECW @ RW0+, @ RW0+RW7	DECW @ RW0+, @ RW0+RW7	MOVW A, @ RW0+, @ RW0+RW7	MOVW A, @ RW0+, @ RW0+RW7	MOVW @ RW0+, #16, @ RW0+RW7, #16	MOVW @ RW0+, #16, @ RW0+RW7, #16	MOVW @ RW0+, #16, @ RW0+RW7, #16	MOVW @ RW0+, #16, @ RW0+RW7, #16	XCHW A, @ RW0+, @ RW0+RW7	XCHW A, @ RW0+, @ RW0+RW7
+D	JMP @ RW1+, @ RW1+RW7	JMP @ RW1+, @ RW1+RW7	CALL @ RW1+, @ RW1+RW7	CALL @ RW1+, @ RW1+RW7	INCW @ RW1+, @ RW1+RW7	INCW @ RW1+, @ RW1+RW7	DECW @ RW1+, @ RW1+RW7	DECW @ RW1+, @ RW1+RW7	MOVW A, @ RW1+, @ RW1+RW7	MOVW A, @ RW1+, @ RW1+RW7	MOVW @ RW1+, #16, @ RW1+RW7, #16	MOVW @ RW1+, #16, @ RW1+RW7, #16	MOVW @ RW1+, #16, @ RW1+RW7, #16	MOVW @ RW1+, #16, @ RW1+RW7, #16	XCHW A, @ RW1+, @ RW1+RW7	XCHW A, @ RW1+, @ RW1+RW7
+E	JMP @ RW2+, @ PC+d16	JMP @ RW2+, @ PC+d16	CALL @ RW2+, @ PC+d16	CALL @ RW2+, @ PC+d16	INCW @ RW2+, @ PC+d16	INCW @ RW2+, @ PC+d16	DECW @ RW2+, @ PC+d16	DECW @ RW2+, @ PC+d16	MOVW A, @ RW2+, @ PC+d16	MOVW A, @ RW2+, @ PC+d16	MOVW @ RW2+, #16, @ PC+d16, #16	MOVW @ RW2+, #16, @ PC+d16, #16	MOVW @ RW2+, #16, @ PC+d16, #16	MOVW @ RW2+, #16, @ PC+d16, #16	XCHW A, @ RW2+, @ PC+d16	XCHW A, @ RW2+, @ PC+d16
+F	JMP @ RW3+, @ addr16	JMP @ RW3+, @ addr16	CALL @ RW3+, @ addr16	CALL @ RW3+, @ addr16	INCW @ RW3+, @ addr16	INCW @ RW3+, @ addr16	DECW @ RW3+, @ addr16	DECW @ RW3+, @ addr16	MOVW A, @ RW3+, @ addr16	MOVW A, @ RW3+, @ addr16	MOVW @ RW3+, #16, @ addr16, #16	MOVW @ RW3+, #16, @ addr16, #16	MOVW @ RW3+, #16, @ addr16, #16	MOVW @ RW3+, #16, @ addr16, #16	XCHW A, @ RW3+, @ addr16	XCHW A, @ RW3+, @ addr16

表 B.9-10 ea 系命令 その5 (第1バイト = 74<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD A, R0, @RW0+d8	ADD A, R0, @RW0+d8	SUB A, R0, @RW0+d8	SUB A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	ADDC A, R0, @RW0+d8	CMP A, R0, @RW0+d8	CMP A, R0, @RW0+d8	AND A, R0, @RW0+d8	AND A, R0, @RW0+d8	OR A, R0, @RW0+d8	OR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	XOR A, R0, @RW0+d8	DBNZ R0, @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<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADD R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUB R0, A', @RW0+d8, A	SUBC A, R0', @RW0+d8	SUBC A, R0', @RW0+d8	NEG R0', @RW0+d8	NEG A, R0', @RW0+d8	AND R0, A', @RW0+d8, A	AND R0, A', @RW0+d8, A	OR R0, A', @RW0+d8, A	OR R0, A', @RW0+d8, A	XOR R0, A', @RW0+d8, A	XOR R0, A', @RW0+d8, A	NOT R0', @RW0+d8	NOT R0', @RW0+d8
+1	ADD R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUB R1, A', @RW1+d8, A	SUBC A, R1', @RW1+d8	SUBC A, R1', @RW1+d8	NEG R1', @RW1+d8	NEG A, R1', @RW1+d8	AND R1, A', @RW1+d8, A	AND R1, A', @RW1+d8, A	OR R1, A', @RW1+d8, A	OR R1, A', @RW1+d8, A	XOR R1, A', @RW1+d8, A	XOR R1, A', @RW1+d8, A	NOT R1', @RW1+d8	NOT R1', @RW1+d8
+2	ADD R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUB R2, A', @RW2+d8, A	SUBC A, R2', @RW2+d8	SUBC A, R2', @RW2+d8	NEG R2', @RW2+d8	NEG A, R2', @RW2+d8	AND R2, A', @RW2+d8, A	AND R2, A', @RW2+d8, A	OR R2, A', @RW2+d8, A	OR R2, A', @RW2+d8, A	XOR R2, A', @RW2+d8, A	XOR R2, A', @RW2+d8, A	NOT R2', @RW2+d8	NOT R2', @RW2+d8
+3	ADD R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUB R3, A', @RW3+d8, A	SUBC A, R3', @RW3+d8	SUBC A, R3', @RW3+d8	NEG R3', @RW3+d8	NEG A, R3', @RW3+d8	AND R3, A', @RW3+d8, A	AND R3, A', @RW3+d8, A	OR R3, A', @RW3+d8, A	OR R3, A', @RW3+d8, A	XOR R3, A', @RW3+d8, A	XOR R3, A', @RW3+d8, A	NOT R3', @RW3+d8	NOT R3', @RW3+d8
+4	ADD R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUB R4, A', @RW4+d8, A	SUBC A, R4', @RW4+d8	SUBC A, R4', @RW4+d8	NEG R4', @RW4+d8	NEG A, R4', @RW4+d8	AND R4, A', @RW4+d8, A	AND R4, A', @RW4+d8, A	OR R4, A', @RW4+d8, A	OR R4, A', @RW4+d8, A	XOR R4, A', @RW4+d8, A	XOR R4, A', @RW4+d8, A	NOT R4', @RW4+d8	NOT R4', @RW4+d8
+5	ADD R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUB R5, A', @RW5+d8, A	SUBC A, R5', @RW5+d8	SUBC A, R5', @RW5+d8	NEG R5', @RW5+d8	NEG A, R5', @RW5+d8	AND R5, A', @RW5+d8, A	AND R5, A', @RW5+d8, A	OR R5, A', @RW5+d8, A	OR R5, A', @RW5+d8, A	XOR R5, A', @RW5+d8, A	XOR R5, A', @RW5+d8, A	NOT R5', @RW5+d8	NOT R5', @RW5+d8
+6	ADD R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUB R6, A', @RW6+d8, A	SUBC A, R6', @RW6+d8	SUBC A, R6', @RW6+d8	NEG R6', @RW6+d8	NEG A, R6', @RW6+d8	AND R6, A', @RW6+d8, A	AND R6, A', @RW6+d8, A	OR R6, A', @RW6+d8, A	OR R6, A', @RW6+d8, A	XOR R6, A', @RW6+d8, A	XOR R6, A', @RW6+d8, A	NOT R6', @RW6+d8	NOT R6', @RW6+d8
+7	ADD R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUB R7, A', @RW7+d8, A	SUBC A, R7', @RW7+d8	SUBC A, R7', @RW7+d8	NEG R7', @RW7+d8	NEG A, R7', @RW7+d8	AND R7, A', @RW7+d8, A	AND R7, A', @RW7+d8, A	OR R7, A', @RW7+d8, A	OR R7, A', @RW7+d8, A	XOR R7, A', @RW7+d8, A	XOR R7, A', @RW7+d8, A	NOT R7', @RW7+d8	NOT R7', @RW7+d8
+8	ADD @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUB @RW0, A', @RW0+d16, A	SUBC A, @RW0', @RW0+d16	SUBC A, @RW0', @RW0+d16	NEG @RW0', @RW0+d16	NEG A, @RW0', @RW0+d16	AND @RW0, A', @RW0+d16, A	AND @RW0, A', @RW0+d16, A	OR @RW0, A', @RW0+d16, A	OR @RW0, A', @RW0+d16, A	XOR @RW0, A', @RW0+d16, A	XOR @RW0, A', @RW0+d16, A	NOT @RW0', @RW0+d16	NOT @RW0', @RW0+d16
+9	ADD @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUB @RW1, A', @RW1+d16, A	SUBC A, @RW1', @RW1+d16	SUBC A, @RW1', @RW1+d16	NEG @RW1', @RW1+d16	NEG A, @RW1', @RW1+d16	AND @RW1, A', @RW1+d16, A	AND @RW1, A', @RW1+d16, A	OR @RW1, A', @RW1+d16, A	OR @RW1, A', @RW1+d16, A	XOR @RW1, A', @RW1+d16, A	XOR @RW1, A', @RW1+d16, A	NOT @RW1', @RW1+d16	NOT @RW1', @RW1+d16
+A	ADD @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUB @RW2, A', @RW2+d16, A	SUBC A, @RW2', @RW2+d16	SUBC A, @RW2', @RW2+d16	NEG @RW2', @RW2+d16	NEG A, @RW2', @RW2+d16	AND @RW2, A', @RW2+d16, A	AND @RW2, A', @RW2+d16, A	OR @RW2, A', @RW2+d16, A	OR @RW2, A', @RW2+d16, A	XOR @RW2, A', @RW2+d16, A	XOR @RW2, A', @RW2+d16, A	NOT @RW2', @RW2+d16	NOT @RW2', @RW2+d16
+B	ADD @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUB @RW3, A', @RW3+d16, A	SUBC A, @RW3', @RW3+d16	SUBC A, @RW3', @RW3+d16	NEG @RW3', @RW3+d16	NEG A, @RW3', @RW3+d16	AND @RW3, A', @RW3+d16, A	AND @RW3, A', @RW3+d16, A	OR @RW3, A', @RW3+d16, A	OR @RW3, A', @RW3+d16, A	XOR @RW3, A', @RW3+d16, A	XOR @RW3, A', @RW3+d16, A	NOT @RW3', @RW3+d16	NOT @RW3', @RW3+d16
+C	ADD @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUB @RW0+, A', @RW0+RW7, A	SUBC A, @RW0+', @RW0+RW7	SUBC A, @RW0+', @RW0+RW7	NEG @RW0+', @RW0+RW7	NEG A, @RW0+', @RW0+RW7	AND @RW0+, A', @RW0+RW7, A	AND @RW0+, A', @RW0+RW7, A	OR @RW0+, A', @RW0+RW7, A	OR @RW0+, A', @RW0+RW7, A	XOR @RW0+, A', @RW0+RW7, A	XOR @RW0+, A', @RW0+RW7, A	NOT @RW0+', @RW0+RW7	NOT @RW0+', @RW0+RW7
+D	ADD @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUB @RW1+, A', @RW1+RW7, A	SUBC A, @RW1+', @RW1+RW7	SUBC A, @RW1+', @RW1+RW7	NEG @RW1+', @RW1+RW7	NEG A, @RW1+', @RW1+RW7	AND @RW1+, A', @RW1+RW7, A	AND @RW1+, A', @RW1+RW7, A	OR @RW1+, A', @RW1+RW7, A	OR @RW1+, A', @RW1+RW7, A	XOR @RW1+, A', @RW1+RW7, A	XOR @RW1+, A', @RW1+RW7, A	NOT @RW1+', @RW1+RW7	NOT @RW1+', @RW1+RW7
+E	ADD @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUB @RW2+, A', @PC+d16, A	SUBC A, @RW2+', @PC+d16	SUBC A, @RW2+', @PC+d16	NEG @RW2+', @PC+d16	NEG A, @RW2+', @PC+d16	AND @RW2+, A', @PC+d16, A	AND @RW2+, A', @PC+d16, A	OR @RW2+, A', @PC+d16, A	OR @RW2+, A', @PC+d16, A	XOR @RW2+, A', @PC+d16, A	XOR @RW2+, A', @PC+d16, A	NOT @RW2+', @PC+d16	NOT @RW2+', @PC+d16
+F	ADD @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUB @RW3+, A', addr16, A	SUBC A, @RW3+', addr16	SUBC A, @RW3+', addr16	NEG @RW3+', addr16	NEG A, @RW3+', addr16	AND @RW3+, A', addr16, A	AND @RW3+, A', addr16, A	OR @RW3+, A', addr16, A	OR @RW3+, A', addr16, A	XOR @RW3+, A', addr16, A	XOR @RW3+, A', addr16, A	NOT @RW3+', addr16	NOT @RW3+', addr16



表 B.9-12 ea 系命令 その 7 (第 1 バイト = 76<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW A, RW0, @RW0+d8	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW 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	SUBW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	SUBW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	ADDCW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	CMPW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ANDW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	ORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	XORW A, @RW3+, addr 16	DWBZ @RW3+, r' addr 16, r	DWBZ @RW3+, r' addr 16, r

表 B.9-13 ea 系命令 その 8 (第 1 バイト = 77<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	ADDW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBW RW0, A', @RW0+d8, A	SUBCW A, RW0', @RW0+d8	SUBCW A, RW0', @RW0+d8	NEGW RW0', @RW0+d8	NEGW RW0', @RW0+d8	ANDW RW0, A', @RW0+d8, A	ANDW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	ORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	XORW RW0, A', @RW0+d8, A	NOTW RW0', @RW0+d8	NOTW RW0', @RW0+d8
+1	ADDW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBW RW1, A', @RW1+d8, A	SUBCW A, RW1', @RW1+d8	SUBCW A, RW1', @RW1+d8	NEGW RW1', @RW1+d8	NEGW RW1', @RW1+d8	ANDW RW1, A', @RW1+d8, A	ANDW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	ORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	XORW RW1, A', @RW1+d8, A	NOTW RW1', @RW1+d8	NOTW RW1', @RW1+d8
+2	ADDW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBW RW2, A', @RW2+d8, A	SUBCW A, RW2', @RW2+d8	SUBCW A, RW2', @RW2+d8	NEGW RW2', @RW2+d8	NEGW RW2', @RW2+d8	ANDW RW2, A', @RW2+d8, A	ANDW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	ORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	XORW RW2, A', @RW2+d8, A	NOTW RW2', @RW2+d8	NOTW RW2', @RW2+d8
+3	ADDW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBW RW3, A', @RW3+d8, A	SUBCW A, RW3', @RW3+d8	SUBCW A, RW3', @RW3+d8	NEGW RW3', @RW3+d8	NEGW RW3', @RW3+d8	ANDW RW3, A', @RW3+d8, A	ANDW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	ORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	XORW RW3, A', @RW3+d8, A	NOTW RW3', @RW3+d8	NOTW RW3', @RW3+d8
+4	ADDW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBW RW4, A', @RW4+d8, A	SUBCW A, RW4', @RW4+d8	SUBCW A, RW4', @RW4+d8	NEGW RW4', @RW4+d8	NEGW RW4', @RW4+d8	ANDW RW4, A', @RW4+d8, A	ANDW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	ORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	XORW RW4, A', @RW4+d8, A	NOTW RW4', @RW4+d8	NOTW RW4', @RW4+d8
+5	ADDW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBW RW5, A', @RW5+d8, A	SUBCW A, RW5', @RW5+d8	SUBCW A, RW5', @RW5+d8	NEGW RW5', @RW5+d8	NEGW RW5', @RW5+d8	ANDW RW5, A', @RW5+d8, A	ANDW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	ORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	XORW RW5, A', @RW5+d8, A	NOTW RW5', @RW5+d8	NOTW RW5', @RW5+d8
+6	ADDW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBW RW6, A', @RW6+d8, A	SUBCW A, RW6', @RW6+d8	SUBCW A, RW6', @RW6+d8	NEGW RW6', @RW6+d8	NEGW RW6', @RW6+d8	ANDW RW6, A', @RW6+d8, A	ANDW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	ORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	XORW RW6, A', @RW6+d8, A	NOTW RW6', @RW6+d8	NOTW RW6', @RW6+d8
+7	ADDW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBW RW7, A', @RW7+d8, A	SUBCW A, RW7', @RW7+d8	SUBCW A, RW7', @RW7+d8	NEGW RW7', @RW7+d8	NEGW RW7', @RW7+d8	ANDW RW7, A', @RW7+d8, A	ANDW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	ORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	XORW RW7, A', @RW7+d8, A	NOTW RW7', @RW7+d8	NOTW RW7', @RW7+d8
+8	ADDW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBW @RW0, A', @RW0+d16, A	SUBCW A, @RW0', @RW0+d16	SUBCW A, @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	NEGW @RW0', @RW0+d16	ANDW @RW0, A', @RW0+d16, A	ANDW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	ORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	XORW @RW0, A', @RW0+d16, A	NOTW @RW0', @RW0+d16	NOTW @RW0', @RW0+d16
+9	ADDW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBW @RW1, A', @RW1+d16, A	SUBCW A, @RW1', @RW1+d16	SUBCW A, @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	NEGW @RW1', @RW1+d16	ANDW @RW1, A', @RW1+d16, A	ANDW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	ORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	XORW @RW1, A', @RW1+d16, A	NOTW @RW1', @RW1+d16	NOTW @RW1', @RW1+d16
+A	ADDW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBW @RW2, A', @RW2+d16, A	SUBCW A, @RW2', @RW2+d16	SUBCW A, @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	NEGW @RW2', @RW2+d16	ANDW @RW2, A', @RW2+d16, A	ANDW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	ORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	XORW @RW2, A', @RW2+d16, A	NOTW @RW2', @RW2+d16	NOTW @RW2', @RW2+d16
+B	ADDW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBW @RW3, A', @RW3+d16, A	SUBCW A, @RW3', @RW3+d16	SUBCW A, @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	NEGW @RW3', @RW3+d16	ANDW @RW3, A', @RW3+d16, A	ANDW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	ORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	XORW @RW3, A', @RW3+d16, A	NOTW @RW3', @RW3+d16	NOTW @RW3', @RW3+d16
+C	ADDW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBW @RW0+, A', @RW0+RW7, A	SUBCW A, @RW0+', @RW0+RW7	SUBCW A, @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	NEGW @RW0+', @RW0+RW7	ANDW @RW0+, A', @RW0+RW7, A	ANDW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	ORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	XORW @RW0+, A', @RW0+RW7, A	NOTW @RW0+', @RW0+RW7	NOTW @RW0+', @RW0+RW7
+D	ADDW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBW @RW1+, A', @RW1+RW7, A	SUBCW A, @RW1+', @RW1+RW7	SUBCW A, @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	NEGW @RW1+', @RW1+RW7	ANDW @RW1+, A', @RW1+RW7, A	ANDW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	ORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	XORW @RW1+, A', @RW1+RW7, A	NOTW @RW1+', @RW1+RW7	NOTW @RW1+', @RW1+RW7
+E	ADDW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBW @RW2+, A', @PC+d16, A	SUBCW A, @RW2+', @PC+d16	SUBCW A, @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	NEGW @RW2+', @PC+d16	ANDW @RW2+, A', @PC+d16, A	ANDW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	ORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	XORW @RW2+, A', @PC+d16, A	NOTW @RW2+', @PC+d16	NOTW @RW2+', @PC+d16
+F	ADDW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBW @RW3+, A', addr16, A	SUBCW A, @RW3+', addr16	SUBCW A, @RW3+', addr16	NEGW @RW3+', addr16	NEGW @RW3+', addr16	ANDW @RW3+, A', addr16, A	ANDW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	ORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	XORW @RW3+, A', addr16, A	NOTW @RW3+', addr16	NOTW @RW3+', addr16

表 B.9-14 ea 系命令 その9 (第1バイト=78<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MULU A, R0', @RW0+d8	MULU A, MULW A, RW0', @RW0+d8	MULW A, RW0', @RW0+d8	MULW A, MULW A, RW0', @RW0+d8	MUL A, R0', @RW0+d8	MUL A, R0', @RW0+d8	MULW A, RW0', @RW0+d8	MULW A, MULW A, RW0', @RW0+d8	DIVU A, R0', @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	DIVW A, RW0', @RW0+d8	DIVW A, RW0', @RW0+d8
+1	MULU A, R1', @RW1+d8	MULU A, MULW A, RW1', @RW1+d8	MULW A, RW1', @RW1+d8	MULW A, MULW A, RW1', @RW1+d8	MUL A, R1', @RW1+d8	MUL A, R1', @RW1+d8	MULW A, RW1', @RW1+d8	MULW A, MULW A, RW1', @RW1+d8	DIVU A, R1', @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	DIVW A, RW1', @RW1+d8	DIVW A, RW1', @RW1+d8
+2	MULU A, R2', @RW2+d8	MULU A, MULW A, RW2', @RW2+d8	MULW A, RW2', @RW2+d8	MULW A, MULW A, RW2', @RW2+d8	MUL A, R2', @RW2+d8	MUL A, R2', @RW2+d8	MULW A, RW2', @RW2+d8	MULW A, MULW A, RW2', @RW2+d8	DIVU A, R2', @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	DIVW A, RW2', @RW2+d8	DIVW A, RW2', @RW2+d8
+3	MULU A, R3', @RW3+d8	MULU A, MULW A, RW3', @RW3+d8	MULW A, RW3', @RW3+d8	MULW A, MULW A, RW3', @RW3+d8	MUL A, R3', @RW3+d8	MUL A, R3', @RW3+d8	MULW A, RW3', @RW3+d8	MULW A, MULW A, RW3', @RW3+d8	DIVU A, R3', @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	DIVW A, RW3', @RW3+d8	DIVW A, RW3', @RW3+d8
+4	MULU A, R4', @RW4+d8	MULU A, MULW A, RW4', @RW4+d8	MULW A, RW4', @RW4+d8	MULW A, MULW A, RW4', @RW4+d8	MUL A, R4', @RW4+d8	MUL A, R4', @RW4+d8	MULW A, RW4', @RW4+d8	MULW A, MULW A, RW4', @RW4+d8	DIVU A, R4', @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	DIVW A, RW4', @RW4+d8	DIVW A, RW4', @RW4+d8
+5	MULU A, R5', @RW5+d8	MULU A, MULW A, RW5', @RW5+d8	MULW A, RW5', @RW5+d8	MULW A, MULW A, RW5', @RW5+d8	MUL A, R5', @RW5+d8	MUL A, R5', @RW5+d8	MULW A, RW5', @RW5+d8	MULW A, MULW A, RW5', @RW5+d8	DIVU A, R5', @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	DIVW A, RW5', @RW5+d8	DIVW A, RW5', @RW5+d8
+6	MULU A, R6', @RW6+d8	MULU A, MULW A, RW6', @RW6+d8	MULW A, RW6', @RW6+d8	MULW A, MULW A, RW6', @RW6+d8	MUL A, R6', @RW6+d8	MUL A, R6', @RW6+d8	MULW A, RW6', @RW6+d8	MULW A, MULW A, RW6', @RW6+d8	DIVU A, R6', @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	DIVW A, RW6', @RW6+d8	DIVW A, RW6', @RW6+d8
+7	MULU A, R7', @RW7+d8	MULU A, MULW A, RW7', @RW7+d8	MULW A, RW7', @RW7+d8	MULW A, MULW A, RW7', @RW7+d8	MUL A, R7', @RW7+d8	MUL A, R7', @RW7+d8	MULW A, RW7', @RW7+d8	MULW A, MULW A, RW7', @RW7+d8	DIVU A, R7', @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	DIVW A, RW7', @RW7+d8	DIVW A, RW7', @RW7+d8
+8	MULU A, @RW0, @RW0+d16	MULU A, MULW A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	MULW A, MULW A, @RW0, @RW0+d16	MUL A, @RW0, @RW0+d16	MUL A, @RW0, @RW0+d16	MULW A, @RW0, @RW0+d16	MULW A, MULW A, @RW0, @RW0+d16	DIVU 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	DIVW A, @RW0, @RW0+d16	DIVW A, @RW0, @RW0+d16
+9	MULU A, @RW1, @RW1+d16	MULU A, MULW A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	MULW A, MULW A, @RW1, @RW1+d16	MUL A, @RW1, @RW1+d16	MUL A, @RW1, @RW1+d16	MULW A, @RW1, @RW1+d16	MULW A, MULW A, @RW1, @RW1+d16	DIVU 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	DIVW A, @RW1, @RW1+d16	DIVW A, @RW1, @RW1+d16
+A	MULU A, @RW2, @RW2+d16	MULU A, MULW A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	MULW A, MULW A, @RW2, @RW2+d16	MUL A, @RW2, @RW2+d16	MUL A, @RW2, @RW2+d16	MULW A, @RW2, @RW2+d16	MULW A, MULW A, @RW2, @RW2+d16	DIVU 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	DIVW A, @RW2, @RW2+d16	DIVW A, @RW2, @RW2+d16
+B	MULU A, @RW3, @RW3+d16	MULU A, MULW A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	MULW A, MULW A, @RW3, @RW3+d16	MUL A, @RW3, @RW3+d16	MUL A, @RW3, @RW3+d16	MULW A, @RW3, @RW3+d16	MULW A, MULW A, @RW3, @RW3+d16	DIVU 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	DIVW A, @RW3, @RW3+d16	DIVW A, @RW3, @RW3+d16
+C	MULU A, @RW0+, @RW0+RW7	MULU A, MULW A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	MULW A, MULW A, @RW0+, @RW0+RW7	MUL A, @RW0+, @RW0+RW7	MUL A, @RW0+, @RW0+RW7	MULW A, @RW0+, @RW0+RW7	MULW A, MULW A, @RW0+, @RW0+RW7	DIVU 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	DIVW A, @RW0+, @RW0+RW7	DIVW A, @RW0+, @RW0+RW7
+D	MULU A, @RW1+, @RW1+RW7	MULU A, MULW A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	MULW A, MULW A, @RW1+, @RW1+RW7	MUL A, @RW1+, @RW1+RW7	MUL A, @RW1+, @RW1+RW7	MULW A, @RW1+, @RW1+RW7	MULW A, MULW A, @RW1+, @RW1+RW7	DIVU 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	DIVW A, @RW1+, @RW1+RW7	DIVW A, @RW1+, @RW1+RW7
+E	MULU A, @RW2+, @PC+d16	MULU A, MULW A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	MULW A, MULW A, @RW2+, @PC+d16	MUL A, @RW2+, @PC+d16	MUL A, @RW2+, @PC+d16	MULW A, @RW2+, @PC+d16	MULW A, MULW A, @RW2+, @PC+d16	DIVU 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	DIVW A, @RW2+, @PC+d16	DIVW A, @RW2+, @PC+d16
+F	MULU A, @RW3+, addr16	MULU A, MULW A, @RW3+, addr16	MULW A, @RW3+, addr16	MULW A, MULW A, @RW3+, addr16	MUL A, @RW3+, addr16	MUL A, @RW3+, addr16	MULW A, @RW3+, addr16	MULW A, MULW A, @RW3+, addr16	DIVU A, @RW3+, addr16	DIVU A, @RW3+, addr16	DIVU A, @RW3+, addr16	DIVUW A, @RW3+, addr16	DIVUW 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<sub>H</sub>)[illegible]

表 B.9-16 MOV Ri, ea 命令 (第1バイト = 7A<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0 @RW0+d8	MOV R0, R0 @RW0+d8	MOV R1, R0 @RW0+d8	MOV R2, R0 @RW0+d8	MOV R3, R0 @RW0+d8	MOV R4, R0 @RW0+d8	MOV R5, R0 @RW0+d8	MOV R6, R0 @RW0+d8	MOV R7, R0 @RW0+d8	MOV R8, R0 @RW0+d8	MOV R9, R0 @RW0+d8	MOV R10, R0 @RW0+d8	MOV R11, R0 @RW0+d8	MOV R12, R0 @RW0+d8	MOV R13, R0 @RW0+d8	MOV R14, R0 @RW0+d8
+1	MOV R0, R1 @RW1+d8	MOV R0, R1 @RW1+d8	MOV R1, R1 @RW1+d8	MOV R2, R1 @RW1+d8	MOV R3, R1 @RW1+d8	MOV R4, R1 @RW1+d8	MOV R5, R1 @RW1+d8	MOV R6, R1 @RW1+d8	MOV R7, R1 @RW1+d8	MOV R8, R1 @RW1+d8	MOV R9, R1 @RW1+d8	MOV R10, R1 @RW1+d8	MOV R11, R1 @RW1+d8	MOV R12, R1 @RW1+d8	MOV R13, R1 @RW1+d8	MOV R14, R1 @RW1+d8
+2	MOV R0, R2 @RW2+d8	MOV R0, R2 @RW2+d8	MOV R1, R2 @RW2+d8	MOV R2, R2 @RW2+d8	MOV R3, R2 @RW2+d8	MOV R4, R2 @RW2+d8	MOV R5, R2 @RW2+d8	MOV R6, R2 @RW2+d8	MOV R7, R2 @RW2+d8	MOV R8, R2 @RW2+d8	MOV R9, R2 @RW2+d8	MOV R10, R2 @RW2+d8	MOV R11, R2 @RW2+d8	MOV R12, R2 @RW2+d8	MOV R13, R2 @RW2+d8	MOV R14, R2 @RW2+d8
+3	MOV R0, R3 @RW3+d8	MOV R0, R3 @RW3+d8	MOV R1, R3 @RW3+d8	MOV R2, R3 @RW3+d8	MOV R3, R3 @RW3+d8	MOV R4, R3 @RW3+d8	MOV R5, R3 @RW3+d8	MOV R6, R3 @RW3+d8	MOV R7, R3 @RW3+d8	MOV R8, R3 @RW3+d8	MOV R9, R3 @RW3+d8	MOV R10, R3 @RW3+d8	MOV R11, R3 @RW3+d8	MOV R12, R3 @RW3+d8	MOV R13, R3 @RW3+d8	MOV R14, R3 @RW3+d8
+4	MOV R0, R4 @RW4+d8	MOV R0, R4 @RW4+d8	MOV R1, R4 @RW4+d8	MOV R2, R4 @RW4+d8	MOV R3, R4 @RW4+d8	MOV R4, R4 @RW4+d8	MOV R5, R4 @RW4+d8	MOV R6, R4 @RW4+d8	MOV R7, R4 @RW4+d8	MOV R8, R4 @RW4+d8	MOV R9, R4 @RW4+d8	MOV R10, R4 @RW4+d8	MOV R11, R4 @RW4+d8	MOV R12, R4 @RW4+d8	MOV R13, R4 @RW4+d8	MOV R14, R4 @RW4+d8
+5	MOV R0, R5 @RW5+d8	MOV R0, R5 @RW5+d8	MOV R1, R5 @RW5+d8	MOV R2, R5 @RW5+d8	MOV R3, R5 @RW5+d8	MOV R4, R5 @RW5+d8	MOV R5, R5 @RW5+d8	MOV R6, R5 @RW5+d8	MOV R7, R5 @RW5+d8	MOV R8, R5 @RW5+d8	MOV R9, R5 @RW5+d8	MOV R10, R5 @RW5+d8	MOV R11, R5 @RW5+d8	MOV R12, R5 @RW5+d8	MOV R13, R5 @RW5+d8	MOV R14, R5 @RW5+d8
+6	MOV R0, R6 @RW6+d8	MOV R0, R6 @RW6+d8	MOV R1, R6 @RW6+d8	MOV R2, R6 @RW6+d8	MOV R3, R6 @RW6+d8	MOV R4, R6 @RW6+d8	MOV R5, R6 @RW6+d8	MOV R6, R6 @RW6+d8	MOV R7, R6 @RW6+d8	MOV R8, R6 @RW6+d8	MOV R9, R6 @RW6+d8	MOV R10, R6 @RW6+d8	MOV R11, R6 @RW6+d8	MOV R12, R6 @RW6+d8	MOV R13, R6 @RW6+d8	MOV R14, R6 @RW6+d8
+7	MOV R0, R7 @RW7+d8	MOV R0, R7 @RW7+d8	MOV R1, R7 @RW7+d8	MOV R2, R7 @RW7+d8	MOV R3, R7 @RW7+d8	MOV R4, R7 @RW7+d8	MOV R5, R7 @RW7+d8	MOV R6, R7 @RW7+d8	MOV R7, R7 @RW7+d8	MOV R8, R7 @RW7+d8	MOV R9, R7 @RW7+d8	MOV R10, R7 @RW7+d8	MOV R11, R7 @RW7+d8	MOV R12, R7 @RW7+d8	MOV R13, R7 @RW7+d8	MOV R14, R7 @RW7+d8
+8	MOV R0, @RW0	MOV R0, @RW0	MOV R1, @RW0	MOV R2, @RW0	MOV R3, @RW0	MOV R4, @RW0	MOV R5, @RW0	MOV R6, @RW0	MOV R7, @RW0	MOV R8, @RW0	MOV R9, @RW0	MOV R10, @RW0	MOV R11, @RW0	MOV R12, @RW0	MOV R13, @RW0	MOV R14, @RW0
+9	MOV R0, @RW1	MOV R0, @RW1	MOV R1, @RW1	MOV R2, @RW1	MOV R3, @RW1	MOV R4, @RW1	MOV R5, @RW1	MOV R6, @RW1	MOV R7, @RW1	MOV R8, @RW1	MOV R9, @RW1	MOV R10, @RW1	MOV R11, @RW1	MOV R12, @RW1	MOV R13, @RW1	MOV R14, @RW1
+A	MOV R0, @RW2	MOV R0, @RW2	MOV R1, @RW2	MOV R2, @RW2	MOV R3, @RW2	MOV R4, @RW2	MOV R5, @RW2	MOV R6, @RW2	MOV R7, @RW2	MOV R8, @RW2	MOV R9, @RW2	MOV R10, @RW2	MOV R11, @RW2	MOV R12, @RW2	MOV R13, @RW2	MOV R14, @RW2
+B	MOV R0, @RW3	MOV R0, @RW3	MOV R1, @RW3	MOV R2, @RW3	MOV R3, @RW3	MOV R4, @RW3	MOV R5, @RW3	MOV R6, @RW3	MOV R7, @RW3	MOV R8, @RW3	MOV R9, @RW3	MOV R10, @RW3	MOV R11, @RW3	MOV R12, @RW3	MOV R13, @RW3	MOV R14, @RW3
+C	MOV R0, @RW0+	MOV R0, @RW0+	MOV R1, @RW0+	MOV R2, @RW0+	MOV R3, @RW0+	MOV R4, @RW0+	MOV R5, @RW0+	MOV R6, @RW0+	MOV R7, @RW0+	MOV R8, @RW0+	MOV R9, @RW0+	MOV R10, @RW0+	MOV R11, @RW0+	MOV R12, @RW0+	MOV R13, @RW0+	MOV R14, @RW0+
+D	MOV R0, @RW1+	MOV R0, @RW1+	MOV R1, @RW1+	MOV R2, @RW1+	MOV R3, @RW1+	MOV R4, @RW1+	MOV R5, @RW1+	MOV R6, @RW1+	MOV R7, @RW1+	MOV R8, @RW1+	MOV R9, @RW1+	MOV R10, @RW1+	MOV R11, @RW1+	MOV R12, @RW1+	MOV R13, @RW1+	MOV R14, @RW1+
+E	MOV R0, @RW2+	MOV R0, @RW2+	MOV R1, @RW2+	MOV R2, @RW2+	MOV R3, @RW2+	MOV R4, @RW2+	MOV R5, @RW2+	MOV R6, @RW2+	MOV R7, @RW2+	MOV R8, @RW2+	MOV R9, @RW2+	MOV R10, @RW2+	MOV R11, @RW2+	MOV R12, @RW2+	MOV R13, @RW2+	MOV R14, @RW2+
+F	MOV R0, @RW3+	MOV R0, @RW3+	MOV R1, @RW3+	MOV R2, @RW3+	MOV R3, @RW3+	MOV R4, @RW3+	MOV R5, @RW3+	MOV R6, @RW3+	MOV R7, @RW3+	MOV R8, @RW3+	MOV R9, @RW3+	MOV R10, @RW3+	MOV R11, @RW3+	MOV R12, @RW3+	MOV R13, @RW3+	MOV R14, @RW3+

表 B.9-17 MOVW RWi, ea 命令 (第 1 バイト = 7BH)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOVW RW0, RW0, @RW0+d8	MOVW RW1, RW0, @RW0+d8	MOVW RW1, RW1, @RW1+d8	MOVW RW1, RW1, @RW1+d8	MOVW RW2, RW0, @RW0+d8	MOVW RW2, RW2, @RW2+d8	MOVW RW3, RW0, @RW0+d8	MOVW RW3, RW0, @RW0+d8	MOVW RW4, RW0, @RW0+d8	MOVW RW4, RW1, @RW1+d8	MOVW RW5, RW0, @RW0+d8	MOVW RW5, RW1, @RW1+d8	MOVW RW6, RW0, @RW0+d8	MOVW RW6, RW1, @RW1+d8	MOVW RW7, RW0, @RW0+d8	MOVW RW7, RW1, @RW1+d8
+1	MOVW RW0, RW1, @RW1+d8	MOVW RW1, RW1, @RW1+d8	MOVW RW1, RW1, @RW1+d8	MOVW RW1, RW1, @RW1+d8	MOVW RW2, RW1, @RW1+d8	MOVW RW2, RW2, @RW2+d8	MOVW RW3, RW1, @RW1+d8	MOVW RW3, RW1, @RW1+d8	MOVW RW4, RW1, @RW1+d8	MOVW RW4, RW2, @RW2+d8	MOVW RW5, RW1, @RW1+d8	MOVW RW5, RW2, @RW2+d8	MOVW RW6, RW1, @RW1+d8	MOVW RW6, RW2, @RW2+d8	MOVW RW7, RW1, @RW1+d8	MOVW RW7, RW2, @RW2+d8
+2	MOVW RW0, RW2, @RW2+d8	MOVW RW1, RW2, @RW2+d8	MOVW RW1, RW2, @RW2+d8	MOVW RW1, RW2, @RW2+d8	MOVW RW2, RW2, @RW2+d8	MOVW RW2, RW2, @RW2+d8	MOVW RW3, RW2, @RW2+d8	MOVW RW3, RW2, @RW2+d8	MOVW RW4, RW2, @RW2+d8	MOVW RW4, RW3, @RW3+d8	MOVW RW5, RW2, @RW2+d8	MOVW RW5, RW3, @RW3+d8	MOVW RW6, RW2, @RW2+d8	MOVW RW6, RW3, @RW3+d8	MOVW RW7, RW2, @RW2+d8	MOVW RW7, RW3, @RW3+d8
+3	MOVW RW0, RW3, @RW3+d8	MOVW RW1, RW3, @RW3+d8	MOVW RW1, RW3, @RW3+d8	MOVW RW1, RW3, @RW3+d8	MOVW RW2, RW3, @RW3+d8	MOVW RW2, RW3, @RW3+d8	MOVW RW3, RW3, @RW3+d8	MOVW RW3, RW3, @RW3+d8	MOVW RW4, RW3, @RW3+d8	MOVW RW4, RW4, @RW4+d8	MOVW RW5, RW3, @RW3+d8	MOVW RW5, RW4, @RW4+d8	MOVW RW6, RW3, @RW3+d8	MOVW RW6, RW4, @RW4+d8	MOVW RW7, RW3, @RW3+d8	MOVW RW7, RW4, @RW4+d8
+4	MOVW RW0, RW4, @RW4+d8	MOVW RW1, RW4, @RW4+d8	MOVW RW1, RW4, @RW4+d8	MOVW RW1, RW4, @RW4+d8	MOVW RW2, RW4, @RW4+d8	MOVW RW2, RW4, @RW4+d8	MOVW RW3, RW4, @RW4+d8	MOVW RW3, RW4, @RW4+d8	MOVW RW4, RW4, @RW4+d8	MOVW RW4, RW5, @RW5+d8	MOVW RW5, RW4, @RW4+d8	MOVW RW5, RW5, @RW5+d8	MOVW RW6, RW4, @RW4+d8	MOVW RW6, RW5, @RW5+d8	MOVW RW7, RW4, @RW4+d8	MOVW RW7, RW5, @RW5+d8
+5	MOVW RW0, RW5, @RW5+d8	MOVW RW1, RW5, @RW5+d8	MOVW RW1, RW5, @RW5+d8	MOVW RW1, RW5, @RW5+d8	MOVW RW2, RW5, @RW5+d8	MOVW RW2, RW5, @RW5+d8	MOVW RW3, RW5, @RW5+d8	MOVW RW3, RW5, @RW5+d8	MOVW RW4, RW5, @RW5+d8	MOVW RW4, RW6, @RW6+d8	MOVW RW5, RW5, @RW5+d8	MOVW RW5, RW6, @RW6+d8	MOVW RW6, RW5, @RW5+d8	MOVW RW6, RW6, @RW6+d8	MOVW RW7, RW5, @RW5+d8	MOVW RW7, RW6, @RW6+d8
+6	MOVW RW0, RW6, @RW6+d8	MOVW RW1, RW6, @RW6+d8	MOVW RW1, RW6, @RW6+d8	MOVW RW1, RW6, @RW6+d8	MOVW RW2, RW6, @RW6+d8	MOVW RW2, RW6, @RW6+d8	MOVW RW3, RW6, @RW6+d8	MOVW RW3, RW6, @RW6+d8	MOVW RW4, RW6, @RW6+d8	MOVW RW4, RW7, @RW7+d8	MOVW RW5, RW6, @RW6+d8	MOVW RW5, RW7, @RW7+d8	MOVW RW6, RW6, @RW6+d8	MOVW RW6, RW7, @RW7+d8	MOVW RW7, RW6, @RW6+d8	MOVW RW7, RW7, @RW7+d8
+7	MOVW RW0, RW7, @RW7+d8	MOVW RW1, RW7, @RW7+d8	MOVW RW1, RW7, @RW7+d8	MOVW RW1, RW7, @RW7+d8	MOVW RW2, RW7, @RW7+d8	MOVW RW2, RW7, @RW7+d8	MOVW RW3, RW7, @RW7+d8	MOVW RW3, RW7, @RW7+d8	MOVW RW4, RW7, @RW7+d8	MOVW RW4, RW8, @RW8+d8	MOVW RW5, RW7, @RW7+d8	MOVW RW5, RW8, @RW8+d8	MOVW RW6, RW7, @RW7+d8	MOVW RW6, RW8, @RW8+d8	MOVW RW7, RW7, @RW7+d8	MOVW RW7, RW8, @RW8+d8
+8	MOVW RW0, @RW0	MOVW RW1, @RW0	MOVW RW1, @RW0	MOVW RW1, @RW0	MOVW RW2, @RW0	MOVW RW2, @RW0	MOVW RW3, @RW0	MOVW RW3, @RW0	MOVW RW4, @RW0	MOVW RW4, @RW1	MOVW RW5, @RW0	MOVW RW5, @RW1	MOVW RW6, @RW0	MOVW RW6, @RW1	MOVW RW7, @RW0	MOVW RW7, @RW1
+9	MOVW RW0, @RW1	MOVW RW1, @RW1	MOVW RW1, @RW1	MOVW RW1, @RW1	MOVW RW2, @RW1	MOVW RW2, @RW1	MOVW RW3, @RW1	MOVW RW3, @RW1	MOVW RW4, @RW1	MOVW RW4, @RW2	MOVW RW5, @RW1	MOVW RW5, @RW2	MOVW RW6, @RW1	MOVW RW6, @RW2	MOVW RW7, @RW1	MOVW RW7, @RW2
+A	MOVW RW0, @RW2	MOVW RW1, @RW2	MOVW RW1, @RW2	MOVW RW1, @RW2	MOVW RW2, @RW2	MOVW RW2, @RW2	MOVW RW3, @RW2	MOVW RW3, @RW2	MOVW RW4, @RW2	MOVW RW4, @RW3	MOVW RW5, @RW2	MOVW RW5, @RW3	MOVW RW6, @RW2	MOVW RW6, @RW3	MOVW RW7, @RW2	MOVW RW7, @RW3
+B	MOVW RW0, @RW3	MOVW RW1, @RW3	MOVW RW1, @RW3	MOVW RW1, @RW3	MOVW RW2, @RW3	MOVW RW2, @RW3	MOVW RW3, @RW3	MOVW RW3, @RW3	MOVW RW4, @RW3	MOVW RW4, @RW4	MOVW RW5, @RW3	MOVW RW5, @RW4	MOVW RW6, @RW3	MOVW RW6, @RW4	MOVW RW7, @RW3	MOVW RW7, @RW4
+C	MOVW RW0, @RW4	MOVW RW1, @RW4	MOVW RW1, @RW4	MOVW RW1, @RW4	MOVW RW2, @RW4	MOVW RW2, @RW4	MOVW RW3, @RW4	MOVW RW3, @RW4	MOVW RW4, @RW4	MOVW RW4, @RW5	MOVW RW5, @RW4	MOVW RW5, @RW5	MOVW RW6, @RW4	MOVW RW6, @RW5	MOVW RW7, @RW4	MOVW RW7, @RW5
+D	MOVW RW0, @RW5	MOVW RW1, @RW5	MOVW RW1, @RW5	MOVW RW1, @RW5	MOVW RW2, @RW5	MOVW RW2, @RW5	MOVW RW3, @RW5	MOVW RW3, @RW5	MOVW RW4, @RW5	MOVW RW4, @RW6	MOVW RW5, @RW5	MOVW RW5, @RW6	MOVW RW6, @RW5	MOVW RW6, @RW6	MOVW RW7, @RW5	MOVW RW7, @RW6
+E	MOVW RW0, @RW6	MOVW RW1, @RW6	MOVW RW1, @RW6	MOVW RW1, @RW6	MOVW RW2, @RW6	MOVW RW2, @RW6	MOVW RW3, @RW6	MOVW RW3, @RW6	MOVW RW4, @RW6	MOVW RW4, @RW7	MOVW RW5, @RW6	MOVW RW5, @RW7	MOVW RW6, @RW6	MOVW RW6, @RW7	MOVW RW7, @RW6	MOVW RW7, @RW7
+F	MOVW RW0, @RW7	MOVW RW1, @RW7	MOVW RW1, @RW7	MOVW RW1, @RW7	MOVW RW2, @RW7	MOVW RW2, @RW7	MOVW RW3, @RW7	MOVW RW3, @RW7	MOVW RW4, @RW7	MOVW RW4, @RW8	MOVW RW5, @RW7	MOVW RW5, @RW8	MOVW RW6, @RW7	MOVW RW6, @RW8	MOVW RW7, @RW7	MOVW RW7, @RW8

表 B.9-18 MOV Ri, ea 命令 (第1バイト = 7C<sub>H</sub>)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV R0, R0, @RW0+d8, R0	MOV R0, R1, @RW0-d8, R1	MOV R0, R1, @RW0-d8, R1	MOV R0, R2, @RW0+d8, R2	MOV R0, R3, @RW0+d8, R3	MOV R0, R4, @RW0+d8, R4	MOV R0, R5, @RW0+d8, R5	MOV R0, R6, @RW0+d8, R6	MOV R0, R7, @RW0+d8, R7	MOV R0, R8, @RW0+d8, R8	MOV R0, R9, @RW0+d8, R9	MOV R0, R10, @RW0+d8, R10	MOV R0, R11, @RW0+d8, R11	MOV R0, R12, @RW0+d8, R12	MOV R0, R13, @RW0+d8, R13	MOV R0, R14, @RW0+d8, R14
+1	MOV R1, R0, @RW1+d8, R0	MOV R1, R1, @RW1+d8, R1	MOV R1, R2, @RW1+d8, R2	MOV R1, R3, @RW1+d8, R3	MOV R1, R4, @RW1+d8, R4	MOV R1, R5, @RW1+d8, R5	MOV R1, R6, @RW1+d8, R6	MOV R1, R7, @RW1+d8, R7	MOV R1, R8, @RW1+d8, R8	MOV R1, R9, @RW1+d8, R9	MOV R1, R10, @RW1+d8, R10	MOV R1, R11, @RW1+d8, R11	MOV R1, R12, @RW1+d8, R12	MOV R1, R13, @RW1+d8, R13	MOV R1, R14, @RW1+d8, R14	MOV R1, R15, @RW1+d8, R15
+2	MOV R2, R0, @RW2+d8, R0	MOV R2, R1, @RW2+d8, R1	MOV R2, R2, @RW2+d8, R2	MOV R2, R3, @RW2+d8, R3	MOV R2, R4, @RW2+d8, R4	MOV R2, R5, @RW2+d8, R5	MOV R2, R6, @RW2+d8, R6	MOV R2, R7, @RW2+d8, R7	MOV R2, R8, @RW2+d8, R8	MOV R2, R9, @RW2+d8, R9	MOV R2, R10, @RW2+d8, R10	MOV R2, R11, @RW2+d8, R11	MOV R2, R12, @RW2+d8, R12	MOV R2, R13, @RW2+d8, R13	MOV R2, R14, @RW2+d8, R14	MOV R2, R15, @RW2+d8, R15
+3	MOV R3, R0, @RW3+d8, R0	MOV R3, R1, @RW3+d8, R1	MOV R3, R2, @RW3+d8, R2	MOV R3, R3, @RW3+d8, R3	MOV R3, R4, @RW3+d8, R4	MOV R3, R5, @RW3+d8, R5	MOV R3, R6, @RW3+d8, R6	MOV R3, R7, @RW3+d8, R7	MOV R3, R8, @RW3+d8, R8	MOV R3, R9, @RW3+d8, R9	MOV R3, R10, @RW3+d8, R10	MOV R3, R11, @RW3+d8, R11	MOV R3, R12, @RW3+d8, R12	MOV R3, R13, @RW3+d8, R13	MOV R3, R14, @RW3+d8, R14	MOV R3, R15, @RW3+d8, R15
+4	MOV R4, R0, @RW4+d8, R0	MOV R4, R1, @RW4+d8, R1	MOV R4, R2, @RW4+d8, R2	MOV R4, R3, @RW4+d8, R3	MOV R4, R4, @RW4+d8, R4	MOV R4, R5, @RW4+d8, R5	MOV R4, R6, @RW4+d8, R6	MOV R4, R7, @RW4+d8, R7	MOV R4, R8, @RW4+d8, R8	MOV R4, R9, @RW4+d8, R9	MOV R4, R10, @RW4+d8, R10	MOV R4, R11, @RW4+d8, R11	MOV R4, R12, @RW4+d8, R12	MOV R4, R13, @RW4+d8, R13	MOV R4, R14, @RW4+d8, R14	MOV R4, R15, @RW4+d8, R15
+5	MOV R5, R0, @RW5+d8, R0	MOV R5, R1, @RW5+d8, R1	MOV R5, R2, @RW5+d8, R2	MOV R5, R3, @RW5+d8, R3	MOV R5, R4, @RW5+d8, R4	MOV R5, R5, @RW5+d8, R5	MOV R5, R6, @RW5+d8, R6	MOV R5, R7, @RW5+d8, R7	MOV R5, R8, @RW5+d8, R8	MOV R5, R9, @RW5+d8, R9	MOV R5, R10, @RW5+d8, R10	MOV R5, R11, @RW5+d8, R11	MOV R5, R12, @RW5+d8, R12	MOV R5, R13, @RW5+d8, R13	MOV R5, R14, @RW5+d8, R14	MOV R5, R15, @RW5+d8, R15
+6	MOV R6, R0, @RW6+d8, R0	MOV R6, R1, @RW6+d8, R1	MOV R6, R2, @RW6+d8, R2	MOV R6, R3, @RW6+d8, R3	MOV R6, R4, @RW6+d8, R4	MOV R6, R5, @RW6+d8, R5	MOV R6, R6, @RW6+d8, R6	MOV R6, R7, @RW6+d8, R7	MOV R6, R8, @RW6+d8, R8	MOV R6, R9, @RW6+d8, R9	MOV R6, R10, @RW6+d8, R10	MOV R6, R11, @RW6+d8, R11	MOV R6, R12, @RW6+d8, R12	MOV R6, R13, @RW6+d8, R13	MOV R6, R14, @RW6+d8, R14	MOV R6, R15, @RW6+d8, R15
+7	MOV R7, R0, @RW7+d8, R0	MOV R7, R1, @RW7+d8, R1	MOV R7, R2, @RW7+d8, R2	MOV R7, R3, @RW7+d8, R3	MOV R7, R4, @RW7+d8, R4	MOV R7, R5, @RW7+d8, R5	MOV R7, R6, @RW7+d8, R6	MOV R7, R7, @RW7+d8, R7	MOV R7, R8, @RW7+d8, R8	MOV R7, R9, @RW7+d8, R9	MOV R7, R10, @RW7+d8, R10	MOV R7, R11, @RW7+d8, R11	MOV R7, R12, @RW7+d8, R12	MOV R7, R13, @RW7+d8, R13	MOV R7, R14, @RW7+d8, R14	MOV R7, R15, @RW7+d8, R15
+8	MOV @RW0, R0, @RW0+d16, R0	MOV @RW0, R1, @RW0+d16, R1	MOV @RW0, R2, @RW0+d16, R2	MOV @RW0, R3, @RW0+d16, R3	MOV @RW0, R4, @RW0+d16, R4	MOV @RW0, R5, @RW0+d16, R5	MOV @RW0, R6, @RW0+d16, R6	MOV @RW0, R7, @RW0+d16, R7	MOV @RW0, R8, @RW0+d16, R8	MOV @RW0, R9, @RW0+d16, R9	MOV @RW0, R10, @RW0+d16, R10	MOV @RW0, R11, @RW0+d16, R11	MOV @RW0, R12, @RW0+d16, R12	MOV @RW0, R13, @RW0+d16, R13	MOV @RW0, R14, @RW0+d16, R14	MOV @RW0, R15, @RW0+d16, R15
+9	MOV @RW1, R0, @RW1+d16, R0	MOV @RW1, R1, @RW1+d16, R1	MOV @RW1, R2, @RW1+d16, R2	MOV @RW1, R3, @RW1+d16, R3	MOV @RW1, R4, @RW1+d16, R4	MOV @RW1, R5, @RW1+d16, R5	MOV @RW1, R6, @RW1+d16, R6	MOV @RW1, R7, @RW1+d16, R7	MOV @RW1, R8, @RW1+d16, R8	MOV @RW1, R9, @RW1+d16, R9	MOV @RW1, R10, @RW1+d16, R10	MOV @RW1, R11, @RW1+d16, R11	MOV @RW1, R12, @RW1+d16, R12	MOV @RW1, R13, @RW1+d16, R13	MOV @RW1, R14, @RW1+d16, R14	MOV @RW1, R15, @RW1+d16, R15
+A	MOV @RW2, R0, @RW2+d16, R0	MOV @RW2, R1, @RW2+d16, R1	MOV @RW2, R2, @RW2+d16, R2	MOV @RW2, R3, @RW2+d16, R3	MOV @RW2, R4, @RW2+d16, R4	MOV @RW2, R5, @RW2+d16, R5	MOV @RW2, R6, @RW2+d16, R6	MOV @RW2, R7, @RW2+d16, R7	MOV @RW2, R8, @RW2+d16, R8	MOV @RW2, R9, @RW2+d16, R9	MOV @RW2, R10, @RW2+d16, R10	MOV @RW2, R11, @RW2+d16, R11	MOV @RW2, R12, @RW2+d16, R12	MOV @RW2, R13, @RW2+d16, R13	MOV @RW2, R14, @RW2+d16, R14	MOV @RW2, R15, @RW2+d16, R15
+B	MOV @RW3, R0, @RW3+d16, R0	MOV @RW3, R1, @RW3+d16, R1	MOV @RW3, R2, @RW3+d16, R2	MOV @RW3, R3, @RW3+d16, R3	MOV @RW3, R4, @RW3+d16, R4	MOV @RW3, R5, @RW3+d16, R5	MOV @RW3, R6, @RW3+d16, R6	MOV @RW3, R7, @RW3+d16, R7	MOV @RW3, R8, @RW3+d16, R8	MOV @RW3, R9, @RW3+d16, R9	MOV @RW3, R10, @RW3+d16, R10	MOV @RW3, R11, @RW3+d16, R11	MOV @RW3, R12, @RW3+d16, R12	MOV @RW3, R13, @RW3+d16, R13	MOV @RW3, R14, @RW3+d16, R14	MOV @RW3, R15, @RW3+d16, R15
+C	MOV @RW0+, R0, @RW0+RW7, R0	MOV @RW0+, R1, @RW0+RW7, R1	MOV @RW0+, R2, @RW0+RW7, R2	MOV @RW0+, R3, @RW0+RW7, R3	MOV @RW0+, R4, @RW0+RW7, R4	MOV @RW0+, R5, @RW0+RW7, R5	MOV @RW0+, R6, @RW0+RW7, R6	MOV @RW0+, R7, @RW0+RW7, R7	MOV @RW0+, R8, @RW0+RW7, R8	MOV @RW0+, R9, @RW0+RW7, R9	MOV @RW0+, R10, @RW0+RW7, R10	MOV @RW0+, R11, @RW0+RW7, R11	MOV @RW0+, R12, @RW0+RW7, R12	MOV @RW0+, R13, @RW0+RW7, R13	MOV @RW0+, R14, @RW0+RW7, R14	MOV @RW0+, R15, @RW0+RW7, R15
+D	MOV @RW1+, R0, @RW1+RW7, R0	MOV @RW1+, R1, @RW1+RW7, R1	MOV @RW1+, R2, @RW1+RW7, R2	MOV @RW1+, R3, @RW1+RW7, R3	MOV @RW1+, R4, @RW1+RW7, R4	MOV @RW1+, R5, @RW1+RW7, R5	MOV @RW1+, R6, @RW1+RW7, R6	MOV @RW1+, R7, @RW1+RW7, R7	MOV @RW1+, R8, @RW1+RW7, R8	MOV @RW1+, R9, @RW1+RW7, R9	MOV @RW1+, R10, @RW1+RW7, R10	MOV @RW1+, R11, @RW1+RW7, R11	MOV @RW1+, R12, @RW1+RW7, R12	MOV @RW1+, R13, @RW1+RW7, R13	MOV @RW1+, R14, @RW1+RW7, R14	MOV @RW1+, R15, @RW1+RW7, R15
+E	MOV @RW2+, R0, @PC+d16, R0	MOV @RW2+, R1, @PC+d16, R1	MOV @RW2+, R2, @PC+d16, R2	MOV @RW2+, R3, @PC+d16, R3	MOV @RW2+, R4, @PC+d16, R4	MOV @RW2+, R5, @PC+d16, R5	MOV @RW2+, R6, @PC+d16, R6	MOV @RW2+, R7, @PC+d16, R7	MOV @RW2+, R8, @PC+d16, R8	MOV @RW2+, R9, @PC+d16, R9	MOV @RW2+, R10, @PC+d16, R10	MOV @RW2+, R11, @PC+d16, R11	MOV @RW2+, R12, @PC+d16, R12	MOV @RW2+, R13, @PC+d16, R13	MOV @RW2+, R14, @PC+d16, R14	MOV @RW2+, R15, @PC+d16, R15
+F	MOV @RW3+, R0, addr16, R0	MOV @RW3+, R1, addr16, R1	MOV @RW3+, R2, addr16, R2	MOV @RW3+, R3, addr16, R3	MOV @RW3+, R4, addr16, R4	MOV @RW3+, R5, addr16, R5	MOV @RW3+, R6, addr16, R6	MOV @RW3+, R7, addr16, R7	MOV @RW3+, R8, addr16, R8	MOV @RW3+, R9, addr16, R9	MOV @RW3+, R10, addr16, R10	MOV @RW3+, R11, addr16, R11	MOV @RW3+, R12, addr16, R12	MOV @RW3+, R13, addr16, R13	MOV @RW3+, R14, addr16, R14	MOV @RW3+, R15, addr16, R15

表 B.9-19 MOVW RWi, ea 命令 (第1バイト = 7D<sub>H</sub>)[illegible]



表 B.9-20 XCH Ri, ea 命令 (第1バイト=7EH)

	00	10	20	30	40	50	60	70	80	90	A	B0	C0	D0	E0	F0
+0	XCH R0, R0' @RW0+d8	XCH R0, @RW0+d8	XCH R1, R0' @RW0+d8	XCH R1, R0' @RW0+d8	XCH R2, R0' @RW0+d8	XCH R2, @RW0+d8	XCH R3, R0' @RW0+d8	XCH R3, @RW0+d8	XCH R4, R0' @RW0+d8	XCH R4, @RW0+d8	XCH R5, R0' @RW0+d8	XCH R5, @RW0+d8	XCH R6, R0' @RW0+d8	XCH R6, @RW0+d8	XCH R7, R0' @RW0+d8	XCH R7, @RW0+d8
+1	XCH R0, R1' @RW1+d8	XCH R0, @RW1+d8	XCH R1, R1' @RW1+d8	XCH R1, R1' @RW1+d8	XCH R2, R1' @RW1+d8	XCH R2, @RW1+d8	XCH R3, R1' @RW1+d8	XCH R3, @RW1+d8	XCH R4, R1' @RW1+d8	XCH R4, @RW1+d8	XCH R5, R1' @RW1+d8	XCH R5, @RW1+d8	XCH R6, R1' @RW1+d8	XCH R6, @RW1+d8	XCH R7, R1' @RW1+d8	XCH R7, @RW1+d8
+2	XCH R0, R2' @RW2+d8	XCH R0, @RW2+d8	XCH R1, R2' @RW2+d8	XCH R1, R2' @RW2+d8	XCH R2, R2' @RW2+d8	XCH R2, @RW2+d8	XCH R3, R2' @RW2+d8	XCH R3, @RW2+d8	XCH R4, R2' @RW2+d8	XCH R4, @RW2+d8	XCH R5, R2' @RW2+d8	XCH R5, @RW2+d8	XCH R6, R2' @RW2+d8	XCH R6, @RW2+d8	XCH R7, R2' @RW2+d8	XCH R7, @RW2+d8
+3	XCH R0, R3' @RW3+d8	XCH R0, @RW3+d8	XCH R1, R3' @RW3+d8	XCH R1, R3' @RW3+d8	XCH R2, R3' @RW3+d8	XCH R2, @RW3+d8	XCH R3, R3' @RW3+d8	XCH R3, @RW3+d8	XCH R4, R3' @RW3+d8	XCH R4, @RW3+d8	XCH R5, R3' @RW3+d8	XCH R5, @RW3+d8	XCH R6, R3' @RW3+d8	XCH R6, @RW3+d8	XCH R7, R3' @RW3+d8	XCH R7, @RW3+d8
+4	XCH R0, R4' @RW4+d8	XCH R0, @RW4+d8	XCH R1, R4' @RW4+d8	XCH R1, R4' @RW4+d8	XCH R2, R4' @RW4+d8	XCH R2, @RW4+d8	XCH R3, R4' @RW4+d8	XCH R3, @RW4+d8	XCH R4, R4' @RW4+d8	XCH R4, @RW4+d8	XCH R5, R4' @RW4+d8	XCH R5, @RW4+d8	XCH R6, R4' @RW4+d8	XCH R6, @RW4+d8	XCH R7, R4' @RW4+d8	XCH R7, @RW4+d8
+5	XCH R0, R5' @RW5+d8	XCH R0, @RW5+d8	XCH R1, R5' @RW5+d8	XCH R1, R5' @RW5+d8	XCH R2, R5' @RW5+d8	XCH R2, @RW5+d8	XCH R3, R5' @RW5+d8	XCH R3, @RW5+d8	XCH R4, R5' @RW5+d8	XCH R4, @RW5+d8	XCH R5, R5' @RW5+d8	XCH R5, @RW5+d8	XCH R6, R5' @RW5+d8	XCH R6, @RW5+d8	XCH R7, R5' @RW5+d8	XCH R7, @RW5+d8
+6	XCH R0, R6' @RW6+d8	XCH R0, @RW6+d8	XCH R1, R6' @RW6+d8	XCH R1, R6' @RW6+d8	XCH R2, R6' @RW6+d8	XCH R2, @RW6+d8	XCH R3, R6' @RW6+d8	XCH R3, @RW6+d8	XCH R4, R6' @RW6+d8	XCH R4, @RW6+d8	XCH R5, R6' @RW6+d8	XCH R5, @RW6+d8	XCH R6, R6' @RW6+d8	XCH R6, @RW6+d8	XCH R7, R6' @RW6+d8	XCH R7, @RW6+d8
+7	XCH R0, R7' @RW7+d8	XCH R0, @RW7+d8	XCH R1, R7' @RW7+d8	XCH R1, R7' @RW7+d8	XCH R2, R7' @RW7+d8	XCH R2, @RW7+d8	XCH R3, R7' @RW7+d8	XCH R3, @RW7+d8	XCH R4, R7' @RW7+d8	XCH R4, @RW7+d8	XCH R5, R7' @RW7+d8	XCH R5, @RW7+d8	XCH R6, R7' @RW7+d8	XCH R6, @RW7+d8	XCH R7, R7' @RW7+d8	XCH R7, @RW7+d8
+8	XCH R0, @RW0' @RW0+d16	XCH R0, @RW0+d16	XCH R1, @RW0' @RW0+d16	XCH R1, @RW0' @RW0+d16	XCH R2, @RW0' @RW0+d16	XCH R2, @RW0+d16	XCH R3, @RW0' @RW0+d16	XCH R3, @RW0+d16	XCH R4, @RW0' @RW0+d16	XCH R4, @RW0+d16	XCH R5, @RW0' @RW0+d16	XCH R5, @RW0+d16	XCH R6, @RW0' @RW0+d16	XCH R6, @RW0+d16	XCH R7, @RW0' @RW0+d16	XCH R7, @RW0+d16
+9	XCH R0, @RW1' @RW1+d16	XCH R0, @RW1+d16	XCH R1, @RW1' @RW1+d16	XCH R1, @RW1' @RW1+d16	XCH R2, @RW1' @RW1+d16	XCH R2, @RW1+d16	XCH R3, @RW1' @RW1+d16	XCH R3, @RW1+d16	XCH R4, @RW1' @RW1+d16	XCH R4, @RW1+d16	XCH R5, @RW1' @RW1+d16	XCH R5, @RW1+d16	XCH R6, @RW1' @RW1+d16	XCH R6, @RW1+d16	XCH R7, @RW1' @RW1+d16	XCH R7, @RW1+d16
+A	XCH R0, @RW2' W2-d16, A	XCH R0, W2-d16, A	XCH R1, @RW2' W2-d16, A	XCH R1, @RW2' W2-d16, A	XCH R2, @RW2' W2-d16, A	XCH R2, W2-d16, A	XCH R3, @RW2' W2-d16, A	XCH R3, W2-d16, A	XCH R4, @RW2' W2-d16, A	XCH R4, W2-d16, A	XCH R5, @RW2' W2-d16, A	XCH R5, W2-d16, A	XCH R6, @RW2' W2-d16, A	XCH R6, W2-d16, A	XCH R7, @RW2' W2-d16, A	XCH R7, W2-d16, A
+B	XCH R0, @RW3' @RW3+d16	XCH R0, @RW3+d16	XCH R1, @RW3' @RW3+d16	XCH R1, @RW3' @RW3+d16	XCH R2, @RW3' @RW3+d16	XCH R2, @RW3+d16	XCH R3, @RW3' @RW3+d16	XCH R3, @RW3+d16	XCH R4, @RW3' @RW3+d16	XCH R4, @RW3+d16	XCH R5, @RW3' @RW3+d16	XCH R5, @RW3+d16	XCH R6, @RW3' @RW3+d16	XCH R6, @RW3+d16	XCH R7, @RW3' @RW3+d16	XCH R7, @RW3+d16
+C	XCH R0, @RW0+ ' @RW0+RW7	XCH R0, @RW0+ ' @RW0+RW7	XCH R1, @RW0+ ' @RW0+RW7	XCH R1, @RW0+ ' @RW0+RW7	XCH R2, @RW0+ ' @RW0+RW7	XCH R2, @RW0+RW7	XCH R3, @RW0+ ' @RW0+RW7	XCH R3, @RW0+RW7	XCH R4, @RW0+ ' @RW0+RW7	XCH R4, @RW0+RW7	XCH R5, @RW0+ ' @RW0+RW7	XCH R5, @RW0+RW7	XCH R6, @RW0+ ' @RW0+RW7	XCH R6, @RW0+RW7	XCH R7, @RW0+ ' @RW0+RW7	XCH R7, @RW0+RW7
+D	XCH R0, @RW1+ ' @RW1+RW7	XCH R0, @RW1+ ' @RW1+RW7	XCH R1, @RW1+ ' @RW1+RW7	XCH R1, @RW1+ ' @RW1+RW7	XCH R2, @RW1+ ' @RW1+RW7	XCH R2, @RW1+RW7	XCH R3, @RW1+ ' @RW1+RW7	XCH R3, @RW1+RW7	XCH R4, @RW1+ ' @RW1+RW7	XCH R4, @RW1+RW7	XCH R5, @RW1+ ' @RW1+RW7	XCH R5, @RW1+RW7	XCH R6, @RW1+ ' @RW1+RW7	XCH R6, @RW1+RW7	XCH R7, @RW1+ ' @RW1+RW7	XCH R7, @RW1+RW7
+E	XCH R0, @RW2+ ' @PC+dt16	XCH R0, @PC+dt16	XCH R1, @RW2+ ' @PC+dt16	XCH R1, @RW2+ ' @PC+dt16	XCH R2, @RW2+ ' @PC+dt16	XCH R2, @PC+dt16	XCH R3, @RW2+ ' @PC+dt16	XCH R3, @PC+dt16	XCH R4, @RW2+ ' @PC+dt16	XCH R4, @PC+dt16	XCH R5, @RW2+ ' @PC+dt16	XCH R5, @PC+dt16	XCH R6, @RW2+ ' @PC+dt16	XCH R6, @PC+dt16	XCH R7, @RW2+ ' @PC+dt16	XCH R7, @PC+dt16
+F	XCH R0, @RW3+ ' R0, addr16	XCH R0, @RW3+ ' R0, addr16	XCH R1, @RW3+ ' R1, addr16	XCH R1, @RW3+ ' R1, addr16	XCH R2, @RW3+ ' R2, addr16	XCH R2, @RW3+R2, addr16	XCH R3, @RW3+ ' R3, addr16	XCH R3, @RW3+R3, addr16	XCH R4, @RW3+ ' R4, addr16	XCH R4, @RW3+R4, addr16	XCH R5, @RW3+ ' R5, addr16	XCH R5, @RW3+R5, addr16	XCH R6, @RW3+ ' R6, addr16	XCH R6, @RW3+R6, addr16	XCH R7, @RW3+ ' R7, addr16	XCH R7, @RW3+R7, addr16

表 B.9-21 XCHW RWi, ea 命令 (第 1 バイト = 7FH)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	XCHW RW0, RW0	XCHW RW0, @RW0-d8	XCHW RW1, RW0	XCHW RW1, @RW0+d8	XCHW RW2, RW0	XCHW RW2, @RW0+d8	XCHW RW3, RW0	XCHW RW3, @RW0-d8	XCHW RW4, RW0	XCHW RW4, @RW0-d8	XCHW RW5, RW0	XCHW RW5, @RW0-d8	XCHW RW6, RW0	XCHW RW6, @RW0+d8	XCHW RW7, RW0	XCHW RW7, @RW0+d8
+1	XCHW RW0, RW1	XCHW RW0, @RW1-d8	XCHW RW1, RW1	XCHW RW1, @RW1+d8	XCHW RW2, RW1	XCHW RW2, @RW1+d8	XCHW RW3, RW1	XCHW RW3, @RW1-d8	XCHW RW4, RW1	XCHW RW4, @RW1+d8	XCHW RW5, RW1	XCHW RW5, @RW1-d8	XCHW RW6, RW1	XCHW RW6, @RW1+d8	XCHW RW7, RW1	XCHW RW7, @RW1+d8
+2	XCHW RW0, RW2	XCHW RW0, @RW2-d8	XCHW RW1, RW2	XCHW RW1, @RW2+d8	XCHW RW2, RW2	XCHW RW2, @RW2+d8	XCHW RW3, RW2	XCHW RW3, @RW2-d8	XCHW RW4, RW2	XCHW RW4, @RW2+d8	XCHW RW5, RW2	XCHW RW5, @RW2-d8	XCHW RW6, RW2	XCHW RW6, @RW2+d8	XCHW RW7, RW2	XCHW RW7, @RW2+d8
+3	XCHW RW0, RW3	XCHW RW0, @RW3-d8	XCHW RW1, RW3	XCHW RW1, @RW3+d8	XCHW RW2, RW3	XCHW RW2, @RW3+d8	XCHW RW3, RW3	XCHW RW3, @RW3-d8	XCHW RW4, RW3	XCHW RW4, @RW3+d8	XCHW RW5, RW3	XCHW RW5, @RW3-d8	XCHW RW6, RW3	XCHW RW6, @RW3+d8	XCHW RW7, RW3	XCHW RW7, @RW3+d8
+4	XCHW RW0, RW4	XCHW RW0, @RW4-d8	XCHW RW1, RW4	XCHW RW1, @RW4+d8	XCHW RW2, RW4	XCHW RW2, @RW4+d8	XCHW RW3, RW4	XCHW RW3, @RW4-d8	XCHW RW4, RW4	XCHW RW4, @RW4+d8	XCHW RW5, RW4	XCHW RW5, @RW4+d8	XCHW RW6, RW4	XCHW RW6, @RW4+d8	XCHW RW7, RW4	XCHW RW7, @RW4+d8
+5	XCHW RW0, RW5	XCHW RW0, @RW5-d8	XCHW RW1, RW5	XCHW RW1, @RW5+d8	XCHW RW2, RW5	XCHW RW2, @RW5+d8	XCHW RW3, RW5	XCHW RW3, @RW5-d8	XCHW RW4, RW5	XCHW RW4, @RW5+d8	XCHW RW5, RW5	XCHW RW5, @RW5-d8	XCHW RW6, RW5	XCHW RW6, @RW5+d8	XCHW RW7, RW5	XCHW RW7, @RW5+d8
+6	XCHW RW0, RW6	XCHW RW0, @RW6-d8	XCHW RW1, RW6	XCHW RW1, @RW6+d8	XCHW RW2, RW6	XCHW RW2, @RW6+d8	XCHW RW3, RW6	XCHW RW3, @RW6-d8	XCHW RW4, RW6	XCHW RW4, @RW6+d8	XCHW RW5, RW6	XCHW RW5, @RW6-d8	XCHW RW6, RW6	XCHW RW6, @RW6+d8	XCHW RW7, RW6	XCHW RW7, @RW6+d8
+7	XCHW RW0, RW7	XCHW RW0, @RW7-d8	XCHW RW1, RW7	XCHW RW1, @RW7+d8	XCHW RW2, RW7	XCHW RW2, @RW7+d8	XCHW RW3, RW7	XCHW RW3, @RW7-d8	XCHW RW4, RW7	XCHW RW4, @RW7+d8	XCHW RW5, RW7	XCHW RW5, @RW7-d8	XCHW RW6, RW7	XCHW RW6, @RW7+d8	XCHW RW7, RW7	XCHW RW7, @RW7+d8
+8	XCHW RW0, RW0	XCHW RW0, @RW0-d16	XCHW RW1, RW1	XCHW RW1, @RW0+d16	XCHW RW2, RW0	XCHW RW2, @RW0+d16	XCHW RW3, RW0	XCHW RW3, @RW0-d16	XCHW RW4, RW0	XCHW RW4, @RW0-d16	XCHW RW5, RW0	XCHW RW5, @RW0-d16	XCHW RW6, RW0	XCHW RW6, @RW0+d16	XCHW RW7, RW0	XCHW RW7, @RW0+d16
+9	XCHW RW0, RW0	XCHW RW0, @RW1-d16	XCHW RW1, RW1	XCHW RW1, @RW1+d16	XCHW RW2, RW1	XCHW RW2, @RW1+d16	XCHW RW3, RW1	XCHW RW3, @RW1-d16	XCHW RW4, RW1	XCHW RW4, @RW1+d16	XCHW RW5, RW1	XCHW RW5, @RW1-d16	XCHW RW6, RW1	XCHW RW6, @RW1+d16	XCHW RW7, RW1	XCHW RW7, @RW1+d16
+A	XCHW RW0, RW2	XCHW RW0, @RW2-d16	XCHW RW1, RW2	XCHW RW1, @RW2+d16	XCHW RW2, RW2	XCHW RW2, @RW2+d16	XCHW RW3, RW2	XCHW RW3, @RW2-d16	XCHW RW4, RW2	XCHW RW4, @RW2+d16	XCHW RW5, RW2	XCHW RW5, @RW2-d16	XCHW RW6, RW2	XCHW RW6, @RW2+d16	XCHW RW7, RW2	XCHW RW7, @RW2+d16
+B	XCHW RW0, RW3	XCHW RW0, @RW3-d16	XCHW RW1, RW3	XCHW RW1, @RW3+d16	XCHW RW2, RW3	XCHW RW2, @RW3+d16	XCHW RW3, RW3	XCHW RW3, @RW3-d16	XCHW RW4, RW3	XCHW RW4, @RW3+d16	XCHW RW5, RW3	XCHW RW5, @RW3-d16	XCHW RW6, RW3	XCHW RW6, @RW3+d16	XCHW RW7, RW3	XCHW RW7, @RW3+d16
+C	XCHW RW0, RW0	XCHW RW0, @RW0-RW7	XCHW RW1, RW1	XCHW RW1, @RW0+RW7	XCHW RW2, RW0	XCHW RW2, @RW0+RW7	XCHW RW3, RW0	XCHW RW3, @RW0-RW7	XCHW RW4, RW0	XCHW RW4, @RW0+RW7	XCHW RW5, RW0	XCHW RW5, @RW0-RW7	XCHW RW6, RW0	XCHW RW6, @RW0+RW7	XCHW RW7, RW0	XCHW RW7, @RW0-RW7
+D	XCHW RW0, RW0	XCHW RW0, @RW1+RW7	XCHW RW1, RW1	XCHW RW1, @RW1+RW7	XCHW RW2, RW1	XCHW RW2, @RW1+RW7	XCHW RW3, RW1	XCHW RW3, @RW1-RW7	XCHW RW4, RW1	XCHW RW4, @RW1+RW7	XCHW RW5, RW1	XCHW RW5, @RW1+RW7	XCHW RW6, RW1	XCHW RW6, @RW1+RW7	XCHW RW7, RW1	XCHW RW7, @RW1+RW7
+E	XCHW RW0, RW2	XCHW RW0, @PC-d16	XCHW RW1, RW2	XCHW RW1, @PC+d16	XCHW RW2, RW2	XCHW RW2, @PC+d16	XCHW RW3, RW2	XCHW RW3, @PC-d16	XCHW RW4, RW2	XCHW RW4, @PC+d16	XCHW RW5, RW2	XCHW RW5, @PC-d16	XCHW RW6, RW2	XCHW RW6, @PC+d16	XCHW RW7, RW2	XCHW RW7, @PC+d16
+F	XCHW RW0, RW3	XCHW RW0, add16	XCHW RW1, RW3	XCHW RW1, add16	XCHW RW2, RW3	XCHW RW2, @RW3+RW2, add16	XCHW RW3, RW3	XCHW RW3, add16	XCHW RW4, @RW3+RW4, add16	XCHW RW4, add16	XCHW RW5, @RW3+RW5, add16	XCHW RW5, add16	XCHW RW6, @RW3+RW6, add16	XCHW RW6, add16	XCHW RW7, @RW3+RW7, add16	XCHW RW7, add16



# 索引

## <数字>

### 1

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	212
16ビットアウトプットコンペアの動作	211
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	213
16ビットタイマレジスタ	
16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR)	221
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	198
16ビット入出力タイマ	
16ビット入出力タイマのレジスタ	199
16ビットフリーランタイマ	
16ビットフリーランタイマ(×1)	196
16ビットフリーランタイマのカウントタイミング	210
16ビットフリーランタイマの動作	209
16ビットリロードタイマ	
16ビットリロードタイマ(イベントカウント機能付)の概要	216
16ビットリロードタイマ(イベントカウント機能付)のブロックダイアグラム	216
16ビットリロードタイマ(イベントカウント機能付)のレジスタ	217
16ビットリロードレジスタ(TMRLR)	
16ビットタイマレジスタ(TMR)/16ビットリロードレジスタ(TMRLR)	221
1Mビットフラッシュメモリ	
1Mビットフラッシュメモリの特長	380

### 8

8/16ビットPPG	
8/16ビットPPGの動作モード	240
8/16ビットPPG	
8/16ビットPPGの概要	228
8/16ビットPPGの動作	238
8/16ビットPPGのレジスタ	231
8/16ビットPPGの割込み	238
8ビットPPG	
8ビットPPGのブロックダイアグラム	229

## <アルファベット>

### A

A/Dコンバータ	
A/Dコンバータ使用上の注意	258
A/Dコンバータの概要	258
A/Dコンバータの電源端子の処理	19
A/Dコンバータのブロックダイアグラム	259
A/Dコンバータのレジスタ	260
ADCR1,2	
データレジスタ(ADCR1,2)	265
ADCS1,2	
コントロールステータスレジスタ(ADCS1,2)	261
ADER	
ポート5 アナログ入力許可レジスタ(ADER)	145
ARSR	
自動レディ機能選択レジスタ(ARSR)	121

### B

BAP	
バッファアドレスポインタ(BAP)	68

### C

CDCR	
クロック分周コントロールレジスタ(CDCR)	280
CKSCR	
クロック選択レジスタ(CKSCR)	86
CLKR	
クロック出力許可レジスタ(CLKR)	365
CLK同期モード	
CLK同期モード使用時のレジスタの設定値	300
CMR	
コモンレジスタバンクプリフィックス(CMR)	41
CMRH	
コマンドレジスタ上位8ビット(CMRH)	316
CMRL	
コマンドレジスタ下位8ビット(CMRL)	318
CPU間欠動作機能	
CPU間欠動作機能	96

## D

<b>D/Aコントロールレジスタ</b>	
D/Aコントロールレジスタ (DACR0/1).....	277
<b>D/Aコンバータ</b>	
D/Aコンバータの電源端子の処理.....	19
<b>D/Aコンバータ</b>	
D/Aコンバータの動作.....	278
D/Aコンバータのブロックダイアグラム	
.....	275
D/Aコンバータのレジスタ一覧.....	274
<b>D/Aコンバータレジスタ</b>	
D/Aコンバータレジスタ (DAT0/1).....	276
<b>DACR0/1</b>	
D/Aコントロールレジスタ (DACR0/1).....	277
<b>DAT0/1</b>	
D/Aコンバータレジスタ (DAT0/1).....	276
<b>DCRR</b>	
同報コントロールビットリードレジスタ (DCRR).....	326
<b>DCT</b>	
データカウンタ (DCT) .....	66
<b>DCWR</b>	
同報コントロールビット設定レジスタ (DCWR) .....	314
<b>DDR<sub>x</sub></b>	
ポート方向レジスタ (DDR <sub>x</sub> ).....	141
<b>DERR</b>	
電文長ビットリードレジスタ下位8ビット (DERR).....	327
<b>DEWR</b>	
電文長ビット設定レジスタ (DEWR).....	315
<b>DIRR</b>	
遅延割込み発生モジュールのレジスタ (DIRR) .....	254
<b>DIV A, Ri 命令</b>	
「DIV A, Ri」, 「DIVW A Rwi」命令の使用上の注意事項.....	22
<b>DIVR</b>	
分周比コントロールレジスタ (DIVR).....	175
<b>DIVW A Rwi 命令</b>	
「DIV A, Ri」, 「DIVW A Rwi」命令の使用上の注意事項.....	22
<b>DTP</b>	
DTP/外部割込みのブロックダイアグラム .....	246
DTP/外部割込みのレジスタ.....	246
DTPの動作.....	249
外部割込み/DTPの動作手順.....	251
<b>DTP許可レジスタ</b>	
割込み/DTP許可レジスタ (ENIR).....	247
<b>DTP要求の切替え</b>	
外部割込み要求とDTP要求の切替え.....	250

## E

<b>ECSR</b>	
バス制御信号選択レジスタ (ECSR).....	124
<b>EEPROMメモリマップ</b>	
EEPROMメモリマップ.....	372
<b>EI<sup>2</sup>OS</b>	
EI <sup>2</sup> OSを使った変換動作.....	268
<b>EI<sup>2</sup>OSステータスレジスタ</b>	
EI <sup>2</sup> OSステータスレジスタ (ISCS) .....	67
<b>EIRR</b>	
割込み/DTP要因レジスタ (EIRR).....	247
<b>ELVR</b>	
要求レベル設定レジスタ (ELVR).....	248
<b>ENIR</b>	
割込み/DTP許可レジスタ (ENIR).....	247

## F

<b>F<sup>2</sup>MC-16LX</b>	
F <sup>2</sup> MC-16LX 命令一覧表 .....	446
<b>FMCS</b>	
コントロールステータスレジスタ (FMCS) .....	382

## H

<b>HACR</b>	
外部アドレス出力制御レジスタ (HACR) .....	123

## I

<b>I/Oアクセス命令</b>	
I/Oアクセス命令.....	40
<b>I/Oポート</b>	
I/Oポートの概要.....	134
I/Oポートのレジスタ.....	137
I/Oポートブロックダイアグラム.....	135
<b>I/Oマップ</b>	
I/Oマップ.....	416
<b>I/Oレジスタアドレスポインタ</b>	
I/Oレジスタアドレスポインタ (IOA).....	67
<b>ICR</b>	
割込み制御レジスタ (ICR).....	63
<b>ICS23/01</b>	
コントロールステータスレジスタ (ICS23/01) .....	207
<b>IEBus<sup>TM</sup></b>	
IEBus <sup>TM</sup> 未使用時のTX, RX端子の処理.....	19
<b>IEBus<sup>TM</sup>コントローラ</b>	
IEBus <sup>TM</sup> コントローラの送信時の動作.....	343
IEBus <sup>TM</sup> コントローラの特長 .....	308
IEBus <sup>TM</sup> コントローラのブロックダイアグラム .....	309

IEBus™コントローラのレジスタ	310
<b>IEBus™プロトコル動作</b>	
IEBus™プロトコル動作の概要	345
<b>IOA</b>	
I/Oレジスタアドレスポインタ (IOA)	67
<b>IPCP0~3</b>	
インプットキャプチャデータレジスタ (IPCP0~3)	207
<b>ISCS</b>	
EI <sup>2</sup> OSステータスレジスタ (ISCS)	67
<b>ISD</b>	
拡張インテリジェントI/Oサービスディスクリプタ (ISD)	66

## L

<b>LPMCR</b>	
低消費電力モード制御レジスタ (LPMCR)	84
<b>LRRH, LRRL</b>	
ロックリードレジスタ (LRRH, LRRL)	324

## M

<b>MARH, MARL</b>	
マスタアドレスリードレジスタ (MARH, MARL)	325
<b>MAWH, MAWL</b>	
自局アドレス設定レジスタ (MAWH, MAWL)	312
<b>MOV ILM, #imm8</b>	
MOV ILM, #imm8	40

## N

<b>NCC</b>	
フラグ変化抑止プリフィックス (NCC)	41

## O

<b>OCCP0, 1</b>	
コンペアレジスタ (OCCP0, 1)	204
<b>OCS0, 1</b>	
コントロールステータスレジスタ (OCS0, 1)	204
<b>ODR4</b>	
ポート4出力端子レジスタ (ODR4)	143

## P

<b>PACSR</b>	
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)	369

<b>PADRO/PADR1</b>	
プログラムアドレス検出レジスタ (PADRO/PADR1)	369
<b>PDRx</b>	
ポートデータレジスタ (PDRx)	139
<b>PLLクロック</b>	
メインクロック/PLLクロックの切り替え	98
<b>POPW ps</b>	
POPW ps	40
<b>PPG0, 1出力端子制御レジスタ</b>	
PPG0, 1出力端子制御レジスタ (PPGOE)	236
<b>PPG0動作モード制御レジスタ</b>	
PPG0動作モード制御レジスタ (PPGC0)	232
<b>PPGC0</b>	
PPG0動作モード制御レジスタ (PPGC0)	232
<b>PPGC1</b>	
PPG1動作モード制御レジスタ (PPGC1)	234
<b>PPGOE</b>	
PPG0, 1出力端子制御レジスタ (PPGOE)	236
<b>PPG出力動作</b>	
PPG出力動作	241
<b>PRLL/PRLH</b>	
リロードレジスタ (PRLL/PRLH)	237
<b>PWCR</b>	
PWCデータバッファレジスタ (PWCR)	174
モード変更時のPWCRとタイマの値	193
<b>PWCSR</b>	
PWCコントロールステータスレジスタ (PWCSR)	169
<b>PWCSRのSTRTビットとSTOPビット</b>	
PWCSRのSTRTビットとSTOPビット	192
<b>PWCコントロールステータスレジスタ</b>	
PWCコントロールステータスレジスタ (PWCSR)	169
<b>PWCタイマ</b>	
PWCタイマの動作	166
PWCタイマの特長	166
PWCタイマのブロックダイアグラム	167
PWCタイマのレジスタ一覧	168
<b>PWCデータバッファレジスタ</b>	
PWCデータバッファレジスタ (PWCR)	174
<b>PWCノイズフィルタレジスタ</b>	
PWCノイズフィルタレジスタ (RNCR)	176

## R

<b>RDB</b>	
リードデータバッファ (RDB)	328
<b>RDR0, 1, 6</b>	
入力プルアップ抵抗設定レジスタ (RDR0, 1, 6)	144

RETI	
RETI .....	40
RNCR	
PWCノイズフィルタレジスタ (RNCR) .....	176
ROMM	
ROMミラー機能選択レジスタ (ROMM) .....	377

## S

SAWH, SAWL	
スレーブアドレス設定レジスタ (SAWH, SAWL) .....	313
SCR0~4	
シリアルコントロールレジスタ (SCR0~4) .....	289
SIDR0~4	
シリアルインプットデータレジスタ (SIDR0~4)/シリアルアウトプットデータレジスタ (SODR0~4) の構成 .....	291
SMR0~4	
シリアルモードレジスタ (SMR0~4) .....	287
SODR0~4	
シリアルインプットデータレジスタ (SIDR0~4)/シリアルアウトプットデータレジスタ (SODR0~4) .....	291
SSR	
スレーブステータス (SSR) の読み込み (コントロールビット : 0H, 6H) .....	359
SSR0~4	
シリアルステータスレジスタ (SSR0~4) .....	292
STOPビット	
PWCSRのSTRTビットとSTOPビット .....	192
STRH	
ステータスレジスタ上位8ビット (STRH) .....	320
STRL	
ステータスレジスタ下位8ビット (STRL) .....	322
STRTビット	
PWCSRのSTRTビットとSTOPビット .....	192

## T

TBTC	
タイムベースタイマ制御レジスタ (TBTC) .....	150
TCCS	
タイマコントロールステータスレジスタ (TCCS) .....	201
TCDTH, TCDTL	
タイマデータレジスタ (TCDTH, TCDTL) .....	201
TMCSR	
タイマコントロールステータスレジスタ (TMCSR) .....	218

TMR	
16ビットタイマレジスタ (TMR)/16ビットリロードレジスタ (TMRLR) .....	221
TMRLR	
16ビットタイマレジスタ (TMR)/16ビットリロードレジスタ (TMRLR) .....	221

## U

UART	
UART (モード1 使用時) の応用例 .....	304
UARTの動作 .....	294
UARTの特長 .....	284
UARTのブロックダイヤグラム .....	285
UARTのレジスタ .....	286

## W

WDB	
ライトデータバッファ (WDB) .....	327
WDTC	
ウォッチドッグタイマ制御レジスタ (WDTC) .....	155
WTC	
時計タイマ制御レジスタ (WTC) .....	162

<五十音>

## あ

アービトレーション	
バス占有権の決定(アービトレーション)	346
アウトプットコンペア	
アウトプットコンペア (×2)	196
アキュムレータ	
アキュムレータ (A)	30
アクリリジビット	
アクリリジビット	356
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)	26
アドレス一致検出機能	
アドレス一致検出機能の動作	371
アドレス一致検出機能の	
ブロックダイアグラム	368
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	372
アドレッシング	
アドレッシング	423
間接アドレッシング	432
直接アドレッシング	425
アンダフロー動作	
アンダフロー動作	223

## い

インターバル割込み機能	
インターバル割込み機能	152
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)	34
インプットキャプチャ	
インプットキャプチャ (×4)	197
インプットキャプチャ入力タイミグ	
インプットキャプチャ入力タイミグ	213

## う

ウォッチドッグ	
ウォッチドッグ停止	157
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	157
ウォッチドッグタイマのクリア	157
ウォッチドッグタイマの	
ブロックダイアグラム	154
ウォッチドッグタイマのレジスター一覧	154
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)	155

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	157

## か

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)	123
外部イベントカウンタ	
外部イベントカウンタ	222
外部クロック	
外部クロック	296
サブクロックモードと外部クロックの使用	21
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)	119
外部メモリアクセス (外部バス端子制御回路)	
のブロックダイアグラム	119
外部メモリアクセス (外部バス端子制御回路)	
のレジスタ	120
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)	119
外部メモリアクセス (外部バス端子制御回路)	
のブロックダイアグラム	119
外部メモリアクセス (外部バス端子制御回路)	
のレジスタ	120
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	127
外部割込み	
DTP/外部割込みのブロックダイアグラム	246
DTP/外部割込みのレジスタ	246
外部割込み/DTPの動作手順	251
外部割込みの動作	249
外部割込み要求	
外部割込み要求とDTP要求の切替え	250
外部割込み要求レベル	
外部割込み要求レベル	251
カウンタ	
カウンタの動作状態	226
カウントクロック	
カウントクロックと最大周期	184
カウントクロックの選択	179
カウントクロックの選択に関する注意事項	242
書き込み	
フラッシュメモリ書き込み/消去の方法	380
フラッシュメモリ書き込み手順	392
書き込む	
フラッシュメモリヘデータを書き込む	392



**拡張インテリジェントI/Oサービス**

拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS)	294
拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) の概要	61
拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) の構造	62
拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) の実行時間	71

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

拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) の動作フロー	69
---	----

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

拡張インテリジェントI/Oサービス (EI <sup>2</sup> OS) 機能と割込み	223
---	-----

**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)	66
--------------------------------	----

**間接アドレッシング**

間接アドレッシング	432
-----------	-----

**き****擬似時計モード**

擬似時計モードの解除	92
擬似時計モードへの遷移	92

**く****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)	365
---------------------	-----

**クロック選択ビット**

クロック選択ビット	192
-----------	-----

**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
-----------------	----

**クロック分周コントロールレジスタ**

クロック分周コントロールレジスタ (CDCR)	280
-------------------------	-----

**クロックモニタ機能**

クロックモニタ機能のブロックダイアグラム	364
----------------------	-----

**こ****コマンドシーケンス**

コマンドシーケンス表	384
------------	-----

**コマンドレジスタ下位8ビット**

コマンドレジスタ下位8ビット (CMRL)	318
-----------------------	-----

**コマンドレジスタ上位8ビット**

コマンドレジスタ上位8ビット (CMRH)	316
-----------------------	-----

**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)	41
-------------------------	----

**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)	33
----------------------	----

**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 (ADCS1, 2)	261
コントロールステータスレジスタ (FMCS)	382
コントロールステータスレジスタ (ICS23/01)	207
コントロールステータスレジスタ (OCS0/1)	204

**コントロールビット**

スレーブステータス (SSR) の読み込み (コントロールビット: 0H, 6H)	359
データコマンド転送(コントロールビット: 読み込み(3H, 7H), 書き込み(AH, BH, EH, FH))	360

**コントロールビット受信**

マスタユニットによるコントロールビット 受信に関する注意事項	331
-----------------------------------	-----

**コンペアレジスタ**

コンペアレジスタ (OCCP0, 1)	204
---------------------	-----

**さ****サブクロック**

メインクロック/サブクロックの切り替え	98
---------------------	----

**サブクロックモード**

サブクロックモードと外部クロックの使用	21
---------------------	----

**し****自局アドレス設定レジスタ**

自局アドレス設定レジスタ (MAWH, MAWL)	312
---------------------------	-----

**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)	32
---------------------------------------	----

**実効アドレス**

実効アドレスフィールド	441
-------------	-----

**実効アドレスフィールド**

実効アドレスフィールド	424
-------------	-----

**実行サイクル数**

実行サイクル数	439
実行サイクル数計算方法	439

**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)	121
----------------------	-----

**周期**

パルス幅/周期測定の範囲	187
パルス幅/周期の計算	187

**受信動作**

受信動作	297
------	-----

出力端子機能	
出力端子機能.....	225
消去	
フラッシュメモリ書き込み/消去の方法	
.....	380
状態遷移	
状態遷移.....	101
初期設定処理ルーチン	
初期設定処理ルーチン.....	338
初期値	
各ハードウェアの初期値.....	238
シリアルアウトデータレジスタ	
シリアルインデータレジスタ	
(SIDR0~4)/シリアルアウトデータ	
レジスタ(SODR0~4)の構成.....	291
シリアルインデータレジスタ	
シリアルインデータレジスタ	
(SIDR0~4)/シリアルアウトデータ	
レジスタ(SODR0~4)の構成.....	291
シリアル書き込み接続	
MB90F583C/CAシリアル書き込み接続の	
基本構成.....	404
シリアル書き込み接続例	
シリアル書き込み接続例(ユーザ電源使用時)	
.....	407
シリアル書き込み接続例(ライターから	
電源供給時).....	409
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	
周波数について.....	405
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(SCR0~4)	
.....	289
シリアルステータスレジスタ	
シリアルステータスレジスタ(SSR0~4)	
.....	292
シリアルモードレジスタ	
シリアルモードレジスタ(SMR0~4).....	287
す	
スタートビット	
スタートビット.....	349
スタック	
スタック内へのレジスタの退避.....	54
スタック操作命令	
スタック操作命令(PUSHW, POPW).....	40
スタンバイ	
スタンバイからの復帰.....	251
ステータスレジスタ下位8ビット	
ステータスレジスタ下位8ビット(STRL)	
.....	322
ストップモード	
ストップモードの解除.....	94
ストップモードへの遷移.....	94

ストリング命令	
ストリング命令 [MOVS, MOVSW, SCEQ, SCWEQ,	
FILS, FILSW] .....	41
スリープモード	
スリープモードの解除.....	91
スリープモードへの遷移.....	91
スレーブアドレス設定レジスタ	
スレーブアドレス設定レジスタ(SAWH, SAWL)	
.....	313
スレーブアドレスフィールド	
スレーブアドレスフィールド.....	351
スレーブ受信	
スレーブ受信.....	334
スレーブ受信時	
スレーブ受信時(受信割込みが発生).....	336
スレーブステータス	
スレーブステータス(SSR)の読み込み(コント	
ロールビット: 0H, 6H).....	359
スレーブステータスとロックアドレスの送信	
.....	332
スレーブ送信	
スレーブ送信.....	331
スレーブデータ送信処理ルーチン	
スレーブデータ送信処理ルーチン.....	340

## せ

セクタ構成	
1Mビットフラッシュメモリのセクタ構成	
.....	381
セクタ消去	
フラッシュメモリのセクタ消去手順.....	395
フラッシュメモリのセクタ消去を	
一時停止する.....	397
フラッシュメモリのセクタ消去を再開する	
.....	398
フラッシュメモリの任意のデータを消去する	
(セクタ消去) フラッシュメモリ	
.....	395
セクタ消去タイマフラグ	
セクタ消去タイマフラグ(DQ3).....	389
専用レジスタ	
専用レジスタ.....	28

## そ

送信動作	
送信動作.....	298
測定	
測定モードと測定動作.....	189
測定結果データ	
測定結果データ.....	187
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	
.....	60
ソフトウェア割込みの概要.....	59

ソフトウェア割込みの構造.....	59
ソフトウェア割込みの動作.....	59

## た

### タイマ

タイマのクリア.....	182, 192
タイマの周期.....	184
タイマ／パルス幅測定の開始と停止.....	181
モード変更時のPWCRとタイマの値.....	193

### タイマ機能

タイマ機能.....	177
------------	-----

### タイマコントロールステータスレジスタ

タイマコントロールステータスレジスタ (TCCS).....	201
タイマコントロールステータスレジスタ (TMCSSR).....	218

### タイマ値とリロード値

タイマ値とリロード値.....	183
-----------------	-----

### タイマデータレジスタ

タイマデータレジスタ (TCDTH, TCDTL) .....	201
------------------------------------	-----

### タイマモード

タイマモードでの測定終了フラグ.....	192
タイマモード動作のフローチャート.....	185

### 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) .....	388
-------------------------------	-----

### タイムベースタイマ

タイムベースタイマの動作.....	152
タイムベースタイマのレジスタ一覧.....	148

### タイムベースタイマのブロックダイアグラム

タイムベースタイマのブロックダイアグラム .....	149
-------------------------------	-----

### 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) .....	150
---------------------------------	-----

### 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).....	36
-------------------------	----

### 多重割込み

多重割込み.....	54
------------	----

### 多バイト長データ

多バイト長データのアクセス.....	27
メモリ空間における多バイト長データの配置 .....	27

### 端子配列図

FPT-100P-M06の端子配列図.....	10
FTP-100P-M05端子配列図.....	9

### 単発測定

単発測定モードと連続測定モード.....	186
----------------------	-----

### 単発モード

単発モード.....	267
単発モード時のEI <sup>2</sup> OSの起動例.....	269

## 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動作.....	255
遅延割込み発生モジュールの ブロックダイアグラム.....	254
遅延割込み発生モジュールのレジスタ (DIRR) .....	254

### 遅延割込み要求ラッチ

遅延割込み要求ラッチの使用上の注意 .....	255
----------------------------	-----

### チップ消去

フラッシュメモリのデータを消去する (チップ消去).....	394
-----------------------------------	-----

### 直接アドレッシング

直接アドレッシング.....	425
----------------	-----

## つ

### 通信アドレス

通信アドレス.....	346
-------------	-----

### 通信開始

通信開始.....	300
-----------	-----

### 通信終了

通信終了.....	300
-----------	-----

### 通信プリスケアラ

通信プリスケアラ.....	295
---------------	-----

### 通信プリスケアラの動作

通信プリスケアラの動作.....	282
------------------	-----

### 通信モード

通信モード.....	346
------------	-----

## て

### 停止モード

停止モード.....	267
停止モード時のEI <sup>2</sup> OS起動例.....	271

### 低消費電力回路

低消費電力回路の概要.....	82
低消費電力回路のブロックダイアグラム .....	83

### 低消費電力制御回路

低消費電力制御回路の動作.....	89
-------------------	----

### 低消費電力モード

低消費電力モードの状態遷移図 (1系統品) .....	109
低消費電力モードの状態遷移図 (2系統品) .....	104

### 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
---------------------------------	----

### データカウンタ

データカウンタ (DCT).....	66
--------------------	----

データコマンド転送	
データコマンド転送(コントロールビット :	
読み込み(3H, 7H),	
書き込み(AH, BH, EH, FH)).....	360
データバンクレジスタ	
データバンクレジスタ (DTB).....	25
データフィールド	
データフィールド.....	354
データポーリングフラグ	
データポーリングフラグ (DQ7).....	386
データレジスタ	
データレジスタ (ADCR1, 2).....	265
電源端子	
A/Dコンバータの電源端子の処理.....	19
D/Aコンバータの電源端子の処理.....	19
電源端子.....	19
伝送データ	
伝送データ.....	358
転送データフォーマット	
転送データフォーマット.....	297
伝送プロトコル	
伝送プロトコル.....	348
伝送プロトコルのコントロールフィールド	
伝送プロトコルのコントロールフィールド	
.....	352
電文長ビット設定レジスタ	
電文長ビット設定レジスタ (DEWR).....	315
電文長ビットリードレジスタ下位8ビット	
電文長ビットリードレジスタ下位8ビット	
(DERR).....	327
電文長フィールド	
電文長フィールド.....	353
と	
動作モード	
動作モードの選択.....	180
同報コントロールビット設定レジスタ (DCWR)	
同報コントロールビット設定レジスタ (DCWR)	
.....	314
同報コントロールビットリードレジスタ	
同報コントロールビットリードレジスタ	
(DCRR).....	326
同報受信	
同報受信.....	334
同報受信時	
同報受信時(受信割込みが発生).....	336
同報通信	
同報通信.....	347
同報ビット	
同報ビット.....	349
トグルビットフラグ	
トグルビットフラグ (DQ6).....	387
時計タイマ	
時計タイマのインターバル割込み機能	
.....	164
時計タイマの動作.....	164

時計タイマのブロックダイアグラム.....	161
時計タイマのレジスタ.....	160
時計タイマ制御レジスタ	
時計タイマ制御レジスタ (WTC).....	162
時計モード	
時計モードの解除.....	93
時計モードへの遷移.....	93

## な

内部クロック動作	
内部クロック動作.....	222
内部タイマ	
内部タイマ.....	295

## に

入出力回路形式	
入出力回路形式.....	16
入力端子	
未使用入力端子の処理.....	19
入力端子機能	
入力端子機能 (内部クロックモード時)	
.....	224
入力抵抗レジスタ	
入力プルアップ抵抗設定レジスタ (RDRO, 1, 6)	
.....	144

## 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.....	385
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	
.....	95
ハードウェアスタンバイモードへの遷移	
.....	95
ハードウェア割込み	
ハードウェア割込み使用手順例.....	58
ハードウェア割込みの概要.....	53
ハードウェア割込みの構造.....	53
ハードウェア割込みの使用上の注意.....	54
ハードウェア割込みの動作.....	55
ハードウェア割込みの動作フロー.....	57
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).....	124
バスモード別メモリ空間	
バスモード別メモリ空間.....	117
バスモード別メモリ空間推奨設定例	
バスモード別メモリ空間推奨設定例.....	118
パッケージ外形寸法図	
FPT-100P-M05パッケージ外形寸法図.....	7
FPT-100P-M06パッケージ外形寸法図.....	8
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	
周波数について.....	405

<b>バッファアドレスポインタ</b>	
バッファアドレスポインタ (BAP).....	68
<b>パリティビット</b>	
パリティビット.....	355
<b>パルス</b>	
最小入力パルス幅.....	187
最短入力パルス幅.....	193
タイマ/パルス幅測定の開始と停止.....	181
パルスの端子出力の制御.....	243
パルス幅/周期測定の範囲.....	187
パルス幅/周期の計算.....	187
パルス幅測定動作のフローチャート.....	191
<b>パルス幅測定機能</b>	
パルス幅測定機能.....	178
<b>パルス幅測定モード</b>	
連続測定モードを用いたパルス幅測定モード.....	193
<b>バンクセレクトプリフィックス</b>	
バンクセレクトプリフィックス.....	40
<b>バンク方式</b>	
バンク方式によるアドレス指定.....	25
<b>バンクレジスタ</b>	
バンクレジスタ.....	37
<b>汎用レジスタ</b>	
汎用レジスタ.....	38
<b>ひ</b>	
<b>ビットフォーマット</b>	
ビットフォーマット.....	361
<b>品種構成</b>	
品種構成.....	5
<b>ふ</b>	
<b>不定</b>	
ポート0, 1からの出力が不定になる場合.....	21
<b>フラグ</b>	
5つのフラグ (PE/ORE/FRE/RDRF/TDRE) と2つの割込み要因.....	301
<b>フラグ変化抑止プリフィックス</b>	
フラグ変化抑止プリフィックス (NCC).....	41
<b>フラグ変更命令</b>	
フラグ変更命令.....	40
<b>フラッシュマイコンプログラマ</b>	
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).....	411
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).....	413
フラッシュマイコンプログラマのシステム構成 (株式会社ワイ・ディ・シー製).....	406
<b>フラッシュメモリ</b>	
フラッシュメモリのレジスタ.....	380

<b>フラッシュメモリ書込み</b>	
フラッシュメモリ書込み/消去の詳細説明.....	390
<b>フラッシュメモリ消去</b>	
フラッシュメモリ書込み/消去の詳細説明.....	390
<b>プリフィックスコード</b>	
プリフィックスコードが連続している場合.....	43
<b>プリフィックス命令</b>	
割込み抑止命令とプリフィックス命令に関する制約.....	42
<b>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</b>	
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).....	369
<b>プログラムアドレス検出レジスタ</b>	
プログラムアドレス検出レジスタ (PADRO/PADRI).....	369
<b>プログラムカウンタ</b>	
プログラムカウンタ (PC).....	35
<b>プログラムパッチ処理フロー図</b>	
プログラムパッチ処理フロー図.....	374
<b>プログラムパッチ処理例</b>	
プログラムパッチ処理例.....	373
<b>プログラムバンクレジスタ</b>	
プログラムバンクレジスタ (PCB).....	25
<b>プロセッサステータス</b>	
プロセッサステータス (PS).....	33
<b>ブロックダイアグラム</b>	
MB90580Cシリーズのブロックダイアグラム.....	6
<b>分周周期測定モード</b>	
分周周期測定モード.....	193
<b>分周比コントロールレジスタ</b>	
分周比コントロールレジスタ (DIVR).....	175

## へ

<b>ペリフェラル</b>	
DTPを用いた時の外部に接続するペリフェラルの条件.....	251
<b>変換データ保護機能</b>	
変換データ保護機能.....	272

## ほ

<b>ポート0, 1</b>	
ポート0, 1からの出力が不定になる場合.....	21
<b>ポート4出力端子レジスタ</b>	
ポート4出力端子レジスタ (ODR4).....	143
<b>ポート5アナログ入力許可レジスタ</b>	
ポート5アナログ入力許可レジスタ (ADER).....	145

ポートデータレジスタ	
ポートデータレジスタ (PDRx) .....	139
ポート方向レジスタ	
ポート方向レジスタ (DDRx) .....	141
ホールド機能	
ホールド機能 .....	131

## ま

マシクロックの初期化	
マシクロックの初期化 .....	98
マスタアドレスフィールド	
マスタアドレスフィールド .....	350
マスタアドレスリードレジスタ	
マスタアドレスリードレジスタ (MARH, MARL) .....	325
マスタ受信	
マスタ受信 .....	333
マスタ受信時	
マスタ受信時(受信割込みが発生) .....	335
マスタ受信処理ルーチン	
マスタ受信処理ルーチン .....	341
マスタスレーブデータ送信時	
マスタスレーブデータ送信時 (送信割込みが発生) .....	335
マスタ送信	
マスタ送信 .....	330
マスタ送信処理ルーチン	
マスタ送信処理ルーチン .....	339
マスタユニット	
マスタユニットによるコントロールビット 受信に関する注意事項 .....	331

## み

未定義命令	
未定義命令の実行による例外発生 .....	72

## め

命令	
命令の種類 .....	422
命令マップの構造 .....	460
命令一覧表	
F <sup>3</sup> MC-16LX 命令一覧表 .....	446
メインクロック	
メインクロック/PLLクロックの切り替え .....	98
メインクロック/サブクロックの切り替え .....	98
メインクロック発振安定待ち時間	
メインクロック発振安定待ち時間設定 .....	97
メインルーチン	
メインルーチン .....	337

メモリアクセスモード	
メモリアクセスモードの概要 .....	114
メモリ空間	
メモリ空間 .....	24

## も

モード端子	
モード端子 .....	115
モードデータ	
モードデータ .....	116

## ゆ

ユーザ	
シリアル書き込み接続例 (ユーザ電源使用時) .....	407
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) .....	26
ユーザスタックポインタ	
ユーザスタックポインタ (USP) とシステムスタックポインタ (SSP) .....	32
ユーザ電源	
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) .....	411

## よ

要求レベル設定レジスタ	
要求レベル設定レジスタ (ELVR) .....	248
読み出し	
フラッシュメモリを読み出し/リセット状態にする .....	391

## ら

ライター	
シリアル書き込み接続例 (ライターから電源供給時) .....	409
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) .....	413
ライトデータバッファ	
ライトデータバッファ (WDB) .....	329
ラッチアップ	
ラッチアップの防止 .....	19

## り

リードデータバッファ	
リードデータバッファ (RDB) .....	328
リセット	
フラッシュメモリを読み出し/リセット状態にする .....	391

リセット解除後の動作.....	77
<b>リセット入力</b>	
リセット入力によって初期化されないレジスタ.....	78
<b>リセット要因</b>	
リセット要因.....	75
<b>リニア方式</b>	
リニア方式によるアドレス指定.....	25
<b>リロード値</b>	
リロード値とパルス幅の関係.....	241
<b>リロード値</b>	
タイマ値とリロード値.....	183
<b>リロード動作モード</b>	
リロード動作モード.....	183
<b>リロードレジスタ</b>	
リロードレジスタ (PRL/PR LH).....	237
リロードレジスタへのライトタイミング.....	244
<b>れ</b>	
<b>例外</b>	
未定義命令の実行による例外発生.....	72
<b>レジスタ</b>	
レジスタの値の変更.....	192
<b>レジスタバンク</b>	
レジスタバンク.....	38
<b>レジスタバンクポインタ</b>	
レジスタバンクポインタ (RP).....	34
<b>レディ機能</b>	
レディ機能.....	129
<b>連続測定</b>	
単発測定モードと連続測定モード.....	186
<b>連続測定モード</b>	
連続測定モードを用いたパルス幅測定モード.....	193
<b>連続モード</b>	
連続モード.....	267
連続モード時のEI <sup>2</sup> OSの起動例.....	270

## ろ

<b>ロック</b>	
ロックの設定, 解除.....	360
<b>ロックアドレス</b>	
スレーブステータスとロックアドレスの送信.....	332
<b>ロックリードレジスタ</b>	
ロックリードレジスタ (LRRH, LRRL).....	324

## わ

<b>割込み</b>	
割込み/DTP許可レジスタ (ENIR).....	247
割込み/DTP要因レジスタ (EIRR).....	247
割込みの概要.....	48
<b>割込みシーケンス</b>	
割込みシーケンスの実行時間.....	56
<b>割込み処理ルーチン</b>	
割込み処理ルーチン.....	337
<b>割込み制御レジスタ</b>	
割込み制御レジスタ (ICR).....	63
<b>割込みフラグ</b>	
各動作モードによる割込みフラグのセットタイミング.....	301
<b>割込みベクタ</b>	
割込みベクタ.....	51
<b>割込み要因</b>	
5つのフラグ (PE/ORE/FRE/RDRF/TDRE) と2つの割込み要因.....	301
割込み要因.....	49
<b>割込み要求</b>	
割込み要求の発生.....	184, 188
<b>割込み抑止</b>	
割込み抑止命令とプリフィックス命令に関する制約.....	42
<b>割込み抑止命令</b>	
割込み抑止命令.....	42, 54
<b>ワンショット動作モード</b>	
ワンショット動作モード.....	183

GM44-10111-3

---

**富士通マイクロエレクトロニクス・CONTROLLER MANUAL**

F<sup>2</sup>MC-16LX  
16ビット・マイクロコントローラ  
MB90580Cseries  
ハードウェアマニュアル

---

2008年7月 第3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

---



