



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

商品仕様の継続性について

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

オーダ型格および品名について

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

詳しいお問い合わせ先

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

サイプレスについて

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

FR60Lite

32 ビット・マイクロコントローラ

MB91265A Series

ハードウェアマニュアル

FR60Lite

32 ビット・マイクロコントローラ

MB91265A Series

ハードウェアマニュアル

開発における最新の注意事項に関しては、必ず『Check Sheet』を参照してください。

『Check Sheet』は、下記サポートページに掲載されています。

URL : <http://jp.fujitsu.com/microelectronics/products/micom/support/index.html>

『Check Sheet』は、システム開発において、問題を未然に防ぐことを目的として、最低限必要と思われるチェック項目をリストにしたものです。

富士通株式会社

はじめに

■ 本書の目的と対象読者

富士通半導体製品につきまして、平素より格別のご愛顧を賜り厚くお礼申し上げます。

MB91265A シリーズは、32 ビット高性能 RISC CPU を使用し、高性能 / 高速な CPU 処理が要求される組込み制御用に各種 I/O リソースやバス制御機構を内蔵した、シングルチップマイクロコントローラです。32 ビット CPU がアクセスする広大なアドレス空間をサポートするために、外部バスアクセスが基本となっていますが、CPU の命令実行を高速化するために RAM (データ用) を内蔵しています。

デジタルビデオカメラ、ナビゲーションシステム、DVD プレーヤなど、高性能な CPU 処理パワーを要求される組込み用途に最適な仕様となっています。

MB91265A シリーズは、FR30/40 CPU をベースにバスアクセスの強化を行い、より高速での使用に対応した FR60 Lite です。

本書は、実際に MB91265A シリーズを使用して製品を開発される技術者を対象に、MB91265A シリーズの機能や動作について解説したものです。本書を一通り読んでください。

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

*: FR は、FUJITSU RISC controller の略で、富士通株式会社の製品です。

■ 商標

その他の記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

■ 本書の全体構成

本書は、以下に示す 20 の章、および付録から構成されています。

第 1 章 概要

この章では、MB91265A シリーズの特長、ブロックダイアグラム、外形寸法など全体を知るための基本的なことについて説明します。

第 2 章 デバイスの取扱いについて

この章では、デバイスを取り扱う際の注意事項について説明します。

第 3 章 CPU および制御部

この章では、MB91265A シリーズの CPU コアの機能を知るために、アーキテクチャ、仕様、命令などの基本的なことについて説明します。

第 4 章 I/O ポート

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

第 5 章 割込みコントローラ

この章では、割込みコントローラの概要、レジスタの構成 / 機能、および動作について説明します。

第 6 章 外部割込み・NMI 制御部

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

第 7 章 REALOS 関連ハード

この章では、遅延割込みモジュールおよびビットサーチモジュールの概要、レジスタ構成 / 機能、および動作について説明します。

第 8 章 リロードタイマ

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

第 9 章 タイミングジェネレータ

この章では、タイミングジェネレータの概要、レジスタの構成 / 機能、および動作について説明します。

第 10 章 PPG

この章では、PPG の概要、レジスタの構成 / 機能、および動作について説明します。

第 11 章 PWC (Pulse Width Count: パルス幅測定)

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

第 12 章 多機能タイマ

この章では、多機能タイマの概要、レジスタの構成 / 機能、および動作について説明します。

第 13 章 U-TIMER (UART ボーレート発生用 16 ビットタイマ)

この章では、U-TIMER の概要、レジスタの構成 / 機能、および動作について説明します。

第 14 章 UART

この章では、UART の概要、レジスタの構成 / 機能、および動作について説明します。

第 15 章 C-CAN

この章では、C-CAN の機能と動作について説明します。

第 16 章 8/10 ビット A/D コンバータ

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

第 17 章 積和演算回路

この章では、積和演算回路の概要、レジスタの構成 / 機能、および動作について説明します。

第 18 章 DMAC (DMA コントローラ)

この章では、DMAC の概要、レジスタの構成 / 機能、および動作について説明します。

第 19 章 フラッシュメモリ

この章では、フラッシュメモリの概要、レジスタの構成 / 機能、および動作について説明します。

第 20 章 シリアル書込み接続

この章では、シリアル書込み基本構成、シリアルオンボード書込みに使用する端子、シリアル書込み接続例、およびフラッシュマイコンプログラマシステム構成について説明します。

付録

付録では、各 CPU ステートにおける端子状態、リトル・エンディアン領域を利用する際の注意事項、FR ファミリの命令一覧、および MB91265A シリーズを使用する際の注意事項について説明します。

- 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、外国為替および外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

目次

第 1 章	概要	1
1.1	概要	2
1.2	ブロックダイアグラム	4
1.3	端子配列図	5
1.3.1	外形寸法図	6
1.4	端子機能一覧	7
1.5	入出力回路形式	15
第 2 章	デバイスの取扱いについて	19
2.1	デバイスの取扱いについて	20
第 3 章	CPU および制御部	23
3.1	メモリ空間	24
3.2	メモリマップ	25
3.3	内部アーキテクチャ	26
3.4	プログラミングモデル	31
3.4.1	レジスタ	32
3.5	データ構造	38
3.6	メモリマップ	40
3.7	分岐命令	42
3.8	EIT (例外・割込み・トラップ)	45
3.9	動作モード	59
3.10	リセット (デバイス初期化)	62
3.10.1	リセットレベル	63
3.10.2	リセット要因	64
3.10.3	リセットシーケンス	66
3.10.4	発振安定待ち時間	67
3.10.5	リセット動作モード	69
3.11	クロック生成制御	70
3.11.1	ソースクロックの選択	71
3.11.2	PLL 制御	72
3.11.3	発振安定待ち・PLL ロック待ち時間	73
3.11.4	クロック分配	75
3.11.5	クロック分周	77
3.11.6	クロック生成制御部のブロックダイアグラム	78
3.11.7	クロック生成制御部のレジスタ詳細説明	79
3.11.8	クロック制御部が持つ周辺回路	93
3.12	デバイス状態制御	96
第 4 章	I/O ポート	103
4.1	ポート基本ブロックダイアグラム	104
4.1.1	プルアップ抵抗付き入出力ポート	105
4.2	I/O ポートのレジスタ	106
4.3	アナログ入力ポート	112

第 5 章	割込みコントローラ	115
5.1	概要	116
5.2	レジスタ一覧	117
5.3	ブロックダイヤグラム	119
5.4	レジスタ詳細説明	120
5.5	割込みコントローラの動作	122
第 6 章	外部割込み・NMI 制御部	129
6.1	外部割込み・NMI 制御部	130
第 7 章	REALOS 関連ハード	139
7.1	遅延割込みモジュール	140
7.2	ビットサーチモジュール	142
第 8 章	リロードタイマ	149
8.1	概要	150
8.2	ブロックダイヤグラム	151
8.3	リロードタイマのレジスタ	152
8.4	リロードタイマの動作	156
8.5	注意点	160
第 9 章	タイミングジェネレータ	161
9.1	タイミングジェネレータ	162
第 10 章	PPG	167
10.1	概要	168
10.2	ブロックダイヤグラム	171
10.3	PPG のレジスタ	175
10.4	動作説明	181
第 11 章	PWC (Pulse Width Count: パルス幅測定)	187
11.1	概要	188
11.2	ブロックダイヤグラム	189
11.3	PWC のレジスタ	190
11.4	動作説明	197
第 12 章	多機能タイマ	207
12.1	概要	208
12.2	ブロックダイヤグラム	210
12.3	多機能タイマの端子	217
12.4	多機能タイマのレジスタ	218
12.4.1	コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2, CPCLRBL0 ~ CPCLRBL2) / コンペアクリアレジスタ (CPCLRH0 ~ CPCLRH2, CPCLRL0 ~ CPCLRL2)	224
12.4.2	タイマデータレジスタ (TCDTH0 ~ TCDTH2, TCDTL0 ~ TCDTL2)	226
12.4.3	タイマ状態制御レジスタ (TCCSH0 ~ TCCSH2, TCCSL0 ~ TCCSL2)	227
12.4.4	A/D トリガ制御レジスタ (ADTRGC)	233

12.4.5	アウトプットコンペアバッファレジスタ (OCCPBH0 ~ OCCPBH5, OCCPBL0 ~ OCCPBL5) / アウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5, OCCPL0 ~ OCCPL5)	235
12.4.6	コンペア制御レジスタ (OCSH0 ~ OCSH5, OCSL0 ~ OCSL5)	237
12.4.7	コンペアモード制御レジスタ (OCMOD)	244
12.4.8	インプットキャプチャデータレジスタ (IPCPH0 ~ IPCPH3, IPCPL0 ~ IPCPL3)	246
12.4.9	インプットキャプチャ状態制御 / PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01)	247
12.4.10	16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2)	255
12.4.11	16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2)	256
12.4.12	波形制御レジスタ (SIGCR1/SIGCR2)	265
12.4.13	A/D 起動コンペアレジスタ (ADCOMP1, ADCOMP2, ADCOMPC1, ADCOMPC2)	268
12.4.14	フリーランタイムセレクトレジスタ (FSR0 ~ FSR2)	270
12.5	多機能タイマ割込み	272
12.6	多機能タイマの動作	275
12.6.1	16 ビットフリーランタイムの動作	276
12.6.2	16 ビットアウトプットコンペアの動作	283
12.6.3	16 ビットインプットキャプチャの動作	294
12.6.4	波形ジェネレータの動作	296
12.6.5	A/D 起動コンペアの動作	308
12.7	多機能タイマの使用上の注意	310
12.8	多機能タイマのプログラム例	312
第 13 章	U-TIMER (UART ボーレート発生用 16 ビットタイマ)	315
13.1	概要	316
13.2	動作説明	320
第 14 章	UART	321
14.1	概要	322
14.2	レジスタ詳細説明	325
14.3	UART の動作	333
14.4	応用例	339
14.5	ボーレートと U-TIMER のリロード値の設定例	341
第 15 章	C-CAN	343
15.1	C-CAN の特長	344
15.2	C-CAN のブロックダイアグラム	345
15.3	C-CAN のレジスタ	346
15.4	C-CAN レジスタ機能	352
15.4.1	全体コントロールレジスタ	353
15.4.2	メッセージインタフェースレジスタ	366
15.4.3	メッセージオブジェクト	379
15.4.4	メッセージハンドラレジスタ	385
15.4.5	CAN クロックプリスケアラレジスタ	394
15.5	C-CAN 機能	395
15.5.1	メッセージオブジェクト	396
15.5.2	メッセージ送信動作	398

15.5.3	メッセージ受信動作	400
15.5.4	FIFO バッファ機能	403
15.5.5	割込み機能	405
15.5.6	ビットタイミング	407
15.5.7	テストモード	410
15.5.8	ソフトウェア初期化	414
15.5.9	CAN クロックプリスケアラ	415
第 16 章 8/10 ビット A/D コンバータ		419
16.1	概要	420
16.2	構成	421
16.3	端子	424
16.4	レジスタ	426
16.4.1	A/D チャンネル制御レジスタ (ADCH)	427
16.4.2	A/D モード設定レジスタ (ADM0)	429
16.4.3	A/D 制御ステータスレジスタ (ADCS)	432
16.4.4	A/D データレジスタ (ADCD)	435
16.4.5	アナログ入力制御レジスタ (AICR)	436
16.5	割込み	437
16.6	動作説明	438
16.7	A/D 変換データ保護機能	442
16.8	使用上の注意	443
第 17 章 積和演算回路		445
17.1	特長	446
17.2	命令定義	450
17.3	レジスタ説明	451
17.4	動作説明	457
17.5	命令詳細説明	460
第 18 章 DMAC (DMA コントローラ)		465
18.1	概要	466
18.2	レジスタ詳細説明	469
18.3	DMAC (DMA コントローラ) の動作	483
18.3.1	転送要求の設定	485
18.3.2	転送シーケンス	486
18.3.3	DMA 転送全般	488
18.4	動作フローチャート	497
18.5	データパス	499
第 19 章 フラッシュメモリ		501
19.1	概要	502
19.2	レジスタ	508
19.2.1	フラッシュメモリステータスレジスタ (FLCR)	509
19.2.2	フラッシュウェイトレジスタ (FLWC)	511
19.3	フラッシュメモリのアクセスモード	513
19.4	自動アルゴリズム起動方法	515
19.5	自動アルゴリズム実行状態	520
19.6	デュアルオペレーション動作	526

第 20 章	シリアル書き込み接続	529
20.1	概要	530
付録		535
付録 A	I/O マップ	536
付録 B	割込みベクタ	545
付録 C	各 CPU ステートにおける端子状態	548
付録 D	リトル・エンディアン領域を利用する際の注意事項	550
付録 E	命令一覧表	557
付録 F	使用上の注意	572
索引		575

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。）
2	1.1 概要 内蔵周辺機能 の変更 ("MB91F267NA" "MB91F267NA /MB91267NA") (" 表 1.1-1 内蔵周辺機能 " の挿入および置き換え)
4	図 1.2-1 ブロックダイアグラム の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
5	1.3 端子配列図 LQFP-64 (MB91F267A/MB91F267NA/MB91267NA/MB91267A) の変更 ("MB91266A" "MB91267NA/MB91267A"), ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
10	表 1.4-1 端子機能一覧表 (4 / 7) の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
13	表 1.4-1 端子機能一覧表 (7 / 7) の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
25	MB91F267A/MB91F267NA/MB91267NA/MB91267A のメモリマップ の変更 ("MB91F267A" "MB91F267A/MB91267A"), ("MB91F267NA" "MB91F267NA /MB91267NA")
25	3.2 メモリマップ MB91266A のメモリマップ の変更 (" 図 3.2-2 MB91266A のメモリマップ " の削除)
52	表 3.8-3 ベクタテーブル (3 / 3) の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
79	RSRR : リセット要因レジスタ / ウォッチドッグタイマ制御レジスタ の変更 ("(reserverd bit)" " - : 予約ビット")
80	RSRR : リセット要因レジスタ / ウォッチドッグタイマ制御レジスタ の変更 ("(reserved bit)" " - : 予約ビット")
83	STCR : スタンバイ制御レジスタ の変更 ("(reserverd bit)" " - : 予約ビット")
85	TBCR : タイムベースカウンタ制御レジスタ の変更 ("(reserved bit)" " - : 予約ビット")
87	CLKR : クロックソース制御レジスタ の変更 ("(reserved bit)" " - : 予約ビット")
92	DIVR1: 基本クロック分周設定レジスタ 1 の変更 ("(reserved bit)" " - : 予約ビット")

ページ	変更内容（詳細は本文を参照してください。）
109	4.2 I/O ポートのレジスタ Port Function Register(PFR: PFR0, PFR1, PTFR0) の変更 ("MB91F267NA のみ " "MB91F267NA /MB91267NA"), ("MB91F267NA " "MB91F267NA /MB91267NA")
110	4.2 I/O ポートのレジスタ Port Function Register(PFR: PFR0, PFR1, PTFR0) の変更 ("MB91F267NA のみ " "MB91F267NA /MB91267NA")
111	4.2 I/O ポートのレジスタ Port Function Register(PFR: PFR0, PFR1, PTFR0) の変更 ("MB91F267NA " "MB91F267NA /MB91267NA")
120	Interrupt Control Register (ICR) の変更 ("-: 未定義 " の追加)
121	Hold Request Cancel Level register (HRCL) の変更 ("-: 未定義 " の追加)
123	5.5 割込みコントローラの動作 表 5.5-1 割込み要因と割込み番号, 割込みレベルの関係 (2 / 4) の変更 ("MB91F267NA のみ " "MB91F267NA /MB91267NA")
140	DICR (Delayed Interrupt Control Register) の変更 ("-: 未定義 " の追加)
179	10.3 PPG のレジスタ PPG 起動レジスタ (TRG) の変更 ("PEN07 PEN06 PEN05 PEN04 PEN03 PEN02 PEN01 PEN00" "PEN7 PEN6 PEN5 PEN4 PEN3 PEN2 PEN1 PEN0")
179	10.3 PPG のレジスタ 出力反転レジスタ (REVC) の変更 ("REV07 REV06 REV05 REV04 REV03 REV02 REV01 REV00" "REV7 REV6 REV5 REV4 REV3 REV2 REV1 REV0")
190	PWC コントロール / ステータスレジスタ (PWCSR: PWCSR0) の変更 (" " "R/W")
190	PWC コントロール / ステータスレジスタ (PWCSR: PWCSR0) の変更 ("-: 未定義 " の削除)
192	PWC コントロール / ステータスレジスタ (PWCSR: PWCSR0) の変更 ("(reserved)" " - : 予約ビット")
318	U-TIMEr Control register (UTIMC: UTIMC0, UTIMC1) の変更 ("(reserved)" "-: 未定義ビット")
325	シリアルモードレジスタ (SMR: SMR0, SMR1) の変更 ("R/W" の追加), ("(reserved)" " - : 予約ビット"), (" 未定義 " " 予約 ")
326	シリアルモードレジスタ (SMR: SMR0, SMR1) の変更 ("(reserved)" " - : 予約ビット"), ("(reserved)" "-: 未定義ビット")

ページ	変更内容（詳細は本文を参照してください。）
347	表 15.3-1 全体コントロールレジスタ一覧 の変更 ("Int-Id" "IntId")
350	表 15.3-3 メッセージハンドラレジスタ一覧 の変更 ("32 メッセージバッファ以上をサポートするための予約領域 " を削除)
354	15.4.1.1 CAN 制御レジスタ (CTRLR) CAN 制御レジスタ (CTRLR) の変更 ("res" " - "), (": 予約ビット " " - : 予約ビット ")
355	15.4.1.1 CAN 制御レジスタ (CTRLR) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")
357	15.4.1.2 CAN ステータスレジスタ (STATR) CAN ステータスレジスタ (STATR) の変更 ("res" " - "), (" 予約ビット " " - : 予約ビット ")
361	15.4.1.4 CAN ビットタイミングレジスタ (BTR) CAN ビットタイミングレジスタ (BTR) ("res" " - ")
361	15.4.1.4 CAN ビットタイミングレジスタ (BTR) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")
363	15.4.1.6 CAN テストレジスタ (TESTR) CAN テストレジスタ (TESTR) の変更 ("res" " - ")
363	15.4.1.6 CAN テストレジスタ (TESTR) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")
364	15.4.1.6 CAN テストレジスタ (TESTR) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")
365	15.4.1.7 CAN プリスケーラ拡張レジスタ (BRPER) CAN プリスケーラ拡張レジスタ (BRPER) の変更 ("res" " - ")
365	15.4.1.7 CAN プリスケーラ拡張レジスタ (BRPER) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")
367	15.4.2.1 IFx コマンド要求レジスタ (IFxCREQ) IFx コマンド要求レジスタ (IFxCREQ) の変更 ("res" " - ")
368	15.4.2.1 IFx コマンド要求レジスタ (IFxCREQ) レジスタ機能 の変更 ("res: 予約ビット " " - : 予約ビット ")

ページ	変更内容（詳細は本文を参照してください。）
368	15.4.2.1 IFx コマンド要求レジスタ (IFxCREQ) レジスタ機能 の変更 ("[bit4 ~ bit0] Message Number: メッセージ番号 (128 message buffer CAN 用)" の記述削除)
369	15.4.2.2 IFx コマンドマスクレジスタ (IFxCMSK) IFx コマンドマスクレジスタ (IFxCMSK) の変更 ("res" " - ")
369	15.4.2.2 IFx コマンドマスクレジスタ (IFxCMSK) レジスタ機能 の変更 ("res: 予約ビット" " - : 予約ビット")
374	15.4.2.3 IFx マスクレジスタ 1,2 (IFxMSK1, IFxMSK2) IFx マスクレジスタ 1,2 (IFxMSK1, IFxMSK2) の変更 ("res" " - ")
376	15.4.2.5 IFx メッセージ制御レジスタ (IFxMCTR) IFx メッセージ制御レジスタ (IFxMCTR) の変更 ("res" " - ")
379	15.4.3 メッセージオブジェクト の変更 ("(品種によって 128 まで)" の削除)
387	15.4.4.1 CAN 送信要求レジスタ (TREQR1, TREQR2) レジスタ機能 の変更 ("32 メッセージバッファ以上 " の記述削除)
389	15.4.4.2 CAN データ更新レジスタ (NEWDT1, NEWDT2) レジスタ機能 の変更 ("32 メッセージバッファ以上 " の記述削除)
391	15.4.4.3 CAN 割込みペンディングレジスタ (INTPND1, INTPND2) レジスタ機能 の変更 ("32 メッセージバッファ以上 " の記述削除)
393	15.4.4.4 CAN メッセージ有効レジスタ (MSGVAL1, MSGVAL2) レジスタ機能 の変更 ("32 メッセージバッファ以上 " の記述削除)
394	15.4.5 CAN クロックプリスケアラレジスタ レジスタ機能 の変更 ("res" " - "), ("res: 予約ビット" " - : 予約ビット")
432	16.4.3 A/D 制御ステータスレジスタ (ADCS) A/D 制御ステータスレジスタ (ADCS: ADCS1, ADCS2) の変更 (" " " 予約 ")
434	16.4.3 A/D 制御ステータスレジスタ (ADCS) 表 16.4-3 A/D 制御ステータスレジスタ (ADCS) の各ビットの機能説明 (2 / 2) の変更 (" 未使用ビット" " 未定義ビット")

ページ	変更内容（詳細は本文を参照してください。）
469	18.2 レジスタ詳細説明 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 コントロール / ステータスレジスタ A[DMACA: DMACA0 ~ DMACA4] の変更 ("* マークは DMAC 転送中に設定すると..." の削除)
472	18.2 レジスタ詳細説明 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 コントロール / ステータスレジスタ A [DMACA: DMACA0 ~ DMACA4] の変更 ("(Reserved)" " ")
482	18.2 レジスタ詳細説明 DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 DMAC 全体制御レジスタ [DMACR] の変更 ("(Reserved)" " ")
509	19.2.1 フラッシュメモリステータスレジスタ (FLCR) 図 19.2-2 フラッシュメモリステータスレジスタ (FLCR) のビット構成 ("-: 未定義" の削除)
509	19.2.1 フラッシュメモリステータスレジスタ (FLCR) フラッシュメモリステータスレジスタ (FLCR) の変更 ("(reserved)" " - : 予約ビット")
510	19.2.1 フラッシュメモリステータスレジスタ (FLCR) 制限事項 の変更 ("(reserved)" " - : 予約ビット")
511	19.2.2 フラッシュウェイトレジスタ (FLWC) 図 19.2-3 フラッシュウェイトレジスタ (FLWC) のビット構成 の変更 ("-: 未定義" の削除)
511	19.2.2 フラッシュウェイトレジスタ (FLWC) フラッシュウェイトレジスタ (FLWC) の変更 ("(reserved)" " - : 予約ビット")
544	付録 A I/O マップ 付表 A-1 I/O マップ (8 / 8) の変更 (128 メッセージバッファ関連記述削除)
544	付録 A I/O マップ 付表 A-1 I/O マップ (8 / 8) の変更 ("Reserved" " 予約")
544	付録 A I/O マップ 付表 A-1 I/O マップ (8 / 8) の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
547	付録 B 割込みベクタ 付表 B-1 ベクタテーブル (3 / 3) の変更 ("MB91F267NA のみ" "MB91F267NA /MB91267NA")
549	付録 C 各 CPU ステートにおける端子状態 表 C-1 シングルチップモード ("MB91F267NA のみ" "MB91F267NA /MB91267NA")

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

MB91265A シリーズの特長，ブロックダイアグラム，外形寸法など全体を知るための基本的なことについて説明します。

- 1.1 概要
- 1.2 ブロックダイアグラム
- 1.3 端子配列図
- 1.4 端子機能一覧
- 1.5 入出力回路形式

1.1 概要

MB91265A シリーズは高速処理を要求される組込み制御用途向けに設計された、汎用の富士通 32 ビット RISC マイクロコントローラです。CPU には、FR ファミリと互換の FR60Lite を使用しています。

MB91F267NA/MB91267NA には、C-CAN (1 チャンネル) が搭載されています。

■ 特長

● FR60Lite CPU

- 32 ビット RISC, ロード / ストアアーキテクチャ, パイプライン 5 段
- 最大動作周波数: 33 MHz (原発振 4.192 MHz 原発振 8 通倍 (PLL クロック通倍方式))
- 16 ビット固定長命令 (基本命令)
- 命令実行速度: 1 命令 / 1 サイクル
- メモリ - メモリ間転送命令, ビット処理命令, パレルシフト命令など: 組込み用途に適した命令
- 関数入口 / 出口命令, レジスタ内容のマルチロードストア命令: C 言語対応命令
- レジスタのインターロック機能: アセンブラ記述も容易に可能
- 乗算器の内蔵 / 命令レベルでのサポート
符号付き 32 ビット乗算: 5 サイクル
符号付き 16 ビット乗算: 3 サイクル
- 割込み (PC/PS 退避) : 6 サイクル (16 プライオリティレベル)
- ハーバードアーキテクチャにより, プログラムアクセスとデータアクセスを同時に実行可能
- FR ファミリとの命令互換

● 内蔵周辺機能

表 1.1-1 内蔵周辺機能

	MB91V265A	MB91F267A	MB91F267NA	MB91267A	MB91267NA
	評価用品	フラッシュメモリ品		マスク ROM 品	
パッケージ	PGA-401 (リードピッチ 2.54 mm インタースティシャル)	LQFP-64 (リードピッチ 0.65 mm)			
ROM/Flash 容量	外部 SRAM	128 K バイト			
RAM 容量	24 K バイト	4 K バイト			
C-CAN	1 チャンネル	なし	1 チャンネル	なし	1 チャンネル

- A/D コンバータ (逐次比較型)
 - 分解能 : 10 ビット: 4 チャンネル × 1 ユニット, 7 チャンネル × 1 ユニット
 - 変換時間: 1.2 μs (最小変換時間 システムクロック 33 MHz 時)
1.35 μs (最小変換時間 システムクロック 20 MHz 時)
- 外部割込み入力: 8 チャンネル

- ビットサーチモジュール (REALOS 使用)
 - 1 ワード中の MSB (上位ビット) から最初の "1" "0" 変化ビット位置をサーチする機能
- UART (全二重ダブルバッファ方式) : 2 チャンネル
 - パリティあり / なし選択可能
 - 非同期 (調歩同期) / クロック同期通信の選択可能
 - 専用ボーレートタイマ (U-Timer) をチャンネルごとに内蔵
 - 外部クロックを転送クロックとして使用可能
 - パリティ, フレーム, オーバランエラー検出機能あり
- 8/16 ビット PPG タイマ : 8 チャンネル (8 ビット時) / 4 チャンネル (16 ビット時)
- タイミングジェネレータ
- 16 ビットリロードタイマ : 3 チャンネル (カスケードモードあり, リロードタイマ 0 の出力なし)
- 16 ビットフリーランタイマ : 3 チャンネル
- 16 ビット PWC タイマ : 1 チャンネル
- インプットキャプチャ : 4 チャンネル (フリーランタイマと連動)
- アウトプットコンペア : 6 チャンネル (フリーランタイマと連動)
- 波形ジェネレータ

アウトプットコンペア出力, 16 ビット PPG タイマ 0, 16 ビットデッドタイマを使用してさまざまな波形を生成することができます。
- 積和演算回路
 - RAM : 命令 RAM(I-RAM) 256 × 16 ビット
 - XRAM 64 × 16 ビット
 - YRAM 64 × 16 ビット
 - 積和演算 (16 ビット × 16 ビット + 40 ビット) を 1 サイクル実行
 - 演算結果は 40 ビットから 16 ビットへの丸め処理で抽出
- DMAC (DMA コントローラ) : 5 チャンネル

内蔵ペリフェラル割込み, ソフトウェアによって転送起動が可能
- ウォッチドッグタイマ
- 低消費電力モード

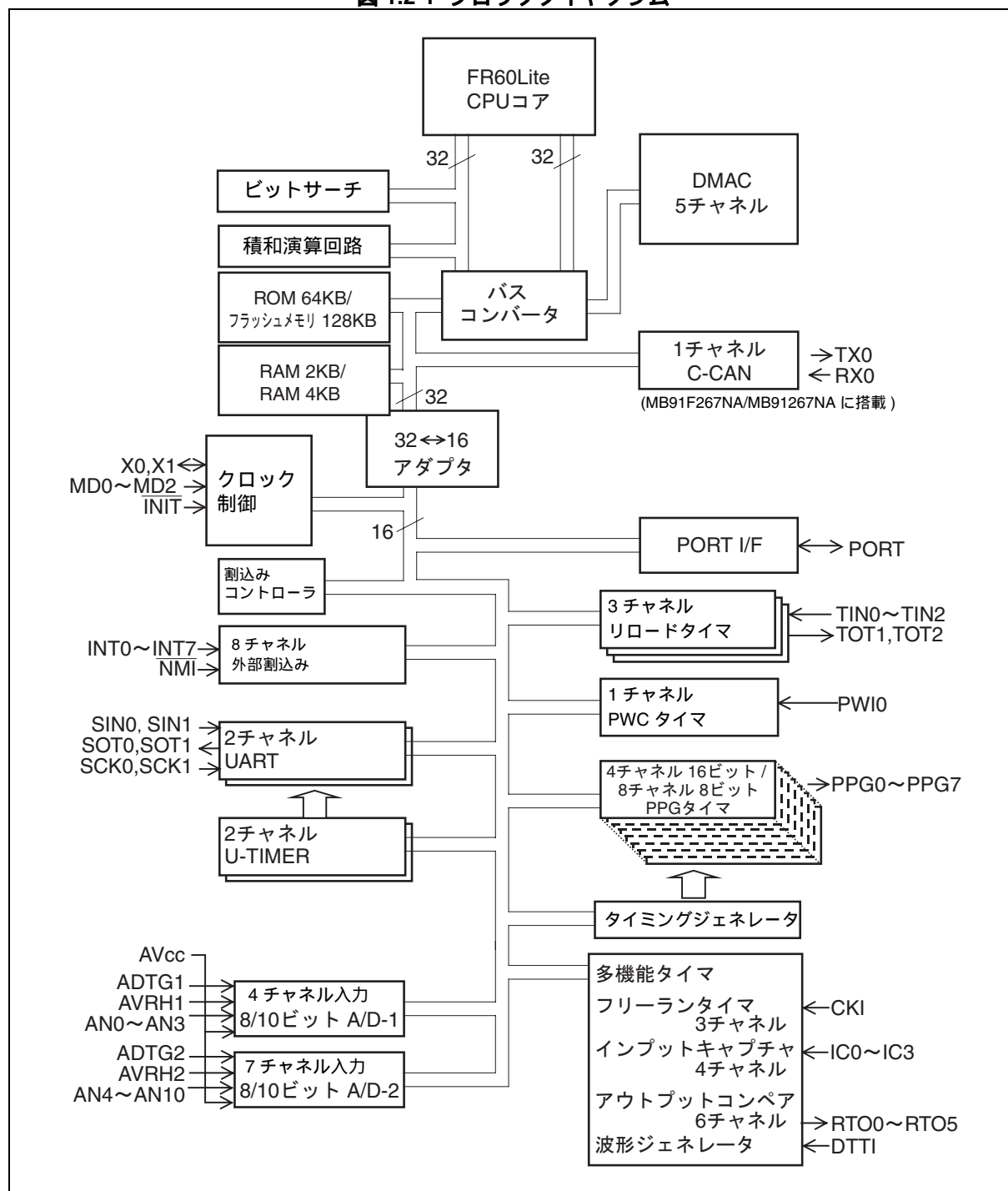
スリープ / ストップ機能
- パッケージ : LQFP-64
- CMOS テクノロジ : 0.35 μm
- 電源 : 1 電源 (Vcc=4.0V ~ 5.5V)

1.2 ブロックダイアグラム

図 1.2-1 に , MB91265A シリーズのブロックダイアグラムを示します。

■ ブロックダイアグラム

図 1.2-1 ブロックダイアグラム

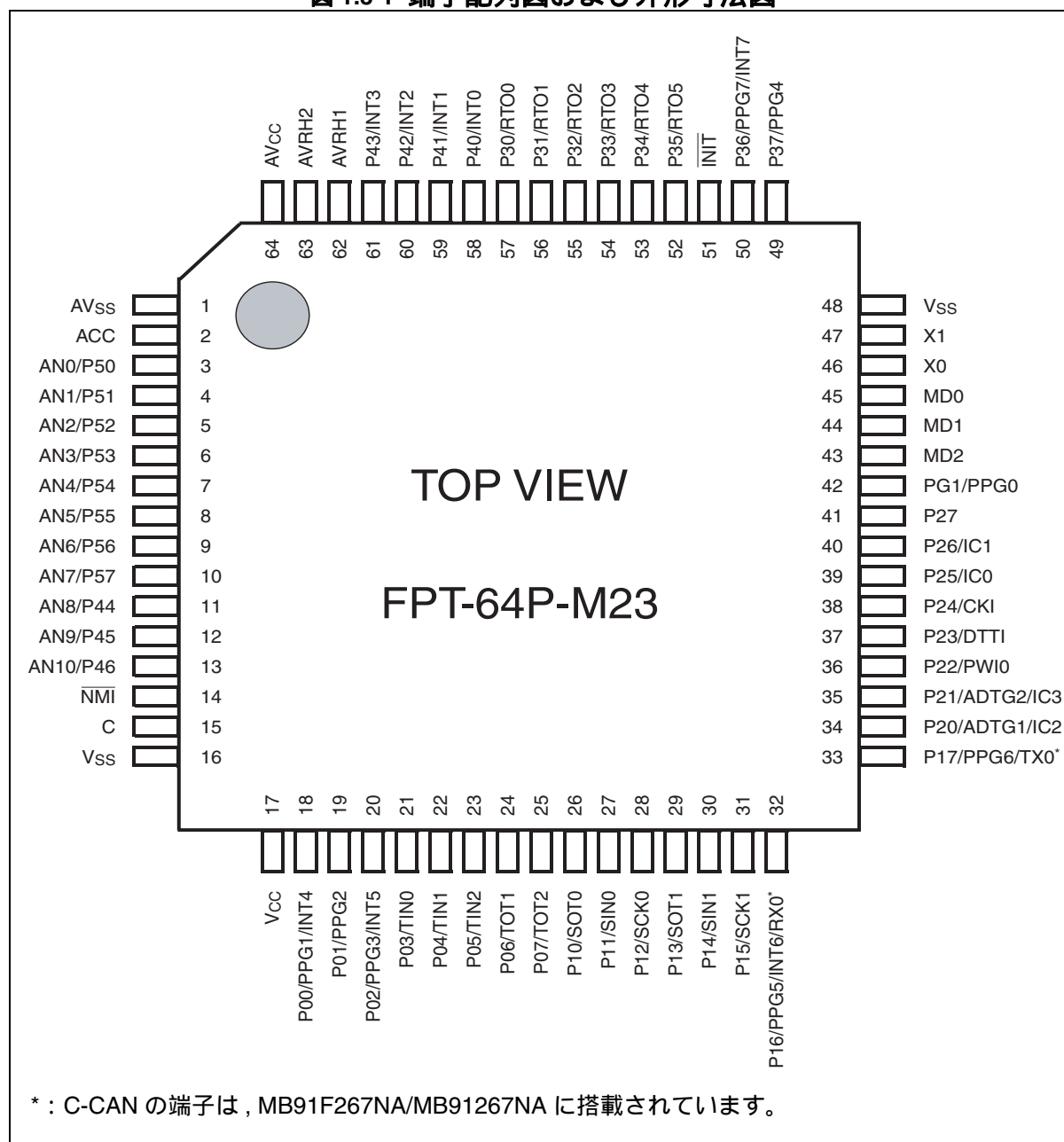


1.3 端子配列図

FPT-64P-M23 の端子配列図および外形寸法図を示します。

■ LQFP-64 (MB91F267A/MB91F267NA/MB91267A/MB91267NA)

図 1.3-1 端子配列図および外形寸法図

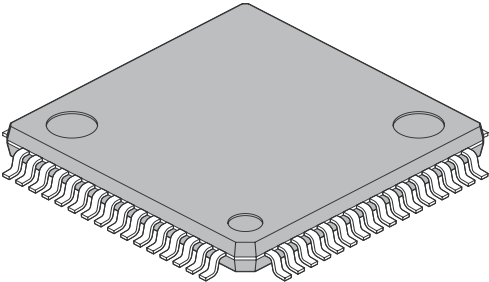


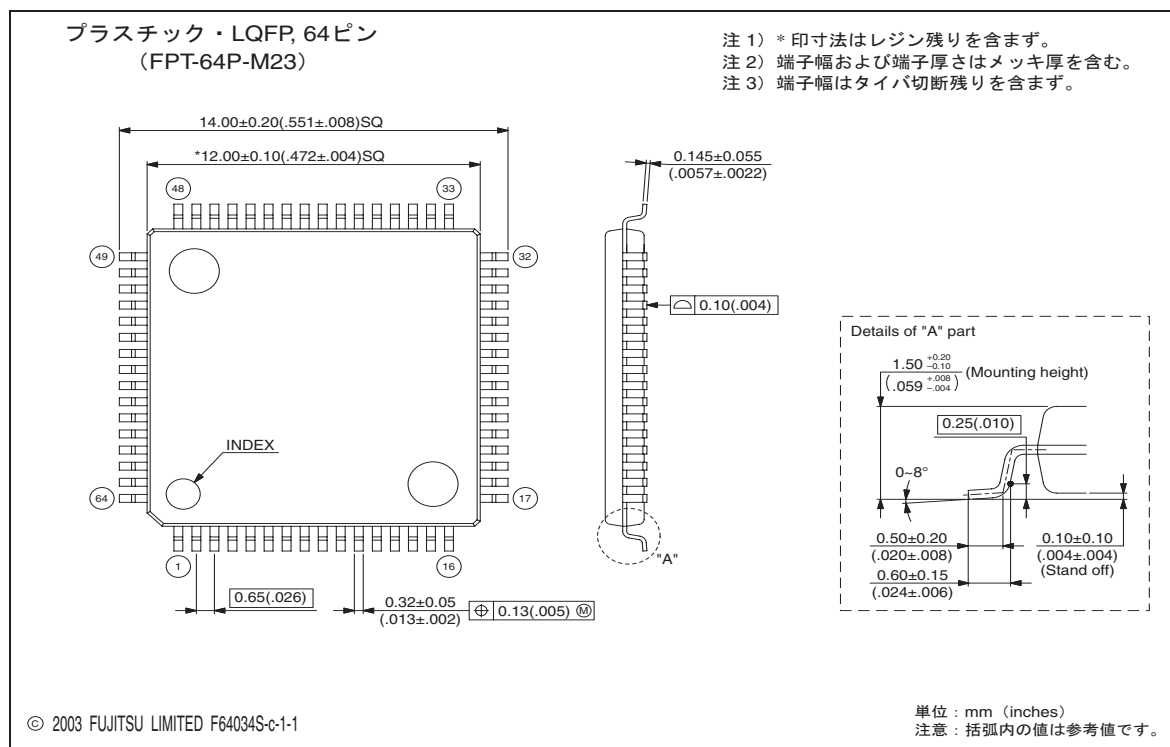
1.3.1 外形寸法図

図 1.3-2 に , FPT-64P-M23 の外形寸法図を示します。

■ パッケージ外形寸法図

図 1.3-2 外形寸法図

<p>プラスチック・LQFP, 64ピン</p>  <p>(FPT-64P-M23)</p>	リードピッチ	0.65mm
	パッケージ幅 × パッケージ長さ	12.0 × 12.0mm
	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
	コード (参考)	P-LFQFP64-12 × 12-0.65



最新の外形寸法図については , 下記の URL にてご確認ください。

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

1.4 端子機能一覧

表 1.4-1 に、端子機能について説明します。

■ 端子機能一覧

端子配列図を参照してください。

表 1.4-1 端子機能一覧表 (1 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
3	AN0	G	A/D コンバータ 1 のアナログ入力です。 AICR1 レジスタの指定がアナログ入力のときに有効になります。
	P50		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
4	AN1	G	A/D コンバータ 1 のアナログ入力です。 AICR1 レジスタの指定がアナログ入力のときに有効になります。
	P51		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
5	AN2	G	A/D コンバータ 1 のアナログ入力です。 AICR1 レジスタの指定がアナログ入力のときに有効になります。
	P52		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
6	AN3	G	A/D コンバータ 1 のアナログ入力です。 AICR1 レジスタの指定がアナログ入力のときに有効になります。
	P53		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
7	AN4	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P54		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
8	AN5	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P55		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
9	AN6	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P56		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
10	AN7	G	A/D コンバータの 2 アナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P57		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
11	AN8	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P44		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。

表 1.4-1 端子機能一覧表 (2 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
12	AN9	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P45		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
13	AN10	G	A/D コンバータ 2 のアナログ入力です。 AICR2 レジスタの指定がアナログ入力のときに有効になります。
	P46		汎用入出力ポートです。 この機能はアナログ入力指定が禁止のときに有効になります。
14	$\overline{\text{NMI}}$	H	$\overline{\text{NMI}}$ (Non Maskable Interrupt) 入力です。
18	INT4	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	PPG1		PPG タイマ 1 の出力です。 この機能は PPG タイマ 1 の出力指定が許可のときに有効になります。
	P00		汎用入出力ポートです。 PPG タイマ 1 の出力指定および外部割込み入力が禁止のときに有効になります。
19	PPG2	D	PPG タイマ 2 の出力です。 この機能は PPG タイマ 2 の出力指定が許可のときに有効になります。
	P01		汎用入出力ポートです。 PPG タイマ 2 の出力指定が禁止のときに有効になります。
20	INT5	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	PPG3		PPG タイマ 3 の出力です。 この機能は PPG タイマ 3 の出力指定が許可のときに有効になります。
	P02		汎用入出力ポートです。 PPG タイマ 3 の出力指定および外部割込み入力が禁止のときに有効になります。
21	TIN0	D	リロードタイマ 0 の外部トリガ入力です。 トリガ入力を許可している間は、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ポートによる出力を停止させておく必要があります。
	P03		汎用入出力ポートです。 この機能はリロードタイマ 0 の外部クロック入力指定が禁止のときに有効になります。
22	TIN1	D	リロードタイマ 1 の外部トリガ入力です。 トリガ入力を許可している間は、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ポートによる出力を停止させておく必要があります。
	P04		汎用入出力ポートです。 この機能はリロードタイマ 1 の外部クロック入力指定が禁止のときに有効になります。

表 1.4-1 端子機能一覧表 (3 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
23	TIN2	D	リロードタイマ 2 の外部トリガ入力です。 トリガ入力を許可している間は、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ポートによる出力を停止させておく必要があります。
	P05		汎用入出力ポートです。 この機能はリロードタイマ 2 の外部クロック入力指定が禁止のときに有効になります。
24	TOT1	D	リロードタイマ 1 の出力です。 この機能はリロードタイマ 1 出力許可のときに有効になります。
	P06		汎用入出力ポートです。 この機能はリロードタイマ 1 出力指定が禁止のときに有効になります。
25	TOT2	D	リロードタイマ 2 の出力です。 この機能はリロードタイマ 2 出力許可のときに有効になります。
	P07		汎用入出力ポートです。 この機能はリロードタイマ 2 出力指定が禁止のときに有効になります。
26	SOT0	D	UART0 のデータ出力です。 この機能は UART0 のデータ出力指定が許可のときに有効になります。
	P10		汎用入出力ポートです。 UART0 のデータ出力指定が禁止のときに有効になります。
27	SIN0	D	UART0 のデータ入力です。 UART0 が入力動作をしている間は、この入力を随時使用しますので、意図的に行う以外は、ポートによる出力を停止させておく必要があります。
	P11		汎用入出力ポートです。 UART0 のデータ入力指定が禁止のときに有効になります。
28	SCK0	D	UART0 のクロック入出力です。 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。
	P12		汎用入出力ポートです。 この機能は UART0 のクロック出力指定が禁止のときに有効になります。
29	SOT1	D	UART1 のデータ出力です。 この機能は UART1 のデータ出力指定が許可のときに有効になります。
	P13		汎用入出力ポートです。 この機能は UART1 のデータ出力指定が禁止のときに有効になります。
30	SIN1	D	UART1 のデータ入力です。 UART1 が入力動作をしている間は、この入力を随時使用しますので、意図的に行う以外は、ポートによる出力を停止させておく必要があります。
	P14		汎用入出力ポートです。 この機能は UART1 のデータ入力指定が禁止のときに有効になります。
31	SCK1	D	UART1 のクロック入出力です。 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。
	P15		汎用入出力ポートです。 この機能は UART1 のクロック出力指定が禁止のときに有効になります。

表 1.4-1 端子機能一覧表 (4 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
32	INT6	E	外部割込み入力です。 対応する外部割込みを許可している間は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	PPG5		PPG タイマ 5 の出力です。 この機能は PPG タイマ 5 の出力指定が許可のときに有効になります。
	RX0		C-CAN0 の RX0 入力端子です (MB91F267NA/MB91267NA)。 この入力は随時使用しますので、意図的に行う以外は、ポートによる出力を停止させておく必要があります。
	P16		汎用入出力ポートです。 この機能は PPG タイマ 5 の出力指定および C-CAN0 の RX0 入力指定 *2 が禁止のときに有効になります。
33	PPG6	D	PPG タイマ 6 の出力です。 この機能は PPG タイマ 6 の出力指定が許可のときに有効になります。
	TX0		C-CAN0 の TX0 出力端子です (MB91F267NA/MB91267NA)。 この機能は C-CAN0 の TX0 出力指定が許可のときに有効になります。
	P17		汎用入出力ポートです。 この機能は PPG タイマ 6 の出力指定および C-CAN0 の TX0 出力指定 *2 が禁止のときに有効になります。
34	ADTG1	D	A/D コンバータ 1 の外部トリガ入力です。 A/D の起動要因として選択した場合は、この入力を随時使用しますので、意図的に行う以外は、ポートによる出力を停止させておく必要があります。
	IC2		インプットキャプチャ 2 のトリガ入力です。 インプットキャプチャトリガ入力を設定し、ポートを入力設定した場合に入力できます。インプットキャプチャ入力として選択した場合は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	P20		汎用入出力ポートです。 この機能は A/D コンバータの外部トリガ入力指定およびインプットキャプチャトリガ入力指定が禁止のときに有効になります。
35	ADTG2	D	A/D コンバータ 2 の外部トリガ入力です。 A/D の起動要因として選択した場合は、この入力を随時使用しますので、意図的に行う以外は、ポートによる出力を停止させておく必要があります。
	IC3		インプットキャプチャ 3 のトリガ入力です。 インプットキャプチャトリガ入力を設定し、ポートを入力設定した場合に入力できます。インプットキャプチャ入力として選択した場合は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	P21		汎用入出力ポートです。 この機能は A/D コンバータの外部トリガ入力指定およびインプットキャプチャトリガ入力指定が禁止のときに有効になります。

表 1.4-1 端子機能一覧表 (5 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
36	PWIO	D	PWC タイマ 0 のパルス幅カウンタ入力です。 この機能は PWC タイマ 0 のパルス幅カウンタ入力許可のときに有効になります。
	P22		汎用入出力ポートです。 この機能は PWC タイマ 0 のパルス幅カウンタ入力指定が禁止のときに有効になります。
37	DTTI	D	多機能タイマの波形ジェネレータ出力 RTO0 ~ RTO5 を制御する入力信号です。 この機能は DTTI 端子入力が許可のときに有効になります。
	P23		汎用入出力ポートです。 この機能は DTTI 端子入力指定が禁止のときに有効になります。
38	CKI	D	フリーランタイムの外部クロック入力端子です。 フリーランタイムの外部クロック入力として選択した場合は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	P24		汎用入出力ポートです。 フリーランタイムの外部クロック入力指定が禁止のときに有効になります。
39	IC0	D	インプットキャプチャ 0 のトリガ入力です。 インプットキャプチャ 0 のトリガ入力を設定し、ポートを入力設定した場合に入力できます。インプットキャプチャ入力として選択した場合は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	P25		汎用入出力ポートです。 この機能はインプットキャプチャ 0 のトリガ入力指定が禁止のときに有効になります。
40	IC1	D	インプットキャプチャ 1 のトリガ入力です。 インプットキャプチャ 1 のトリガ入力を設定し、ポートを入力設定した場合に入力できます。インプットキャプチャ入力として選択した場合は、この入力を随時使用しますので意図的に行う以外はポートによる出力を停止させておく必要があります。
	P26		汎用入出力ポートです。 この機能はインプットキャプチャ 1 のトリガ入力指定が禁止のときに有効になります。
41	P27	D	汎用入出力ポートです。
42	PPG0	D	PPG タイマ 0 の出力です。 この機能は PPG タイマ 0 の出力指定が許可のときに有効になります。
	PG1		汎用入出力ポートです。 この機能は、PPG タイマ 0 の出力指定が禁止のときに有効になります。
43	MD2	H, K	モード端子 2 です。 この端子の設定により基本動作モードを設定します。Vcc または Vss に接続してください。フラッシュメモリ品は回路形式は K です。
44	MD1	H, K	モード端子 1 です。 この端子の設定により基本動作モードを設定します。Vcc または Vss に接続してください。フラッシュメモリ品は回路形式は K です。

表 1.4-1 端子機能一覧表 (6 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
45	MD0	H	モード端子 0 です。 この端子の設定により基本動作モードを設定します。Vcc または Vss に接続してください。
46	X0	A	クロック (発振) 出力です。
47	X1	A	クロック (発振) 入力です。
49	PPG4	D	PPG タイマ 4 の出力です。 この機能は PPG タイマ 4 の出力指定が許可のときに有効になります。
	P37		汎用入出力ポートです。 この機能は、PPG タイマ 4 の出力指定が禁止のときに有効になります。
50	INT7	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	PPG7		PPG タイマ 7 の出力です。 この機能は PPG タイマ 7 の出力指定が許可のときに有効になります。
	P36		汎用入出力ポートです。 この機能は、PPG タイマ 7 の出力指定が禁止のときに有効になります。
51	$\overline{\text{INIT}}$	I	外部リセット入力です。
52	RTO5	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P35		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。
53	RTO4	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P34		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。
54	RTO3	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P33		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。
55	RTO2	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P32		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。
56	RTO1	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P31		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。

表 1.4-1 端子機能一覧表 (7 / 7)

端子番号	端子名	入出力回路形式 *1	機能説明
57	RTO0	J	多機能タイマの波形ジェネレータ出力です。 この端子は、波形ジェネレータに指定した波形を出力します。波形の出力は、波形ジェネレータ出力が許可のときに有効になります。
	P30		汎用入出力ポートです。 この機能は波形ジェネレータ出力指定が禁止のときに有効になります。
58	INT0	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	P40		汎用入出力ポートです。 この機能は外部割込み入力指定が禁止のときに有効になります。
59	INT1	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	P41		汎用入出力ポートです。 この機能は外部割込み入力指定が禁止のときに有効になります。
60	INT2	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	P42		汎用入出力ポートです。 この機能は外部割込み入力指定が禁止のときに有効になります。
61	INT3	E	外部割込み入力です。 対応する外部割込みを許可している間は、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ポートによる出力を停止させておく必要があります。
	P43		汎用入出力ポートです。 この機能は外部割込み入力指定が禁止のときに有効になります。

*1：入出力回路形式については、「1.5 入出力回路形式」を参照してください。

*2：C-CAN の設定は、MB91F267NA/MB91267NA の設定です。

第 1 章 概要

[電源・GND 端子]

端子番号	端子名	機能説明
16, 48	Vss	GND 端子です。すべて同電位でお使いください。
17	Vcc	電源端子です。すべて同電位でお使いください。
64	AVcc	A/D コンバータ用のアナログ電源端子です。
63	AVRH2	A/D コンバータ 2 用のアナログ基準電源端子です。
62	AVRH1	A/D コンバータ 1 用のアナログ基準電源端子です。
1	AVss	A/D コンバータ用のアナログ GND 端子です。
15	C	内部レギュレータ用のコンデンサ結合端子です。
2	ACC	アナログ用のコンデンサ結合端子です。

1.5 入出力回路形式

表 1.5-1 に，入出力回路形式を示します。

■ 入出力回路形式

表 1.5-1 入出力回路形式 (1 / 3)

分類	回路形式	備考
A	<p>クロック入力</p> <p>スタンバイ制御</p>	<p>高速用 (メインクロック原発振)</p> <p>発振帰還抵抗 : 約 1 MΩ</p>
D	<p>プルアップ制御</p> <p>デジタル出力</p> <p>デジタル出力</p> <p>スタンバイ制御</p> <p>デジタル入力</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 <p>スタンバイ制御あり</p> <p>プルアップ制御あり</p> <p>プルアップ抵抗値 = 約 50 kΩ (Typ)</p> <p>$I_{OL} = 4 \text{ mA}$</p>
E	<p>プルアップ制御</p> <p>デジタル出力</p> <p>デジタル出力</p> <p>スタンバイ制御</p> <p>デジタル入力</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 <p>スタンバイ制御なし</p> <p>プルアップ制御あり</p> <p>プルアップ抵抗値 = 約 50 kΩ (Typ)</p> <p>$I_{OL} = 4 \text{ mA}$</p>

表 1.5-1 入出力回路形式 (2 / 3)

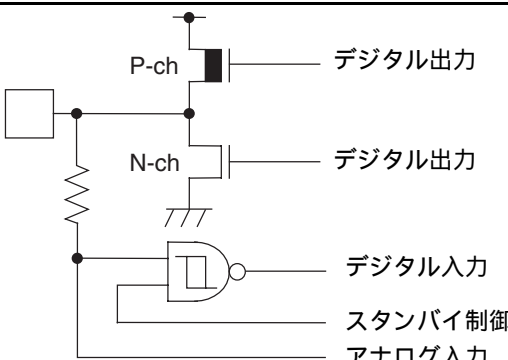
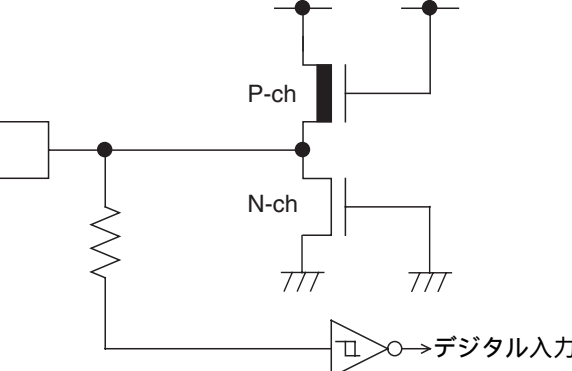
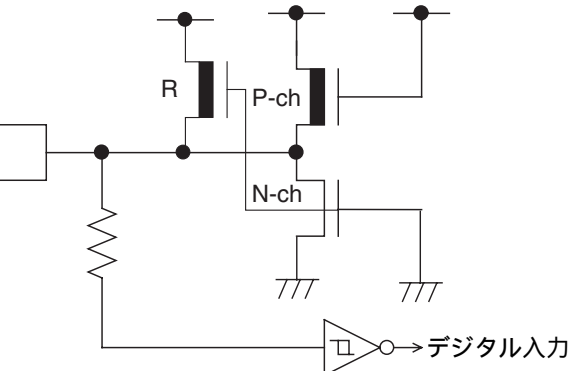
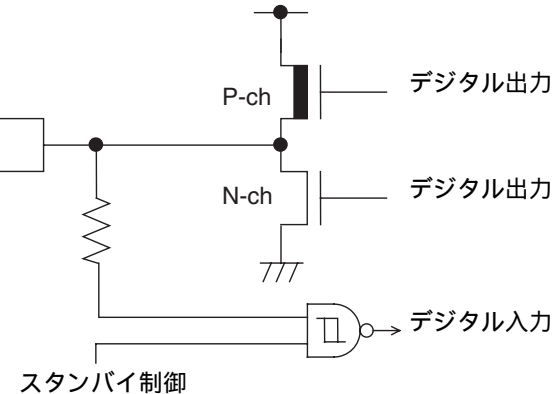
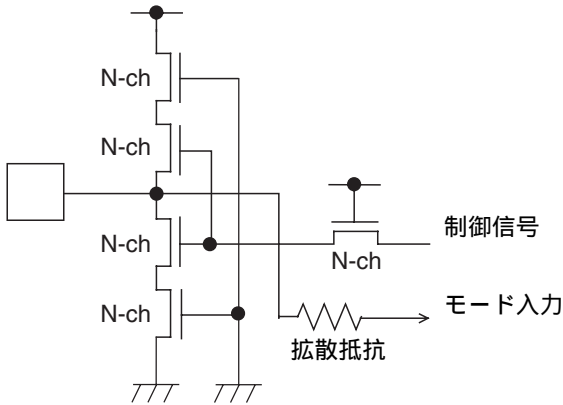
分類	回路形式	備考
G	 <p>デジタル出力</p> <p>デジタル出力</p> <p>デジタル入力</p> <p>スタンバイ制御</p> <p>アナログ入力</p>	<p>アナログ /CMOS レベルヒステリシス入出力端子</p> <p>CMOS レベル出力</p> <p>CMOS レベルヒステリシス入力 (スタンバイ制御付き)</p> <p>アナログ入力 (AICR の対応するビットが "1" のときアナログ入力が有効になります)</p> <p>$I_{OL}=4\text{ mA}$</p>
H	 <p>デジタル入力</p>	<p>CMOS レベルヒステリシス入力</p> <p>スタンバイ制御なし</p>
I	 <p>デジタル入力</p>	<p>CMOS レベルヒステリシス入力</p> <p>プルアップ抵抗付き</p> <p>プルアップ抵抗値 = 約 $50\text{ k}\Omega$ (Typ)</p> <p>スタンバイ制御なし</p>
J	 <p>デジタル出力</p> <p>デジタル出力</p> <p>デジタル入力</p> <p>スタンバイ制御</p>	<ul style="list-style-type: none"> CMOS レベル出力 CMOS レベルヒステリシス入力 <p>スタンバイ制御あり</p> <p>$I_{OL} = 12\text{ mA}$</p>

表 1.5-1 入出力回路形式 (3 / 3)

分類	回路形式	備考
K	 <p>The diagram shows four N-channel MOSFETs connected in a stack. The gates of the top three transistors are connected to a common control signal line. The gate of the bottom transistor is connected to a mode input line through a diffusion resistance (represented by a zigzag line). The source of the bottom transistor is connected to ground. The drain of the top transistor is connected to a supply voltage. The drains of the middle two transistors are connected to a common node that is also connected to the mode input line. The source of the second transistor from the top is connected to the drain of the third transistor. The source of the third transistor is connected to the drain of the bottom transistor. The source of the bottom transistor is connected to ground.</p>	フラッシュメモリ品のみ • CMOS レベル入力 • フラッシュテスト用の高電圧制御あり

第2章

デバイスの取扱いについて

デバイスを取り扱う際の注意事項について説明します。

2.1 デバイスの取扱いについて

2.1 デバイスの取扱いについて

デバイスを扱う際の注意事項について説明します。

■ デバイスの取扱いについて

- ラッチアップ防止のために

CMOSIC では入力端子や出力端子に V_{CC} より高い電圧や V_{SS} より低い電圧を印加した場合、または V_{CC} と V_{SS} の間に定格を超える電圧を印加した場合に、ラッチアップ現象を生じることがあります。ラッチアップが生じると電源電流が激増し、素子の熱破壊に至ることがありますので使用に際しては最大定格を超えることのないよう十分に注意してください。

- 未使用入力端子の処理について

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の原因となることがありますので、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。

- 電源端子について

V_{CC} または V_{SS} が複数ある場合、デバイス設計上はラッチアップなどの誤動作を防止するためにデバイス内部で同電位にすべきものどうし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ブ信号の誤動作の防止・総出力電流規格を遵守などのために、必ずそれらすべてを外部で同電位の電源およびグラウンドに接続してください。また、電流供給源からできるかぎり低インピーダンスで本デバイスの V_{CC} または V_{SS} に接続するような配慮をお願い致します。

さらに、本デバイスの近くで、 V_{CC} と V_{SS} の間に $0.1\ \mu\text{F}$ 程度のセラミックコンデンサをバイパスコンデンサとして接続することをお勧め致します。

- 水晶発振回路について

X0, X1 端子の近辺のノイズは本デバイスの誤動作の原因となります。X0 と X1 および水晶発振子（あるいはセラミック発振子）さらにグラウンドへのバイパスコンデンサはできるかぎり近くに配置するようにプリント板を設計してください。

また、X0, X1 端子の回りをグラウンドで囲むようなプリント板アートワークは安定した動作を期待できますので、強くお勧め致します。

各量産品において、ご使用される発振子メーカーに発振評価依頼をしてください。

- モード端子 (MD0 ~ MD2) について

これらの端子は、 V_{CC} または V_{SS} に直接つないで使用してください。ノイズにより誤ってテストモードに入ってしまうことを防ぐために、プリント板上の各モード端子と V_{CC} または V_{SS} 間のパターン長をできるかぎり短くし、これらを低インピーダンスで接続するようにしてください。

- 電源投入時について

電源投入直後は必ず $\overline{\text{INIT}}$ 端子にて設定初期化リセット (INIT) をかけてください。

また、電源投入直後は、発振回路の発振安定待ち時間およびレギュレータの安定待ち時間を確保するため、 $\overline{\text{INIT}}$ 端子への "L" レベル入力を発振回路の要求する安定待ち時間の間持続してください ($\overline{\text{INIT}}$ 端子による INIT では、発振安定待ち時間の設定は最小値に初期化されています)。

● 電源投入順序について

電源投入は、 V_{cc} AV_{cc} $AVRH$ の順に投入し、切断時は、この逆の順で行ってください。

A/D コンバータを使用しない場合でも、 $AV_{cc}=V_{cc}$ レベル、 $AV_{ss}=V_{ss}$ レベルに接続してください。

● 電源投入時の原振入力について

電源投入時は、必ず発振安定待ち解除されるまでの間クロックを入力してください。

● PLL クロックモード動作中の注意について

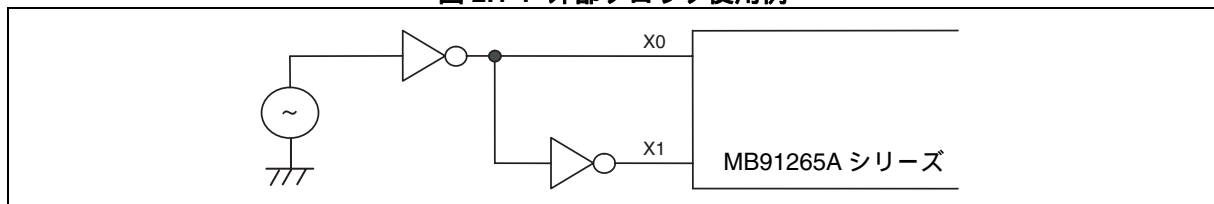
本デバイスで、PLL クロックを選択しているときに発振子が外れたり、あるいはクロック入力が停止した場合、本デバイスは PLL 内部の自励発振回路の自走周波数で動作を継続し続ける場合があります。この動作は保証外の動作です。

● 外部クロックについて

外部クロックを使用する際には、原則として X0 端子、また X1 端子には X0 と逆相のクロックを同時に供給してください。ただし、ストップモード（発振停止モード）を併用する場合は、ストップモード時に X1 端子が "H" 出力で停止しますので、出力どうしの衝突を避けるために、外部に $1\text{ k}\Omega$ 程度の抵抗を入れるようにしてください。

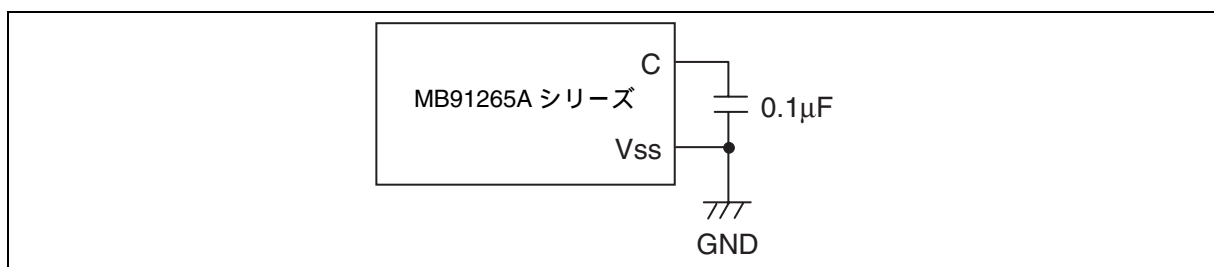
下図に外部クロック使用方法例について示します。

図 2.1-1 外部クロック使用例



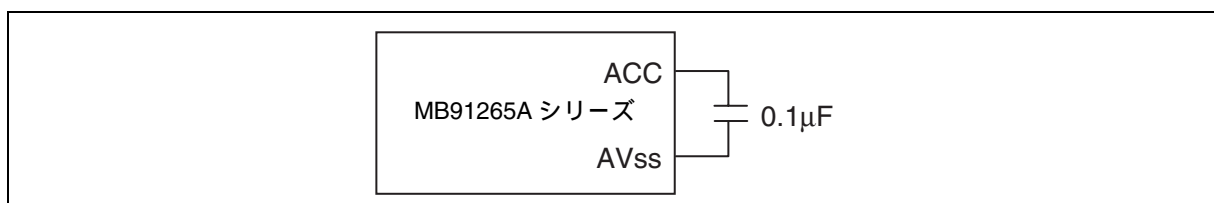
● C 端子について

本品種は、レギュレータを内蔵しており、C 端子には、レギュレータ用に $0.1\text{ }\mu\text{F}$ 程度のパスコンを必ず入れてください。



● ACC 端子について

本品種は、A/D コンバータを内蔵しており、ACC 端子と AV_{ss} 端子の端子間には、 $0.1\text{ }\mu\text{F}$ 程度のコンデンサを必ず入れてください。



第2章 デバイスの取扱いについて

● 同期モードのソフトウェアリセットについて

同期モードのソフトウェアリセットを使用するときは、STCR (スタンバイ制御レジスタ) の SRST ビットに "0" を設定する前に、以下2つの条件を必ず満たしてください。

- 割込み許可フラグ (I-Flag) を割込み禁止 (I-Flag=0) に設定する。
- NMI を使用しない。

第3章

CPU および制御部

MB91265A シリーズの CPU コアの機能を知るために、アーキテクチャ、仕様、命令などの基本的な
ことについて説明します。

- 3.1 メモリ空間
- 3.2 メモリマップ
- 3.3 内部アーキテクチャ
- 3.4 プログラミングモデル
- 3.5 データ構造
- 3.6 メモリマップ
- 3.7 分岐命令
- 3.8 EIT (例外・割込み・トラップ)
- 3.9 動作モード
- 3.10 リセット (デバイス初期化)
- 3.11 クロック生成制御
- 3.12 デバイス状態制御

3.1 メモリ空間

MB91265A シリーズの論理アドレス空間は 4G バイト (2^{32} 番地) あり, CPU はリニアにアクセスを行います。

■ ダイレクトアドレッシング領域

アドレス空間の下記の領域は I/O 用に使用されます。

この領域をダイレクトアドレッシング領域とよび, 命令中で直接オペランドのアドレスを指定できます。

ダイレクト領域は, アクセスするデータのサイズにより, 以下のように異なります。

バイトデータアクセス : 000_H ~ 0FF_H

ハーフワードデータアクセス : 000_H ~ 1FF_H

ワードデータアクセス : 000_H ~ 3FF_H

3.2 メモリマップ

MB91265A シリーズのメモリマップを示します。

■ MB91F267A/MB91F267NA/MB91267NA/MB91267A のメモリマップ

図 3.2-1 MB91F267A/MB91F267NA/MB91267NA/MB91267A のメモリマップ

シングルチップモード		
0000 0000 _H	I/O	ダイレクト アドレッシング領域
0000 0400 _H		
0001 0000 _H	I/O	I/O マップ参照
0003 F000 _H	アクセス禁止	
0004 0000 _H	内蔵 RAM4K バイト	
	アクセス禁止	
000E 0000 _H	内蔵 ROM128K バイト	
0010 0000 _H	アクセス禁止	
FFFF FFFF _H		

3.3 内部アーキテクチャ

MB91265A シリーズ CPU は、RISC アーキテクチャを採用すると同時に、組込み用途に向けた高機能命令を導入した高性能コアです。

■ 特長

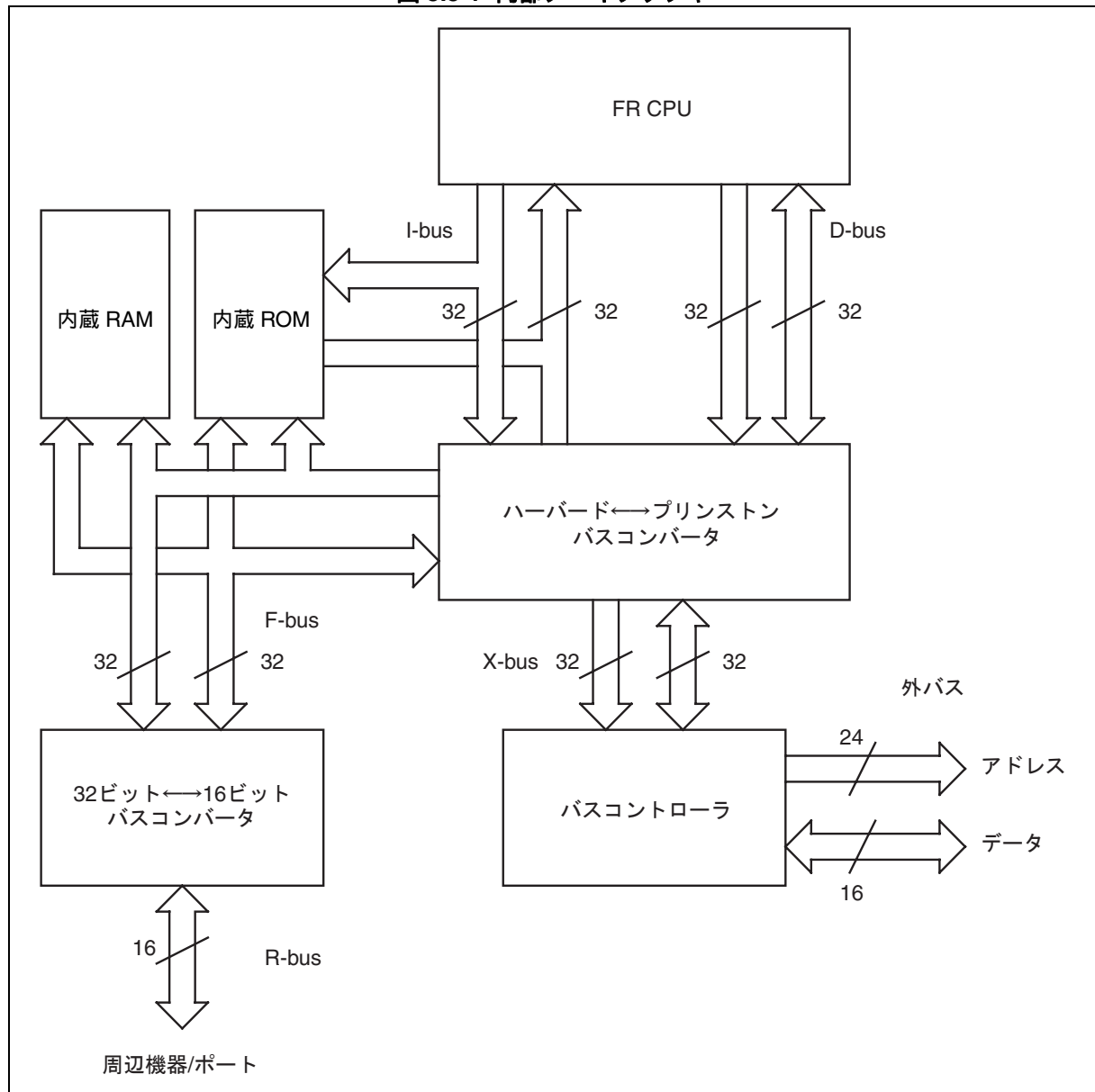
- RISC アーキテクチャの採用
基本命令 : 1 命令 1 サイクル
- 32 ビットアーキテクチャ
汎用レジスタ 32 ビット × 16 本
- 4G バイトのリニアなメモリ空間
- 乗算器の搭載
32 ビット × 32 ビット乗算 : 5 サイクル
16 ビット × 16 ビット乗算 : 3 サイクル
- 割込み処理機能の強化
高速応答速度 (6 サイクル)
多重割込みのサポート
レベルマスク機能 (16 レベル)
- I/O 操作用命令の強化
メモリ - メモリ転送命令
ビット処理命令
- 高いコード効率
基本命令語長 : 16 ビット
- 低消費電力
スリープモード、ストップモード
- クロック分周比設定機能

■ 内部アーキテクチャ

FR ファミリの CPU は命令バスとデータバスが独立したハーバードアーキテクチャ構造を採用しています。

32 ビット 16 ビットバスコンバータは 32 ビットバス (F-bus) に接続され、CPU と周辺リソースとのインタフェースを実現します。ハーバード プリンストンバスコンバータは I-bus、D-bus 双方に接続され、CPU とバスコントローラとのインタフェースを実現します。

図 3.3-1 内部アーキテクチャ



(注意事項) 外バス機能は、サポートしておりません。

■ CPU

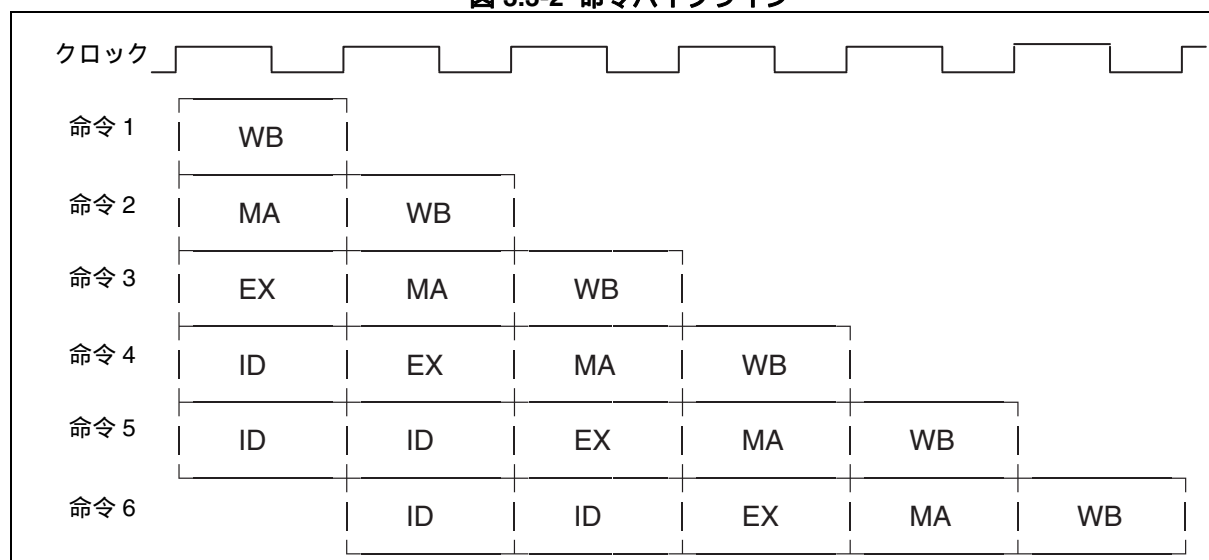
CPU は 32 ビット RISC の FR アーキテクチャをコンパクトにインプリメントしたものです。

1 サイクルあたり 1 命令の実行を行うため、5 段階の命令パイプライン方式を採用しています。

パイプラインは以下のステージから構成されています。

- 命令フェッチ (IF) : 命令アドレスを出力し、命令をフェッチします。
- 命令デコード (ID) : フェッチした命令をデコードします。レジスタの読出しも行います。
- 実行 (EX) : 演算を実行します。
- メモリアクセス (MA) : メモリに対するロードまたはストアのアクセスを行います。
- ライトバック (WB) : 演算結果 (またはロードされたメモリデータ) をレジスタに書き込みます。

図 3.3-2 命令パイプライン



命令は、順不同で実行されることはありません。すなわち、命令 A が命令 B の前にパイプラインに入ると、命令 A は必ず命令 B の前にライトバックステージに達します。

命令の実行は、原則として 1 サイクルあたり 1 命令の速度で行われます。ただし、メモリウェイトを伴ったロード・ストア命令、遅延スロットを持たない分岐命令、複数サイクル命令では命令の実行に複数のサイクルが必要となります。また、命令の供給が遅い場合も命令の実行速度が低下します。

■ 32 ビット 16 ビットバスコンバータ

32 ビット幅で高速アクセスされる F-bus と、16 ビット幅でアクセスされる R-bus とのインタフェースを行い、CPU から内蔵周辺回路へのデータアクセスを実現します。

CPU から 32 ビット幅のアクセスがあった場合、このバスコンバータがそれを 2 回の 16 ビット幅アクセスに変換して R-bus へのアクセスを行います。内蔵周辺回路の一部にはアクセス幅に関して制限のあるものがあります。

■ ハーバード プリンストンバスコンバータ

CPU の命令アクセスとデータアクセスの整合をとり、外部バスとのスムーズなインタフェースを実現します。

CPU は命令バスとデータバスが独立したハーバードアーキテクチャ構造です。一方、外部バスの制御を行うバスコントローラは単一バスのプリンストンアーキテクチャ構造です。このバスコンバータは CPU の命令アクセスとデータアクセスに優先順位を付け、バスコントローラへのアクセスの制御を行います。この働きにより、外部へのバスアクセス順位が常に最適化されます。

■ 命令概要

FR ファミリは、一般的な RISC の命令体系に加え、組み込み用途に最適化された論理演算とビット操作およびダイレクトアドレッシング命令をサポートしています。命令セットの一覧は付録を参照してください。各命令は 16 ビット長（一部命令は 32 ビット長、48 ビット長）ですので、優れたメモリ使用効率を持ちます。

命令セットは以下の機能グループに分けることができます。

- 算術演算
- ロードとストア
- 分岐
- 論理演算とビット操作
- ダイレクトアドレッシング
- その他

● 算術演算

標準の算術演算命令（加算、減算、比較）およびシフト命令（論理シフト、算術演算シフト）があります。加算と減算については、多ワード長演算で使用するキャリ付き演算や、アドレス計算に便利なフラグ値を変化させない演算も可能です。

さらに、32 ビット × 32 ビット、16 ビット × 16 ビットの乗算命令と、32 ビット ÷ 32 ビットのステップ除算命令を持ちます。

また、レジスタに即値をセットする即値転送命令や、レジスタ間転送命令も備えています。

算術演算命令はすべて CPU 内の汎用レジスタおよび乗除算レジスタを用いて演算を行います。

● ロードとストア

ロードとストアは外部メモリに対して読出しと書込みを行う命令です。また、チップ内の周辺リソース (I/O) への読出しと書込みにも使用されます。

ロードとストアはバイト、ハーフワード、ワードの 3 種類のアクセス長を持ちます。また、一般的なレジスタ間接のメモリアドレッシングに加え、一部の命令についてはディスプレースメント付きレジスタ間接やレジスタインクリメント・デクリメント付きレジスタ間接のメモリアドレッシングも可能です。

● 分岐

分岐、コール、割込みおよび復帰の命令です。分岐命令は、遅延スロットを持つものと持たないものがあり、用途に応じて最適化を行うことができます。分岐命令の詳細については「3.7 分岐命令」を参照してください。

- 論理演算とビット操作

論理演算命令は汎用レジスタ間、または汎用レジスタとメモリ（および I/O）間で AND, OR, EOR の論理演算を行えます。また、ビット操作命令はメモリ（および I/O）の内容を直接操作することができます。メモリアドレッシングは一般的なレジスタ間接です。

- ダイレクトアドレッシング

ダイレクトアドレッシング命令は I/O と汎用レジスタ間、または I/O とメモリ間のアクセスに使用する命令です。I/O のアドレスをレジスタ間接ではなく命令中で直接指定することにより、高速、高効率なアクセスが行えます。一部の命令についてはレジスタインクリメント・デクリメント付きレジスタ間接のメモリアドレッシングも可能です。

- その他

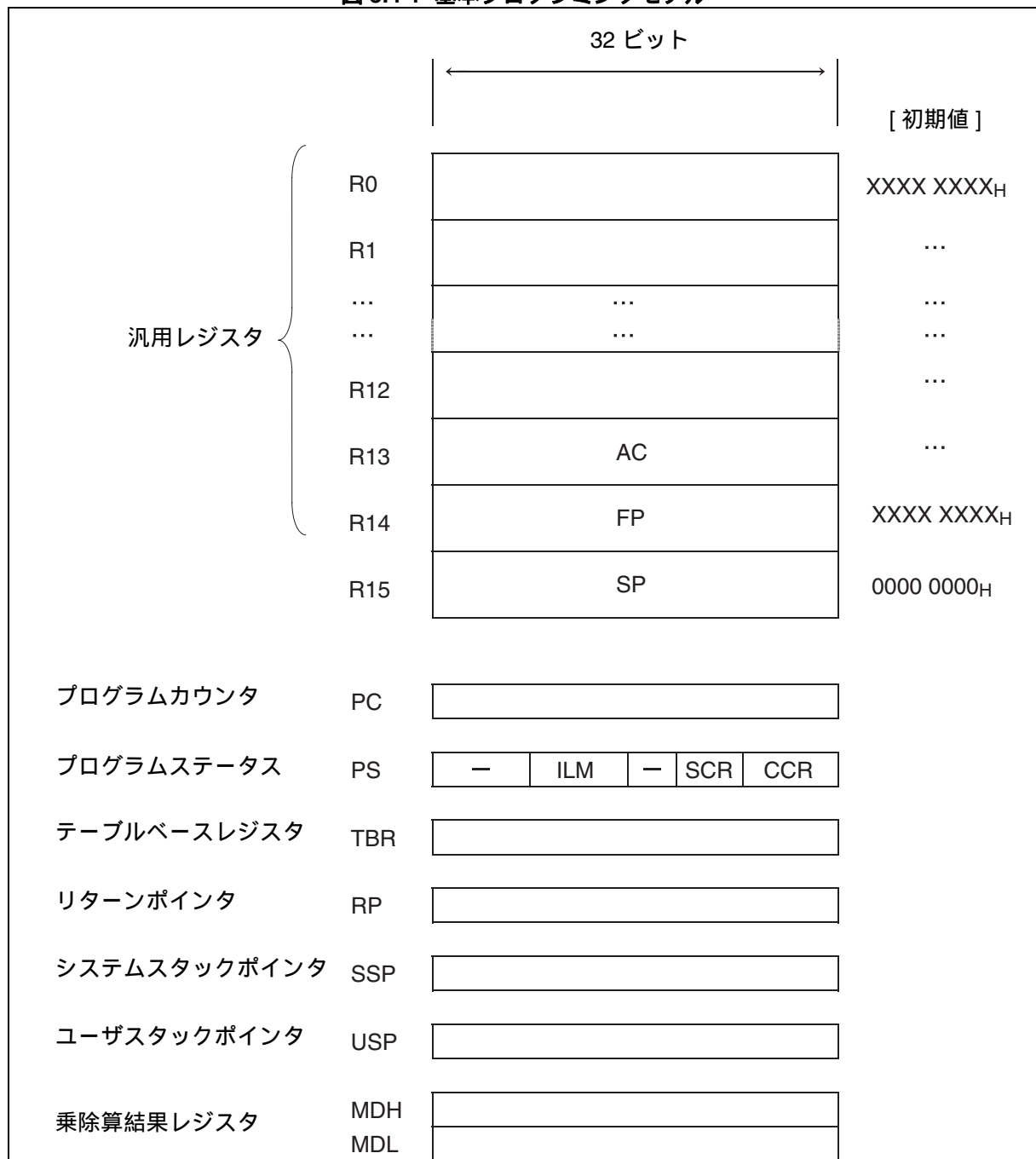
PS レジスタ内のフラグ設定、スタック操作、符号 / ゼロ拡張などを行う命令です。また、高級言語対応の関数入口 / 出口、レジスタマルチロード / ストア命令も備えています。

3.4 プログラミングモデル

基本プログラミングモデルと各レジスタについて説明します。

■ 基本プログラミングモデル

図 3.4-1 基本プログラミングモデル

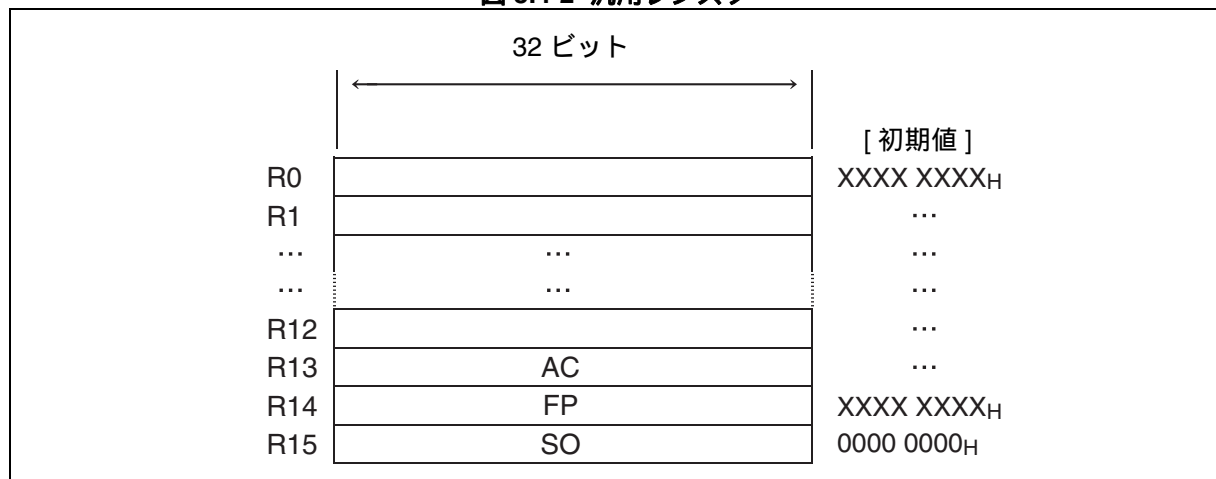


3.4.1 レジスタ

各レジスタについて説明します。

■ 汎用レジスタ

図 3.4-2 汎用レジスタ



レジスタ R0 ~ R15 は汎用レジスタです。各種演算におけるアキュムレータおよびメモリアクセスのポインタとして使用されます。

16本のレジスタのうち、以下に示すレジスタは特殊な用途を想定しており、そのために一部の命令が強化されています。

R13：仮想アキュムレータ

R14：フレームポインタ

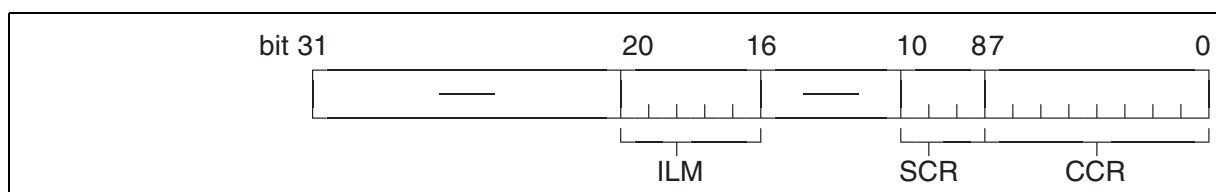
R15：スタックポインタ

リセットによる初期値は、R0 ~ R14 は不定です。R15 は、00000000_H (SSP の値) となります。

■ PS (Program Status)

プログラムステータスを保持するレジスタで、ILM, SCR と CCR の3つのパートに分かれています。

図中、未定義のビットはすべて予約ビットです。読出し時、常に "0" が読み出されます。書込みは無効です。



■ CCR (Condition Code Register)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	[初期値]
-	-	S	I	N	Z	V	C	--00XXXX _B

[bit5] S: スタックフラグ

R15 として使用されるスタックポインタを指定します。

値	内容
0	SSP が R15 として使用されます。 EIT 発生時, 自動的に "0" となります。 (ただし, スタックに退避される値はクリアされる前の値です。)
1	USP が R15 として使用されます。

リセットにより "0" にクリアされます。

RETI 命令実行時は "0" にしてください。

[bit4] I: 割込み許可フラグ

ユーザ割込み要求の許可・禁止を制御します。

値	内容
0	ユーザ割込み禁止です。 INT 命令実行時, "0" にクリアされます。 (ただし, スタック退避させる値はクリアする前の値です。)
1	ユーザ割込み許可です。 ユーザ割込み要求のマスク処理は, ILM の保持する値により制御されます。

リセットにより "0" にクリアされます。

[bit3] N: ネガティブフラグ

演算結果を "2" の補数で表現された整数とみなしたときの符号を示します。

値	内容
0	演算結果が正の値であったことを示します。
1	演算結果が負の値であったことを示します。

リセットによる初期状態は不定です。

[bit2] Z: ゼロフラグ

演算結果が "0" であったかどうかを示します。

値	内容
0	演算結果が "0" 以外の値であったことを示します。
1	演算結果が "0" であったことを示します。

リセットによる初期状態は不定です。

[bit1] V: オーバフローフラグ

演算に用いたオペランドを "2" の補数で表現される整数であるとみなし、演算の結果、オーバーフローが生じたかどうかを示します。

値	内容
0	演算の結果、オーバーフローは生じていないことを示します。
1	演算の結果、オーバーフローが生じたことを示します。

リセットによる初期状態は不定です。

[bit0] C: キャリフラグ

演算により、最上位ビットからのキャリ、またはボローが発生したかどうかを示します。

値	内容
0	キャリもボローも発生していないことを示します。
1	キャリまたはボローが発生したことを示します。

リセットによる初期状態は不定です。

■ SCR (System Condition Code Register)

bit10	bit9	bit8	[初期値]
D1	D0	T	XX0 _B

[bit10, bit9] D1, D0: ステップ除算用フラグ

ステップ除算実行時の中間データを保持します。

除算処理の実行途中に、変更しないでください。

ステップ除算実行途中にほかの処理を行う場合は、PS レジスタの値を退避・復帰することによりステップ除算の再開が保証されます。

リセットによる初期状態は不定です。

DIV0S 命令の実行により、被除数と除数を参照して設定されます。

DIV0U 命令の実行により、強制的にクリアされます。

[bit8] T: ステップトレーストラップフラグ

ステップトレーストラップを有効にするかどうかを指定するフラグです。

値	内容
0	ステップトレーストラップは無効です。
1	ステップトレーストラップは有効です。 このとき、ユーザ用 NMI とユーザ割込みがすべて割込み禁止となります。

リセットにより "0" に初期化されます。

ステップトレーストラップの機能はエミュレータが使用します。エミュレータ使用時、ユーザプログラム中で使用することはできません。

■ ILM

bit20	bit19	bit18	bit17	bit16	[初期値]
ILM4	ILM3	ILM2	ILM1	ILM0	01111 _B

割込みレベルマスク値を保持するレジスタで、この ILM の保持する値がレベルマスクに使用されます。

CPU に入力される割込み要求の中で、対応する割込みレベルが、この ILM で示されるレベルよりも強い場合にのみ割込み要求が受け付けられます。

レベル値は、0 (00000_B) が最強で、31 (11111_B) が最弱です。

プログラムから設定可能な値には制限があります。

元の値が 16 ~ 31 のとき

新たな値として設定できるのは 16 ~ 31 です。0 ~ 15 を設定する命令の実行により、(指定した値 + 16) という値が転送されます。

元の値が 0 ~ 15 のとき

0 ~ 31 の任意の値が設定可能です。

リセットにより、15 (01111_B) に初期化されます。

[PS レジスタに関する注意事項]

一部の命令で PS レジスタを先行処理しているため、下記の例外動作により、デバッガの使用時に割込み処理ルーチンでブレークしたり、PS フラグの表示内容が更新されたりする場合があります。

いずれの場合も、EIT から復帰後以降に、正しく再処理を行うように設計されていますので、EIT 前後の動作は仕様どおりの処理を行います。

1. DIV0U/DIV0S 命令の直前の命令で、(a) ユーザ割込み・NMI を受け付けた場合、(b) ステップ実行を行った場合、(c) データイベントまたはエミュレータメニューにてブレークした場合、以下の動作を行う場合があります。

(1) D0, D1 フラグが先行して更新されます。

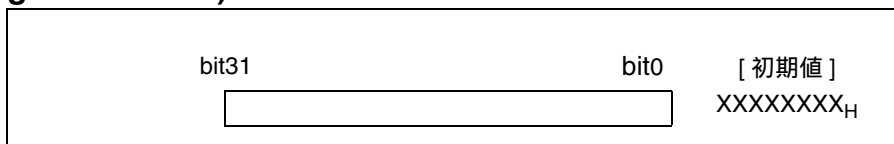
(2) EIT 処理ルーチン (ユーザ割込み・NMI, またはエミュレータ) を実行します。

(3) EIT から復帰後、DIV0U/DIV0S 命令が実行され、D0/D1 フラグが (1) と同じ値に更新されます。

2. ユーザ割込み・NMI 要因が発生している状態で、割込みを許可するために ORCCR, STILM, MOV Ri, PS の各命令が実行されると、以下の動作を行います。

- (1) PS レジスタが先行して更新されます。
- (2) EIT 処理ルーチン (ユーザ割込み・NMI) を実行します。
- (3) EIT から復帰後、上記命令が実行され、PS レジスタが(1)と同じ値に更新されます。

■ PC (Program Counter)



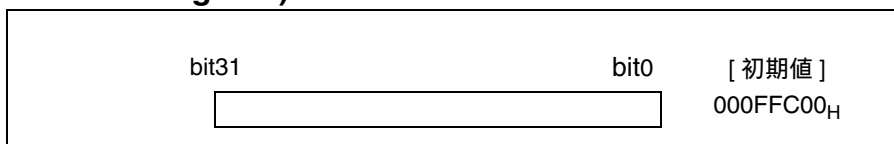
プログラムカウンタで、実行している命令のアドレスを示しています。

命令の実行を伴う PC の更新時に、bit0 は "0" に設定されます。bit0 が "1" になる可能性があるのは、分岐先アドレスとして奇数番地を指定した場合だけです。

ただし、その場合でも bit0 は無効であり、命令は "2" の倍数のアドレスに置く必要があります。

リセットによる初期値は不定です。

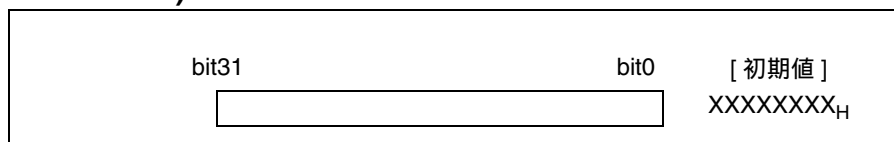
■ TBR (Table Base Register)



テーブルベースレジスタで、EIT 処理の際に使用されるベクタテーブルの先頭アドレスを保持します。

リセットによる初期値は、000FFC00_H です。

■ RP (Return Pointer)



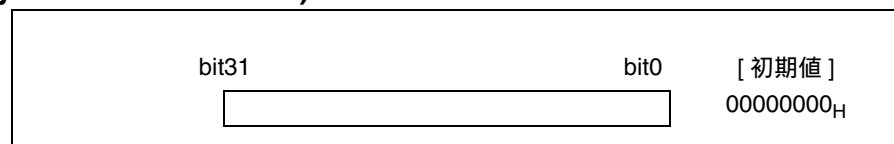
リターンポインタで、サブルーチンから復帰するアドレスを保持します。

CALL 命令実行時、PC の値がこの RP に転送されます。

RET 命令実行時、RP の内容が PC に転送されます。

リセットによる初期値は不定です。

■ SSP (System Stack Pointer)



SSP は、システムスタックポインタです。

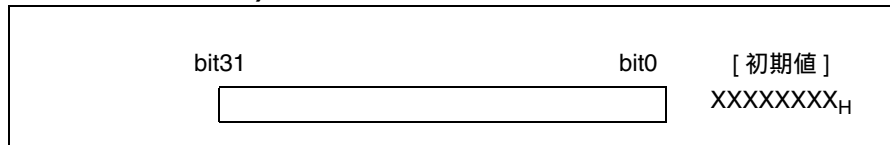
S フラグが "0" のとき、R15 として機能します。

SSP を明確に指定することも可能です。

また, EIT 発生時に, PS と PC を退避するスタックを指定するスタックポインタとしても使用されます。

リセットによる初期値は 00000000_H です。

■ USP (User Stack Pointer)



USP は, ユーザスタックポインタです。

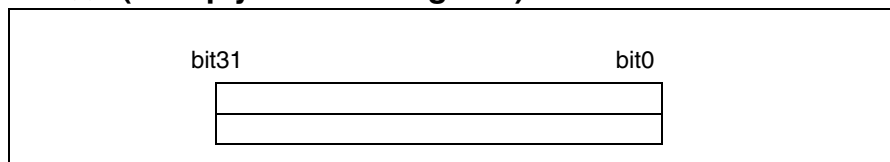
S フラグが "1" のとき, R15 として機能します。

USP を明確に指定することも可能です。

リセットによる初期値は不定です。

RETI 命令で使用することはできません。

■ 乗除算レジスタ (Multiply & Divide register)



乗除算用のレジスタで, 各々 32 ビット長です。

リセットによる初期値は不定です。

乗算実行時

32 ビット × 32 ビットの乗算のとき, 64 ビット長の演算結果は, 以下の配置で乗除算結果格納レジスタに格納されます。

MDH : 上位 32 ビット

MDL : 下位 32 ビット

16 ビット × 16 ビットの乗算のときは, 以下のように結果が格納されます。

MDH : 不定

MDL : 結果 32 ビット

除算実行時

計算開始時, 被除数を MDL に格納します。

DIV0S/DIV0U, DIV1, DIV2, DIV3, DIV4S 命令の実行により除算を計算すると, 結果が MDL と MDH に格納されます。

MDH : 剰余

MDL : 商

3.5 データ構造

ビットオーダーリング, バイトオーダーリングおよびワードアライメントについて説明します。

■ ビットオーダーリング

FR ファミリでは, ビットオーダーリングとして, リトルエンディアンを採用しています。

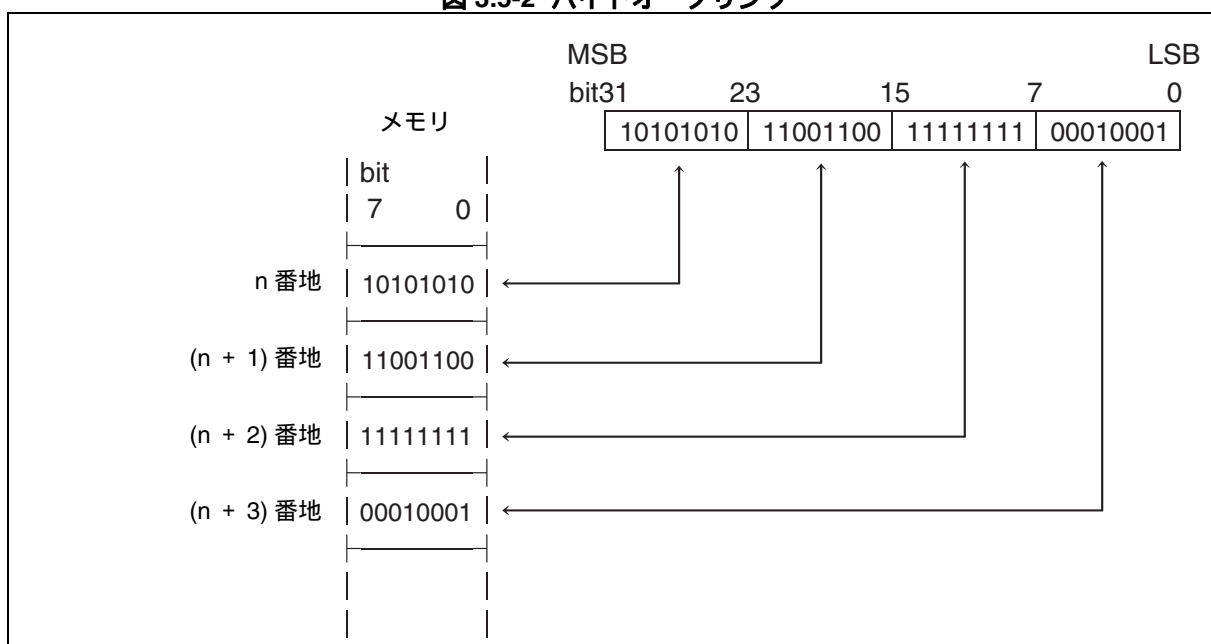
図 3.5-1 ビットオーダーリング



■ バイトオーダーリング

FR ファミリでは, バイトオーダーリングとして, ビッグエンディアンを採用しています。

図 3.5-2 バイトオーダーリング



■ ワードアライメント

● プログラムアクセス

FR ファミリのプログラムは, "2" の倍数のアドレスに配置する必要があります。

PC の bit0 は, 命令の実行に伴う PC の更新時に, "0" に設定されます。"1" になる可能性があるのは, 分岐先アドレスとして奇数番地を指定した場合だけです。

ただし, その場合でも bit0 は無効であり, 命令は "2" の倍数のアドレスに置かなくてはなりません。

奇数アドレスの例外はありません。

● データアクセス

FR ファミリでは, データアクセスを行う際, その幅により以下のように強制的なアライメントがアドレスに対して施されます。

ワードアクセス : アドレスは 4 の倍数 (最下位 2 ビットは強制的に "00")

ハーフワードアクセス : アドレスは 2 の倍数 (最下位ビットは強制的に "0")

バイトアクセス :

ワードおよびハーフワードデータアクセス時に, 一部のビットが強制的に "0" にされるのは, 実効アドレスの計算結果に対してです。例えば, @ (R13, Ri) のアドレッシングモードの場合, 加算前のレジスタは (たとえ最下位ビットが "1" であっても) そのまま計算に使用され, 加算結果の下位ビットがマスクされます。計算前のレジスタがマスクされるわけではありません。

[例] LD @ (R13, R2), R0

R13	0 0 0 0 2 2 2 2 H
R2	0 0 0 0 0 0 0 3 H
+))	
<hr/>	
加算結果	0 0 0 0 2 2 2 5 H
	下位 2 ビット強制マスク
アドレス端子	0 0 0 0 2 2 2 4 H

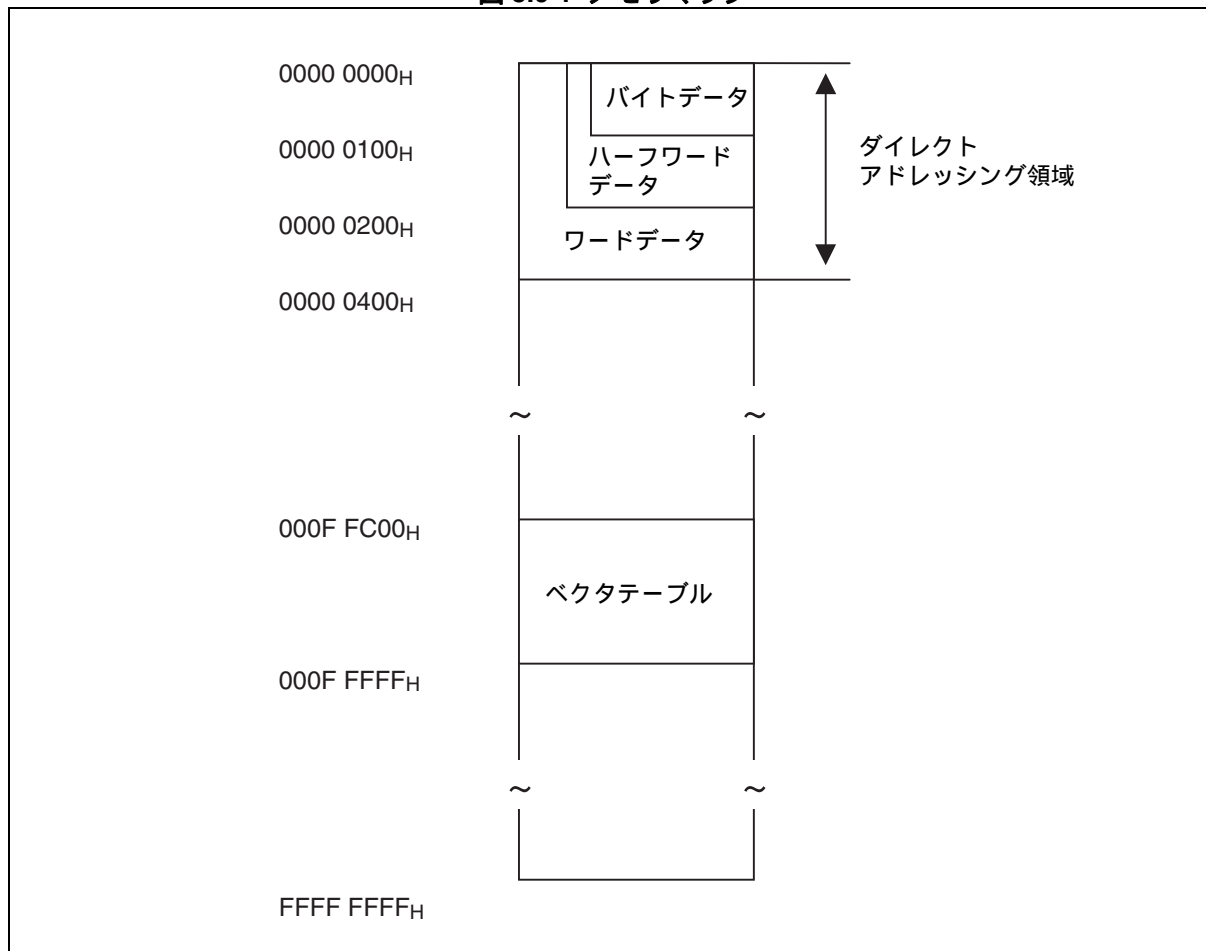
3.6 メモリマップ

MB91265A シリーズのメモリマップについて説明します。

■ メモリマップ

アドレス空間は 32 ビットリニアです。

図 3.6-1 メモリマップ



■ ダイレクトアドレッシング領域

アドレス空間の下記の領域は I/O 用の領域です。この領域は、ダイレクトアドレッシングにより、命令中で直接オペランドアドレスを指定することができます。

ダイレクトアドレス指定可能なアドレス領域の大きさは、データ長ごとに異なります。

- バイトデータ (8 ビット) : 000_H ~ 0FF_H
- ハーフワードデータ (16 ビット) : 000_H ~ 1FF_H
- ワードデータ (32 ビット) : 000_H ~ 3FF_H

■ ベクタテーブル初期領域

000FFC00_H ~ 000FFFFF_H の領域は EIT ベクタテーブル初期領域です。

EIT 処理時に使用されるベクタテーブルは, TBR を書き換えることにより任意のアドレスに配置可能ですが, リセットによる初期化によってこのアドレスに配置されます。

3.7 分岐命令

FR ファミリでは、分岐命令によって遅延スロット付きの動作と遅延スロットなしの動作を指定します。

■ 遅延スロット付き動作

● 命令

以下に示す表記をした命令が、遅延スロット付きの分岐動作を行います。

JMP:D @Ri	CALL:D label12	CALL:D @Ri	RET:D
BRA:D label9	BNO:D label9	BEQ:D label9	BNE:D label9
BC:D label9	BNC:D label9	BN:D label9	BP:D label9
BV:D label9	BNV:D label9	BLT:D label9	BGE:D label9
BLE:D label9	BGT:D label9	BLS:D label9	BHI:D label9

● 動作説明

遅延スロット付きの動作では、分岐先の命令を実行する前に、分岐命令の直後（「遅延スロット」とよびます）に置かれた命令を実行した後に、分岐します。分岐動作の前に遅延スロットの命令を実行するため、見掛け上の実行速度が1サイクルとなります。そのかわり、遅延スロットに有効な命令を入れられないときは、NOP命令を置いてください。

[例]

```

;      命令の並び
ADD    R1, R2 ;
BRA:D  LABEL ;分岐命令
MOV    R2, R3 ;遅延スロット.....分岐の前に実行される
...
LABEL: ST    R3,@R4 ;分岐先

```

条件分岐命令の場合、分岐条件が成立する場合も成立しない場合も遅延スロットに置かれた命令は実行されます。

遅延分岐命令では、一部の命令の実行順序が反転するように見えますが、それはPCの更新動作だけについてであり、その他の動作（レジスタの更新・参照など）はあくまで記述された順番で実行されます。

次に、具体的な説明をします。

(1) JMP:D @Ri / CALL:D @Ri 命令で参照する Ri は、遅延スロットの中の命令が Ri を更新しても影響を受けません。

[例]

```

LDI:32  #Label, R0
JMP:D   @R0          ; Label に分岐
LDI:8   #0,          R0 ; 分岐先アドレスには影響を与えない。
...

```

- (2) RET:D命令が参照する RP は、遅延スロットの中の命令が RP を更新しても影響を受けません。

[例]

```
RET:D                ;これより前に設定された RP の示すアドレスへ分岐
MOV    R8,    RP    ;リターン動作には影響を与えない。
```

...

- (3) Bcc:D rel 命令が参照するフラグも、遅延スロットの命令の影響を受けません。

[例]

```
ADD    #1,    R0    ;フラグ変化
BC:D    Overflow    ;上記の命令の実行結果により分岐
AND CCR#0                ;このフラグ更新は上記分岐命令では参照しない。
```

...

- (4) CALL:D 命令の遅延スロット中の命令で RP を参照すると、CALL:D 命令により更新された内容が読み出されます。

[例]

```
CALL:D Label          ;RP を更新して分岐
MOV    RP,    R0      ;上記 CALL:D の実行結果の RP を転送
```

...

● 制限事項

- (1) 遅延スロットに置ける命令

遅延スロット内で実行できるのは、以下の条件を満たす命令のみです。

- 1 サイクル命令
- 分岐命令ではないこと
- 順番が変化した場合でも動作に影響を与えない命令

「1 サイクル命令」とは、命令一覧表中でサイクル数の欄が「1」または「a」または「b」または「c」または「d」と記載された命令です。

- (2) ステップトレーストラップ

遅延スロットを持つ分岐命令の実行と遅延スロットの間では、ステップトレーストラップは発生しません。

- (3) 割り込み・NMI

遅延スロットを持つ分岐命令の実行と遅延スロットの間では、割り込み・NMI を受理しません。

- (4) 未定義命令例外

遅延スロットに未定義命令があった場合、未定義命令例外は発生しません。このとき、未定義命令は NOP 命令として動作します。

■ 遅延スロットなし動作

● 命令

以下に示す表記をした命令が、遅延スロットなしの分岐動作を行います。

JMP @Ri	CALL label12	CALL @Ri	RET
BRA label9	BNO label9	BEQ label9	BNE label9
BC label9	BNC label9	BN label9	BP label9
BV label9	BNV label9	BLT label9	BGE label9
BLE label9	BGT label9	BLS label9	BHI label9

● 動作説明

遅延スロットなしの動作では、あくまで命令の並びの順に実行します。直後の命令が分岐前に実行されることはありません。

[例]

```

;      命令の並び
ADD    R1, R2 ;
BRA    LABEL ;分岐命令(遅延スロットなし)
MOV    R2, R3 ;実行されない
...
LABEL: STR3, @R4 ;分岐先

```

遅延スロットなしの分岐命令の実行サイクル数は、分岐するとき2サイクル、分岐しないとき1サイクルとなります。

遅延スロットに適当な命令を入れることができないためにNOPを明記した遅延スロット付き分岐命令に比べ、命令コード効率を上げることができます。

遅延スロットに有効な命令を設置できるときは遅延スロット付きの動作を選択し、そうでないときは遅延スロットなしの動作を選択することで、実行速度と命令コード効率を両立させることが可能となります。

3.8 EIT (例外・割り込み・トラップ)

EIT とは、現プログラム実行時にイベントの発生により、そのプログラムの実行を中断しほかのプログラムを実行することを指し、例外 (Exception)、割り込み (Interrupt)、トラップ (Trap) の総称です。

例外とは実行中のコンテキストに関連して発生する事象です。例外を起こした命令から再実行します。

割り込みとは実行中のコンテキストに無関係に発生する事象です。イベント要因は、ハードウェアです。

トラップとは実行中のコンテキストに関連して発生する事象です。システムコールのようにプログラムで指示するものがあります。トラップを起こした命令の次の命令から再実行します。

■ 特長

- 割り込みに多重割り込みをサポート
- 割り込みにレベルマスク機能 (15 レベルをユーザが使用可能)
- トラップ命令 (INT)
- エミュレータ起動用 EIT (ハードウェア / ソフトウェア)

■ EIT 要因

EIT 要因として、次のものがあります。

- リセット
- ユーザ割り込み (内部リソース、外部割り込み)
- NMI
- 遅延割り込み
- 未定義命令例外
- トラップ命令 (INT)
- トラップ命令 (INTE)
- ステップトレーストラップ
- コプロセッサ不在トラップ
- コプロセッサエラートラップ

■ EIT からの復帰

EIT から復帰するためには RETI 命令を実行します。

■ 割込みレベル

割込みレベルは 0 ~ 31 で、5 ビットによって管理されます。

各レベルの割り当ては、以下のとおりです。

表 3.8-1 割込みレベル

レベル		割込み要因	注意事項
2 進数	10 進数		
00000	0	(システム予約)	ILM の元の値が 16 ~ 31 のとき、 この範囲の値をプログラムにより ILM に設定できません。
...	
...	
00011	3	(システム予約)	
00100	4	INTE 命令	
		ステップトレーストラップ	
00101	5	(システム予約)	
...	
...	
01110	14	(システム予約)	
01111	15	NMI (ユーザ用)	
10000	16	割込み	ILM 設定時、ユーザ割込み禁止
10001	17	割込み	
...	
...	
11110	30	割込み	
11111	31		
			ICR 設定時、割込み禁止

操作が可能なのは、16 ~ 31 のレベルです。

未定義命令例外、コプロセッサ不在トラップ、コプロセッサエラートラップ、そして INT 命令は、割込みレベルの影響を受けません。また、ILM を変化させることもありません。

■ I フラグ

割込みの許可・禁止を指定するフラグです。PS レジスタの CCR の bit4 として設けられています。

値	内容
0	割込み禁止です。 INT 命令実行時 "0" にクリアされます。 (ただし、スタック退避させる値はクリアする前の値です。)
1	割込み許可です。 割込み要求のマスク処理は、ILM の保持する値により制御されます。

■ ILM

割込みレベルマスク値を保持する PS レジスタ (bit20 ~ bit16) です。

CPU に入力される割込み要求の中で、対応する割込みレベルが、この ILM で示されるレベルよりも強い場合にのみ割込み要求が受け付けられます。

レベル値は、0 (00000_B) が最強で、31 (11111_B) が最弱です。

プログラムから設定可能な値には制限があります。元の値が 16 ~ 31 のとき、新たな値として設定できるのは 16 ~ 31 です。0 ~ 15 の値を設定する命令を実行すると、(指定した値 + 16) という値が転送されます。

元の値が 0 ~ 15 のときは、0 ~ 31 の任意の値が設定可能です。設定するには STILM 命令を使用します。

■ 割込み・NMI に対するレベルマスク

NMI および割込み要求が発生したときは、割込み要因の持つ割込みレベル (表 3.8-1 を参照) が ILM の保持するレベルマスク値と比較されます。そして、次の条件が成立したときはマスクされ、要求は受理されません。

要因の持つ割込みレベル レベルマスク値

■ ICR (Interrupt Control Register)

割り込みコントローラ内に設けられたレジスタで、割り込みの各要求に対するレベルを設定します。ICR は割り込み要求入力 of 各々に対応して用意されています。ICR は I/O 空間にマッピングされており、CPU からはバスを通してアクセスされます。

● ICR ビット構成

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
-	-	-	R	R/W	R/W	R/W	R/W	

[bit4] 本ビットは、常に "1" です。

[bit3 ~ bit0] ICR3 ~ ICR0

対応する割り込み要因の割り込みレベルの下位 4 ビットです。読出しおよび書込みが可能です。

bit4 と合わせて、ICR は 16 ~ 31 の範囲で値を設定できます。

● ICR マッピング

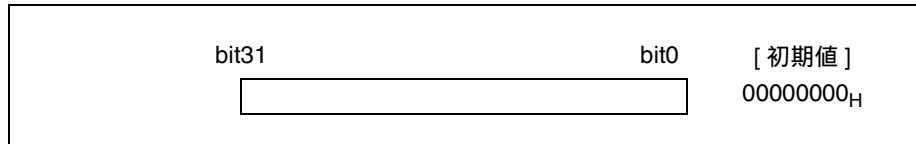
表 3.8-2 割り込み要因と割り込み制御レジスタ、割り込みベクタ

割り込み要因	割り込み制御レジスタ		対応する割り込みベクタ		
			番号		アドレス
			16 進	10 進	
IRQ00	ICR00	00000440 _H	10 _H	16	TBR + 3BC _H
IRQ01	ICR01	00000441 _H	11 _H	17	TBR + 3B8 _H
IRQ02	ICR02	00000442 _H	12 _H	18	TBR + 3B4 _H
...
...
IRQ45	ICR45	0000046D _H	3D _H	61	TBR + 308 _H
IRQ46	ICR46	0000046E _H	3E _H	62	TBR + 304 _H
IRQ47	ICR47	0000046F _H	3F _H	63	TBR + 300 _H

TBR 初期値：000F FC00_H

(注意事項) 「第5章 割り込みコントローラ」を参照してください。

■ SSP (System Stack Pointer)



SSP が、EIT の受理および復帰動作時のデータ退避・復帰用スタックを示すポイントとして使用されます。

SSP は、EIT 処理時に内容が 8 減ぜられ、RETI 命令の実行による EIT からの復帰動作時に 8 加算されます。

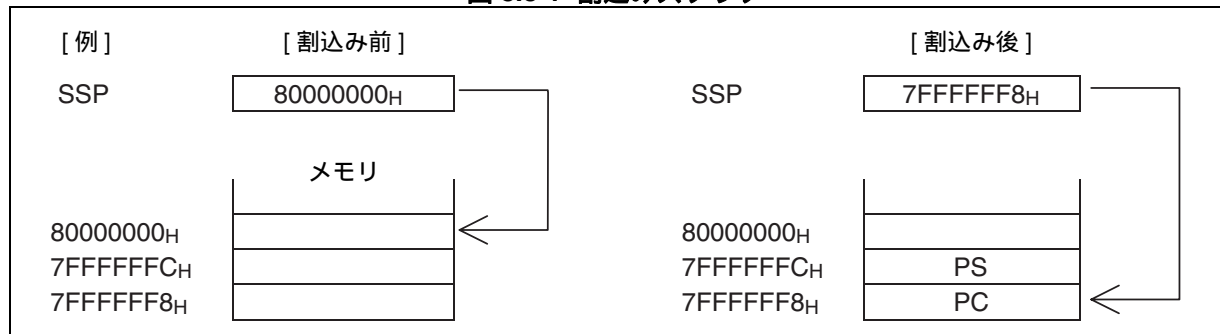
リセットによる初期値は 00000000_H です。

SSP は、CCR 中の S フラグが "0" のとき、汎用レジスタ R15 としても機能します。

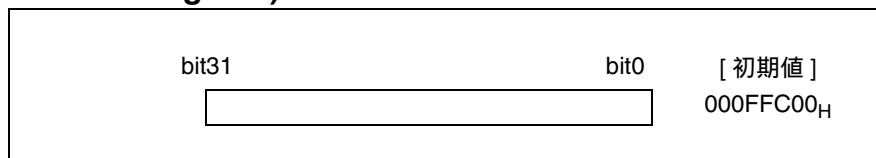
■ 割込みスタック

SSP により示される領域で、PC および PS の値が退避・復帰されます。割込み後は SSP の示すアドレスに PC、(SSP + 4) のアドレスに PS が格納されています。

図 3.8-1 割込みスタック



■ TBR (Table Base Register)



EIT 用ベクタテーブルの先頭アドレスを示すレジスタです。

TBR と EIT 要因ごとに決められたオフセット値を加算したアドレスが、ベクタアドレスとなります。

リセットによる初期値は 000FFC00_H です。

■ EIT ベクタテーブル

TBR の示すアドレスから 1 K バイトの領域が、EIT 用ベクタ領域となっています。

1 ベクタあたりの大きさは 4 バイトで、ベクタ番号とベクタアドレスの関係は下記のように表されます。

$$\begin{aligned} \text{vctadr} &= \text{TBR} + \text{vctofs} \\ &= \text{TBR} + (3\text{FC}_\text{H} - 4 \times \text{vct}) \end{aligned}$$

vctadr : ベクタアドレス

vctofs : ベクタオフセット

vct : ベクタ番号

加算結果の下位 2 ビットは常に "00_B" として扱われます。

000FFC00_H ~ 000FFFFF_H の領域がリセットによるベクタテーブルの初期領域です。

ベクタの一部には特殊な機能が割り当てられています。なお、アーキテクチャ上のベクタテーブルを表 3.8-3 に示します。

表 3.8-3 ベクタテーブル (1 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
リセット *1	0	00	-	3FC _H	000FFFFC _H
モードベクタ *1	1	01	-	3F8 _H	000FFFF8 _H
システム予約	2	02	-	3F4 _H	000FFFF4 _H
システム予約	3	03	-	3F0 _H	000FFFF0 _H
システム予約	4	04	-	3EC _H	000FFFE _C
システム予約	5	05	-	3E8 _H	000FFFE8 _H
システム予約	6	06	-	3E4 _H	000FFFE4 _H
コプロセッサ不在トラップ	7	07	-	3E0 _H	000FFFE0 _H
コプロセッサエラートラップ	8	08	-	3DC _H	000FFFD _C
INTE 命令	9	09	-	3D8 _H	000FFFD8 _H
システム予約	10	0A	-	3D4 _H	000FFFD4 _H
システム予約	11	0B	-	3D0 _H	000FFFD0 _H
ステップトレーストラップ	12	0C	-	3CC _H	000FFFC _C
NMI 要求 (tool)	13	0D	-	3C8 _H	000FFFC8 _H
未定義命令例外	14	0E	-	3C4 _H	000FFFC4 _H
NMI 要求	15	0F	15(F _H) 固定	3C0 _H	000FFFC0 _H
外部割込み 0	16	10	ICR00	3BC _H	000FFFB _C
外部割込み 1	17	11	ICR01	3B8 _H	000FFFB8 _H
外部割込み 2	18	12	ICR02	3B4 _H	000FFFB4 _H
外部割込み 3	19	13	ICR03	3B0 _H	000FFFB0 _H
外部割込み 4	20	14	ICR04	3AC _H	000FFFA _C
外部割込み 5	21	15	ICR05	3A8 _H	000FFFA8 _H
外部割込み 6/C-CAN ウェイクアップ *2	22	16	ICR06	3A4 _H	000FFFA4 _H
外部割込み 7	23	17	ICR07	3A0 _H	000FFFA0 _H

表 3.8-3 ベクタテーブル (2 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
リロードタイマ 0	24	18	ICR08	39C _H	000FFF9C _H
リロードタイマ 1	25	19	ICR09	398 _H	000FFF98 _H
リロードタイマ 2	26	1A	ICR10	394 _H	000FFF94 _H
UART0 (受信)	27	1B	ICR11	390 _H	000FFF90 _H
UART0 (送信)	28	1C	ICR12	38C _H	000FFF8C _H
DTTI 端子	29	1D	ICR13	388 _H	000FFF88 _H
DMAC0 (終了, エラー)	30	1E	ICR14	384 _H	000FFF84 _H
DMAC1 (終了, エラー)	31	1F	ICR15	380 _H	000FFF80 _H
DMAC2/3/4 (終了, エラー)	32	20	ICR16	37C _H	000FFF7C _H
UART1 (受信完了)	33	21	ICR17	378 _H	000FFF78 _H
UART1 (送信完了)	34	22	ICR18	374 _H	000FFF74 _H
C-CAN0 ^{*2}	35	23	ICR19	370 _H	000FFF70 _H
システム予約	36	24	ICR20	36C _H	000FFF6C _H
積和	37	25	ICR21	368 _H	000FFF68 _H
PPG0/PPG1	38	26	ICR22	364 _H	000FFF64 _H
PPG2/PPG3	39	27	ICR23	360 _H	000FFF60 _H
PPG4/PPG5/PPG6/PPG7	40	28	ICR24	35C _H	000FFF5C _H
システム予約	41	29	ICR25	358 _H	000FFF58 _H
波形 (アンダフロー) 0/1/2	42	2A	ICR26	354 _H	000FFF54 _H
フリーランタイマ 1 (コンペアクリア)	43	2B	ICR27	350 _H	000FFF50 _H
フリーランタイマ 1 (0 検出)	44	2C	ICR28	34C _H	000FFF4C _H
フリーランタイマ 2 (コンペアクリア)	45	2D	ICR29	348 _H	000FFF48 _H
フリーランタイマ 2 (0 検出)	46	2E	ICR30	344 _H	000FFF44 _H
タイムベースタイマ オーバフロー	47	2F	ICR31	340 _H	000FFF40 _H
フリーランタイマ 0 (コンペアクリア)	48	30	ICR32	33C _H	000FFF3C _H
フリーランタイマ 0 (0 検出)	49	31	ICR33	338 _H	000FFF38 _H
システム予約	50	32	ICR34	334 _H	000FFF34 _H
A/D コンバータ 1	51	33	ICR35	330 _H	000FFF30 _H
A/D コンバータ 2	52	34	ICR36	32C _H	000FFF2C _H
PWC0 (測定完了)	53	35	ICR37	328 _H	000FFF28 _H
システム予約	54	36	ICR38	324 _H	000FFF24 _H
PWC0 (オーバフロー)	55	37	ICR39	320 _H	000FFF20 _H
システム予約	56	38	ICR40	31C _H	000FFF1C _H
ICU 0 (取込み)	57	39	ICR41	318 _H	000FFF18 _H

表 3.8-3 ベクタテーブル (3 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
ICU 1 (取込み)	58	3A	ICR42	314 _H	000FFF14 _H
ICU2/ICU3 (取込み)	59	3B	ICR43	310 _H	000FFF10 _H
OCU0/OCU1 (一致)	60	3C	ICR44	30C _H	000FFF0C _H
OCU2/OCU3 (一致)	61	3D	ICR45	308 _H	000FFF08 _H
OCU4/OCU5 (一致)	62	3E	ICR46	304 _H	000FFF04 _H
遅延割込み要因ビット	63	3F	ICR47	300 _H	000FFF00 _H
システム予約 (REALOS にて使用)	64	40	-	2FC _H	000FFEFC _H
システム予約 (REALOS にて使用)	65	41	-	2F8 _H	000FFE8 _H
システム予約	66	42	-	2F4 _H	000FFE4 _H
システム予約	67	43	-	2F0 _H	000FFE0 _H
システム予約	68	44	-	2EC _H	000FFEEC _H
システム予約	69	45	-	2E8 _H	000FFEE8 _H
システム予約	70	46	-	2E4 _H	000FFEE4 _H
システム予約	71	47	-	2E0 _H	000FFEE0 _H
システム予約	72	48	-	2DC _H	000FFEDC _H
システム予約	73	49	-	2D8 _H	000FFED8 _H
システム予約	74	4A	-	2D4 _H	000FFED4 _H
システム予約	75	4B	-	2D0 _H	000FFED0 _H
システム予約	76	4C	-	2CC _H	000FFECC _H
システム予約	77	4D	-	2C8 _H	000FFEC8 _H
システム予約	78	4E	-	2C4 _H	000FFEC4 _H
システム予約	79	4F	-	2C0 _H	000FFEC0 _H
INT 命令で使用	80 ~ 255	50 ~ FF	-	2BC _H ~ 000 _H	000FFEC _H ~ 000FFC00 _H

*1: TBRの値を変更しても,リセットベクタとモードベクタは常に固定アドレス000FFFCH, 000FFF8_Hが使用されます。

*2: C-CAN の割込みは, MB91F267NA/MB91267NA に搭載されている機能です。

■ 多重 EIT 処理

複数の EIT 要因が同時に発生した場合、CPU は 1 つの EIT 要因を選択して受理し、EIT シーケンスを実行した後、再び EIT 要因の検出を行う、という動作を繰り返します。

EIT 要因検出の際に、受理可能な EIT 要因がなくなったとき、最後に受理した EIT 要因のハンドラの命令を実行します。

そのため、複数の EIT 要因が同時に発生した場合、各要因のハンドラの実行順序は、次の 2 つの要素により決まります。

EIT 要因受理の優先順位

受理した場合にほかの要因をどのようにマスクするか

■ EIT 要因の優先度

EIT 要因受理の優先度とは、PS と PC を退避して PC を更新し（必要に応じて）ほかの要因のマスク処理を行うという、EIT シーケンスを実行する要因を選ぶときの順番です。

必ずしも、先に受理された要因のハンドラが先に実行されるわけではありません。

EIT 要因受理の優先度は、表 3.8-4 のようになっています。

表 3.8-4 EIT 要因の受理の優先度とほかの要因へのマスク

受理の優先順位	要因	ほかの要因に対するマスク
1	リセット	ほかの要因は破棄されます。
2	未定義命令例外	取消し
3	INTE 命令	ILM = 4 ほかの要因は破棄されます。
4	INT 命令	I フラグ = 0
5	コプロセッサ不在トラップ コプロセッサエラートラップ	
6	ユーザ割込み	ILM = 受理した要因のレベル
7	NMI (ユーザ用)	ILM = 15
8	NMI (エミュレータ用)	ILM = 4
9	ステップトレーストラップ	ILM = 4

EIT 要因を受理した後のほかの要因に対してマスクの処理を考慮すると、同時に発生した EIT 要因の各ハンドラの実行順序は、表 3.8-5 のようになります。

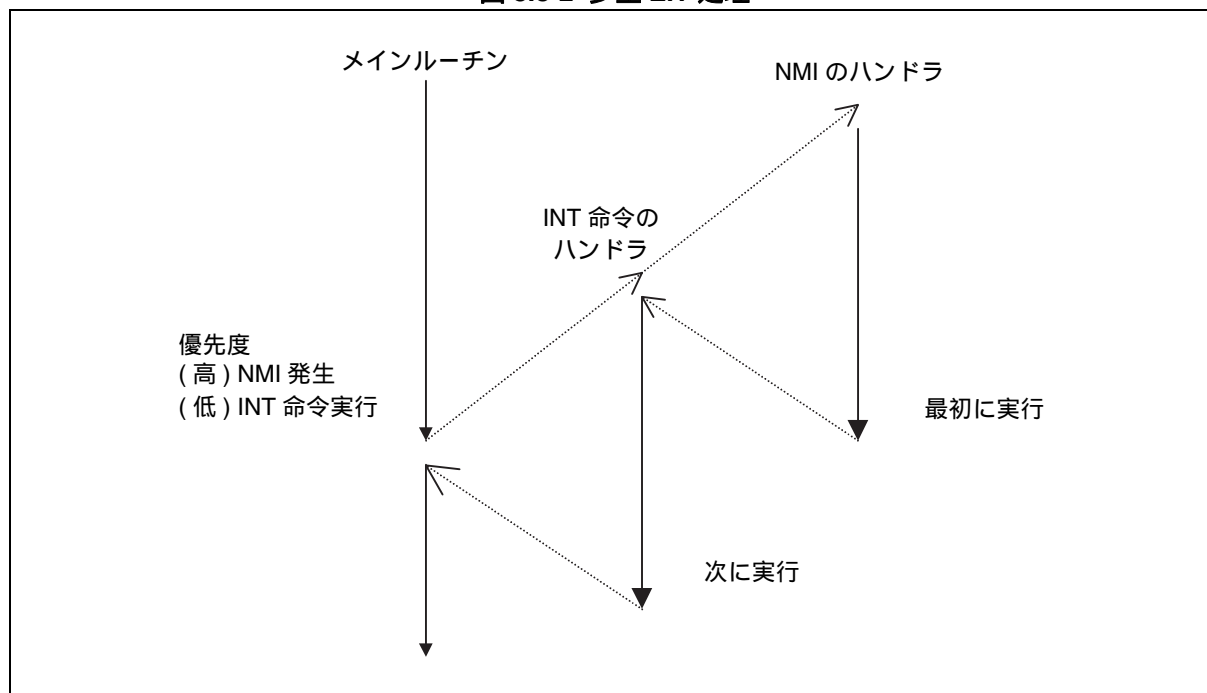
表 3.8-5 EIT ハンドラの実行順序

ハンドラの実行順序	要因
1	リセット *
2	未定義命令例外
3	INTE 命令 *
4	ステップトレーストラップ
5	NMI (ユーザ用)
6	INT 命令
7	ユーザ割込み
8	コプロセッサ不在トラップ, コプロセッサエラートラップ

*: ほかの要因は破棄されます。

[例]

図 3.8-2 多重 EIT 処理



■ EIT の動作

以下の説明で、転送元の「PC」とは各 EIT 要因を検出した命令のアドレスを示します。
また、「次の命令のアドレス」とは、EIT を検出した命令により、以下を意味します。

- LDI : 32 のとき : PC + 6
- LDI : 20, COPOP, COPLD, COPST, COPSV のとき : PC + 4
- その他の命令のとき : PC + 2

● ユーザ割込み・NMI の動作

ユーザ割込みまたはユーザ用 NMI の割込み要求が発生すると、以下の順序で要求受理の可否が判定されます。

[割込み要求受理の可否判定]

同時に発生した要求の割込みレベルを比較し、最も強いレベル (最も小さい数値) を保持するものが選択されます。

比較に使用されるレベルは、マスク可能割込みについては対応する ICR の保持する値が、NMI についてはあらかじめ定められた定数が使用されます。

同じレベルを持つ割込み要求が複数発生しているときは、最も若い割込み番号を持つ割込み要求が選択されます。

割込みレベル レベルマスク値のとき、割込み要求はマスクされ受理されません。

割込みレベル < レベルマスク値のとき、へ。

選択された割込み要求がマスク可能割込みであるとき、I フラグが "0" ならば割込み要求はマスクされ、受理されません。I フラグが "1" ならば へ。

選択された割込み要求が NMI であるとき、I フラグの値にかかわらず へ。

上記の条件が成立したとき、命令処理の切れ目で割込み要求が受理されます。

EIT 要求検出時にユーザ割込み・NMI の要求が受理されると、受理された割込み要求に対応した割込み番号を使用して、CPU は以下のように動作します。

なお、下記において [動作] における () はレジスタの指すアドレスを表します。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
次の命令のアドレス	(SSP)
受理した要求の割込みレベル	ILM
"0"	S フラグ
(TBR + 受理した割込み要求のベクタオフセット) PC	

割込みシーケンス終了後、ハンドラの先頭の命令を実行する前に、新たな EIT の検出を行います。この時点で受理可能な EIT が発生していると、CPU は EIT 処理シーケンスに遷移します。

● INT 命令の動作

INT #u8

u8 で示されるベクタの割込みハンドラへ分岐します。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
PC + 2	(SSP)
"0"	I フラグ
"0"	S フラグ
(TBR + 3FC _H - 4 × u8)	PC

● INTE 命令の動作

INTE

ベクタ番号 #9 のベクタの割込みハンドラへ分岐します。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
PC + 2	(SSP)
"00100 _B "	ILM
"0"	S フラグ
(TBR + 3D8 _H)	PC

INTE 命令, およびステップトレーストラップの処理ルーチン中では, INTE 命令は使用しないでください。

また, ステップ実行中は INTE による EIT の発生はありません。

● ステップトレーストラップの動作

PS 中の SCR における T フラグをセットしてステップトレースの機能を許可にしておくと, 1 命令実行ごとにトラップが発生し, ブレークします。

[ステップトレーストラップ検出の条件]

T フラグ = 1

遅延分岐命令ではないとき。

INTE 命令, ステップトレーストラップの処理ルーチン以外を実行中であるとき。

以上の条件が成立すると, 命令動作の切れ目でブレークします。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
次の命令のアドレス	(SSP)
"00100 _B "	ILM
"0"	S フラグ
(TBR + 3CC _H)	PC

T フラグをセットしてステップトレーストラップを許可にしたとき、ユーザ用の NMI とユーザ割込みは禁止状態となります。

また、INTE 命令による EIT は発生しなくなります。

FR ファミリでは、T フラグをセットした次の命令からトラップが発生します。

● 未定義命令例外の動作

命令のデコード時に未定義命令であることを検出すると、未定義命令例外が発生します。

[未定義命令例外の検出条件]

命令のデコード時に、未定義命令であることを検出。

遅延スロット外に置かれている（遅延分岐命令の直後ではないこと）。

以上の条件が成立すると、未定義命令例外が発生し、ブレイクします。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
PC	(SSP)
"0"	S フラグ
(TBR + 3C4 _H)	PC

PC として退避されるのは、未定義命令例外を検出した命令自身のアドレスです。

● コプロセッサ不在トラップ

実装していないコプロセッサを使用するコプロセッサ命令を実行すると、コプロセッサ不在トラップが発生します。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
次の命令のアドレス	(SSP)
"0"	S フラグ
(TBR + 3E0 _H)	PC

● コプロセッサエラートラップ

コプロセッサを使用しているときにエラーが発生した場合，次にそのコプロセッサを操作するコプロセッサ命令を実行したとき，コプロセッサエラートラップが発生します。

[動作]

SSP - 4	SSP
PS	(SSP)
SSP - 4	SSP
次の命令のアドレス	(SSP)
"0"	S フラグ
(TBR + 3DC _H)	PC

● RETI 命令の動作

RETI 命令は，EIT 処理ルーチンから復帰する命令です。

[動作]

(R15)	PC
R15 + 4	R15
(R15)	PS
R15 + 4	R15

RETI 命令は，S フラグが "0" の状態で実行する必要があります。

■ 注意事項

● 遅延スロット

分岐命令の遅延スロットには，EIT に関して制約があります。

分岐命令については「3.7 分岐命令」を参照してください。

3.9 動作モード

MB91265A シリーズの動作モードには、バスモードとアクセスモードがあります。ただし、シングルチップモードのみサポートします。

■ 動作モード

● 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指し、モード設定端子 (MD2, MD1, MD0) で指定します。

< 注意事項 >

本品種では、外バスモードをサポートしていません。

■ バスモード

バスモードの詳細は、「3.2 メモリマップ」を参照してください。

● シングルチップモード

内部 I/O, 内蔵 RAM, 内蔵 ROM が有効で、それ以外の領域へのアクセスは無効であるモードです。外部端子は、周辺リソースまたは汎用ポートとして機能します。バス端子としては機能しません。

< 注意事項 >

本品種では、外バスモードをサポートしていません。

■ モード設定

動作モードの設定は、モード端子 (MD2, MD1, MD0) とモードデータによって行います。

● モード端子

MD2, MD1, MD0 の 3 端子で、モードベクタ、リセットベクタフェッチに関する指定を行います。

表に示した設定以外は禁止です。

モード端子 MD2, MD1, MD0	モード名	リセットベクタ アクセス領域
000 _B	内 ROM モードベクタ	内部

< 注意事項 >

MD2, MD1, MD0: 000_B 以外の設定は禁止です。

● モードデータ


モードベクタフェッチ (「3.10 リセット (デバイス初期化)」の「3.10.3 リセットシーケンス」を参照) によって内部のモードレジスタ (MODR) に書き込むデータをモードデータとよびます。

モードレジスタに設定が行われた後、本レジスタの設定に従った動作モードで動作します。

モードデータは、すべてのリセット要因で設定されます。また、ユーザプログラムからは設定することはできません。

< モードデータ詳細説明 >

bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
0	0	0	0	0	1	1	1


 動作モード設定ビット

[bit31 ~ bit24] 予約ビット

必ず "00000111_B" を設定してください。"00000111_B" 以外の値を設定したときの動作は保証できません。

< 注意事項 >

モードベクタに設定するモードデータは，000FFFF8_H にバイトデータとして配置する必要があります。

MB91265Aシリーズは，バイトエンディアンとしてビッグエンディアンを採用していますので，下図のように，bit31 ~ bit24 の最上位バイトに配置してください。

		bit31	24	23	16	15	8	7	0
誤	000FFFF8 _H	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	モードデータ
正	000FFFF8 _H	モードデータ	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
	000FFFFC _H	リセットベクタ							

3.10 リセット (デバイス初期化)

リセット動作について説明します。

■ 概要

リセット要因が発生すると、デバイスはすべてのプログラムおよびハードウェア動作を停止し、状態を初期化します。この状態をリセット状態とよびます。

リセット要因の消失により、デバイスは初期状態からプログラムおよびハードウェア動作を開始します。このリセット状態から動作開始に至る一連の動作をリセットシーケンスとよびます。

3.10.1 リセットレベル

本品種のリセット動作は、2種類のレベルに分けられ、それぞれ発生要因および初期化の内容が異なります。以下、各リセットレベルについて説明します。

■ 設定初期化リセット (INIT)

すべての設定を初期化する最強レベルのリセットを設定初期化リセット (INIT) とよびます。

設定初期化リセット (INIT) により初期化される主な内容は以下のとおりです。

[設定初期化リセット (INIT) による初期化箇所]

- デバイスの動作モード (バスモードおよび外部バス幅の設定)
- 内部クロックに関するすべての設定 (クロックソース選択, PLL 制御, 分周比設定)
- 外部バスの CS0 領域に関するすべての設定
- その他端子状態に関するすべての設定
- 動作初期化リセット (RST) で初期化されるすべての箇所

詳細はそれぞれの機能の解説を参照してください。

なお、電源投入後は必ず $\overline{\text{INIT}}$ 端子にて設定初期化リセット (INIT) をかけてください。

■ 動作初期化リセット (RST)

プログラム動作を初期化する通常レベルのリセットを動作初期化リセット (RST) とよびます。

設定初期化リセット (INIT) 時には、同時に動作初期化リセット (RST) も発生します。

動作初期化リセット (RST) により初期化される主な内容は以下のとおりです。

[動作初期化リセット (RST) による初期化箇所]

- プログラム動作
- CPU および内部バス
- 周辺回路のレジスタ設定値
- I/O ポート設定
- デバイスの動作モード (バスモードおよび外部バス幅の設定)

詳細はそれぞれの機能の解説を参照してください。

3.10.2 リセット要因

各リセット発生要因と発生するリセットレベルについて説明します。

■ リセット要因

過去に発生したリセット要因は、RSRR (リセット要因レジスタ) を読み出すことにより知ることが可能です (各説明のレジスタ、フラグの詳細は、「3.11 クロック生成制御」の「3.11.6 クロック生成制御部のブロックダイアグラム」, 「3.11.7 クロック生成制御部のレジスタ詳細説明」を参照してください)。

■ $\overline{\text{INIT}}$ 端子入力 (設定初期化リセット端子)

外部端子の $\overline{\text{INIT}}$ 端子は、設定初期化リセット端子として機能します。

本端子へ "L" レベル入力を行っている間、設定初期化リセット (INIT) 要求が発生します。

本端子へ "H" レベルを入力することにより、設定初期化リセット (INIT) 要求は解除されます。

本端子要求による設定初期化リセット (INIT) が発生した場合、RSRR (リセット要因レジスタ) 中の bit15:INIT がセットされます。

本端子要求による設定初期化リセット (INIT) は、すべてのリセット要因中で最強のものであり、すべての入力・動作・状態よりも優先されます。

なお、電源投入直後は必ず $\overline{\text{INIT}}$ 端子にて設定初期化リセット (INIT) をかけてください。また、電源投入直後は、発振回路の発振安定待ち時間を確保するため、 $\overline{\text{INIT}}$ 端子への "L" レベル入力を発振回路の要求する安定待ち時間の間持続してください ($\overline{\text{INIT}}$ 端子による INIT では、発振安定待ち時間の設定は最小値に初期化されています)。

- ・ 発生要因 : 外部 $\overline{\text{INIT}}$ 端子への "L" レベル入力
- ・ 解除要因 : 外部 $\overline{\text{INIT}}$ 端子への "H" レベル入力
- ・ 発生レベル : 設定初期化リセット (INIT)
- ・ 対応フラグ : bit15:INIT

■ STCR:SRST ビット書込み (ソフトウェアリセット)

STCR (スタンバイ制御レジスタ) 中の bit4:SRST に "0" が書き込まれると、ソフトウェアリセット要求が発生します。

ソフトウェアリセット要求は、動作初期化リセット (RST) 要求です。

要求が受け付けられ動作初期化リセット (RST) が発生すると、ソフトウェアリセット要求は解除されます。

ソフトウェアリセット要求による動作初期化リセット (RST) が発生した場合、RSRR (リセット要因レジスタ) 中の bit11:SRST がセットされます。

ソフトウェアリセット要求による動作初期化リセット (RST) は、TBCR (タイムベースカウンタ制御レジスタ) 中の bit9:SYNCR がセットされている場合 (同期リセットモード)、すべてのバスアクセスが停止した後でないと発生しません。

このため、バスの使用状況によっては動作初期化リセット (RST) が発生するまでに長時間を要する場合があります。

- 発生要因 : STCR (スタンバイ制御レジスタ) 中の bit4:SRST への "0" 書き込み
- 解除要因 : 動作初期化リセット (RST) の発生
- 発生レベル : 動作初期化リセット (RST)
- 対応フラグ : bit11:SRST

■ ウォッチドッグリセット

RSRR (ウォッチドッグタイマ制御レジスタ) に対して書き込みを行うと、ウォッチドッグタイマが起動します。その後、RSRR 中の bit9, bit8:WT1, WT0 ビットにて設定した周期内に CTBR (タイムベースカウンタクリアレジスタ) への $A5_H/5A_H$ 書き込みが行われないと、ウォッチドッグリセット要求が発生します。

ウォッチドッグリセット要求は、設定初期化リセット (INIT) 要求です。要求が受け付けられ、設定初期化リセット (INIT) が発生するか、または動作初期化リセット (RST) が発生すると、ウォッチドッグリセット要求は解除されます。

ウォッチドッグリセット要求による設定初期化リセット (INIT) が発生した場合、RSRR (リセット要因レジスタ) 中の bit13:WDOG がセットされます。

なお、ウォッチドッグリセット要求による設定初期化リセット (INIT) が発生した場合は、発振安定待ち時間の設定は初期化されません。また、メインランおよびサブラン中にメイン発振を停止しない場合は、発振安定待ち時間は不要です。

- 発生要因 : ウォッチドッグタイマの設定周期経過
- 解除要因 : 設定初期化リセット (INIT) または動作初期化リセット (RST) の発生
- 発生レベル : 設定初期化リセット (INIT)
- 対応フラグ : bit13:WDOG

3.10.3 リセットシーケンス

リセット要因の消失により、デバイスはリセットシーケンスの実行を開始します。リセットシーケンスは、リセットレベルによりそれぞれ動作内容が異なります。各リセットレベルにおけるリセットシーケンスの動作内容について説明します。

■ 設定初期化リセット (INIT) 解除シーケンス

設定初期化リセット (INIT) 要求が解除されると、デバイスは以下の動作を順に実行します。

ただし、メインランおよびサブラン中にメイン発振を停止しない場合のウォッチドッグリセットでは、(2) の発振安定待ち時間を確保しません。

- (1) 設定初期化リセット (INIT) の解除、発振安定待ち状態へ遷移
- (2) 発振安定待ち時間 (STCR の bit3, bit2:OS1, OS0 にて設定) の間、動作初期化リセット (RST) 状態を保持、内部クロック停止
- (3) 動作初期化リセット (RST) 状態、内部クロック動作開始
- (4) 動作初期化リセット (RST) の解除、通常動作状態へ遷移
- (5) 000FFFF8_H 番地より、モードベクタの読出し
- (6) MODR (モードレジスタ) へ、モードベクタの書込み
- (7) 000FFFFC_H 番地より、リセットベクタの読出し
- (8) PC (プログラムカウンタ) へ、リセットベクタの書込み
- (9) PC (プログラムカウンタ) の示す番地より、プログラム動作開始

■ 動作初期化リセット (RST) 解除シーケンス

本リセットは、ソフトウェアリセットで発生します。

動作初期化リセット (RST) 要求が解除されると、デバイスは以下の動作を順に実行します。

- (1) 動作初期化リセット (RST) の解除、通常動作状態へ遷移
- (2) 000FFFF8_H 番地より、モードベクタの読出し
- (3) MODR (モードレジスタ) へ、モードベクタの書込み
- (4) 000FFFFC_H 番地より、リセットベクタの読出し
- (5) PC (プログラムカウンタ) へ、リセットベクタの書込み
- (6) PC (プログラムカウンタ) の示す番地より、プログラム動作開始

3.10.4 発振安定待ち時間

デバイスの原発振が停止していたか、またはその可能性がある状態から復帰したとき、自動的に発振安定待ち状態に遷移します。本機能により発振開始後の安定していない発振器出力を使用しないようにします。

発振安定待ち時間中は、内部および外部へのクロック供給は停止し、内蔵タイムベースカウンタのみが動作して、STCR (スタンバイ制御レジスタ) にて設定された安定待ち時間の経過を待ちます。

ここでは、発振安定待ち動作の詳細について説明します。

■ 発振安定待ち発生要因

要因を以下に示します。

(1) 設定初期化リセット (INIT) の解除時

各種要因による設定初期化リセット (INIT) が解除された直後に、発振安定待ち状態へ遷移します。

発振安定待ち時間経過後は、動作初期化リセット (RST) 状態へ遷移します。

なお、 $\overline{\text{INIT}}$ 端子による初期化時の発振安定待ち時間は最小値に設定されるため、発振安定待ちは行いません。 $\overline{\text{INIT}}$ 端子の入力幅にて発振安定待ち時間を確保してください。

(2) ストップモードからの復帰時

ストップモードが解除された直後に、発振安定待ち状態へ遷移します。

ただし、設定初期化リセット (INIT) 要求により解除された場合は、設定初期化リセット (INIT) 状態に遷移し、設定初期化リセット (INIT) が解除された後、発振安定待ち状態へ遷移します。

発振安定待ち時間経過後は、ストップモードが解除された要因に対応した状態へと遷移します。

- 有効な外部割込み要求入力 (NMI 含む) の発生による復帰時
通常動作状態へ遷移します。
- 設定初期化リセット (INIT) 要求による復帰時
設定初期化リセット (INIT) 状態へ遷移します。

(3) PLL 選択時の異常状態からの復帰時

PLL をソースクロックとして動作しているときに、PLL 制御に何らかの異常*が発生した場合、PLL ロックタイムの確保のために、自動的に発振安定待ち時間に遷移します。

発振安定待ち時間経過後は、通常動作状態へ遷移します。

*: PLL 使用中の逡倍率変更や、PLL 動作許可ビット化けの発生など

■ 発振安定待ち時間の選択

発振安定待ち時間は、内蔵タイムベースカウンタを用いてカウントされます。

発振安定待ち要因が発生し、発振安定待ち状態へ遷移すると、内蔵タイムベースカウンタはいったん初期化された後、発振安定待ち時間の計測を開始します。

STCR (スタンバイ制御レジスタ) の bit3, bit2:OS1, OS0 ビットにより、発振安定待ち時間を4種類のうちから選択して設定することができます。

いったん選択した設定は、外部 $\overline{\text{INIT}}$ 端子による設定初期化リセット (INIT) 以外では初期化されません。それ以外の設定初期化リセット (INIT) や動作初期化リセット (RST) では、リセット発生以前に設定した発振安定待ち時間が保持されます。

発振安定待ち時間として選択可能な4種類の設定は、それぞれ以下の用途を想定してあります。

- OS1, OS0 = 00_B: 発振安定待ち時間なし
(ストップモードで PLL も発振器を停止させない場合)
- OS1, OS0 = 01_B: PLL ロック待ち時間
(外部クロック入力や、ストップモードで発振器を停止させない場合)
- OS1, OS0 = 10_B: 発振安定待ち時間 (中)
(セラミック振動子などの安定が速い発振子を使用する場合)
- OS1, OS0 = 11_B: 発振安定待ち時間 (長)
(一般の水晶発振子などを使用する場合)

なお、電源投入直後は必ず $\overline{\text{INIT}}$ 端子にて設定初期化リセット (INIT) をかけてください。

また、下記状態では、発振回路の発振安定待ち時間を確保するため、 $\overline{\text{INIT}}$ 端子への "L" レベル入力を発振回路の要求する安定待ち時間の間持続してください ($\overline{\text{INIT}}$ 端子による INIT では、発振安定待ち時間の設定は最小値に初期化されています)。

- 電源投入直後の $\overline{\text{INIT}}$ 端子入力
- ストップモードで発振停止中の $\overline{\text{INIT}}$ 端子入力
- クロックソースとしてサブクロック選択中、かつメイン発振停止状態での $\overline{\text{INIT}}$ 端子入力

したがって、安定発振を行うためには、 $\overline{\text{INIT}}$ 端子入力はメインクロックの発振安定待ち時間を満たす期間、"L" レベルを入力してください。

3.10.5 リセット動作モード

動作初期化リセット (RST) には、通常 (非同期) リセットモードと同期リセットモードの2つのモードがあり、TBCR (タイムベースカウンタ制御レジスタ) 中の bit9:SYNCR によってどちらのモードで動作するかを設定します。
本モード設定は、設定初期化リセット (INIT) のみで初期化されます。
設定初期化リセット (INIT) は、常に非同期でリセット動作を行います。
ここでは、各モード動作について説明します。

■ 通常リセット動作

動作初期化リセット (RST) 要求が発生した際に、即座に動作初期化リセット (RST) 状態への遷移を行う動作を通常リセット動作とよびます。

本モードにおいては、リセット (RST) 要求が受け付けられると、内部バスアクセスの動作状態にかかわらず、即時にリセット (RST) 状態へ遷移します。

本モードでは、各状態へ遷移する時点で行われていたバスアクセスについては、その結果を保証できません。しかし、動作初期化リセット (RST) 要求を確実に受け付けることが可能です。

TBCR (タイムベースカウンタ制御レジスタ) 中の bit9:SYNCR が "0" のとき、通常リセットモードとなります。

設定初期化リセット (INIT) の発生後の初期値は、通常リセットモードとなります。

■ 同期リセット動作

動作初期化リセット (RST) 要求が発生した際に、すべてのバスアクセスが停止した後、動作初期化リセット (RST) 状態への遷移を行う動作を、同期リセット動作とよびます。
本モードにおいては、リセット (RST) 要求が受け付けられても、内部バスアクセスが行われている間は、リセット (RST) 状態への遷移は行いません。

上記要求が受け付けられると、それにより内部バスに対してスリープ要求が発行されます。各バスが動作を切り上げてスリープ状態に移行すると、動作初期化リセット (RST) 状態へ遷移します。

本モードでは、各状態へ遷移する時点ではすべてのバスアクセスが停止しているため、すべてのバスアクセスの結果を保証できます。

しかし、バスアクセスが何らかの理由により停止しない場合、その間各要求を受け付けることができなくなります。このような場合でも、設定初期化リセット (INIT) は即座に有効となります。

バスアクセスが停止しない要因は以下のとおりです。

- 外部拡張バスインタフェースに対し RDY (レディ要求) が入力され続け、バスウェイトが有効となっている場合。

(注意事項) DMA コントローラについては、各要求の受付けにより転送停止を行いますので、各状態への遷移を遅延させることはありません。

TBCR (タイムベースカウンタ制御レジスタ) 中の bit9:SYNCR が "1" のとき、同期リセットモードとなります。

設定初期化リセット (INIT) の発生後の初期値は、通常リセットモードに戻ります。

同期モードのソフトウェアリセットの使用に関しては TBCR (タイムベースカウンタ制御レジスタ) の bit9:SYNCR ビットの制限事項を参照してください。

3.11 クロック生成制御

クロック生成制御について説明します。

■ 概要

内部動作クロックは以下のようにして生成されます。

- ベースクロックの生成：ソースクロックを 2 分周または PLL 発振させ、基本クロックを生成します。
- 各内部クロックの生成：ベースクロックを分周し、各部に供給する動作クロックを生成します。

以下、各クロック生成とその制御について解説します。

各レジスタ、フラグの詳細な説明は、「3.11.6 クロック生成制御部のブロックダイアグラム」、「3.11.7 クロック生成制御部のレジスタ詳細説明」を参照してください。

3.11.1 ソースクロックの選択

ソースクロックの選択について解説します。

■ ソースクロックの選択

外部発振端子である X0/X1 端子入力に発振子を接続し、内蔵発振回路にて生成した原発振がソースクロックとなります。

外部バスクロックを含むすべてのクロック供給源は、本品種自身となります。

外部発振端子および内蔵発振回路は、メインクロックを動作中に任意に切り換えて使用することが可能です。

- メインクロック：X0/X1 端子入力から生成し、高速クロックとして使用することを想定したクロックです。

内部ベースクロックは、以下のソースクロックのうちから選択して生成します。

- メインクロックを2分周したもの
- メインクロックをPLLで逡倍したもの

ϕ はソースクロックを2分周または、PLL発振させた基本クロックを指します。したがって、システムベースクロックは、上記の内部ベースクロック発生ところで生成されるクロックです。ソースクロックの選択制御は、CLKR（クロックソース制御レジスタ）の設定によって行います。

3.11.2 PLL 制御

メインクロックに対応した PLL 発振回路について、動作（発振）許可・禁止と通倍率設定を制御することが可能です。

各制御は、CLKR（クロックソース制御レジスタ）の設定によって行います。

ここでは、各制御内容について説明します。

■ PLL 動作許可

メイン PLL 発振動作の許可 / 停止は、CLKR（クロックソース制御レジスタ）の bit10:PLL1EN の設定によって行います。

PLL1EN ビットは、設定初期化リセット (INIT) 後は "0" に初期化され、メイン PLL の発振動作は停止しています。停止中は、ソースクロックとしてメイン PLL 出力を選択することはできません。

プログラム動作を開始したら、まずクロックソースとして使用するメイン PLL の通倍率を設定しかつ動作許可した後、PLL のロック待ち時間経過後にソースクロックを切り換えてください。この際の PLL ロック待ち時間は、タイムベースタイマ割込みを使用することを推奨致します。

ソースクロックとしてメイン PLL 出力を選択している間は、PLL は動作停止させることはできません。

レジスタへの書込みは無効となります。ストップモードに移行する際などで PLL を停止させたい場合は、いったんソースクロックをメインクロックの2分周したものに選択し直した後、PLL を停止させてください。

なお、STCR（スタンバイ制御レジスタ）の bit0:OSCD1 ビットにより、ストップモード中の発振が停止するように設定してある場合、PLL はストップモード遷移時に自動的に停止しますので、動作停止を改めて設定する必要はありません。その後、ストップモードから復帰する際、PLL は自動的に発振動作を開始します。ストップモード中の発振が停止しないように設定してある場合は、PLL は自動では停止しません。この場合は、必要であればストップモード移行前にあらかじめ動作停止を設定してください。

■ PLL 通倍率

メイン PLL の通倍率は、CLKR（クロックソース制御レジスタ）の bit14 ~ bit12:PLL1S2, PLL1S1, PLL1S0 ビットによって設定します。

設定初期化リセット (INIT) 後は全ビット "0" に初期化されています。

PLL 通倍率設定を初期値より変更する場合、プログラム動作開始後、PLL を動作許可する前または同時に設定してください。通倍率変更後は、ロック待ち時間経過後にソースクロックを切り換えてください。この際の PLL ロック待ち時間は、タイムベースタイマ割込みを使用することを推奨します。

動作中に PLL 通倍率設定を変更する場合、いったんソースクロックを PLL 以外に切り換えてから変更してください。通倍率変更後は、上記と同様にロック待ち時間経過後にソースクロックを切り換えてください。

PLL 通倍率設定の変更を、PLL 使用中に変更することも可能ですが、この際は通倍率設定書換え後から自動的に発振安定待ち状態に遷移し、設定された発振安定待ち時間が経過するまでの間はプログラム動作が停止します。PLL 以外にクロックソースを切り換えた場合は、プログラム動作は停止しません。

3.11.3 発振安定待ち・PLL ロック待ち時間

ソースクロックとして選択するクロックが安定動作状態にない場合、発振安定待ち時間が必要です（「3.10.4 発振安定待ち時間」を参照）。

PLL については、動作開始後、設定された周波数に出力が安定するまでにはロック待ち時間が必要です。

■ 電源投入後の待ち時間

電源投入後は、まずメインクロック用発振回路の発振安定待ち時間が必要となります。

発振安定待ち時間の設定は、INIT 端子入力（設定初期化リセット端子）により最小値に初期化されるため、この発振安定待ち時間は、INIT 端子入力へ "L" レベルを入力する時間により確保します。

この状態においては、PLL は動作許可されていないため、ロック待ち時間はここでは考慮する必要はありません。

■ 設定初期化後の待ち時間

設定初期化リセット (INIT) が解除されると、発振安定待ち状態へ遷移します。ここでは、設定された発振安定待ち時間を内部的に発生します。

INIT 端子入力後の初めの発振安定待ち状態では、設定時間は最小値に初期化されているため、すぐに本状態は終了し、動作初期化リセット (RST) 状態へと遷移します。また、メインランおよびサブラン中にメイン発振を停止しない場合のウォッチドッグリセットでは、発振安定待ち時間をとりません。

これらの状態においては、PLL はいずれも動作許可されていないため、ロック待ち時間はここでは考慮する必要がありません。

■ PLL 動作許可後の待ち時間

プログラム動作開始後、停止状態の PLL を動作許可した場合、ロック待ち時間が経過しないとその PLL 出力を使用しないでください。

ソースクロックとしてメイン PLL を選択していなければ、ロック待ち時間中もプログラム動作は実行可能です。この際の PLL ロック待ち時間は、タイムベースタイマ割込みを使用することを推奨します。

■ PLL 通倍率変更後の待ち時間

プログラム動作開始後、動作状態の PLL の通倍率設定を変更した場合も、ロック待ち時間が経過しないとその PLL 出力を使用しないでください。

ソースクロックとしてメイン PLL を選択していなければ、ロック待ち時間中もプログラム動作は実行可能です。この際の PLL ロック待ち時間は、タイムベースタイマ割込みを使用することを推奨します。

■ ストップモード復帰後の待ち時間

プログラム動作開始後、ストップモードに遷移した場合の解除時には、プログラムにて設定された時間の発振安定待ち時間を内部的に発生します。

ストップモード中に、ソースクロックとして選択しているクロック用発振回路を停止させる設定の場合、その発振回路の発振安定待ち時間と、使用している PLL のロック待ち時間のうち、長い方の時間が必要となります。ストップモードに遷移させる前に、あらかじめ長い方の発振安定待ち時間を設定しておいてください。

ストップモード中に、ソースクロックとして選択しているクロック用発振回路を停止させない設定の場合は、PLL は自動では動作を停止しません。したがって、PLL を停止させない限り発振安定待ち時間は不要です。

ストップモードに遷移させる前に、あらかじめ発振安定待ち時間を最小値に設定しておくことを推奨致します。

3.11.4 クロック分配

ソースクロックより生成されたベースクロックを元に、各機能用の動作クロックをそれぞれ作成します。

内部動作クロックは、CPU クロック、周辺クロック、外部バスクロックがあり、それぞれが独立に分周比を設定できます。

ここでは、各内部動作クロックについて説明します。

■ CPU クロック (CLKB)

CPU と内部メモリおよび内部バスに使用されるクロックです。

本クロックを使用する回路には、以下のものがあります。

- CPU
- 内蔵 RAM, 内蔵 ROM
- ビットサーチモジュール
- I-bus, D-bus, F-bus, X-bus
- DMA コントローラ
- On chip Debug Support Unit (DSU)

動作可能な上限周波数を超える周波数になる逡倍率と分周比の組合せは設定しないでください。

■ 周辺クロック (CLKP)

周辺リソースおよびペリフェラルバスに使用されるクロックです。

本クロックを使用する回路には、以下のものがあります。

- ペリフェラル (周辺) バス
- クロック制御部 (バスインタフェース部のみ)
- 割込みコントローラ
- I/O ポート
- 外部割込み入力, UART, 16 ビットタイマなどの周辺リソース

動作可能な上限周波数を超える周波数になる逡倍率と分周比の組合せは設定しないでください。

■ 外部バスクロック (CLKT)

外部拡張バスインタフェースに使用されるクロックです。

本クロックを使用する回路には、以下のものがあります。

- 外部拡張バスインタフェース
- 外部クロック出力

動作可能な上限周波数を超える周波数になる逡倍率と分周比の組合せは設定しないでください。

< 注意事項 >

- 本品種では、外バスモードをサポートしていません。
 - CPU の処理能力はフラッシュウェイトレジスタ (FLWC) の設定にも影響を受けます。必ず本レジスタの設定を最適な値にしてご利用ください。「19.2.2 フラッシュウェイトレジスタ (FLWC)」をご参照ください。
-

3.11.5 クロック分周

各内部動作クロックは、それぞれ独立にベースクロックからの分周比を設定できます。この機能により、各回路に最適な動作周波数を設定します。

■ クロック分周

分周比は、DIVR0 (基本クロック分周設定レジスタ 0) および DIVR1 (基本クロック分周設定レジスタ 1) にて設定します。各レジスタには各クロックに対応する 4 ビットずつの設定ビットがあり、(レジスタ設定値 + 1) がそのクロックのベースクロックに対する分周比となります。分周比設定が奇数であっても、常にデューティ比は 50% となります。

設定値の変更があった場合、設定後、次のクロックの立上りから変更後の分周比が有効となります。

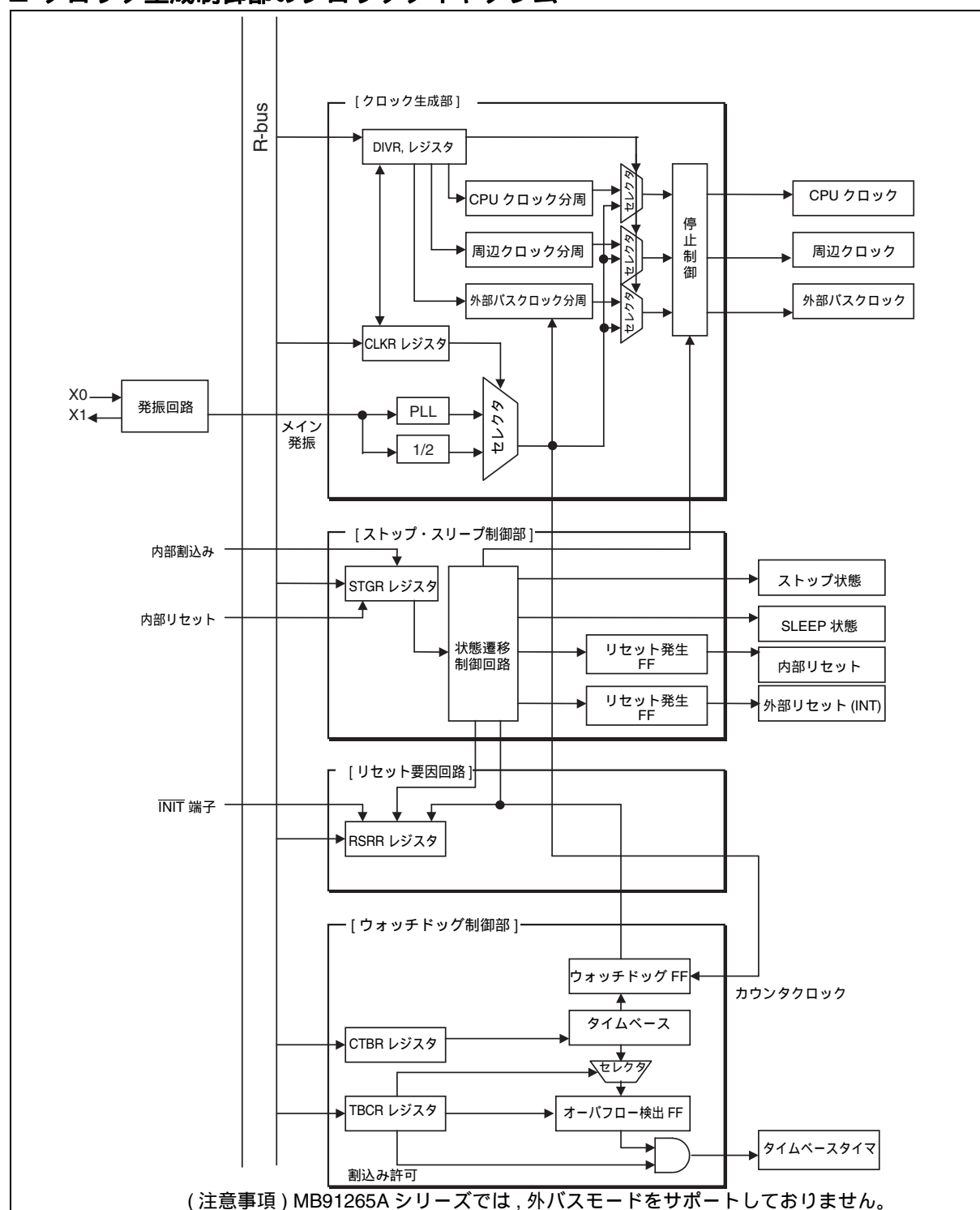
分周比設定は、動作初期化リセット (RST) の発生では初期化されず、リセット発生前の設定が維持されます。設定初期化リセット (INIT) の発生によってのみ初期化されます。初期状態からソースクロックを高速なものに変更する前に、必ず分周比の設定を行ってください。

ソースクロックの選択、メイン PLL の通倍率の設定、分周比の設定の組合せで、上限周波数を超える設定をした場合、動作は保証されません。十分にご注意願います (ソースクロック選択の変更設定との順序を間違えないように特に留意願います)。

3.11.6 クロック生成制御部のブロックダイアグラム

以下にクロック生成制御部のブロックダイアグラムを示します。図中のレジスタの詳細な解説は、「3.11.7 クロック生成制御部のレジスタ詳細説明」を参照してください。

■ クロック生成制御部のブロックダイアグラム



3.11.7 クロック生成制御部のレジスタ詳細説明

クロック生成制御部のレジスタについて説明します。

■ RSRR : リセット要因レジスタ / ウォッチドッグタイマ制御レジスタ

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
アドレス : 000480 _H	INIT	-	WDOG	-	SRST	-	WT1	WT0
	R	R	R	R	R	R	R/W	R/W
初期値 ($\overline{\text{INIT}}$ 端子)	1	0	0	0	0	0	0	0
初期値 (INIT)	*	*	*	x	x	*	0	0
初期値 (RST)	x	x	x	*	*	x	0	0

* : 要因により変化します。
x : 初期化されません。

直前に発生したリセットの要因の保持, ウォッチドッグタイマの周期設定, および起動制御を行うレジスタです。

本レジスタを読み出すと, 保持されたリセット要因は読出し後にクリアされます。読み出すまでの間に複数回のリセットが発生した場合, リセット要因フラグは累積され, 複数のフラグがセットされることになります。

本レジスタに書き込むと, ウォッチドッグタイマが起動されます。それ以降は, リセット (RST) が発生するまで, ウォッチドッグタイマは動作を続けます。

[bit15] INIT (INITialize reset occurred)

$\overline{\text{INIT}}$ 端子入力によるリセット (INIT) の発生の有無を示します。

0	$\overline{\text{INIT}}$ 端子入力による INIT は発生していません。
1	$\overline{\text{INIT}}$ 端子入力による INIT が発生しました。

- ・ 読出し直後に "0" に初期化されます。
- ・ 読出し可能で, 書込みはビット値に影響を与えません。

[bit14] - : 予約ビット

[bit13] WDOG (WatchDOG reset occurred)

ウォッチドッグタイマによるリセット (INIT) の発生の有無を示します。

0	ウォッチドッグタイマによる INIT は発生していません。
1	ウォッチドッグタイマによる INIT が発生しました。

- ・ $\overline{\text{INIT}}$ 端子入力によるリセット (INIT), または読出し直後に "0" に初期化されます。
- ・ 読出し可能で, 書込みはビット値に影響を与えません。

[bit12] - : 予約ビット

[bit11] SRST (Software ReSeT occurred)

STCRレジスタのSRSTビット書込み (ソフトウェアリセット) によるリセット (RST) の発生の有無を示します。

同期モードのソフトウェアリセットの使用に関しては TBCR (タイムベースカウンタ制御レジスタ) の bit9:SYNCR ビットの制限事項を参照してください。

0	ソフトウェアリセットによる RST は発生していません。
1	ソフトウェアリセットによる RST が発生しました。

- $\overline{\text{INIT}}$ 端子入力によるリセット (INIT), または読出し直後に "0" に初期化されます。
- 読出し可能で, 書込みはビット値に影響を与えません。

[bit10] - : 予約ビット

[bit9, bit8] WT1, WT0 (Watchdog interval Time select)

ウォッチドッグタイマの周期を設定します。

本ビットに書き込む値により, ウォッチドッグタイマの周期を下表に示す 4 種類のうちから選択します。

WT1	WT0	ウォッチドッグリセットの発生を 抑止するために最低限必要な CTBR への書込み間隔	CTBR への最終 5A _H 書込みから ウォッチドッグリセットが 発生するまでの時間
0	0	$\phi \times 2^{16}$ (初期値)	$\phi \times 2^{16} \sim \phi \times 2^{17}$
0	1	$\phi \times 2^{18}$	$\phi \times 2^{18} \sim \phi \times 2^{19}$
1	0	$\phi \times 2^{20}$	$\phi \times 2^{20} \sim \phi \times 2^{21}$
1	1	$\phi \times 2^{22}$	$\phi \times 2^{22} \sim \phi \times 2^{23}$

(ϕ は内部ベースクロックの周期)

- リセット (RST) により "00_B" に初期化されます。
- 読出し可能で, 書込みはリセット (RST) 後 1 回のみ有効で, それ以降の書込みは無効です。

■ STCR : スタンバイ制御レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
アドレス : 000481 _H	STOP	SLEEP	HIZ	SRST	OS1	OS0	-	OSCD1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 ($\overline{\text{INIT}}$ 端子)	0	0	1	1	0	0	1	1
初期値 (INIT)	0	0	1	1	x	x	1	1
初期値 (RST)	0	0	x	1	x	x	x	x

R/W: リード / ライト可能
x : 初期化されません。

デバイスの動作モードを制御するレジスタです。

ストップ、スリープの2つのスタンバイモードへの遷移、ストップモード中の端子および発振停止制御を行うほか、発振安定待ち時間の設定、ソフトウェアリセットの発行を行います。

同期モードのソフトウェアリセットの使用に関してはTBCR (タイムベースカウンタ制御レジスタ) の bit9:SYNCR ビットの制限事項を参照してください。

< 注意事項 >

スタンバイモードに入れる場合は、同期スタンバイモード (TBCR: タイムベースカウンタ制御レジスタの bit8: SYNCS ビットにて設定します) を使用したうえで、以下のシーケンスを必ず使用してください。

/* STCR ライト */

```
ldi    #_STCR, R0          ; STCR レジスタ (0481H)
ldi    #_Val_of_Stby, r1    ; Val_of_Stby は、STCR へのライトデータ
stb    r1, @r0              ; STCR へのライト
```

/* STBR ライト */

```
ldi    #_CTBR, r2           ; CTBR レジスタ (0483H)
ldi    #0xA5, r1             ; クリアコマンド (1)
stb    r1, @r2               ; CTBR への A5 ライト
ldi    #0xA5, r1             ; クリアコマンド (2)
stb    r1, @r2               ; CTBR への A5 ライト
```

/* ここでタイムベースカウンタクリア */

```
ldub    @r0, r1              ; STCR リード
```

/* 同期スタンバイ遷移開始 */

```
ldub    @r0, r1              ; STCR ダミーリード
nop                                           ; タイミング調整用の NOP x 5
nop
nop
nop
nop
nop
```

以下に、スタンバイ制御レジスタ (STCR) の各ビットの機能を説明します。

[bit7] STOP (STOP mode)

ストップモードへの遷移を指示します。bit6:SLEEP ビットと本ビット両方に "1" を書き込んだ場合は、本ビットの方が優先となり、ストップモードへ遷移します。

0	ストップモードへの遷移は行いません (初期値)。
1	ストップモードへ遷移します。

- ・ リセット (RST) およびストップ復帰要因により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit6] SLEEP (SLEEP mode)

スリープモードへの遷移を指示します。bit7:STOP ビットと本ビット両方に "1" を書き込んだ場合は、bit7:STOP ビットの方が優先となり、ストップモードへ遷移します。

0	スリープモードへの遷移は行いません (初期値)。
1	スリープモードへ遷移します。

- ・ リセット (RST) およびスリープ復帰要因により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit5] HIZ (HIZ mode)

ストップモード時の端子状態を制御します。

0	ストップモード遷移前の端子状態を維持します。
1	ストップモード中は端子出力をハイインピーダンス状態にします (初期値)。

- ・ リセット (INIT) により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit4] SRST (Software ReSeT)

ソフトウェアリセット (RST) の発行を指示します。

0	ソフトウェアリセットを発行します。
1	ソフトウェアリセットの発行は行いません (初期値)。

- ・ リセット (RST) により "1" に初期化されます。
- ・ 読出しおよび書込みが可能です。読出し値は、常に "1" となります。

[bit3, bit2] OS1, OS0 (Oscillation Stabilization time select)

リセット (INIT) 後, およびストップモード復帰後などにおける発振安定待ち時間を設定します。

本ビットに書き込む値により, 発振安定待ち時間を下表に示す 4 種類のうちから選択します。

OS1	OS0	発振安定待ち時間	メイン発振 4 MHz の場合
0	0	$\phi \times 2^1$ (初期値)	1.0 μ s
0	1	$\phi \times 2^{11}$	1.0 ms
1	0	$\phi \times 2^{16}$	32.7 ms
1	1	$\phi \times 2^{22}$	2.0 s

(ϕ は内部ベースクロックの周期で, メイン発振の 2 倍の周期)

- $\overline{\text{INIT}}$ 端子入力によるリセット (INIT) により "00_B" に初期化されます。
- 読出しおよび書込みが可能です。

[bit1] - : 予約ビット

予約ビットです。本品種では, 本ビットへの書込みは常に"1"を書き込んでください。

[bit0] OSCD1 (Oscillation Disable mode for XIN1)

メイン発振入力 (X0, X1) における, ストップモード時の発振停止を制御します。

0	ストップモード中もメイン発振は停止しません。
1	ストップモード中はメイン発振を停止します (初期値)。

- リセット (INIT) により "1" に初期化されます。
- 読出しおよび書込みが可能です。

■ TBCR : タイムベースカウンタ制御レジスタ

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
アドレス : 000482 _H	TBIF	TBIE	TBC2	TBC1	TBC0	-	SYNCR	SYNCS
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 (INIT)	0	0	x	x	x	x	0	0
初期値 (RST)	0	0	x	x	x	x	x	x
R/W: リード / ライト可能								
X: 初期化されません。								

タイムベースタイマ割込みなどを制御するレジスタです。

タイムベースタイマ割込みの許可、割込みインターバル時間の選択を行うほか、リセット動作のオプション機能の設定を行います。

以下に、タイムベースカウンタ制御レジスタ (TBCR) の各ビットの機能を説明します。

[bit15] TBIF (Time-Base timer Interrupt Flag)

タイムベースタイマ割込みフラグです。

タイムベースカウンタが設定されたインターバル時間 (bit13 ~ bit11: TBC2 ~ TBC0 ビットにて設定) を経過したことを示します。

bit14: TBIE ビットにより割込み発生が許可 (TBIE=1) されているときに、本ビットが "1" になると、タイムベースタイマ割込み要求が発生します。

クリア要因	命令による "0" の書込み
セット要因	設定されたインターバル時間の経過 (タイムベースカウンタ出力の立下りエッジ検出)

- リセット (RST) により "0" に初期化されます。
- 読出し / 書込み可能です。ただし、書込みは "0" のみ可能で、"1" を書込みしてもビット値は変化しません。
また、リードモディファイライト (RMW) 系命令での読出し値は、常に "1" となります。

[bit14] TBIE (Time-Base timer Interrupt Enable)

タイムベースタイマ割込み要求出力許可ビットです。

タイムベースカウンタのインターバル時間経過による割込み要求出力を制御します。本ビットが "1" のときに、bit15: TBIF ビットが "1" になると、タイムベースタイマ割込み要求が発生します。

0	タイムベースタイマ割込み要求出力禁止 (初期値)
1	タイムベースタイマ割込み要求出力許可

- リセット (RST) により "0" に初期化されます。
- 読出しおよび書込みが可能です。

[bit13 ~ bit11] TBC2, TBC1, TBC0 (Time-Base timer Counting time select)

タイムベースタイマで使用するタイムベースカウンタのインターバル時間を設定します。

本ビットに書き込む値により、インターバル時間を下表に示す 8 種類のうちから選択します。

TBC2	TBC1	TBC0	タイマインターバル時間	原発振 4 MHz かつ PLL が 8 通倍の場合
0	0	0	$\phi \times 2^{11}$	64 μ s
0	0	1	$\phi \times 2^{12}$	128 μ s
0	1	0	$\phi \times 2^{13}$	256 μ s
0	1	1	$\phi \times 2^{22}$	131 ms
1	0	0	$\phi \times 2^{23}$	262 ms
1	0	1	$\phi \times 2^{24}$	524 ms
1	1	0	$\phi \times 2^{25}$	1048 ms
1	1	1	$\phi \times 2^{26}$	2097 ms

(ϕ は内部ベースクロックの周期で、メイン PLL の出力周期)

- ・ 初期値は不定です。割込みを許可する前に必ず値を設定してください。
- ・ 読出しおよび書込みが可能です。

[bit10] - : 予約ビット

予約ビットです。読出し値は不定で、書込みは動作に影響を与えません。

[bit9] SYNCR (SYNChronous Reset enable)

同期リセット動作許可ビットです。

動作初期化リセット (RST) 要求またはハードウェアスタンバイ要求が発生した際に、即座にリセット (RST) またはハードウェアスタンバイ遷移を行う通常リセット動作を行うか、すべてのバスアクセスが停止した後、動作初期化リセット (RST) またはハードウェアスタンバイ遷移を行う同期リセット動作を行うかを選択します。

0	通常リセット動作 (初期値)
1	同期リセット動作

- ・ リセット (INIT) により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

< 注意事項 >

同期モードのソフトウェアリセットを使用するときは、STCR(スタンバイ制御レジスタ)の SRST ビットに "0" を設定する前に、以下 2 つの条件を必ず満たしてください。

- ・ 割込み許可フラグ (I-Flag) を割込み禁止 (I-Flag=0) に設定する。
- ・ NMI を使用しない。

[bit8] SYNCS (SYNChronous Standby enable)

同期スタンバイ動作許可ビットです。

スタンバイモード(スリープモードまたはストップモード)を使用する際は必ず"1"を設定してください。

0	通常スタンバイ動作 (初期値)
1	同期スタンバイ動作

- ・ リセット (INIT) により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

■ CTBR : タイムベースカウンタクリアレジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
アドレス : 000483 _H	D7	D6	D5	D4	D3	D2	D1	D0
	W	W	W	W	W	W	W	W
初期値 (INIT)	x	x	x	x	x	x	x	x
初期値 (RST)	x	x	x	x	x	x	x	x

W: ライトオンリ
x: 初期化されません。

タイムベースカウンタを初期化するためのレジスタです。

本レジスタに連続して {A5_H}, {5A_H} を書き込むと, {5A_H} 書込みの直後にタイムベースカウンタを全ビット "0" にクリアします。{A5_H} 書込みと {5A_H} 書込みの間の時間に制限はありませんが, {A5_H} 書込みの後に {5A_H} 以外のデータを書き込むと, 再度 {A5_H} を書き込まないと {5A_H} を書き込んでもクリア動作は行いません。

ストップ, スリープ, DMA 転送中など, CPU が動作していない間は自動的にクリアを行いますので, これらの条件が発生すると, ウォッチドッグリセットは自動的に延期されます。

本レジスタの読出し値は不定です。

< 注意事項 >

本レジスタを使用してタイムベースカウンタをクリアすると, 発振安定待ち時間, ウォッチドッグタイマ周期, およびタイムベースタイマの周期が一時的に変動します。

■ CLKR : クロックソース制御レジスタ

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
アドレス : 000484 _H	-	PLL1S2	PLL1S1	PLL1S0	-	PLL1EN	CLKS1	CLKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 (INIT)	0	0	0	0	0	0	0	0
初期値 (RST)	x	x	x	x	x	x	x	x
R/W: リード / ライト可能								
X: 初期化されません。								

内部ベースクロックとなるクロックソースの選択やメイン PLL の制御を行うレジスタです。

本レジスタにてクロックソースを選択します。また、メイン PLL の動作許可および通倍率の選択を制御します。

[bit15] - : 予約ビット

予約ビットです。本品種では、本ビットへの書込みは常に"0"を書き込んでください。

[bit14 ~ bit12] PLL1S2, PLL1S1, PLL1S0 (PLL1 ratio Select 2 ~ 0)

メイン PLL の通倍率選択ビットです。

メイン PLL の通倍率を下記の組合せのうちから選択します。

本ビットは、クロックソースとしてメイン PLL を選択している間は書換えを禁止します。

動作可能な上限周波数を超える設定をしないでください。

PLL1S2	PLL1S1	PLL1S0	メイン PLL 通倍率	メイン発振 4 MHz の場合
0	0	0	× 1 (等倍)	$\phi = 250 \text{ ns}$ (4 MHz 時)
0	0	1	× 2 (2 通倍)	$\phi = 125 \text{ ns}$ (8 MHz 時)
0	1	0	× 3 (3 通倍)	$\phi = 83.33 \text{ ns}$ (12 MHz 時)
0	1	1	× 4 (4 通倍)	$\phi = 62.50 \text{ ns}$ (16 MHz 時)
1	0	0	× 5 (5 通倍)	$\phi = 50.00 \text{ ns}$ (20 MHz 時)
1	0	1	× 6 (6 通倍)	$\phi = 41.67 \text{ ns}$ (24 MHz 時)
1	1	0	× 7 (7 通倍)	$\phi = 35.71 \text{ ns}$ (28 MHz 時)
1	1	1	× 8 (8 通倍)	$\phi = 31.25 \text{ ns}$ (32 MHz 時)

(ϕ は内部ベースクロックの周期)

- ・ リセット (INIT) により "000_B" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit11] - : 予約ビット

予約ビットです。本品種では、本ビットへの書込みは常に"0"を書き込んでください。

[bit10] PLL1EN (PLL1 ENable)

メイン PLL の動作許可ビットです。

本ビットは、クロックソースとしてメイン PLL を選択している間は書換えを禁止します。

また、本ビットが "0" の間はクロックソースとしてメイン PLL を選択することを禁止します。

bit9, bit8: CLKS1, CLKS0 ビットの設定を参照してください。

STCR の bit0:OSCD1 が "1" であると、ストップモード中は本ビットが "1" であってもメイン PLL は停止します。ストップモードからの復帰後は動作許可に戻ります。

0	メイン PLL 停止 (初期値)
1	メイン PLL 動作許可

- ・ リセット (INIT) により "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit9, bit8] CLKS1, CLKS0 (CLOCK source Select)

使用するクロックソースを設定します。

本ビットに書き込む値により、クロックソースを下表に示す 3 種類のうちから選択します。

CLKS1	CLKS0	クロックソース設定
0	0	X0/X1 からの原発振入力の 2 分周 (初期値)
0	1	設定禁止
1	0	メイン PLL
1	1	設定禁止

- ・ リセット (INIT) により "00_B" に初期化されます。
- ・ 読出しおよび書込みが可能です。

< 注意事項 >

bit9:CLKS1 が "1" の間は bit8:CLKS0 の値を変更することができません。

[変更可能な組合せ]	
"00 _B "	"10 _B "
"10 _B "	"00 _B "

上記の組合せ以外は設定禁止です。

■ DIVR0: 基本クロック分周設定レジスタ 0

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
アドレス : 000486 _H	B3	B2	B1	B0	P3	P2	P1	P0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 (INIT)	0	0	0	0	0	0	1	1
初期値 (RST)	×	×	×	×	×	×	×	×

R/W: リード / ライト可能
 × : 初期化されません。

各内部クロックのベースクロックに対する分周比を制御するレジスタです。

本レジスタでは、CPU および内部バスのクロック (CLKB) と、周辺回路およびペリフェラルバスクロック (CLKP) の分周比の設定を行います。

ソースクロックの選択、メイン PLL の通倍率の設定、分周比の設定の組合せで、上限周波数を超える設定をした場合、動作は保証されません。十分にご注意願います。また、ソースクロック選択の変更設定との順序を間違えないように注意願います。

本レジスタの設定の変更があった場合、設定後、次のクロックレートから変更後の分周比が有効となります。

[bit15 ~ bit12] B3, B2, B1, B0 (clkB divide select 3 ~ 0)

CPU クロック (CLKB) クロック分周比設定ビットです。

CPUと内部メモリおよび内部バスのクロック (CLKB) のクロック分周比を設定します。

本ビットに書き込む値により、CPU および内部バスのクロックのベースクロックに対する分周比 (クロック周波数) を下表に示す 16 種類のうちから選択します。

動作可能な上限周波数を超える周波数になる分周比は設定しないでください。

B3	B2	B1	B0	クロック分周比	クロック周波数：原発振 4 MHz およびメイン PLL8 通倍の場合
0	0	0	0	ϕ	32 MHz (初期値)
0	0	0	1	$\phi \times 2$ (2 分周)	16 MHz
0	0	1	0	$\phi \times 3$ (3 分周)	10.7 MHz
0	0	1	1	$\phi \times 4$ (4 分周)	8 MHz
0	1	0	0	$\phi \times 5$ (5 分周)	6.4 MHz
0	1	0	1	$\phi \times 6$ (6 分周)	5.33 MHz
0	1	1	0	$\phi \times 7$ (7 分周)	4.57 MHz
0	1	1	1	$\phi \times 8$ (8 分周)	4 MHz
...
1	1	1	1	$\phi \times 16$ (16 分周)	2 MHz

(ϕ は内部ベースクロックの周期)

- リセット (INIT) により "0000_B" に初期化されます。
- 読出しおよび書込みが可能です。

[bit11 ~ bit8] P3, P2, P1, P0 (clkP divide select 3 ~ 0)

周辺クロック (CLKP) クロック分周比設定ビットです。

周辺回路およびペリフェラルバスのクロック (CLKP) のクロック分周比を設定します。

本ビットに書き込む値により，周辺回路およびペリフェラルバスのクロックのベースクロックに対する分周比（クロック周波数）を下表に示す 16 種類のうちから選択します。

動作可能な上限周波数を超える周波数になる分周比は設定しないでください。

P3	P2	P1	P0	クロック分周比	クロック周波数：原発振 4 MHz およびメイン PLL8 逡倍の場合
0	0	0	0	ϕ	32 MHz
0	0	0	1	$\phi \times 2$ (2 分周)	16 MHz
0	0	1	0	$\phi \times 3$ (3 分周)	10.7 MHz
0	0	1	1	$\phi \times 4$ (4 分周)	8 MHz (初期値)
0	1	0	0	$\phi \times 5$ (5 分周)	6.4 MHz
0	1	0	1	$\phi \times 6$ (6 分周)	5.33 MHz
0	1	1	0	$\phi \times 7$ (7 分周)	4.57 MHz
0	1	1	1	$\phi \times 8$ (8 分周)	4 MHz
...
1	1	1	1	$\phi \times 16$ (16 分周)	2 MHz

(ϕ はシステムベースクロックの周期)

- リセット (INIT) により "0011_B" に初期化されます。
- 読出しおよび書込みが可能です。

■ DIVR1: 基本クロック分周設定レジスタ 1

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
アドレス: 000487 _H	T3	T2	T1	T0	-	-	-	-
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値 (INIT)	0	0	0	0	0	0	0	0
初期値 (RST)	x	x	x	x	x	x	x	x

R/W: リード / ライト可能
X: 初期化されません。

各内部クロックのベースクロックに対する分周比を制御するレジスタです。

本レジスタでは、外部バスインタフェースのクロック (CLKT) の分周比の設定を行います。

ソースクロックの選択、メイン PLL の通倍率の設定、分周比の設定の組合せで、上限周波数を超える設定をした場合、動作は保証されません。十分にご注意願います。また、ソースクロック選択の変更設定との順序を間違えないように注意願います。

本レジスタの設定の変更があった場合、設定後次のクロックレートから変更後の分周比が有効となります。

[bit7 ~ bit4] T3, T2, T1, T0 (clkT divide select 3 ~ 0)

外部バスクロック (CLKT) クロック分周比設定ビットです。

外部バスインタフェースのクロック (CLKT) のクロック分周比を設定します。

本ビットに書き込む値により、外部拡張バスインタフェースのクロックのベースクロックに対する分周比 (クロック周波数) を下表に示す16種類のうちから選択します。動作可能な上限周波数を超える周波数になる分周比は設定しないでください。

T3	T2	T1	T0	クロック分周比	クロック周波数: 原発振 4 MHz およびメイン PLL8 通倍の場合
0	0	0	0	ϕ	32 MHz (初期値)
0	0	0	1	$\phi \times 2$ (2 分周)	16 MHz
0	0	1	0	$\phi \times 3$ (3 分周)	10.7 MHz
0	0	1	1	$\phi \times 4$ (4 分周)	8 MHz
0	1	0	0	$\phi \times 5$ (5 分周)	6.4 MHz
0	1	0	1	$\phi \times 6$ (6 分周)	5.33 MHz
0	1	1	0	$\phi \times 7$ (7 分周)	4.57 MHz
0	1	1	1	$\phi \times 8$ (8 分周)	4 MHz
...
1	1	1	1	$\phi \times 16$ (16 分周)	2 MHz

(ϕ はシステムベースクロックの周期)

- リセット (INIT) により "0000_B" に初期化されます。
- 読出しおよび書込みが可能です。

外部バスインタフェースを使用されない場合は、"1111_B" (16 分周) に設定することを推奨致します。

[bit3 ~ bit0] - : 予約ビット

- ・リセット (INIT) により "0000_B" に初期化されます。
- ・本ビットへの書込みは、常に "0000_B" を書き込んでください。

(注意事項) 本品種では、外バスモードをサポートしていません。

3.11.8 クロック制御部が持つ周辺回路

クロック制御部内の周辺回路機能について説明します。

■ タイムベースカウンタ

クロック制御部には、26ビット長のタイムベースカウンタがあり、内部ベースクロックで動作しています。

タイムベースカウンタは、発振安定待ち時間の計測（「3.10.4 発振安定待ち時間」を参照）のほか、以下の用途に使用されます。

- ウォッチドッグタイマ
システムの暴走検出用のウォッチドッグタイマをタイムベースカウンタのビット出力を用いて計測します。
- タイムベースタイマ
タイムベースカウンタ出力を用いてインターバル割込みを発生します。

以下、これらの機能について解説します。

● ウォッチドッグタイマ

ウォッチドッグタイマは、タイムベースカウンタ出力を用いた暴走検出用タイマです。プログラムの暴走などで設定したインターバルの間にウォッチドッグリセットの発生延期動作が行われなくなると、ウォッチドッグリセットとして設定初期化リセット (INIT) 要求を発生します。

[ウォッチドッグタイマの起動と周期設定]

ウォッチドッグタイマは、リセット (RST) 後の 1 回目の RSRR (リセット要因レジスタ / ウォッチドッグタイマ制御レジスタ) への書込み動作により起動します。このとき、ウォッチドッグタイマのインターバル時間を bit9, bit8: WT1, WT0 ビットにより設定します。インターバル時間の設定は、この最初の書込みで設定した時間のみが有効となり、それ以降の書込みはすべて無視されます。

[ウォッチドッグリセットの発生延期]

いったんウォッチドッグタイマを起動すると、プログラムで定期的に CTBR (タイムベースカウンタクリアレジスタ) に対して {A5_H}, {5A_H} の順でデータを書き込む必要があります。この操作により、ウォッチドッグリセット発生用フラグが初期化されます。

[ウォッチドッグリセットの発生]

ウォッチドッグリセット発生用フラグは、設定したインターバルのタイムベースカウンタ出力の立下りエッジによってセットされます。2 度目の立下りエッジの検出時にフラグがセットされていると、ウォッチドッグリセットとして設定初期化リセット (INIT) 要求を発生します。

[ウォッチドッグタイマの停止]

いったんウォッチドッグタイマを起動すると、動作初期化リセット (RST) が発生するまでは、ウォッチドッグタイマを停止することはできません。

動作初期化リセット (RST) の発生する以下の状態ではウォッチドッグタイマは停止し、再度プログラム動作にて起動するまでは、機能しません。

- 動作初期化リセット (RST) 状態
- 設定初期化リセット (INIT) 状態
- 発振安定待ちリセット (RST) 状態

[ウォッチドッグタイマの一時停止 (自動発生延期)]

ウォッチドッグタイマは、CPU のプログラム動作が停止している場合には、いったんウォッチドッグリセット発生用フラグを初期化し、ウォッチドッグリセットの発生を延期します。プログラム動作の停止とは具体的には以下の動作を示します。

- スリープ状態
- ストップ状態
- 発振安定待ち RUN 状態
- D-bus (データバス) に対する DMA 転送中
- エミュレータデバッグを使用中のブレーク中

また、タイムベースカウンタのクリアを行うと、同時にウォッチドッグリセット発生用フラグも初期化され、ウォッチドッグリセットの発生が延期されます。

● タイムベースタイマ

タイムベースタイマは、タイムベースカウンタ出力を用いたインターバル割込み発生用タイマです。

メイン PLL のロック待ち時間やサブクロックなどの発振安定待ち時間など、最大で { ベースクロック $\times 2^{27}$ } サイクルまでの比較的長時間の時間計測を行う用途に適しています。

設定したインターバルに対応するタイムベースカウンタの出力の立下りエッジを検出すると、タイムベースタイマ割込み要求が発生します。

[タイムベースタイマの起動とインターバル設定]

タイムベースタイマは、TBCR (タイムベースカウンタ制御レジスタ) の bit13, bit12, bit11:TBC2, TBC1, TBC0 ビットにてインターバル時間を設定します。

設定したインターバルに対応するタイムベースカウンタの出力の立下りエッジは常に検出されているため、インターバル時間の設定後はまず bit15:TBIF ビットをクリアした後、bit14:TBIE ビットを "1" にして割込み要求出力を許可してください。

インターバル時間を変更する際は、あらかじめ bit14:TBIE ビットを "0" にして割込み要求出力を禁止しておいてください。

タイムベースカウンタはこれら設定に影響されず常にカウント動作を行っていますので、正確なインターバル割込み時間を得るためには、割込みを許可する前にタイムベースカウンタをクリアしてください。そうでないと、割込み許可直後に割込み要求が発生することもあり得ます。

[プログラムによるタイムベースカウンタのクリア]

CTBR (タイムベースカウンタクリアレジスタ) に対して {A5_H}, {5A_H} の順でデータを書き込むと, {5A_H} 書込みの直後に, タイムベースカウンタを全ビット"0"にクリアします。{A5_H} 書込みと {5A_H} 書込み間の時間は制限がありませんが, {A5_H} 書込みの後に {5A_H} 以外のデータを書き込むと, 再度 {A5_H} を書き込まないと {5A_H} を書き込んででもクリア動作は行いません。

このタイムベースカウンタのクリアを行うことにより, 同時にウォッチドッグリセット発生用フラグも初期化され, ウォッチドッグリセットの発生がいったん延期されます。

[デバイス状態によるタイムベースカウンタのクリア]

タイムベースカウンタは, 以下のデバイス状態の遷移時に, 同時に全ビット"0"にクリアされます。

- ストップ状態
- 設定初期化リセット (INIT) 状態
- ハードウェアスタンバイ状態

特にストップ状態の場合, 発振安定待ち時間の計測のためにタイムベースカウンタが使用されるため, 意図せずにタイムベースタイマのインターバル割込みが発生してしまうことがあります。そのため, ストップモードを設定する前には, タイムベースタイマ割込みを禁止し, タイムベースタイマを使用しないようにしてください。

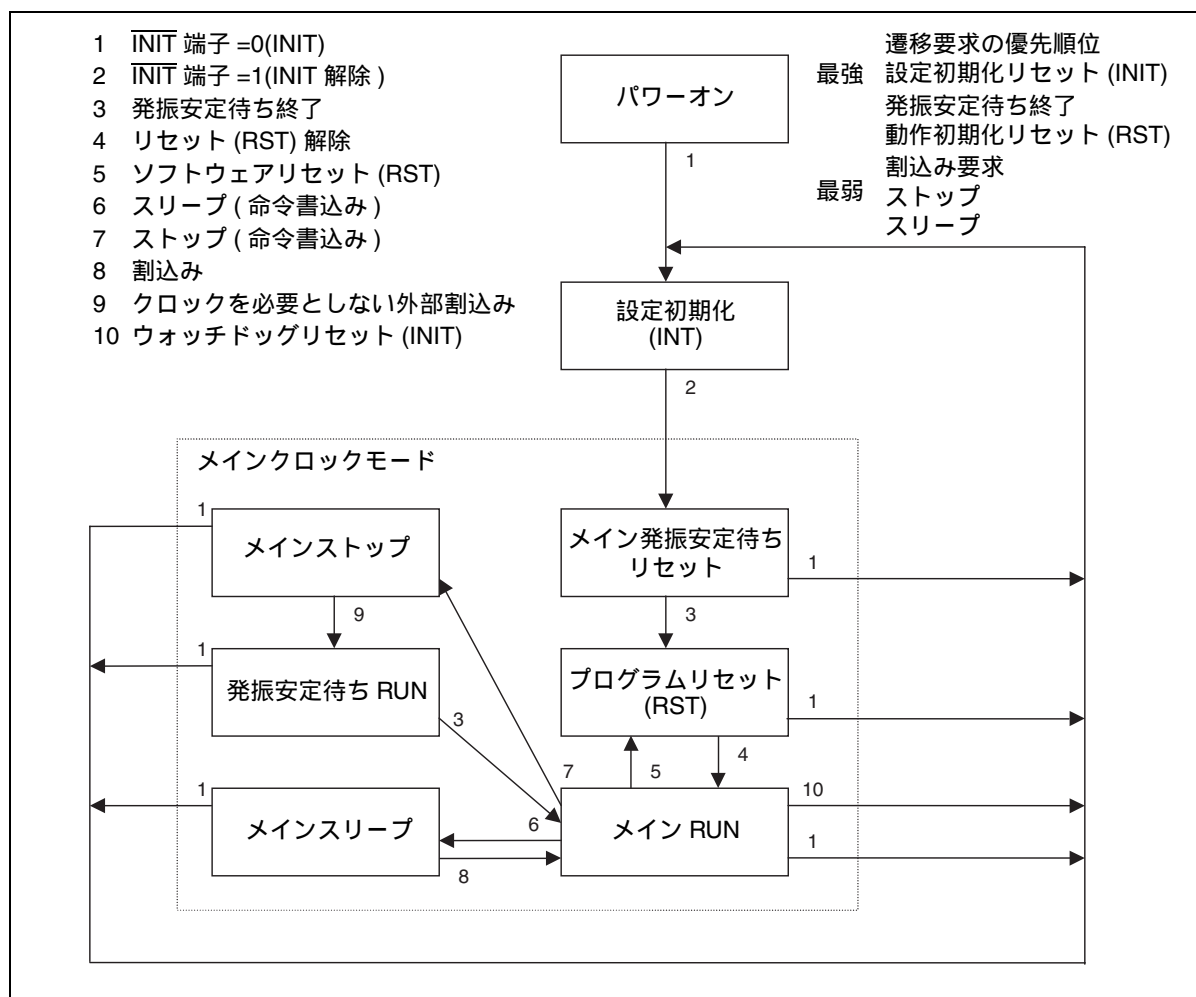
それ以外の状態については, 動作初期化リセット (RST) が発生するため, タイムベースタイマ割込みは自動的に禁止されます。

3.12 デバイス状態制御

各種状態とその制御について解説します。

■ デバイス状態と各遷移

状態遷移を下記に示します。



本品種のデバイスの動作状態には以下のものがあります。

● RUN 状態 (通常動作)

プログラム実行状態です。

すべての内部クロックが供給され、すべての回路が動作可能な状態です。

ただし、16 ビット周辺 (ペリフェラル) バスのみは、アクセスが行われていない期間はバスクロックのみ停止しています。

各状態遷移要求を受け付けますが、同期リセットモードを選択している場合、通常リセットモードの場合と一部要求に対する状態遷移動作が異なります。詳細は、「3.10.5 リセット動作モード」の「■ 同期リセット動作」を参照してください。

● スリープ状態

プログラム停止状態です。プログラム動作により遷移します。
 CPU のプログラム実行のみ停止し、周辺回路は動作可能な状態です。各種内蔵メモリおよび内部 / 外部バスは DMA コントローラが要求しない限り停止状態です。
 有効な割込み要求の発生により、本状態は解除され、RUN 状態 (通常動作) へ遷移します。
 設定初期化リセット (INIT) 要求の発生により、設定初期化リセット (INIT) 状態へ遷移します。
 動作初期化リセット (RST) 要求の発生により、動作初期化リセット (RST) 状態へ遷移します。

● ストップ状態

デバイス停止状態です。プログラム動作により遷移します。
 すべての内部回路が停止します。内部クロックはすべて停止し、発振回路およびメイン PLL は設定により停止させることが可能です。
 また、設定により、外部端子を一律ハイインピーダンスにすることが可能です (一部端子を除く)。
 特定の (クロックを必要としない) 有効な割込み要求の発生により、発振安定待ち RUN 状態へ遷移します。
 設定初期化リセット (INIT) 要求の発生により、設定初期化リセット (INIT) 状態へ遷移します。
 動作初期化リセット (RST) 要求の発生により、発振安定待ちリセット (RST) 状態へ遷移します。

● 発振安定待ち RUN 状態

デバイス停止状態です。ストップ状態からの復帰後に遷移します。
 クロック発生制御部 (タイムベースカウンタおよびデバイス状態制御部) を除くすべての内部回路が停止します。内部クロックはすべて停止しますが、発振回路および動作許可されていたメイン PLL は動作しています。
 ストップ状態などでの外部端子のハイインピーダンス制御は解除されます。
 設定された発振安定待ち時間の経過により、RUN 状態 (通常動作) へ遷移します。
 設定初期化リセット (INIT) 要求の発生により、設定初期化リセット (INIT) 状態へ遷移します。
 動作初期化リセット (RST) 要求の発生により、発振安定待ちリセット (RST) 状態へ遷移します。

● 発振安定待ちリセット (RST) 状態

デバイス停止状態です。ストップ状態または設定初期化リセット (INIT) 状態からの復帰後に遷移します。
 クロック発生制御部 (タイムベースカウンタおよびデバイス状態制御部) を除くすべての内部回路が停止します。内部クロックはすべて停止しますが、発振回路および動作許可されていたメイン PLL は動作しています。
 ストップ状態などでの外部端子のハイインピーダンス制御は解除されます。
 内部回路に対し、動作初期化リセット (RST) を出力します。
 設定された発振安定待ち時間の経過により、発振安定待ちリセット (RST) 状態へ遷移します。
 設定初期化リセット (INIT) 要求の発生により、設定初期化リセット (INIT) 状態へ遷移します。

● 動作初期化リセット (RST) 状態

プログラム初期化状態です。動作初期化リセット (RST) 要求の受付け, または発振安定待ちリセット (RST) 状態の終了により遷移します。

CPU のプログラム実行は停止し, プログラムカウンタは初期化されます。周辺回路は一部を除き初期化されます。すべての内部クロックと発振回路および動作許可されていたメイン PLL は動作しています。

内部回路に対し, 動作初期化リセット (RST) を出力します。

動作初期化リセット (RST) 要求の消失により, RUN 状態 (通常動作) へ遷移し, 動作初期化リセットシーケンスを実行します。設定初期化リセット (INIT) 状態からの復帰後だった場合は, 設定初期化リセットシーケンスを実行します。

設定初期化リセット (INIT) 要求の発生により, 設定初期化リセット (INIT) 状態へ遷移します。

● 設定初期化リセット (INIT) 状態

全設定初期化状態です。設定初期化リセット (INIT) 要求の受付けにより遷移します。

CPU のプログラム実行は停止し, プログラムカウンタは初期化されます。周辺回路はすべて初期化されます。発振回路は動作しますが, メイン PLL は動作を停止します。すべての内部クロックは, 外部 $\overline{\text{INIT}}$ 端子への "L" レベル入力期間は停止しますが, それ以外では動作します。

内部回路に対し, 設定初期化リセット (INIT) および動作初期化リセット (RST) を出力します。

設定初期化リセット (INIT) 要求の消失により, 本状態は解除され, 発振安定待ちリセット (RST) 状態へ遷移します。その後, 動作初期化リセット (RST) 状態を経て, 設定初期化リセットシーケンスを実行します。

● 各状態遷移要求の優先順位

どの状態においても, 各状態遷移要求は以下の優先順位に従います。ただし, 一部要求は特定の状態でしか発生しませんので, その状態でしか有効になりません。

[最強] 設定初期化リセット (INIT) 要求

発振安定待ち時間の終了 (発振安定待ちリセット状態および発振安定待ち RUN 状態のみ発生)

動作初期化リセット (RST) 要求

有効な割込み要求 (RUN, スリープ, ストップ状態のみ発生)

ストップモード要求 (レジスタ書込み) (RUN 状態のみ発生)

[最弱] スリープモード要求 (レジスタ書込み) (RUN 状態のみ発生)

■ 低消費電力モード

本品種の状態のうち、各低消費電力モードとその使用方法について説明します。

本品種の低消費電力モードには、以下のものがあります。

- スリープモード
レジスタ書込みにより、デバイスをスリープ状態へ遷移させます。
 - ストップモード
レジスタ書込みにより、デバイスをストップ状態へ遷移させます。
- 以下、各モードについて説明します。

● スリープモード

STCR (スタンバイ制御レジスタ) の bit6:SLEEP ビットに "1" を書き込むと、スリープモードとなり、スリープ状態へ遷移します。以降、スリープ状態からの復帰要因が発生するまでは、スリープ状態を維持します。

スリープ状態については、「3.12 デバイス状態制御 デバイス状態と各遷移 スリープ状態」も参照してください。

[スリープモードへの移行]

スリープモードに入れる場合は、同期スタンバイモード (TBCR: タイムベースカウンタ制御レジスタの bit8: SYNCSC ビットにて設定します) を使用したうえで、以下のシーケンスを必ず使用してください。

/* STCR ライト */

```
ldi    #_STCR, R0          ; STCR レジスタ (0481H)
ldi    #_Val_of_Stby, r1    ; Val_of_Stby は、STCR へのライトデータ
stb    r1, @r0             ; STCR へのライト
```

/* STBR ライト */

```
ldi    #_CTBR, r2          ; CTBR レジスタ (0483H)
ldi    #0xA5, r1           ; クリアコマンド (1)
stb    r1, @r2             ; CTBR への A5 ライト
ldi    #0xA5, r1           ; クリアコマンド (2)
stb    r1, @r2             ; CTBR への A5 ライト
```

/* ここでタイムベースカウンタクリア */

```
ldub    @r0, r1            ; STCR リード
```

/* 同期スタンバイ遷移開始 */

```
ldub    @r0, r1            ; STCR ダミーリード
nop                                           ; タイミング調整用の NOP × 5
nop
nop
nop
nop
nop
```

STCR (スタンバイ制御レジスタ) の bit7:STOP ビットと本ビット両方に "1" を書き込んだ場合は、bit7:STOP ビットの方が優先となり、ストップ状態へ遷移します。

[スリープ状態で停止する回路]

- CPU のプログラム実行

以下は、DMA 転送が発生した場合は動作します。

- ビットサーチモジュール
- 各種内蔵メモリ
- 内部 / 外部バス

(注意事項) 本品種では、外バスモードをサポートしておりません。

[スリープ状態で停止しない回路]

- 発振回路
- 動作許可されたメイン PLL
- クロック発生制御部
- 割込みコントローラ
- 周辺回路
- 時計タイマ
- メイン発振安定待ちタイマ
- DMA コントローラ
- On chip Debug Support Unit (DSU)

[スリープ状態からの復帰要因]

- 有効な割込み要求の発生

ICRレジスタの設定が割込み禁止(1111_B)でない割込み要求が発生すると、スリープモードは解除され、RUN 状態 (通常動作) へ遷移します。このとき、CPU の PS レジスタの I フラグを "1" に設定して割込み受付けを許可状態にし、スリープ復帰後、割込みハンドラを実行するようにしてください。

ICR レジスタの設定が割込み禁止 (1111_B) の割込み要求が発生しても、スリープモードは解除されません。

- 設定初期化リセット (INIT) 要求の発生

設定初期化リセット (INIT) 要求が発生すると、無条件で設定初期化リセット (INIT) 状態へ遷移します。

- 動作初期化リセット (RST) 要求の発生

動作初期化リセット (RST) 要求が発生すると、無条件で動作初期化リセット (RST) 状態へ遷移します。

(注意事項) 各要因の優先順位については、「3.12 デバイス状態制御 デバイス状態と各遷移 各状態遷移要求の優先順位」を参照してください。

[同期スタンバイ動作]

タイムベースカウンタ制御レジスタ (TBCR) の bit8(SYNCS ビット) に "1" が設定してある場合、同期スタンバイ動作が許可されます。この場合、SLEEP ビットへの書き込みのみではスリープ状態へは遷移しません。その後、STCR レジスタを読み出すことによってスリープ状態へ遷移します。

スリープモードを使用する場合は、[スリープモードへの移行]にあるシーケンスを必ず使用してください。

● ストップモード

STCR (スタンバイ制御レジスタ) の bit7:STOP ビットに "1" を書き込むと、ストップモードとなり、ストップ状態へ遷移します。以降、ストップ状態からの復帰要因が発生するまでは、ストップ状態を維持します。

ストップ状態については、「3.12 デバイス状態制御 デバイス状態と各遷移 ストップ状態」を参照してください。

[ストップモードへの移行]

ストップモードに入れる場合は、同期スタンバイモード (TBCR: タイムベースカウンタ制御レジスタの bit8:SYNCS ビットにて設定します) を使用したうえで、以下のシーケンスを必ず使用してください。

```
/* STCR ライト */
ldi    #_STCR, R0          ; STCR レジスタ (0481H)
ldi    #_Val_of_Stby, r1    ; Val_of_Stby は、STCR へのライトデータ
stb     r1, @r0             ; STCR へのライト

/* STBR ライト */
ldi    #_CTBR, r2          ; CTBR レジスタ (0483H)
ldi    #0xA5, r1           ; クリアコマンド (1)
stb     r1, @r2            ; CTBR への A5 ライト
ldi    #0xA5, r1           ; クリアコマンド (2)
stb     r1, @r2            ; CTBR への A5 ライト

/* ここでタイムベースカウンタクリア */
ldub     @r0, r1           ; STCR リード

/* 同期スタンバイ遷移開始 */
ldub     @r0, r1           ; STCR ダミーリード
nop                                     ; タイミング調整用の NOP × 5
nop
nop
nop
nop
nop
```

STCR (スタンバイ制御レジスタ) の bit6:SLEEP ビットと本ビット両方に "1" を書き込んだ場合は、bit7:STOP ビットの方が優先となり、ストップ状態へ遷移します。

[ストップ状態で停止する回路]

下記を除くすべての内部回路

[ストップ状態で停止しない回路]

- 停止するよう設定されていない発振回路

STCR (スタンバイ制御レジスタ) の bit0:OSCD1 ビットに "0" が設定してあるとき、ストップ状態中のメインクロック用発振回路は停止しません。

- 動作許可されていて、かつ停止するよう設定されていない発振回路に接続されたメイン PLL

STCR (スタンバイ制御レジスタ) の bit0:OSCD1 ビットに "0" が設定してあるとき、CLKR (クロックソース制御レジスタ) の bit10:PLL1EN ビットに "1" が設定してあると、ストップ状態中のメインクロック用 PLL は停止しません。

[ストップ状態での端子のハイインピーダンス制御]

STCR (スタンバイ制御レジスタ) の bit5:HIZ ビットに "1" が設定してあると、ストップ状態中の端子出力をハイインピーダンス状態にします。本制御の対象となる端子は、「付録 C 各 CPU ステートにおける端子状態」を参照してください。

STCR (スタンバイ制御レジスタ) の bit5:HIZ ビットに "0" が設定してあると、ストップ状態中の端子出力はストップ状態への遷移前の値を保持します。詳細は、「付録 C 各 CPU ステートにおける端子状態」を参照してください。

[ストップ状態からの復帰要因]

- 特定の (クロックを必要としない) 有効な割込み要求の発生
一部の外部割込み、および $\overline{\text{NMI}}$ 入力端子のみが有効です。
ICR レジスタの設定が割込み禁止 (1111_B) でない割込み要求が発生すると、ストップモードは解除され、発振安定待ち RUN 状態へ遷移します。このとき、CPU の PS レジスタの I フラグを "1" に設定して割込み受け付けを許可状態にし、ストップ復帰後、割込みハンドラを実行するようにしてください。
ICR レジスタの設定が割込み禁止 (1111_B) の割込み要求が発生しても、ストップモードは解除されません。
- 設定初期化リセット (INIT) 要求の発生
設定初期化リセット (INIT) 要求が発生すると、無条件で設定初期化リセット (INIT) 状態へ遷移します。
- 動作初期化リセット (RST) 要求の発生
動作初期化リセット (RST) 要求が発生すると、無条件で動作初期化リセット (RST) 状態へ遷移します。

(注意事項) 各要因の優先順位については、「3.12 デバイス状態制御 デバイス状態と各遷移 各状態遷移要求の優先順位」を参照してください。

[ストップモード時のクロックソース選択]

自励発振モードでは、ストップモードを設定する前に、メインクロックの 2 分周をソースクロックとするようあらかじめ選択しておいてください。詳細は、「3.11 クロック生成制御」、特に「3.11.2 PLL 制御」を参照してください。

なお、分周比の設定に関しては、通常動作時と制限事項は変わりません。

[同期スタンバイ動作]

タイムベースカウンタ制御レジスタ (TBCR) の bit8:SYNCS ビットに "1" が設定してある場合、同期スタンバイ動作が許可されます。この場合、STOP ビットへの書込みのみではストップ状態へは遷移しません。その後、STCR レジスタを読み出すことによってストップ状態へ遷移します。

ストップモードを使用する場合は、[ストップモードへの移行] にあるシーケンスを必ず使用してください。

第4章

I/O ポート

I/O ポートの概要，レジスタの構成および機能について説明します。

- 4.1 ポート基本ブロックダイアグラム
- 4.2 I/O ポートのレジスタ
- 4.3 アナログ入力ポート

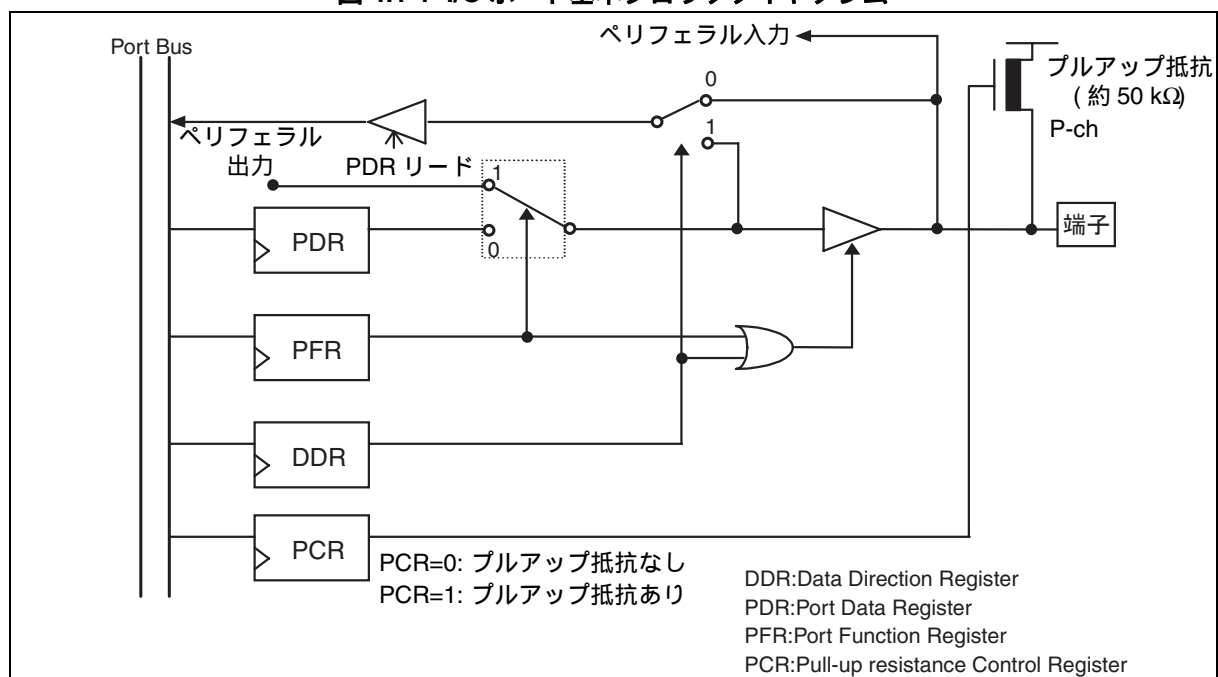
4.1 ポート基本ブロックダイアグラム

MB91265A シリーズは、各端子に対応する外部バスインタフェースやペリフェラルが入出力として端子を使用しない設定になっているとき、I/O ポートとして使用できます。

■ ポート基本ブロックダイアグラム

下図にポートの基本的な構成を示します。

図 4.1-1 I/O ポート基本ブロックダイアグラム



4.1.1 ブルアップ抵抗付き入出力ポート

I/O ポートは、以下の4種類のレジスタにより使用します。

- PDR(ポートデータレジスタ /Port Data Register)
 - DDR(ポート方向レジスタ /Data Direction Register)
 - PFR(ポート機能レジスタ /Port Function Register)
 - PCR(ブルアップ抵抗制御レジスタ /Pull-up resistance Control Register)
-

■ I/O ポートのモード

- ポート入力モード時 (PFR=0 & DDR=0)
 - PDR 読出し：対応する外部端子のレベルが読み出されます。
 - PDR 書込み：PDR に設定値が書き込まれます。
 - ポート出力モード時 (PFR=0 & DDR=1)
 - PDR 読出し：PDR の値が読み出されます。
 - PDR 書込み：PDR の値が対応する外部端子に出力されます。
 - ペリフェラル出力モード時 (PFR=1 & DDR=x)
 - PDR 読出し：対応するペリフェラル出力の値が読み出されます。
 - PDR 書込み：PDR に設定値が書き込まれます。
-

< 注意事項 >

- ポートへのアクセスは、バイトアクセスで行ってください。
 - ストップモード時 (HIZ=0) には、ブルアップ抵抗制御レジスタの設定が優先されます。
 - ストップモード時 (HIZ=1) には、ブルアップ抵抗制御レジスタの設定は無効となります。
 - 汎用ポート入力とペリフェラル入力を切り換えるレジスタはありません。
外部端子から入力された値は常に汎用ポートとペリフェラル回路へ伝播しています。
また、DDR で出力設定時も外部へ出力している値が常に汎用ポートとペリフェラル回路へ伝播しています。
ペリフェラル入力としてご使用の場合は、DDR で入力設定にし、各ペリフェラルの入力信号を有効にしてご使用ください。
-

4.2 I/O ポートのレジスタ

I/O ポートのレジスタについて説明します。

■ Port Data Register(PDR: PDR0 ~ PDR5, PDRG)

PDR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000000 _H	P07	P06	P05	P04	P03	P02	P01	P00	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR1									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000001 _H	P17	P16	P15	P14	P13	P12	P11	P10	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR2									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000002 _H	P27	P26	P25	P24	P23	P22	P21	P20	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR3									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000003 _H	P37	P36	P35	P34	P33	P32	P31	P30	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR4									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000004 _H	-	P46	P45	P44	P43	P42	P41	P40	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDR5									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000005 _H	P57	P56	P55	P54	P53	P52	P51	P50	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PDRG									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000010 _H	-	-	-	-	-	-	PG1	-	-----X- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

PDR0 ~ PDR5, PDRG は, I/O ポートの入出力データレジスタです。対応する DDR0 ~ DDR5, DDRG, PFR0 ~ PFR5, PFRG で, 入出力制御が行われます。

■ Data Direction Register (DDR: DDR0 ~ DDR5, DDRG)

DDR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000400 _H	P07	P06	P05	P04	P03	P02	P01	P00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR1									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000401 _H	P17	P16	P15	P14	P13	P12	P11	P10	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR2									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000402 _H	P27	P26	P25	P24	P23	P22	P21	P20	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR3									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000403 _H	P37	P36	P35	P34	P33	P32	P31	P30	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR4									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000404 _H	-	P46	P45	P44	P43	P42	P41	P40	-0000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDR5									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000405 _H	P57	P56	P55	P54	P53	P52	P51	P50	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
DDRG									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000410 _H	-	-	-	-	-	-	PG1	-	-----0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

DDR0 ~ DDR5, DDRG は、対応するI/Oポートの入出力方向をビット単位で制御します。

PFR=0 のとき DDR=0 : ポート入力

DDR=1 : ポート出力

PFR=1 のとき DDR=0 : ペリフェラル入力

DDR=1 : ペリフェラル出力

■ Pull-up Resistance Control Register (PCR: PCR0 ~ PCR4, PCRG)

PCR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000600 _H	P07	P06	P05	P04	P03	P02	P01	P00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR1									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000601 _H	P17	P16	P15	P14	P13	P12	P11	P10	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR2									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000602 _H	P27	P26	P25	P24	P23	P22	P21	P20	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR3									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000603 _H	P37	P36	-	-	-	-	-	-	00----- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCR4									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000604 _H	-	-	-	-	P43	P42	P41	P40	----0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
PCRG									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000610 _H	-	-	-	-	-	-	PG1	-	-----0- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

PCR0 ~ PCR4, PCRG は、対応する I/O ポートのプルアップ抵抗制御を行います。

PCR=0 : プルアップ抵抗なし

PCR=1 : プルアップ抵抗あり

■ Port Function Register(PFR: PFR0, PFR1, PTFR0)

PFR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000420 _H	TOT2E	TOT1E	-	-	-	-	-	-	00----- _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PFR1									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000421 _H	TX0E*	-	SCK1E	-	SOT1	SCK0E	-	SOT0E	0-0-00-0 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PTFR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000433 _H	PPG7E	PPG6E	PPG5E	PPG4E	PPG3E	PPG2E	PPG1E	PPG0E	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

PFR0, PFR1, PTFR0 は , 対応するペリフェラルの出力をビット単位で制御します。
PFR の空きビットには , 必ず "0" を書いてください。

* : PFR1 レジスタの bit7 の TX0E(C-CAN の TX0 出力許可ビット) は , MB91F267NA/
MB91267NA の設定です。
その他の品種は , 必ず "0" を書いてください。

< 注意事項 >

MB91F267NA/MB91267NA の P17 の PPG6 と TX0 出力は , PPG6E=1, TX0E=1 の設
定の場合 , PPG6 の出力が優先されます。

以下に各 PFR レジスタについてその初期値と機能をまとめます。

レジスタ名	ビット	ビット名	設定値	機能
PFR0	7	TOT2E	0	汎用ポート [初期値]
			1	リロードタイマ2出力
	6	TOT1E	0	汎用ポート [初期値]
			1	リロードタイマ1出力
PFR1	7	TX0E *	0	汎用ポート [初期値]
			1	C-CAN データ出力
	5	SCK1E	0	汎用ポート [初期値]
			1	UART1 のクロック出力
	3	SOT1	0	汎用ポート [初期値]
			1	UART1 のデータ出力
	2	SCK0E	0	汎用ポート [初期値]
			1	UART0 のクロック出力
	0	SOT0E	0	汎用ポート [初期値]
			1	UART0 のデータ出力

* : PFR1 レジスタの bit7 の TX0E(C-CAN の TX0 出力許可ビット) は, MB91F267NA/MB91267NA の設定です。

その他の品種は, 必ず "0" を書いてください。

レジスタ名	ビット	ビット名	設定値	機能
PTFR0	7	PPG7E	0	汎用ポート [初期値]
			1	PPG タイマ7出力
	6	PPG6E	0	汎用ポート [初期値]
			1	PPG タイマ6出力
	5	PPG5E	0	汎用ポート [初期値]
			1	PPG タイマ5出力
	4	PPG4E	0	汎用ポート [初期値]
			1	PPG タイマ4出力
	3	PPG3E	0	汎用ポート [初期値]
			1	PPG タイマ3出力
	2	PPG2E	0	汎用ポート [初期値]
			1	PPG タイマ2出力
	1	PPG1E	0	汎用ポート [初期値]
			1	PPG タイマ1出力
	0	PPG0E	0	汎用ポート [初期値]
			1	PPG タイマ0出力

< 注意事項 >

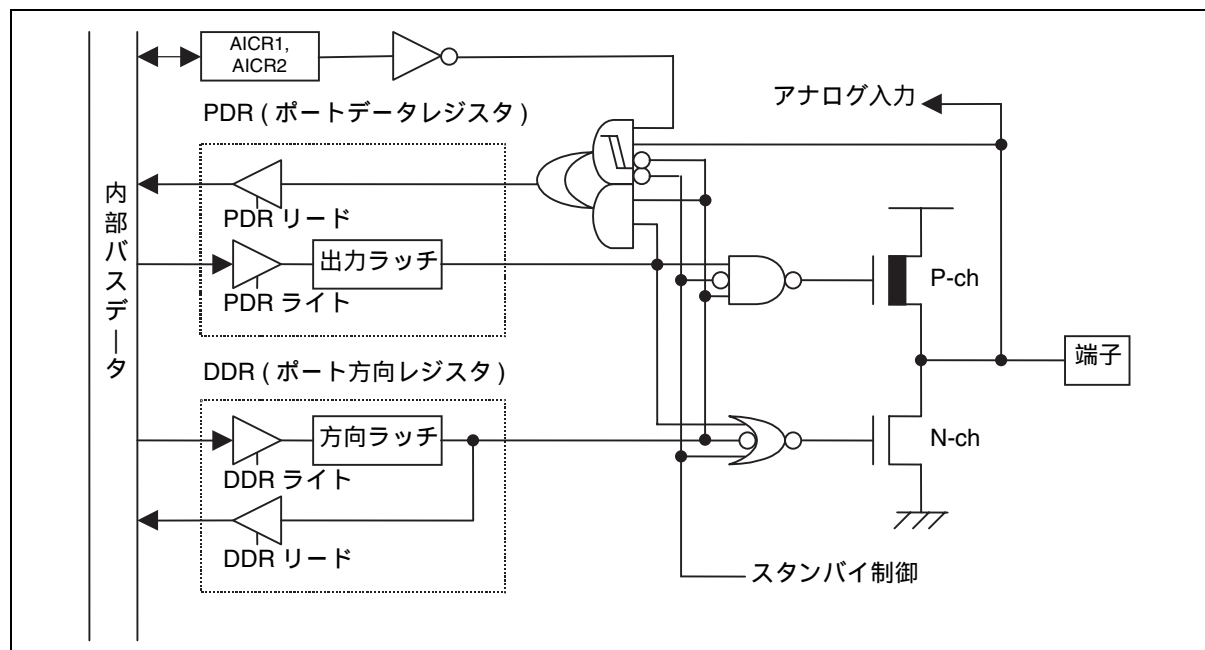
MB91F267NA/MB91267NA の P17 の PPG6 と TX0 出力は , PPG6E=1, TX0E=1 の設定の場合 , PPG6 の出力が優先されます。

4.3 アナログ入力ポート

A/D コンバータのブロックダイアグラムと使用しているレジスタについて説明します。

■ 8/10 ビット A/D コンバータの端子のブロックダイアグラム

図 4.3-1 AN0 ~ AN10 端子のブロックダイアグラム



< 注意事項 >

- 入力ポートとして使用する端子は、対応する DDR レジスタのビットを "0" に設定し、かつ外部端子にプルアップ抵抗を付加してください。また AICR レジスタの対応するビットを "0" に設定してください。
- アナログ入力端子として使用する端子は、対応する AICR レジスタのビットを "1" に設定してください。このときの PDR レジスタの読出し値は、PDR のレジスタ値になります。

■ Analog Input Control Register (AICR: AICR1, AICR2)

AICR1									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00000086 _H	-	-	-	-	AN3E	AN2E	AN1E	AN0E	----0000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

AICR2									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000008E _H	-	AN10E	AN9E	AN8E	AN7E	AN6E	AN5E	AN4E	-0000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

AICR は , 対応する I/O ポートの各端子を以下のように制御します。

AICR=0 : ポート入力モード

AICR=1 : アナログ入力モード

リセットで "0" になります。

第5章

割込みコントローラ

割込みコントローラの概要，レジスタの構成 / 機能および動作について説明します。

- 5.1 概要
- 5.2 レジスター一覧
- 5.3 ブロックダイアグラム
- 5.4 レジスタ詳細説明
- 5.5 割込みコントローラの動作

5.1 概要

割込みコントローラは、割込み受付け / 調停処理を管理します。

■ ハードウェア構成

割込みコントローラは、以下のレジスタおよび回路より構成されます。

- ICR レジスタ
- 割込み優先度判定回路
- 割込みレベル、割込み番号（ベクタ）発生部
- ホールドリクエスト取下げ要求発生部

■ 主要機能

割込みコントローラには、主に以下の機能があります。

- NMI 要求 / 割込み要求の検出
- 優先度判定（レベルおよび番号による）
- 判定結果の要因の割込みレベル伝達（CPU へ）
- 判定結果の要因の割込み番号伝達（CPU へ）
- NMI/ 割込みレベルが "1111_B" 以外の割込み発生によるストップモードからの復帰指示（CPU へ）
- DMAC へのホールドリクエスト取下げ要求発生

5.2 レジスタ一覧

図 5.2-1 に割込みコントローラのレジスタ一覧を示します。

■ レジスタ一覧

図 5.2-1 割込みコントローラのレジスタ一覧

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000440 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR00
000441 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01
000442 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02
000443 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03
000444 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04
000445 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05
000446 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06
000447 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07
000448 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR08
000449 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR09
00044A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR10
00044B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR11
00044C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR12
00044D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR13
00044E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR14
00044F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR15
000450 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR16
000451 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR17
000452 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR18
000453 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR19
000454 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR20
000455 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR21
000456 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR22
000457 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR23
000458 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR24
000459 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR25
00045A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR26
00045B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR27
00045C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR28
00045D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR29
00045E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR30
00045F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR31
				R	R/W	R/W	R/W	R/W	

(続く)

第 5 章 割込みコントローラ

(続き)

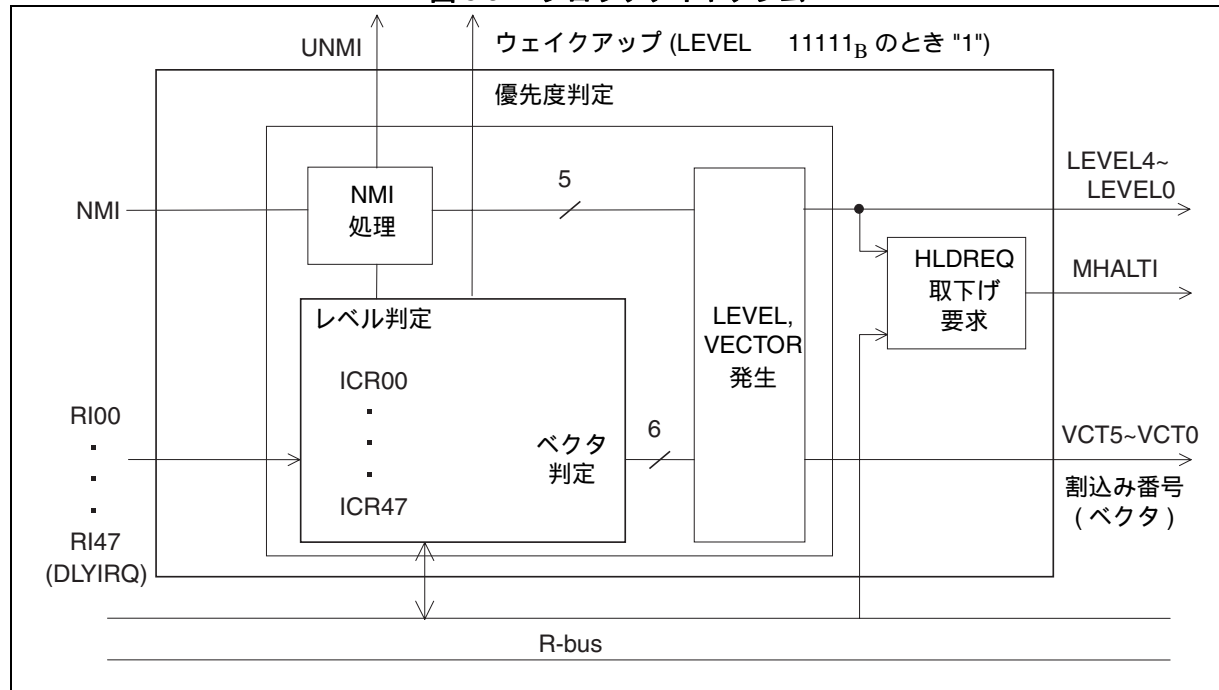
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000460 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR32
000461 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR33
000462 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR34
000463 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR35
000464 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR36
000465 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR37
000466 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38
000467 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR39
000468 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR40
000469 _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR41
00046A _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR42
00046B _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR43
00046C _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR44
00046D _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR45
00046E _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR46
00046F _H	-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	ICR47
				R	R/W	R/W	R/W	R/W	
000045 _H	MHALTI	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	HRCL
	R/W			R	R/W	R/W	R/W	R/W	

5.3 ブロックダイアグラム

図 5.3-1 に、割り込みコントローラのブロックダイアグラムを示します。

■ ブロックダイアグラム

図 5.3-1 ブロックダイアグラム



5.4 レジスタ詳細説明

割り込みコントローラで使用するレジスタの詳細について説明します。

■ Interrupt Control Register (ICR)

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
-	-	-	ICR4	ICR3	ICR2	ICR1	ICR0	---1111 _B
			R	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能
 R : リードオンリ
 - : 未定義

割り込み制御レジスタです。各割り込み入力に対して1つずつ設けられており、対応する割り込み要求の割り込みレベルを設定します。

[bit4 ~ bit0] ICR4 ~ ICR0

割り込みレベル設定ビットで、対応する割り込み要求の割り込みレベルを指定します。

本レジスタに設定した割り込みレベルがCPUのILMレジスタに設定されたレベルマスク値以上の場合は、CPU側にて割り込み要求はマスクされます。

リセットにより、"1111_B"に初期化されます。

設定可能な割り込みレベル設定ビットと割り込みレベルの対応を表5.4-1に示します。

表 5.4-1 設定可能な割り込みレベル設定ビットと割り込みレベルの対応

ICR4	ICR3	ICR2	ICR1	ICR0	割り込みレベル	
0	0	0	0	0	0	システム予約
0	1	1	1	0	14	
0	1	1	1	1	15	
1	0	0	0	0	16	設定可能な最強レベル
1	0	0	0	1	17	(強)  (弱)
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	
1	1	1	1	1	31	割り込み禁止

ICR4は"1"固定で、"0"を書き込むことはできません。

■ Hold Request Cancel Level register (HRCL)

HRCL									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000045 _H	MHALTI	-	-	LVL4	LVL3	LVL2	LVL1	LVL0	0--11111 _B
	R/W			R	R/W	R/W	R/W	R/W	
R/W : リード/ライト可能									
R : リードオンリ									
- : 未定義									

ホールドリクエスト取下げ要求発生のためのレベル設定レジスタです。

[bit7] MHALTI

MHALTI は、NMI の要求による DMA 転送抑止ビットです。NMI 要求によって "1" にセットされ、"0" を書き込むことによってクリアされます。NMI ルーチンの最後で通常の割込みルーチンと同様にクリアしてください。

[bit4 ~ bit0] LVL4 ~ LVL0

バスマスタへのホールドリクエスト取下げ要求を出すための割込みレベルを設定します。

本レジスタに設定した割込みレベルより強いレベルを持つ割込み要求が発生した場合は、バスマスタに対してホールドリクエスト取下げ要求を出します。

LVL4 ビットは "1" 固定で、"0" を書き込むことはできません。

5.5 割込みコントローラの動作

割込みコントローラの動作を説明します。

- 優先順位判定
- NMI
- ホールドリクエストの取下げ要求
- スタンバイモード (ストップ / スリープ) からの復帰

■ 優先順位判定

割込みコントローラでは、同時に発生している割込み要因の中で最も優先度の高い要因を選択し、その要因の割込みレベルと割込み番号を CPU へ出力します。

割込み要因の優先順位判定基準は次のとおりです。

1. NMI
2. 以下の条件を満たす要因
 - 割込みレベルの数値が 31 以外 (31 は割込み禁止)
 - 割込みレベルの数値が最も小さい要因
 - その中で、最も小さい割込み番号を持つ要因

上記の判定基準により割込み要因が 1 つも選択されなかった場合は、割込みレベルとして 31 ("11111_B") を出力します。そのときの割込み番号は不定です。

割込み要因と割込み番号、割込みレベルの関係を表 5.5-1 に示します。

表 5.5-1 割込み要因と割込み番号、割込みレベルの関係 (1 / 4)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
リセット	0	00	-	3FC _H	000FFFC _H
モードベクタ	1	01	-	3F8 _H	000FFFF8 _H
システム予約	2	02	-	3F4 _H	000FFFF4 _H
システム予約	3	03	-	3F0 _H	000FFFF0 _H
システム予約	4	04	-	3EC _H	000FFFE4 _H
システム予約	5	05	-	3E8 _H	000FFFE0 _H
システム予約	6	06	-	3E4 _H	000FFFE4 _H
コプロセッサ不在トラップ	7	07	-	3E0 _H	000FFFE0 _H
コプロセッサエラートラップ	8	08	-	3DC _H	000FFFD8 _H
INTE 命令	9	09	-	3D8 _H	000FFFD8 _H
システム予約	10	0A	-	3D4 _H	000FFFD4 _H
システム予約	11	0B	-	3D0 _H	000FFFD0 _H
ステップトレーストラップ	12	0C	-	3CC _H	000FFFC4 _H

表 5.5-1 割込み要因と割込み番号，割込みレベルの関係 (2 / 4)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
NMI 要求 (tool)	13	0D	-	3C8 _H	000FFFC8 _H
未定義命令例外	14	0E	-	3C4 _H	000FFFC4 _H
NMI 要求	15	0F	15(F _H) 固定	3C0 _H	000FFFC0 _H
外部割込み 0	16	10	ICR00	3BC _H	000FFFBC _H
外部割込み 1	17	11	ICR01	3B8 _H	000FFF8 _H
外部割込み 2	18	12	ICR02	3B4 _H	000FFF4 _H
外部割込み 3	19	13	ICR03	3B0 _H	000FFFB0 _H
外部割込み 4	20	14	ICR04	3AC _H	000FFFAC _H
外部割込み 5	21	15	ICR05	3A8 _H	000FFFA8 _H
外部割込み 6/C-CAN ウェイクアップ*	22	16	ICR06	3A4 _H	000FFFA4 _H
外部割込み 7	23	17	ICR07	3A0 _H	000FFFA0 _H
リロードタイマ 0	24	18	ICR08	39C _H	000FFF9C _H
リロードタイマ 1	25	19	ICR09	398 _H	000FFF98 _H
リロードタイマ 2	26	1A	ICR10	394 _H	000FFF94 _H
UART0 (受信完了)	27	1B	ICR11	390 _H	000FFF90 _H
UART0 (送信完了)	28	1C	ICR12	38C _H	000FFF8C _H
DTTI 端子	29	1D	ICR13	388 _H	000FFF88 _H
DMAC0 (終了, エラー)	30	1E	ICR14	384 _H	000FFF84 _H
DMAC1 (終了, エラー)	31	1F	ICR15	380 _H	000FFF80 _H
DMAC2/3/4 (終了, エラー)	32	20	ICR16	37C _H	000FFF7C _H
UART1 (受信完了)	33	21	ICR17	378 _H	000FFF78 _H
UART1 (送信完了)	34	22	ICR18	374 _H	000FFF74 _H
C-CAN*	35	23	ICR19	370 _H	000FFF70 _H
システム予約	36	24	ICR20	36C _H	000FFF6C _H
積和	37	25	ICR21	368 _H	000FFF68 _H
PPG0/PPG1	38	26	ICR22	364 _H	000FFF64 _H
PPG2/PPG3	39	27	ICR23	360 _H	000FFF60 _H
PPG4/PPG5/PPG6/PPG7	40	28	ICR24	35C _H	000FFF5C _H
システム予約	41	29	ICR25	358 _H	000FFF58 _H
波形 (アンドフロー) 0/1/2	42	2A	ICR26	354 _H	000FFF54 _H

* : C-CAN の割込みは，MB91F267NA/MB91267NA に搭載されている機能です。

表 5.5-1 割込み要因と割込み番号，割込みレベルの関係 (3 / 4)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
フリーランタイム 1(コンペアクリア)	43	2B	ICR27	350 _H	000FFF50 _H
フリーランタイム 1 (0 検出)	44	2C	ICR28	34C _H	000FFF4C _H
フリーランタイム 2(コンペアクリア)	45	2D	ICR29	348 _H	000FFF48 _H
フリーランタイム 2 (0 検出)	46	2E	ICR30	344 _H	000FFF44 _H
タイムベースタイムオーバーフロー	47	2F	ICR31	340 _H	000FFF40 _H
フリーランタイム 0(コンペアクリア)	48	30	ICR32	33C _H	000FFF3C _H
フリーランタイム 0 (0 検出)	49	31	ICR33	338 _H	000FFF38 _H
システム予約	50	32	ICR34	334 _H	000FFF34 _H
A/D コンバータ 1	51	33	ICR35	330 _H	000FFF30 _H
A/D コンバータ 2	52	34	ICR36	32C _H	000FFF2C _H
PWC0 (測定完了)	53	35	ICR37	328 _H	000FFF28 _H
システム予約	54	36	ICR38	324 _H	000FFF24 _H
PWC0 (オーバフロー)	55	37	ICR39	320 _H	000FFF20 _H
システム予約	56	38	ICR40	31C _H	000FFF1C _H
ICU 0 (取込み)	57	39	ICR41	318 _H	000FFF18 _H
ICU 1 (取込み)	58	3A	ICR42	314 _H	000FFF14 _H
ICU2/ICU3 (取込み)	59	3B	ICR43	310 _H	000FFF10 _H
OCU0/OCU1 (一致)	60	3C	ICR44	30C _H	000FFF0C _H
OCU2/OCU3 (一致)	61	3D	ICR45	308 _H	000FFF08 _H
OCU4/OCU5 (一致)	62	3E	ICR46	304 _H	000FFF04 _H
遅延割込み要因ビット	63	3F	ICR47	300 _H	000FFF00 _H
システム予約 (REALOS にて使用)	64	40	-	2FC _H	000FFEFC _H
システム予約 (REALOS にて使用)	65	41	-	2F8 _H	000FFE8 _H
システム予約	66	42	-	2F4 _H	000FFE4 _H
システム予約	67	43	-	2F0 _H	000FFE0 _H
システム予約	68	44	-	2EC _H	000FEEC _H
システム予約	69	45	-	2E8 _H	000FEE8 _H
システム予約	70	46	-	2E4 _H	000FEE4 _H
システム予約	71	47	-	2E0 _H	000FEE0 _H
システム予約	72	48	-	2DC _H	000FEDC _H
システム予約	73	49	-	2D8 _H	000FED8 _H

表 5.5-1 割り込み要因と割り込み番号，割り込みレベルの関係 (4 / 4)

割り込み要因	割り込み番号		割り込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
システム予約	74	4A	-	2D4 _H	000FFED4 _H
システム予約	75	4B	-	2D0 _H	000FFED0 _H
システム予約	76	4C	-	2CC _H	000FFECC _H
システム予約	77	4D	-	2C8 _H	000FFEC8 _H
システム予約	78	4E	-	2C4 _H	000FFEC4 _H
システム予約	79	4F	-	2C0 _H	000FFEC0 _H
INT 命令で使用	80 ~ 255	50 ~ FF	-	2BC _H ~ 000 _H	000FFEBC _H ~ 000FFC00 _H

■ NMI (Non Maskable Interrupt)

NMI は，本モジュールが取り扱う割り込み要因の中では最も優先順位が高くなっています。

そのため，ほかの割り込み要因と同時発生した場合は常に NMI が選択されます。

- NMI が発生すると，CPU に対して次の情報を伝えます。

割り込みレベル : 15 ("01111_B")

割り込み番号 : 15 ("0001111_B")

- NMI 検出

NMI の設定および検出は外部割り込み /NMI モジュールで行います。本モジュールでは NMI 要求により，割り込みレベル / 割り込み番号，MHALTI の生成のみを行います。

- NMI による DMA 転送の抑止

NMI 要求が発生すると，HRCL レジスタの MHALTI ビットが "1" になり，DMA 転送が抑止されます。DMA 転送の抑止を解除したい場合は，NMI ルーチンの最後にて MHALTI ビットを "0" にクリアしてください。

■ ホールドリクエスト取下げ要求 (Hold Request Cancel Request)

優先度の高い割込み処理を、CPU のホールド中 (DMA 転送中) に行いたい場合は、ホールドリクエスト発生元においてリクエストを取り下げてもら必要があります。この取下げ要求発生基準となる割込みレベルを HRCL レジスタに設定します。

● 発生基準

HRCL レジスタに設定した割込みレベルより強いレベルの割込み要因が発生した場合は、DMAC に対して、ホールドリクエスト取下げ要求を発生します。

HRCL レジスタの割込みレベル > 優先度判定後の割込みレベル 取下げ要求発生

HRCL レジスタの割込みレベル ≤ 優先度判定後の割込みレベル 取下げ要求なし

取下げ要求発生原因となった割込み要因をクリアしない限り、この取下げ要求は有効であり、結果的にいつまでも DMA 転送が起こらないことになります。必ず対応する割込み要因をクリアしてください。

また、NMI を使用したときは、HRCL レジスタの MHALTI ビットが "1" となっているため、取下げ要求が有効になっています。

● 設定可能なレベル

HRCL レジスタに設定可能な値は、ICR と同様に "10000_B" から "11111_B" までです。

"11111_B" に設定した場合はすべての割込みレベルに対して取下げ要求を発生し、また、"10000_B" に設定した場合は NMI のみ取下げ要求を発生します。

ホールドリクエスト取下げ要求発生となる割込みレベルの設定を表 5.5-2 に示します。

表 5.5-2 ホールドリクエスト取下げ要求発生となる割込みレベルの設定

HRCL レジスタ	取下げ要求発生となる割込みレベル
16	NMI のみ
17	NMI, 割込みレベル 16
18	NMI, 割込みレベル 16, 17
:	:
31	NMI, 割込みレベル 16 ~ 30 [初期値]

リセット後は、すべての割込みレベルに対して DMA 転送を抑止します。すなわち、割込みが発生していると DMA 転送が行われませんので、HRCL レジスタの値を必要な値に設定してください。

■ スタンバイモード (ストップ/スリープ) からの復帰

割込み要求の発生により、ストップモードから復帰する機能を本モジュールで実現します。NMI を含む、周辺からの割込み要求 (割込みレベルが "11111_B" 以外) が 1 つでも発生すると、クロック制御部に対してストップモードからの復帰要求を発生します。

優先度判定部はストップ復帰後、クロックが供給されてから動作を再開しますので、優先度判定部の結果が出るまでの間、CPU は命令を実行します。

スリープ状態からの復帰においても、同様に動作します。

また、スリープ中であっても、本モジュール内のレジスタはアクセス可能です。

< 注意事項 >

- NMI 要求においてもストップモードからの復帰を行います。ただし、ストップ時に有効な入力レベルを NMI 端子に与えるようにしてください。
- ストップおよびスリープからの復帰の要因としない割込み要因は、対応する周辺の制御レジスタにて割込みレベルを "11111_B" にしてください。

■ ホールドリクエスト取下げ要求機能 (HRCR) の使用例

DMA 転送中に、CPU が優先度の高い処理を行いたいときは、DMA に対してホールドリクエストを取り下げてもらってホールド状態を解除する必要があります。ここでは、割込みを利用して DMA に対してホールドリクエストの取下げ、すなわち、CPU の優先動作を実現します。

● 制御レジスタ

1. HRCL (ホールドリクエストキャンセルレベル設定レジスタ) : 本モジュール

本レジスタに設定した割込みレベルより強いレベルの割込みが発生した場合に、DMA に対してホールドリクエスト取下げ要求を発生します。その基準となるレベルを設定します。

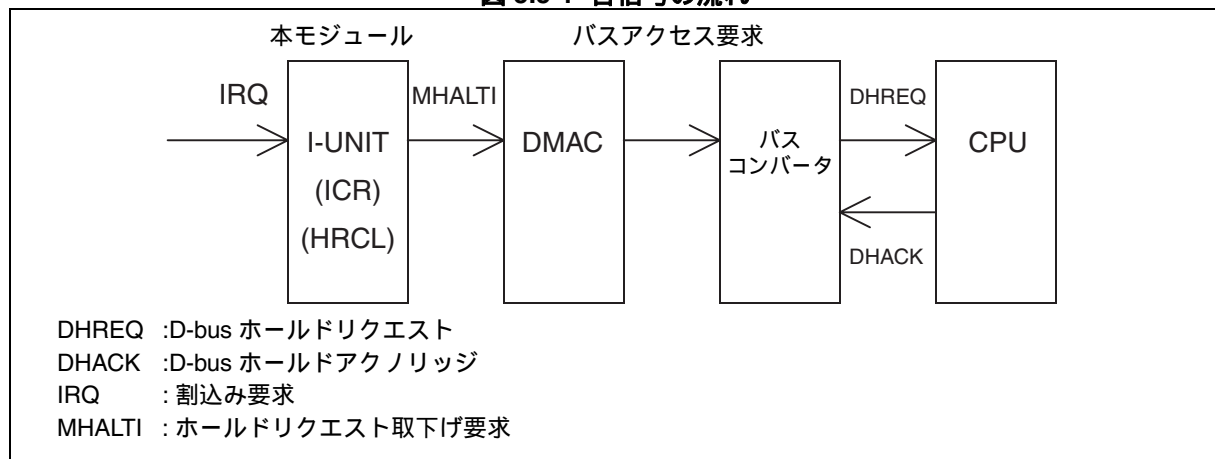
2. ICR : 本モジュール

使用する割込み要因に対応する ICR に、HRCL レジスタよりも強いレベルを設定します。

● ハードウェア構成

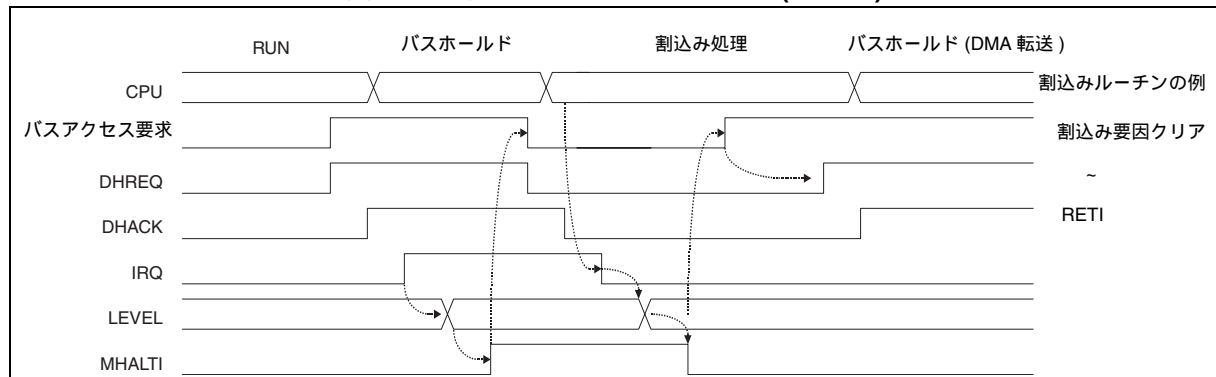
各信号の流れは、次のようになっています。

図 5.5-1 各信号の流れ



● シーケンス

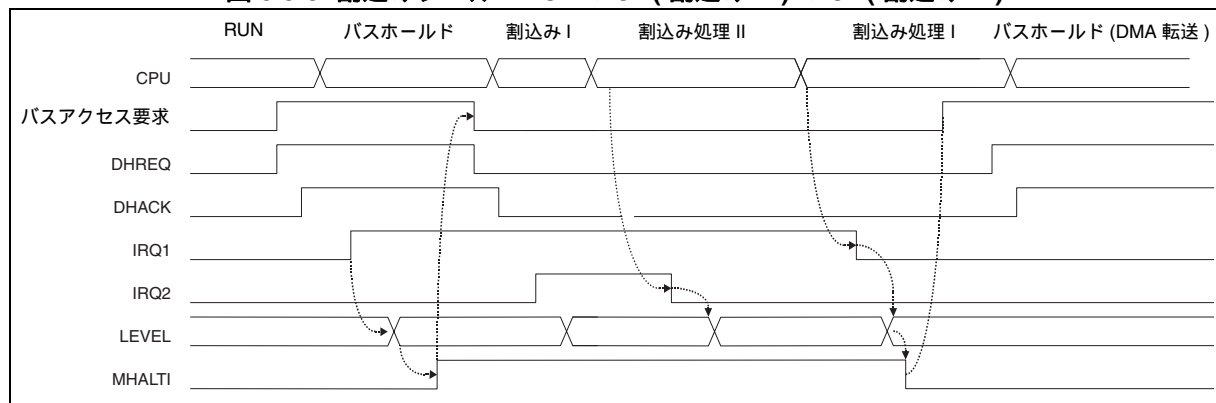
図 5.5-2 割り込みレベル $HRCL < ICR (LEVEL)$



割り込み要求が発生すると割り込みレベルが変化して、これが HRCL レジスタに設定したレベルより強いと DMA に対して MHALTI をアクティブにします。これによって DMA はアクセス要求を下げ、CPU はホールド状態から復帰して割り込み処理を行います。

多重割り込みの場合を以下に示します。

図 5.5-3 割り込みレベル $HRCL < ICR(割り込み I) < ICR(割り込み II)$



割り込みルーチンの例

， ：割り込み要因クリア

， ： RETI

上記例では、割り込みルーチン I を実行中にそれよりも優先度の高い割り込みが発生した場合を示しています。

HRCL レジスタに設定した割り込みレベルより高い割り込みレベルが発生している間は、DHREQ は "L" レベルになっています。

< 注意事項 >

HRCL レジスタと ICR に設定する割り込みレベルの関係には、十分注意してください。

第6章

外部割込み・NMI 制御部

外部割込み /NMI 制御部の概要，レジスタの構成 / 機能および外部割込み /NMI 制御部の動作について説明します。

6.1 外部割込み・NMI 制御部

6.1 外部割込み・NMI 制御部

外部割込み制御部は、 $\overline{\text{NMI}}$ および INT0 ~ INT7 に入力される外部割込み要求の制御を行うブロックです。

外部割込み入力は、検出する要求のレベルとして、"H", "L", "立上リエッジ", "立下リエッジ" から選択できます。

■ レジスター一覧

EIRR0

アドレス

000040_H

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初期値
00000000_B

ENIR0

アドレス

000041_H

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初期値
00000000_B

ELVR0

アドレス

000042_H

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初期値
00000000_B

ELVR0

アドレス

000043_H

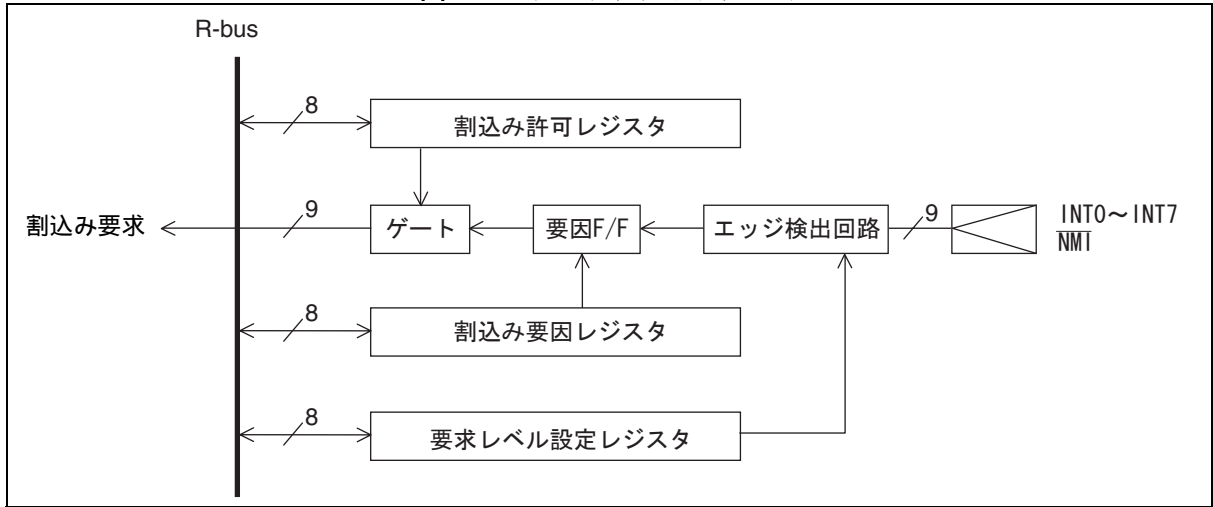
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

初期値
00000000_B

R/W : リード / ライト可能

■ ブロックダイアグラム

図 6.1-1 ブロックダイアグラム



■ レジスタ詳細説明

● 割り込み許可レジスタ (ENIR: ENable Interrupt request Register)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000041 _H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

ENIR は外部割込み要求出力のマスク制御を行います。このレジスタの "1" を書かれたビットに対応する割込み要求出力は許可され (INT0 の許可を EN0 が制御), 割込みコントローラに対して要求が出力されます。"0" が書かれたビットの対応する端子は割込み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。
NMI に対する許可ビットは存在しません。

● 外部割込み要因レジスタ (EIRR: External Interrupt Request Register)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000040 _H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

EIRR は読み出したときには対応する外部割込み要求があることを示し, 書き込み時にはこの要求を示すフリップフロップの内容をクリアするレジスタです。
この EIRR レジスタを読み出したときに "1" であった場合, そのビットに対応する端子に外部割込み要求があることを示します。
またこのレジスタに "0" を書き込むと, 対応するビットの要求フリップフロップがクリアされます。
"1" の書き込みは無効です。
リードモディファイライト (RMW) 系命令の読出し時には "1" が読み出されます。
NMI に対するフラグは, ユーザからアクセスできません。

● 外部割込み要求レベル設定レジスタ (ELVR: External LeVel Register)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000042 _H	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000043 _H	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード/ライト可能									

ELVR は要求検出の選択を行うレジスタです。INT0 ~ INT7 に 2 ビットずつが割り当てられていて、以下の設定になります。要求入力レベルの場合、EIRR の各ビットをクリアしても入力アクティブレベルならば該当するビットは再びセットされます。

表 6.1-1 ELVR 割当て表

LBx	LAx	動 作
0	0	"L" レベルで要求あり
0	1	"H" レベルで要求あり
1	0	立上りエッジで要求あり
1	1	立下りエッジで要求あり

NMI の検出レベルは、常に立下りエッジです。

また、ストップ状態からの復帰に NMI を使用する場合は、"L" レベル検出となります。

< 注意事項 >

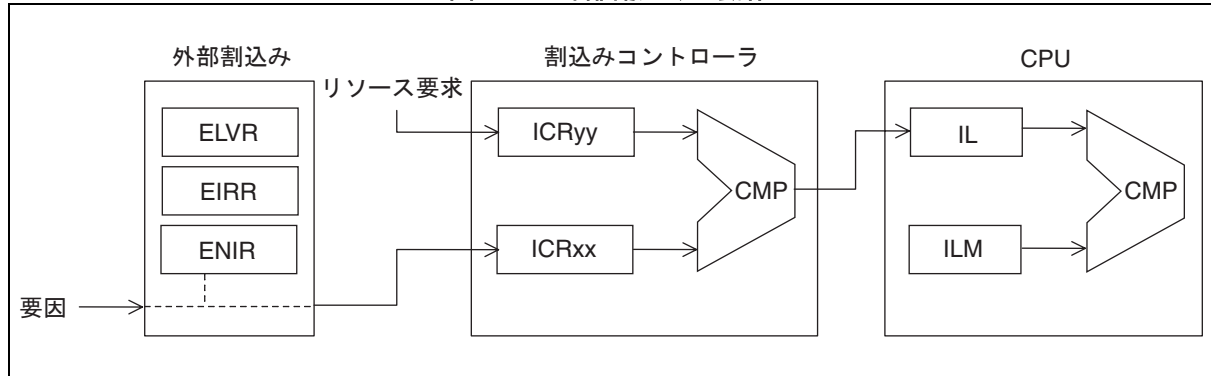
外部割込み要求レベルを変更すると、内部で割込み要因が発生する場合がありますので、外部割込み要求レベル変更後に外部割込み要因レジスタ (EIRR) をクリアしてください。外部割込み要因レジスタをクリアする際はいったん、外部割込み要求レベルレジスタを読み出してからクリアの書き込みを行ってください。

■ 動作説明

● 外部割込みの動作について

要求レベル，許可レジスタの設定の後，対応する端子に ELVR レジスタで設定された要求が入力されると本モジュールは，割込みコントローラに対して割込み要求信号を発生します。割込みコントローラ内で同時発生した割込みの優先順位を識別した結果，本リソースからの割込みが最も優先順位が高かったときに，該当する割込みが発生します。

図 6.1-2 外部割込みの動作



● スタンバイからの復帰について

使用しないチャンネルは，スタンバイに入る前に，必ず禁止状態にしてください。

● 外部割込みの動作手順について

外部割込み部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. 外部割込み入力として使用する端子と兼用する汎用入出力ポートを入力ポートに設定する。
2. 割込み許可レジスタ (ENIR) の対象となるビットを禁止状態にする。
3. 外部割込み要求レベル設定レジスタ (ELVR) の対象となるビットを設定する。
4. 外部割込み要求レベル設定レジスタ (ELVR) を読み出す。
5. 外部割込み要因レジスタ (EIRR) の対象となるビットをクリアする。
6. 割込み許可レジスタ (ENIR) の対象となるビットを許可状態にする。

(ただし，5. と 6. は 16 ビットデータによる同時書込み可)

本モジュール内のレジスタを設定するときには必ず許可レジスタを禁止状態に設定しておかなくてはなりません。また，許可レジスタを許可状態にする前に必ず要因レジスタをクリアしておく必要があります。これは，レジスタ設定時や割込み許可状態時に誤って割込み要因が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について

- 要求レベルがエッジ要求のとき，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（周辺系クロックマシンサイクル）必要とします。
- 要求入力レベルがレベル設定の場合，パルス幅は最小 3 マシンサイクル必要とします。また割込み入力端子がアクティブレベルを保持し続けている限りは，外部割込み要因レジスタをクリアしても，割込みコントローラへの割込み要求は発生し続けます。
- 要求入力レベルがレベル設定のとき，外部より要求入力が入ってその後取り下げられても内部に要因保持回路が存在するので，割込みコントローラへの要求はアクティブのままです。
割込みコントローラへの要求を取り下げるには外部割込み要因レジスタをクリアする必要があります。

図 6.1-3 レベル設定時の外部割込み要因レジスタのクリア

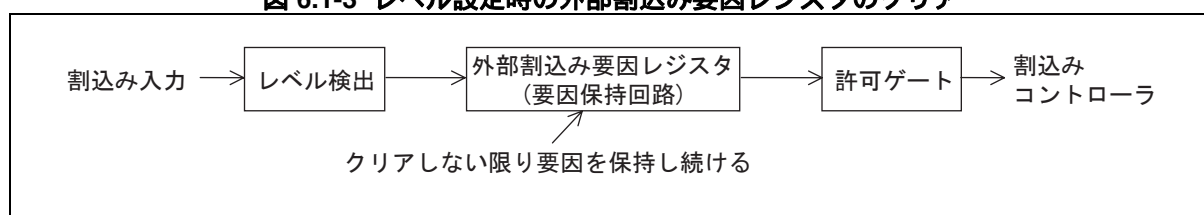
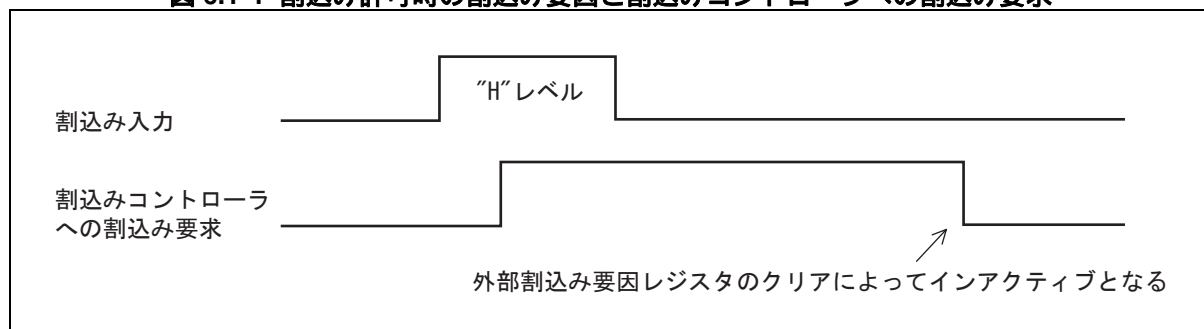


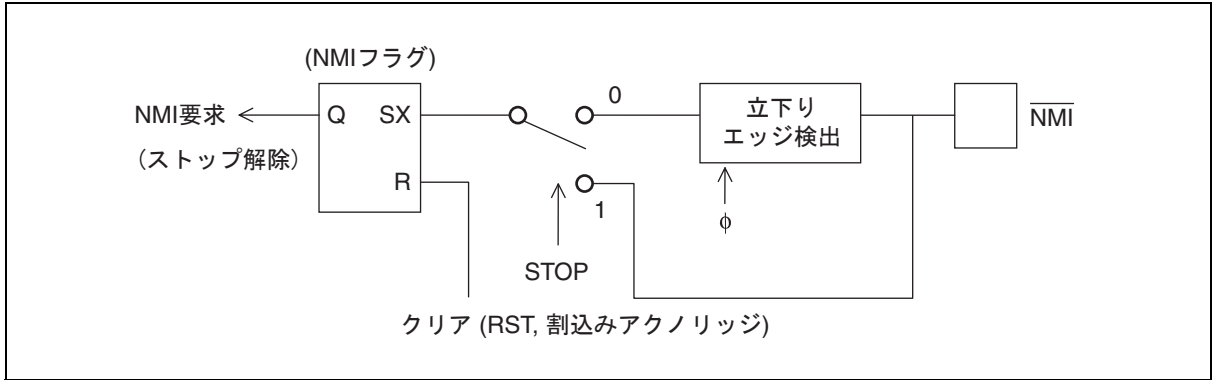
図 6.1-4 割込み許可時の割込み要因と割込みコントローラへの割込み要求



● NMI

- NMI は，ユーザ割込みの中で最強の割込みで，マスクすることは不可能です。
例外として，NMI を使用前に ILM を設定しないで NMI を起動すると，NMI 要因は検出されますが，CPU は NMI 要求を受け付けません。このとき，ILM を NMI が受け付けられるレベルに設定されるまで，NMI 要因は保持され続けます。このため，リセット後，ILM を 16 以上の値に設定してから，NMI を使用してください。
また，NMI の内部要因フラグは，CPU からアクセスできませんので，リセット後の NMI 端子は "H" レベルを保持してください。
- NMI の受け付けは下記のとおりです。
通常時 : 立下りエッジ
ストップ時 : "L" レベル
- NMI によってストップモードの解除が可能です。ストップ状態で "L" レベルが入力されるとストップ状態が解除され発振安定待ち時間がとられます。
NMI 要求検出部は NMI フラグがあり，NMI 要求によりセットされ，NMI 自身の割込みの受け付け，もしくは，リセットでのみクリアされます。なお，このビットは読出し / 書込みができません。

図 6.1-5 NMI 要求検出

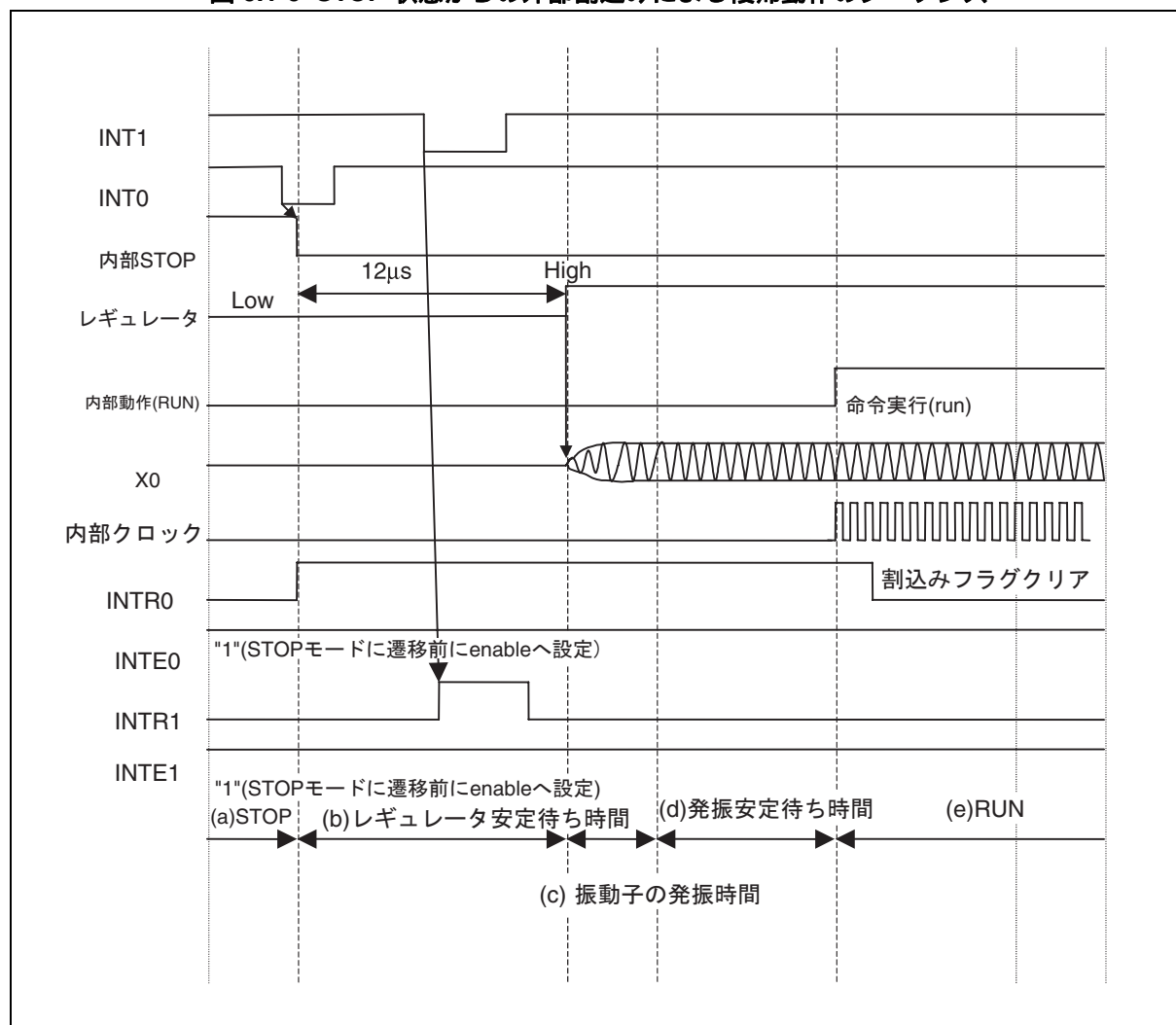


■ 外部割込みを使用した STOP 状態からの復帰時における注意事項

STOP 状態時に、INT 端子への最初に入力された外部割込み信号は非同期で入力され、STOP 状態から復帰することが可能です。ただし、その STOP 解除から、発振安定待ち時間経過するまでの期間においては、ほかの外部割込み信号の入力を認識できない期間が存在します（図 6.1-6 の b+c+d 期間）。STOP 解除後の外部入力信号を内部クロックに同期させるため、クロックが安定していない期間内は、その割込み要因を保持できないためです。

STOP 解除後の外部割込み入力を行う場合には、発振安定待ち時間経過後に外部割込み信号を入力してください。

図 6.1-6 STOP 状態からの外部割込みによる復帰動作のシーケンス



■ STOP 状態からの復帰動作について

現行回路の外部割込みによる STOP 復帰動作は、以下の動作によって行われます。

● STOP 遷移前の処理

外部割込みの入力

STOP 状態から復帰させたいとき、外部割込み信号は非同期で入力信号を伝える状態となっています。この割込み信号が有効になると直ちに内部 STOP 信号を立ち下げる動作が行われます。同時に外部割込み回路ではほかのレベル割込み入力の同期化を行うように切り換わります。

● レギュレータ安定待ち時間

内部 STOP 信号が立ち下ると STOP 時のレギュレータから RUN 時のレギュレータへ切換え動作が開始されます。RUN 時のレギュレータの電圧出力が安定する前に内部動作が開始されると動作が不安定になるため、内部出力電圧の安定待ち時間を確保しています。この期間はクロックが停止しています。

● 振動子の発振時間

レギュレータ安定待ち時間が終了後、クロックの発振が開始されます。振動子の発振時間は使用される振動子により異なります。

● 発振安定待ち時間

振動子の発振時間後にデバイス内部で発振安定待ち時間がとられます。発振安定待ち時間はスタンバイ制御レジスタの OS1, OS0 ビットにより指定します。発振安定待ち時間終了後、内部クロックが供給され、外部割込みによる割込み命令動作が開始されるとともに、STOP からの復帰要因以外の外部割込み要因が受け付け可能になります。

第7章

REALOS 関連ハード

REALOS 関連ハードは、リアルタイム OS により使用されます。したがって、REALOS を使用する場合にはユーザプログラムで使用することはできません。

遅延割込みモジュールおよびビットサーチモジュールの概要、レジスタ構成 / 機能および動作について説明します。

7.1 遅延割込みモジュール

7.2 ビットサーチモジュール

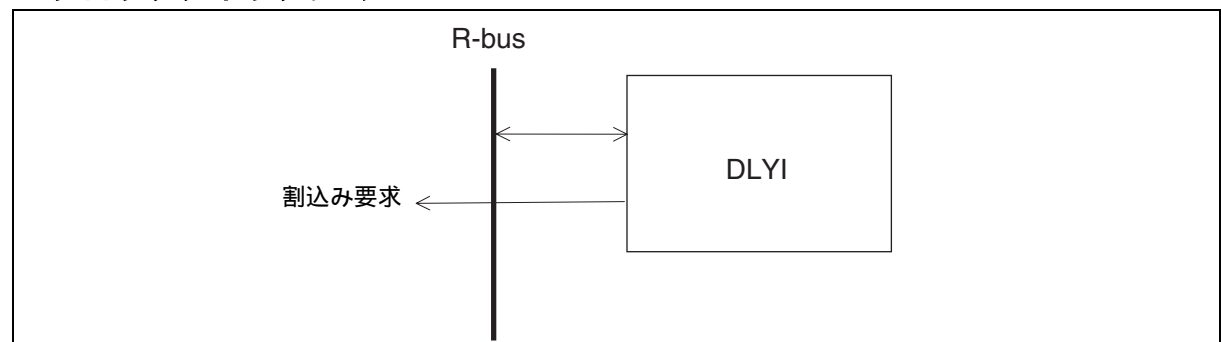
7.1 遅延割込みモジュール

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

■ レジスタ一覧

DICR									初期値
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
000044 _H	-	-	-	-	-	-	-	DLYI	-----0 _B
R/W									
R/W : リード / ライト可能									
- : 未定義									

■ ブロックダイアグラム



■ レジスタ詳細説明

● DICR (Delayed Interrupt Control Register)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000044 _H	-	-	-	-	-	-	-	DLYI	-----0 _B
R/W									
R/W : リード / ライト可能									
- : 未定義									

遅延割込みを制御するレジスタです。

[bit0] DLYI

DLYI	説明
0	遅延割込み要因の解除・要求なし [初期値]
1	遅延割込み要因の発生

本ビットにより、該当する割込み要因の発生・解除を制御します。

■ 動作説明

遅延割込みは、タスク切換え用の割込みを発生します。本機能を使用することにより、ソフトウェアで CPU に対して割込み要求の発生、取消しを行うことができます。

● 割込み番号

遅延割込みは、最も大きな割込み番号に対応した割込み要因に割り当てられています。

本品種では、遅延割込みを割込み番号 63 (3F_H) に割り当てています。

● DICR の DLYI ビット

本ビットに "1" を書き込むことで、遅延割込み要因が発生します。また、"0" を書き込むことで、遅延割込み要因を解除します。

本ビットは、一般の割込みにおける割込み要因フラグと同じものであり、割込みルーチンの中で本ビットをクリアし、併せてタスクの切換えを行うようにしてください。

7.2 ビットサーチモジュール

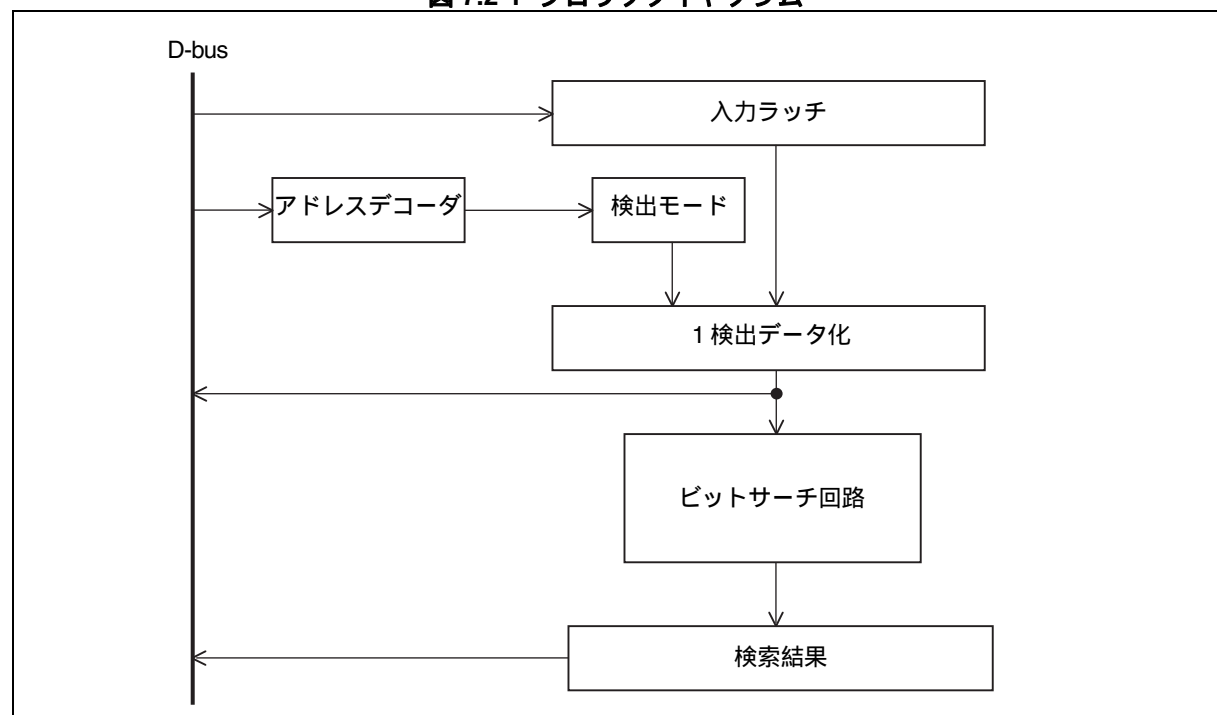
入力レジスタに書き込まれたデータに対して, "0", "1" または変化点を検索し, 検出したビット位置を返します。

■ レジスタ一覧

	bit31	bit0
アドレス : 0003F0 _H	BSD0	0 検出用データレジスタ
アドレス : 0003F4 _H	BSD1	1 検出用データレジスタ
アドレス : 0003F8 _H	BSDC	変化点検出用データレジスタ
アドレス : 0003FC _H	BSRR	検出結果レジスタ

■ ブロックダイアグラム

図 7.2-1 ブロックダイアグラム



■ レジスタ詳細説明

● 0 検出用データレジスタ (BSD0)

アドレス	bit31	bit0
0003F0 _H		
属性	ライトオンリ	
初期値	XXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	

書き込んだ値に対して, 0 検出を行います。

リセットによる初期値は不定です。読出し値は不定です。

データ転送には, 32 ビット長のデータ転送命令を使用してください。

(8 ビット, 16 ビット長のデータ転送命令は使わないでください。)

● 1 検出用データレジスタ (BSD1)

アドレス	bit31	bit0
0003F4 _H		
属性	リード / ライト可能	
初期値	XXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	

データ転送には, 32 ビット長のデータ転送命令を使用してください。

(8 ビット, 16 ビット長のデータ転送命令は使わないでください。)

書込み時

書き込んだ値に対して, "1" を検出します。

読出し時

ビットサーチモジュールの内部状態の退避用データが読み出されます。割込みハンドラなどがビットサーチモジュールを使用する場合に, 元の状態を退避・復帰するときに使用します。

0 検出, 変化点検出, データレジスタにデータを書き込んだ場合でも, 1 検出用データレジスタのみ操作することで退避・復帰できます。

リセットによる初期値は不定です。

● 変化点検出用データレジスタ (BSDC)

アドレス	bit31	bit0
0003F8 _H		
属性	ライトオンリ	
初期値	XXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX	

書き込んだ値に対して, 変化点の検出を行います。

リセットによる初期値は不定です。

読出し値は不定です。

データ転送には, 32 ビット長のデータ転送命令を使用してください。

(8 ビット, 16 ビット長のデータ転送命令は使わないでください。)

● 検出結果レジスタ (BSRR)

0 検出, 1 検出, または変化点検出の結果が読み出されます。

どの検出結果が読み出されるかは, 最後に関与したデータレジスタによって決定されます。

アドレス	bit31	bit0
0003FC _H		
属性	リードオンリ	
初期値	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX	

■ 動作説明

● 0 検出

0 検出用データレジスタに書き込まれたデータを MSB から LSB へスキャンし、最初の "0" を検出した位置を返します。

検出結果は、検出結果レジスタを読み出すことで得られます。

検出した位置と、返す数値の関係は、表 7.2-1 のとおりです。

"0"が存在しないとき(すなわち"FFFFFFFF_H"という数値のとき), "32"という値をサーチ結果として返します。

[実行例]

書込みデータ	読出し値(10進)
11111111111111111111000000000000 _B (FFFFF000 _H)	20
11111000010010011110000010101010 _B (F849E0AA _H)	5
100000000000000101010101010101010 _B (8002AAAA _H)	1
11111111111111111111111111111111 _B (FFFFFFFF _H)	32

● 1 検出

1 検出用データレジスタに書き込まれたデータを MSB から LSB へスキャンし、最初の "1" を検出した位置を返します。

検出結果は、検出結果レジスタを読み出すことで得られます。

検出した位置と返す値との関係は、表 7.2-1 のとおりです。

"1" が存在しないとき (すなわち "00000000_H" という数値のとき), "32" という値をサーチ結果として返します。

[実行例]

書込みデータ	読出し値 (10 進)
00100000000000000000000000000000 _B (20000000 _H)	2
00000001001000110100010101100111 _B (01234567 _H)	7
00000000000000011111111111111111 _B (0003FFFF _H)	14
00000000000000000000000000000001 _B (00000001 _H)	31
00000000000000000000000000000000 _B (00000000 _H)	32

● 変化点検出

変化点検出用データレジスタに書き込まれたデータを bit30 から LSB ヘスキャンし、MSB の値と比較します。最初に MSB と異なる値を検出した位置を返します。

検出結果は、検出結果レジスタを読み出すことで得られます。

検出した位置と返す値は、表 7.2-1 の示すとおりです。

変化点が存在しないときは、"32" を返します。

変化点検出では、結果として "0" を返すことはありません。

[実行例]

書込みデータ	読出し値 (10 進)
00100000000000000000000000000000 _B (20000000 _H)	2
00000001001000110100010101100111 _B (01234567 _H)	7
00000000000000011111111111111111 _B (0003FFFF _H)	14
00000000000000000000000000000001 _B (00000001 _H)	31
00000000000000000000000000000000 _B (00000000 _H)	32
11111111111111111111000000000000 _B (FFFFFF00 _H)	20
11111000010010011110000010101010 _B (F849E0AA _H)	5
10000000000000010101010101010101 _B (8002AAAA _H)	1
11111111111111111111111111111111 _B (FFFFFFFF _H)	32

表 7.2-1 ビット位置と返す値 (10 進)

検出した ビット位置	返す値	検出した ビット位置	返す値	検出した ビット位置	返す値	検出した ビット位置	返す値
31	0	23	8	15	16	7	24
30	1	22	9	14	17	6	25
29	2	21	10	13	18	5	26
28	3	20	11	12	19	4	27
27	4	19	12	11	20	3	28
26	5	18	13	10	21	2	29
25	6	17	14	9	22	1	30
24	7	16	15	8	23	0	31
						存在しない	32

■ 退避・復帰の処理

割込みハンドラ中で、ビットサーチモジュールを使う場合のように、ビットサーチモジュールの内部状態を退避・復帰させる必要がある場合は、以下の手順に従ってください。

1. 検出用データレジスタを読み出し、この内容を保存する（退避）
2. ビットサーチモジュールを使用
3. 1. で退避したデータを 1 検出用データレジスタに書き込む（復帰）

以上の操作により、次に検出結果レジスタを読み出したときに得られる値は、1. 以前に、ビットサーチモジュールに書き込まれた内容に応じたものとなります。最後に書き込まれたデータレジスタが 0 検出用または変化点検出用であっても、上記手順で正しく元に戻ります。

第8章

リロードタイマ

リロードタイマの概要，レジスタの構成 / 機能および動作について説明します。

- 8.1 概要
- 8.2 ブロックダイアグラム
- 8.3 リロードタイマのレジスタ
- 8.4 リロードタイマの動作
- 8.5 注意点

8.1 概要

16 ビットリロードタイマは、16 ビットのダウンカウンタ、16 ビットのリロードレジスタ、内部カウンタ、クロック作成用プリスケアラ、コントロールレジスタで構成されています。

クロックソースとして、CPU クロック、周辺クロックおよび外部バスクロックの3種類の内部クロック（マシンのクロックの 2/8/32/64/128 分周まで選択可能）、外部トリガから選択できます。

MB91265A シリーズでは、本タイマを 3 チャンネル内蔵しています。

リロードタイマ 0 の端子への出力はありません。

■ レジスタ一覧

コントロールステータスレジスタ (TMCSR0 ~ TMCSR3)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 00004E _H	-	-	-	CSL2	CSL1	CSL0	MOD2	MOD1	---00000 _B
ch.1 : 000056 _H	-	-	-	R/W	R/W	R/W	R/W	R/W	
ch.2 : 00005E _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	MOD0	-	OUTL	RELD	INTE	UF	CNTE	TRG	00000000 _B
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	

16 ビットタイマレジスタ (TMR0 ~ TMR3)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 00004A _H									XXXXXXXX _B
ch.1 : 000052 _H	R	R	R	R	R	R	R	R	
ch.2 : 00005A _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
		-							XXXXXXXX _B
	R	R	R	R	R	R	R	R	

16 ビットリロードレジスタ (TMRLR0 ~ TMRLR3)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 000048 _H									XXXXXXXX _B
ch.1 : 000050 _H	W	W	W	W	W	W	W	W	
ch.2 : 000058 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
		-							XXXXXXXX _B
	W	W	W	W	W	W	W	W	

R/W : リード/ライト可能

R : リードオンリ

W : ライトオンリ

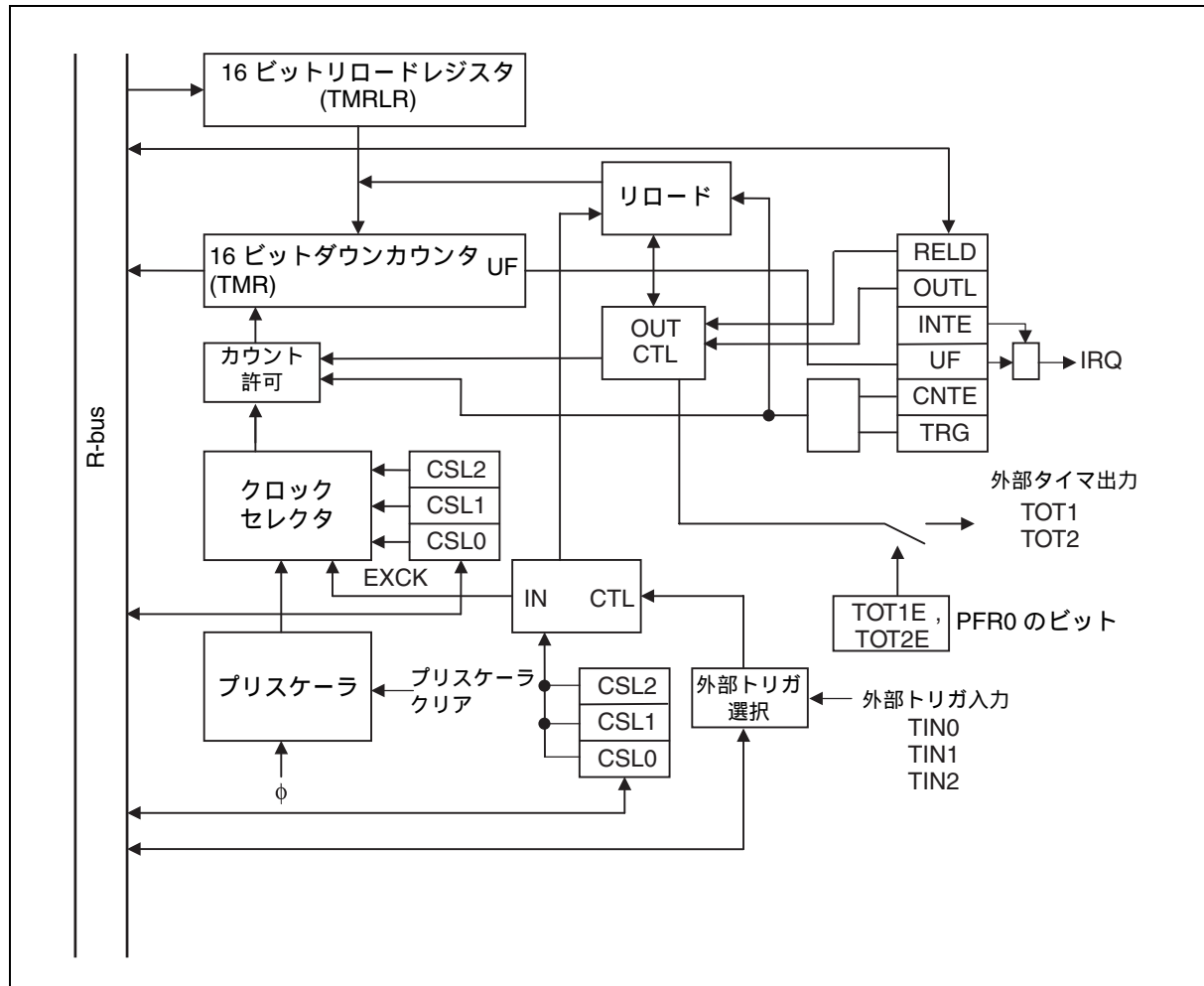
X : 不定

- : 未定義

8.2 ブロックダイアグラム

リロードタイマのブロックダイアグラムを示します。

■ リロードタイマのブロックダイアグラム



8.3 リロードタイマのレジスタ

リロードタイマで使用するレジスタの構成と機能について説明します。

■ コントロールステータスレジスタ (TMCSR: TMCSR0 ~ TMCSR2)

16 ビットタイマの動作モードおよび割込みの制御を行います。

UF, CNTE, TRG ビット以外の書換えは, CNTE=0 のときに行うようにしてください。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 00004E _H	-	-	-	CSL2	CSL1	CSL0	MOD2	MOD1	---00000 _B
ch.1 : 000056 _H	-	-	-	R/W	R/W	R/W	R/W	R/W	
ch.2 : 00005E _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	MOD0	-	OUTL	RELD	INTE	UF	CNTE	TRG	00000000 _B
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能
 R : リードオンリ
 - : 未定義

[bit12 ~ bit10] CSL2, CSL1, CSL0 (Count source select)

カウントソースセレクトビットです。カウントソースは内部クロックと外部トリガを選択できます。選択可能なカウントソースは以下のとおりです。

CSL2	CSL1	CSL0	カウントソース (ϕ マシンクロック)	$\phi=32$ MHz	$\phi=16$ MHz
0	0	0	内部クロック $\phi 2^1$ (初期値)	62.5 ns	125 ns
0	0	1	内部クロック $\phi 2^3$	250 ns	0.5 μ s
0	1	0	内部クロック $\phi 2^5$	1.0 μ s	2.0 μ s
0	1	1	外部トリガ	-	-
1	0	0	設定禁止	-	-
1	0	1	内部クロック $\phi 2^6$	2.0 μ s	4.0 μ s
1	1	0	内部クロック $\phi 2^7$	4.0 μ s	8.0 μ s
1	1	1	ch.1 タイマ出力 (ch.2 のみ設定可能)	ch.1	ch.1

外部トリガをカウントソースに設定した場合のカウント有効エッジは, MOD1, MOD0 ビットにより設定されます。

外部トリガに必要な最小パルス幅は, $2 \times T$ (T: マシンクロックサイクル) です。

CSL2, CSL1, CSL0=111_B は, ch.1 + ch.2 のカスケード接続で, ch.2 のレジスタのみ設定可能です。ch.1 では設定禁止です。

[bit9 ~ bit7] MOD2, MOD1, MOD0 (Mode)

動作モードを選択するビットです。カウントソースが「内部クロック」の場合と「外部トリガ」の場合で機能が変わります。

- ・ 内部クロックモードのとき：リロードトリガ設定
- ・ 外部トリガモードのとき：カウント有効エッジの設定

また、MOD2 は必ず "0" を設定してください。

[内部クロック選択時のリロードトリガ設定]

カウントソースとして、内部クロックが選択されている場合は、MOD2 ~ MOD0ビットの設定により有効エッジが入力されるとリロードレジスタの内容をロードしてカウント動作を継続します。

MOD2	MOD1	MOD0	有効エッジ
0	0	0	ソフトウェアトリガ (初期値)
0	0	1	外部トリガ (立上りエッジ)
0	1	0	外部トリガ (立下りエッジ)
0	1	1	外部トリガ (両エッジ)
1	X	X	設定禁止

[外部トリガ選択時の有効エッジ設定]

カウントソースとして、外部クロックが設定されている場合は、MOD2 ~ MOD0ビットの設定により外部トリガの有効エッジをカウントします。

MOD2	MOD1	MOD0	有効エッジ
X	0	0	-
X	0	1	外部トリガ (立上りエッジ)
X	1	0	外部トリガ (立下りエッジ)
X	1	1	外部トリガ (両エッジ)
X	X	X	設定禁止

外部トリガ選択時のリロードは、ソフトウェアトリガとアンダフローで発生します。

[bit6] 未定義ビット

常に "0" が読み出されます。

[bit5] OUTL

外部タイマ出力レベルを設定するビットです。本ビットが "0" のときと "1" のときでは出力レベルが逆になります。

[bit4] RELD

リロード許可ビットです。"1"のときリロードモードになり、カウンタの値が"0000_H" "FFFF_H" へのアンダフローと同時にリロードレジスタの内容をカウンタへロードしてカウント動作を続けます。

"0" のときワンショットモードになり、カウンタの値が"0000_H" "FFFF_H" へのアンダフローによりカウント動作を停止します。

TOxE	OUTL	RELD	出力波形
0	X	X	出力禁止
1	0	0	カウント中 "H" の矩形波
1	1	0	カウント中 "L" の矩形波
1	0	1	カウントスタート時 "L" のトグル出力
1	1	1	カウントスタート時 "H" のトグル出力

TOxE は PFR0 (Port Function Register) 中の TOT1E, TOT2E を示します。

(注意事項) リロードタイマ 0 の端子への出力はありません。

[bit3] INTE

割込み要求許可ビットです。"1" のとき UF ビットが "1" になると割込み要求を発生します。"0" のときは割込み要求を発生しません。

[bit2] UF

タイマ割込み要求フラグです。カウンタの値が "0000_H" "FFFF_H" へのアンダフローにより "1" にセットされます。"0" の書込み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では "1" が読み出されます。

[bit1] CNTE

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

[bit0] TRG

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" 書込みは意味を持ちません。読出し値は常に "0" です。

本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 のときには何も起こりません。

■ TMR レジスタ (16 ビットタイマレジスタ)

16 ビットタイマのカウント値を読み出すことができるレジスタです。初期値は不定です。本レジスタの読出しは必ず 16 ビットデータ転送命令で行ってください。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 00004A _H									XXXXXXXX _B
ch.1 : 000052 _H	R	R	R	R	R	R	R	R	
ch.2 : 00005A _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
		-							XXXXXXXX _B
	R	R	R	R	R	R	R	R	

R : リードオンリ
 X : 不定
 - : 未定義

■ TMRLR レジスタ (16 ビットリロードレジスタ)

本レジスタは、カウントの初期値を保持しておくためのレジスタです。初期値は不定です。本レジスタの読出しは必ず 16 ビットデータ転送命令で行ってください。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 : 000048 _H									XXXXXXXX _B
ch.1 : 000050 _H	W	W	W	W	W	W	W	W	
ch.2 : 000058 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
		-							XXXXXXXX _B
	W	W	W	W	W	W	W	W	

W : ライトオンリ
 X : 不定
 - : 未定義

8.4 リロードタイマの動作

リロードタイマの内部クロック動作，アンダフロー動作について説明します。

■ 内部クロック動作

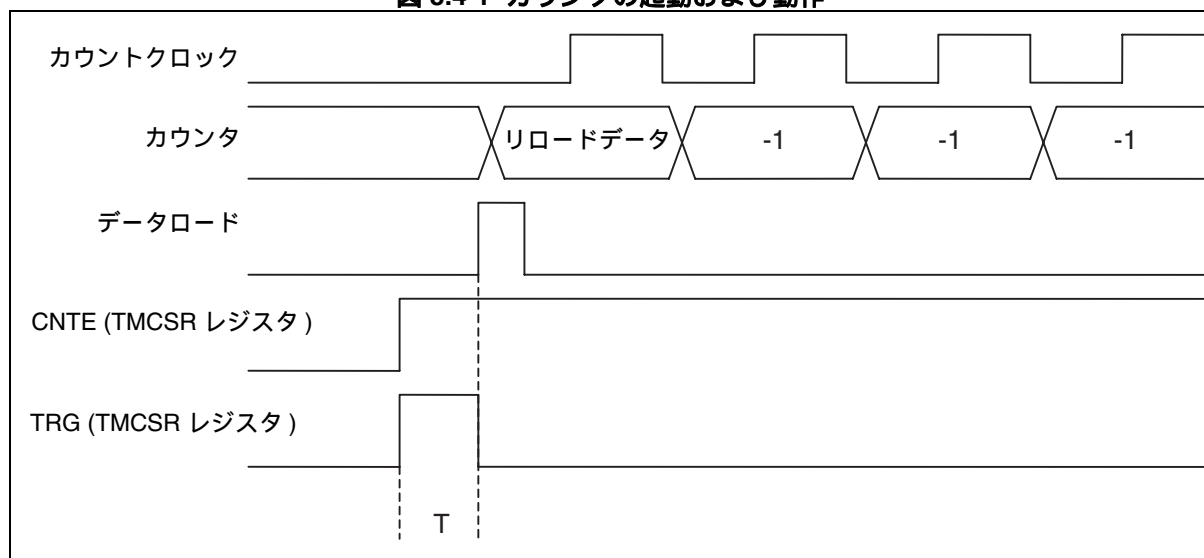
内部クロックの分周クロックでタイマを動作させる場合，クロックソースとしてマシンクロックの 2, 8, 32, 64, 128 分周のクロックから選択することができます。

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，コントロールステータスレジスタの CNTE ビットと TRG ビットの両方に "1" を書き込んでください。TRG ビットによるトリガ入力，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。

カウンタの起動およびカウンタの動作について，図 8.4-1 に示します。

カウント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：周辺系クロックマシンサイクル) の時間がかかります。

図 8.4-1 カウンタの起動および動作



■ アンダフロー動作

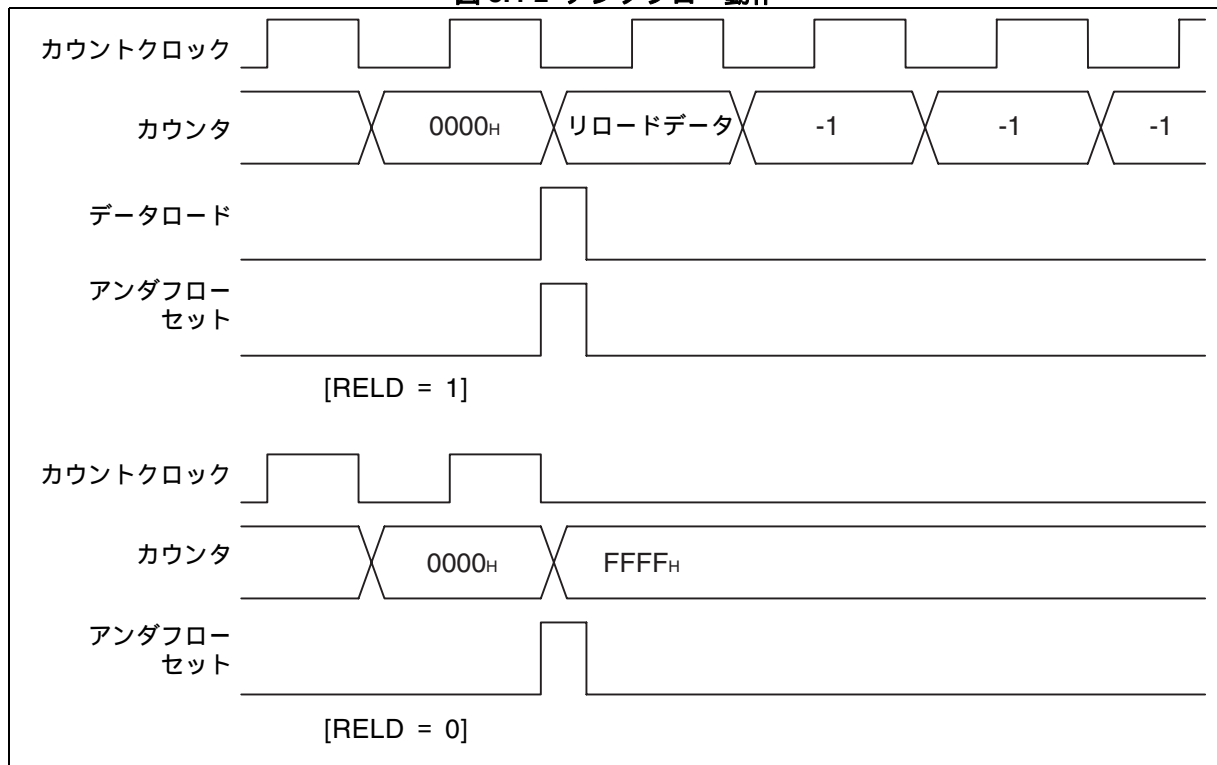
本タイマは、カウンタの値が "0000_H" から "FFFF_H" になるときをアンダフローと定義しています。したがって、[リロードレジスタの設定値 + 1] カウントでアンダフローが発生します。

アンダフロー発生時コントロールステータスレジスタの 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ます。RELD ビットが "0" のとき、カウンタは "FFFF_H" で停止します。

アンダフローが発生すると、コントロールステータスレジスタの UF ビットをセットします。このときに INTE ビットが "1" の場合、割込み要求が発生します。

アンダフロー発生時の動作について図 8.4-2 に示します。

図 8.4-2 アンダフロー動作



■ 出力端子機能

TOT1, TOT2 出力端子は、リロードモード時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はカウント中を示すパルス出力として機能します。出力極性はレジスタの OUTL ビットにより設定できます。OUTL=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=1 にすると出力波形は反転します。

図 8.4-3 出力端子機能 [RELD=1, OUTL=0]

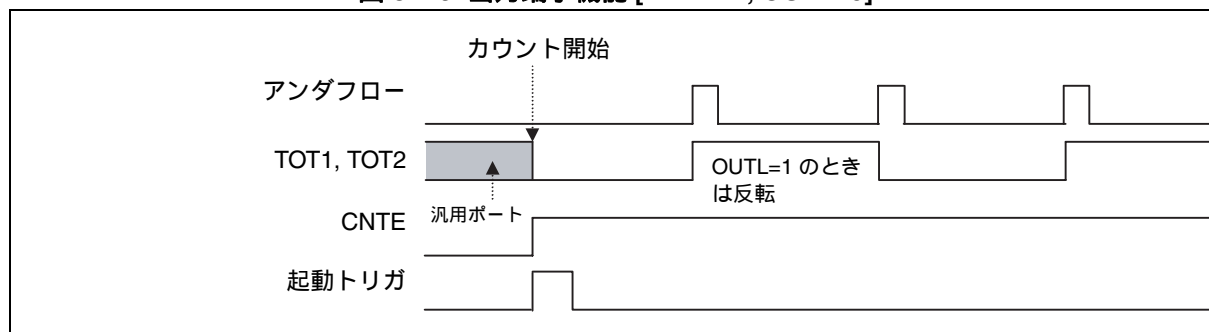
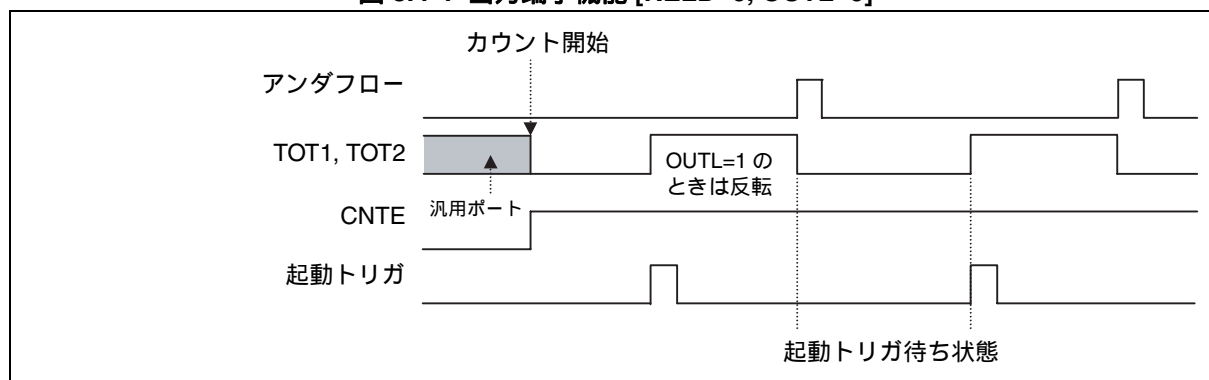


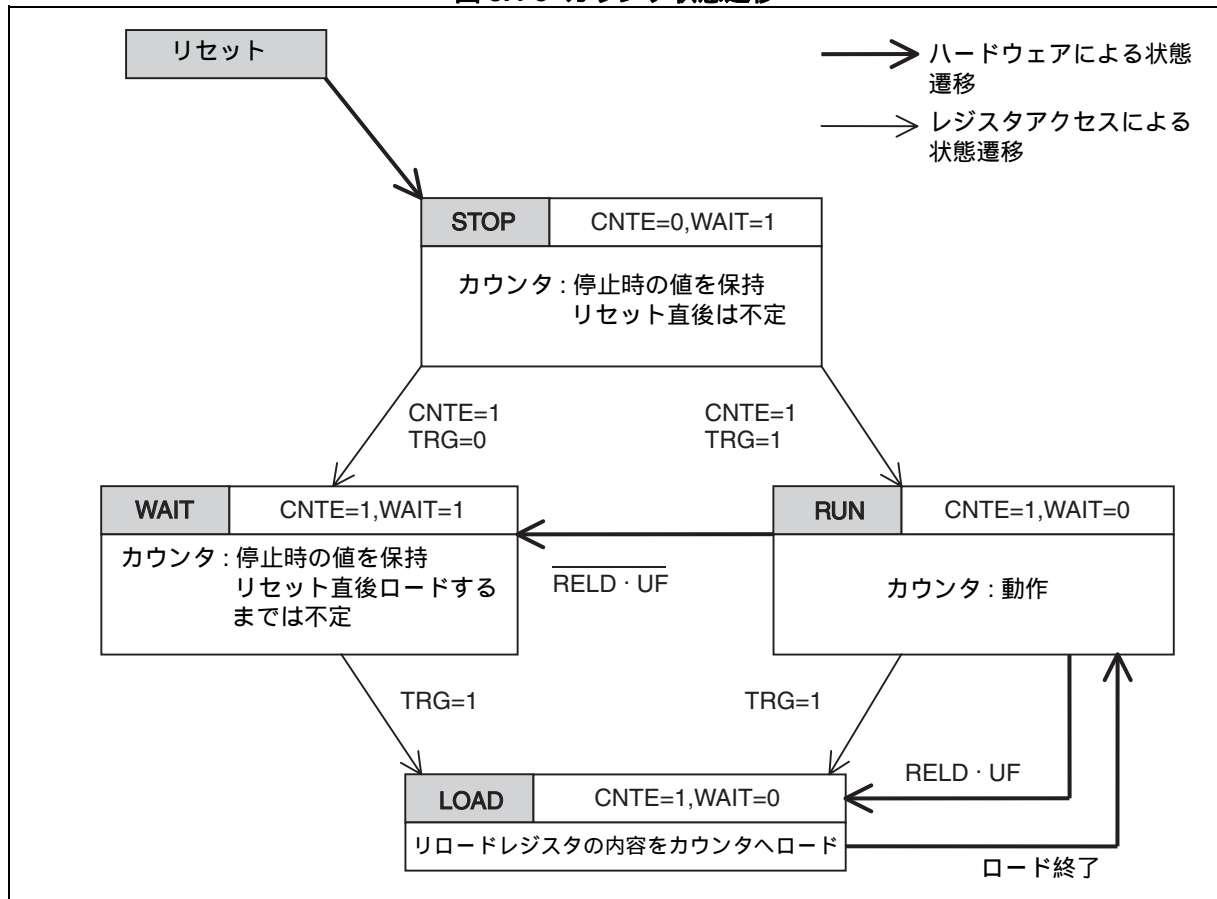
図 8.4-4 出力端子機能 [RELD=0, OUTL=0]



■ 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 信号によって決まっています。設定可能な状態として 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。各状態の遷移を図 8.4-5 に示します。

図 8.4-5 カウンタ状態遷移



8.5 注意点

リロードタイマの使用上の注意点について説明します。

■ 注意点

- 内部プリスケアラは、コントロールステータスレジスタの bit1 (タイマ許可 : CNTE) が "1" に設定されている状態でトリガ (ソフトトリガ , あるいは外部トリガ) がかけられることにより動作可能になります。
- 割込み要求フラグセットタイミングとクリアタイミングが重複した場合には、フラグセットが優先し、クリア動作は無効になります。
- 16 ビットタイマリロードレジスタへの書込みと、リロードのタイミングが重なった場合には、旧データと新データがカウンタにロードされるのは、次のリロードのときになります。
- 16 ビットタイマレジスタのロードとカウントのタイミングが重複した場合には、ロード (リロード) 動作の方が優先されます。

第9章

タイミングジェネレータ

タイミングジェネレータの概要，レジスタの構成 / 機能および動作について説明します。

9.1 タイミングジェネレータ

9.1 タイミングジェネレータ

タイミングジェネレータは、複数の PPG タイマをタイマ間で同期し遅延起動を行わせるための機能です。

■ レジスタ一覧

コントロールレジスタ (TTCR)

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	初期値
000144 _H	TRG60	TRG40	TRG20	TRG00	CS1	CS0	MONI	STR	11110000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

テストレジスタ (TSTPR (書き込み禁止, 読出し値意味なし))

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000147 _H	—	—	—	—	—	—	—	—	00000000 _B
	R	R	R	R	R	R	R	R	

コンペアレジスタ 0 (COMP0)

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	初期値
000148 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

コンペアレジスタ 2 (COMP2)

アドレス	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	初期値
000149 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

コンペアレジスタ 4 (COMP4)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00014A _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

コンペアレジスタ 6 (COMP6)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00014B _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

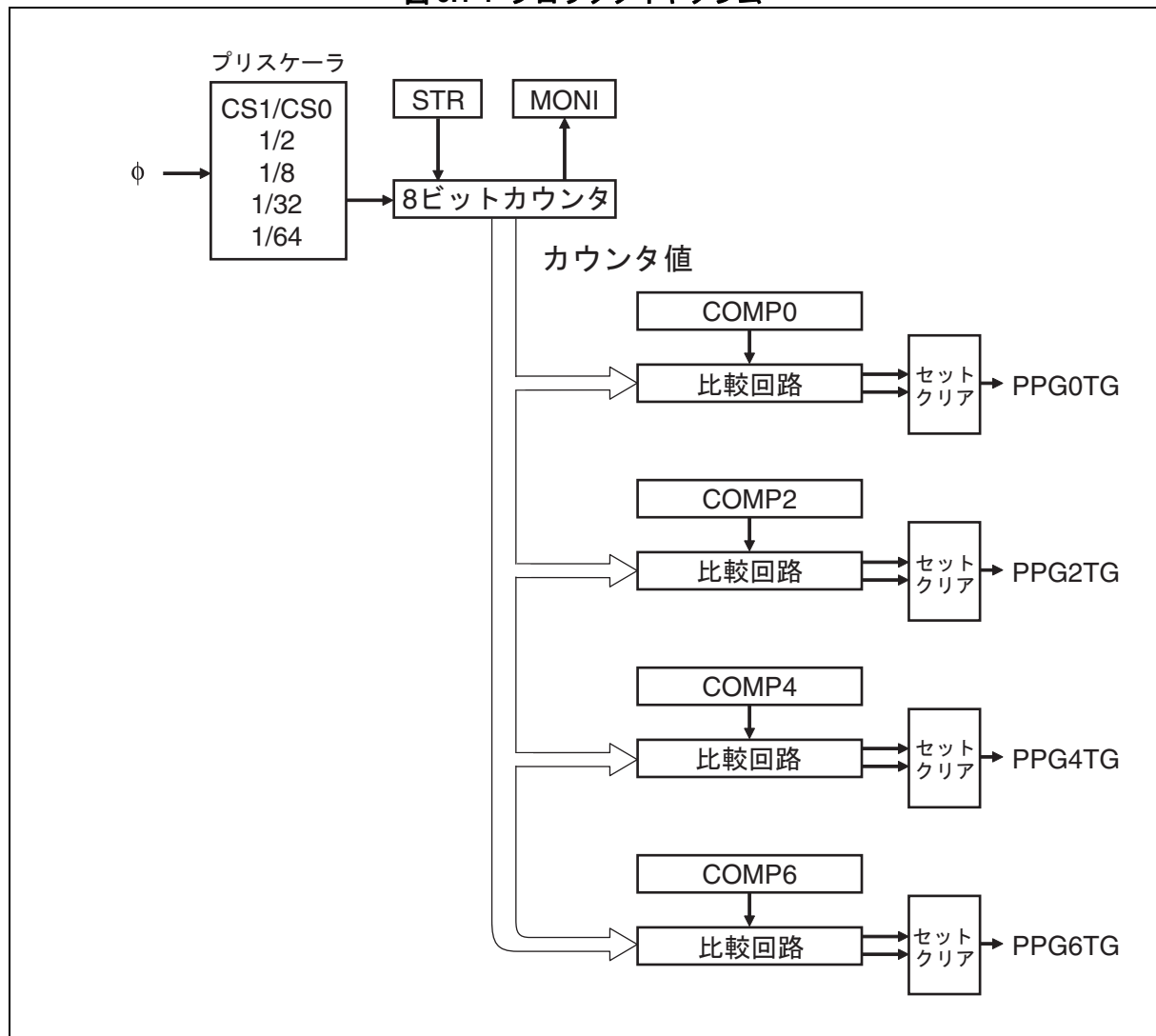
R/W : リード / ライト可能

R : リードオンリ

— : 未定義

■ ブロックダイアグラム

図 9.1-1 ブロックダイアグラム



■ タイミングジェネレータのレジスタ

● TTCR (Control Register)

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	初期値
000144 _H	TRG60	TRG40	TRG20	TRG00	CS1	CS0	MONI	STR	11110000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

タイミングジェネレータの制御レジスタです。

[bit31, bit30, bit29, bit28] TRG60/TRG40/TRG20/TRG00 (PPG Trigger Clear bit):
PPG トリガクリアビット

本ビットに "0" を書き込むことにより, 出力されている PPG 起動トリガをクリアします。

各ビットのトリガとの対応は以下のとおりです。

TRG00: PPG0TG

TRG20: PPG2TG

TRG40: PPG4TG

TRG60: PPG6TG

本レジスタの読出し値は常に "1" です。

[bit27, bit26] CS1/CS0 (Count Select bit): カウントクロック選択ビット

8 ビットカウンタの動作クロックを以下のように選択します。

CS1	CS0	クロックソース
0	0	マシクロック /2 (62.5 ns @32 MHz 時)
0	1	マシクロック /8 (250 ns @32 MHz 時)
1	0	マシクロック /32 (1 μs @32 MHz 時)
1	1	マシクロック /64 (2 μs @32 MHz 時)

[bit25] MONI (MONITOR bit):8 ビットカウンタ動作中モニタビット

8 ビットカウンタの動作を以下のように選択します。

0	カウンタ動作停止中
1	カウンタ動作中

書込み値は意味を持ちません。

[bit24] STR (START bit):8 ビットカウンタ動作許可ビット

8 ビットカウンタの動作を以下のように選択します。

0	意味を持ちません。
1	カウンタ動作開始

"0" 書込みは意味を持ちません。読出し値は常に "0" です。

コンペアレジスタ : COMP0, COMP2, COMP4, COMP6									
アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	初期値
000148 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	初期値
000149 _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00014A _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00014B _H	D7	D6	D5	D4	D3	D2	D1	D0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

8ビットカウンタと本レジスタ値が一致したときに PPG 起動信号がセットされます。
カウント動作中に本レジスタの書換えを行わないでください。
本レジスタ値が "00000000_B" の場合は PPG 起動信号はセットされません。

■ タイミングジェネレータの動作概要

- プリスケアラの動作
8ビットカウンタ用のカウントクロックをマシクロックより分周したものを設定するものです。
- 8ビットカウンタ
8ビットカウンタは、STR ビットによりプリスケアラからのカウントクロックでカウントを行います。8ビットカウンタはカウントアップを開始し、オーバーフローでカウント停止します。カウント中のカウンタスタートは無視されます。
8ビットカウンタがカウント中は MONI ビットには "1" が読めます。停止すると "0" が読めます。
8ビットカウンタのカウント値は各比較器に入力されています。

図 9.1-2 8ビットカウンタの動作・停止タイミング

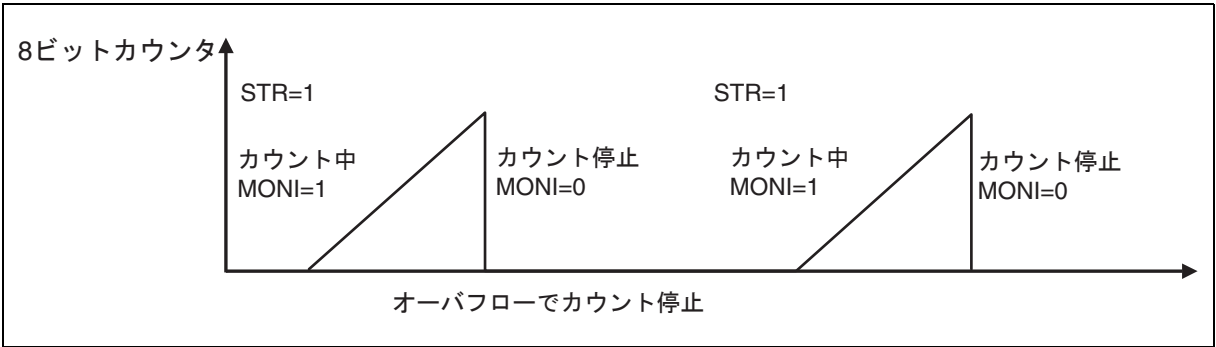
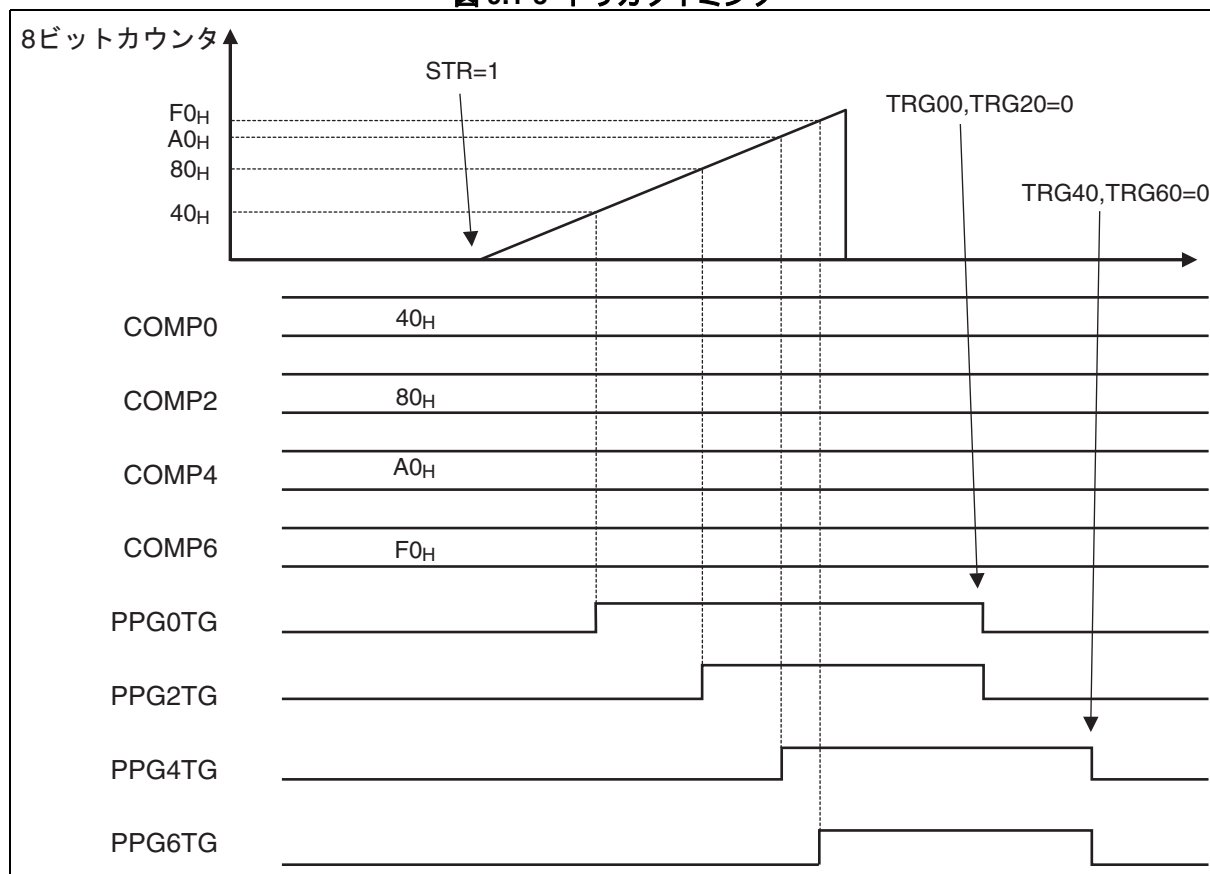


図 9.1-3 トリガタイミング



第10章

PPG

PPG の概要，レジスタの構成 / 機能および動作について説明します。

- 10.1 概要
- 10.2 ブロックダイヤグラム
- 10.3 PPG のレジスタ
- 10.4 動作説明

10.1 概要

PPG は、8 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。

ハードウェアとして、8 ビットダウンカウンタ 8 個、8 ビットリロードレジスタ 16 個、制御レジスタ、外部パルス出力 8 本、割込み出力 8 本を有します。

MB91265A シリーズは、8 ビット PPG として 8 チャンネル分、16 ビット PPG として 4 チャンネル分搭載しています。

■ PPG の機能

- 8 ビット PPG 出力独立動作モード
独立した PPG 出力動作が可能です。
- 16 ビット PPG 出力動作モード
1 チャンネルの 16 ビットの PPG 出力動作が可能です。
- 8 + 8 ビット PPG 出力動作モード
ch.(n + 1) の出力を ch.(n) のクロック入力とすることにより、任意周期の 8 ビット PPG 出力動作が可能です (n=0, 2, 4, 6)。
- 16 + 16 ビット PPG 出力動作モード
ch.(n + 3) + ch.(n + 2) の 16 ビットプリスケアラ出力を ch.(n + 1) + ch.(n) の 16 ビット PPG のクロック入力とするモードです (n=0, 4)。
- PPG 出力動作
任意周期・デューティ比のパルス波を出力します。
外付け回路により、D/A コンバータとしても使用可能です。
- 出力反転機能
PPG の出力値を反転させることが可能です。

■ レジスタ一覧

PPG 起動レジスタ (TRG)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000130 _H	-	-	-	-	-	-	-	-	00000000 _B
	-	-	-	-	-	-	-	-	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

出力反転レジスタ (REVC)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000134 _H	-	-	-	-	-	-	-	-	00000000 _B
	-	-	-	-	-	-	-	-	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

GATE 機能制御レジスタ (GATEC)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000133 _H	-	-	-	-	-	-	STGR	EDGE	XXXXXX00 _B
	-	-	-	-	-	-	R/W	R/W	

PPG0 ~ PPG7 動作モード制御レジスタ (PPGC0 ~ PPGC7)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000108 _H	PIEn	PUFn	INTMn	PCS1	PCS0	MD1*	MD0*	TTRGn	00000000 _B
ch.1 :000109 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :00010A _H									
ch.3 :00010B _H									
ch.4 :000114 _H									
ch.5 :000115 _H									
ch.6 :000116 _H									
ch.7 :000117 _H									

n=0 ~ 7

R/W : リード / ライト可能

X : 不定

- : 未定義

*: MD1, MD0 は、偶数チャンネルのみ存在し、奇数チャンネルには存在しません。奇数チャンネルの初期値は不定です。書込みは意味がありません。

● リロードレジスタ : 8 ビット PPG モード

リロードレジスタ H (PRLH0 ~ PRLH7)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000100 _H									XXXXXXXX _B
ch.1 :000102 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :000104 _H									
ch.3 :000106 _H									
ch.4 :00010C _H									
ch.5 :00010E _H									
ch.6 :000110 _H									
ch.7 :000112 _H									

リロードレジスタ L (PRL0 ~ PRL7)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000101 _H									XXXXXXXX _B
ch.1 :000103 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :000105 _H									
ch.3 :000107 _H									
ch.4 :00010D _H									
ch.5 :00010F _H									
ch.6 :000111 _H									
ch.7 :000113 _H									

R/W : リード / ライト可能

X : 不定

● リロードレジスタ : 16 ビット PPG モード

リロードレジスタ H (PRLH0, PRLH2, PRLH4, PRLH6)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000100 _H									XXXXXXXX _B
ch.2 :000104 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.4 :00010C _H									
ch.6 :000110 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
									XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

リロードレジスタ L (PRL0, PRL2, PRL4, PRL6)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000102 _H									XXXXXXXX _B
ch.2 :000106 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.4 :00010E _H									
ch.6 :000112 _H									
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
									XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

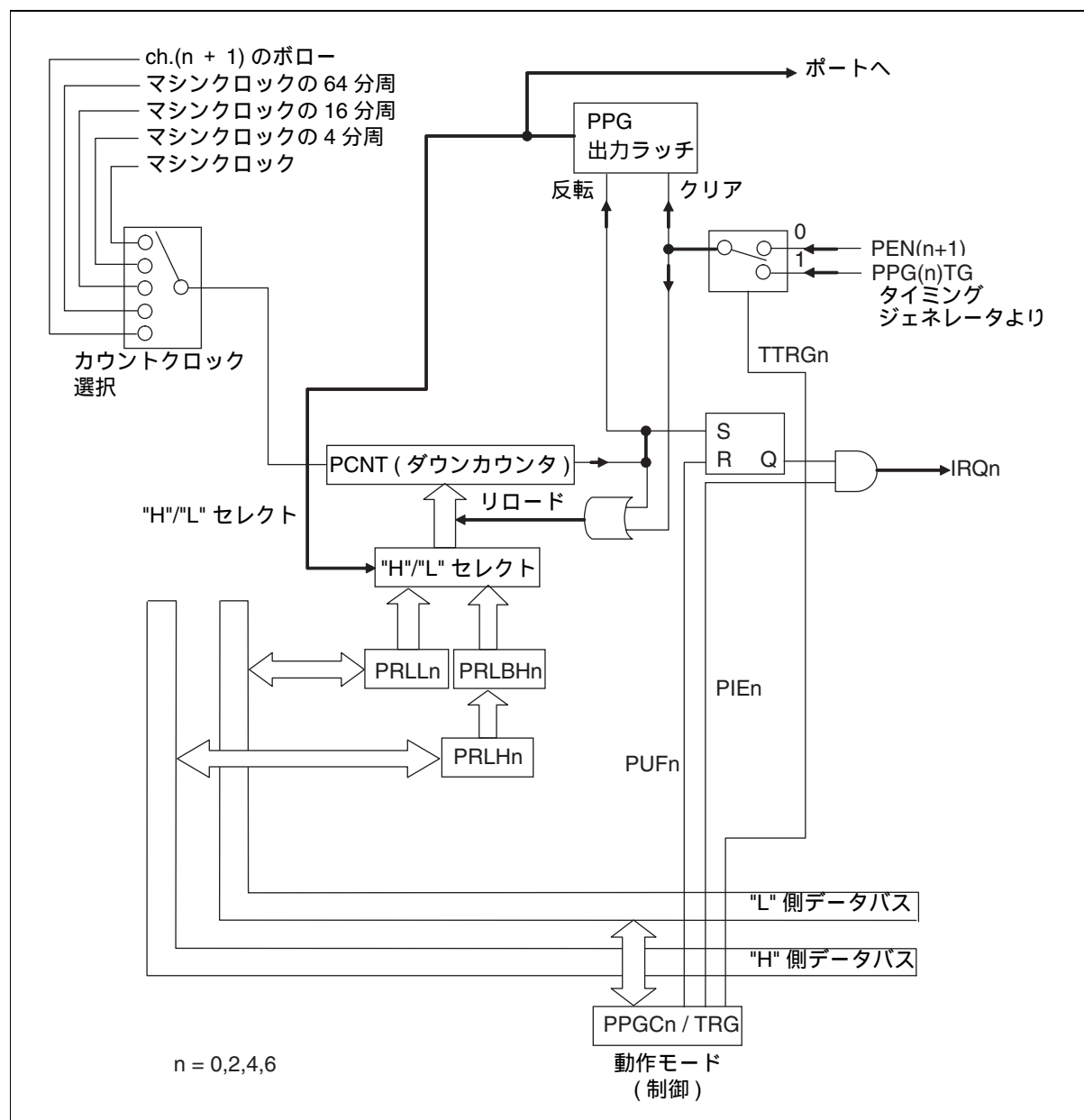
R/W : リード / ライト可能

X : 不定

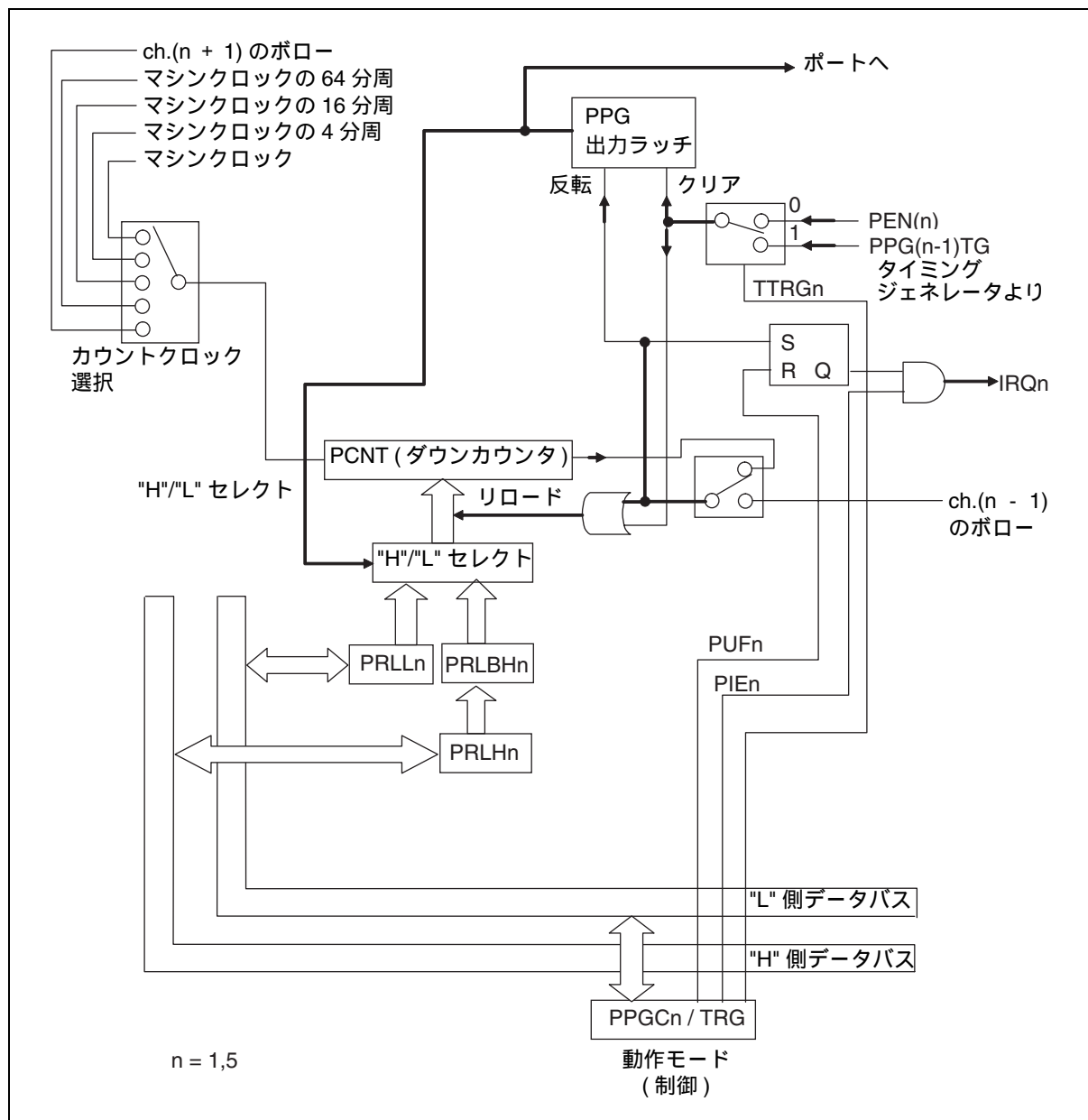
10.2 ブロックダイアグラム

PPG のブロックダイアグラムを示します。

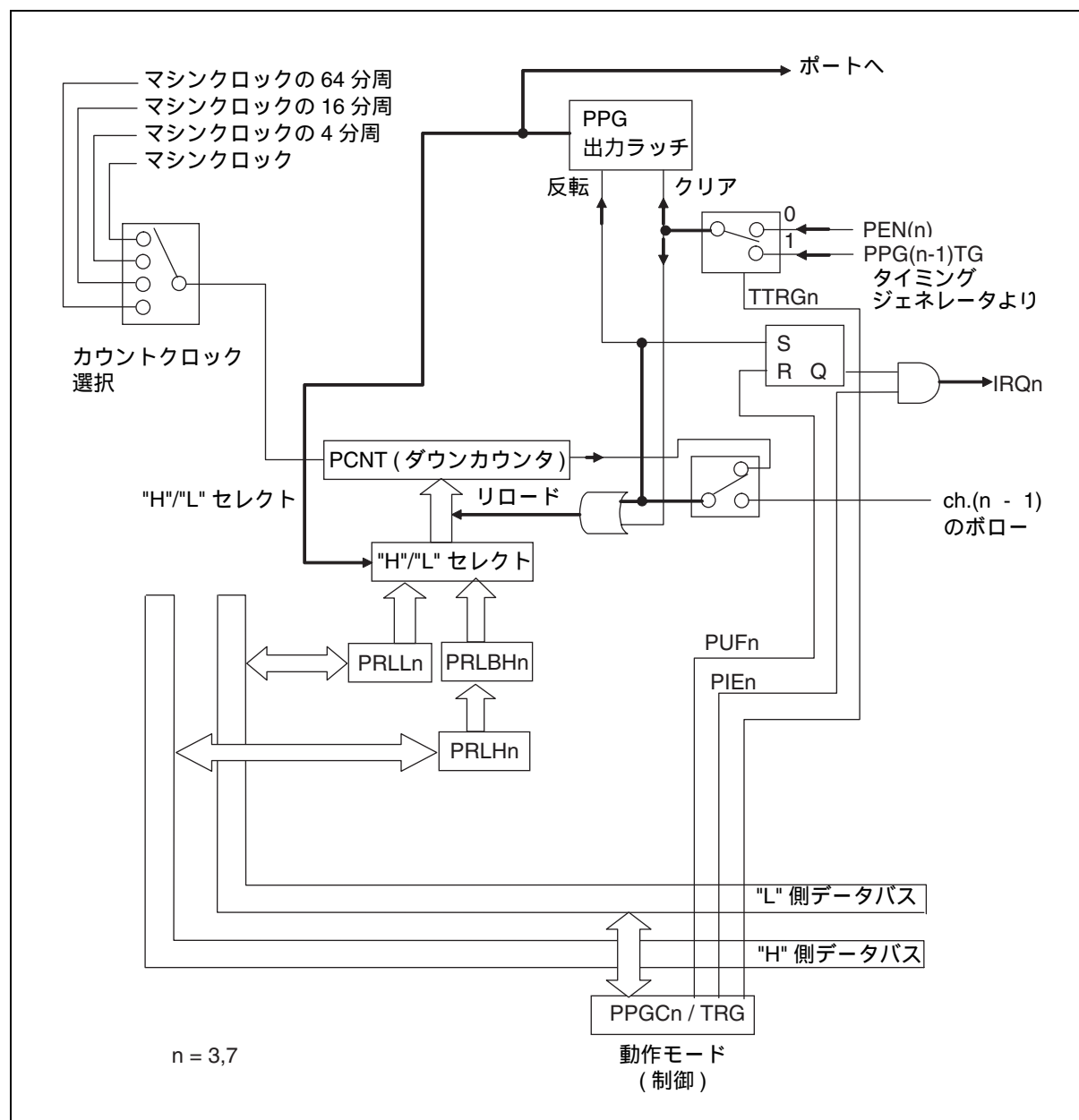
■ 8 ビット PPG ch.0, ch.2, ch.4, ch.6 のブロックダイアグラム



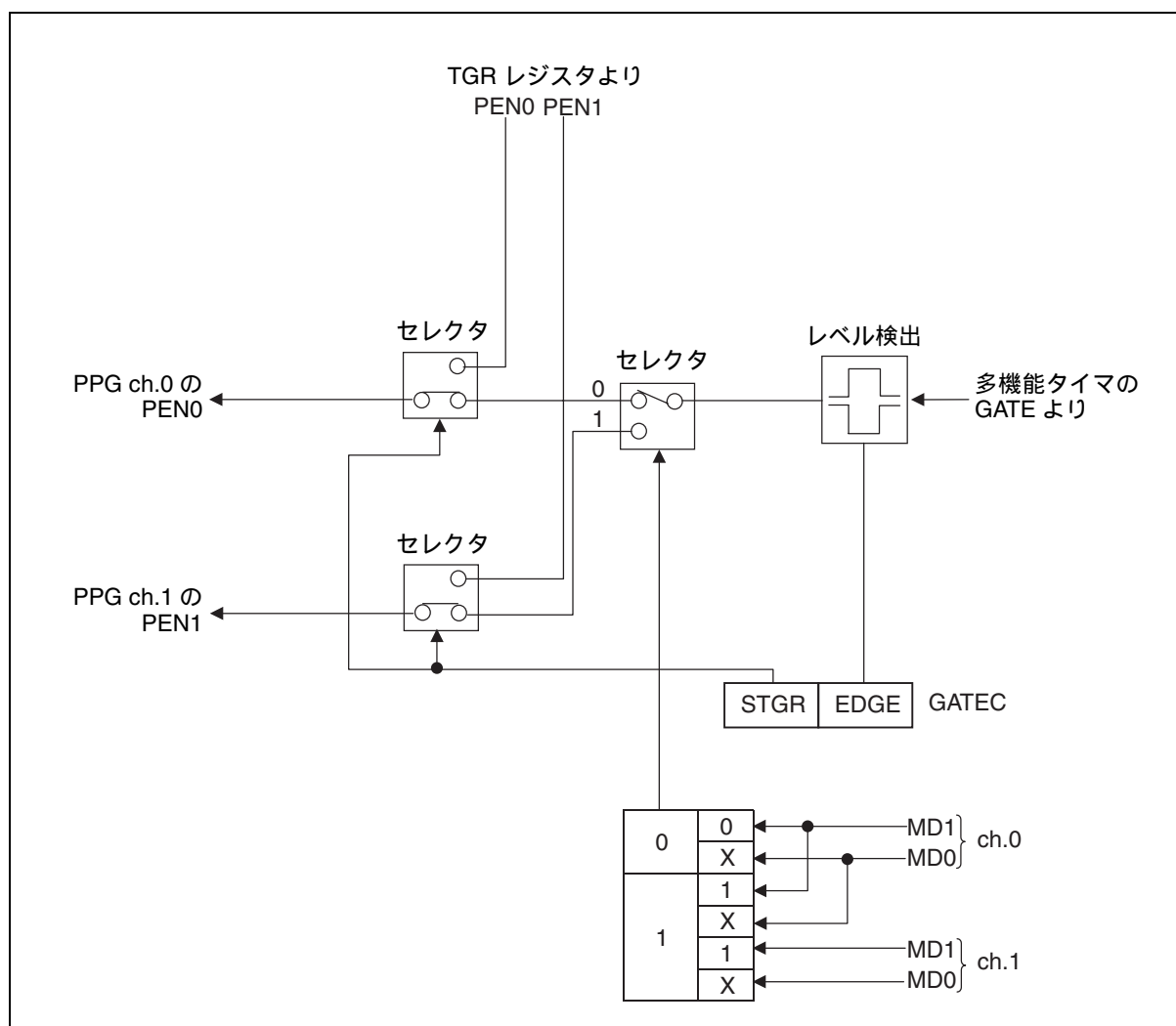
■ 8 ビット PPG ch.1, ch.5 のブロックダイヤグラム



■ 8 ビット PPG ch.3, ch.7 のブロックダイヤグラム



■ ゲート機能のブロックダイアグラム



10.3 PPG のレジスタ

PPG のレジスタについて説明します。

■ PPGCn レジスタ (PPGn 動作モード制御レジスタ) n=0, 1, 2, 3, 4, 5, 6, 7

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000108 _H	PIEn	PUFn	INTMn	PCS1	PCS0	MD1	MD0	TTRGn	00000000 _B
ch.1 :000109 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :00010A _H									
ch.3 :00010B _H									
ch.4 :000114 _H									
ch.5 :000115 _H									
ch.6 :000116 _H									
ch.7 :000117 _H									
R/W : リード / ライト可能									
X : 不定									

[bit7] PIEn(Ppg Interrupt Enable) : PPG 割込み許可ビット

PPG の割込み許可を以下のように制御します。

0	割込み禁止
1	割込み許可

- 本ビットが "1" のとき , PUFn が "1" になると割込み要求が発生します。
- 本ビットが "0" のときは , 割込み要求を発生しません。
- リセットにより , "0" に初期化されます。
- 読出し / 書込み可能です。

[bit6] PUFn(Ppg Underflow Flag) : PPG カウンタアンダフロービット

PPG カウンタアンダフロービットを以下のように制御します。

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

- 8ビットPPG 2チャンネルモードおよび8ビットプリスケラ + 8ビットPPG モード時には , ch.0 のカウント値が "00_H" ~ "FF_H" になったときのアンダフローにより "1" にセットされます。
- 16 ビット PPG 1 チャンネルモード時には , ch.1/ch.0 のカウント値が "0000_H" ~ "FFFF_H" になったときのアンダフローにより "1" にセットされます。
- "0" 書込みにより , "0" になります。
- このビットへの "1" 書込みは意味がありません。
- リードモディファイライトへの読出し時は , "1" が読み出されます。
- リセットにより , "0" に初期化されます。
- 読出し / 書込み可能です。

[bit5] INTMn(Interrupt Mode) : 割込みモードビット

PUFn のビットの検出を PRLBHn からのアンダフロー時のみに限定することができます。

0	アンダフロー時, PUFn を "1" にします。
1	PRLBHn からのアンダフロー時のみ, PUFn を "1" にします。

- リセットにより, "0" に初期化されます。
- 読出し / 書込み可能です。
- 本ビットを "1" にすれば, PPG の波形の 1 周期出力時に割込みをかけることが可能となります。
- 本ビットは, 割込み許可時に書き換えしないでください。

[bit4, bit3] PCS1, PCS0 (Ppg Count Select) : カウントクロック選択ビット

ダウンカウンタの動作クロックを以下のように選択します。

PCS1	PCS0	動作モード
0	0	マシナクロック (62.5 ns マシナクロック 16 MHz 時)
0	1	マシナクロック /4 (250 ns マシナクロック 16 MHz 時)
1	0	マシナクロック /16 (1 μ s マシナクロック 16 MHz 時)
1	1	マシナクロック /64 (4 μ s マシナクロック 16 MHz 時)

- リセットにより, "00_B" に初期化されます。
- 読出し / 書込み可能です。

[bit2, bit1] MD1, MD0 (ppg count MoDe) : 動作モード選択ビット

PPG タイマの動作モードを以下のように選択します。

MD1	MD0	動作モード
0	0	8 ビット PPG 2 チャンネル独立モード
0	1	8 ビットプリスケラ + 8 ビット PPG モード
1	0	16 ビット PPG モード
1	1	16 ビットプリスケラ + 16 ビット PPG モード

- リセットにより, "00_B" に初期化されます。
- 読出し / 書込み可能です。
- 本ビットは偶数チャンネルのみに存在します。

[bit0] TTRGn(Timing TRGer) : タイミングトリガ選択ビット

タイミングジェネレータからの起動信号によってのみPPGを起動させることができます。

0	TRG レジスタもしくは多機能タイマにより起動します。
1	タイミングジェネレータによる起動のみとなります。

- リセットにより, "00_B" に初期化されます。
- 読出し / 書込み可能です。

■ PRLH/PRLH レジスタ (リロードレジスタ : PRL0 ~ PRL7/PRLH0 ~ PRLH7)

リロードレジスタ H (PRLH0 ~ PRLH7)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000100 _H									XXXXXXXX _B
ch.1 :000102 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :000104 _H									
ch.3 :000106 _H									
ch.4 :00010C _H									
ch.5 :00010E _H									
ch.6 :000110 _H									
ch.7 :000112 _H									

リロードレジスタ L (PRL0 ~ PRL7)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000101 _H									XXXXXXXX _B
ch.1 :000103 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ch.2 :000105 _H									
ch.3 :000107 _H									
ch.4 :00010D _H									
ch.5 :00010F _H									
ch.6 :000111 _H									
ch.7 :000113 _H									

R/W : リード / ライト可能

X : 不定

ダウンカウンタ PCNT へのリロード値を保持するレジスタです。それぞれ、以下に示す役割を持っています。

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

どのレジスタも読出し / 書込み可能です。

< 注意事項 >

8 ビットプリスケラ + 8 ビット PPG モードおよび 16 ビットプリスケラ + 16 ビット PPG モードで使用する場合には、プリスケラ側の PRL と PRLH に異なる値を設定すると、PPG 波形がサイクルごとに異なる場合があるので、プリスケラ側の PRL と PRLH は、同じ値に設定することを推奨します。

■ PPG 起動レジスタ (TRG)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000130 _H	-	-	-	-	-	-	-	-	00000000 _B
	-	-	-	-	-	-	-	-	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能
- : 未定義

[bit7 ~ bit0] PEN7 ~ PEN0 (Ppg ENable) : PPG 動作許可ビット
PPG の動作開始および動作モードを以下のように選択します。

PEN7 ~ PEN0	動作状態
0	動作停止 ("L" レベル出力保持)
1	PPG 動作許可

- リセットにより, "0" に初期化されます。
- 読出し / 書込み可能です。

■ 出力反転レジスタ (REVC)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000134 _H	-	-	-	-	-	-	-	-	00000000 _B
	-	-	-	-	-	-	-	-	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能
- : 未定義

[bit7 ~ bit0] REV7 ~ REV0 : 出力反転ビット
PPG の出力値を初期レベルも含めて反転します。

REV7 ~ REV0	出力レベル
0	通常
1	反転

- リセットにより, "0" に初期化されます。
- 読出し / 書込み可能です。
- 単に, PPG 出力を反転するだけですので, 初期レベルも反転します。
また, リロードレジスタの "L", "H" の関係も逆になります。

■ GATE 機能制御レジスタ (GATEC)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000133 _H	-	-	-	-	-	-	STGR	EDGE	XXXXXX00 _B
	-	-	-	-	-	-	R/W	R/W	

R/W : リード / ライト可能
 X : 不定
 - : 未定義

[bit1] STGR : ゲート機能選択ビット

多機能タイマからの起動信号を用いるか, TRG レジスタによる起動を行うかを以下のように選択します。

STGR	動作モード
0	TRG レジスタによる起動
1	多機能タイマからの起動信号による起動

- リセットにより, "0" に初期化されます。
- 読出し / 書込み可能です。

[bit0] EDGE : 起動有効エッジ選択ビット

多機能タイマからの起動有効エッジを以下のように選択します。

EDGE	動作モード
0	立上り起動 立下り停止 ^{*1}
1	立下り起動 立上り停止 ^{*2}

- リセットにより, "0" に初期化されます。
- 読出し / 書込み可能です。

*1: "H" の間, 起動します。

*2: "L" の間, 起動します。

10.4 動作説明

PPG には、8 ビット長の PPG ユニットが 8 チャンネルあり、独立モード以外に、連結動作させることにより、8 ビットプリスケラ + 8 ビット PPG モードと、16 ビット PPG 1 チャンネルモード、16 ビットプリスケラ + 16 ビット PPG モードの計 4 種類の動作を行うことができます。

■ 動作説明

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

動作開始 / 再スタートは、レジスタのビット書込みによります。

リロード動作とパルス出力の関係を以下に示します。

リロード動作	端子出力変化
PRLH PCNT	PPGn [0 1] 
PRLH PCNT	PPGn [1 0] 

n=0 ~ 7

また、PPGn レジスタの bit7 : PIE_n が "1" のとき、カウンタの "00_H" ~ "FF_H" へのボロー (16 ビット PPG モードの場合には、"0000_H" ~ "FFFF_H" へのボロー) によって割込み要求が出力されます。

● 動作モードについて

本ブロックは、独立モードと、8 ビットプリスケラ + 8 ビット PPG モードと、16 ビット PPG 1 チャンネルモードと、16 ビットプリスケラ + 16 ビット PPG モードの計 4 種類の動作モードがあります。

- 独立モードは、8 ビット PPG として独立に動作させる動作モードです。PPGn 端子は ch.(n) の PPG 出力が接続されます (n=0 ~ 7)。
- 8 ビットプリスケラ + 8 ビット PPG モードは、1 チャンネルを 8 ビットプリスケラとして動作させ、そのボロー出力でカウントすることにより、任意周期の 8 ビット PPG 波形を出力できるようにする動作モードです。例えば、PPG1 端子は、ch.1 のプリスケラ出力が接続され、PPG0 端子は、ch.0 の PPG 出力が接続されます。
- 16 ビット PPG 1 チャンネルモードは、2 つのチャンネルを連結させ、16 ビット PPG として動作させる動作モードです。例えば、ch.0 と ch.1 を連結させると、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。

● PPG 出力動作について

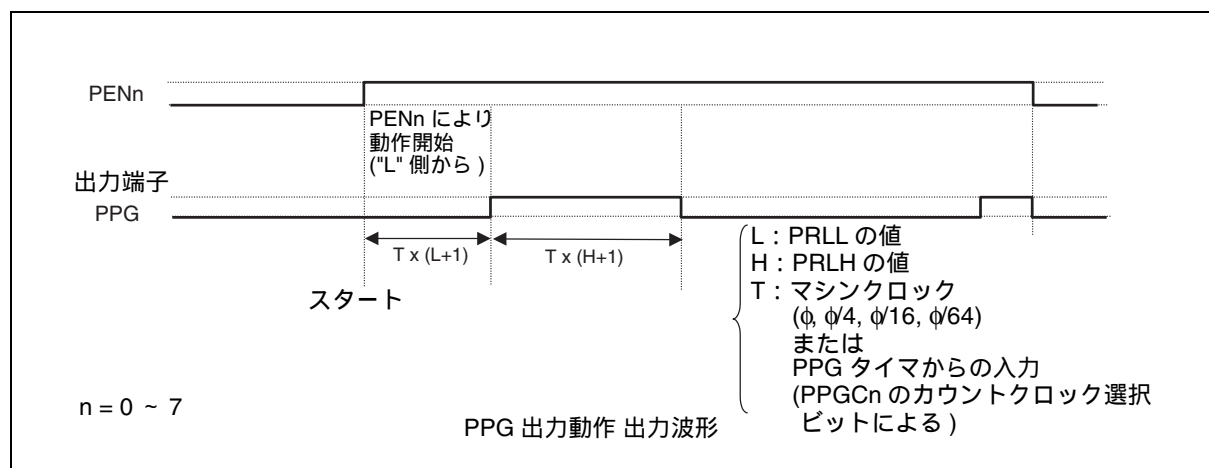
PPG は、TRG レジスタ (PPG 起動レジスタ) の各チャネルのビットを "1" にセットすることによって起動され、カウントを開始します。動作を開始した後は、TRG レジスタの各チャネルビットに "0" を書き込むことによってカウント動作を停止し、停止した後、パルス出力は "L" レベルを保持します。

8 ビットプリスケアラ + 8 ビット PPG モードおよび 16 ビットプリスケアラ + 16 ビット PPG モード時には、プリスケアラチャネルを停止状態で、PPG チャネルを動作状態に設定しないでください。

16 ビット PPG モード時には、各チャネルの TRG レジスタの PEN_n をそれぞれ、同時に開始 / 停止の制御を行ってください ($n=0 \sim 7$)。

以下に PPG 出力動作について説明します。

PPG 動作時は、任意周波数 / 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すると、動作停止を設定するまで停止しません。



● リロード値とパルス幅の関係について

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_H のとき、および 16 ビット PPG 動作時のリロードレジスタ値が 0000_H のときは、カウントクロック 1 周期分のパルス幅をもつこととなりますので注意してください。また、8 ビット PPG 動作時のリロードレジスタ値が FF_H のとき、カウントクロック 256 周期分のパルス幅をもつことになり、16 ビット PPG 動作時のリロードレジスタ値が $FFFF_H$ のときは、カウントクロック 65536 周期分のパルス幅をもつこととなりますので注意してください。

パルス幅の計算式を以下に示します。

$$\begin{aligned} Pl &= T \times (L + 1) \\ Ph &= T \times (H + 1) \end{aligned} \quad \left\{ \begin{array}{l} L : \text{PRLH の値} \\ H : \text{PRLH の値} \\ T : \text{入力クロック周期} \\ Ph : \text{"H" パルス幅} \\ Pl : \text{"L" パルス幅} \end{array} \right.$$

● カウントクロックの選択について

本ブロックの動作に使用するカウントクロックは、周辺クロックおよびタイムベースカウンタの入力を使用しており、4 種類のカウントクロック入力を選択できます。

カウントクロックは以下のように動作します。

PPGC0 ~ PPGC7 レジスタ		カウントクロック動作
PCS1	PCS0	
0	0	カウントクロックは、周辺クロックごとに 1 カウント
0	1	カウントクロックは、周辺クロック 4 サイクルごとに 1 カウント
1	0	カウントクロックは、周辺クロック 16 サイクルごとに 1 カウント
1	1	カウントクロックは、周辺クロック 64 サイクルごとに 1 カウント

8 ビットプリスケアラ + 8 ビット PPG モードおよび 16 ビットプリスケアラ + 16 ビット PPG モードで、プリスケアラ側が動作状態で、PPG 側が停止状態であるときに、PPG 側の起動を行うと、最初のカウント周期がずれる可能性がありますので注意してください。

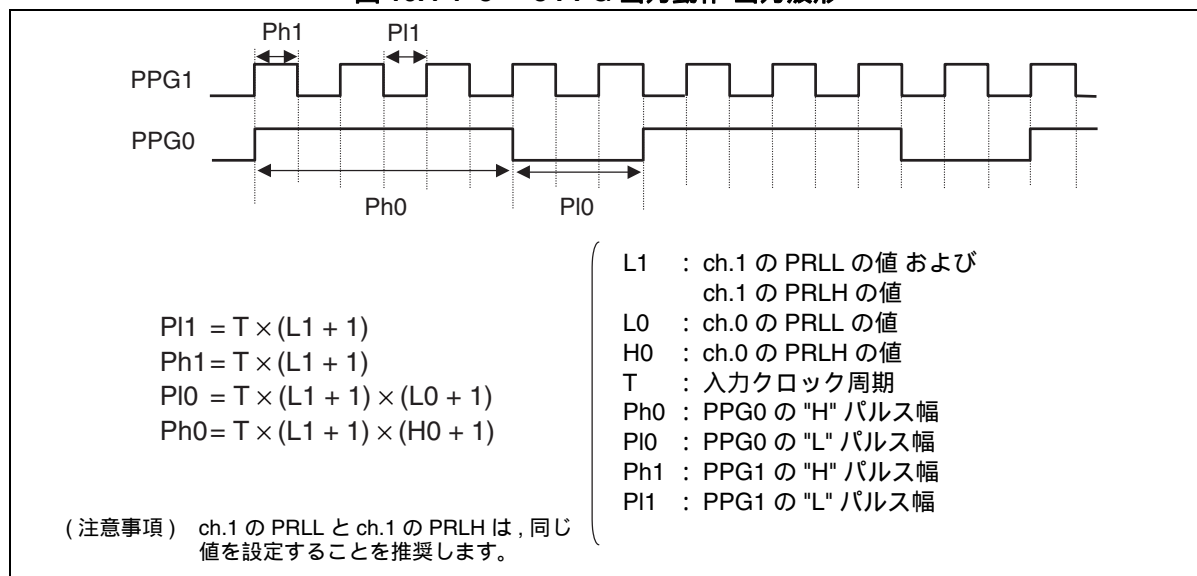
● パルスの端子出力の制御について

本モジュールの動作によって生成されたパルス出力は、外部端子 PPG0 ~ PPG7 より出力させることができます。

16 ビット PPG モードでは、PPG(m) と PPG(m + 1) 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 (m = 0, 2, 4, 6)。

8 ビットプリスケアラ + 8 ビット PPG モードおよび 16 ビットプリスケアラ + 16 ビット PPG モードでは、プリスケアラ側は 8 ビットプリスケアラのトグル波形が出力され、PPG 側は 8 ビット PPG の波形が出力されます。このモードのときの出力波形の例を以下に示します。

図 10.4-1 8 + 8 PPG 出力動作 出力波形



● 割込みについて

本モジュールの割込みは、リロード値がカウントアウトし、ポローが発生したときにアクティブになります。

ただし、INTM_n ビットを "1" にしたときは、PRLBH_n からのアンダフロー時（ポロー）のみアクティブになります。つまり、"H" 幅パルス終了時に割込みが発生します。

8 ビット PPG モードおよび 8 ビットプリスケラ + 8 ビット PPG モードのときには、それぞれのカウンタのポローにより、それぞれの割込み要求が行われますが、16 ビット PPG モードおよび 16 ビットプリスケラ + 16 ビット PPG モードでは、16 ビットカウンタのポローにより、PUF(m) と PUF(m + 1) 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、PIE(m) または PIE(m + 1) 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 PUF(m) と PUF(m + 1) を同時にを行うことを推奨します (m = 0, 2, 4, 6)。

● GATE 機能について

多機能タイマからの信号により、PPG を起動 停止させることができます。

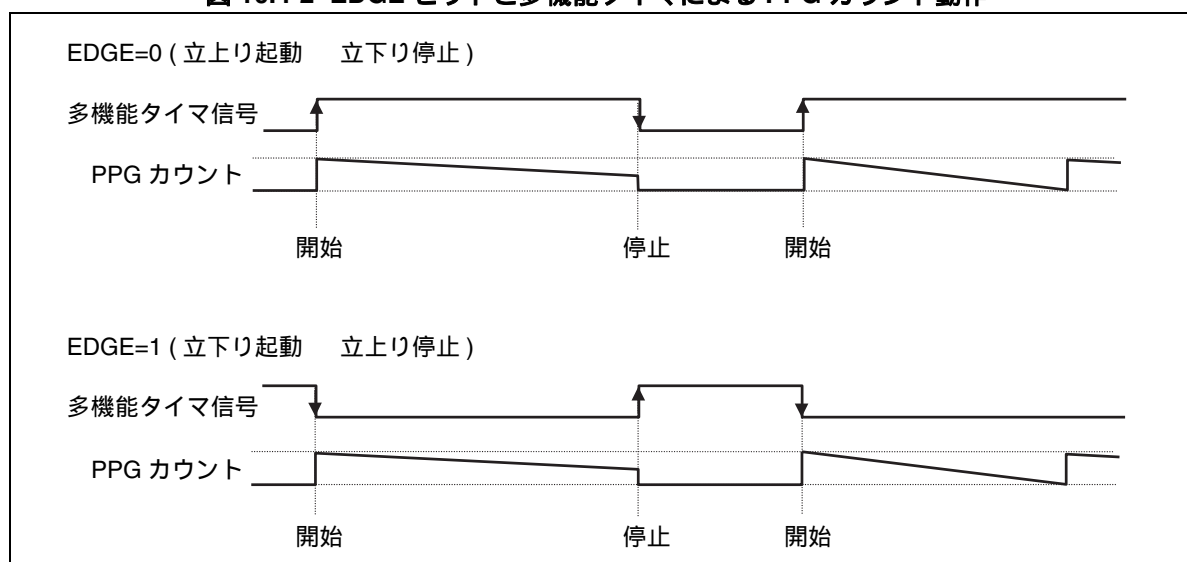
- 8 ビット PPG モード、8 ビットプリスケラ + 8 ビット PPG モードにおいて、PPG ch.0 を本機能により起動できます。
- 16 ビット PPG モード、16 ビットプリスケラ + 16 ビット PPG モードにおいて、PPG ch.0, ch.1 を本機能により起動できます。

各モードの起動切換えは、PPG 動作モード制御レジスタの MD1, MD0 ビットの設定によって定まります。

- PPG ch.0 : MD1, MD0 = 0, X 時、PPG ch.0 が起動 (8 ビット PPG)
- PPG ch.0 : MD1, MD0 = 1, X 時、PPG ch.0, ch.1 が起動 (16 ビット PPG)

EDGE ビットと多機能タイマの信号により、PPG の起動有効期間の制御ができます。

図 10.4-2 EDGE ビットと多機能タイマによる PPG カウント動作



● 各ハードウェアの初期値について

本ブロック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 >	PPGC(n)	0000000X _B
< パルス出力 >	PPG(n)	"L"
< 割込み要求 >	IRQ(n)	"L" (n=0 ~ 7)

上記以外のハードウェアは、初期化されません。

● PPG の組合せについて

ch.0: PPGC		ch.2: PPGC		ch.0	ch.1	ch.2	ch.3
MD1	MD0	MD1	MD0				
0	0	0	0	8 ビット PPG	8 ビット PPG	8 ビット PPG	8 ビット PPG
0	0	0	1	8 ビット PPG	8 ビット PPG	8 ビット PPG ←	8 ビット プリスケーラ
0	0	1	0	8 ビット PPG	8 ビット PPG	16 ビット PPG	
0	0	1	1	設定禁止			
0	1	0	0	8 ビット PPG ←	8 ビット プリスケーラ	8 ビット PPG	8 ビット PPG
0	1	0	1	8 ビット PPG ←	8 ビット プリスケーラ	8 ビット PPG ←	8 ビット プリスケーラ
0	1	1	0	8 ビット PPG ←	8 ビット プリスケーラ	16 ビット PPG	
0	1	1	1	設定禁止			
1	0	0	0	16 ビット PPG		8 ビット PPG	8 ビット PPG
1	0	0	1	16 ビット PPG		8 ビット PPG ←	8 ビット プリスケーラ
1	0	1	0	16 ビット PPG		16 ビット PPG	
1	0	1	1	設定禁止			
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1	16 ビット PPG ←		16 ビットプリスケーラ	

ch.(4, 5, 6, 7) も、それぞれ、ch.(0, 1, 2, 3) と同じ動作組合せが可能です。

以下のように置き換えてください。

$$\left\{ \begin{array}{l} \text{ch.0}=\text{ch.4} \\ \text{ch.1}=\text{ch.5} \\ \text{ch.2}=\text{ch.6} \\ \text{ch.3}=\text{ch.7} \end{array} \right.$$

第11章

PWC (Pulse Width Count: パルス幅測定)

PWC の概要，レジスタの構成 / 機能および動作について説明します。

- 11.1 概要
- 11.2 ブロックダイアグラム
- 11.3 PWC のレジスタ
- 11.4 動作説明

11.1 概要

入力信号のパルス幅測定機能です。

■ PWC の機能

ハードウェアとして、16 ビットアップカウンタ 1 個、入力パルス分周器 & 分周比制御レジスタ 1 個、測定入力端子 1 本、16 ビット制御レジスタ 1 個を有し、これらによって以下の機能を実現します。

● パルス幅測定機能

外部からのパルス入力の任意イベント間の時間を測定します。

基準となる内部クロックは 3 種類の中から選択可能です

(マシンのクロックの 4/16/32 分周)。

各種測定モード	"H" パルス幅 (~) / "L" パルス幅 (~)
	立上り周期 (~) / 立下り周期 (~)
	エッジ間測定 (or ~ or)

8 ビット入力分周器で、入力パルスを 2^n 分周 ($n=1, 2, 3, 4, 5, 6, 7, 8$) して周期測定を行うことが可能です。

測定終了時に割込み要求を発生することが可能です。

1 回のみの測定か、連続測定かを選択することが可能です。

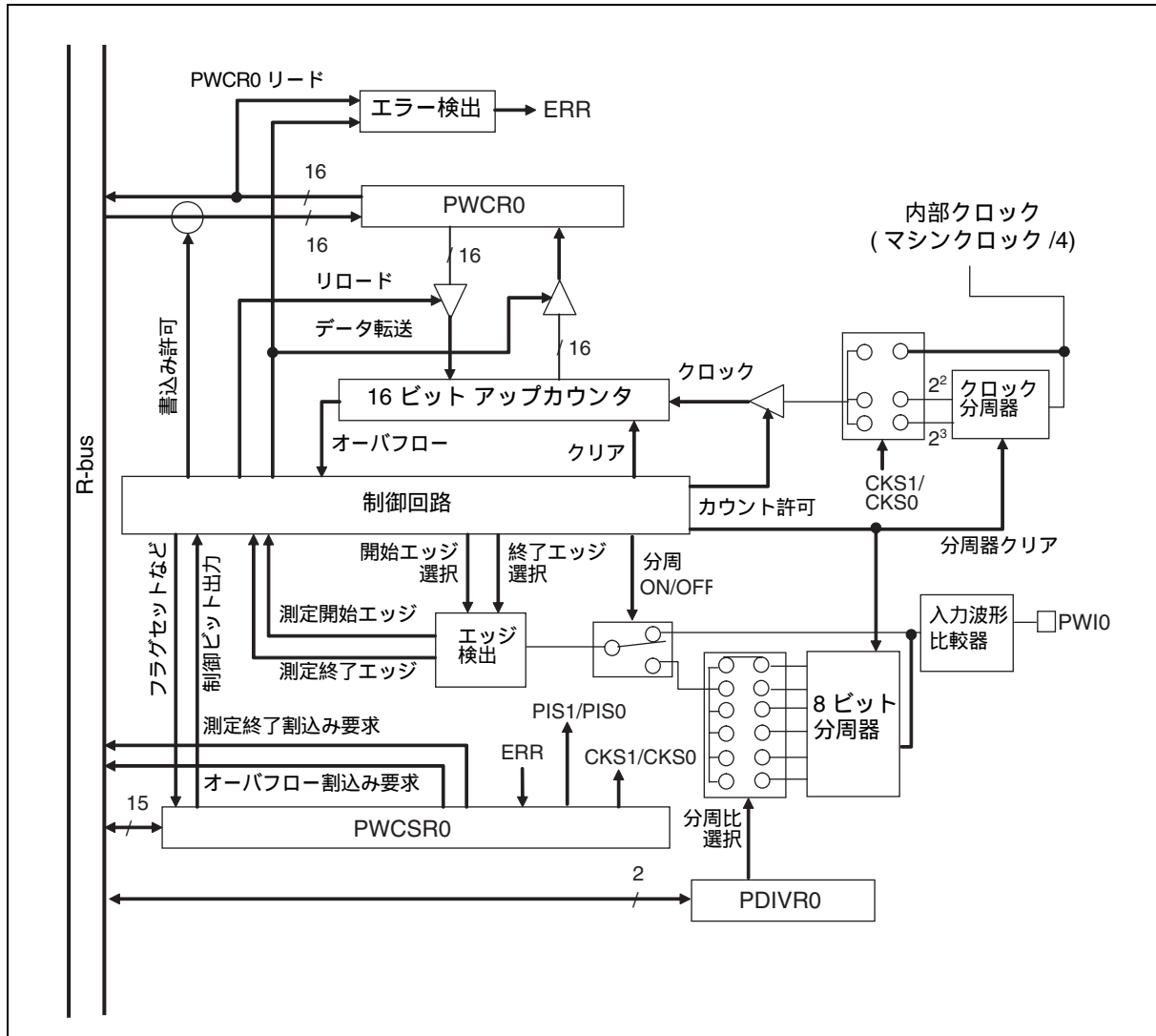
■ レジスタ一覧

アドレス	bit15	~	bit8	bit7	~	bit0	
0000E9 _H				PDIR0			分周比制御レジスタ 0
0000E0 _H	0000E1 _H	PWCSR0					コントロール / ステータスレジスタ 0
0000E2 _H	0000E3 _H	PWCR0					データバッファレジスタ 0

11.2 ブロックダイアグラム

PWC のブロックダイアグラムを示します。

■ PWC のブロックダイアグラム



11.3 PWC のレジスタ

PWC のレジスタ詳細を説明します。

■ PWC コントロール / ステータスレジスタ (PWCSR: PWCSR0)

PWCSR0 (上位)									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0: 0000E0 _H	STRT	STOP	EDIR	EDIE	OVIR	OVIE	ERR	-	00000000 _B
	R/W	R/W	R	R/W	R/W	R/W	R	R/W	

PWCSR0 (下位)									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0: 0000E1 _H	CKS1	CKS0	PIS1	PIS0	SC	MOD2	MOD1	MOD0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W	: リード / ライト可能
R	: リードオンリ

[bit15] STRT : カウンタスタートビット

[bit14] STOP : カウンタストップビット

16 ビットアップカウンタの起動 / 再起動 / 停止を行うビットで、読出し時にはカウンタの動作状態を表示します。以下にビット機能を示します。

表 11.3-1 書き込み時機能 (動作制御)

STRT	STOP	動作制御機能
0	0	機能なし / 動作に影響なし
0	1	カウンタ起動 / 再起動 (カウント許可) (注意事項) クリアビット命令使用可能
1	0	カウンタ動作強制停止 (カウント禁止) (注意事項) クリアビット命令使用可能
1	1	機能なし / 動作に影響なし

表 11.3-2 読出し時機能 (動作状態表示)

STRT	STOP	動作状態表示
0	0	カウント停止中 (起動されていないかまたは測定終了) [初期値]
1	1	カウント動作中 (測定中)

- リセット時: "00_B" に初期化されます。
- 読出し / 書き込み可能です。ただし、書き込み時と読出し時では上記のように意味が異なります。

- ・ リードモディファイライト (RMW) 系命令における読出しでは、動作にかかわらず "11_B" が読み出されます。
- ・ カウンタの起動 / 停止のための STRT, STOP ビットへの書込みは、それぞれのビットに対応するビット処理命令 (ビットクリアなど) を用いることが可能ですが、動作状態の読出しにはビット処理命令は使用できません (読み出すと必ず動作中となりますので注意してください)。

[bit13] EDIR : 測定終了割込み要求フラグ

パルス幅測定モード時、測定終了したことを示すフラグです。測定終了割込み要求が許可されているとき (bit12: EDIE=1) に、本ビットがセットされると、測定終了割込み要求が発生します。

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る)
クリア要因	PWCR (測定結果) を読み出すことによりクリア

- ・ リセット時: "0" に初期化されます。
- ・ 読出しのみ可能です。書込みしてもビット値は変化しません。

[bit12] EDIE : 測定終了割込み要求許可ビット

パルス幅測定モード時の測定終了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出力禁止 (EDIR がセットされても割込みは発生せず) [初期値]
1	測定終了割込み要求出力許可 (EDIR がセットされると割込みが発生する)

- ・ リセット時: "0" に初期化されます。
- ・ 読出し / 書込み可能です。

[bit11] OVIR : カウンタオーバーフロー割込み要求フラグ

全モードにおいて、16 ビットアップカウンタが "FFFF_H" から "0000_H" へオーバーフローしたことを示すフラグです。カウンタオーバーフロー割込み要求が許可されているとき (bit10: OVIE=1) に、本ビットがセットされると、カウンタオーバーフロー割込み要求が発生します。

セット要因	カウンタオーバーフローが発生するとセット ("FFFF _H " から "0000 _H " へ)
クリア要因	"0" 書込みによりクリア

- ・ リセット時: "0" に初期化されます。
- ・ 読出し / 書込み可能です。ただし、書込みは "0" のみ可能です。"1" を書き込んでもビット値は変化しません。
- ・ リードモディファイライト (RMW) 系命令における読出し値は、ビット値にかかわらず "1" です。

[bit10] OVIE : カウンタオーバーフロー割込み要求許可ビット

カウンタオーバーフロー割込み要求を以下のように制御します。

0	オーバーフロー割込み要求出力禁止 (OVIR がセットされても割込みは発生せず)	[初期値]
1	オーバーフロー割込み要求出力許可 (OVIR がセットされると割込みが発生する)	

- リセット時 : "0" に初期化されます。
- 読出し / 書込み可能です。

[bit9] ERR : エラーフラグ

パルス幅測定モードの連続測定モード時において、PWCR 内の測定結果を読み出さないうちに、次の測定が終了してしまったことを示すフラグです。この際、PWCR の値は新しい測定結果に更新され、1 つ前の測定結果は消失します。測定は、本ビット値に関係なく続行されます。

セット要因	読み出していない測定結果が次の結果により消失するとセット
クリア要因	PWCR (測定結果) を読み出すことによりクリア

- リセット時 : "0" に初期化されます。
- 読出しのみ可能です。書込みしてもビット値は変化しません。

[bit8] - : 予約ビット

本ビットは予約ビットです。読出し値は "0" です。

必ず "0" を書き込んでください。

[bit7, bit6] CKS1, CKS0 : クロック選択ビット

内部カウントを以下のように選択します。

CKS1	CKS0	カウントクロック選択
0	0	マシクロックの 4 分周クロック [初期値]
0	1	マシクロックの 16 分周クロック
1	0	マシクロックの 32 分周クロック
1	1	設定禁止

- リセット時 : "00_B" に初期化されます。
- 読出し / 書込み可能です。ただし、"11_B" を設定しないでください。

(注意事項) 起動後の書換えは禁止します。必ず起動前か停止後に書き込んでください。

[bit5, bit4] PIS1, PIS0 : パルス幅測定入力端子選択ビット

これらのビットはパルス幅測定入力端子を選択します。

PIS1	PIS0	入力クロック選択
0	0	(PWI0 端子を選択) [初期値]
0	1	設定禁止
1	0	設定禁止
1	1	設定禁止

- リセット時 : "00_B" に初期化されます。
 - 読出し / 書込み可能です。ただし, "00_B" 以外は設定しないでください。
- (注意事項) 起動後の書換えは禁止します。必ず起動前か停止後に書き込んでください。

[bit3] SC : 測定モード (単発 / 連続) 選択ビット

測定モードを以下のように選択します。

SC	測定モード選択	パルス幅測定モード時
0	単発測定モード [初期値]	1 回測定後停止
1	連続測定モード	連続測定 : バッファレジスタ有効

- リセット時 : "0" に初期化されます。
 - 読出し / 書込み可能です。
- (注意事項) 起動後の書換えは禁止します。必ず起動前か停止後に書き込んでください。

[bit2 ~ bit0] MOD2, MOD1, MOD0 : 動作モード / 測定エッジ選択ビット

動作モードおよび幅測定を行うエッジを以下のように選択します。

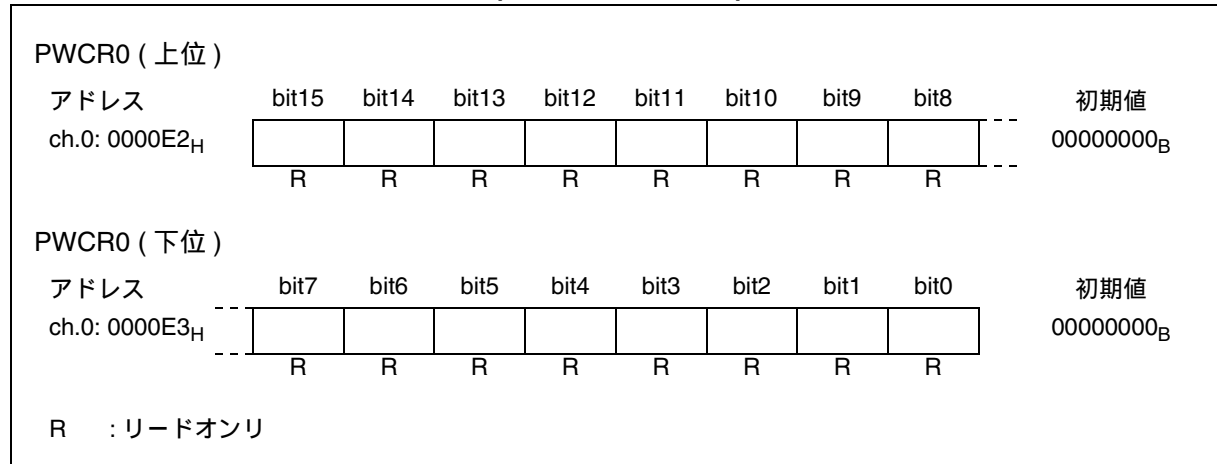
MOD2	MOD1	MOD0	動作モード / 測定エッジ選択
0	0	0	全エッジ間パルス幅測定モード (or ~ or) [初期値]
0	0	1	分周周期測定モード (入力分周器有効)
0	1	0	立上りエッジ間周期測定モード (~)
0	1	1	"H" パルス幅測定モード (~)
1	0	0	"L" パルス幅測定モード (~)
1	0	1	立下りエッジ間周期測定モード (~)
1	1	0	設定禁止
1	1	1	

- リセット時 : "000_B" に初期化されます。

- 読出し / 書込み可能です。

(注意事項) 起動後の書換えは禁止します。必ず起動前か停止後に書き込んでください。

■ PWC データバッファレジスタ (PWCR: PWCR0)



● パルス幅測定モード

連続測定モード時 (PWCSR レジスタ bit3 : SC=1) は、前回の測定結果を保持するバッファレジスタとなります。この場合は読出しのみ可能で、書込みしてもレジスタ値は変化しません。

単発測定モード時 (PWCSR レジスタ bit3 : SC=0) は、アップカウンタを直接アクセスする窓口となります。この場合も、読出しのみ可能で、書込みしてもレジスタ値は変化しません。読出しは随時可能で、カウント中のカウント値が得られます。測定終了後は、測定結果を保存します。

(注意事項) 本レジスタのアクセスは、必ずハーフワードもしくはワード転送命令で行ってください。

- リセット時 : "0000_H" に初期化されます。
- 読出しのみ可能

■ 分周比制御レジスタ (PDIVR: PDIVR0)

PDIVR0									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0: 0000E9 _H	-	-	-	-	-	DIV2	DIV1	DIV0	XXXXXX000 _B
	-	-	-	-	-	R/W	R/W	R/W	
R/W : リード / ライト可能									
X : 不定									
- : 未定義									

分周周期測定モード (PWCSR レジスタ bit2 ~ bit0:MOD2, MOD1, MOD0=001_B) のときに使用するレジスタでほかのモードでは意味を持ちません。

分周周期測定モード時には、本レジスタにより設定された分周比だけ測定端子に入力されたパルスを分周し、分周後の 1 周期幅を測定します。分周比は以下のように選択します。

DIV2	DIV1	DIV0	分周比選択
0	0	0	$2^1=2$ 分周 [初期値]
0	0	1	$2^2=4$ 分周
0	1	0	$2^3=8$ 分周
0	1	1	$2^4=16$ 分周
1	0	0	$2^5=32$ 分周
1	0	1	$2^6=64$ 分周
1	1	0	$2^7=128$ 分周
1	1	1	$2^8=256$ 分周

- リセット時: "000_B" に初期化されます。

- 読出し / 書込み可能です。

(注意事項) 起動後の書換えは禁止します。必ず起動前か停止後に書き込んでください。

11.4 動作説明

PWC は測定入力端子と 8 ビット入力分周などを組み込んでいます。PWC は、パルス幅測定機能があり、3 種類のカウンタクロックを選択可能です。以下に、パルス幅測定機能における基本機能 / 動作について示します。

● パルス幅測定機能

入力パルスの任意イベント間の時間・周期をカウンタで測定できます。

起動後、設定した測定開始エッジが入力されるまでカウントは行われません。開始エッジを検出するカウンタを "0000_H" にクリア後、カウントアップを開始し、停止エッジを検出するとカウントを停止します。この間のカウンタ値がパルス幅としてレジスタに保存されます。

測定終了時、およびオーバフロー発生時に割込み要求が発生できます。

測定終了後は、測定モードに応じて以下のように動作します。

- 単発測定モード時：動作を停止します。
- 連続測定モード時：カウンタ値をバッファレジスタに転送後、再度測定開始エッジが入力されるまでカウントを停止します。

図 11.4-1 パルス幅測定動作 (単発測定モード / "H" 幅測定)

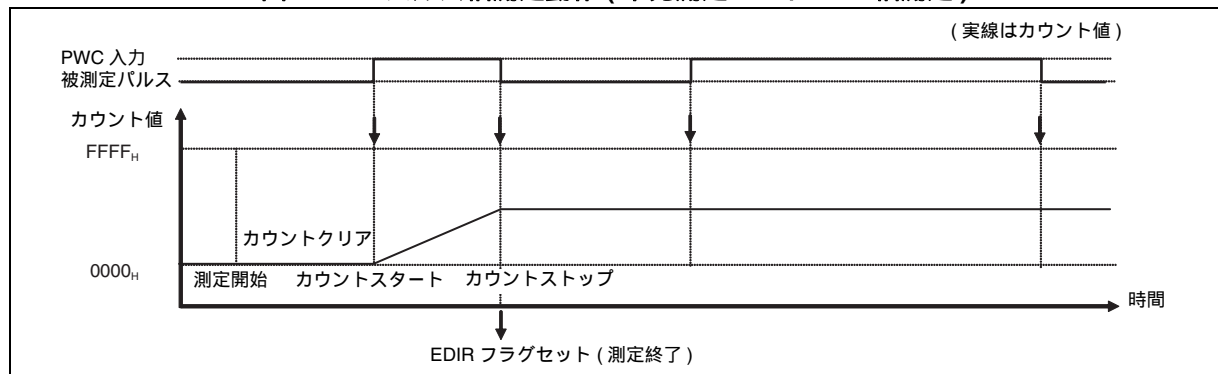
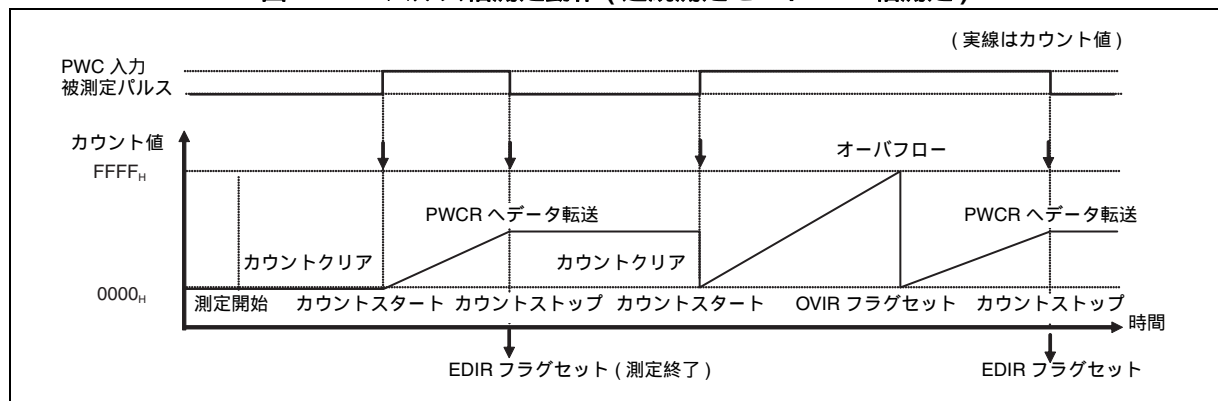


図 11.4-2 パルス幅測定動作 (連続測定モード / "H" 幅測定)



■ カウントクロックの選択

カウンタのカウントクロックは、PWCSR レジスタの bit7, bit6 : CKS1, CKS0 の設定によって、内部クロックソースのうち 3 種類を選択できます。

選択できるカウントクロックは以下のとおりです。

PWCSR	選択される内部カウントクロック
CKS1, CKS0	
00 _B	マシンのクロックの 4 分周 [初期値]
01 _B	マシンのクロックの 16 分周
10 _B	マシンのクロックの 32 分周

リセット後の初期値では、マシンのクロックの 4 分周クロックが選択されています。

(注意事項) カウントクロックの選択は、必ずカウンタ起動前に行ってください。

■ 動作モードの選択

各動作モード / 測定モードの選択は、PWCSR レジスタの設定により行います。

- 動作モードの設定 : PWCSR レジスタ bit2 ~ bit0 : MOD2, MOD1, MOD0
(パルス幅測定モードの選択, 測定エッジの決定など)
- 測定モードの設定 : PWCSR レジスタ bit3 : SC
(単発測定 / 連続測定の選択)

モード設定ビットの組合せによる動作モードの選択の一覧を以下に示します。

動作モード			SC	MOD2	MOD1	MOD0
パルス幅測定	or ~ or 全エッジ間測定	単発測定：バッファ無効	0	0	0	0
		連続測定：バッファ有効	1	0	0	0
	分周周期測定 (1 分周 ~ 256 分周)	単発測定：バッファ無効	0	0	0	1
		連続測定：バッファ有効	1	0	0	1
	~ 立上り間周 期測定	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
	~ "H" パルス幅測定	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
	~ "L" パルス幅測定	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
~ 立下り間周 期測定	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	
設定禁止			0	1	1	0
			1	1	1	0
			0	1	1	1
			1	1	1	1

リセット後の初期値では、全エッジ間測定 - 単発測定モードが選択されています。

(注意事項) 起動モードの選択は、必ずカウンタ起動前に行ってください。

■ パルス幅測定の起動と停止

各動作の起動 / 再起動 / 強制停止は、PWCSR レジスタの bit15, bit14 : STRT, STOP ビットにより行います。

パルス幅測定の起動 / 再起動は STRT ビット、強制停止は STOP ビットと機能が分かれており、それぞれ、"0" を書き込むことにより機能しますが、この際、両ビットに書き込む値が排他でないと機能しません。ビット操作命令以外の命令にて（バイト単位以上にて）書き込む際は、必ず以下に示す組合せを書き込んでください。

機能	STRT	STOP
パルス幅測定の起動 / 再起動	0	1
パルス幅測定の強制停止	1	0

ビット操作命令（ビットクリア命令）を用いる場合、ハードウェアにより自動的に上記組合せにて書き込まれるため、特に意識する必要がありません。

● 起動後の動作

パルス幅測定モードの起動後の動作は、以下のとおりです。

- パルス幅測定モード：測定開始エッジが入力されるまでカウントは行われません。測定開始エッジ検出後、16 ビットアップカウンタを "0000_H" にクリアし、カウントを開始します。

● 再起動について

パルス幅測定起動後、動作中に起動をかける（STRT ビットに "0" を書き込む）ことを再起動とよびます。再起動をかけると、以下の動作が行われます。

- 測定開始エッジ待ち状態の場合、動作に影響はありません。測定中の場合、カウントを停止し、再度測定開始エッジ待ち状態となります。この際、測定終了エッジ検出と再起動が同時になると、測定終了フラグ (EDIR) がセットされ、連続測定モード時は測定結果が PWCR に転送されます。

● 停止について

単発測定モードでは、カウンタのオーバフローまたは測定終了により、自動的にカウント動作を停止しますので、特に意識する必要はありません。それ以外のモードや自動停止する前に停止させたい場合は、強制停止させる必要があります。

● 動作状態の確認

前述 STRT, STOP ビットは読出し時には動作状態表示ビットとして機能します。表示される値は以下の内容を示します。

STRT	STOP	動作状態
0	0	カウンタ停止中（測定開始エッジ待ち状態を除く） 起動されていないか、または測定が終了したことを示します。
1	1	カウント動作中、または測定開始エッジ待ち状態

(注意事項) STRT, STOP のどちらのビットを読み出しても同じ値となります。ただし、リードモディファイライト (RMW) 系命令 (ビット処理命令など) にて本ビットを読み出すと常に "11_B" になるため、これらの命令を使用して読み出さないでください。

■ カウンタのクリア

16 ビットアップカウンタは、以下に示す場合に "0000_H" にクリアされます。

- リセット時
- パルス幅測定モードにて、測定開始エッジを検出しカウントを開始する場合

■ パルス幅測定動作詳細

● 単発測定と連続測定

パルス幅測定には、1 回のみの測定を行うモードと、連続して測定を行うモードがあります。各モードは PWCSR の SC ビットによって選択します (動作モードの選択参照)。両モードにおける相違点は以下のとおりです。

- 単発測定モード : 1 回目の測定終了エッジが入力されるとカウンタのカウントは停止し、PWCSR 中の測定終了フラグ (EDIR) がセットされ、以降の測定は行われません。
ただし、同時に再起動がかかった場合は測定開始待ち状態となります。
- 連続測定モード : 測定終了エッジが入力されるとカウンタのカウントは停止し、PWCSR 中の測定終了フラグ (EDIR) がセットされ、再度測定開始エッジが入力されるまでカウントを停止します。再度、測定開始エッジが入力されるとカウンタを "0000_H" にクリアした後、測定を開始します。測定終了時、カウンタの測定結果は PWCR に転送されます。

(注意事項) 測定モードの選択 / 変更は、必ずカウンタ停止中に行ってください。

● 測定結果データ

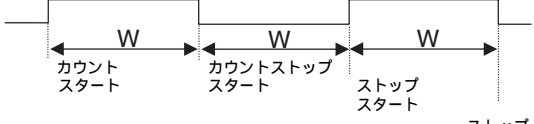
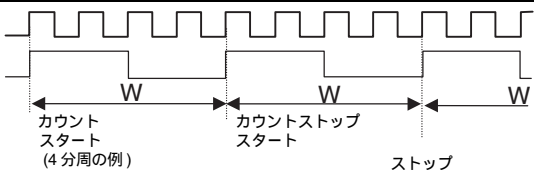
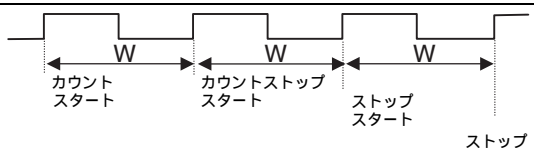
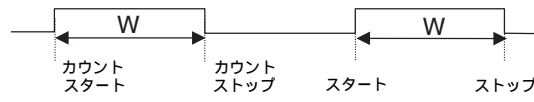
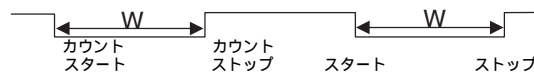
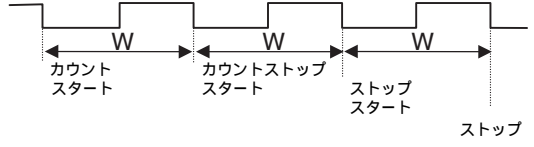
単発測定モードと連続測定モードでは、測定結果とカウンタ値の扱い、PWCR の機能に違いがあります。両モードにおける測定結果の相違点は以下のとおりです。

- 単発測定モード : PWCR を動作中に読み出すと測定中のカウント値が得られます。
PWCR を測定終了後に読み出すと測定結果データが得られます。
- 連続測定モード : 測定終了時、カウンタ内の測定結果は PWCR に転送されます。
PWCR を読み出すと直前の測定結果が得られ、測定動作中も前回の測定結果を保持しています。測定中のカウント値は読み出せません。

連続測定モードにて、測定結果を読み出さないうちに次の測定が終了してしまった場合、前回の測定結果は新しい測定結果に消されてしまいます。この際、PWCSR 中のエラーフラグ (ERR) がセットされます。エラーフラグ (ERR) は、PWCR を読み出すと自動的にクリアされます。

● 測定モードとカウント動作

入力されたパルスのどこを測定するかによって、測定モードは6種類のうちから選択できます。さらに、高い周波数のパルスの幅を精度よく測定できるように、入力されたパルスを任意分周して周期を測定するモードも用意されています。以下に、各モードについて説明します。

測定モード	MOD2	MOD1	MOD0	測定内容 (W：測定するパルス幅)
全エッジ間パルス幅測定	0	0	0	 <p>連続して入力されるエッジ間の幅を測定します。 カウント (測定) 開始：エッジ検出時 カウント (測定) 終了：エッジ検出時</p>
分周周期測定	0	0	1	 <p>分周比設定レジスタ PDIVR で選択した分周比だけ入力パルスを分周してその周期を測定します。 カウント (測定) 開始：起動直後の立上りエッジ検出時 カウント (測定) 終了：分周後の 1 周期終了時</p>
立上りエッジ間周期測定	0	1	0	 <p>立上りエッジ間の周期を測定します。 カウント (測定) 開始：立上りエッジ検出時 カウント (測定) 終了：立上りエッジ検出時</p>
"H" パルス幅測定	0	1	1	 <p>"H" 期間の幅を測定します。 カウント (測定) 開始：立上りエッジ検出時 カウント (測定) 終了：立下りエッジ検出時</p>
"L" パルス幅測定	1	0	0	 <p>"L" 期間の幅を測定します。 カウント (測定) 開始：立下りエッジ検出時 カウント (測定) 終了：立上りエッジ検出時</p>
立下りエッジ間周期測定	1	0	1	 <p>立下りエッジ間の周期を測定します。 カウント (測定) 開始：立下りエッジ検出時 カウント (測定) 終了：立下りエッジ検出時</p>

どのモードでも、測定起動後、測定開始エッジが入力されるまではカウンタはカウント動作を行いません。測定開始エッジ入力されると、カウンタは "0000_H" にクリアされた後、測定終了エッジが入力されるまでの間、カウントクロックごとにアップカウントを続けます。

測定終了エッジが入力されると、以下の動作が行われます。

- PWCSR 中の測定終了フラグ (EDIR) がセットされます。
- カウンタのカウント動作が停止します (再起動と同時であった場合を除く)。
- 連続測定モード時 : カウンタの値 (=測定結果) が PWCR に転送され、次の測定開始エッジが入力されるまでカウントを停止して待ちます。
- 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などを行った場合、終了エッジが次の測定開始エッジとなります。

● 最小入力パルス幅について

パルス幅測定入力端子 (PWI0) に入力できるパルスには以下の制限があります。

- 最小入力幅 : マシンサイクル × 4 以上
(16 MHz のマシンクロックの場合は、0.25 μs 以上とする)

上記パルスより小さい幅のパルスを入力した場合の動作は保証できません。

● パルス幅 / 周期算出方法

測定終了後、PWCR に得られた測定結果データから、被測定パルス幅 / 周期算出方法は以下のように求められます。

$T_W = n \times t / D_{IV} [\mu s]$	T_W : 被測定パルス幅 / 周期 [μs]
	n : PWCR 内の測定結果データ
	t : カウントクロックの周期 [μs]
	D_{IV} : 分周比レジスタ PDIVR で選択した分周比
	(分周周期測定モード以外は "1" を代入)

● パルス幅 / 周期測定レンジ

カウントクロック、および入力分周器の分周比の選択の組合せにより、測定可能なパルス幅 / 周期のレンジが変化します。

一例として、マシクロック (以下 ϕ とする) = 16 MHz 時の測定レンジの一覧表を以下に示します。

分周比	DIV2, DIV1, DIV0	CKS1, CKS0=00 _B ($\phi/4$) 時	CKS1, CKS0=01 _B ($\phi/16$) 時	CKS1, CKS0=10 _B ($\phi/32$) 時
分周なし	-	0.25 μ s ~ 16.4 ms [250 ns]	0.25 μ s ~ 65.5 ms [1.0 μ s]	0.25 μ s ~ 131 ms [2.0 μ s]
2 分周	000 _B	0.25 μ s ~ 8.19 ms [125 ns]	0.25 μ s ~ 32.8 ms [0.5 μ s]	0.25 μ s ~ 65.5 ms [1.0 μ s]
4 分周	001 _B	0.25 μ s ~ 4.10 ms [62.5 ns]	0.25 μ s ~ 16.4 ms [250 ns]	0.25 μ s ~ 32.8 ms [0.5 μ s]
8 分周	010 _B	0.25 μ s ~ 2.05 ms [31.25 ns]	0.25 μ s ~ 8.19 ms [125 ns]	0.25 μ s ~ 16.4 ms [250 ns]
16 分周	011 _B	0.25 μ s ~ 1.02 ms [15.6 ns]	0.25 μ s ~ 4.10 ms [62.5 ns]	0.25 μ s ~ 8.19 ms [125 ns]
64 分周	100 _B	0.25 μ s ~ 256 μ s [3.91 ns]	0.25 μ s ~ 1.024 ms [15.6 ns]	0.25 μ s ~ 2.05 ms [31.25 ns]
256 分周	101 _B	0.25 μ s ~ 64.0 μ s [0.98 ns]	0.25 μ s ~ 256 μ s [3.91 ns]	0.25 μ s ~ 512 μ s [7.81 ns]
-	その他	設定禁止		

(注意事項) ・ マシクロック ϕ = 16 MHz 時

・ [] 内は 1 ビットあたりの分解能を示します。

● 割込み要求発生

パルス幅測定モードにおいては、以下の 2 つの割込み要求を発生することが可能です。

- カウンタのオーバフローによる割込み要求

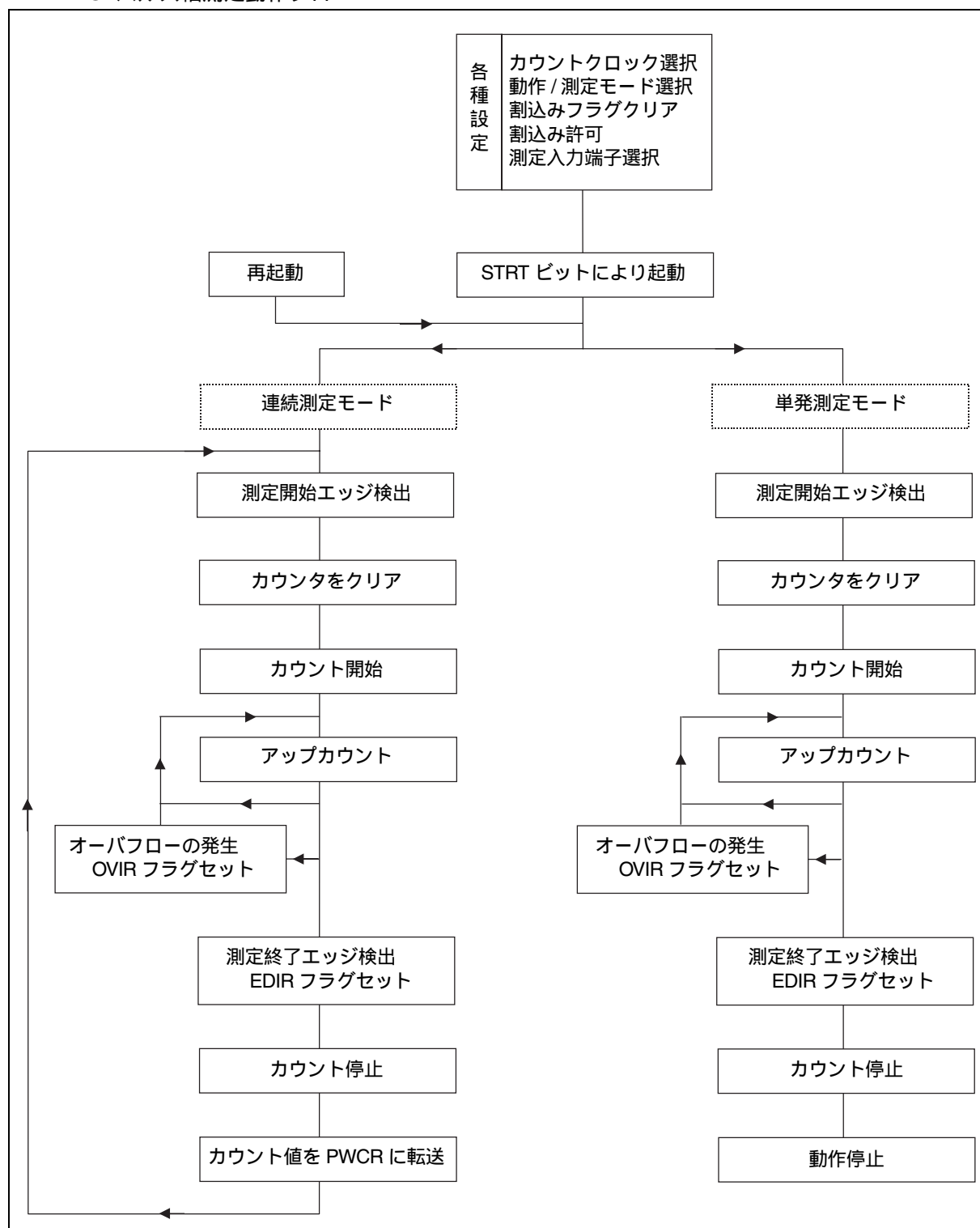
測定中、カウントアップによりオーバフローが発生するとオーバフローフラグがセットされ、オーバフロー割込み要求が許可されていると割込み要求が発生します。

- 測定終了による割込み要求

測定終了エッジを検出すると、PWCSR 中の測定終了フラグ (EDIR) がセットされ、測定終了割込み要求が許可されていると割込み要求が発生します。

測定終了フラグ (EDIR) は、測定結果 PWCR を読み出すと自動的にクリアされます。

● パルス幅測定動作フロー



■ 注意事項

● レジスタ書換えに関する注意事項

PWCSR レジスタの以下に示すビットは動作中に書き換えることを禁止します。書換えは必ず起動前か停止後に行ってください。

[bit7, bit6] CKS1, CKS0	: クロック選択ビット
[bit5, bit4] PIS1, PIS0	: パルス幅測定入力端子選択ビット
[bit3] SC	: 測定モード (単発 / 連続) 選択ビット
[bit2 ~ bit0] MOD2, MOD1, MOD0	: 動作モード / 測定エッジ選択ビット

PDIVR レジスタは動作中に書き換えることを禁止します。書換えは必ず起動前か停止後に行ってください。

● PWCSR レジスタの STRT, STOP ビットについて

両ビット共に、書込み時と読出し時では、意味が異なるので注意してください
(PWC コントロール / ステータスレジスタ (PWCSR: PWCSR0) 参照)。

また、リードモディファイライト (RMW) 系命令における読出し値は、ビット値にかかわらず、"11_B" です。このため、動作状態の読出しには、ビット処理命令は使用できません (読み出すと必ず動作中となります) ので注意してください。

カウンタの起動 / 停止のための STRT, STOP ビットへの書込みは、それぞれのビットに対するビット処理命令 (ビットクリア命令など) を用いることが可能です。

● カウンタのクリアについて

パルス幅測定モードの場合、測定開始エッジでカウンタがクリアされますので、起動前にカウンタ中にあったデータは無効になります。

● 最小入力パルス幅について

パルス幅測定入力端子に入力できるパルスには、以下の制限があります。

- 最小入力幅 : マシンサイクル × 4 (マシンサイクルが 62.5 ns 時, 250 ns)
- 最大入力周波数 : マシンクロックの 4 分周 (マシンサイクルが 16 MHz 時, 4 MHz)

上記パルスより小さい幅、高い周波数のパルスを入力した場合の動作は保証できません。入力信号にそのようなノイズがのる可能性がある場合は、チップ外部でフィルタなどを通して除去した後、入力してください。

● 分周周期測定モードについて

パルス幅測定モードのうちの分周周期測定モードでは、入力パルスを分周するため、測定結果より算出して得られるパルス幅は平均値となりますので注意してください。

● クロック選択ビットについて

PWCSR レジスタの bit7, bit6 CKS1, CKS0 : クロック選択ビットにおいて "11_B" は設定禁止です。

● 予約ビットについて

PWCSR レジスタの bit8 は予約ビットになっています。このビットに書込みを行う場合は、必ず "0" にしてください。

● 動作中の再起動について

カウント動作を開始した後に再起動を行う場合は、そのタイミングによっては以下に示すようなことが起こり得ます。

- パルス幅単発測定モード時、測定終了エッジと同時であった場合
再起動を行い、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) はセットされます。
- パルス幅連続測定モード時、測定終了エッジと同時であった場合
再起動を行い、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) はセットされ、その時点での測定結果は PWCR に転送されます。

以上のように、動作中の再起動時には、フラグの動作に注意して割込み制御などを行うようにしてください。

第12章

多機能タイマ

多機能タイマの概要，レジスタの構成 / 機能および動作について説明します。

- 12.1 概要
- 12.2 ブロックダイアグラム
- 12.3 多機能タイマの端子
- 12.4 多機能タイマのレジスタ
- 12.5 多機能タイマ割込み
- 12.6 多機能タイマの動作
- 12.7 多機能タイマの使用上の注意
- 12.8 多機能タイマのプログラム例

12.1 概要

多機能タイマは、3 つの 16 ビットフリーランタイマ、6 つの 16 ビットアウトプットコンペア、4 つの 16 ビットインプットキャプチャ、4 チャンネルの 8/16 ビット PPG タイマ、および 1 つの波形ジェネレータ、2 つの A/D 起動コンペアから構成されています。この波形ジェネレータを使用すると、6 つの別々の波形を 16 ビットフリーランタイマから出力することができ、また、入力パルス幅と外部クロックサイクルを測定することもできます。

■ 多機能タイマの構成

● 16 ビットフリーランタイマ (× 3)

- 16 ビットフリーランタイマは 16 ビットアップ / ダウンカウンタ、制御レジスタ、16 ビットコンペアクリアレジスタ (バッファレジスタを持っています)、およびプリスケラから構成されています。
- 9 種類のカウンタ動作クロック (ϕ , $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, $\phi/64$, $\phi/128$, $\phi/256$) を選択することができます (ϕ : マシンクロック)。
- コンペアクリア割込みは、コンペアクリアレジスタと 16 ビットフリーランタイマが比較され、一致した場合に生成されます。0 検出割込みは、16 ビットフリーランタイマがカウント値 "0" を検出している間に生成されます。
- コンペアクリアレジスタは、選択可能なバッファレジスタを持っています (このバッファレジスタに書き込まれたデータはコンペアクリアレジスタへ転送されます)。16 ビットフリーランタイマが停止し、バッファにデータが書き込まれると、転送は直ちに実行されます。16 ビットフリーランタイマの動作中にタイマ値 "0" が検出されると、バッファからデータが転送されます。
- アップカウントモードにおいてリセットやソフトウェアクリア、あるいはコンペアクリアレジスタとのコンペア一致が発生すると、カウンタ値は "0000_H" にリセットされます。
- このカウンタの出力値は、多機能タイマのアウトプットコンペアとインプットキャプチャのクロックカウントとして使用することができます。
- 0 検出またはコンペア一致時に、A/D 起動が可能です。

● 16 ビットアウトプットコンペア (× 6)

- 16 ビットアウトプットコンペアは、6 つの 16 ビットコンペアレジスタ (選択可能なバッファレジスタを持っています)、コンペア出力ラッチ、コンペア制御レジスタから構成されています。選択された 1 つの 16 ビットフリーランタイマ値とコンペアレジスタが一致すると、割込みが生成され、出力レベルが反転します。
- 6 つのコンペアレジスタは、別々に動作させることができます。出力端子と割込みフラグは各コンペアレジスタに対応しています。
- 2 つのコンペアレジスタを対 (ペア) にして出力端子を制御することができます。2 つのコンペアレジスタを一緒に使用することによって出力端子を反転させます。
- 各出力端子の初期値を設定することができます。
- 割込みはアウトプットコンペアレジスタが 16 ビットフリーランタイマと一致した場合に生成されます。

● 16 ビットインプットキャプチャ (× 4)

- インプットキャプチャは、4つの独立した外部入力端子と、この端子に対応するキャプチャレジスタおよびキャプチャ制御レジスタから構成されています。外部端子において入力信号のエッジを検出すると、選択された1つの16ビットフリーランタイムの値をキャプチャレジスタへ格納することができ、また、割込みも同時に生成されます。
- 外部入力信号の3種類のトリガエッジ (立上りエッジ、立下りエッジ、およびその両方のエッジ) を選択することができ、また、トリガエッジが立上りエッジであるか立下りエッジであるかを示すレジスタを持っています。
- 4つのインプットキャプチャを別々に動作させることができます。
- 割込みは外部入力からの有効エッジが検出されると生成されます。

● 8/16 ビット PPG タイマ (× 4)

PPG タイマ 0 は、波形ジェネレータへ PPG 信号を供給するために使用します。PPG タイマ 0 の詳細については、「第 10 章 PPG」を参照してください。

● 波形ジェネレータ

- 波形ジェネレータは、3つの16ビットデッドタイムレジスタ、3つのタイマ制御レジスタ、および1つの16ビット波形制御レジスタから構成されています。
- 波形ジェネレータは、リアルタイム出力、16ビットPPG波形出力、ノンオーバーラップ3相波形出力 (インバータ制御用)、およびDCチョッパ波形出力を生成することができます。
- 16ビットデッドタイムのデッドタイムに基づいて、ノンオーバーラップ波形出力を生成することができます (デッドタイムタイマ機能)。
- 2チャンネルモード時にリアルタイムアウトプットを動作させることにより、ノンオーバーラップ波形出力を生成することができます (デッドタイムタイマ機能)。
- リアルタイムアウトプットコンペア一致を検出すると、GATE信号が生成され、この信号によりPPGタイマの動作が開始または停止します (GATE 機能)。
- リアルタイムアウトプットコンペア一致が検出されると、16ビットデッドタイムがアクティブになり、PPG動作の制御用GATE信号を生成することによって、PPGタイマ0を容易に開始または停止させることができます (GATE 機能)。
- DTTI 端子を使用することによって、強制的に停止を制御することができます。
- DTTI レジスタにより、強制的に停止を制御することも可能です。

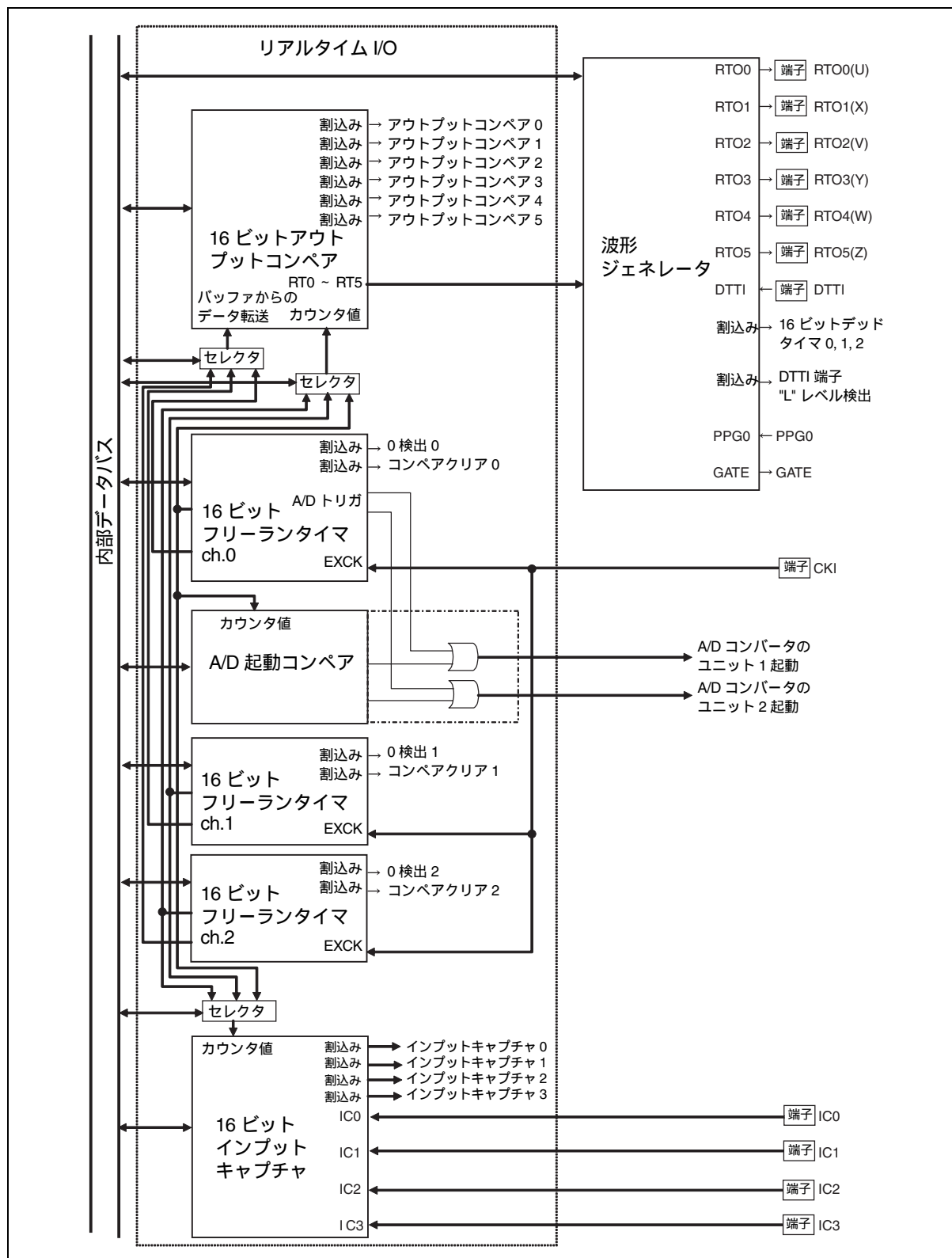
● A/D 起動コンペア (× 2)

- ch.0 の16ビットフリーランタイム値とコンペアレジスタが一致したときに、A/Dを起動することができます (ch.1, ch.2 の16ビットフリーランタイムでは起動できません)。
- コンペア 1 が A/D1 を起動できます。
- コンペア 2 が A/D2 を起動できます。
- 16ビットフリーランタイムのアップカウント時の一致した場合のみの起動を指定できます。
- 16ビットフリーランタイムのダウンカウント時の一致した場合のみの起動を指定できます。
- 16ビットフリーランタイムのアップダウンカウントの両方で一致した場合の起動を指定できます。

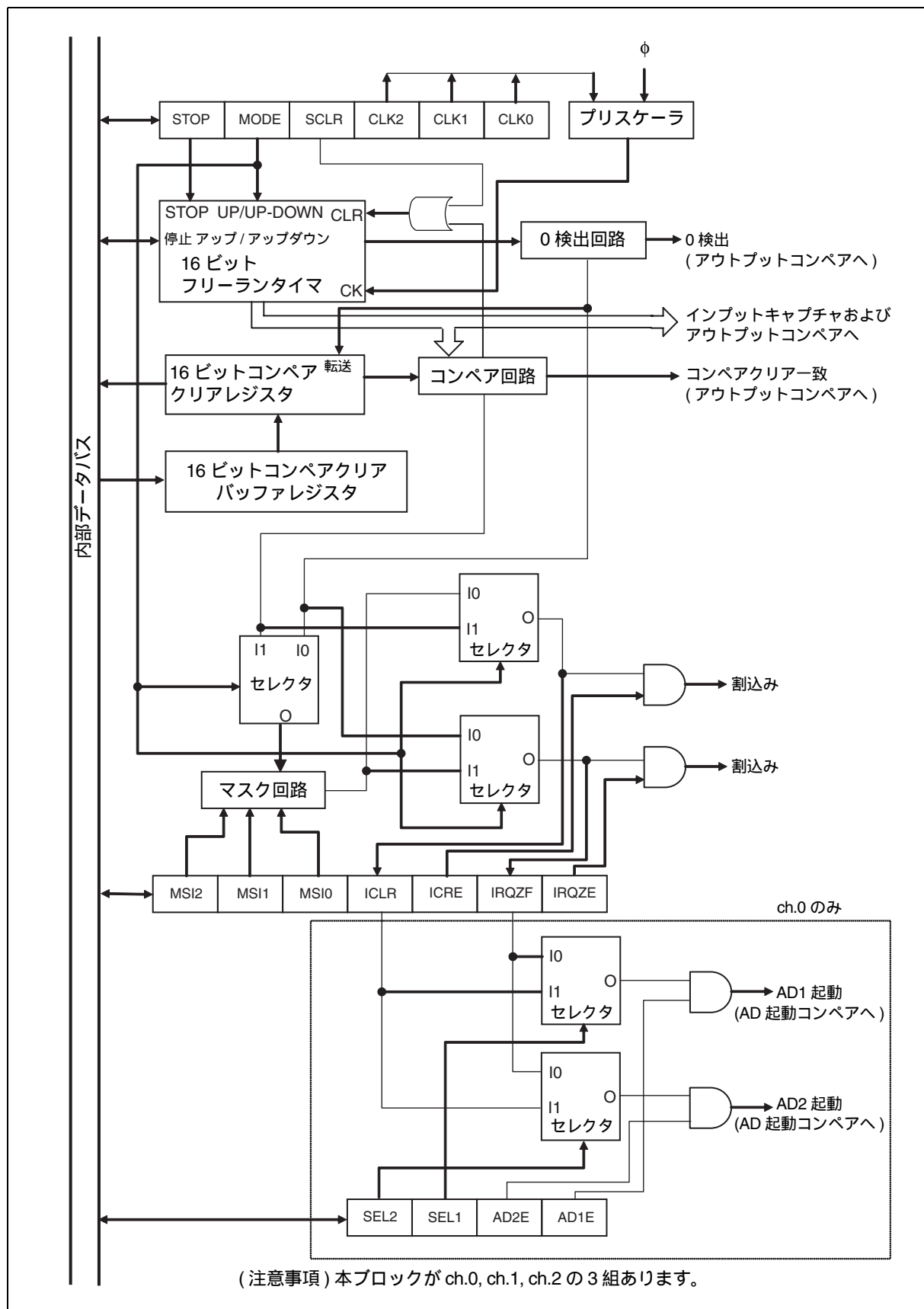
12.2 ブロックダイアグラム

多機能タイマのブロックダイアグラムを示します。

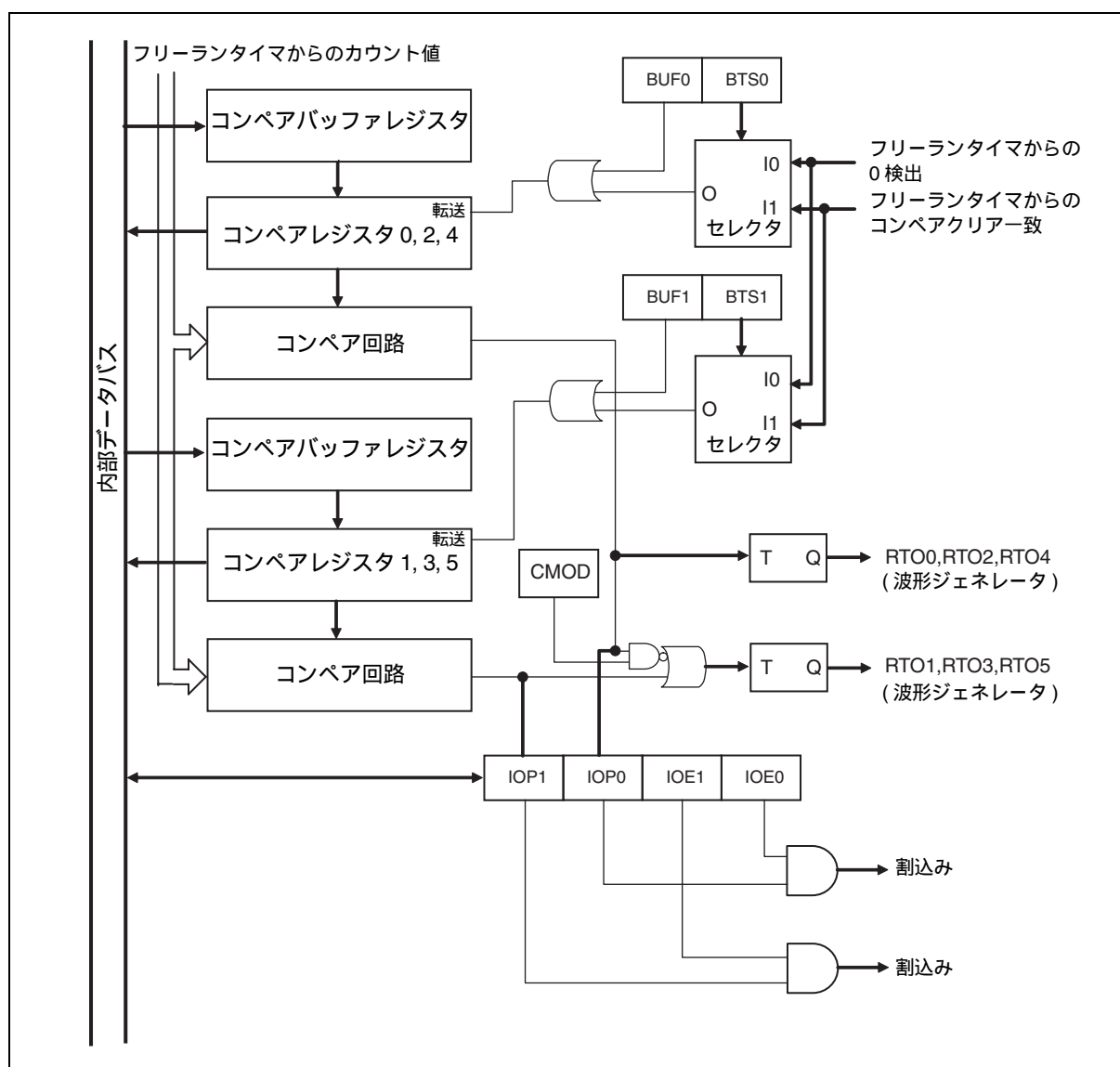
■ 多機能タイマのブロックダイアグラム



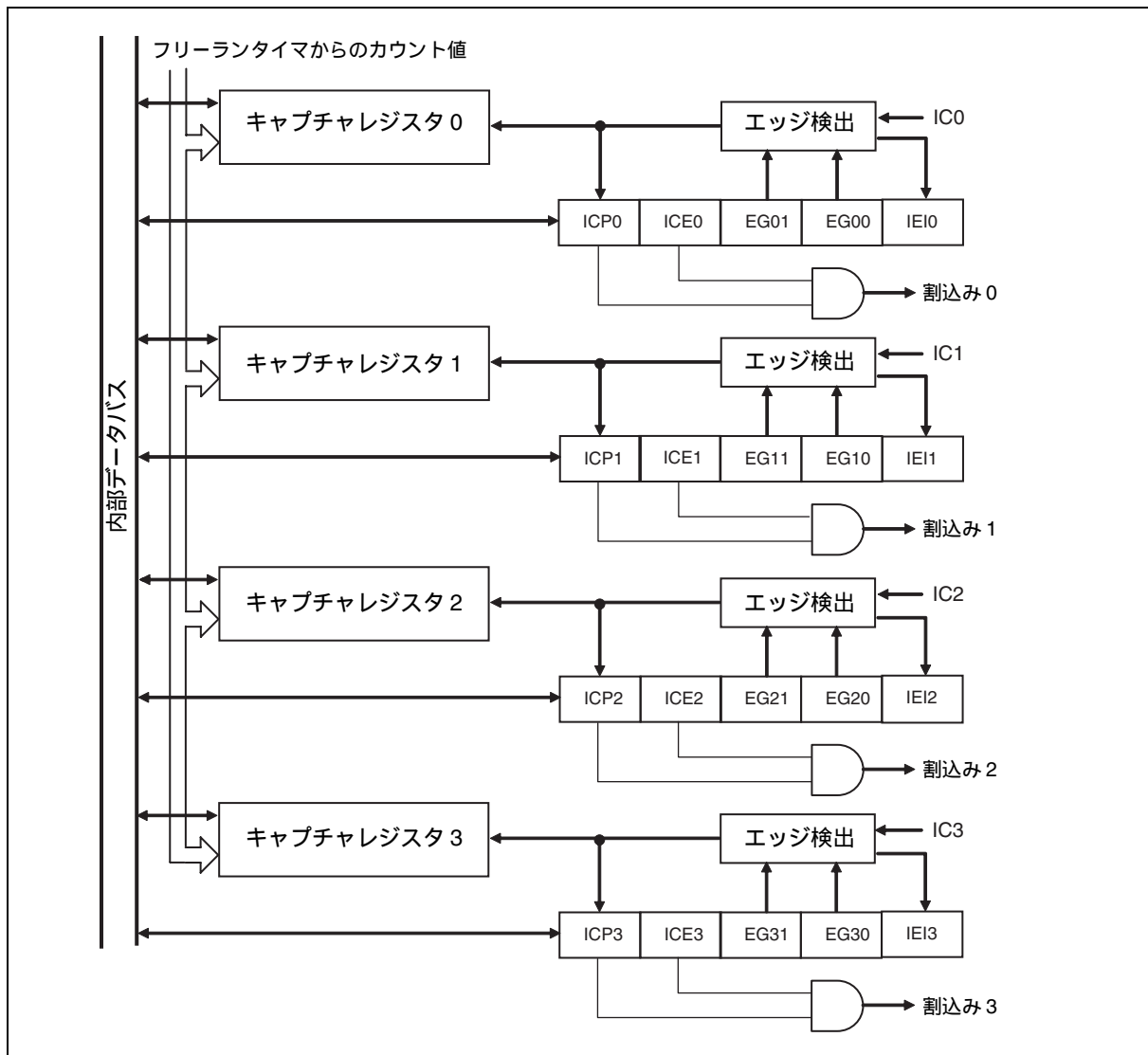
■ 16 ビットフリーランタイムのブロックダイアグラム



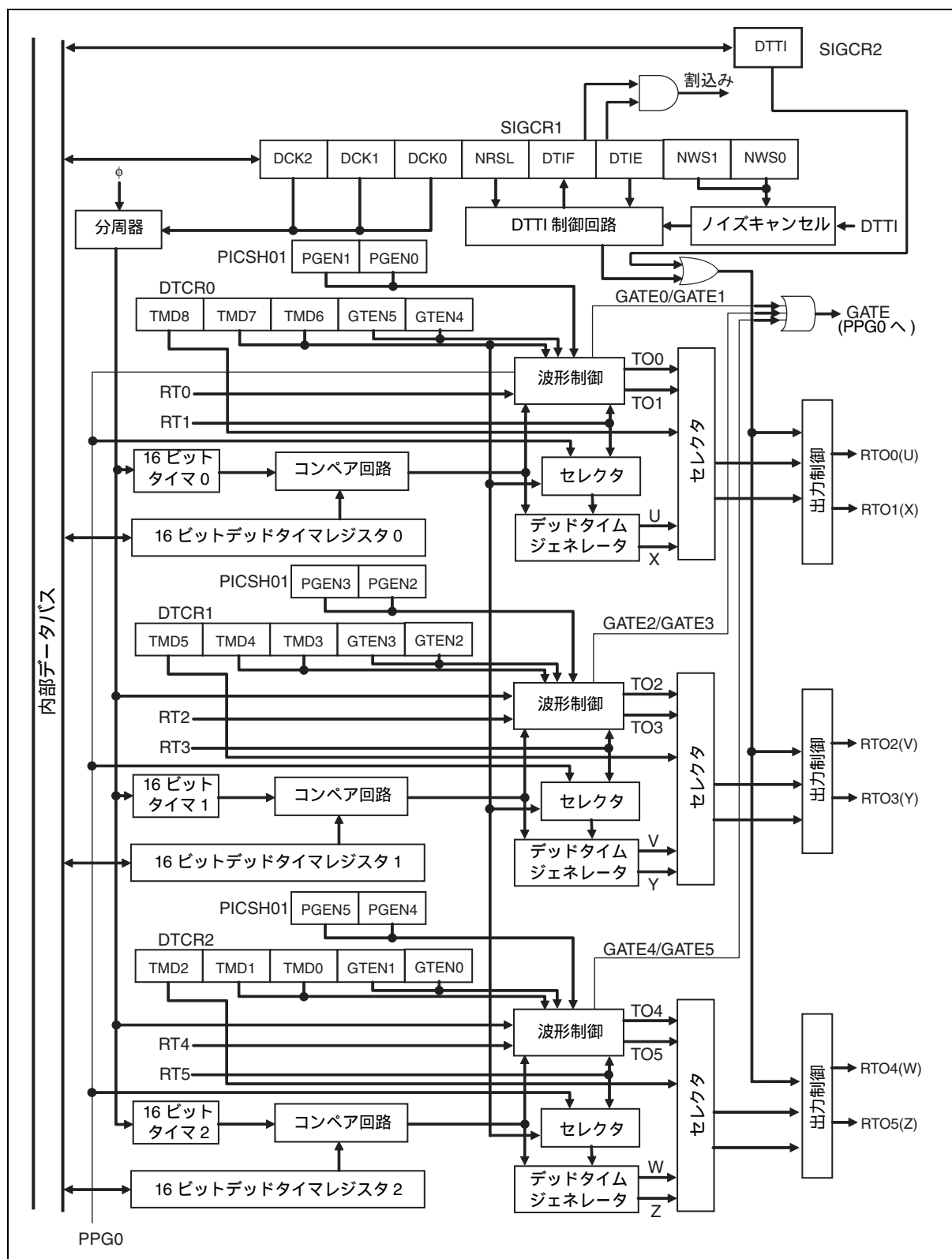
■ 16 ビットアウトプットコンペアのブロックダイアグラム



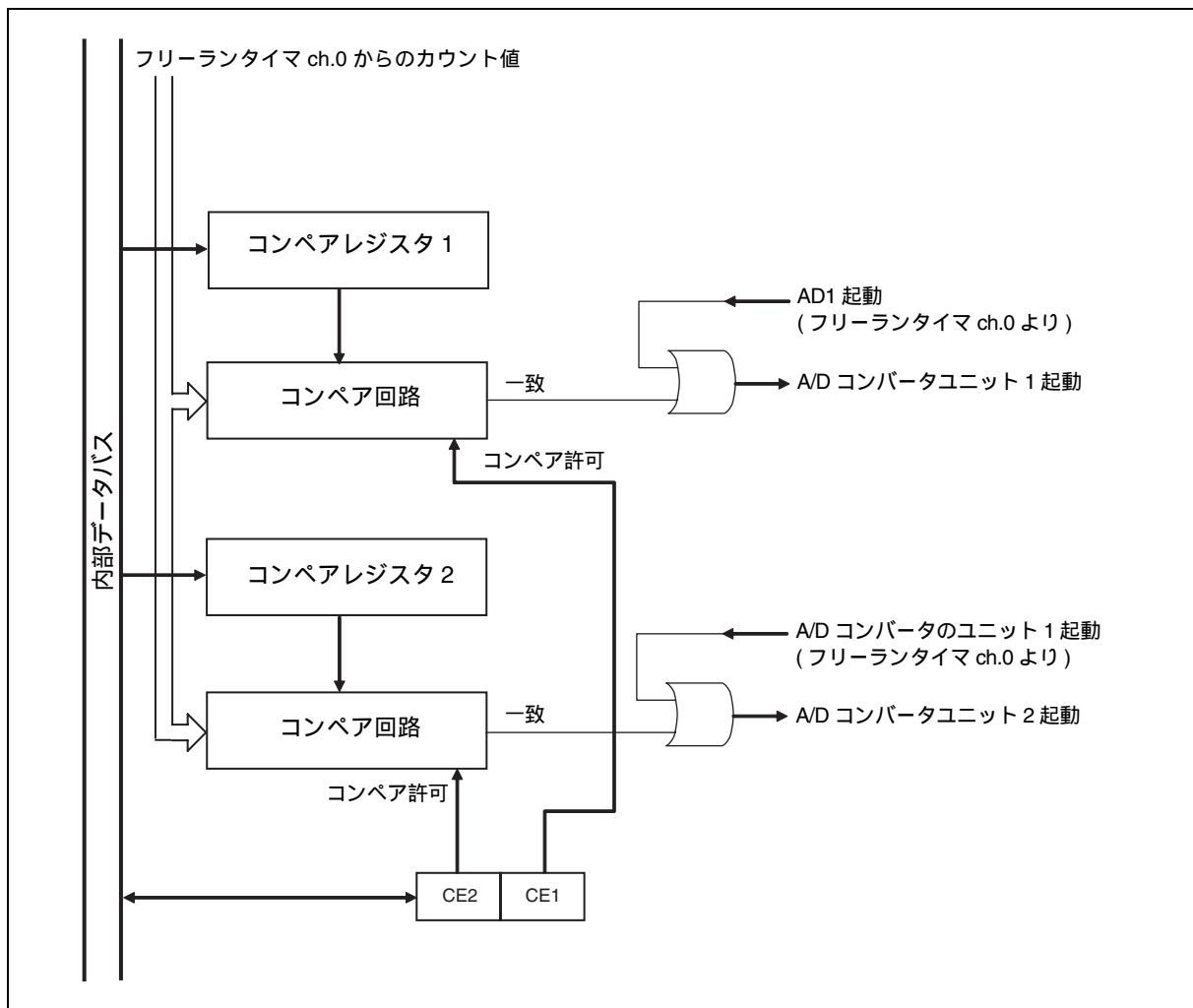
■ 16 ビットインプットキャプチャのブロックダイアグラム



■ 波形ジェネレータのブロックダイアグラム

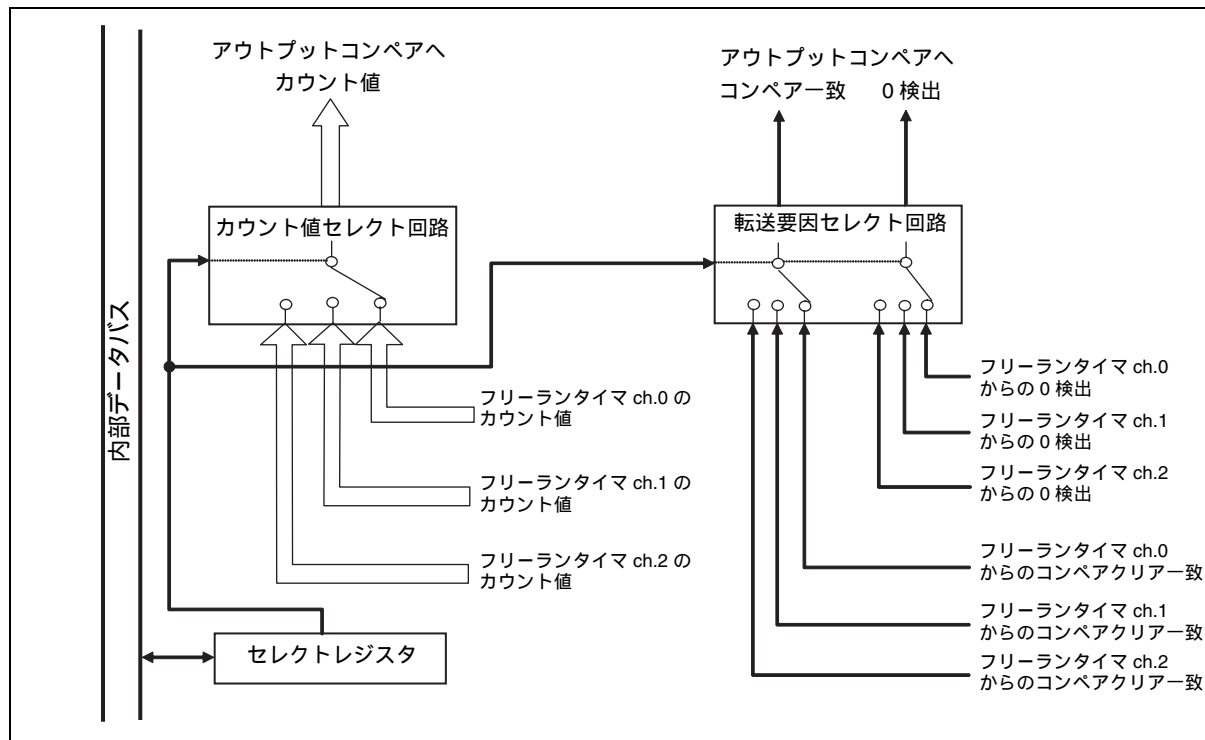


■ A/D 起動コンペアのブロックダイアグラム

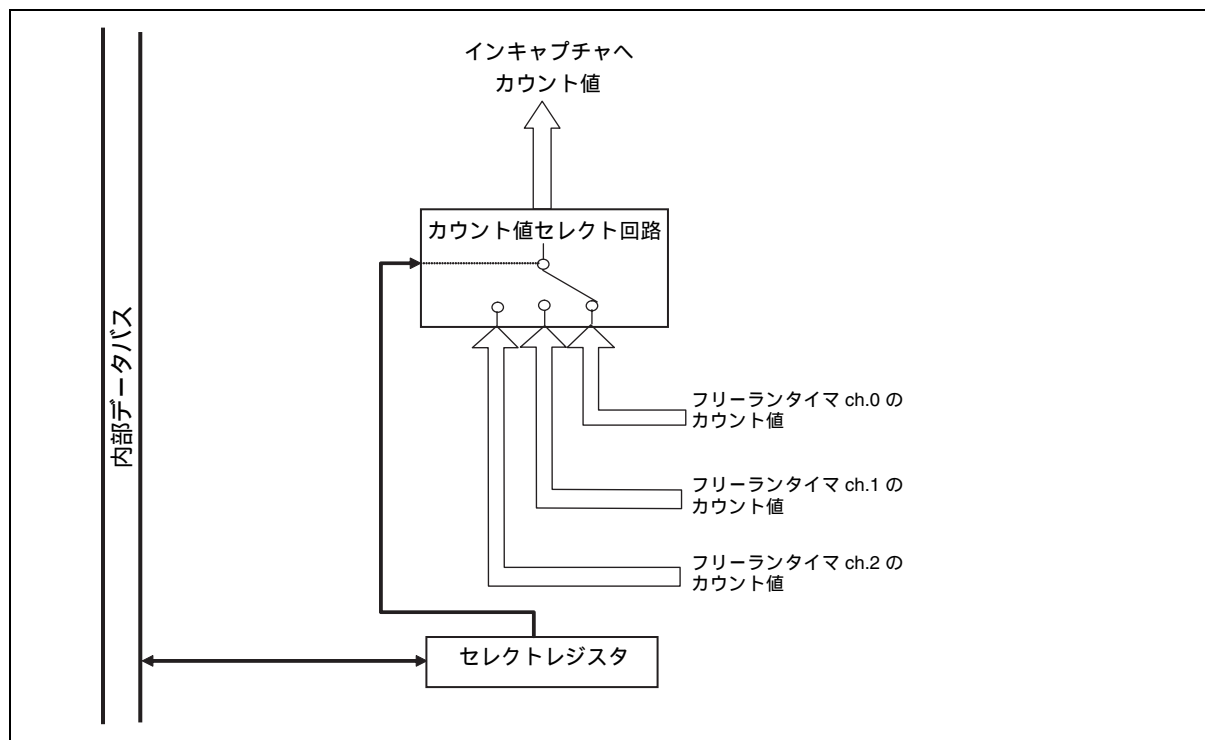


■ フリーランタイムセクタのブロックダイアグラム

- アウトプットコンペア用
(アウトプットコンペアのチャンネルごとに本ブロックはあります。)



- インプットキャプチャ用
(インプットキャプチャのチャンネルごとに本ブロックはあります。)



12.3 多機能タイマの端子

多機能タイマの端子について説明します。

■ 多機能タイマの端子

表 12.3-1 多機能タイマの端子

端子名	端子機能	I/O 形式	プルアップ オプション	スタンバイ 制御	端子設定
P23/DTTI	ポート 2 入出力, DTTI 入力	CMOS 出力, CMOS ヒス テリシス 入力	選択可能	あり	端子を入力ポートとして 設定する (DDR2:bit3=0)
P24/CKI	ポート 2 入出力, 外部クロック				端子を入力ポートとして 設定する (DDR2:bit4=0)
P25/IC0	ポート 2 入出力, インプットキャ プチャ 0				端子を入力ポートとして 設定する (DDR2:bit5=0)
P26/IC1	ポート 2 入出力, インプットキャ プチャ 1				端子を入力ポートとして 設定する (DDR2:bit6=0)
P20/IC2/ ADTG1	ポート 2 入出力, インプットキャ プチャ 2, AD 外 部トリガ入力 1				端子を入力ポートとして 設定する (DDR2:bit0=0)
P21/IC3/ ADTG2	ポート 2 入出力, インプットキャ プチャ 3, AD 外 部トリガ入力 2				端子を入力ポートとして 設定する (DDR2:bit1=0)
RTO0 (U)	ポート 3 入出力, RTO0		なし		RTO0 出力を設定する (OCSH1:OTE0=1)
RTO1 (X)	ポート 3 入出力, RTO1				RTO1 出力を設定する (OCSH1:OTE1=1)
RTO2 (V)	ポート 3 入出力, RTO2				RTO2 出力を設定する (OCSH3:OTE0=1)
RTO3 (Y)	ポート 3 入出力, RTO3				RTO3 出力を設定する (OCSH3:OTE1=1)
RTO4 (W)	ポート 3 入出力, RTO4				RTO4 出力を設定する (OCSH5:OTE0=1)
RTO5 (Z)	ポート 3 入出力, RTO5				RTO5 出力を設定する (OCSH5:OTE1=1)

DDR_x : ポート方向レジスタ

OCSH_x : コンペア制御レジスタ

12.4 多機能タイマのレジスタ

多機能タイマのレジスタについて説明します。

■ 16 ビットフリーランタイマのレジスタ

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (上位)

CPCLRBH0 ~ CPCLRBH2/CPCLRH0 ~ CPCLRH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A4 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	11111111 _B
000154 _H	W	W	W	W	W	W	W	W	CPCLRBH ライト
00015C _H	R	R	R	R	R	R	R	R	CPCLRH リード

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (下位)

CPCLRBL0 ~ CPCLRBL2/CPCLRL0 ~ CPCLRL2

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	11111111 _B
	W	W	W	W	W	W	W	W	CPCLRBH ライト
	R	R	R	R	R	R	R	R	CPCLRH リード

タイマデータレジスタ (上位)

TCDTH0 ~ TCDTH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A6 _H	T15	T14	T13	T12	T11	T10	T09	T08	00000000 _B
000156 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
00015E _H									

タイマデータレジスタ (下位)

TCDTL0 ~ TCDTL2

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	T07	T06	T05	T04	T03	T02	T01	T00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

(続く)

(続き)

タイマ状態制御レジスタ (上位)

TCCSH0 ~ TCCSH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A8 _H	ECKE	IRQZF	IRQZE	MSI2	MSI1	MSI0	ICLR	ICRE	00000000 _B
000158 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
000160 _H									

タイマ状態制御レジスタ (下位)

TCCSL0 ~ TCCSL2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000A9 _H	BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0	01000000 _B
000159 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
000161 _H									

A/D トリガ制御レジスタ

ADTRGC

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000AB _H	-	-	-	-	SEL2	SEL1	AD2E	AD1E	0XXX0000 _B
	-	-	-	-	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

R : リードオンリ

W : ライトオンリ

X : 不定

- : 未定義

■ 16 ビットアウトプットコンペアのレジスタ

アウトプットコンペアバッファレジスタ , アウトプットコンペアレジスタ (上位)

OCCPBH0 ~ OCCPBH5/OCCPH0 ~ OCCPH5

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000090 _H	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	00000000 _B
000092 _H	W	W	W	W	W	W	W	W	OCCPBH ライト OCCPH リード
000094 _H	R	R	R	R	R	R	R	R	
000096 _H									
000098 _H									
00009A _H									

アウトプットコンペアバッファレジスタ , アウトプットコンペアレジスタ (下位)

OCCPBL0 ~ OCCPBL5/OCCPL0 ~ OCCPL5

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000091 _H	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	00000000 _B
000093 _H	W	W	W	W	W	W	W	W	OCCPBL ライト OCCPL リード
000095 _H	R	R	R	R	R	R	R	R	
000097 _H									
000099 _H									
00009B _H									

コンペア制御レジスタ 1, 3, 5 (上位)

OCSH1, OCSH3, OCSH5

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00009C _H	-	BTS1	BTS0	CMOD	OTE1	OTE0	OTD1	OTD0	X1100000 _B
00009E _H	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000A0 _H									

コンペア制御レジスタ 0, 2, 4 (下位)

OCSL0, OCSL2, OCSL4

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
00009D _H	IOP1	IOP0	IOE1	IOE0	BUF1	BUF0	CST1	CST0	00001100 _B
00009F _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000A1 _H									

コンペアモード制御レジスタ

OCMOD

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00009D _H	-	-	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	XX000000 _B
00009F _H	-	-	R/W	R/W	R/W	R/W	R/W	R/W	
0000A2 _H									

R/W : リード / ライト可能

R : リードオンリ

W : ライトオンリ

X : 不定

- : 未定義

■ 16 ビットインプットキャプチャのレジスタ

インプットキャプチャデータレジスタ (上位)

IPCPH0 ~ IPCPH3

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000AC _H	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	XXXXXXXX _B
0000AE _H	R	R	R	R	R	R	R	R	
0000B0 _H									
0000B2 _H									

インプットキャプチャデータレジスタ (下位)

IPCPL0 ~ IPCPL3

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000AD _H	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	XXXXXXXX _B
0000AF _H	R	R	R	R	R	R	R	R	
0000B1 _H									
0000B3 _H									

インプットキャプチャ状態制御レジスタ (ch.2, ch.3) (上位)

ICSH23

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000B6 _H	-	-	-	-	-	-	IEI3	IEI2	XXXXXX00 _B
	-	-	-	-	-	-	R	R	

インプットキャプチャ状態制御レジスタ (ch.2, ch.3) (下位)

ICSL23

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000B7 _H	ICP3	ICP2	ICE3	ICE2	EG31	EG30	EG21	EG20	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PPG 出力制御 / インプットキャプチャ状態制御レジスタ (ch.0, ch.1) (上位)

PICSH01

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000B4 _H	PGEN5	PGEN4	PGEN3	PGEN2	PGEN1	PGEN0	IEI1	IEI0	00000000 _B
	W	W	W	W	W	W	R	R	

インプットキャプチャ状態制御レジスタ (ch.0, ch.1) (下位)

PICSL01

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000B5 _H	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

R : リードオンリ

W : ライトオンリ

X : 不定

- : 未定義

■ 波形ジェネレータのレジスタ

16 ビットデッドタイムレジスタ (上位)

TMRRH0 ~ TMRRH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000BC _H	TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	XXXXXXXX _B
0000BE _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000C0 _H									

16 ビットデッドタイムレジスタ (下位)

TMRRL0 ~ TMRRL2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000BD _H	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	XXXXXXXX _B
0000BF _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000C1 _H									

16 ビットデッドタイム制御レジスタ 0

DTCR0

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000C4 _H	DMOD0	GTEN1	GTEN0	TMIF0	TMIE0	TMD2	TMD1	TMD0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16 ビットデッドタイム制御レジスタ 1

DTCR1

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000C5 _H	DMOD1	GTEN3	GTEN2	TMIF1	TMIE1	TMD5	TMD4	TMD3	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

16 ビットデッドタイム制御レジスタ 2

DTCR2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000C6 _H	DMOD2	GTEN5	GTEN4	TMIF2	TMIE2	TMD8	TMD7	TMD6	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

波形制御レジスタ 1

SIGCR1

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000C9 _H	DTIE	DTIF	NRSL	DCK2	DCK1	DCK0	NWS1	NWS0	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

波形制御レジスタ 2

SIGCR2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000CB _H	-	-	-	-	-	-	-	DTTI	XXXXXXXX _{1B}
	-	-	-	-	-	-	-	R/W	

R/W : リード / ライト可能

X : 不定

- : 未定義

■ A/D 起動コンペアのレジスタ

コンペアレジスタ 2 (上位)

ADCOMP1,ADCOMP2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.1 :0000CE _H	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	00000000 _B
ch.2 :0000D0 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

コンペアレジスタ 2 (下位)

ADCOMP1,ADCOMP2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.1 :0000CF _H	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	00000000 _B
ch.2 :0000D1 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

制御レジスタ 1

ADCOMPC1

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000D3 _H	-	-	-	-	-	CE2	CE1	-	XXXXX00X _B
	-	-	-	-	-	R/W	R/W	-	

制御レジスタ 2

ADCOMPC2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000D2 _H	-	-	SEL21	SEL20	SEL11	SEL10	-	-	XX0000XX _B
	-	-	R/W	R/W	R/W	R/W	-	-	

R/W : リード / ライト可能

X : 不定

- : 未定義

■ フリーランタイムセクタレジスタ

FSR2

アドレス	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	初期値
000169 _H	ICU31	ICU30	ICU21	ICU20	ICU11	ICU10	ICU01	ICU00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

FSR1

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00016A _H	-	-	-	-	OCU51	OCU50	OCU41	OCU40	XXXX0000 _B
	-	-	-	-	R/W	R/W	R/W	R/W	

FSR0

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00016B _H	OCU31	OCU30	OCU21	OCU20	OCU11	OCU10	OCU01	OCU00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

X : 不定

- : 未定義

12.4.1 コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2, CPCLRBL0 ~ CPCLRBL2) / コンペアクリアレジスタ (CPCLRH0 ~ CPCLRH2, CPCLRL0 ~ CPCLRL2)

コンペアクリアバッファレジスタ (CPCLRBH, CPCLRBL) は, コンペアクリアレジスタ (CPCLRH, CPCLRL) に存在する 16 ビットバッファレジスタです。CPCLRBH, CPCLRBL レジスタと CPCLRH, CPCLRL レジスタは, 両方とも同じアドレスに存在します。

■ コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2, CPCLRBL0 ~ CPCLRBL2)

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (上位)

CPCLRBH0 ~ CPCLRBH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A4 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	11111111 _B
000154 _H	W	W	W	W	W	W	W	W	
00015C _H									

コンペアクリアバッファレジスタ, コンペアクリアレジスタ (下位)

CPCLRBL0 ~ CPCLRBL2

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
-- CL07	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	11111111 _B
-- W	W	W	W	W	W	W	W	W	

W : ライトオンリ

コンペアクリアバッファレジスタは, コンペアクリアレジスタ (CPCLRH, CPCLRL) と同じアドレスに存在するバッファレジスタです。バッファ機能が無効になるか (タイマ状態制御レジスタ下位 (TCCSL) の BFE : bit7=0), またはフリーランタイムが停止すると, コンペアクリアバッファレジスタの値が直ちにコンペアクリアレジスタへ転送されます。バッファ機能が有効になると, 16 ビットフリーランタイムのカウント値 "0" が検出されたときに値がコンペアクリアレジスタへ転送されます。

このレジスタへアクセスする場合は, ハーフワードもしくはワードアクセス命令をご使用ください。

■ コンペアクリアレジスタ (CPCLRHO ~ CPCLRHH2, CPCLRL0 ~ CPCLRL2)

コンペアクリアバッファレジスタ , コンペアクリアレジスタ (上位)

CPCLRHO ~ CPCLRHH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A4 _H	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	11111111 _B
000154 _H	R	R	R	R	R	R	R	R	
00015C _H									

コンペアクリアバッファレジスタ , コンペアクリアレジスタ (下位)

CPCLRL0 ~ CPCLRL2

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
--	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	11111111 _B
--	R	R	R	R	R	R	R	R	

R : リードオンリ

コンペアクリアレジスタは , 16 ビットフリーランタイマのカウント値と比較するために使用します。アップカウントモード時は , このレジスタが 16 ビットフリーランタイマのカウント値と一致すると , 16 ビットフリーランタイマは , "0000_H" にリセットされます。アップダウンカウントモード時は , このレジスタが 16 ビットフリーランタイマのカウント値と一致すると , 16 ビットフリーランタイマは , アップカウントからダウンカウントに変わるか , または "0" 検出時にダウンカウントからアップカウントに変わります。

このレジスタへアクセスする場合は , ハーフワードもしくはワードアクセス命令をご使用ください。

12.4.2 タイマデータレジスタ (TCDTH0 ~ TCDTH2, TCDTL0 ~ TCDTL2)

タイマデータレジスタは, (TCDTH, TCDTL) は, 16 ビットフリーランタイマのカウント値を読み出すために使用します。

■ タイマデータレジスタ (TCDTH0 ~ TCDTH2, TCDTL0 ~ TCDTL2)

タイマデータレジスタ (上位)

TCDTH0 ~ TCDTH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000A6 _H	T15	T14	T13	T12	T11	T10	T09	T08	00000000 _B
000156 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
00015E _H									

タイマデータレジスタ (下位)

TCDTL0 ~ TCDTL2

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	T07	T06	T05	T04	T03	T02	T01	T00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

タイマデータレジスタは, 16 ビットフリーランタイマのカウント値を読み出すために使用します。カウント値は, リセットが発生すると直ちに "0000_H" にクリアされます。タイマ値は, このレジスタへ値を書き込むことで設定することができます。ただし, 値の書き込みはタイマの停止中 (タイマ状態制御レジスタ下位 (TCCSL) の STOP : bit6=1) でなければなりません。タイマデータレジスタへアクセスする場合は, ハーフワードもしくはワードアクセス命令をご使用ください。

16 ビットフリーランタイマは, 以下の要因が発生すると直ちに初期化されます。

- リセット
- タイマ状態制御レジスタ (TCCSL) のクリアビット (SCLR : bit4) =1
- アップカウントモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE:bit5=0) 時におけるコンペアクリアレジスタとタイマカウント値の一致

12.4.3 タイマ状態制御レジスタ (TCCSH0 ~ TCCSH2, TCCSL0 ~ TCCSL2)

タイマ状態制御レジスタ (TCCSH, TCCSL) は, 16 ビットフリーランタイムの動作を制御するために使用する 16 ビットレジスタです。

■ タイマ状態制御レジスタ, 上位 (TCCSH0 ~ TCCSH2)

TCCSH0 ~ TCCSH2

アドレス

0000A8_H

000158_H

000160_H

bit15

bit14

bit13

bit12

bit11

bit10

bit9

bit8

ECKE

IRQZF

IRQZE

MSI2

MSI1

MSI0

ICLR

ICRE

R/W

R/W

R/W

R/W

R/W

R/W

R/W

R/W

初期値

00000000_B

ICRE

コンペアクリア割込み要求許可ビット

0

割込み要求を禁止する

1

割込み要求を許可する

ICLR

コンペアクリア割込みフラグビット

読出し時

書込み時

0

コンペアクリア一致なし

このビットをクリアする

1

コンペアクリア一致あり

このビットに影響を与えない

MSI2

MSI1

MSI0

割込みマスク選択ビット

0

0

0

1回目の一致が発生したときに割込み生成

0

0

1

2回目の一致が発生したときに割込み生成

0

1

0

3回目の一致が発生したときに割込み生成

0

1

1

4回目の一致が発生したときに割込み生成

1

0

0

5回目の一致が発生したときに割込み生成

1

0

1

6回目の一致が発生したときに割込み生成

1

1

0

7回目の一致が発生したときに割込み生成

1

1

1

8回目の一致が発生したときに割込み生成

IRQZE

0 検出割込み要求許可ビット

0

割込み要求を禁止にする

1

割込み要求を許可する

IRQZF

0 検出割込みフラグビット

読出し時

書込み時

0

"0" が検出されない

このビットをクリアする

1

"0" が検出される

このビットに影響を与えない

ECKE

クロック選択ビット

0

内部クロック

1

外部クロック

R/W : リード / ライト可能

初期値

表 12.4-1 タイマ状態制御レジスタ，上位 (TCCSH) (1 / 2)

ビット名	機能
bit15 ECKE : クロック選択 ビット	<ul style="list-style-type: none"> このビットは，内部クロックまたは外部クロックを 16 ビットフリーランタイムのカウントクロックとして選択するために使用します。 このビットに "0" を設定した場合： 内部クロックが選択されます。カウントクロック周波数を選択するためには，TCCSL レジスタのクロック周波数選択ビット (CLK3 ~ CLK0 : bit3 ~ bit0) も選択しなければなりません。 このビットに "1" を設定した場合： 外部クロックが選択されます。外部クロックは，"CKI" 端子から入力されます。したがって，ポート方向レジスタ (DDR1) の bit7 へ "0" を書き込んで外部クロック入力を有効にしなければなりません。 <p>(注意事項) カウントクロックは，このビットが設定されると直ちに変更されます。したがって，このビットの変更は，アウトプットコンペアとインプットキャプチャが停止している間でなければなりません。</p>
bit14 IRQZF : 0 検出割込みフ ラグビット	<ul style="list-style-type: none"> 16 ビットフリーランタイムのカウント値が "0000_H" のとき，このビットには "1" がセットされます。 このビットに "0" を設定した場合：このビットはクリアされます。 このビットに "1" を設定した場合：このビットは影響を受けません。 リードモディファイライト (RMW) 時は，必ず "1" が読み出されます。 <p>(注意事項) ソフトウェアクリア (タイマ状態制御レジスタ下位 (TCCSL) の SCLR : bit4 への "1" 書込み) では，このビットは設定されません。アップダウンカウントモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE : bit5=1) 時は，割込みマスク選択ビット (タイマ状態制御レジスタ上位 (TCCSH) の MSI2 ~ MSI0 : bit12 ~ bit10 が "000_B" 以外) で設定した割込みが発生したときにこのビットに "1" が設定されます。割込みが発生しないときは，このビットに "1" は設定されません。アップカウントモード (MODE : bit5=0) 時は，MSI2 ~ MSI0 : bit12 ~ bit10 の値とは無関係に，このビットは 0 検出が発生するたびに設定されます。</p>
bit13 IRQZE : 0 検出割込み要 求許可ビット	<p>このビットと割込みフラグビット (IRQZF : bit14) に "1" が設定されると，CPU に対する割込み要求が生成されます。</p>
bit12 ~ bit10 MSI2 ~ MSI0 : 割込みマスク選 択ビット	<ul style="list-style-type: none"> これらのビットは，アップカウントモード (MODE=0) 時は，コンペアクリア割込みのマスク回数を設定するために使用します。アップダウンカウントモード (MODE=1) 時は，0 検出割込みのマスク回数を設定するために使用します。 このビットに "0" を設定した場合：割込み要因はマスクされません。 <p>(注意事項) 割込み要因を 2 回マスクし，3 回目の割込みを処理する際には，これらのビットに "010_B" を設定しなければなりません。 読出し値はマスクカウンタの値です。</p>

表 12.4-1 タイマ状態制御レジスタ, 上位 (TCCSH) (2 / 2)

ビット名		機能
bit9	ICLR : コンペアクリア 割込みフラグ ビット	<ul style="list-style-type: none"> • コンペアクリア値と 16 ビットフリーランタイム値が一致すると, このビットには "1" が設定されます。 • このビットに "0" を設定した場合: このビットはクリアされます。 • このビットに "1" を設定した場合: このビットは影響を受けません。 • リードモディファイライト (RMW) 時は, 必ず "1" が読み出されます。 <p>(注意事項) アップカウントモード (タイマ状態制御レジスタ下位 (TCCSL) の MODE : bit5=0) 時は, 割込みマスク選択ビットで設定した割込みが発生したときにこのビットに "1" が設定されます。 割込みが発生しないときは, このビットに "1" は設定されません。 アップダウンカウントモード (MODE=1) 時は, MSI2 ~ MSI0 ビットの値とは無関係に, このビットはコンペアクリアが発生するたびに設定されます。</p>
bit8	ICRE : コンペアクリア 割込み要求許可 ビット	このビットとコンペアクリア割込みフラグビット (ICLR : bit9) に "1" が設定されると, CPU に対する割込み要求が生成されます。

■ タイマ制御レジスタ, 下位 (TCCSL0 ~ TCCSL2)

TCCSL0 ~ TCCSL2		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値 01000000 _B
アドレス 0000A9 _H 000159 _H 000161 _H		BFE	STOP	MODE	SCLR	CLK3	CLK2	CLK1	CLK0	
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

				クロック周波数選択ビット					
CLK3	CLK2	CLK1	CLK0	カウント クロック	$\phi/=32\text{MHz}$	$\phi/=16\text{MHz}$	$\phi/=8\text{MHz}$	$\phi/=4\text{MHz}$	$\phi/=1\text{MHz}$
0	0	0	0	ϕ	31.25ns	62.5ns	125ns	0.25 μs	1 μs
0	0	0	1	$\phi/2$	62.5ns	125ns	0.25 μs	0.5 μs	2 μs
0	0	1	0	$\phi/4$	125ns	0.25 μs	0.5 μs	1 μs	4 μs
0	0	1	1	$\phi/8$	0.25 μs	0.5 μs	1 μs	2 μs	8 μs
0	1	0	0	$\phi/16$	0.5 μs	1 μs	2 μs	4 μs	16 μs
0	1	0	1	$\phi/32$	1 μs	2 μs	4 μs	8 μs	32 μs
0	1	1	0	$\phi/64$	2 μs	4 μs	8 μs	16 μs	64 μs
0	1	1	1	$\phi/128$	4 μs	8 μs	16 μs	32 μs	128 μs
1	0	0	0	$\phi/256$	8 μs	16 μs	32 μs	64 μs	256 μs
その他 設定禁止				—	—	—	—	—	—

ϕ : マシンサイクル

SCLR	タイマクリアビット	
	読み出し時	書き込み時
	0	カウンタを初期化しない
1	常に "0" を読み出す	
		カウンタを "0000 _H " に初期化

MODE	タイマカウントモードビット
0	アップカウントモード
1	アップダウンカウントモード

STOP	タイマ許可ビット
0	カウントを許可する (カウント開始する)
1	カウントを禁止する (カウント停止する)

BFE	コンペアクリアバッファ許可ビット
0	コンペアクリアバッファを無効にする
1	コンペアクリアバッファを有効にする

R/W : リード / ライト可能
 : 初期値

表 12.4-2 タイマ状態制御レジスタ, 下位 (TCCSL) (1 / 2)

ビット名		機能
bit7	BFE : コンペアクリア バッファ許可 ビット	<ul style="list-style-type: none"> このビットは, コンペアクリアバッファを有効にするために使用します。 このビットに "0" を設定した場合 : コンペアクリアバッファは無効になります。したがって, コンペアクリアレジスタ (CPCLR_H, CPCLR_L) に直接書き込むことが可能です。 このビットに "1" を設定した場合 : コンペアクリアバッファは有効になります。コンペアクリアバッファに書き込まれ, 保持されていたデータは, 16 ビットフリーランタイムからのカウント値 "0" が検出されると, コンペアクリアレジスタへ転送されます。
bit6	STOP : タイマ許可 ビット	<ul style="list-style-type: none"> このビットは, 16 ビットフリーランタイムのカウントを停止 / 開始するために使用します。 このビットに "0" を設定した場合 : 16 ビットフリーランタイムのカウントは開始します。 このビットに "1" を設定した場合 : 16 ビットフリーランタイムのカウントは停止します。 <p>(注意事項) 16 ビットフリーランタイムが停止すると, アウトプットコンペアの動作も停止します。</p>
bit5	MODE : タイマカウント モードビット	<ul style="list-style-type: none"> このビットは, 16 ビットフリーランタイムのカウントモードを選択するために使用します。 このビットに "0" を設定した場合 : アップカウントモードが選択されます。タイマは, カウント値がコンペアクリアレジスタと一致して "0000_H" にリセットされるまでカウントアップし, その後, 再びカウントアップします。 このビットに "1" を設定した場合 : アップダウンカウントモードが選択されます。タイマは, カウント値がコンペアクリアレジスタと一致するまでカウントアップし, その後ダウンカウントに変わります。その後, カウント値が "0000_H" に達すると再びアップカウントに変わります。 このビットは, タイマが動作中であっても停止されていても書き込みが可能です。タイマが動作中の場合は, このビットに書き込まれた値はバッファに入れられ, その後, タイマ値が "0000_H" になるとバッファの値によりカウントモードが変わります。

表 12.4-2 タイマ状態制御レジスタ, 下位 (TCCSL) (2 / 2)

ビット名		機能
bit4	SCLR : タイマクリア ビット	<ul style="list-style-type: none"> • このビットは, 16 ビットフリーランタイムを "0000_H" に初期化するために使用します。 • このビットに "1" を設定した場合 : 16 ビットフリーランタイムは, その次のカウントクロックで "0000_H" に初期化されます。 • 読出し値は, 必ず "0" です。 <p>(注意事項) このビットに "1" を書き込んでも, 0 検出割込みは生成されません。 "1" を設定した後, 次のカウントクロックが来る前に "0" を書き込むとタイマクリアは行われません。</p>
bit3 ~ bit0	CLK3 ~ CLK0 : クロック周波数 選択ビット	<ul style="list-style-type: none"> • このビットは, 16ビットフリーランタイムのカウントクロック周波数を選択するために使用します。 • カウントクロックは, これらのビットが設定されると直ちに変更されます。したがって, これらのビットの変更は, アウトプットコンペアとインプットキャプチャが停止している間でなければなりません。

12.4.4 A/D トリガ制御レジスタ (ADTRGC)

フリーランタイムのコンペア一致時，もしくは 0 検出時に A/D トリガ信号出力を制御します。

■ A/D トリガ制御レジスタ (ADTRGC)

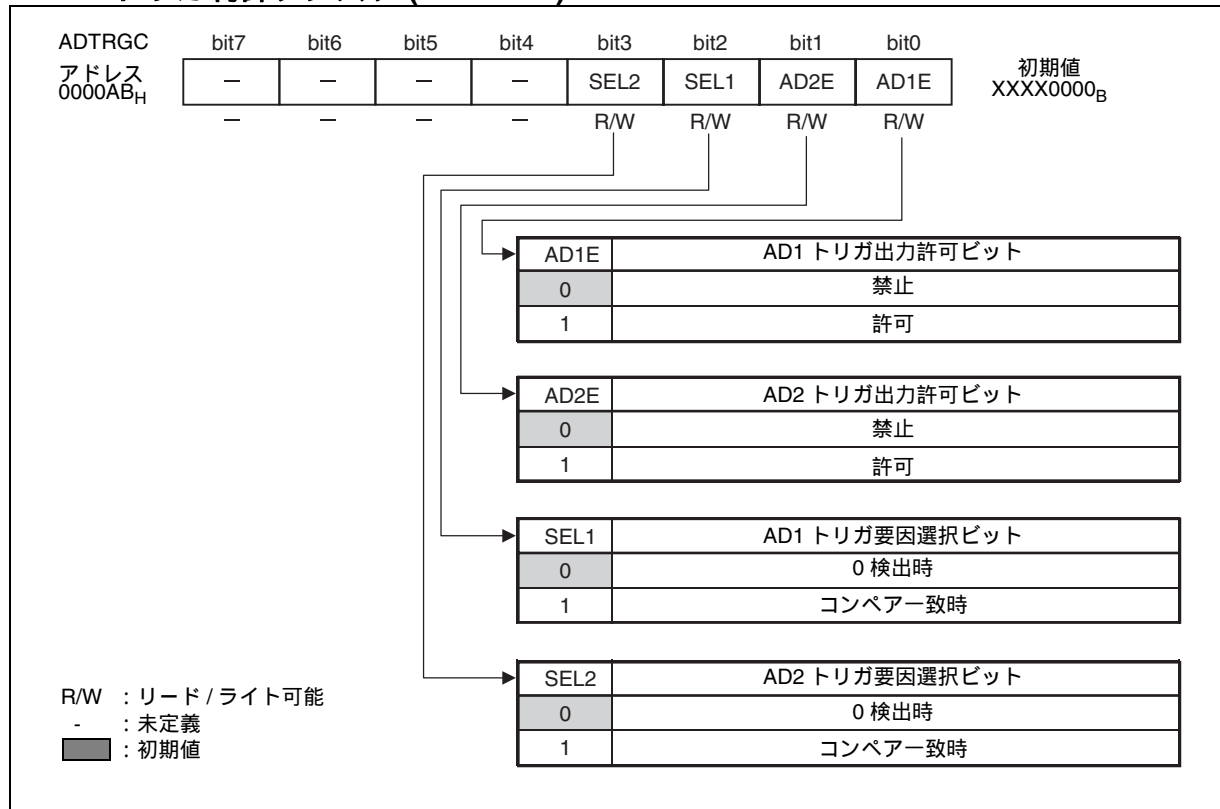


表 12.4-3 AD トリガ制御レジスタ (ADTRGC)

ビット名		機能
bit7 ~ bit4	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは動作に影響しません。
bit3	SEL2 : AD2 トリガ要因選択ビット	AD2 のトリガをフリーランタイムの 0 検出時に出力するか , コンペアー一致時に出力するかを選択ビットです。
bit2	SEL1 : AD1 トリガ要因選択ビット	AD1 のトリガをフリーランタイムの 0 検出時に出力するか , コンペアー一致時に出力するかを選択ビットです。
bit1	AD2E : AD2 トリガ許可ビット	<ul style="list-style-type: none"> "0" のとき , AD2 トリガ信号は出力されません。 "1" のとき , 出力許可となります。
bit0	AD1E : AD1 トリガ許可ビット	<ul style="list-style-type: none"> "0" のとき , AD1 トリガ信号は出力されません。 "1" のとき , 出力許可となります。

12.4.5 アウトプットコンペアバッファレジスタ (OCCPBH0 ~ OCCPBH5, OCCPBL0 ~ OCCPBL5) / アウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5, OCCPL0 ~ OCCPL5)

アウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL) は、アウトプットコンペアレジスタ (OCCPH, OCCPL) 用の 16 ビットバッファレジスタです。OCCPBH, OCCPBL レジスタと OCCPH, OCCPL レジスタは、両方とも同じアドレスに存在しています。

■ アウトプットコンペアバッファレジスタ (OCCPBH0 ~ OCCPBH5, OCCPBL0 ~ OCCPBL5)

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (上位)

OCCPBH0 ~ OCCPBH5

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000090 _H	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	00000000 _B
000092 _H	W	W	W	W	W	W	W	W	
000094 _H									
000096 _H									
000098 _H									
00009A _H									

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (下位)

OCCPBL0 ~ OCCPBL5

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00		00000000 _B
W	W	W	W	W	W	W	W	W	

W : ライトオンリ

アウトプットコンペアバッファレジスタは、アウトプットコンペアレジスタ (OCCPH, OCCPL) 用のバッファレジスタです。バッファ機能が無効になるか (コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の BUF1, BUF0: bit3, bit2=11_B)、またはフリーランタイムが停止すると、アウトプットコンペアバッファレジスタの値は、直ちにアウトプットコンペアレジスタへ転送されます。バッファ機能が有効になると (コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の BUF1, BUF0: bit3, bit2=00_B)、値はコンペア制御レジスタ上位 (OCSH1, OCSH3, OCSL5) の転送選択ビット (BTS1, BTS0: bit14, bit13) に従ってコンペアクリアー致時、または 0 検出時に転送されます。

このレジスタへアクセスする場合、ハーフワードもしくはワードアクセス命令をご使用ください。

以上の説明中のフリーランタイムはアウトプットコンペアが選択しているフリーランタイムの動作状態についてです。

■ アウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5, OCCPL0 ~ OCCPL5)

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (上位)

OCCPH0 ~ OCCPH5

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
000090 _H	OP15	OP14	OP13	OP12	OP11	OP10	OP09	OP08	00000000 _B
000092 _H	R	R	R	R	R	R	R	R	
000094 _H									
000096 _H									
000098 _H									
00009A _H									

アウトプットコンペアバッファレジスタ, アウトプットコンペアレジスタ (下位)

OCCPL0 ~ OCCPL5

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
000091 _H	OP07	OP06	OP05	OP04	OP03	OP02	OP01	OP00	00000000 _B
000093 _H	R	R	R	R	R	R	R	R	
000095 _H									
000097 _H									
000099 _H									
00009B _H									

R : リードオンリ

アウトプットコンペアレジスタは、16 ビットフリーランタイムのカウント値と比較するために使用する 16 ビットレジスタです。タイマの動作を有効にする前にアウトプットコンペアバッファレジスタ (OCCPBH, OCCPBL) に値を設定してください。

アウトプットコンペアレジスタの値が 16 ビットフリーランタイムのカウント値と一致すると、コンペア信号が生成され、アウトプットコンペア割込みフラグビット (コンペア制御レジスタ下位 OCSL0, OCSL2, OCSL4 の IOP1, IOP0 : bit7, bit6) が設定されます。出力レベルが設定されると (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の ODT1, ODT0 : bit9, bit8), アウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5, OCCPL0 ~ OCCPL5) に対応する出力レベル波形ジェネレータ RTO0 ~ RTO5 を反転させることができます。

本レジスタ値と 16 ビットフリーランタイムのアップダウンモード時のピーク値と一致した場合は、コンペア信号は生成されません。

● アップダウンモード

• CMOD=0 時

本レジスタ値に "FFFF_H" を設定した場合は、RT 出力は 16 ビットフリーランタイムの値や反転モードにかかわらず "0" 出力となります。"0000_H" を設定した場合は "1" 出力となります。

• CMOD=1 時

本レジスタ値に "FFFF_H" を設定した場合は、RT 出力は 16 ビットフリーランタイムの値や反転モードにかかわらず "1" 出力となります。"0000_H" を設定した場合は "0" 出力となります。

このレジスタへアクセスする場合、ハーフワードもしくはワードアクセス命令をご使用ください。

以上の説明中のフリーランタイムはアウトプットコンペアが選択しているフリーランタイムの動作状態についてです。

12.4.6 コンペア制御レジスタ (OCSH0 ~ OCSH5, OCSL0 ~ OCSL5)

コンペア制御レジスタは, RT0 ~ RT5 の出力レベル, 出力許可, 出力レベル反転モード, コンペア動作許可, コンペアー一致割込み許可, およびコンペアー一致割込みフラグを制御するために使用します。

■ コンペア制御レジスタ, 上位 (OCSH1, OCSH3, OCSH5)

OCSH1, OCSH3, OCSH5		bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値 X1100000 _B
アドレス 00009C _H 00009E _H 0000A0 _H		—	BTS1	BTS0	CMOD	OTE1	OTE0	OTD1	OTD0	
		—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

OTD0	出力レベルビット	
	読出し時	書込み時
	0	RT0, RT2, RT4 の現出力値
1	RT0, RT2, RT4 が "1" を出力	

OTD1	出力レベルビット	
	読出し時	書込み時
	0	RT1, RT3, RT5 の現出力値
1	RT1, RT3, RT5 が "1" を出力	

OTE0	出力許可ビット	
	汎用入出力ポート	
	0	波形ジェネレータ出力端子 (RTO0, RTO2, RTO4)
1		

OTE1	出力許可ビット	
	汎用入出力ポート	
	0	波形ジェネレータ出力端子 (RTO1, RTO3, RTO5)
1		

CMOD	出力レベル反転モードビット	
	MOD1x=0	MOD1x=1
	0	アップカウント時の一致時は "1" にリセット ダウンカウント時の一致時は "0" にセット
1	アップカウント時の一致時は "0" にリセット ダウンカウント時の一致時は "1" にセット	

BTS0	バッファ転送選択ビット	
	0 検出が発生すると転送が起動 (ch.0, 2, 4)	
	コンペアクリア一致が発生すると転送が起動 (ch.0, 2, 4)	

BTS1	バッファ転送選択ビット	
	0 検出が発生すると転送が起動 (ch.1, 3, 5)	
	コンペアクリア一致が発生すると転送が起動 (ch.1, 3, 5)	

R/W : リード / ライト可能
 — : 未定義
 ■ : 初期値

表 12.4-4 コンペア制御レジスタ , 上位 (OCSH1, OCSH3, OCSH5) (1 / 3)

ビット名		機能
bit15	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは動作に影響しません。
bit14	BTS1 : バッファ転送 選択ビット	<ul style="list-style-type: none"> このビットは , アウトプットコンペアバッファレジスタ (OCCPBH1,OCCPBH3,OCCPBH5, OCCPBL1,OCCPBL3,OCCPBL5) からアウトプットコンペアレジスタ (OCCPH1,OCCPH3,OCCPH5, OCCPL1,OCCPL3,OCCPL5) へのデータ転送時期を選択するために使用します。 このビットに "0" を設定した場合 : データ転送は , 16 ビットフリーランタイムのカウント値 "0" が検出されると起動します。 このビットに "1" を設定した場合 : データ転送は , 16 ビットフリーランタイムでコンペアクリア一致が発生すると起動します。
bit13	BTS0 : バッファ転送 選択ビット	<ul style="list-style-type: none"> このビットは , アウトプットコンペアバッファレジスタ (OCCPBH0,OCCPBH2,OCCPBH4, OCCPBL0,OCCPBL2,OCCPBL4) からアウトプットコンペアレジスタ (OCCPH0,OCCPH2,OCCPH4, OCCPL0,OCCPL2,OCCPL4) へのデータ転送時期を選択するために使用します。 このビットに "0" を設定した場合 : データ転送は , 16 ビットフリーランタイムのカウント値 "0" が検出されると起動します。 このビットに "1" を設定した場合 : データ転送は , 16 ビットフリーランタイムでコンペアクリア一致が発生すると起動します。

表 12.4-4 コンペア制御レジスタ, 上位 (OCSH1, OCSH3, OCSH5) (2 / 3)

ビット名	機能
bit12	<p>CMOD : 出力レベル反転モードビット</p> <ul style="list-style-type: none"> このビットは, 端子出力が有効の間 (OTE1=1 または OTE0=1) に一致が発生した場合に端子出力レベル反転モードを直ちに切り換えるために使用します。 このビットに "0" を設定した場合 : コンペアモード制御レジスタ (OCMOD) : MOD1x=0 のとき <ul style="list-style-type: none"> RT0, RT2, RT4 : レベルは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 が一致すると直ちに反転します。 RT1, RT3, RT5 : レベルは, 16 ビットフリーランタイムとコンペアレジスタ 1, 3, 5 が一致すると直ちに反転します。 コンペアモード制御レジスタ (OCMOD) : MOD1x=1 のとき <ul style="list-style-type: none"> アップカウント時に一致したときは, "1" にセット ダウンカウント時に一致したときは, "0" にリセット このビットに "1" を設定した場合 : コンペアモード制御レジスタ (OCMOD) : MOD1x=0 のとき <ul style="list-style-type: none"> RT0, RT2, RT4 : レベルは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 が一致すると直ちに反転します。 RT1, RT3, RT5 : レベルは, 16 ビットフリーランタイムとコンペアレジスタ (0 または 1) (2 または 3) (4 または 5) が一致すると直ちに反転します。 コンペアレジスタ 0, 2, 4 と 1, 3, 5 が同じ値の場合は, ただ 1 つのコンペアレジスタが使用される場合と同じ動作になります。 コンペアモード制御レジスタ (OCMOD) : MOD1x=1 のとき <ul style="list-style-type: none"> アップカウント時に一致したときは, "0" にリセット ダウンカウント時に一致したときは, "1" にセット
bit11	<p>OTE1 : 出力許可ビット</p> <ul style="list-style-type: none"> このビットは, ポートへの波形ジェネレータ出力 (RTO1, RTO3, RTO5) を許可するために使用します。 このビットの初期値は "0" です。 <p>(注意事項) 波形ジェネレータが無効 (16 ビットデッドタイム制御レジスタ下位 (DTCR0 ~ DTCR2) の TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6 : bit2 ~ bit0=000_B) の場合は, RTO1, RTO3, RTO5 はアウトプットコンペアと同じ値を出力します。</p>
bit10	<p>OTE0 : 出力許可ビット</p> <ul style="list-style-type: none"> このビットは, ポートへの波形ジェネレータ出力 (RTO0, RTO2, RTO4) を許可するために使用します。 このビットの初期値は "0" です。 <p>(注意事項) 波形ジェネレータが無効 (16 ビットデッドタイム制御レジスタ下位 (DTCR0 ~ DTCR2) の TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6 : bit2 ~ bit0=000_B) の場合は, RTO0, RTO2, RTO4 はアウトプットコンペアと同じ値を出力します。</p>

表 12.4-4 コンペア制御レジスタ, 上位 (OCSH1, OCSH3, OCSH5) (3 / 3)

ビット名		機能
bit9	OTD1 : 出力レベル ビット	<ul style="list-style-type: none"> • このビットは , アウトプットコンペア 1, 3, 5 (RT1, RT3, RT5) の端子出力レベルを変更するために使用します。 • コンペア端子出力の初期値は "0" です。 • 値を書き込む場合は , 必ず前もってコンペア動作を停止させてください。このビットの読出し値は , RT1, RT3, RT5 におけるアウトプットコンペア値を示します。
bit8	OTD0 : 出力レベル ビット	<ul style="list-style-type: none"> • このビットは , アウトプットコンペア 0, 2, 4 (RT0, RT2, RT4) の端子出力レベルを変更するために使用します。 • コンペア端子出力の初期値は "0" です。 • 値を書き込む場合は , 必ず前もってコンペア動作を停止させてください。このビットの読出し値は , RT0, RT2, RT4 におけるアウトプットコンペア値を示します。

■ コンペア制御レジスタ , 下位 (OCSL0, OCSL2, OCSL4)

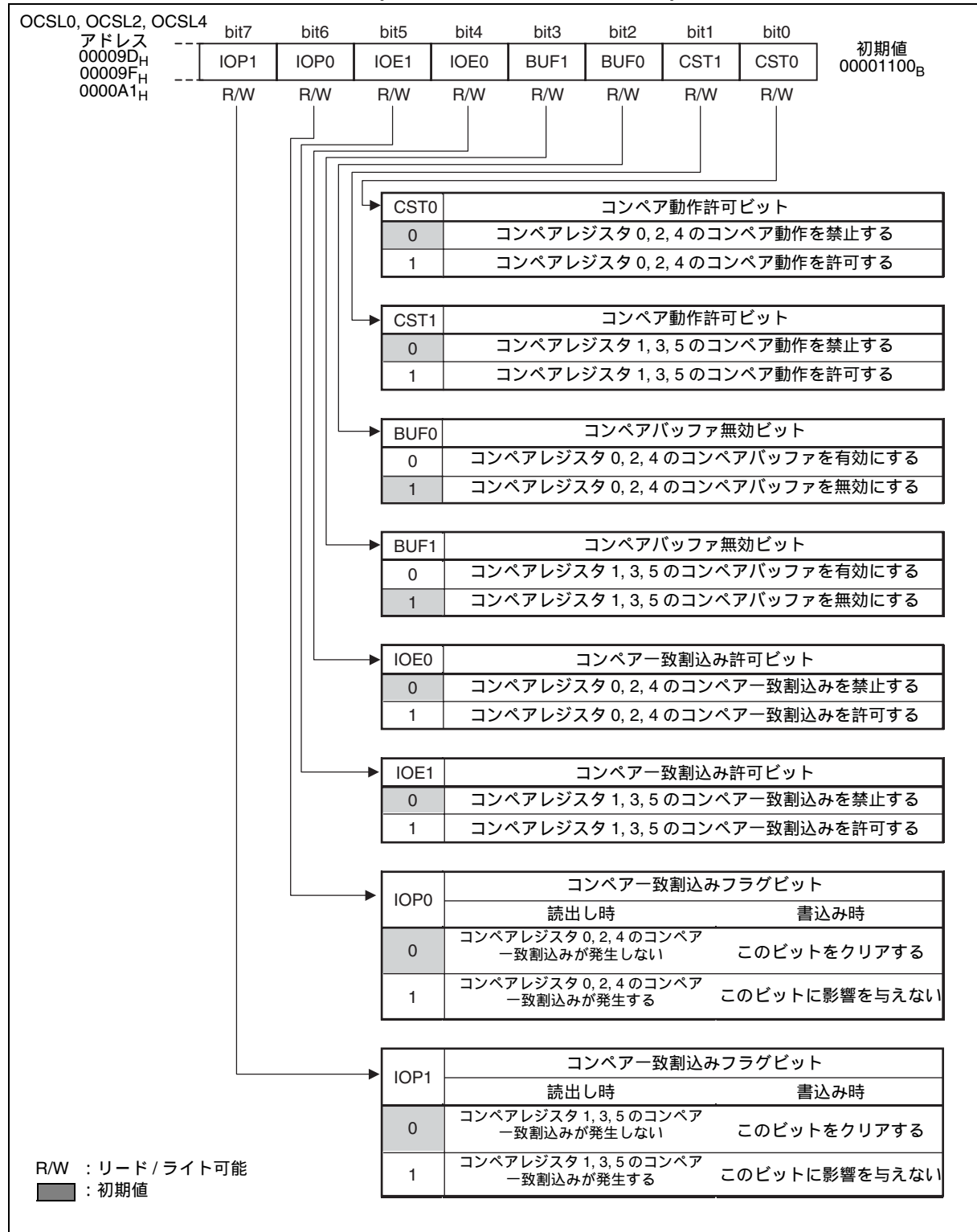


表 12.4-5 コンペア制御レジスタ , 下位 (OCSL0, OCSL2, OCSL4) (1 / 2)

ビット名		機能
bit7	IOP1 : コンペア一致 割込みフラグ ビット	<ul style="list-style-type: none"> このビットは , コンペアレジスタ 1, 3, 5 が 16 ビットフリーランタイムの値と一致したことを示す割込みフラグです。 このビットは , コンペアレジスタ値が 16 ビットフリーランタイム値に一致した場合に "1" が設定されます。 コンペア一致割込み許可ビット (IOE1 : bit5) が "許可" になっている間にこのビットが設定されると , アウトプットコンペア割込みが発生します。 このビットに "0" を設定した場合 : このビットはクリアされます。 このビットに "1" を設定した場合 : このビットは影響を受けません。 リードモディファイライト (RMW) 時は , 必ず "1" が読み出されます。
bit6	IOP0 : コンペア一致 割込みフラグ ビット	<ul style="list-style-type: none"> このビットは , コンペアレジスタ 0, 2, 4 が 16 ビットフリーランタイムの値と一致したことを示す割込みフラグです。 このビットは , コンペアレジスタ値が 16 ビットフリーランタイム値に一致した場合に "1" が設定されます。 コンペア一致割込み許可ビット (IOE0 : bit4) が "許可" になっている間にこのビットが設定されると , アウトプットコンペア割込みが発生します。 このビットに "0" を設定した場合 : このビットはクリアされます。 このビットに "1" を設定した場合 : このビットは影響を受けません。 リードモディファイライト (RMW) 時は , 必ず "1" が読み出されます。
bit5	IOE1 : コンペア一致 割込み許可 ビット	<ul style="list-style-type: none"> このビットは , コンペアレジスタ 1, 3, 5 のアウトプットコンペア割込みを "許可" にするために使用します。 このビットに "0" を設定した場合 : コンペア一致割込みを禁止します。 このビットに "1" が書き込まれている間にコンペア一致割込みフラグビット (IOP1 : bit7) が設定されると , アウトプットコンペア割込みが発生します。
bit4	IOE0 : コンペア一致 割込み許可 ビット	<ul style="list-style-type: none"> このビットは , コンペアレジスタ 0, 2, 4 のアウトプットコンペア割込みを "許可" にするために使用します。 このビットに "0" を設定した場合 : コンペア一致割込みを禁止します。 このビットに "1" が書き込まれている間にコンペア一致割込みフラグビット (IOP0 : bit6) が設定されると , アウトプットコンペア割込みが発生します。
bit3	BUF1 : コンペアバッ ファ無効ビット	<ul style="list-style-type: none"> このビットは , アウトプットコンペアレジスタ 1, 3, 5 のバッファ機能を無効にするために使用します。 このビットに "0" を設定した場合 : バッファ機能が有効になります。 このビットに "1" を設定した場合 : バッファ機能が無効になります。

表 12.4-5 コンペア制御レジスタ, 下位 (OCSL0, OCSL2, OCSL4) (2 / 2)

ビット名		機能
bit2	BUF0 : コンペアバッファ無効ビット	<ul style="list-style-type: none"> このビットは, アウトプットコンペアレジスタ 0, 2, 4 のバッファ機能を無効にするために使用します。 このビットに "0" を設定した場合: バッファ機能が有効になります。 このビットに "1" を設定した場合: バッファ機能が無効になります。
bit1	CST1 : コンペア動作許可ビット	<ul style="list-style-type: none"> このビットは, 16 ビットフリーランタイムとコンペアレジスタ 1, 3, 5 の間のコンペア動作を許可するために使用します。 コンペア動作を許可する場合は, 必ず前もってコンペアレジスタ 1, 3, 5 とタイマデータレジスタ (TCDTH, TCDTL) に値を書き込んでください。 このビットに "0" を設定した場合: コンペアレジスタのコンペア動作を禁止します。 このビットに "1" を設定した場合: コンペアレジスタのコンペア動作を許可します。 <p>(注意事項) アウトプットコンペアは, 16 ビットフリーランタイムクロックと同期化されるので, 16 ビットフリーランタイムを停止するとコンペア動作も停止します。</p>
bit0	CST0 : コンペア動作許可ビット	<ul style="list-style-type: none"> このビットは, 16 ビットフリーランタイムとコンペアレジスタ 0, 2, 4 の間のコンペア動作を許可するために使用します。 コンペア動作を許可する場合は, 必ず前もってコンペアレジスタ 0, 2, 4 とタイマデータレジスタ (TCDTH, TCDTL) に値を書き込んでください。 このビットに "0" を設定した場合: コンペアレジスタのコンペア動作を禁止します。 このビットに "1" を設定した場合: コンペアレジスタのコンペア動作を許可します。 <p>(注意事項) アウトプットコンペアは, 16 ビットフリーランタイムクロックと同期化されるので, 16 ビットフリーランタイムを停止すると 0 検出, コンペア動作も停止します。</p>

12.4.7 コンペアモード制御レジスタ (OCMOD)

コンペアモード制御レジスタは、コンペアー一致時の出力レベルを反転モードもしくはセット/リセットを行うかを制御します。

■ コンペアモード制御レジスタ (OCMOD)

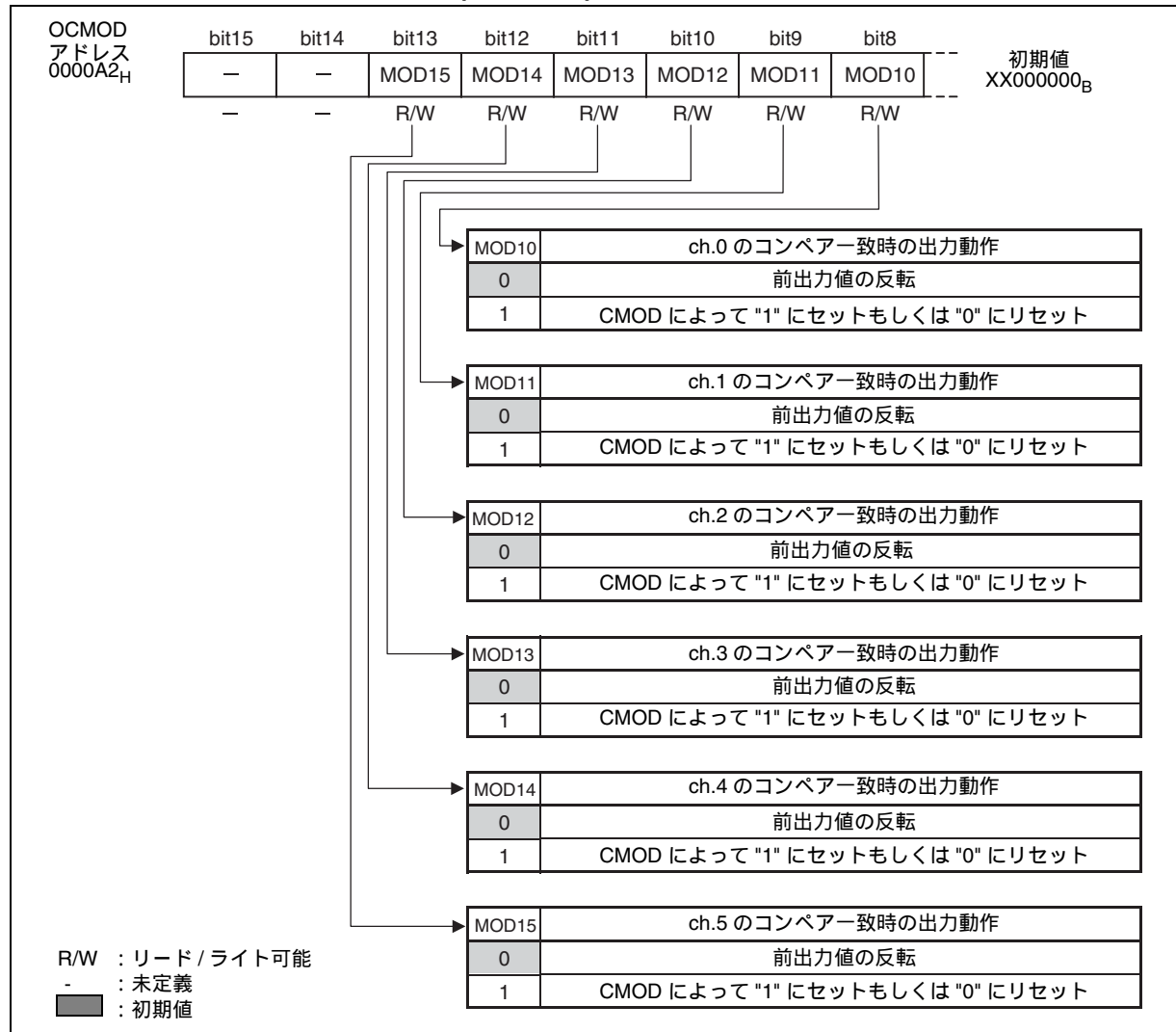


表 12.4-6 コンペアモード制御レジスタ (OCMOD)

ビット名		機能
bit15, bit14	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは動作に影響しません。
bit13	MOD15 : ch.5 のコンペア一致 モード設定ビット	<ul style="list-style-type: none"> このビットによりアウトプットコンペアの出力のコンペア一致時の動作を指示します。 初期値は, "0" です。 "0" のときは, 一致時に出力値を反転します。 "1" のときは, 一致時に出力値を "1" にセット, もしくは "0" にリセットします。セット / リセットの切換えはコンペア制御レジスタ (OCSH) の CMOD ビットにて設定します。 値を書き込む場合は, 必ず前もってコンペア動作を停止させてください。 CMOD の設定は, ch.0, ch.1 と ch.2, ch.3 と ch.4, ch.5 で設定となります。 <ul style="list-style-type: none"> ch.0, ch.1 で独立してリセット / セットの設定はできません。 ch.2, ch.3 で独立してリセット / セットの設定はできません。 ch.4, ch.5 で独立してリセット / セットの設定はできません。
bit12	MOD14 : ch.4 のコンペア一致 モード設定ビット	
bit11	MOD13 : ch.3 のコンペア一致 モード設定ビット	
bit10	MOD12 : ch.2 のコンペア一致 モード設定ビット	
bit9	MOD11 : ch.1 のコンペア一致 モード設定ビット	
bit8	MOD10 : ch.0 のコンペア一致 モード設定ビット	

12.4.8 インพุットキャプチャデータレジスタ (IPCPH0 ~ IPCPH3, IPCPL0 ~ IPCPL3)

インพุットキャプチャデータレジスタは、入力波形の有効エッジが検出されたときのフリーランタイムのカウント値を保持するために使用します。

■ インพุットキャプチャデータレジスタ (IPCPH0 ~ IPCPH3, IPCPL0 ~ IPCPL3)

インพุットキャプチャデータレジスタ (上位)

IPCPH0 ~ IPCPH3

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000AC _H	CP15	CP14	CP13	CP12	CP11	CP10	CP09	CP08	XXXXXXXX _B
0000AE _H	R	R	R	R	R	R	R	R	
0000B0 _H									
0000B2 _H									

インพุットキャプチャデータレジスタ (下位)

IPCPL0 ~ IPCPL3

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000AD _H	CP07	CP06	CP05	CP04	CP03	CP02	CP01	CP00	XXXXXXXX _B
0000AF _H	R	R	R	R	R	R	R	R	
0000B1 _H									
0000B3 _H									

R : リードオンリ

X : 不定

このレジスタは、対応する外部端子入力波形の有効エッジが検出されたときのフリーランタイム値を格納するために使用します (このレジスタへアクセスする場合は、ハーフワードもしくはワードアクセス命令をご使用ください。このレジスタにデータを書き込むことはできません)。

以上の説明中のフリーランタイムはインพุットキャプチャが選択しているフリーランタイムの動作状態についてです。

12.4.9 インพุットキャプチャ状態制御 /PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01)

インพุットキャプチャ状態制御 /PPG 出力制御レジスタ (ICSH23, ICSL23, PICSH01, PICSL01) は、エッジ選択、割込み要求許可、割込み要求フラグおよび PPG 出力を制御するために使用します。また、インพุットキャプチャ 2, 3 において検出された有効なエッジを示すためにも使用します。

■ インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 上位 (ICSH23)

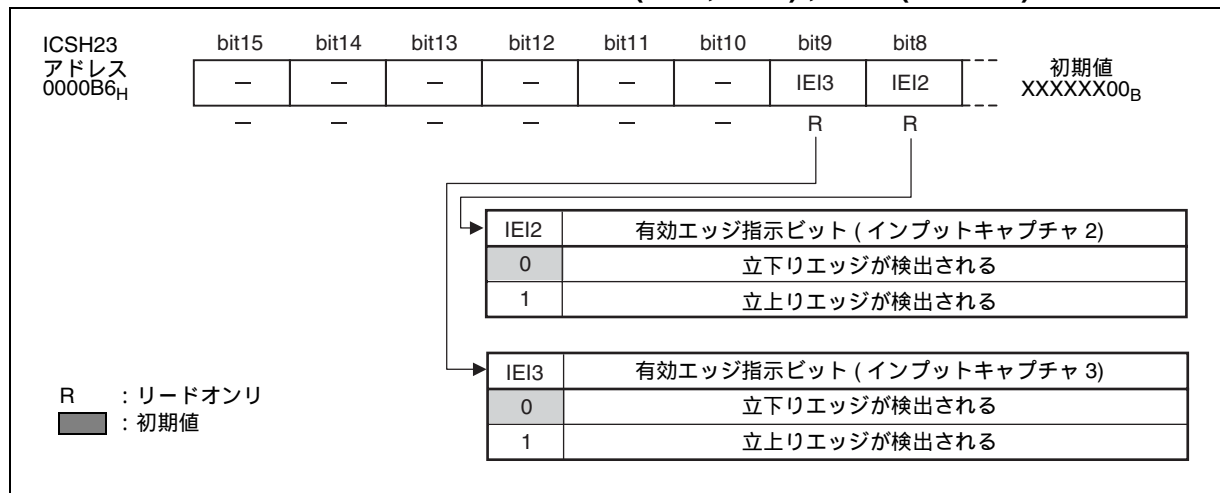


表 12.4-7 インพุットキャプチャ状態制御 (ch.2, ch.3) , 上位 (ICSH23)

ビット名		機能
bit15 ~ bit10	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは動作に影響しません。
bit9	IEI3 : 有効エッジ指示 ビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> このビットは, キャプチャレジスタ 3 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。 立下りエッジが検出されると, このビットに "0" が書き込まれます。 立上りエッジが検出されると, このビットに "1" が書き込まれます。 このビットは読出し専用ビットです。 <p>(注意事項) インพุットキャプチャ状態制御レジスタ下位 (ICSL23) の EG31, EG30 : bit3, bit2=00_B の場合, 読出し値は意味がありません。</p>
bit8	IEI2 : 有効エッジ指示 ビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> このビットは, キャプチャレジスタ 2 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。 立下りエッジが検出されると, このビットに "0" が書き込まれます。 立上りエッジが検出されると, このビットに "1" が書き込まれます。 このビットは, 読出し専用ビットです。 <p>(注意事項) インพุットキャプチャ状態制御レジスタ下位 (ICSL23) の EG21, EG20 : bit1, bit0=00_B の場合, 読出し値は意味がありません。</p>

■ インพุットキャプチャ状態制御レジスタ (ch.2, ch.3) , 下位 (ICSL23)

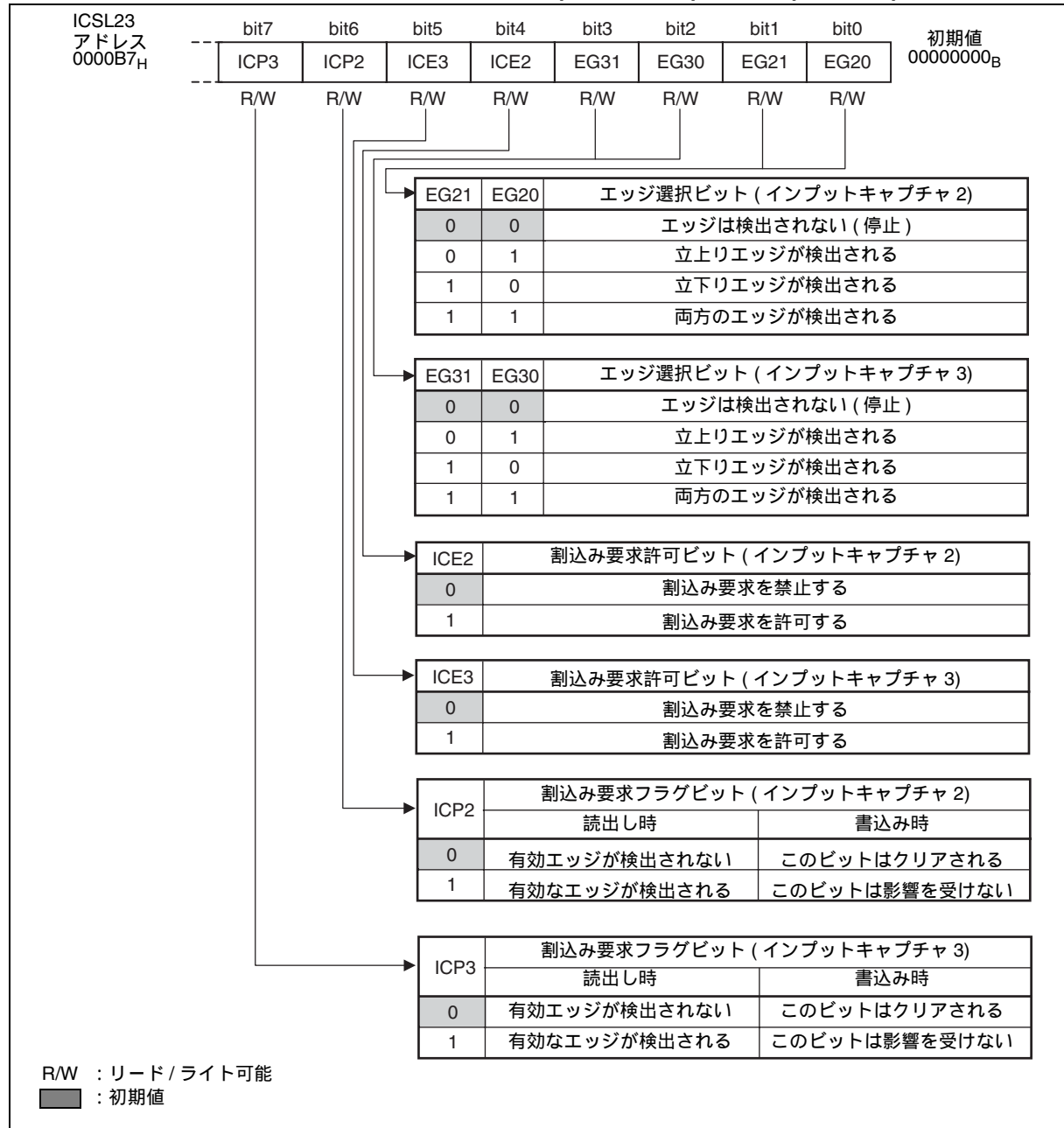


表 12.4-8 インพุットキャプチャ状態制御レジスタ (ch.2, ch.3), 下位 (ICSL23)

ビット名		機能
bit7	ICP3 : 割込み要求フラ グビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ3の割込み要求フラグとして使用します。 このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。 割込み許可ビット (ICE3 : bit5) が設定されている間に有効エッジが検出されると、直ちに割込みを発生することができます。 このビットに "0" を設定した場合：このビットはクリアされます。 このビットに "1" を設定した場合：このビットは影響を受けません。 リードモディファイライト (RMW) 時は、必ず "1" が読み出されます。
bit6	ICP2 : 割込み要求フラ グビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ2の割込み要求フラグとして使用します。 このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。 割込み許可ビット (ICE2 : bit4) が設定されている間に有効エッジが検出されると、直ちに割込みを発生することができます。 このビットに "0" を設定した場合：このビットはクリアされます。 このビットに "1" を設定した場合：このビットは影響を受けません。 リードモディファイライト (RMW) 時は、必ず "1" が読み出されます。
bit5	ICE3 : 割込み要求許可 ビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ3のインพุットキャプチャ割込み要求を許可するために使用します。 このビットに "0" を設定した場合、割込み要求を禁止します。 このビットに "1" が設定されている間に割込みフラグ (ICP3 : bit7) が設定されると、インพุットキャプチャ 3 割込みが生成されます。
bit4	ICE2 : 割込み要求許可 ビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ2のインพุットキャプチャ割込み要求を許可するために使用します。 このビットに "0" を設定した場合、割込み要求を禁止します。 このビットに "1" が設定されている間に割込みフラグ (ICP2 : bit6) が設定されると、インพุットキャプチャ 2 割込みが生成されます。
bit3, bit2	EG31, EG30 : エッジ選択 ビット (インพุット キャプチャ 3)	<ul style="list-style-type: none"> これらのビットは、インพุットキャプチャ3の外部入力の有効エッジ極性を指定するために使用します。 これらのビットは、インพุットキャプチャ3の動作を有効にするためにも使用します。
bit1, bit0	EG21, EG20 : エッジ選択 ビット (インพุット キャプチャ 2)	<ul style="list-style-type: none"> これらのビットは、インพุットキャプチャ2の外部入力の有効エッジ極性を指定するために使用します。 これらのビットは、インพุットキャプチャ2の動作を有効にするためにも使用します。

■ PPG 出力制御レジスタ, 上位 (PICSH01)

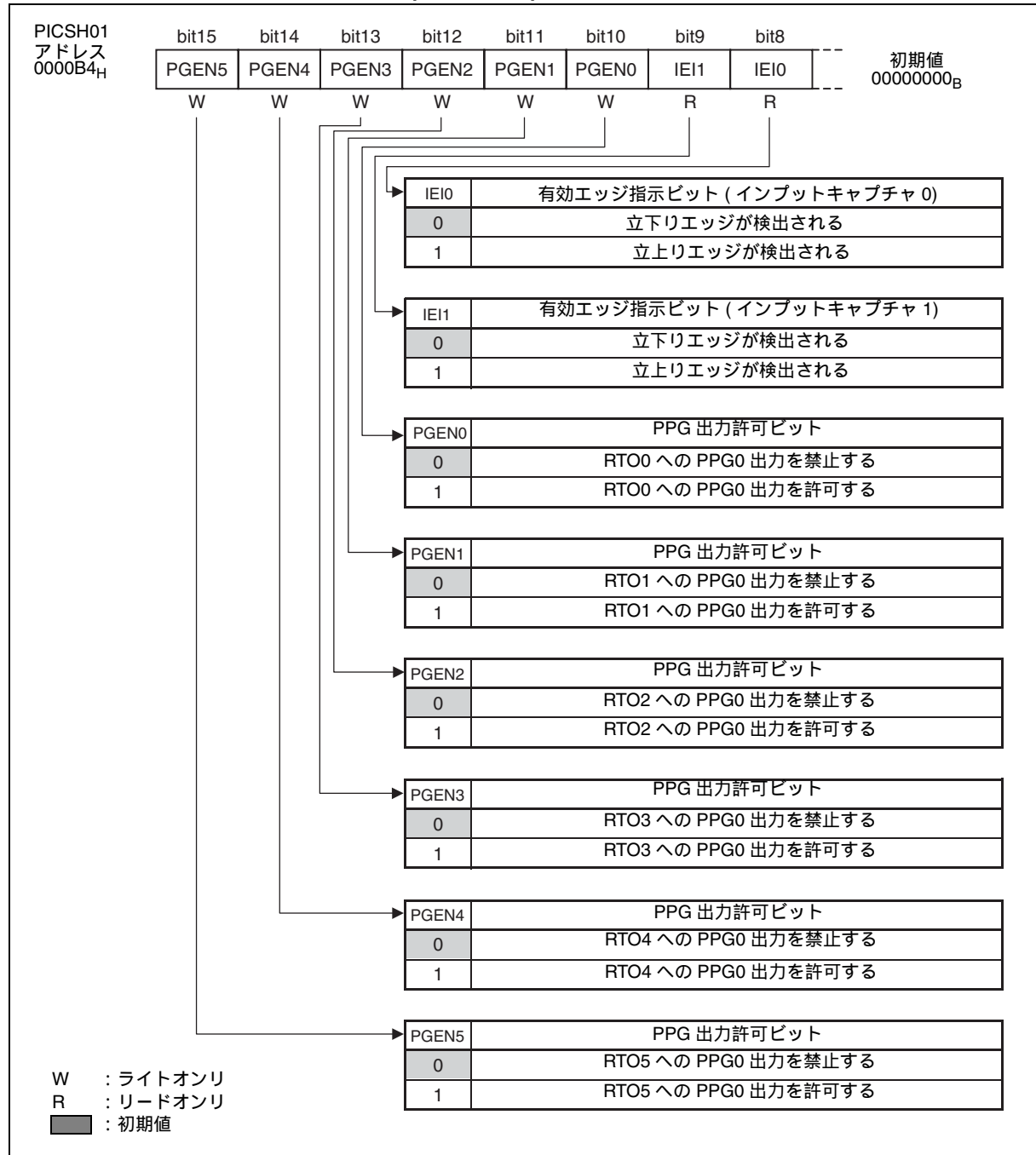


表 12.4-9 PPG 出力制御レジスタ上位 (PICSH01)

ビット名		機能
bit15 ~ bit10	PGEN5 ~ PGEN0 : PPG 出力許可 ビット	<ul style="list-style-type: none"> これらのビットは, RTO0, RTO1, RTO2, RTO3, RTO4, RTO5へのPPG0出力を選択するために使用します。 書込みのみ可能です。
bit9	IEI1 : 有効エッジ指示 ビット (インプット キャプチャ 1)	<ul style="list-style-type: none"> このビットは, キャプチャレジスタ 1 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。 立下りエッジが検出されると, このビットに "0" が書き込まれます。 立上りエッジが検出されると, このビットに "1" が書き込まれます。 このビットは読出し専用ビットです。 <p>(注意事項) インプットキャプチャ状態制御レジスタ下位 (PICSL01) の EG11, EG10: bit3, bit2=00_B の場合, 読出し値は意味がありません。</p>
bit8	IEI0 : 有効エッジ指示 ビット (インプット キャプチャ 0)	<ul style="list-style-type: none"> このビットは, キャプチャレジスタ 0 の有効エッジ指示ビットであり, 立上りエッジまたは立下りエッジが検出されたことを示します。 立下りエッジが検出されると, このビットに "0" が書き込まれます。 立上りエッジが検出されると, このビットに "1" が書き込まれます。 このビットは, 読出し専用ビットです。 <p>(注意事項) インプットキャプチャ状態制御レジスタ下位 (PICSL01) の EG01, EG00: bit1, bit0=00_B の場合, 読出し値は意味がありません。</p>

■ インพุットキャプチャ状態制御レジスタ (ch.0, ch.1) , 下位 (PICSL01)

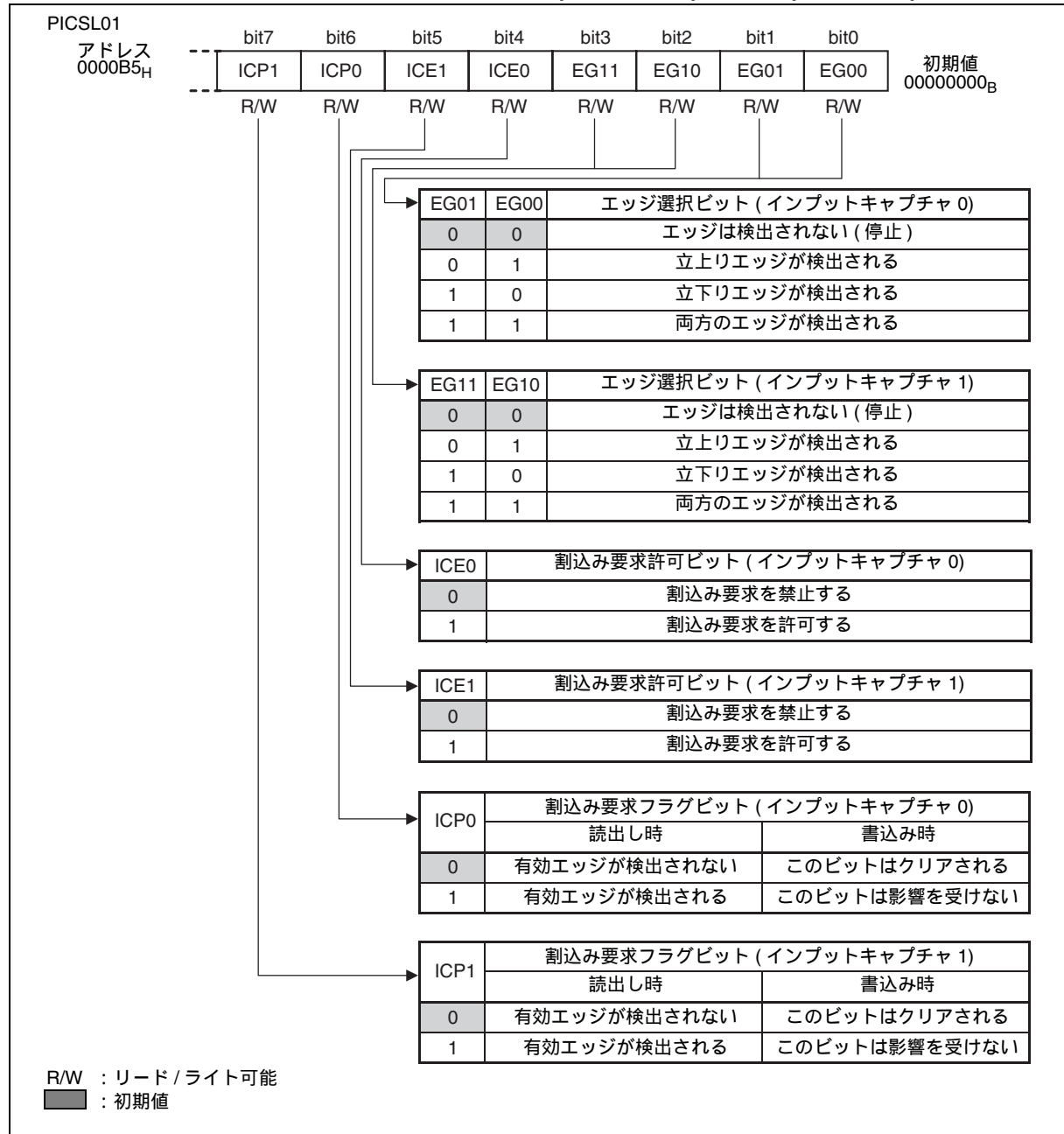


表 12.4-10 インพุットキャプチャ状態制御レジスタ (ch.0, ch.1), 下位 (PICSLO1)

ビット名		機能
bit7	ICP1 : 割込み要求フラ グビット (インพุット キャプチャ 1)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ1の割込み要求フラグとして使用します。 このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。 割込み許可ビット (ICE1 : bit5) が設定されている間に有効エッジが検出されると、直ちに割込みが生成されます。 このビットに "0" を設定した場合：このビットはクリアされます。 このビットに "1" を設定した場合：このビットは影響を受けません。 リードモディファイライト (RMW) 時は、必ず "1" が読み出されます。
bit6	ICP0 : 割込み要求フラ グビット (インพุット キャプチャ 0)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ0の割込み要求フラグとして使用します。 このビットは、外部入力端子の有効エッジが検出されると直ちに "1" が設定されます。 割込み許可ビット (ICE0 : bit4) が設定されている間に有効エッジが検出されると、直ちに割込みが生成されます。 このビットに "0" を設定した場合：このビットはクリアされます。 このビットに "1" を設定した場合：このビットは影響を受けません。 リードモディファイライト (RMW) 時は、必ず "1" が読み出されます。
bit5	ICE1 : 割込み要求許可 ビット (インพุット キャプチャ 1)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ1のインพุットキャプチャ割込み要求を許可するために使用します。 このビットに "0" を設定した場合、割込み要求を禁止します。 このビットに "1" が設定されている間に割込みフラグ (ICP1 : bit7) が設定されると、インพุットキャプチャ 1 割込みが生成されます。
bit4	ICE0 : 割込み要求許可 ビット (インพุット キャプチャ 0)	<ul style="list-style-type: none"> このビットは、インพุットキャプチャ0のインพุットキャプチャ割込み要求を許可するために使用します。 このビットに "0" を設定した場合、割込み要求を禁止します。 このビットに "1" が設定されている間に割込みフラグ (ICP0 : bit6) が設定されると、インพุットキャプチャ 0 割込みが生成されます。
bit3, bit2	EG11, EG10 : エッジ選択 ビット (インพุット キャプチャ 1)	<ul style="list-style-type: none"> これらのビットは、インพุットキャプチャ1の外部入力の有効エッジ極性を指定するために使用します。 これらのビットは、インพุットキャプチャ1の動作を有効にするためにも使用します。
bit1, bit0	EG01, EG00 : エッジ選択 ビット (インพุット キャプチャ 0)	<ul style="list-style-type: none"> これらのビットは、インพุットキャプチャ0の外部入力の有効エッジ極性を指定するために使用します。 これらのビットは、インพุットキャプチャ0の動作を有効にするためにも使用します。

12.4.10 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2)

16 ビットデッドタイムレジスタは、16 ビットデッドタイムのコンペア値を保持します。

■ 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2)

16 ビットデッドタイムレジスタ (上位)

TMRRH0 ~ TMRRH2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000BC _H	TR15	TR14	TR13	TR12	TR11	TR10	TR09	TR08	XXXXXXXX _B
0000BE _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000C0 _H									

16 ビットデッドタイムレジスタ (下位)

TMRRL0 ~ TMRRL2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000BD _H	TR07	TR06	TR05	TR04	TR03	TR02	TR01	TR00	XXXXXXXX _B
0000BF _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0000C1 _H									

R/W : リード / ライト可能

X : 不定

これらのレジスタは、16 ビットデッドタイムの比較値を格納するために使用します。
 これらのレジスタ値は、16 ビットデッドタイムが動作を開始するとリロードされます。
 タイマ動作中にこれらのレジスタに値が再書き込みされると、この新しい値は次のタイマ開始 / 動作時に有効になります。
 このレジスタへアクセスする場合は、ハーフワードもしくはワードアクセス命令をご使用ください。
 デッドタイムタイマモード時は、これらのレジスタはノンオーバーラップ時間を設定するために使用します。

$$\text{ノンオーバーラップ時間} = (\text{設定値}) \times \text{選択されたクロック}$$

< 注意事項 >

"0000_H" を設定することはできません。

タイマモード時は、これらのレジスタは PPG0 タイマ動作の GATE 時間を設定するために使用します。

$$\text{GATE 時間} = (\text{設定値}) \times \text{選択されたクロック}$$

< 注意事項 >

"0000_H" を設定することはできません。

12.4.11 16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2)

16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) は、波形ジェネレータの動作モード、割り込み要求許可、割り込み要求フラグ、GATE 信号許可、および出力レベル極性を制御するために使用します。

■ 16 ビットデッドタイム制御レジスタ, 上位 (DTCR0)

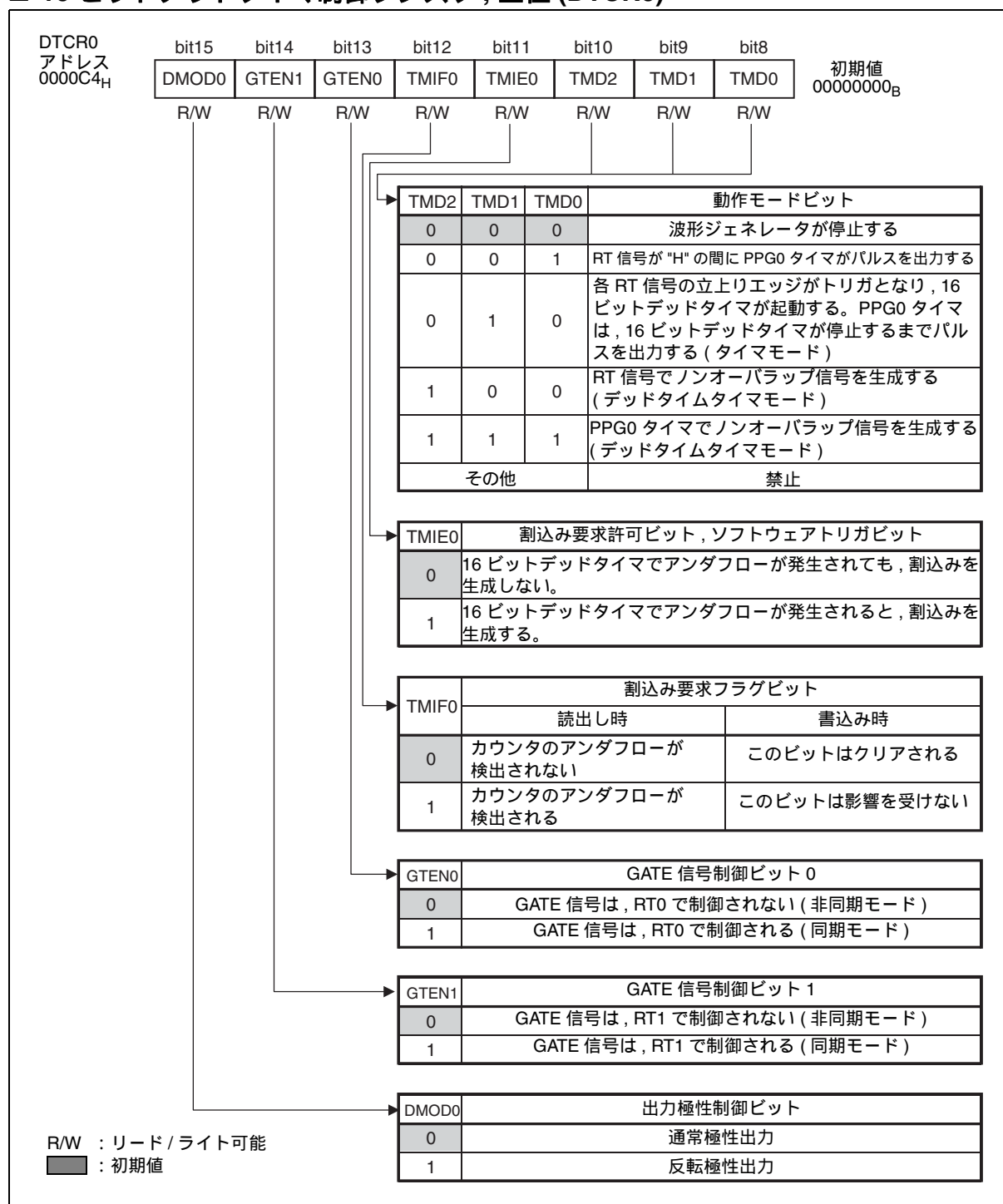


表 12.4-11 16 ビットデッドタイム制御レジスタ, 上位 (DTCR0) (1 / 2)

ビット名		機能
bit15	DMOD0 : 出力極性制御 ビット	<ul style="list-style-type: none"> このビットは、デッドタイムタイマモードにおいて U/V/W の出力を設定するために使用します。 このビットを設定すると、U/V/W の出力極性は反転します。 <p>(注意事項) このビットは、デッドタイムタイマモードが選択されていない場合 (TMD2 : bit10=0) は意味がありません。</p>
bit14	GTEN1 : GATE 信号制御 ビット 1	このビットは、RT1 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit13	GTEN0 : GATE 信号制御 ビット 0	このビットは、RT0 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit12	TMIF0 : 割込み要求フラ グビット	<ul style="list-style-type: none"> このビットは、16 ビットデッドタイムの割込み要求フラグとして使用します。 このビットは、16 ビットデッドタイムでアンダフローが発生すると "1" が設定されます。 このビットに "0" を書き込むと、このビットはクリアされます。"1" を書き込んででも、このビットは影響されません。 リードモディファイライト (RMW) 時は、必ず "1" が読み出されます。 <p>(注意事項) このビットは、TMD2 ~ TMD0 : bit10 ~ bit8 が "000_B" または "001_B" の場合のみ機能し、ほかの値の場合は必ず "0" になります。 ソフトウェアクリア ("0" 書込み) とハードウェアセット (16 ビットデッドタイム 0 でアンダフローが発生する) が同時に発生した場合は、ソフトウェアクリアがハードウェアセットよりも優先され、このビットはクリアされます。</p>
bit11	TMIE0 : 割込み要求許可 ビット, ソフト ウェアトリガ ビット	<ul style="list-style-type: none"> このビットは、16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用します。 TMD2 ~ TMD0 : bit10 ~ bit8 が "000_B" または "001_B" の場合、このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると、16 ビットデッドタイムのトリガとなり、値がリロードされ、ダウンカウントが開始します。 このビットが "1" であり、割込み要求フラグビット (TMIF0 : bit12) が "1" の場合、割込み要求が CPU へ送られます。 <p>(注意事項) 16 ビットデッドタイムを再度トリガとする場合には、このビットに "1" を書き込む前に必ず "0" を書き込んでください。</p>

表 12.4-11 16 ビットデッドタイム制御レジスタ , 上位 (DTCR0) (2 / 2)

ビット名	機能
bit10 ~ bit8	<div data-bbox="343 616 478 750">TMD2 ~ TMD0 : 動作モード ビット</div> <ul style="list-style-type: none"> これらのビットは , 波形ジェネレータの動作モードを選択するために使用します。 TMD2 ~ TMD0 : bit10 ~ bit8 が "000_B" の場合 , アウトプットコンペアの RT0 と RT1 の信号は , RTO0 と RTO1 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 TMD2 ~ TMD0 : bit10 ~ bit8 が "001_B" の場合 , アウトプットコンペアの RT0 と RT1 の信号は , PPG0 出力が禁止 (PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN0Cbit10=0, PGEN1 : bit11=0) になると , RTO0 と RTO1 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 <p>(注意事項) デッドタイムタイマモードで波形ジェネレータを動作させるには , 必ず RT1 に対して 2 チャンネルモード (コンペア制御レジスタ上位 (OCSH1) の CMOD : bit12=1) を選択してください。 TMD2 ~ TMD0 : bit10 ~ bit8 が "111_B" の場合 , RTO0 の出力と RTO1 の出力は , PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN0 : bit10=0, PGEN1 : bit11=0 の設定に依存しません。</p>

■ 16 ビットデッドタイム制御レジスタ, 下位 (DTCR1)

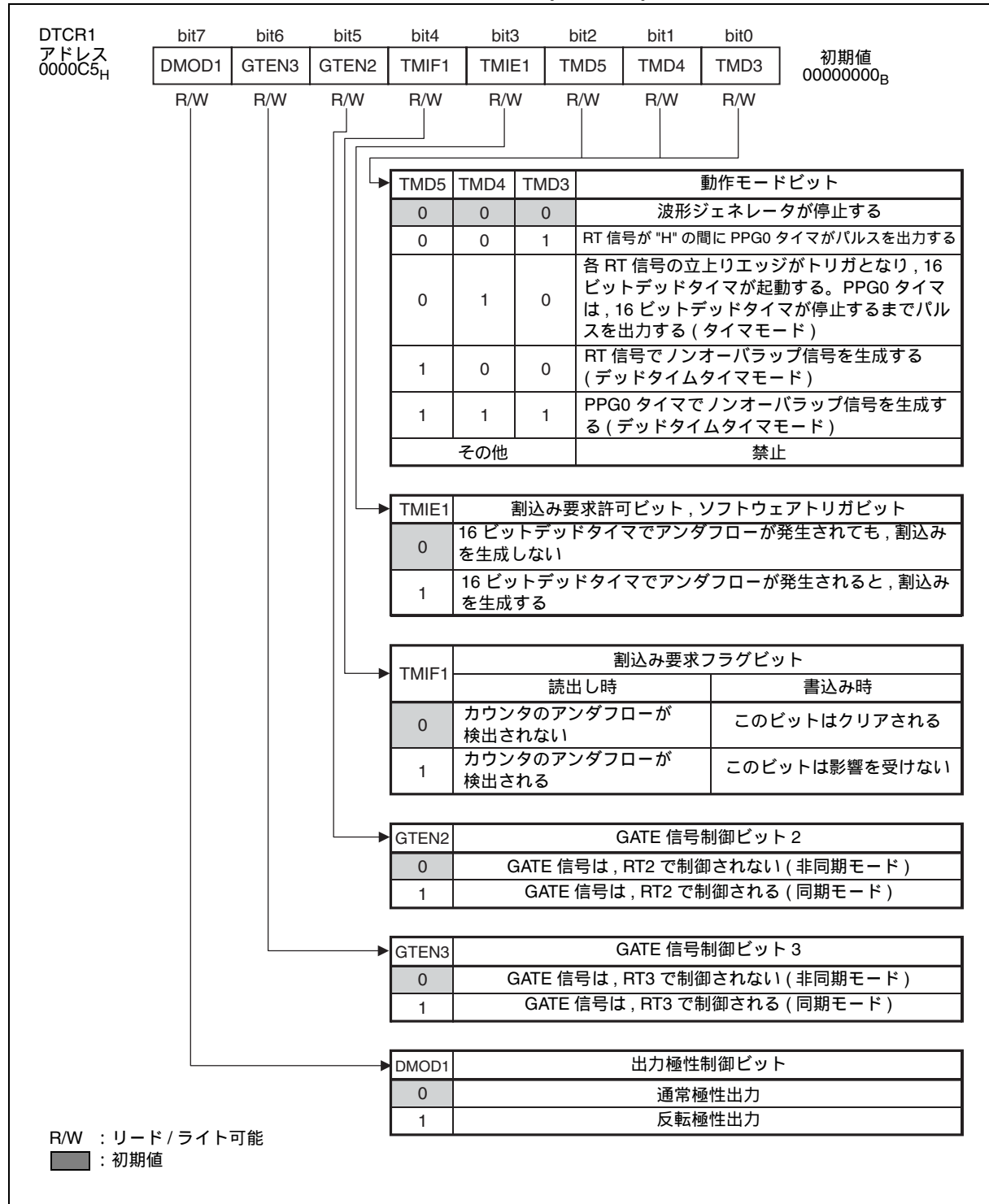


表 12.4-12 16 ビットデッドタイム制御レジスタ, 下位 (DTCR1) (1 / 2)

ビット名		機能
bit7	DMOD1 : 出力極性制御 ビット	<ul style="list-style-type: none"> このビットは, デッドタイムタイマモードにおいて U/V/W の出力を設定するために使用します。 このビットを設定すると, U/V/W の出力極性は反転します。 <p>(注意事項)</p> <p>このビットは, デッドタイムタイマモードが選択されていない場合 (TMD5 : bit2=0) は意味がありません。</p>
bit6	GTEN3 : GATE 信号制御 ビット 3	このビットは, RT3 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit5	GTEN2 : GATE 信号制御 ビット 2	このビットは, RT2 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit4	TMIF1 : 割込み要求フラ グビット	<ul style="list-style-type: none"> このビットは, 16 ビットデッドタイムの割込み要求フラグとして使用します。 このビットは, 16 ビットデッドタイムでアンダフローが発生すると "1" が設定されます。 このビットに "0" を書き込むと, このビットはクリアされます。"1" を書き込んで, このビットは影響されません。 リードモディファイライト (RMW) 時は, 必ず "1" が読み出されます。 <p>(注意事項)</p> <p>このビットは, (TMD5 ~ TMD3 : bit2 ~ bit0) が "000_B" または "001_B" の場合のみ機能し, ほかの値の場合は必ず "0" になります。ソフトウェアクリア ("0" 書込み) とハードウェアセット (16 ビットデッドタイム 1 でアンダフローが発生する) が同時に発生した場合は, ソフトウェアクリアがハードウェアセットよりも優先され, このビットはクリアされます。</p>
bit3	TMIE1 : 割込み要求許可 ビット, ソフト ウェアトリガ ビット	<ul style="list-style-type: none"> このビットは, 16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用します。 TMD5 ~ TMD3 : bit2 ~ bit0 が "000_B" または "001_B" の場合, このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると, 16 ビットデッドタイムのトリガとなり, 値がリロードされ, ダウンカウントが開始します。 このビットが "1" であり, 割込み要求フラグビット (TMIF1 : bit4) が "1" の場合, 割込み要求が CPU へ送られます。 <p>(注意事項)</p> <p>16 ビットデッドタイムを再度トリガとする場合には, このビットに "1" を書き込む前に必ず "0" を書き込んでください。</p>

表 12.4-12 16 ビットデッドタイム制御レジスタ, 下位 (DTCR1) (2 / 2)

ビット名	機能
bit2 ~ bit0	<div data-bbox="343 622 478 757">TMD5 ~ TMD3 : 動作モード ビット</div> <ul style="list-style-type: none"> これらのビットは , 波形ジェネレータの動作モードを選択するために使用します。 TMD5 ~ TMD3 : bit2 ~ bit0 が "000_B" の場合 , アウトプットコンペアの RT2 と RT3 の信号は , RTO2 と RTO3 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 TMD5 ~ TMD3 : bit2 ~ bit0 が "001_B" の場合 , アウトプットコンペアの RT2 と RT3 の信号は , PPG0 出力が禁止 (PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN2 : bit12=0, PGEN3 : bit13=0) になると , RTO2 と RTO3 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 <p>(注意事項) デッドタイムタイマモードで波形ジェネレータを動作させるには , 必ず RT3 に対して 2 チャネルモード (コンペア制御レジスタ上位 (OCSH3) の CMOD : bit12=1) を選択してください。 TMD5 ~ TMD3 : bit2 ~ bit0 が "111_B" の場合 , RTO2 の出力と RTO3 の出力は , PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN2 : bit12=0, PGEN3 : bit13=0 の設定に依存しません。</p>

■ 16 ビットデッドタイム制御レジスタ, 上位 (DTCR2)

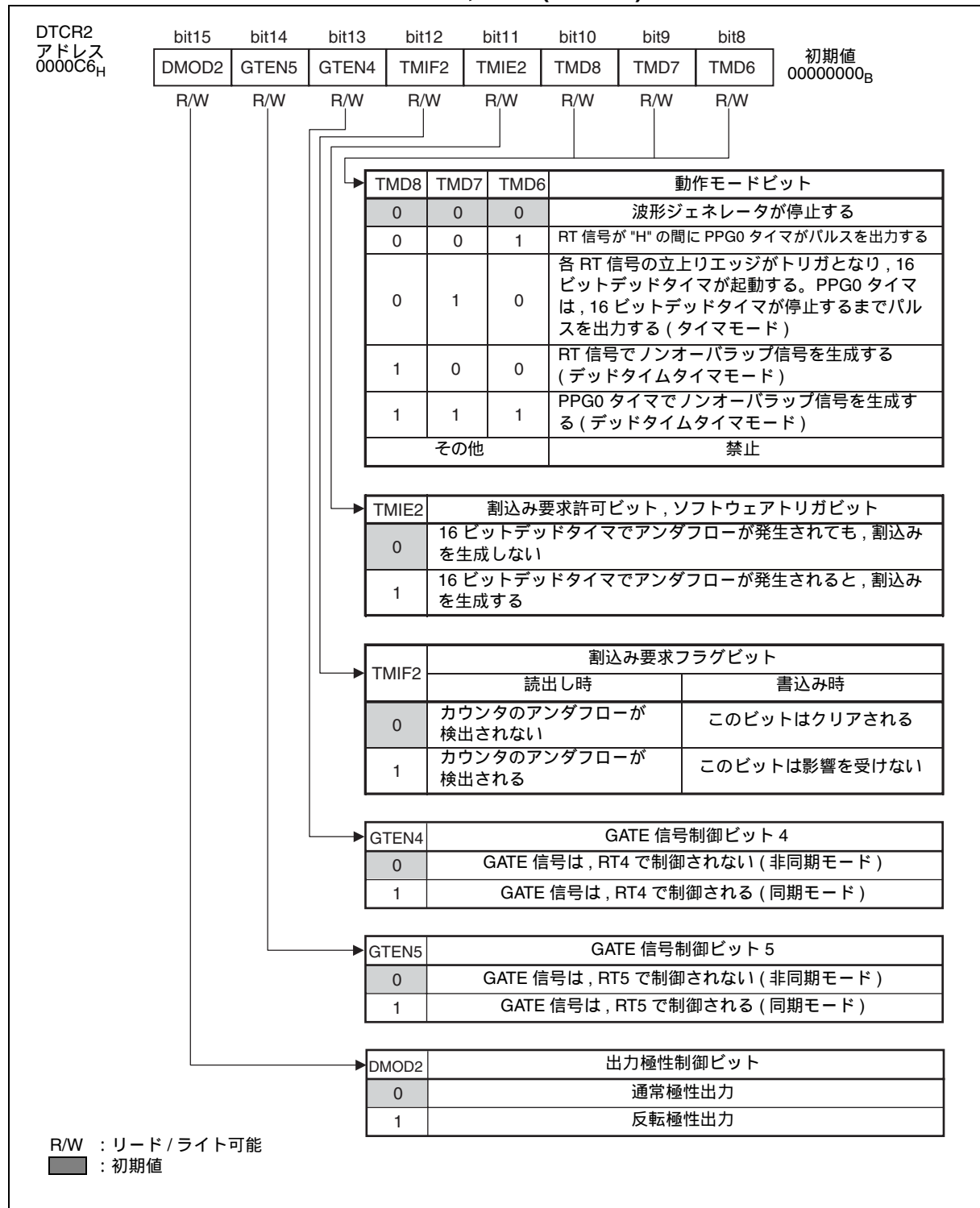


表 12.4-13 16 ビットデッドタイム制御レジスタ, 上位 (DTCR2) (1 / 2)

ビット名		機能
bit15	DMOD2 : 出力極性制御 ビット	<ul style="list-style-type: none"> このビットは, デッドタイムタイマモードにおいて U/V/W の出力を設定するために使用します。 このビットを設定すると, U/V/W の出力極性は反転します。 <p>(注意事項)</p> <p>このビットは, デッドタイムタイマモードが選択されていない場合 (TMD8 : bit10=0) は意味がありません。</p>
bit14	GTEN5 : GATE 信号制御 ビット 5	このビットは, RT5 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit13	GTEN4 : GATE 信号制御 ビット 4	このビットは, RT4 で PPG0 タイマの GATE 信号出力を制御するために使用します。
bit12	TMIF2 : 割込み要求フラ グビット	<ul style="list-style-type: none"> このビットは, 16 ビットデッドタイムの割込み要求フラグとして使用します。 このビットは, 16 ビットデッドタイムでアンダフローが発生すると "1" が設定されます。 このビットに "0" を書き込むと, このビットはクリアされます。"1" を書き込んで, このビットは影響されません。 リードモディファイライト (RMW) 時は, 必ず "1" が読み出されます。 <p>(注意事項)</p> <p>このビットは, (TMD8 ~ TMD6 : bit10 ~ bit8) が "000_B" または "001_B" の場合のみ機能し, ほかの値の場合は必ず "0" になります。 ソフトウェアクリア ("0" 書込み) とハードウェアセット (16 ビットデッドタイム 2 でアンダフローが発生する) が同時に発生した場合は, ソフトウェアクリアがハードウェアセットよりも優先され, このビットはクリアされます。</p>
bit11	TMIE2 : 割込み要求許可 ビット, ソフト ウェアトリガ ビット	<ul style="list-style-type: none"> このビットは, 16 ビットデッドタイムのソフトウェアトリガビットおよび割込み許可ビットとして使用します。 TMD8 ~ TMD6 : bit10 ~ bit8 が "000_B" または "001_B" の場合, このビットは 16 ビットデッドタイムのソフトウェアトリガとして使用されます。このビットを "0" から "1" へ変更すると, 16 ビットデッドタイムのトリガとなり, 値がリロードされ, ダウンカウントが開始します。 このビットが "1" であり, 割込み要求フラグビット (TMIF2 : bit12) が "1" の場合, 割込み要求が CPU へ送られます。 <p>(注意事項)</p> <p>16 ビットデッドタイムを再度トリガとする場合には, このビットに "1" を書き込む前に必ず "0" を書き込んでください。</p>

表 12.4-13 16 ビットデッドタイム制御レジスタ , 上位 (DTCR2) (2 / 2)

ビット名	機能
bit10 ~ bit8	<div data-bbox="344 622 480 757">TMD8 ~ TMD6 : 動作モード ビット</div> <ul style="list-style-type: none"> これらのビットは , 波形ジェネレータの動作モードを選択するために使用します。 TMD8 ~ TMD6 : bit10 ~ bit8 が "000_B" の場合 , アウトプットコンペアの RT4 と RT5 の信号は , RTO4 と RTO5 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 TMD8 ~ TMD6 : bit10 ~ bit8 が "001_B" の場合 , アウトプットコンペアの RT4 と RT5 の信号は , PPG0 出力が禁止 (PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN4 : bit14=0, PGEN5 : bit15=0) になると , RTO4 と RTO5 のそれぞれから出力されます。また , 16 ビットデッドタイムはリロードタイマとしても使用することができます。 <p>(注意事項) デッドタイムタイマモードで波形ジェネレータを動作させるには , 必ず RT5 に対して 2 チャンネルモード (コンペア制御レジスタ上位 (OCSH5) の CMOD : bit12=1) を選択してください。 TMD8 ~ TMD6 : bit10 ~ bit8 が "111_B" の場合 , RTO4 の出力と RTO5 の出力は , PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN4 : bit14=0, PGEN5 : bit15=0 の設定に依存しません。</p>

12.4.12 波形制御レジスタ (SIGCR1/SIGCR2)

波形制御レジスタは、動作クロック周波数、ノイズキャンセル機能有効、DTTI 端子入力有効、および DTTI 端子割込みを制御するために使用します。

■ 波形制御レジスタ 1 (SIGCR1)

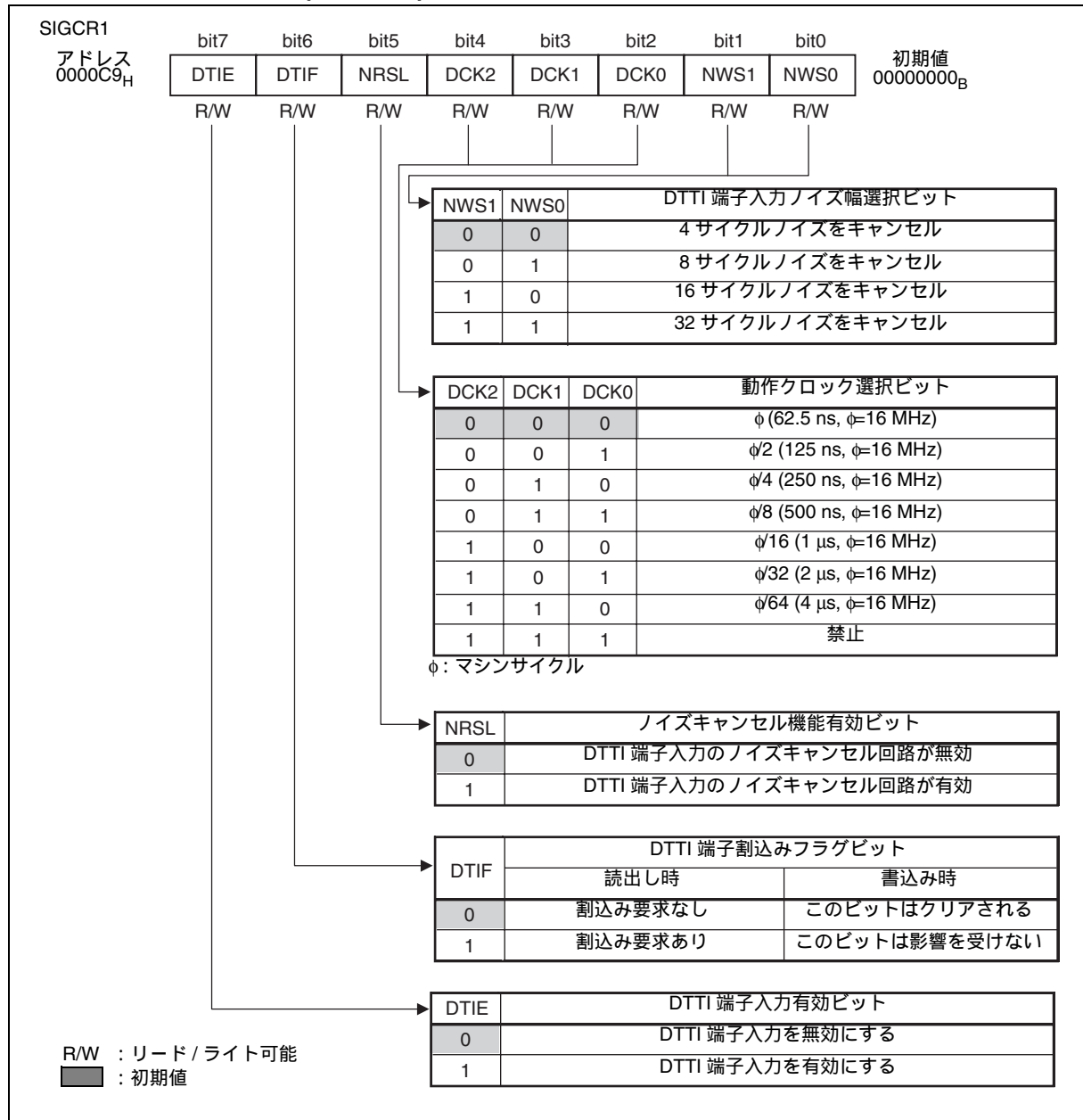


表 12.4-14 波形制御レジスタ 1 (SIGCR1)

ビット名		機能
bit7	DTIE : DTTI 端子入力 有効ビット	このビットは , RTO0 ~ RTO5 端子の出力レベル制御用 DTTI 端子入力を有効にするために使用します。
bit6	DTIF : DTTI 端子 割込みフラグ ビット	<ul style="list-style-type: none"> このビットは , DTTI 端子の割込みフラグです。 DTTI 端子入力が有効になり (DTIE : bit7=1) , DTTI 端子の "L" レベルが検出されると , このビットが設定され , 割込み要求が CPU へ送られます。 このビットに "0" を設定した場合 : このビットはクリアされます。 このビットに "1" を設定した場合 : このビットは影響を受けません。 リードモディファイライト (RMW) 時では , 必ず "1" が読み出されます。 <p>(注意事項) ノイズキャンセル機能が有効 (NRSL : bit5=1) の場合は , ノイズとして指定したパルス幅以上の DTTI 端子 "L" レベル入力により , 本ビットに "1" が設定されます。 ソフトウェアクリア ("0" 書込み) とハードウェアセット (DTTI 端子の "L" レベル検出) が同時に発生した場合は , ソフトウェアクリアがハードウェアセットよりも優先され , このビットはクリアされます。</p>
bit5	NRSL : ノイズキャン セル機能有効 ビット	<ul style="list-style-type: none"> このビットは , ノイズキャンセル機能を有効にするために使用します。 このビットに "0" を設定した場合 , ノイズキャンセル機能は無効になります。 このビットに "1" を設定した場合 , ノイズキャンセル機能は有効になります。 ノイズキャンセル回路は , カウンタでオーバフローが発生するまで "L" レベルが保持されると , DTTI 端子入力信号を DTTI 制御回路に伝えます。カウンタは , DTTI 端子の "L" レベル入力で操作される n ビットカウンタです。n は , NWS1, NWS0 ビット : 1, 0 の設定に基づいて 2, 3, 4 または 5 のいずれかの値になります。 <p>(注意事項) ノイズキャンセル機能は無効とするには , 約 2^n マシンサイクルが必要になります。 ノイズキャンセル機能を有効とすると , 内部クロックが停止するモード (停止モードなど) 時は , DTTI 端子の入力が無効になります。</p>
bit4 ~ bit2	DCK2 ~ DCK0 : 動作クロック 選択ビット	これらのビットは , 16 ビットデッドタイムの動作クロックを選択するために使用します。
bit1, bit0	NWS1, NWS0 : DTTI 端子ノイ ズ幅選択ビット	これらのビットは , 除去する DTTI 端子ノイズパルス幅を選択するために使用します。

■ 波形制御レジスタ 2 (SIGCR2)

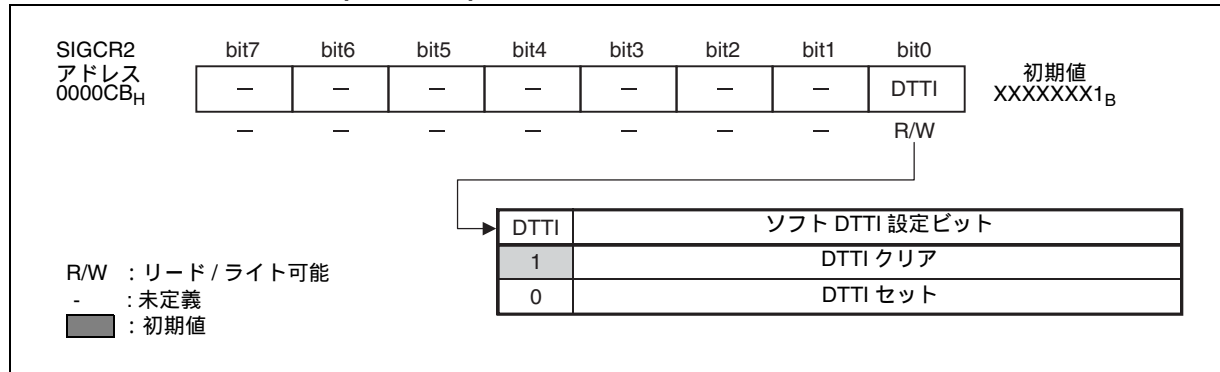


表 12.4-15 波形制御レジスタ 2 (SIGCR2)

ビット名		機能
bit7 ~ bit1	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは動作に影響しません。
bit0	DTTI : ソフト DTTI 設定 ビット	<ul style="list-style-type: none"> "0" を書き込むと, DTTI のセットとなります。 "1" を書き込むとクリアされます。 (注意事項) 本ビットの "1" 書き込みによる DTTI 端子割込み, ならびに DTTI 端子割込みフラグビット (DTIF) の変化はありません。

12.4.13 A/D 起動コンペアレジスタ (ADCOMP1, ADCOMP2, ADCOMPC1, ADCOMPC2)

コンペアレジスタ 1, 2 は, フリーランタイム 0 の値と一致したときに A/D コンバータ 1, 2 を起動させます。コンペアレジスタは, コンペア値を書き込むために使用します。

■ コンペアレジスタ 1, 2 (ADCOMP1, ADCOMP2)

コンペアレジスタレジスタ 1, 2 (上位)

ADCOMP1, ADCOMP2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.1 :0000CE _H	CMP15	CMP14	CMP13	CMP12	CMP11	CMP10	CMP09	CMP08	00000000 _B
ch.2 :0000D0 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

コンペアレジスタレジスタ 1, 2 (下位)

ADCOMP1, ADCOMP2

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.1 :0000CF _H	CMP07	CMP06	CMP05	CMP04	CMP03	CMP02	CMP01	CMP00	00000000 _B
ch.2 :0000D1 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

コンペアレジスタは, 16ビットフリーランタイム0のカウント値と比較するためのデータを書き込むレジスタで, フリーランタイム 0 とコンペア値が一致したときに, A/D を起動させることができます。

コンペアレジスタに書き込まれた値は, すぐに比較動作されます。

コンペアレジスタへの書込みは, ワードもしくはハーフワードで行ってください。

■ 制御レジスタ 1, 2 (ADCOMPC1, ADCOMPC2)

制御レジスタ 1

ADCOMPC1

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0000D3 _H	-	-	-	-	-	CE2	CE1	-	XXXXXX00X _B
	-	-	-	-	-	R/W	R/W	-	

制御レジスタ 2

ADCOMPC2

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0000D2 _H	-	-	SEL21	SEL20	SEL11	SEL10	-	-	XX0000XX _B
	-	-	R/W	R/W	R/W	R/W	-	-	

R/W : リード / ライト可能

X : 不定

- : 未定義

[bit2 ~ bit1] CE2, CE1 : A/D コンペア起動許可ビット

CEn	A/D コンペア起動許可ビット
0	A/D ユニット n のコンペア起動禁止 [初期値]
1	A/D ユニット n のコンペア起動許可

(n=1, 2 : A/D ユニットの番号)

[bit13 ~ bit10] SEL21, SEL20, SEL11, SEL10 : A/D コンペア起動要因選択ビット

SELn1	SELn0	A/D コンペア起動要因選択ビット
0	0	フリーランタイムと一致したときに起動 [初期値]
0	1	フリーランタイムのアップカウント時のみ一致したときに起動
1	0	フリーランタイムのダウンカウント時のみ一致したときに起動
1	1	設定禁止

(n=1, 2 : A/D ユニットの番号)

12.4.14 フリーランタイムセクタレジスタ (FSR0 ~ FSR2)

フリーランタイムセクタレジスタは、各インプットキャプチャ、アウトプットコンペアに対して、3 チャンネルあるフリーランタイムのいずれを割り当てるかを設定します。

■ フリーランタイムセクタレジスタ (FSR0 ~ FSR2)

FSR2									
アドレス	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16	初期値
000169 _H	ICU31	ICU30	ICU21	ICU20	ICU11	ICU10	ICU01	ICU00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
FSR1									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00016A _H	-	-	-	-	OCU51	OCU50	OCU41	OCU40	XXXX0000 _B
	-	-	-	-	R/W	R/W	R/W	R/W	
FSR0									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
00016B _H	OCU31	OCU30	OCU21	OCU20	OCU11	OCU10	OCU01	OCU00	00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									
X : 不定									
- : 未定義									

[bit23 ~ bit16] ICU_n1, ICU_n0 : インพุットキャプチャ用フリーランタイム選択ビット

ICU _n 1	ICU _n 0	インพุットキャプチャ用フリーランタイム選択ビット
0	0	フリーランタイム ch.0 選択 [初期値]
0	1	フリーランタイム ch.1 選択
1	0	フリーランタイム ch.2 選択
1	1	設定禁止

(n=0, 1, 2, 3 : インพุットキャプチャのチャンネル)

(注意事項) インพุットキャプチャ動作中に , 設定を書き換えしないでください。

[bit11 ~ bit0] OCUn1, OCUn0 : アウトพุットコンペア用フリーランタイム選択ビット

OCUn1	OCUn0	アウトพุットコンペア用フリーランタイム選択ビット
0	0	フリーランタイム ch.0 選択 [初期値]
0	1	フリーランタイム ch.1 選択
1	0	フリーランタイム ch.2 選択
1	1	設定禁止

(n=0, 1, 2, 3, 4, 5 : アウトพุットコンペアのチャンネル)

(注意事項) アウトพุットコンペア動作中に , 設定を書き換えしないでください。

12.5 多機能タイマ割込み

多機能タイマは、16 ビットフリーランタイム割込み、16 ビットアウトプットコンペア割込み、16 ビットインプットキャプチャ割込みおよび波形ジェネレータ割込みを生成することができます。

■ 16 ビットフリーランタイム割込み

16 ビットフリーランタイムの割込み制御ビットと割込み要因を表 12.5-1 に示します。

表 12.5-1 16 ビットフリーランタイムの割込み制御ビットと割込み要因

	16 ビットフリーランタイム	
	コンペアクリア	0 検出
割込み要求フラグビット	タイマ状態レジスタ上位 (TCCSH0, TCCSH1, TCCSH2) の ICLR : bit9	タイマ状態レジスタ上位 (TCCSH0, TCCSH1, TCCSH2) の IRQZF : bit14
割込み要求許可ビット	タイマ状態レジスタ上位 (TCCSH0, TCCSH1, TCCSH2) の ICRE : bit8	タイマ状態レジスタ上位 (TCCSH0, TCCSH1, TCCSH2) の IRQZE : bit13
割込み要因	16 ビットフリーランタイム値がコンペアクリアレジスタ (CPCLRH0/CPCLRL0, CPCLRH1/CPCLRL1, CPCLRH2/CPCLRL2) と一致する	16 ビットフリーランタイム値が "0" になる

16 ビットフリーランタイムの値がコンペアクリアレジスタ (CPCLRH0/CPCLRL0, CPCLRH1/CPCLRL1, CPCLRH2/CPCLRL2) と一致すると、タイマ状態レジスタ (TCCSH0, TCCSH1, TCCSH2) の ICLR : bit9 に "1" が設定されます。この状態において割込み要求が許可 (TCCSH0, TCCSH1, TCCSH2 レジスタの ICRE : bit8=1) になると、割込み要求が割込みコントローラへ出力されます。

タイマ値が "0000_H" になると、タイマ状態制御レジスタ (TCCSH0, TCCSH1, TCCSH2) の IRQZF : bit14 に "1" が設定されます。この状態において割込み要求が許可 (TCCSH0, TCCSH1, TCCSH2 レジスタの IRQZE : bit13=1) になると、割込み要求が割込みコントローラへ出力されます。

■ 16 ビットアウトプットコンペア割込み

16 ビットアウトプットコンペアの割込み制御ビットと割込み要因を表 12.5-2 に示します。

表 12.5-2 16 ビットアウトプットコンペア 0 ~ 5 の割込み制御ビットと割込み要因

	16 ビットアウトプット コンペア 0, 1	16 ビットアウトプット コンペア 2, 3	16 ビットアウトプット コンペア 4, 5
割込み要求 フラグビット	コンペア制御レジスタ下位 (OCSL0) の IOP1, IOP0 (bit7, bit6)	コンペア制御レジスタ下位 (OCSL2) の IOP1, IOP0 (bit7, bit6)	コンペア制御レジスタ下位 (OCSL4) の IOP1, IOP0 (bit7, bit6)
割込み要求許可 ビット	コンペア制御レジスタ下位 (OCSL0) の IOE1, IOE0 (bit5, bit4)	コンペア制御レジスタ下位 (OCSL2) の IOE1, IOE0 (bit5, bit4)	コンペア制御レジスタ下位 (OCSL4) の IOE1, IOE0 (bit5, bit4)
割込み要因	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH0, OCCPH1, OCCPL0, OCCPL1) と一致する	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH2, OCCPH3, OCCPL2, OCCPL3) と一致する	16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH4, OCCPH5, OCCPL4, OCCPL5) と一致する

16 ビットフリーランタイム値がアウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5, OCCPL0 ~ OCCPL5) と一致すると、コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の IOP1, IOP0 : bit7, bit6 が "1" に設定されます。この状態において割込み要求が許可 (OCSL0, OCSL2, OCSL4 レジスタの IOE1, IOE0 : bit5, bit4=11_B) になると、割込み要求が割込みコントローラへ出力されます。

■ 16 ビットインプットキャプチャ割込み

16 ビットインプットキャプチャの割込み制御ビットと割込み要因を表 12.5-3 に示します。

表 12.5-3 16 ビットインプットキャプチャ 0 ~ 3 の割込み制御ビットと割込み要因

	16 ビットインプットキャプチャ 0, 1	16 ビットインプットキャプチャ 2, 3
割込み要求 フラグビット	インプットキャプチャ状態制御 レジスタ下位 (PICSL01) の ICP1, ICP0 (bit7, bit6)	インプットキャプチャ状態制御 レジスタ下位 (ICSL23) の ICP3, ICP2 (bit7, bit6)
割込み要求許可ビット	インプットキャプチャ状態制御 レジスタ下位 (PICSL01) の ICE1, ICE0 (bit5, bit4)	インプットキャプチャ状態制御 レジスタ下位 (ICSL23) の ICP3, ICP2 (bit5, bit4)
割込み要因	有効なエッジが IC0, IC1 端子で検出される	有効なエッジが IC2, IC3 端子で検出される

16 ビットインプットキャプチャでは、有効なエッジが IC0, IC1, IC2, IC3 端子で検出されると、インプットキャプチャ状態制御レジスタ (PICSL01, ICSL23) の ICP3 ~ ICP0 : bit7, bit6 に "11_B" が設定されます。この状態において割込み要求が許可 (PICSL01, ICSL23 レジスタの ICE3 ~ ICE0 : bit5, bit4=11_B) になると、割込み要求は割込みコントローラへ出力されます。

■ 波形ジェネレータ割込み

波形ジェネレータの割込み制御ビットと割込み要因を表 12.5-4 に示します。

表 12.5-4 波形ジェネレータの割込み制御ビットと割込み要因

	波形ジェネレータ	
	16 ビットデッドタイム 0, 1, 2	DTTI 端子
割込み要求フラグビット	16 ビットデッドタイム制御レジスタ 上位, 下位 (DTCR0 ~ DTCR2) の TMIF0 ~ TMIF2 (上位は bit12, 下位は bit4)	波形制御レジスタ 1 (SIGCR1) の DTIF (bit6)
割込み要求許可ビット	16 ビットデッドタイム制御レジスタ 上位, 下位 (DTCR0 ~ DTCR2) の TMIE0 ~ TMIE2 (上位は bit11, 下位は bit3)	波形制御レジスタ 1 (SIGCR1) の DTIE(bit7)
割込み要因	16 ビットデッドタイム 0, 1, 2 アンダ フロー	DTTI 端子で "L" レベルが検出さ れる

波形ジェネレータでは, 16ビットデッドタイムのアンダフローが発生し, かつDTCR0 ~ DTCR2 レジスタの TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "000_B" または "001_B" のとき, 16 ビットデッドタイム制御レジスタ (DTCR0 ~ DTCR2) の TMIF0 ~ TMIF2 (上位は bit12, 下位は bit4) には "1" が設定されます。この状態において割込み要求が許可 (DTCR0 ~ DTCR2, レジスタの TMIE0 ~ TMIE2 (上位は bit11, 下位は bit3) =1) になると, 割込み要求は割込みコントローラへ出力されます。

12.6 多機能タイマの動作

多機能タイマの動作について説明します。

■ 多機能タイマの動作

● 16 ビットフリーランタイム

16 ビットフリーランタイムは、カウント動作を許可すると、タイマデータレジスタ (TCDTH, TCDTL) に設定されている値からカウントアップを開始します。カウント値は、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して使用されます。

● 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は、"指定されたアウトプットコンペアレジスタに設定されている値" と "16 ビットフリーランタイム値" の比較に使用します。一致が検出された場合は、割込みフラグが設定され、出力レベルは反転します。

● 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は、指定された有効なエッジを検出するために使用します。

有効なエッジが検出されると、割込みフラグが設定され、16 ビットフリーランタイム値が取り出され、インプットキャプチャデータレジスタへ格納されます

● 波形ジェネレータ

波形ジェネレータは、リアルタイム出力 (RTO0 ~ RTO5)、16 ビット PPG タイマ 0、16 ビットデッドタイムを使用してさまざまな波形 (デッドタイムを含む) を生成します。

● A/D 起動コンペア

16 ビットフリーランタイム 0 の値が指示値となったときに、A/D 起動を行います。

12.6.1 16 ビットフリーランタイマの動作

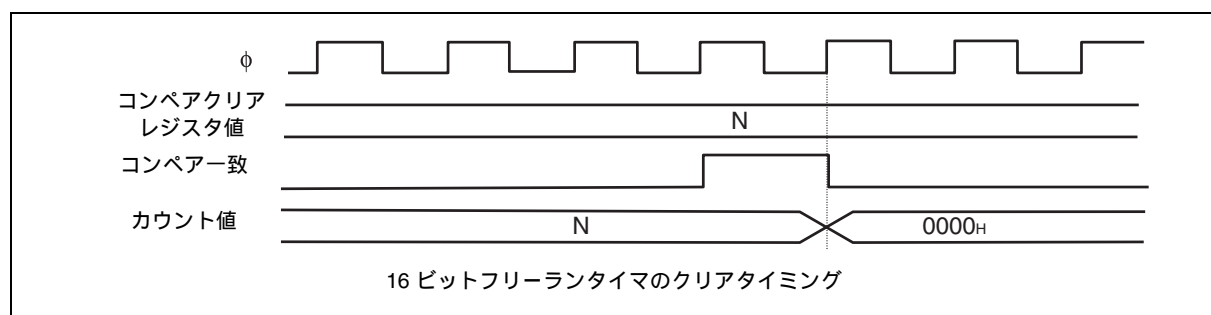
16 ビットフリーランタイマは、3 ユニット搭載しており、リセット完了後、タイマデータレジスタ (TCDTH0 ~ TCDTH2 / TCDTL0 ~ TCDTL2) に設定されている値からカウントアップを開始します。カウント値は、16 ビットアウトプットコンペアと 16 ビットインプットキャプチャの基準時間として使用されます。

■ タイマクリア

16 ビットフリーランタイマのカウント値は、下記のいずれかの場合にクリアされます。

- アップカウントモード (TCCSL レジスタの MODE : bit5=0) によってコンペアクリアレジスタとの一致が検出された場合
- 動作中に TCCSL レジスタの SCLR : bit4 に "1" が書き込まれた場合
- 停止中に TCDTH/TCDTL レジスタに "0000_H" が書き込まれた場合
- リセットされた場合

リセットされると、カウンタは直ちにクリアされます。ソフトウェアクリアされた場合、またはコンペアクリアレジスタとの一致が発生した場合は、カウンタはカウントタイミングと同期してクリアされます。



■ タイマモード

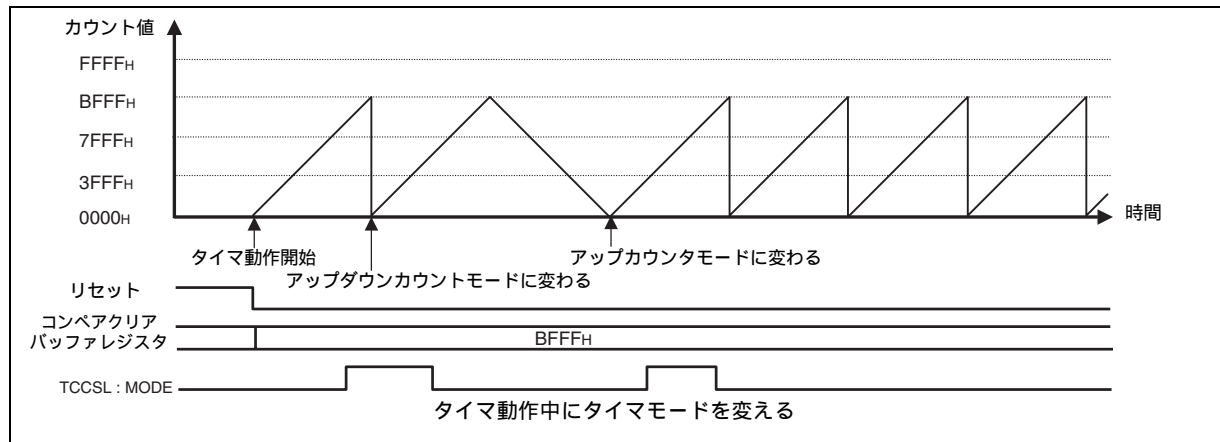
16 ビットフリーランタイマでは、以下のどちらかのモードを選択することができます。

- アップカウントモード (TCCSL レジスタの MODE : bit5=0)
- アップダウンカウントモード (TCCSL レジスタの MODE : bit5=1)

アップカウントモード時は、カウンタは事前に設定されているタイマデータレジスタ (TCDTH/TCDTL) からカウントを開始し、カウント値がコンペアクリアレジスタ (CPCLR_H/CPCLR_L) の値と一致するまでカウントアップし、次にカウンタは、"0000_H" にクリアされ、次に再びカウントアップします。

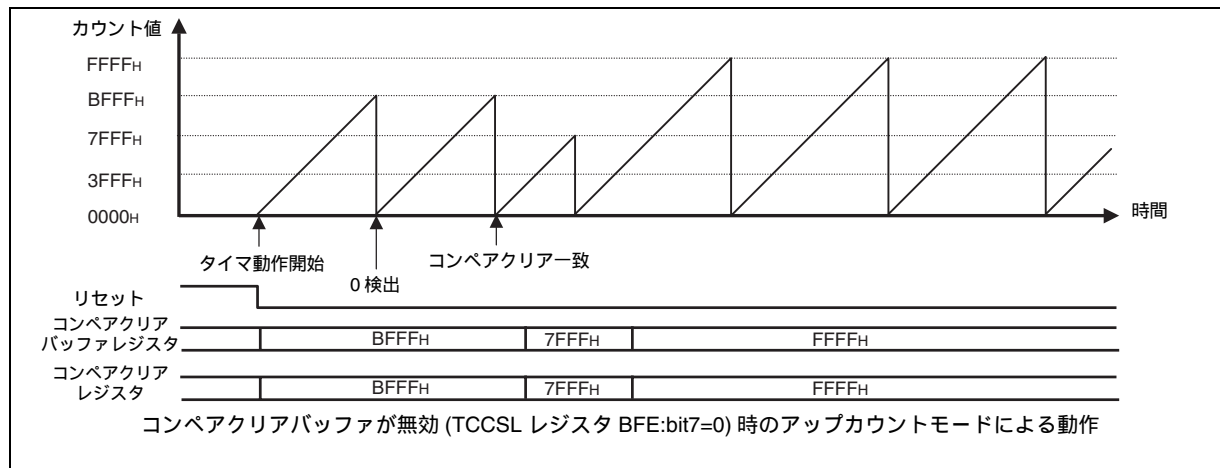
アップダウンカウントモード時は、カウンタは事前に設定されているタイマデータレジスタ (TCDTH/TCDTL) からカウントを開始し、カウント値がコンペアクリアレジスタ (CPCLR_H/CPCLR_L) の値と一致するまでカウントアップし、次にカウントがアップカウントからダウンカウントに変わり、カウンタ値が "0000_H" に達するまでカウントダウンし、次に再びカウントアップします。

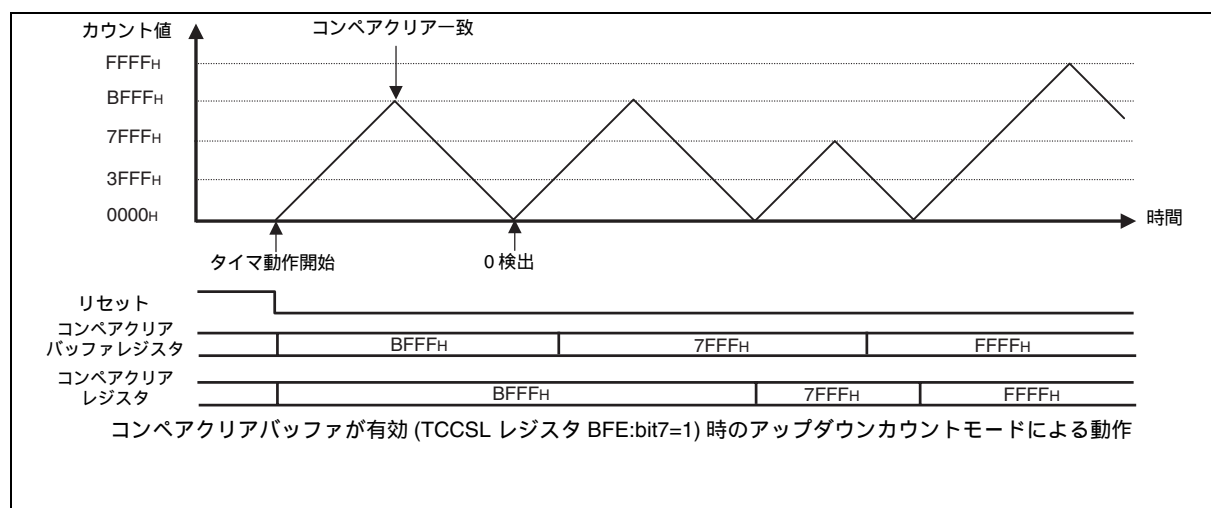
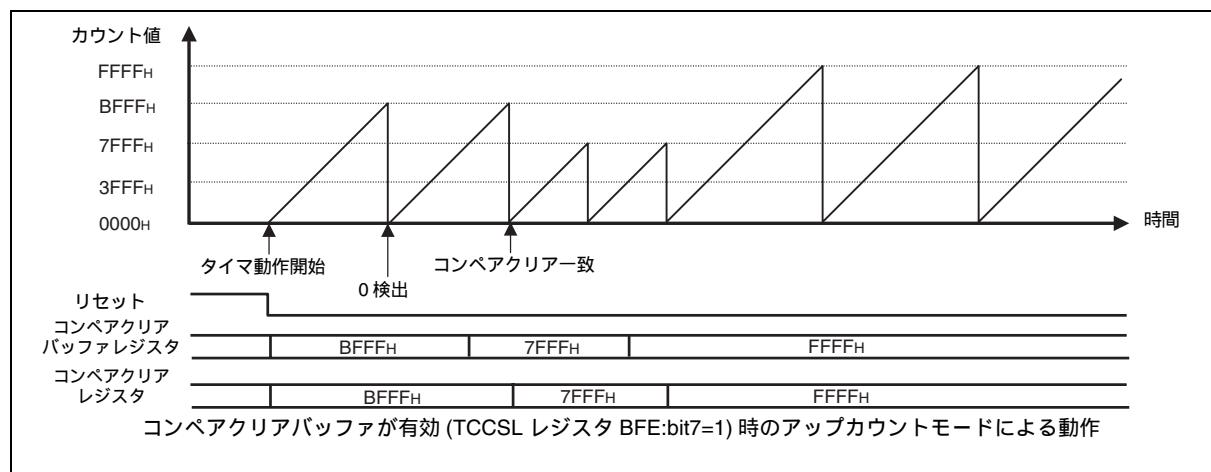
モードビット (TCCSLレジスタのMODE : bit5) には、タイマが動作中であろうと停止していようと、いつでも値を書き込むことができます。タイマ動作中にこのビットに書き込まれた値はバッファに入れられ、カウントモードはタイマ値が"0000_H"になると変わります。



■ コンペアクリアバッファ

コンペアクリアレジスタ (CPCLR_H/CPCLR_L) には、有効または無効にすることができるバッファ機能が存在します。バッファ機能が有効 (TCCSL レジスタの BFE : bit7=1) の場合は、コンペアクリアバッファレジスタ (CPCLR_{BH}/CPCLR_{BL}) に書き込まれたデータは、16 ビットフリーランタイム値 "0" が検出されると CPCLR_H/CPCLR_L レジスタに転送されます。バッファ機能が無効 (TCCSL ビットの BFE : bit7=0) の場合は、CPCLR_{BH}/CPCLR_{BL} レジスタは透過であり、データは CPCLR_H/CPCLR_L レジスタに直接書き込むことができます。





■ タイマ割込み

16 ビットフリーランタイマでは、以下の 2 つの割込みが発生します。

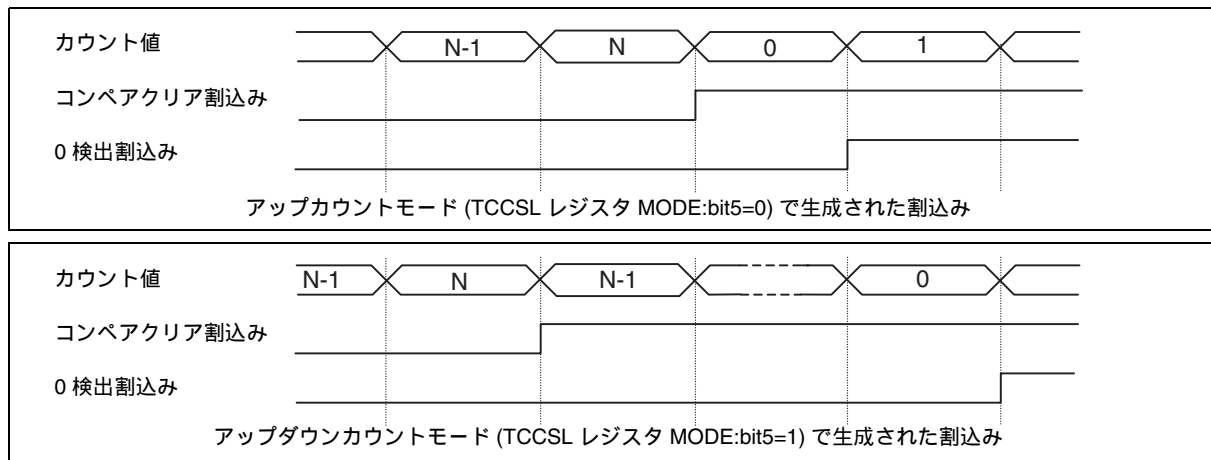
- コンペアクリア割込み
- 0 検出割込み

コンペアクリア割込みは、タイマ値がコンペアクリアレジスタ (COCLRH/COCLRL) の値と一致すると発生します。

0 検出割込みは、タイマ値が "0000_H" に達すると発生します。

< 注意事項 >

ソフトウェアクリア (TCCSL レジスタの SCLR:bit4=1) は, 0 検出割込みを発生しません。



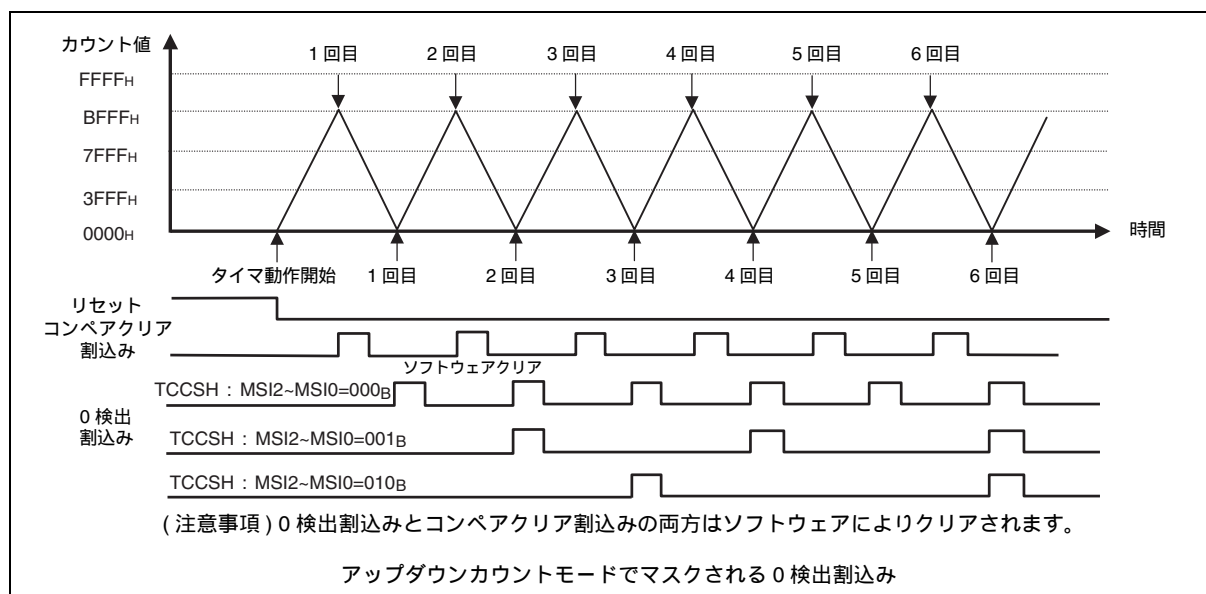
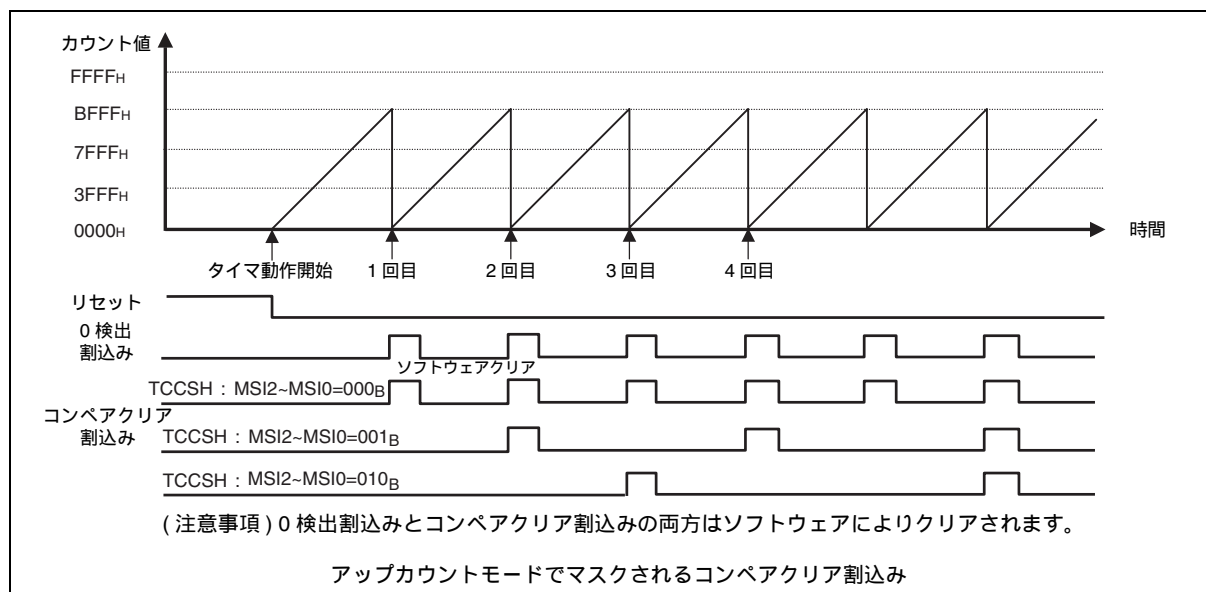
■ 割込みマスク機能

TCCSH レジスタの MSI2 ~ MSI0: bit12 ~ bit10 を設定すると, 割込み要求をマスクすることができます。MSI2 ~ MSI0 ビットは, カウント値が "000_B" に達すると値をリロードする 3 ビットリロードダウンカウンタです。カウント値は, MSI2 ~ MSI0 ビットに直接書き込むことによってもロードすることができます。マスクカウントは, MSI2 ~ MSI0 ビットに設定された値です。MSI2 ~ MSI0 ビットが "000_B" になると, 割込み要因はマスクされません。

割込み要因は, カウントモード (TCCSL レジスタの MODE: bit5) によって異なります。アップカウントモード時は, コンペアクリア割込みのみをマスクすることができ, 0 検出割込みは "0" が検出されるたびに発生します。アップダウンカウントモード時は, 0 検出割込みのみをマスクすることができ, コンペアクリア割込みはコンペアクリアが検出されるたびに発生します。

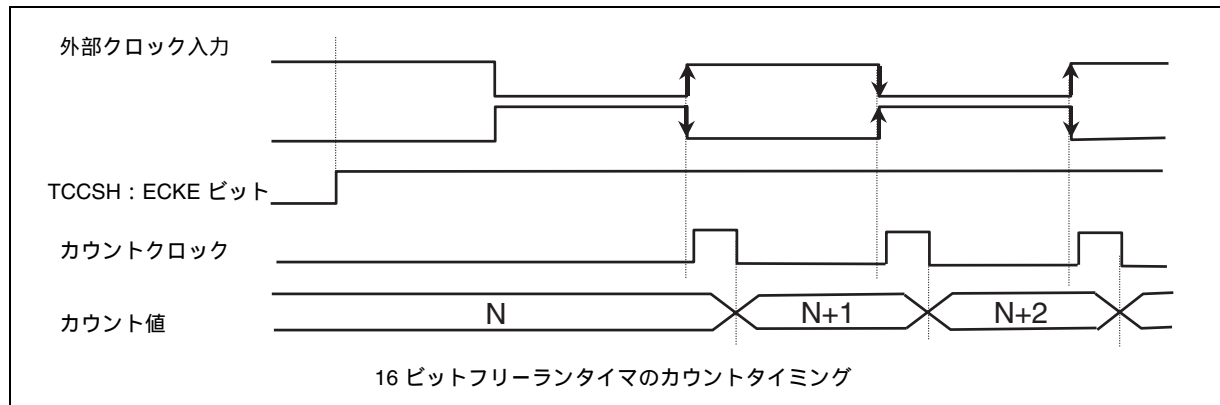
< 注意事項 >

ソフトウェアクリア (TCCSL レジスタの SCLR : bit4=1) は、0 検出割込みを発生しません。



■ 選択された外部カウントクロック

16 ビットフリーランタイムは、入力クロック（内部クロックまたは外部クロック）に基づいてインクリメントされます。外部クロックが選択されると、外部クロックモード (TCCSH レジスタの ECKE : bit15=1) が選択された後、16 ビットフリーランタイムは、外部入力の初期値が "1" の場合、立上りエッジでカウントアップを開始します。その後は両エッジでカウントアップします。外部入力の初期値が "0" の場合、立下りエッジでカウントアップを開始します。その後は両エッジでカウントアップします。



< 注意事項 >

外部クロック入力のカウントは、外部クロックの両エッジとなります。

■ フリーランタイム 0 による A/D 起動

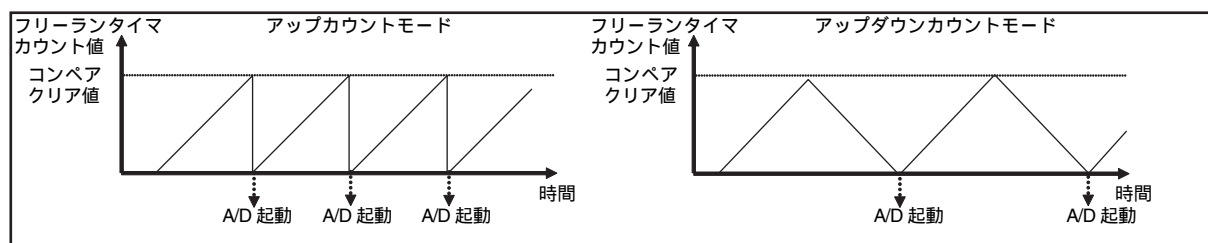
16 ビットフリーランタイム 0 のコンペアー一致時, もしくは 0 検出時に A/D1 および A/D2 の起動が可能です。起動要因は, A/D トリガ制御レジスタ (ADTRGC) の A/D トリガ要因選択ビット (SEL1, SEL2 : bit2, bit3) で選択することが可能です。

A/D 起動信号は, A/D トリガ制御レジスタ (ADTRGC) の A/D トリガ出力許可/禁止ビット (AD1E, AD2E : bit0, bit1) により, コンペアー一致時, もしくは 0 検出時でも, A/D 起動信号を止めることが可能です。

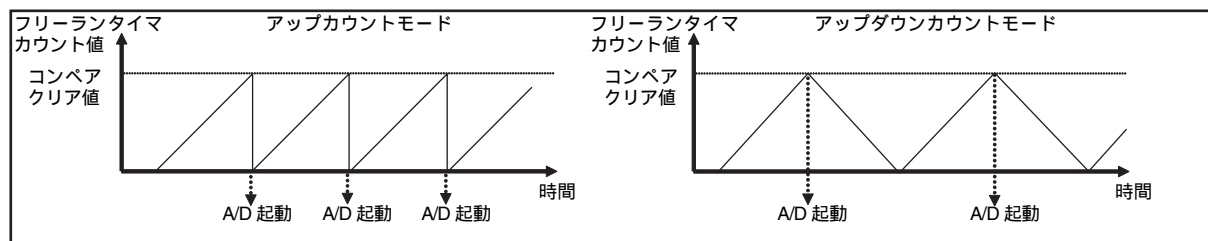
< 注意事項 >

A/D 起動信号の出力を禁止しているときに, 起動要因のコンペアー一致, もしくは 0 検出が出力されている場合, A/D 起動信号の出力を許可すると, 許可と同時に A/D 起動信号が出力されます。

● 0 検出による A/D 起動 (ADTRGC : SELn=0 [n=1, 2 : A/D ユニット番号])



● コンペアークリア一致による A/D 起動 (ADTRGC : SELn=1 [n=1, 2 : A/D ユニット番号])



12.6.2 16 ビットアウトプットコンペアの動作

アウトプットコンペアは," 指定されたコンペアクリアレジスタに設定されている値 " と "16 ビットフリーランタイムの値 " の比較に使用します。一致が検出された場合は, 割込みフラグが設定され, 出力レベルが反転します。
フリーランタイムがアップダウンカウントモードの場合, カウントピークとコンペアレジスタ値が一致したときは, 一致信号を無視します。

■ 16 ビットアウトプットコンペア用フリーランタイムの選択

- 16 ビットアウトプットコンペア各チャンネルに対して, 3 ユニットあるフリーランタイムのいずれかを設定できます。
- 以下の説明中のフリーランタイムはここで選択されたフリーランタイムについてです。
- この設定は, アウトプットコンペア動作中に書き換えないでください。

FSR0 ~ FSR2 レジスタ		選択されるフリーランタイム
OCUn1	OCUn0	
0	0	フリーランタイム ch.0 選択 [初期値]
0	1	フリーランタイム ch.1 選択
1	0	フリーランタイム ch.2 選択
1	1	設定禁止

(n=0, 1, 2, 3, 4, 5 : アウトプットコンペアのチャンネル)

■ 16 ビットアウトプットコンペアの動作 (反転モード, MOD1x=0)

- コンペア動作は, 各チャネル (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の CMOD : bit12=0) において実行することができます。

図 12.6-1 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を別々に使用した際の出力波形例 (フリーランタイムはアップカウントモード)

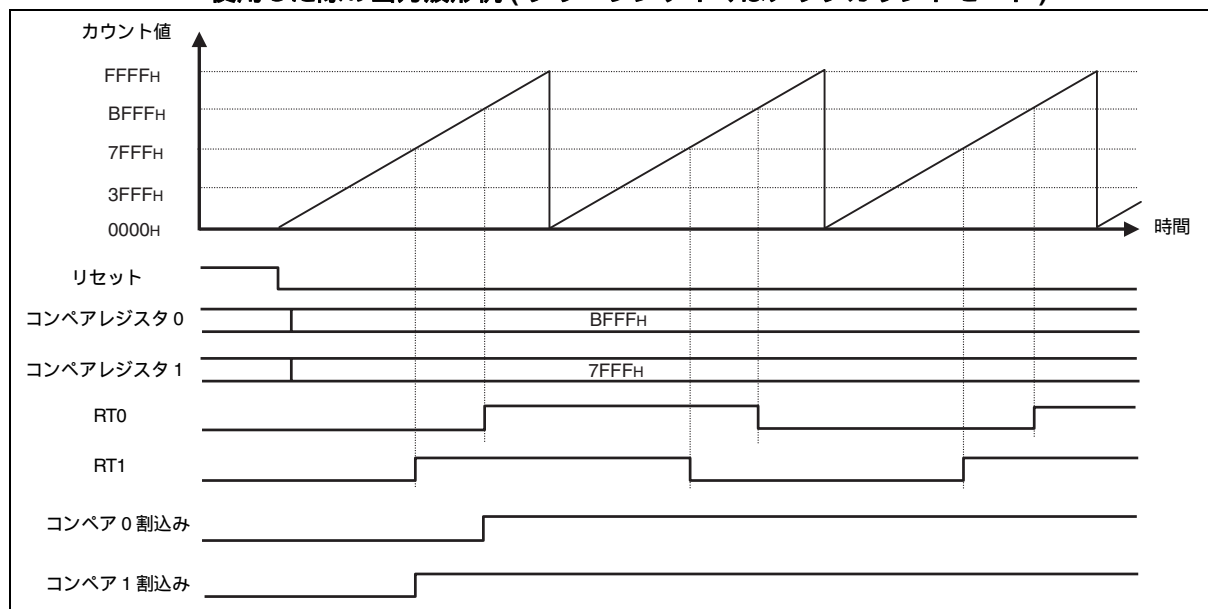
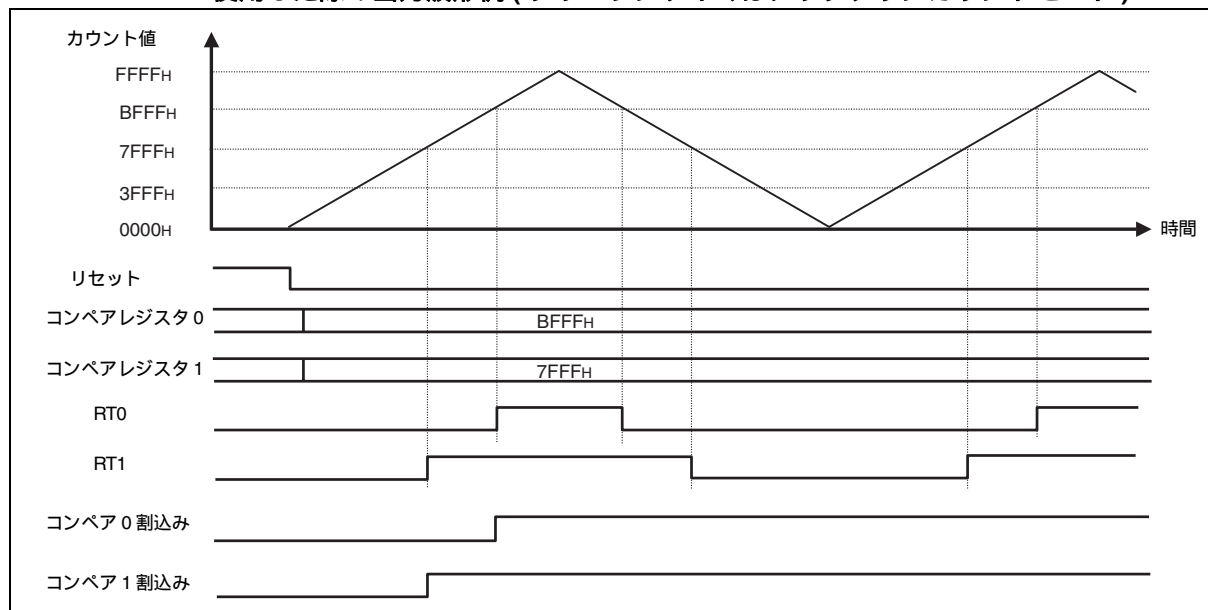


図 12.6-2 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を別々に使用した際の出力波形例 (フリーランタイムはアップダウンカウントモード)



- 出力レベルは、一対のコンペアレジスタ (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の CMOD : bit12=1) を使用して変更することができます。

図 12.6-3 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を一対で使用した際の出力波形例 (フリーランタイムはアップカウントモード)

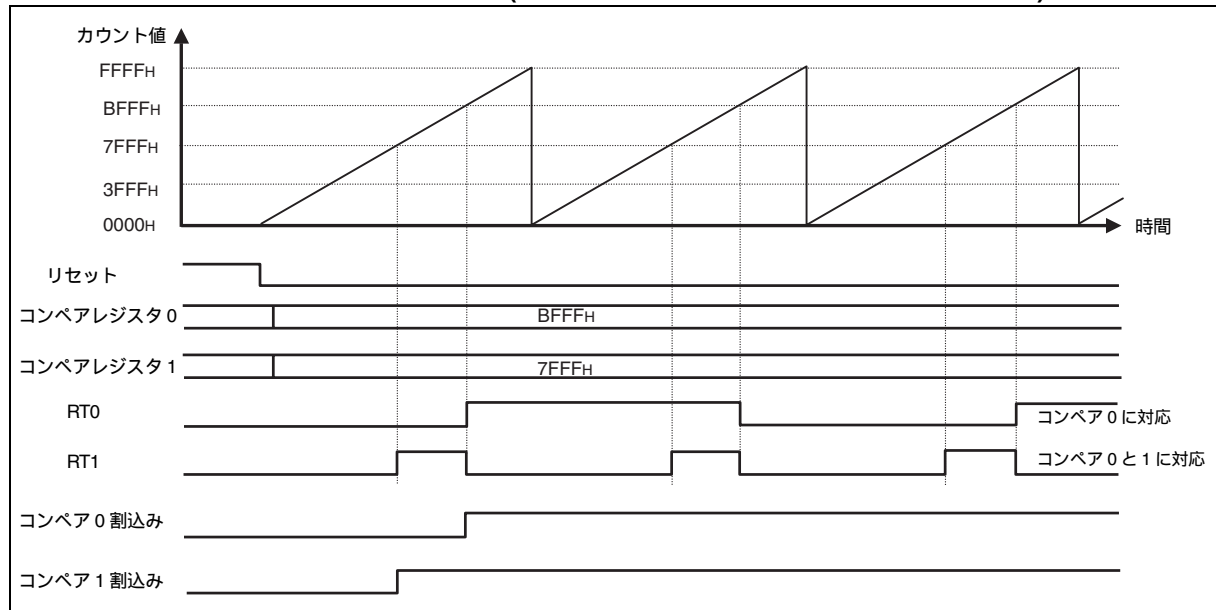
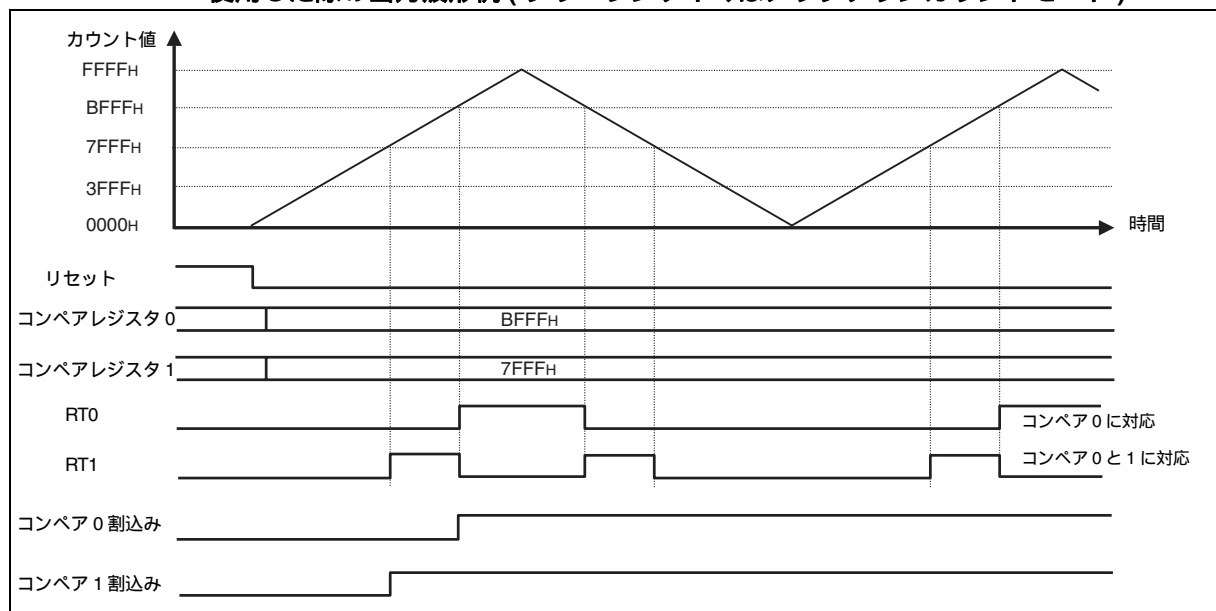
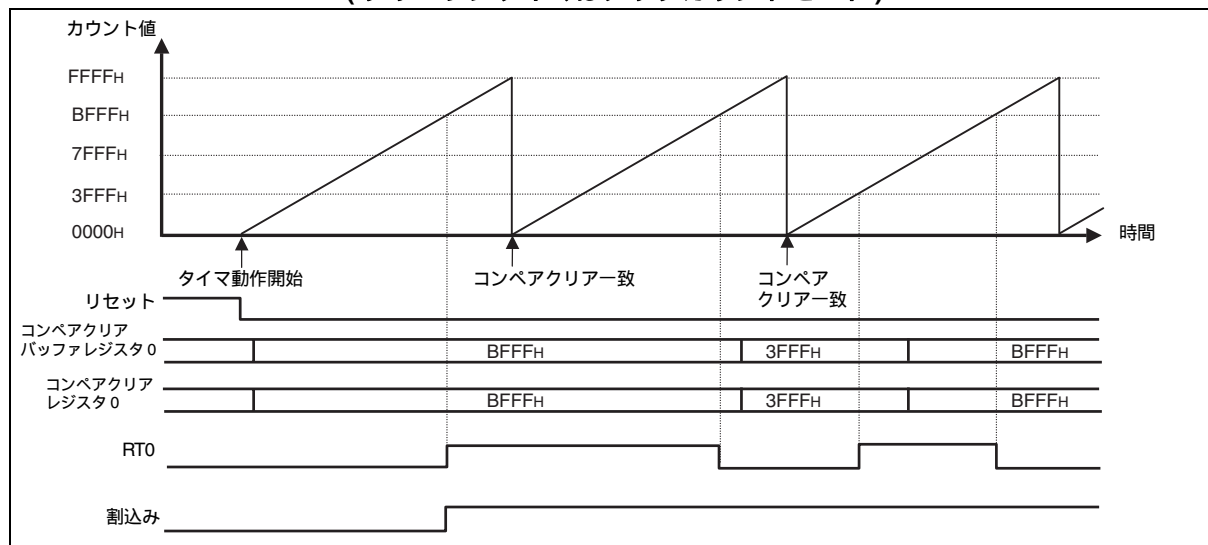


図 12.6-4 出力初期値が "0" のときにコンペアレジスタ 0 とコンペアレジスタ 1 を一緒に使用した際の出力波形例 (フリーランタイムはアップダウンカウントモード)



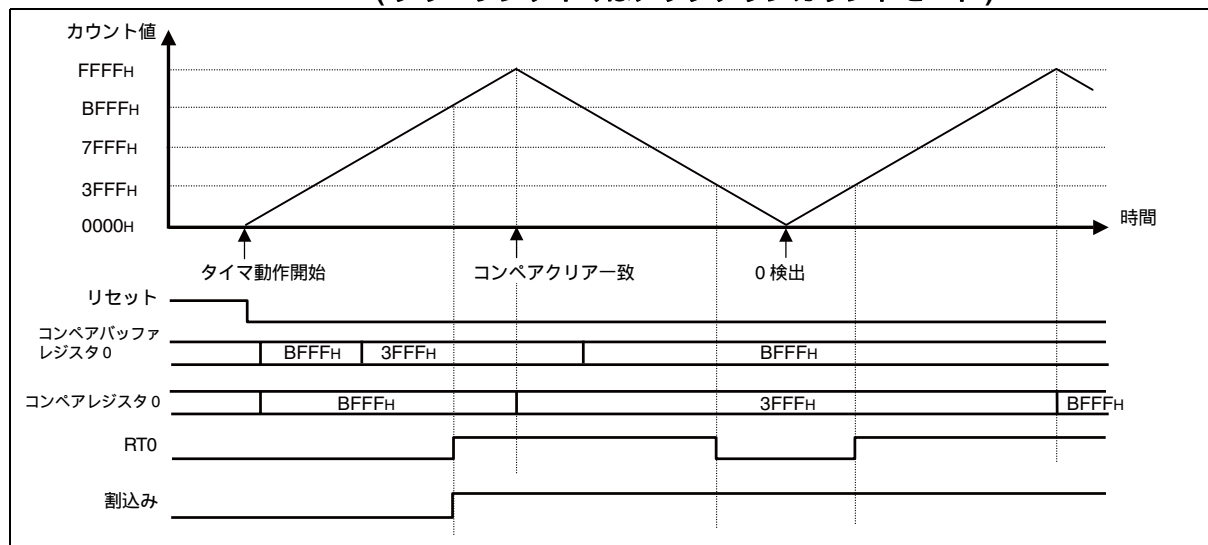
● コンペアバッファが無効になったときの出力レベル

図 12.6-5 コンペアバッファが無効になっているときの出力波形例
(フリーランタイムはアップカウントモード)

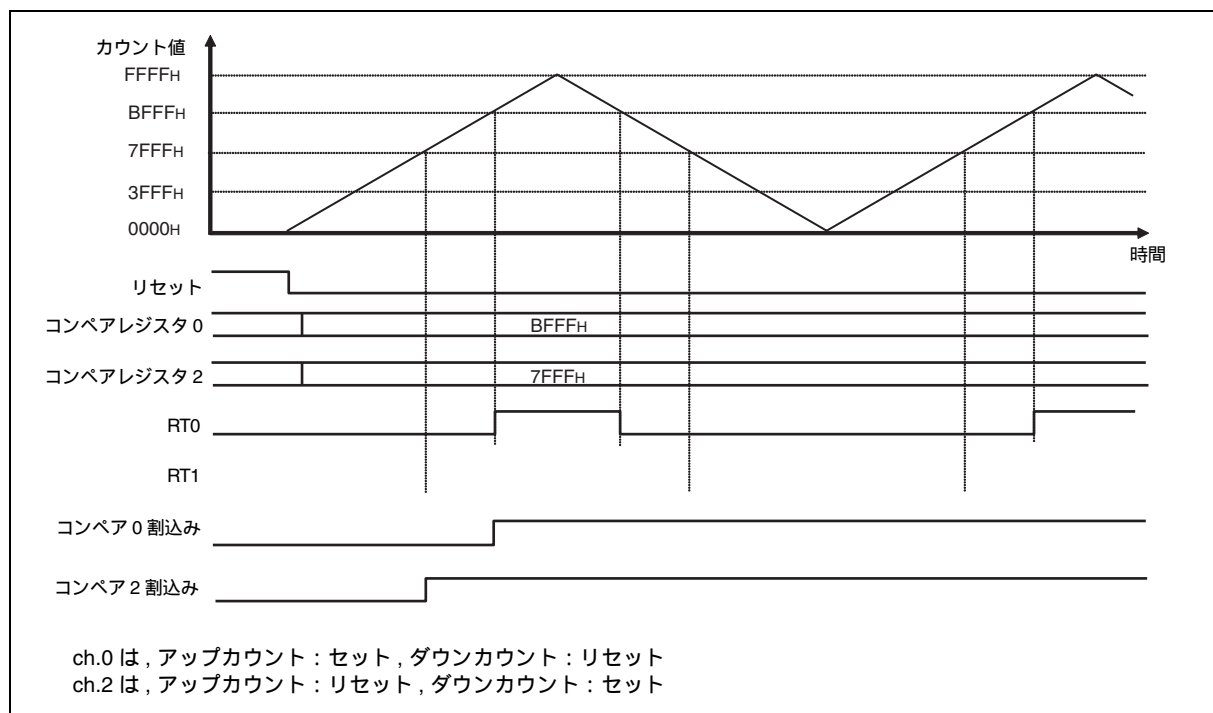
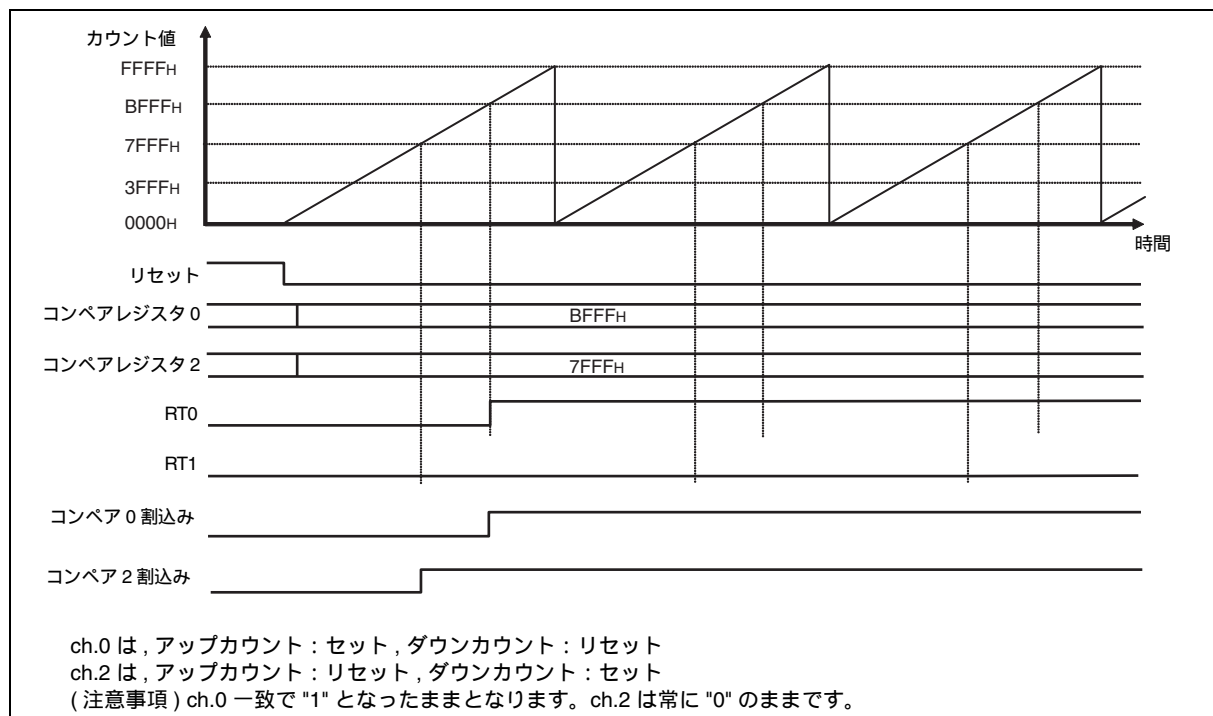


● コンペアクリア一致発生時にコンペアバッファが選択されたときの出力レベル

図 12.6-6 コンペアバッファが有効になったときの出力波形例
(フリーランタイムはアップダウンカウントモード)



■ 16 ビットアウトプットコンペアの動作 (セット/リセットモード, MOD1x=1)



■ 16 ビットアウトプットコンペアタイミング

フリーランタイマがコンペアレジスタ値と一致すると、アウトプットコンペアはコンペア一致信号を発生して出力を反転し、割込みを発生します。コンペア一致が発生すると、出力はカウンタのカウントタイミングと同期して反転します。

図 12.6-7 コンペアレジスタ割込みタイミング

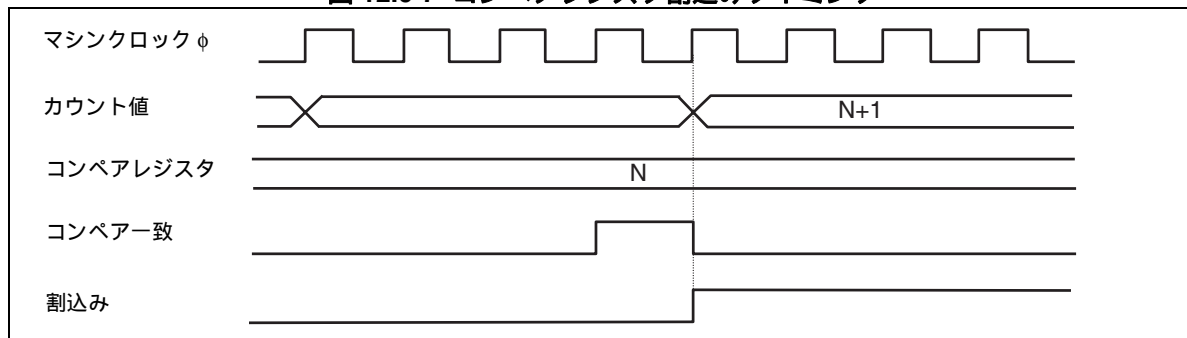
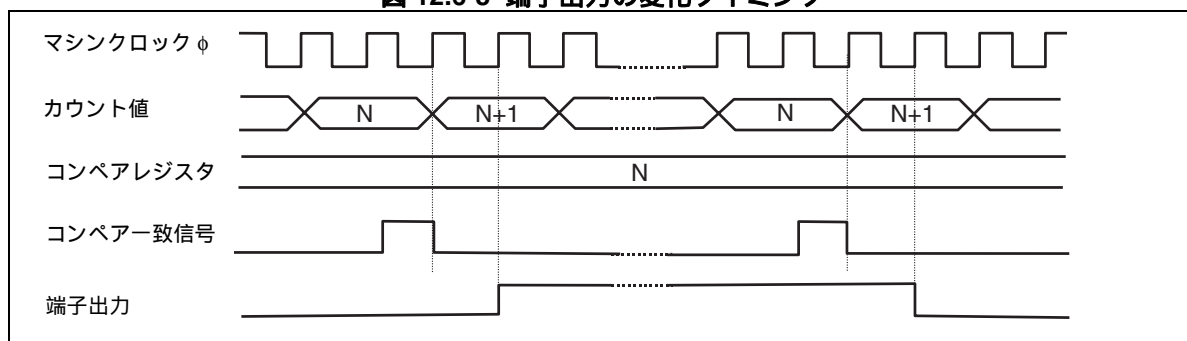
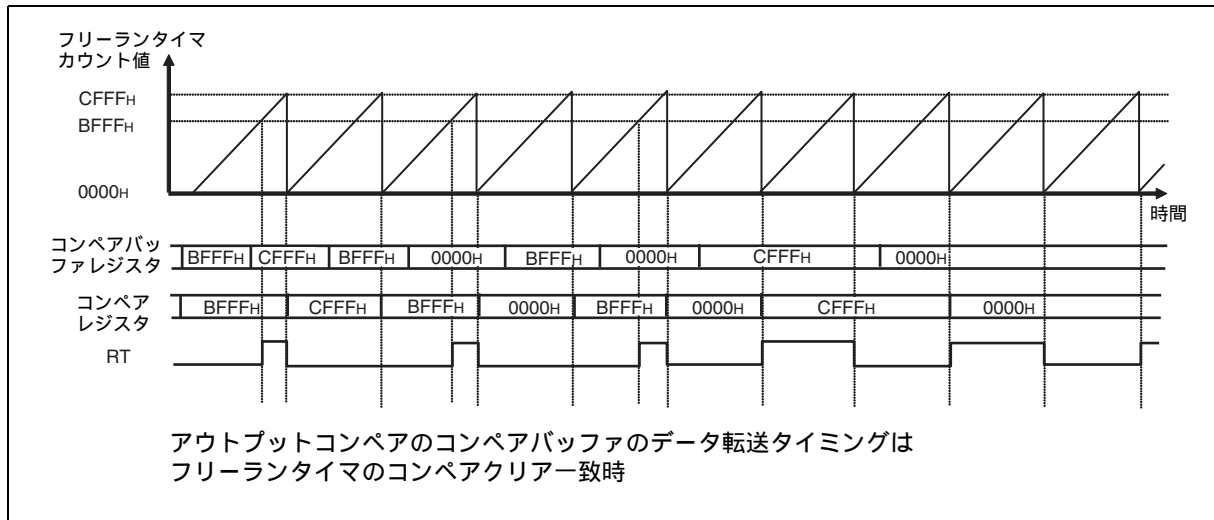


図 12.6-8 端子出力の変化タイミング

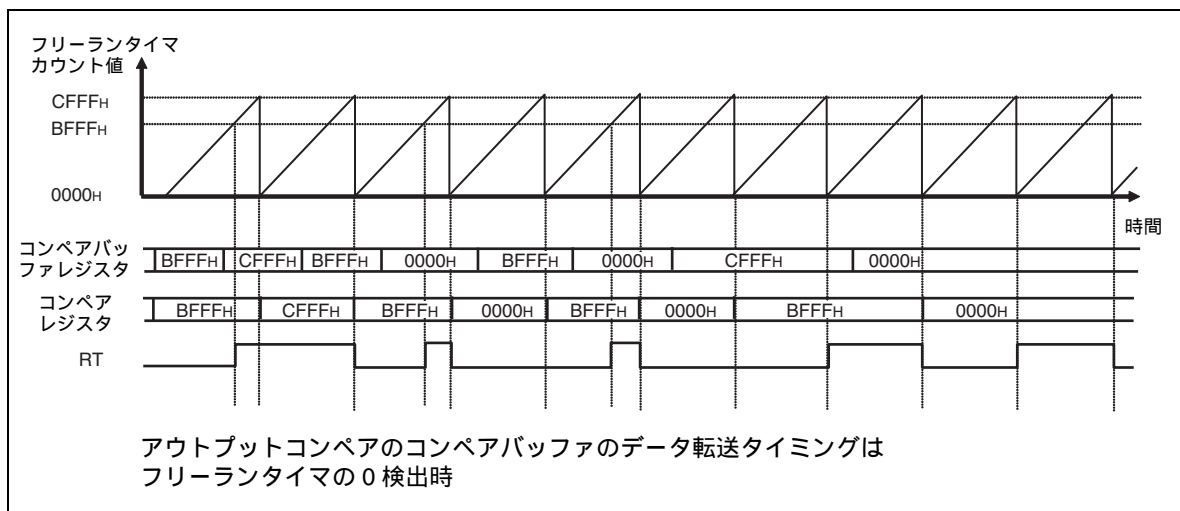


■ 16 ビットアウトプットコンペアとフリーランタイムの動作について

● フリーランタイムがアップカウントの場合

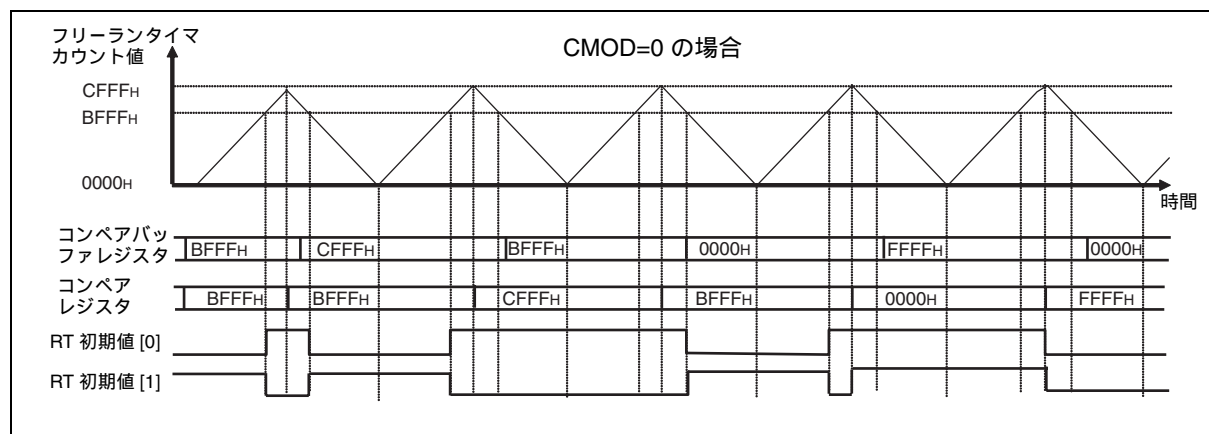


● フリーランタイムがアップカウントの場合



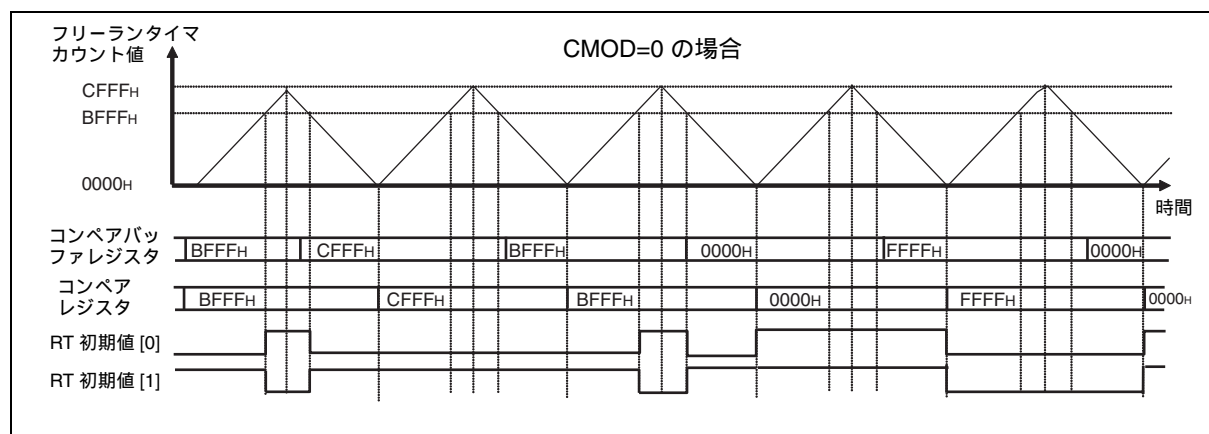
● フリーランタイムがアップダウンカウントの場合

- アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムのコンペアクリアー致時です。
- アウトプットコンペア出力が一致時, 出力反転モードの場合
 - コンペアレジスタ値を "0000_H" に設定したとき, フリーランタイムのカウント値にかかわらず, RT は "1" にセットされます (CMOD = 1 時は "0" にリセット)。
 - コンペアレジスタ値を "FFFF_H" に設定したとき, フリーランタイムのカウント値にかかわらず, RT は "0" にリセットされます (CMOD = 1 時は "1" にセット)。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は, 比較を行いません。このとき, コンペアクリアレジスタ値とコンペアレジスタ値がともに, "FFFF_H" に設定した場合, フリーランタイムのカウント値にかかわらず, RT は "0" にリセットされます。



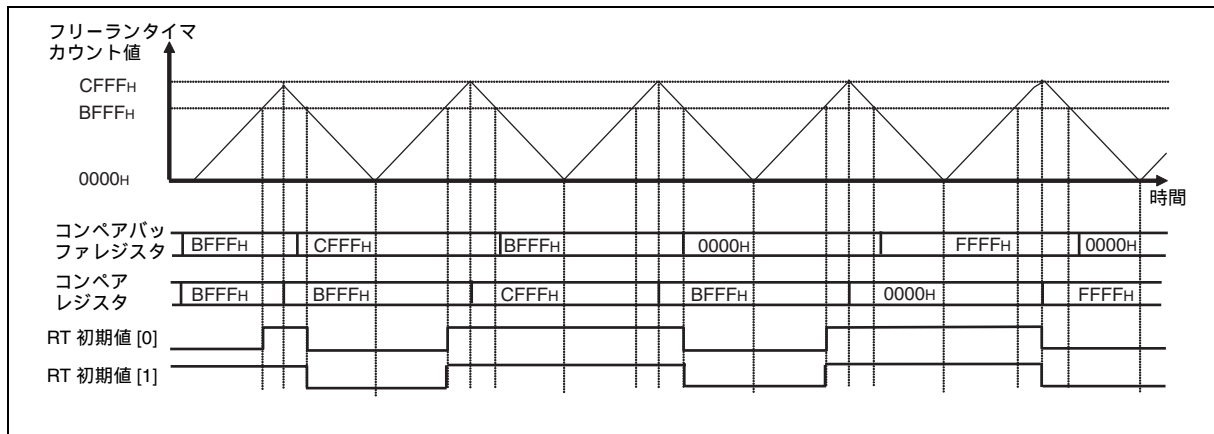
● フリーランタイムがアップダウンカウントの場合

- アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムの 0 検出時です。
- アウトプットコンペア出力は一致時、出力反転モードの場合
 - コンペアレジスタ値を "0000_H" に設定したとき、フリーランタイムのカウント値にかかわらず、RT は "1" にセットされます (CMOD = 1 時は "0" にリセット)。
 - コンペアレジスタ値を "FFFF_H" に設定したとき、フリーランタイムのカウント値にかかわらず、RT は "0" にリセットされます (CMOD = 1 時は "1" にセット)。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は、比較を行いません。このとき、コンペアクリアレジスタ値とコンペアレジスタ値がともに、"FFFF_H" に設定した場合、フリーランタイムのカウント値にかかわらず、RT は "0" にリセットされます。



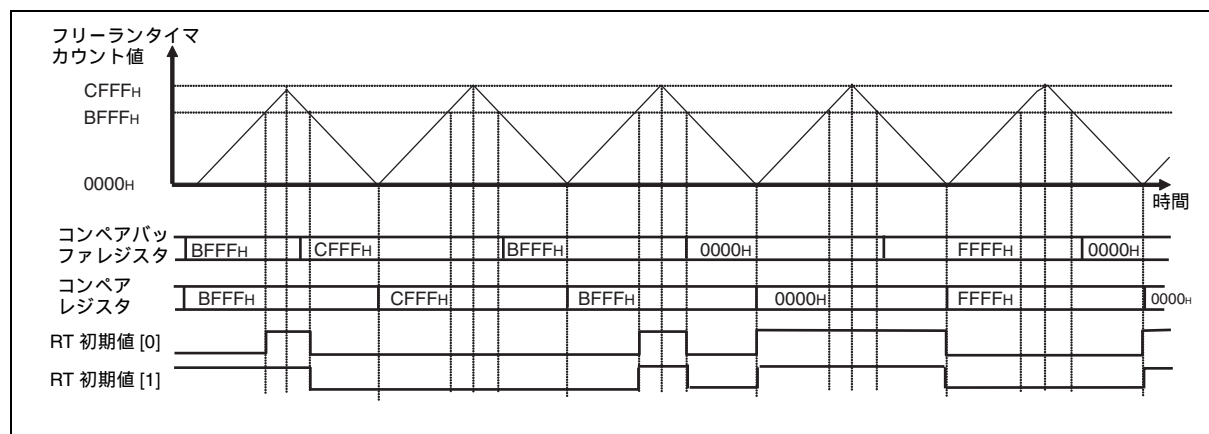
● フリーランタイムがアップダウンカウントの場合

- ・ アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムのコンペアクリア一致時です。
- ・ アウトプットコンペア出力は、アップカウント時の一致は "1" にセット、ダウンカウント時の一致は "0" にリセットする場合 (CMOD=0)
 - コンペアレジスタ値を "0000_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "1" にセットされます。
 - コンペアレジスタ値を "FFFF_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は、比較を行いません。このとき、コンペアクリアレジスタ値とコンペアレジスタ値をともに、"FFFF_H" に設定した場合、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。



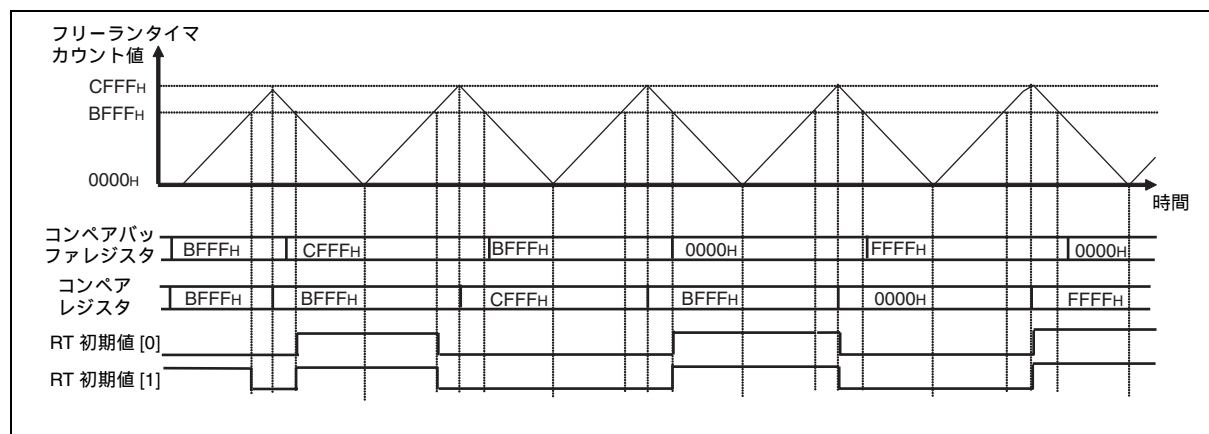
● フリーランタイムがアップダウンカウントの場合

- ・ アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムの 0 検出時です。
- ・ アウトプットコンペア出力は、アップカウント時の一致は "1" にセット、ダウンカウント時の一致は "0" にリセットする場合 (CMOD=0)
 - コンペアレジスタ値を "0000_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "1" にセットされます。
 - コンペアレジスタ値を "FFFF_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は、比較を行いません。このとき、コンペアクリアレジスタ値とコンペアレジスタ値がともに、"FFFF_H" に設定した場合、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。



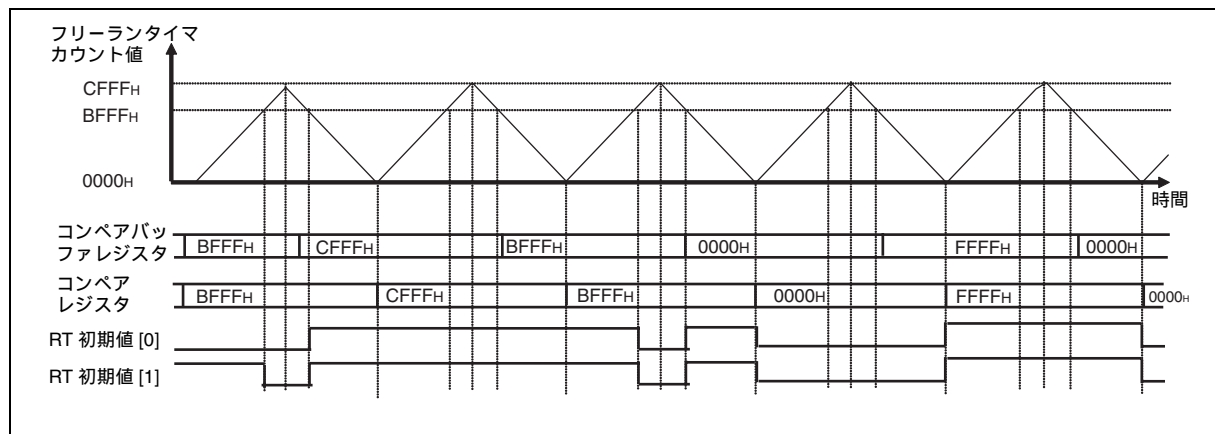
● フリーランタイムがアップダウンカウントの場合

- アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムのコンペアクリア一致時です。
- アウトプットコンペア出力は、アップカウント時の一致は "0" にリセット、ダウンカウント時の一致は "1" にセットする場合 (CMOD=1)
 - コンペアレジスタ値を "0000_H" に設定したとき、フリーランタイムのカウント値にかかわらず、RT は "0" にリセットされます。
 - コンペアレジスタ値を "FFFF_H" に設定したとき、フリーランタイムのカウント値にかかわらず、RT は "1" にセットされます。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は、比較を行いません。このとき、コンペアクリアレジスタ値とコンペアレジスタ値がともに、"FFFF_H" に設定した場合、フリーランタイムのカウント値にかかわらず、RT は "0" にリセットされます。



● フリーランタイムがアップダウンカウントの場合

- アウトプットコンペアのコンペアバッファのデータ転送タイミングはフリーランタイムの 0 検出時です。
- アウトプットコンペア出力は、アップカウント時の一致は "0" にリセット、ダウンカウント時の一致は "1" にセットする場合 (CMOD=1)
 - コンペアレジスタ値を "0000_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。
 - コンペアレジスタ値を "FFFF_H" に設定したとき、フリーランタイムのカウンタ値にかかわらず、RT は "1" にセットされます。
 - フリーランタイムのコンペアクリアレジスタ値とアウトプットコンペアのコンペアレジスタの値が同じ場合は、比較を行いません。このとき、コンペアクリアレジスタ値とコンペアレジスタ値がともに、"FFFF_H" に設定した場合、フリーランタイムのカウンタ値にかかわらず、RT は "0" にリセットされます。



12.6.3 16 ビットインプットキャプチャの動作

インプットキャプチャは、指定された有効なエッジを検出するために使用します。有効なエッジが検出されると、割込みフラグが設定され、16 ビットフリーランタイムの値がキャプチャレジスタへロードされます。

■ 16 ビットインプットキャプチャ用フリーランタイムの選択

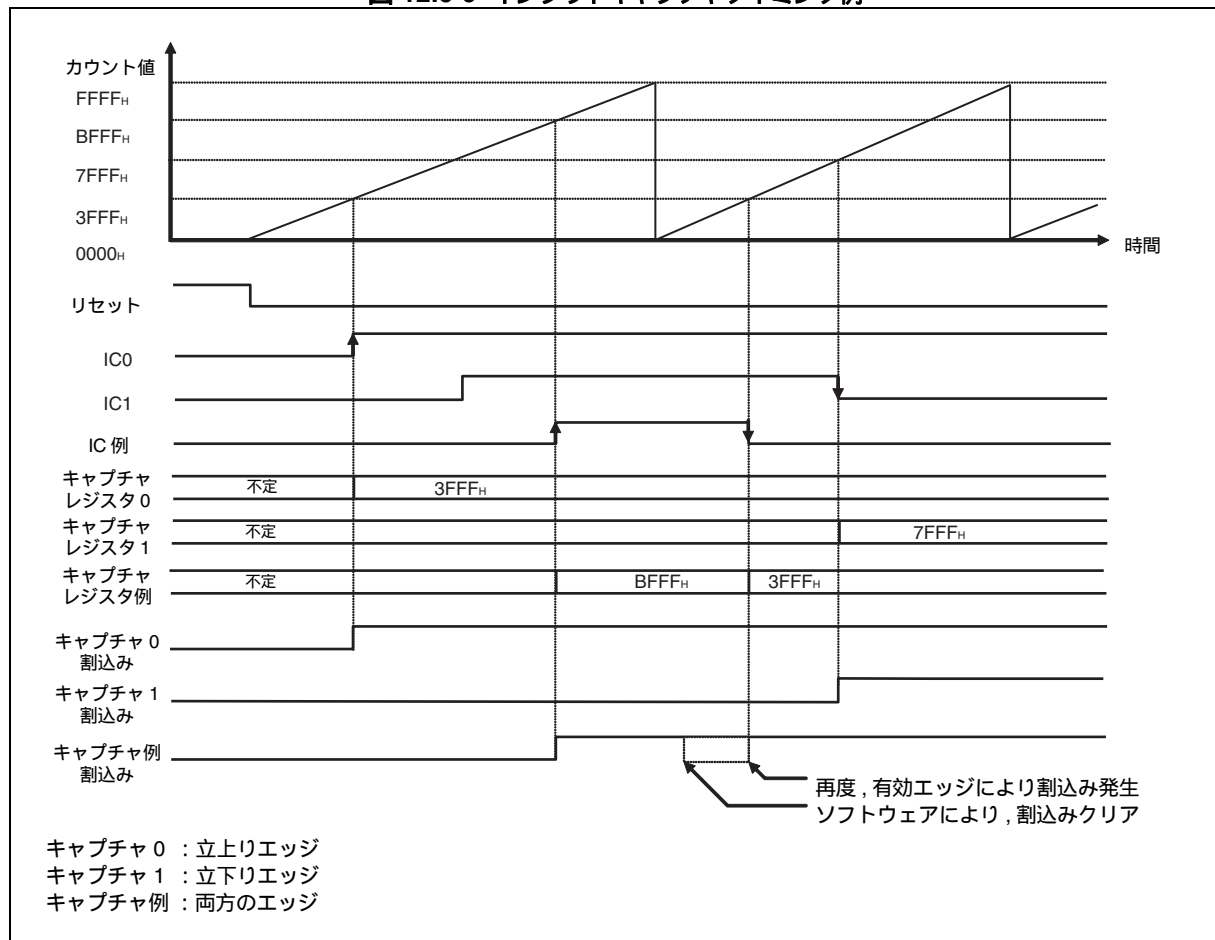
- 16 ビットインプットキャプチャの各チャンネルに対して、3 ユニットあるフリーランタイムのいずれを割り当てるかを設定できます。
- この設定は、インプットキャプチャ動作中に書き換えしないでください。

FSR2 レジスタ		選択されるフリーランタイム
ICUn1	ICUn0	
0	0	フリーランタイム ch.0 選択 [初期値]
0	1	フリーランタイム ch.1 選択
1	0	フリーランタイム ch.2 選択
1	1	設定禁止

(n=0, 1, 2, 3 : インプットキャプチャのチャンネル)

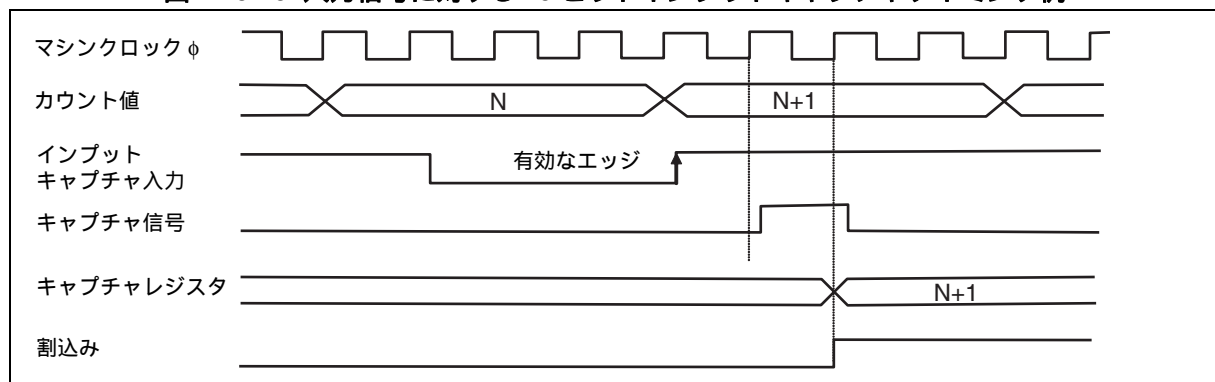
■ 16 ビットインプットキャプチャの動作

図 12.6-9 インプットキャプチャタイミング例



■ 16 ビットインプットキャプチャ入力タイミング

図 12.6-10 入力信号に対する 16 ビットインプットキャプチャタイミング例



12.6.4 波形ジェネレータの動作

波形ジェネレータは、リアルタイム出力 (RTO0 ~ RTO5) , 16 ビット PPG タイマ 0, 16 ビットデッドタイマ 0, 1, 2 を使用してさまざまな波形 (デッドタイムを含む) を生成することができます。

■ RTO0 ~ RTO5 と GATE の出力状態

表 12.6-1 RTO0 ~ RTO5/GATE 出力状態とビット設定 (1 / 2)

TMD2	TMD1	TMD0	GTENx	PGENx	RTOx	GATE
0	0	0	X	X	リアルタイム出力 RTOx (16 ビットアウトプットコンペア出力)	常に "0"
0	0	1	X	0	リアルタイム出力 RTOx (16 ビットアウトプットコンペア出力)	(RTx & GTENx) *3
0	0	1	0	1	RTOx が "H" の期間に PPG0 のパルスを出力 *1	常に "0"
0	0	1	1	1	RTOx が "H" の期間に GATE 信号により起動された PPG0 のパルスを出力	(RTO0/RT1/ RT2/RT3/ RT4/RT5)
0	1	0	X	0	RT0, RT1 の立上りエッジにより 16 ビットデッドタイマ 0 を起動し, 16 ビットデッドタイマ 0 がアンダフローするまで "H" を出力	タイマ動作 期間中は "H" を出力 *4
			X		RT2, RT3 の立上りエッジにより 16 ビットデッドタイマ 1 を起動し, 16 ビットデッドタイマ 1 がアンダフローするまで "H" を出力	
			X		RT4, RT5 の立上りエッジにより 16 ビットデッドタイマ 2 を起動し, 16 ビットデッドタイマ 2 がアンダフローするまで "H" を出力	
0	1	0	0	1	RT0, RT1 の立上りエッジにより 16 ビットデッドタイマ 0 を起動し, 16 ビットデッドタイマ 0 がアンダフローするまで PPG0 のパルスを出力 *1	常に "0"
			0		RT2, RT3 の立上りエッジにより 16 ビットデッドタイマ 1 を起動し, 16 ビットデッドタイマ 1 がアンダフローするまで PPG0 のパルスを出力 *1	
			0		RT4, RT5 の立上りエッジにより 16 ビットデッドタイマ 2 を起動し, 16 ビットデッドタイマ 2 がアンダフローするまで PPG0 のパルスを出力 *1	

表 12.6-1 RTO0 ~ RTO5/GATE 出力状態とビット設定 (2 / 2)

TMD2	TMD1	TMD0	GTEN _x	PGEN _x	RTO _x	GATE
0	1	0	1	1	RT0, RT1 の立上りエッジにより 16 ビットデッドタイム 0 を起動し, 16 ビットデッドタイム 0 がアンダフローするまで, GATE 信号により起動された PPG0 のパルスを出力	タイマ動作期間中は "H" を出力 *4
			1		RT2, RT3 の立上りエッジにより 16 ビットデッドタイム 1 を起動し, 16 ビットデッドタイム 1 がアンダフローするまで, GATE 信号により起動された PPG0 のパルスを出力	
			1		RT4, RT5 の立上りエッジにより 16 ビットデッドタイム 2 を起動し, 16 ビットデッドタイム 2 がアンダフローするまで, GATE 信号により起動された PPG0 のパルスを出力	
1	0	0	X	X	RT1 でノンオーバーラップ信号を生成 *2	常に "0"
			X		RT3 でノンオーバーラップ信号を生成 *2	
			X		RT5 でノンオーバーラップ信号を生成 *2	
1	1	1	0	X	PPG0 でノンオーバーラップ信号を生成	常に "0"
1	1	1	1	X	GATE 信号により起動された PPG0 でノンオーバーラップ信号を生成	(RT0 RT1 RT2 RT3 RT4 RT5)
その他					常に "0"	常に "0"

X=0 ~ 5

*1 : あらかじめ PPG0 を起動しておく必要があります。

*2 : ノンオーバーラップ信号を生成するには, 必ず RT1, RT3, RT5 に対して 2 チャネルモード (コンペア制御レジスタ上位 (OCSH1, OCSH3, OCSH5) の CMOD : bit12=1) を選択してください。

*3 : GTEN_x ビットに "1" を設定した RT_x から GATE 信号が生成されます。*4 : GTEN_x ビットに "1" を設定した RT_x によって起動されるタイマの動作期間中に, GATE 信号が生成されます。複数の GATE_x ビットに "1" を設定した場合, GATE 信号は各々のタイマ動作期間中の信号を OR した信号となります。

< 注意事項 >

RTO0, RTO1 は, 16 ビットデッドタイム制御レジスタ上位 (DTCR0) の TMD2 ~ TMD0 : bit10 ~ bit8 により, RTO2, RTO3 は (DTCR1) 下位レジスタの TMD5 ~ TMD3 : bit2 ~ bit0 により, RTO4, RTO5 は上位レジスタ (DTCR2) の TMD8 ~ TMD6 : bit10 ~ bit8 により制御されます。

■ PPG0 出力制御

RTO0 ~ RTO5 端子への PPG0 出力は、PPG 出力制御 / インプットキャプチャ状態制御レジスタ上位 (PICSH01) の PGEN5 ~ PGEN0 : bit15 ~ bit10 で許可にすることができます。

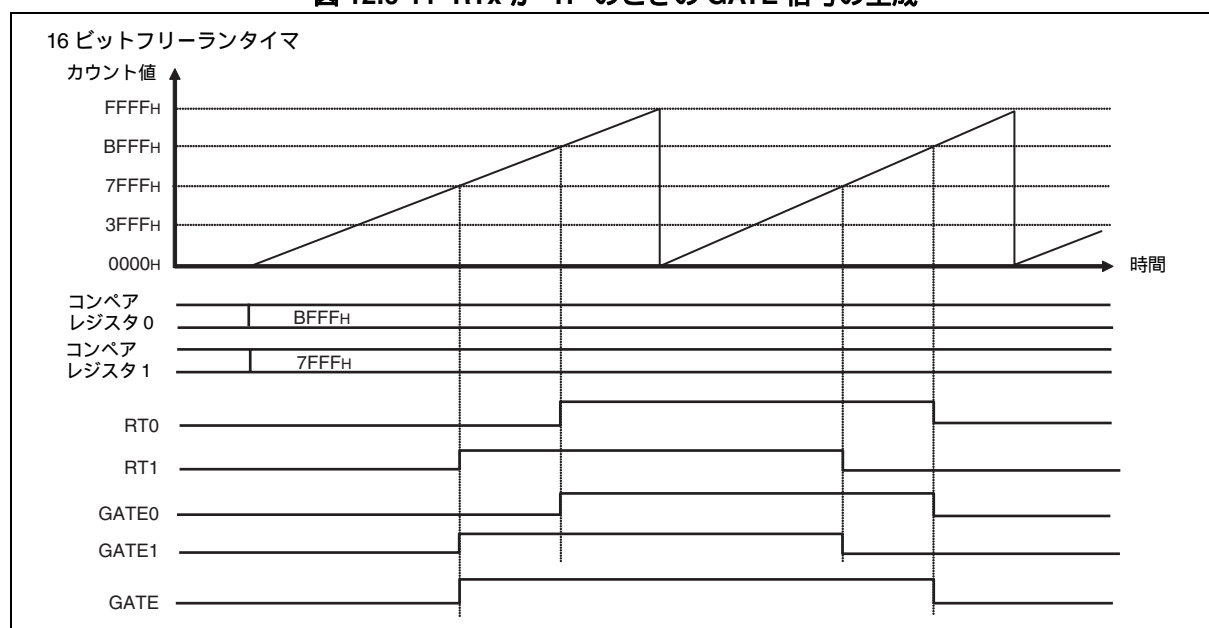
■ ゲートトリガされた PPG0 出力

波形ジェネレータでは、リアルタイム出力 RTO0 ~ RTO5 により、GATE 信号を生成することができます。16 ビットデッドタイム 0, 1, 2 では PPG0 カウントをトリガとして動作することができます。1 つの 16 ビットデッドタイム 0, 1, 2 で 2 つのリアルタイム出力 (RTO0/RTO2/RTO4, RTO1/RTO3/RTO5) が操作され、6 つの別々のゲート信号が生成されます。これら 6 つのゲート信号は論理和がとられて GATE 信号を生成し、PPG0 カウントのトリガとなります。

また、PGEN0 ~ PGEN5 信号を使用すると、PPG0 のみを使用することで RTO0 ~ RTO5 端子に 6 つの異なる波形を出力することができます。

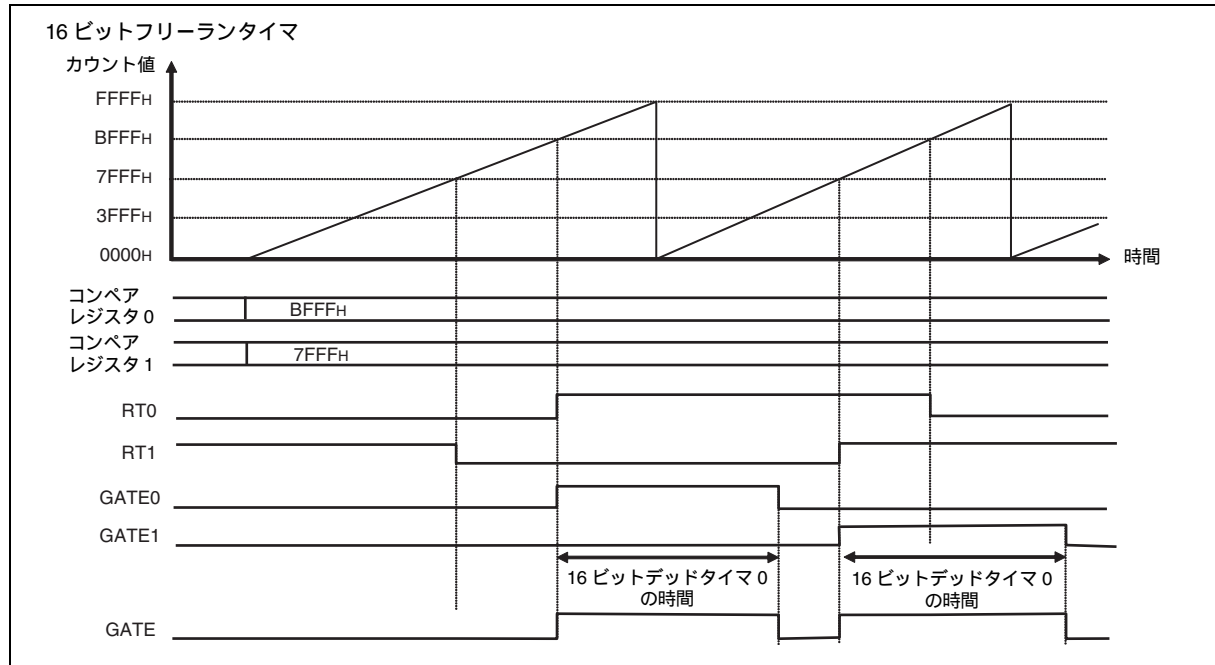
- GATENx がアクティブであり、各 RTx が "H" であるとき (16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "001_B" または "111_B") の GATE 信号生成

図 12.6-11 RTx が "H" のときの GATE 信号の生成



- GTENx がアクティブ (DTCR0, DTCR1, DTCR2 レジスタの TMD8 ~ TMD0=010_B) であるときの RTx 立上りエッジから 16 ビットデッドタイム 0, 1, 2 アンダフローまでにおける GATE 信号の生成

図 12.6-12 RTx 立上りエッジから 16 ビットデッドタイムアンダフローまでにおける GATE 信号の生成



< 注意事項 >

各 16 ビットデッドタイムは、2 つの RT に対して使用されます。すなわち、16 ビットデッドタイム 0 は RT0 と RT1 に対して使用され、16 ビットデッドタイム 1 は RT2 と RT3 に対して使用され、16 ビットデッドタイム 2 は RT4 と RT5 に対して使用されます。したがって、RT を使用して、既に動作中のタイマの起動を試みてはなりません。このような試みを行った場合、GATE 信号出力が拡張され、その結果、誤動作が発生する場合があります。

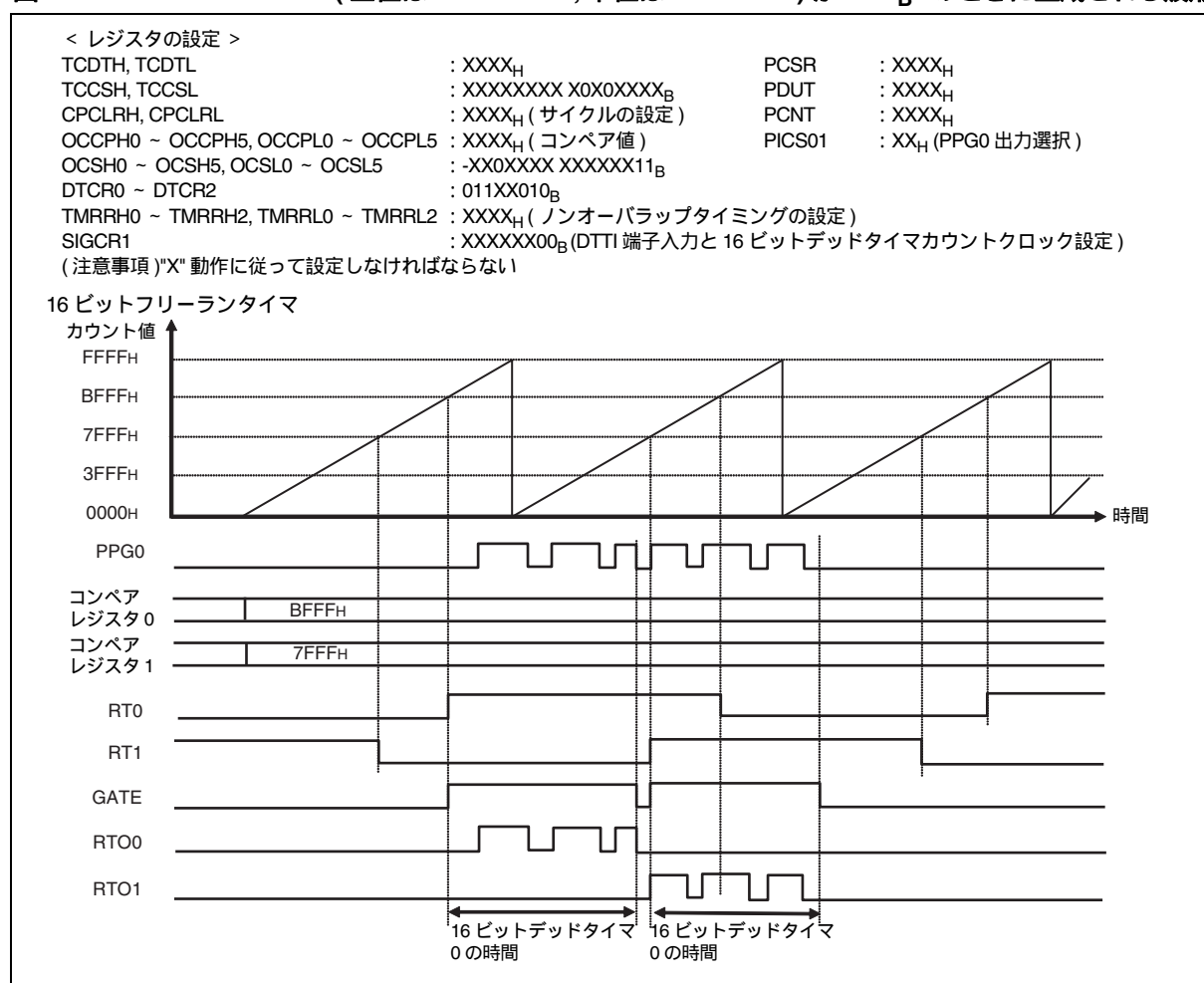
12.6.4.1 タイマモードの動作

RTO0 ~ RTO5 端子の立上りエッジが検出されると, 16 ビットデッドタイマに値がリロードされて, 16 ビットデッドタイマがダウンカウントを開始します。PPG タイマ 0 は, 16 ビットデッドタイマでアンダフローが発生するまで RTO0 ~ RTO5 端子へ出力し続けます。

■ タイマモードの動作

- RT立上りエッジから16ビットデッドタイマアンダフローまでにおけるPPG0出力パルス生成 (DTCR0, DTCR1, DTCR2 レジスタの TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 010_B)

図 12.6-13 TMD2 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "010_B" のときに生成される波形



< 注意事項 >

各 16 ビットデッドタイムは、2 つの RT に対して使用されます。すなわち、16 ビットデッドタイム 0 は RT0 と RT1 に対して使用され、16 ビットデッドタイム 1 は RT2 と RT3 に対して使用され、16 ビットデッドタイム 2 は RT4 と RT5 に対して使用されます。したがって、RT を使用して、既に動作中の PPG0 の起動を試みてはなりません。このような試みを行った場合は、GATE 信号出力が拡張され、その結果、誤動作が発生する場合があります。

12.6.4.2 デッドタイムタイマモード時の動作

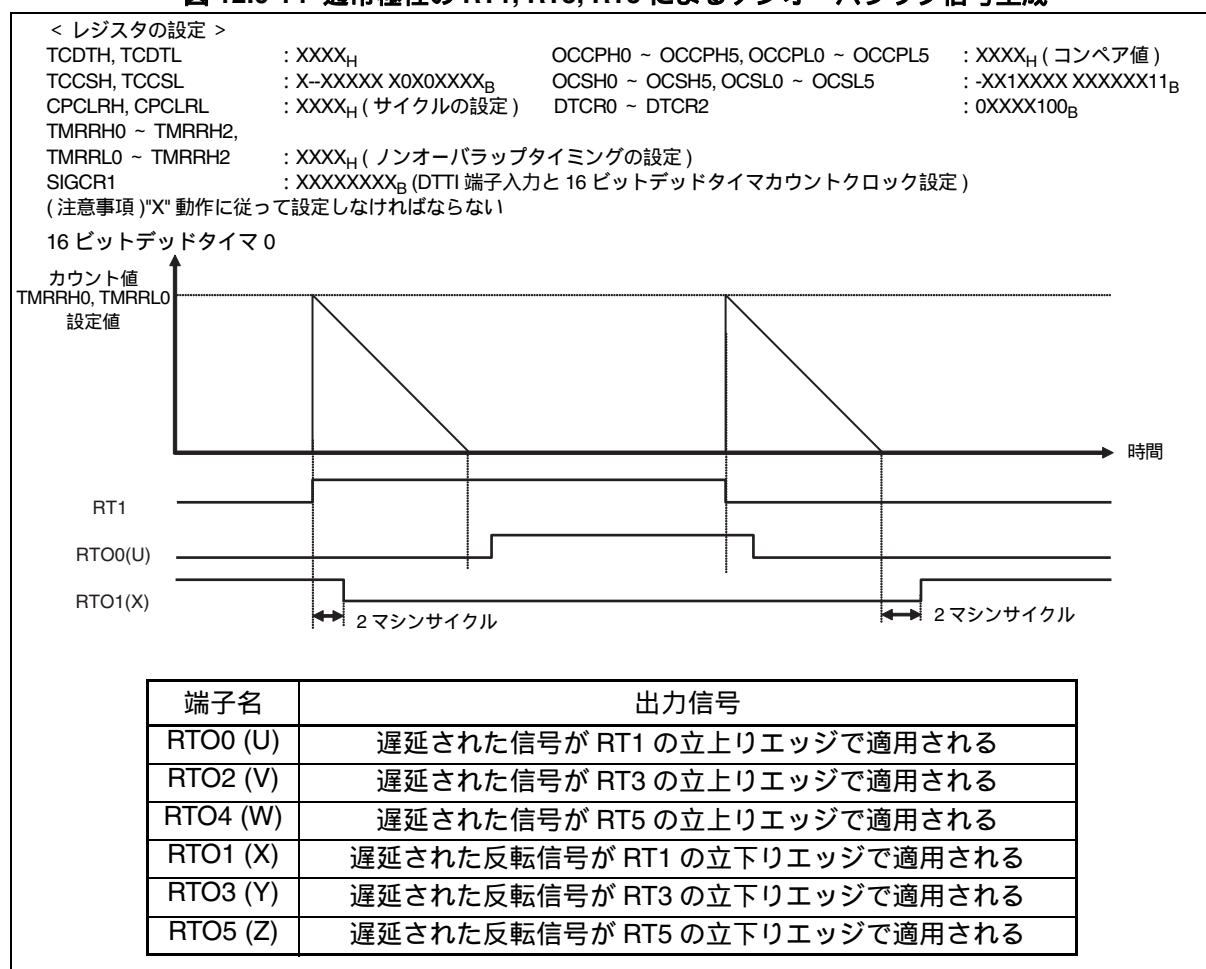
デッドタイムジェネレータは、リアルタイム出力 (RT1, RT3, RT5), もしくは, PPG0 タイマパルス出力を入力し, 外部端子 (RTO0 ~ RTO5) へノンオーバーラップ信号 (反転信号) を出力します。

■ デッドタイムタイマモード時の動作

- 通常極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成 (16ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 100_B)

DTCR0, DTCR1, DTCR2 レジスタの DMOD2 ~ DMOD0 が, "0" (通常極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は RT1, RT3, RT5 端子の立上りエッジまたは立下りエッジで適用されます。RT1, RT3, RT5 パルス幅が, 設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムは, その次の RT エッジの TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2 レジスタ値からダウンカウントを再開します。

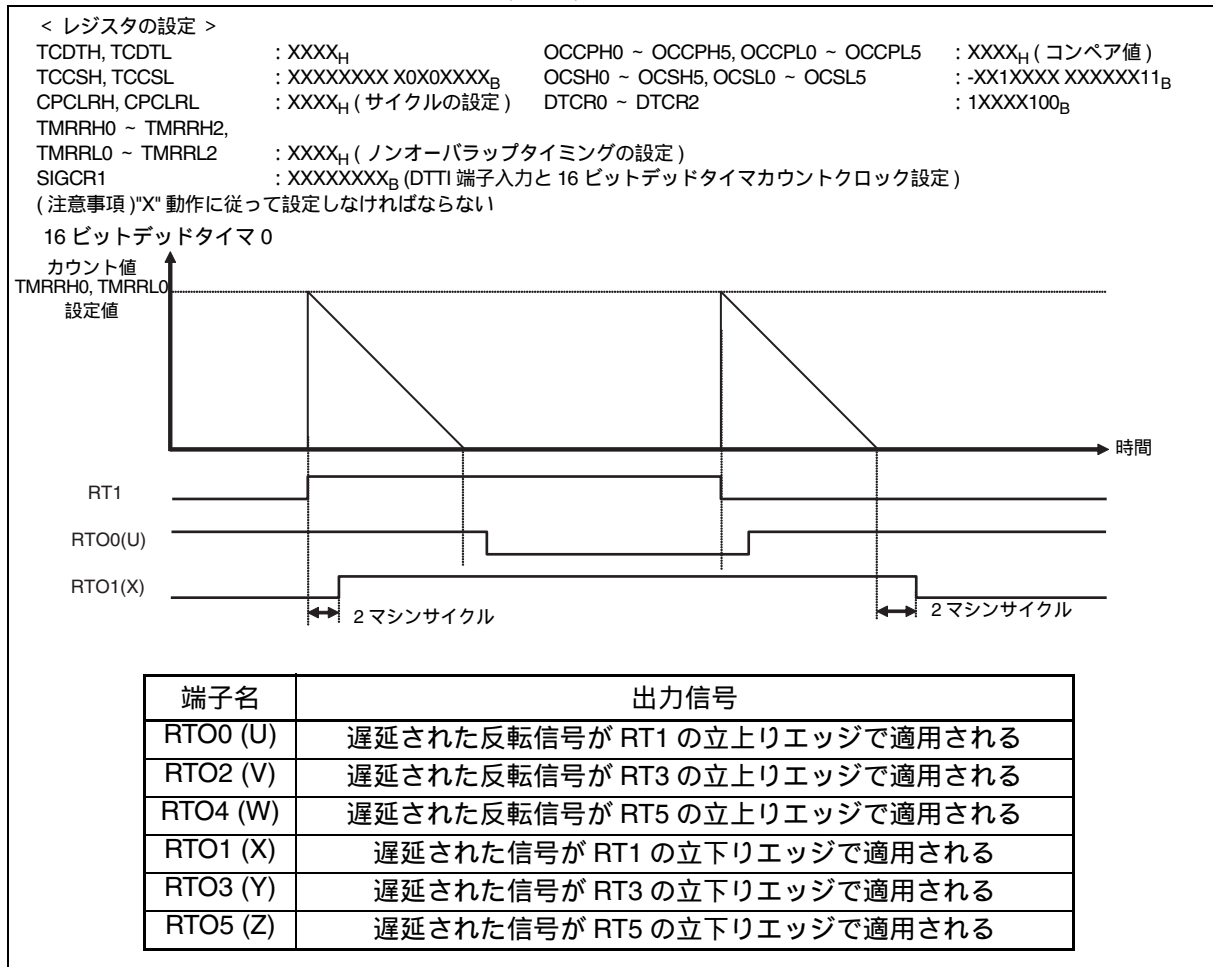
図 12.6-14 通常極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成



- 反転極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 100_B)

DTCR0, DTCR1, DTCR2 レジスタの DMOD2 ~ DMOD0 (上位は bit15, 下位は bit7) が "1" (反転極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, RT1, RT3, RT5 の立上りエッジまたは立下りエッジで適用されます。RT1, RT3, RT5 パルス幅が設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムはその次の RT エッジの TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2 値からダウンカウンタを再開します。

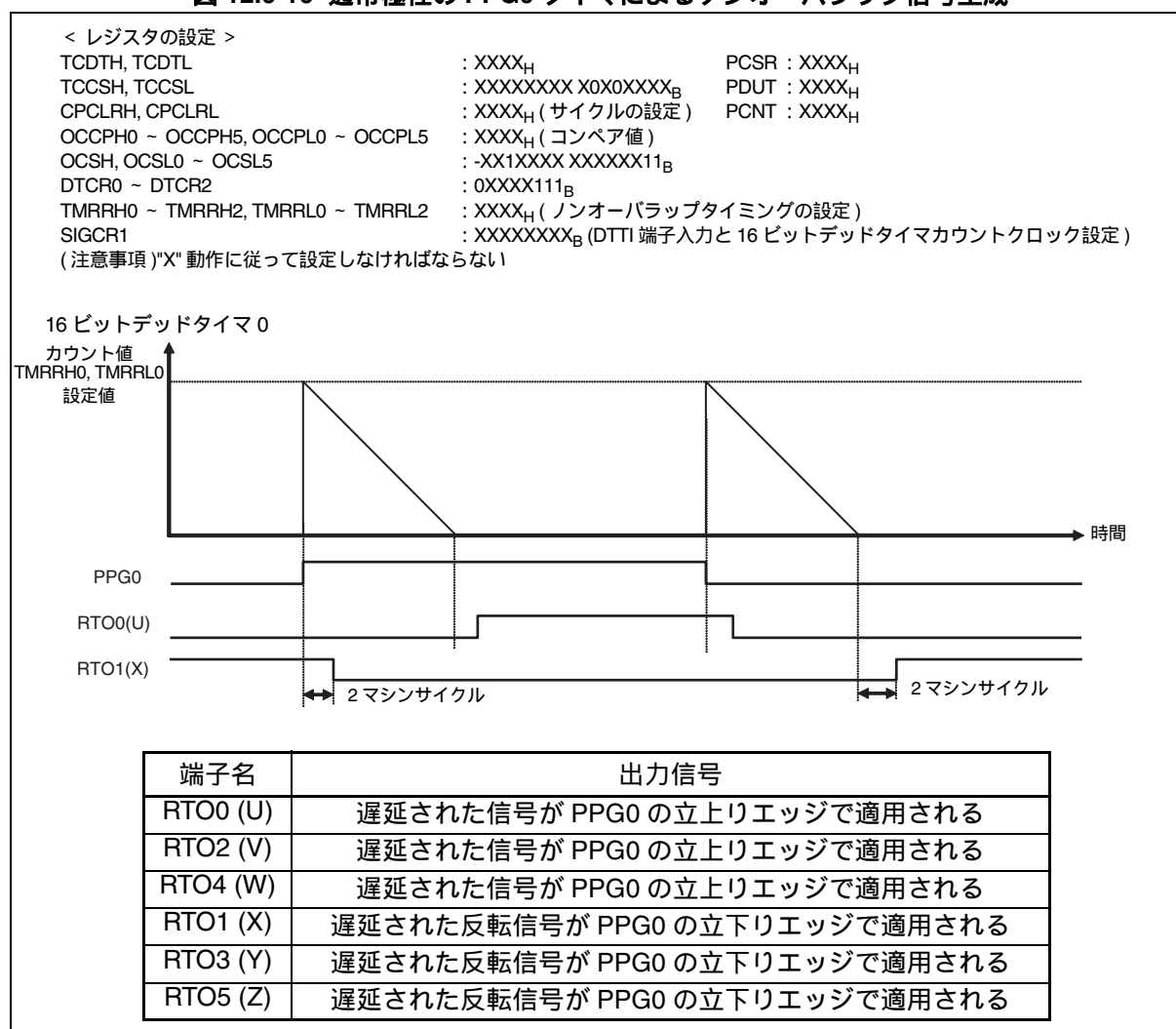
図 12.6-15 通常極性の RT1, RT3, RT5 によるノンオーバーラップ信号生成



- 通常極性の PPG によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 111_B)

DTCR0, DTCR1, DTCR2 レジスタの DMOD2 ~ DMOD0 (上位は bit15, 下位は bit7) が "0" (通常極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, PPG0 タイマパルス信号または反転信号の立上りエッジで適用されます。PPG0 タイマパルス幅が設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムは PPG0 パルスのその次のエッジの TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2 値からダウンカウントを再開します。

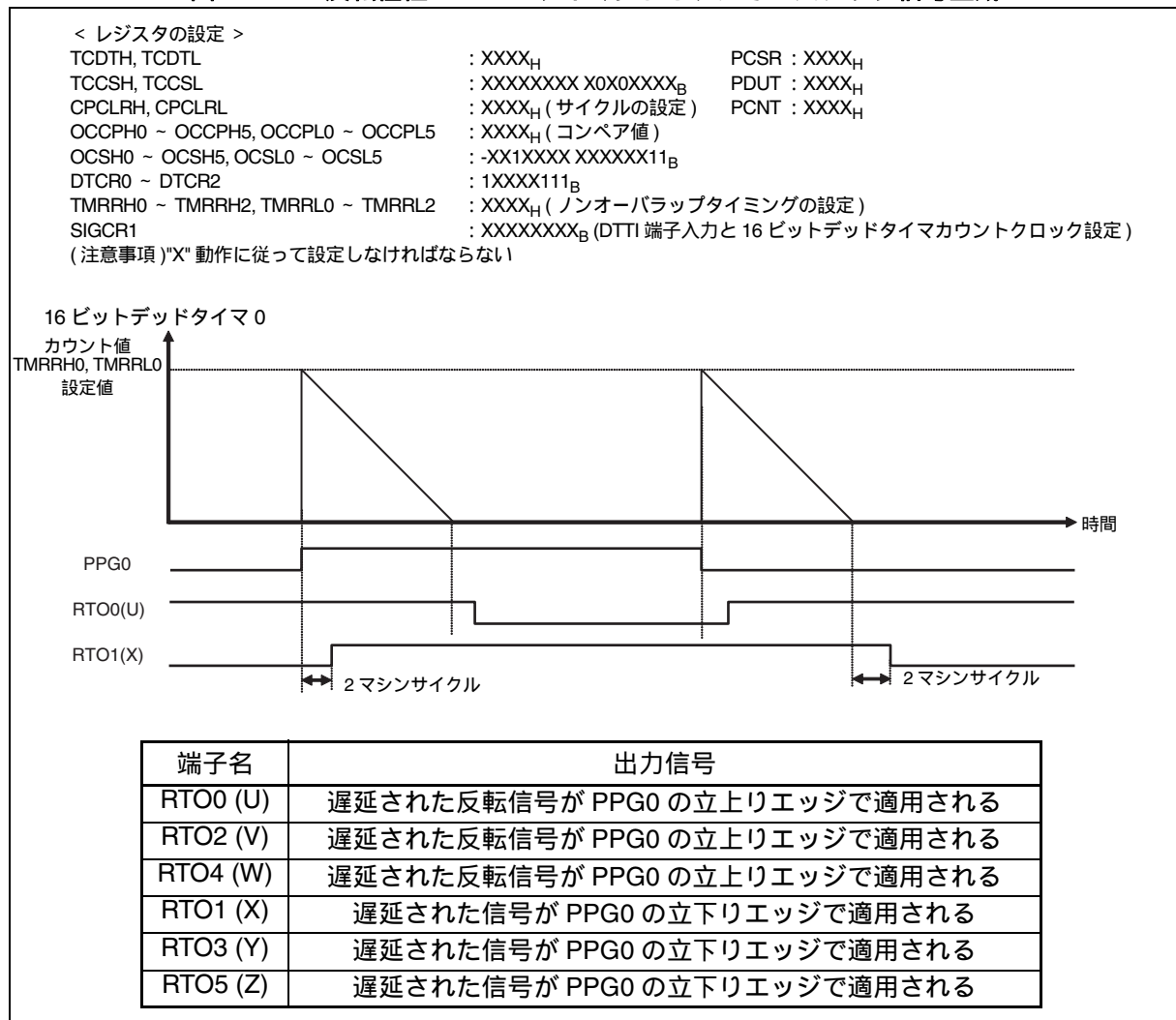
図 12.6-16 通常極性の PPG0 タイマによるノンオーバーラップ信号生成



- 反転極性の PPG によるノンオーバーラップ信号生成 (16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) = 111_B)

DTCR0, DTCR1, DTCR2 レジスタの DMOD2 ~ DMOD0 (上位は bit15, 下位は bit7) が "1" (反転極性) であるノンオーバーラップ信号を選択すると, 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) に設定されているノンオーバーラップ時間に相当する遅延が適用されます。この遅延は, PPG0 タイマパルス信号または反転信号の立上りエッジで適用されます。PPG0 タイマパルス幅が設定されているノンオーバーラップ時間より小さい場合は, 16 ビットデッドタイムは PPG0 パルスのその次のエッジの TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2 値からダウンカウントを再開します。

図 12.6-17 反転極性の PPG0 タイマによるノンオーバーラップ信号生成



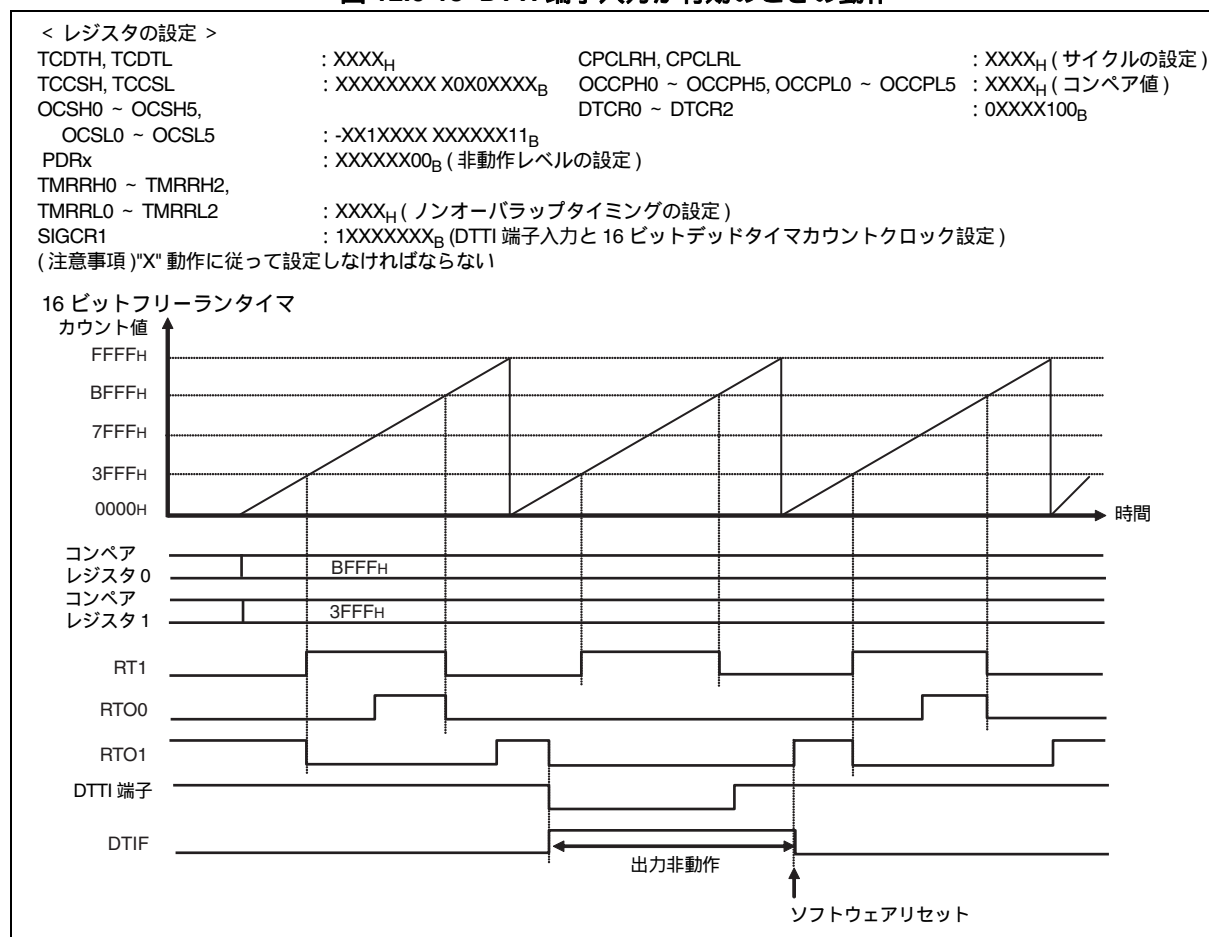
12.6.4.3 DTTI 端子制御の動作

波形制御レジスタ 1 (SIGCR1) の DTIE : bit7 に "1" を設定すると, RTO0 ~ RTO5 出力を DTTI 端子で制御することができます。DTTI 端子の "L" レベルが検出されると, RTO0 ~ RTO5 出力は, 割込みフラグ (SIGCR1 レジスタの DTIF : bit6) がクリアされるまで非動作レベルに固定されます。RTO0 ~ RTO5 の非動作レベルは, これらの端子を共用しているポートデータレジスタ (PDR) を使ってソフトウェア的に設定することができます。また, データディレクションレジスタ (DDR) を使って入力ポートとしておけば, Hi-Z 出力となります。

■ DTTI 端子入力 of 動作

DTTI 端子入力の "L" が検出された場合でも, タイマは波形ジェネレータが動作している間は動作を継続しますが, 波形は外部端子 RTO0 ~ RTO5 へは出力されません。

図 12.6-18 DTTI 端子入力が有効のときの動作



■ 波形制御レジスタ 2 (SIGCR2) の DTTI ビットの動作

- 波形制御レジスタ 2 の DTTI:bit0 を "0" にセットすると、本ビットを "1" にセットするまで RTO0 ~ RTO5 までの出力が非動作レベルに固定されます。
- DTIE ビットの状態にはよらず、DTTI ビットの "0" セットのみにより RTO0 ~ RTO5 の制御が可能です。
- ノイズキャンセル機能の影響は受けずに、DTTI ビットの "0" セット後、直ちに RTO0 ~ RTO5 が非動作レベルに固定されます。
- DTTI ビットへの "0" セットでは DTTI 端子割込み、ならびに割込みフラグ (DTIF) の "1" セットは発生しません。
- DTTI ビットへの "0" セット中に、DTTI 端子への "L" レベル入力検出された場合、DTTI 端子割込みフラグのセット、DTTI 端子割込み要求が発生します。

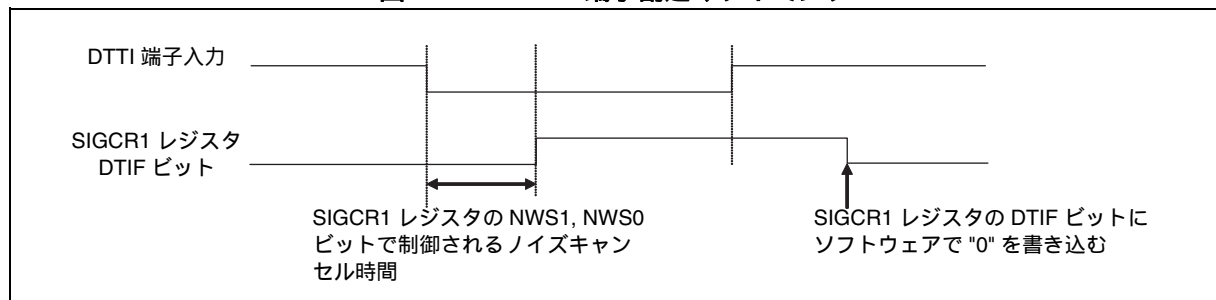
■ DTTI 端子ノイズキャンセル機能

波形制御レジスタ 1 (SIGCR1) の NRSL : bit5 に "1" を設定すると、DTTI 端子入力のノイズキャンセル機能が有効になります。ノイズキャンセル機能が有効になると、出力端子 (RTO0 ~ RTO5) を非動作レベルに固定するために要する時間が 4, 8, 16 または 32 マシンサイクル (SIGCR1 レジスタの NWS1, NWS0 : bit1, bit0 で選択) だけ遅延します。ノイズキャンセル回路はリソースを使用するので、発振が停止するモード (停止モードなど) において DTTI 端子入力が有効になった場合でも入力は無効になります。

■ DTTI 端子割込み

DTTI 端子の "L" レベルが検出されると、ノイズキャンセル時間が経過した後で DTTI 端子割込みフラグ (SIGCR1 レジスタの DTIF : bit6) に "1" が設定され、割込み要求は割込みコントローラへ送信されます。

図 12.6-19 DTTI 端子割込みタイミング



< 注意事項 >

- ノイズキャンセル時間内に SIGCR1 レジスタの NWS1, NWS0 ビットの値が変化した場合、さらに大きな (NWS1, NWS0) ノイズサイクル値が有効になります。
- SIGCR1 レジスタの DTIF : bit6 は、ソフトウェアでのみクリアすることができます。

12.6.5 A/D 起動コンペアの動作

16 ビットフリーランタイム 0 の値が指示値となったときに、A/D 起動を行うことができます。

■ A/D 起動

2 ユニットある A/D コンバータ起動ができます。

- A/D 起動コンペア 1 A/D ユニット 1 起動
- A/D 起動コンペア 2 A/D ユニット 2 起動

■ A/D コンペア起動許可

コンペアレジスタに値をセットし、制御レジスタ (ADCOMP1) の CE1, CE2 : bit1, bit2 に "1" をセットした場合、フリーランタイム 0 とコンペアレジスタ値が一致したときに、A/D に対して起動信号を発生します。

CE1, CE2 を "0" にセットすると、フリーランタイム 0 とコンペアレジスタ値が一致しても、A/D に対して起動信号は発生しません。

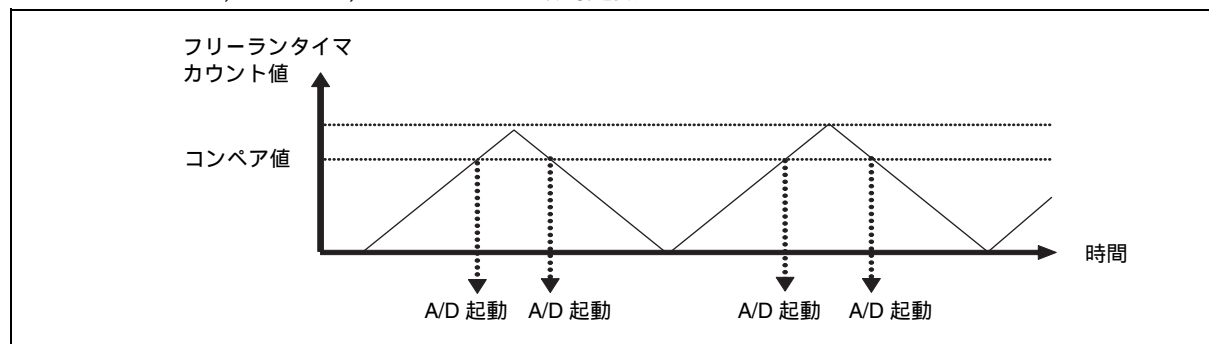
コンペアレジスタは、起動許可中に書き込んだ場合、すぐに、書き込んだ値と一致比較を行います。

フリーランタイム 1, 2 との一致起動はできません。

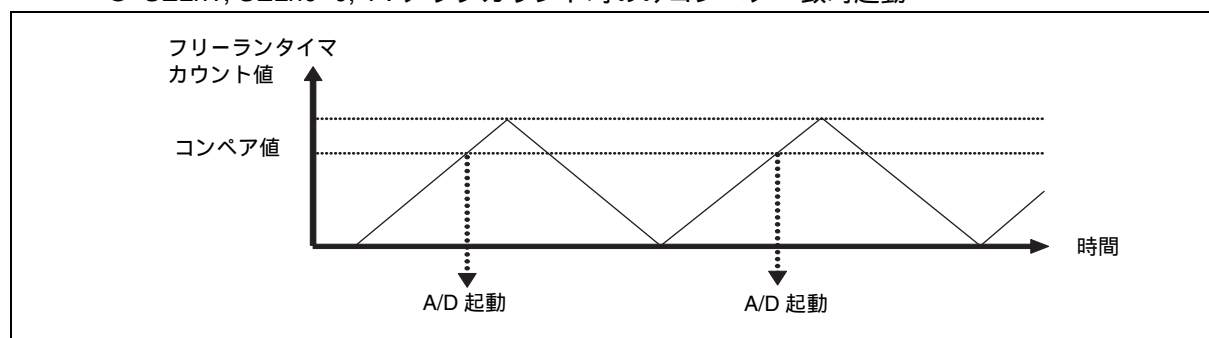
■ A/D コンペア起動モード

ADCOMP2 レジスタの SELn1, SELn0(n=1, 2) ビットにより、A/D 起動モードを設定できます。

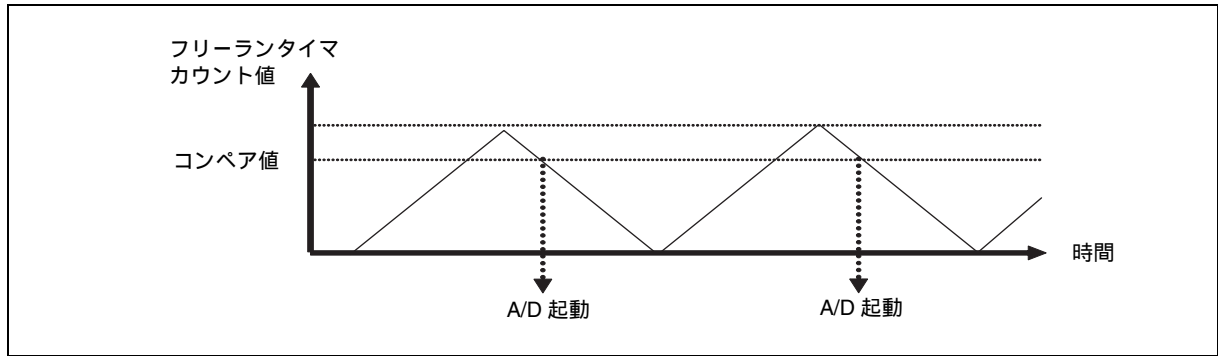
● SELn1, SELn0=0, 0 : コンペアー一致時起動



● SELn1, SELn0=0, 1 : アップカウント時のみコンペアー一致時起動



● SELn1, SELn0=1, 0 : ダウンカウント時のみコンペア一致時起動



● SELn1, SELn0=1, 1 : 設定禁止

12.7 多機能タイマの使用上の注意

多機能タイマの使用上の注意を以下に示します。

■ 16 ビットフリーランタイマの使用上の注意

● プログラムによる設定上の注意

- リセットを実行すると、タイマ値が "0000_H" になりますが、0 検出割込みフラグは設定されません。
- タイマモードビット (TCCSL レジスタの MODE : bit5) は、バッファを持っているので、0 検出後に変更されたタイマモードが有効になります。
- ソフトウェアクリア (TCCSL レジスタの SCLR : bit4=1) は、タイマを初期化しますが、0 検出割込みは発生しません。
- コンペア値とカウント値が一致しているときに、カウントを開始する場合は、コンペアクリアフラグは設定されません。

● 割込みの注意

- タイマ状態制御レジスタ上位 (TCCSH) の IRQZF : bit14 に "1" を設定し、次に割込み要求を許可すると (TCCSH レジスタの IRQZE : bit13=1)、制御は割込み処理から戻ることができません。IRQZF : bit14 は、必ずクリアしてください。
- タイマ状態制御レジスタ上位 (TCCSH) の ICLR : bit9 に "1" を設定し、次に割込み要求を許可すると (TCCSH レジスタの ICRE : bit8=1)、制御は割込み処理から戻ることができません。ICLR : bit9 は、必ずクリアしてください。

■ 16 ビットアウトプットコンペアの使用上の注意

● 割込みの注意

コンペア制御レジスタ下位 (OCSL0, OCSL2, OCSL4) の IOP1, IOP0 : bit7, bit6 に "1_B" を設定し、次に割込み要求を許可にすると (OCSL レジスタの IOE1, IOE0 : bit6, bit5=1_B)、制御は割込み処理から戻ることができません。IOP0, IOP1 ビットは、必ずクリアしてください。

■ 16 ビットインプットキャプチャの使用上の注意

● 割込みの注意

- インプットキャプチャ状態制御レジスタ下位 (PICSL01, ICSL23) の ICP3, ICP2, ICP1, ICP0 (bit7, bit6) に "1_B" を設定し、次に割込み要求を許可にすると (PICSL01, ICSL23 レジスタの ICE3, ICE2, ICE1, ICE0 (bit5, bit4)=1_B)、制御は割込み処理から戻ることができません。ICP3, ICP2, ICP1, ICP0 (bit7, bit6) は、必ずクリアしてください。
- インプットキャプチャ端子 (IC) レベルが、ICP3, ICP2, ICP1, ICP0 がビット設定されてから割込みルーチンが処理されるまでの間に切り換わると、ICP3, ICP2, ICP1, ICP0 の有効エッジ指示ビット (ICSH23 レジスタの IEI3, IEI2 : bit9, bit8, PICSH01 レジスタの IEI1, IEI0 : bit9, bit8) は、検出された最新のエッジを示します。

■ 波形ジェネレータの使用上の注意

● プログラムによる設定上の注意

- 波形ジェネレータが動作中 (DTCR0, DTCR1, DTCR2 レジスタの TMD2 ~ TMD0, TMD5 ~ TMD3, TMD8 ~ TMD6 が "001_B", "010_B", "100_B" または "111_B") に、16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMD8, TMD5, TMD2 (上位は bit10, 下位は bit2), TMD7, TMD4, TMD1 (上位は bit9, 下位は bit1), TMD6, TMD3, TMD0 (上位は bit8, 下位は bit0) ビット値を変更する場合は、トリガソースおよび 16 ビットデッドタイムがカウント中でないことを必ず確認してください。この操作を行わない場合は、以前のトリガでスケジュールされた出力が原因となり、予期しない波形が RTO 端子から出力されます。ただし、RTO 出力は、タイマでアンダフローが発生したり、新しいトリガソースで再トリガされたりすると、正常動作に戻ります。
- トリガソースとは、DTCR0, DTCR1, DTCR2 レジスタの TMD8 ~ TMD0 (上位は bit10 ~ bit8, 下位は bit2 ~ bit0) が "001_B" の場合は "RT の "H" レベル" であり、TMD8 ~ TMD0 ビットが "010_B" の場合は "RT の立上りエッジ" であり、TMD8 ~ TMD0 ビットが "100_B" の場合は "RT の立上りまたは立下りエッジ" であり、TMD8 ~ TMD0 ビットが "111_B" の場合は "PPG0 の立上りエッジまたは立下りエッジ" です。

例えば、TMD8 ~ TMD0 ビットが "100_B" から "111_B" へ変更すると、下記の手順を実行することができます。

1. 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) を "0001_H" のような非常に小さな値を設定する。
 2. RTO1, RTO3, RTO5 の出力を "L" または "H" に設定し、タイマ 0, 1, 2 でアンダフローが発生するまで待つ。
 3. モードビット (TMD8 ~ TMD0)、および対応する設定を変更する。
 4. 修正された出力波形が、1 マシンサイクル後、RTO 端子に現れる。
- タイマがカウント中に 16 ビットデッドタイムレジスタ (TMRRH0 ~ TMRRH2, TMRRL0 ~ TMRRL2) に値が書き込まれると、この新しい値は次のタイマトリガ時に有効になります。タイマレジスタへアクセスする場合は、必ずハーフワードもしくはワード転送命令をご使用ください。
 - タイマがカウントしていない場合のみ、波形制御レジスタ 1 (SIGCR1) の DCK2 ~ DCK0 : bit4 ~ bit2 を変更してください。
 - ノイズキャンセル機能が無効になった場合のみ、波形制御レジスタ 1 (SIGCR1) の NWS1, NWS0 : bit1, bit0 を変更してください。

● 割込みの注意

- 16 ビットデッドタイム制御レジスタ (DTCR0, DTCR1, DTCR2) の TMIF2 ~ TMIF0 (上位は bit12, 下位は bit4) に "1" を設定し、次に割込み要求を許可にすると (DTCR0, DTCR1, DTCR2 レジスタの TMIE2 ~ TMIE0 (上位は bit11, 下位は bit3) = 1)、制御は割込み処理から戻ることができません。TMIF ビットは、必ずクリアしてください。
- 波形制御レジスタ 1 (SIGCR1) の DTIF : bit6 に "1" を設定すると、制御は割込み処理から戻ることができません。DTIF ビットは、必ずクリアしてください。

12.8 多機能タイマのプログラム例

多機能タイマのプログラム例を示します。

■ 16 ビットフリーランタイマのプログラム例

● 処理

- 16 ビットフリーランタイマで 4 ms のときコンペアクリア割込みを発生します。
- 本タイマはアップカウントモード時にコンペアクリアタイマの再発生のために使用します。
- 16 MHz は、マシクロック用で、62.5 ns はカウントクロック用です。

● コーディング例

```

ICR32    .EQU      000460H    ; 16 ビットフリーランタイマのコンペアクリア
                                   ; 割込み制御レジスタ
TCCSH    .EQU      0000A8H    ; タイマ制御状態レジスタ
CPCLRBH  .EQU      0000A4H    ; コンペアクリアバッファレジスタ
; ----- メインプログラム -----
                ORG      C0000H

START:
;           ;           ; スタックポインタ (SP) が既に初期化されて
;           ;           ; いるものと仮定
                AND CCR   #0EFH    ; 割込み禁止
                LDI       #ICR32,r0
                LDI       #00H,r1
                STB       r1,@r0    ; 割込みレベル 16 (最強)
                LDI       #CPCLRBH,r0 ; 16 ビットフリーランタイマのアップカウント
                                   ; モード時に 4 ms でコンペアクリア割込みを
                                   ; 発生する
                LDI       #0FA00H,r1
                STH       r1,@r0    ; ための値をコンペアクリアバッファレジスタ
                                   ; に設定
                LDI       #TCCSH,r3 ; アップカウントダウンモード,
                LDI       #0110H,r1 ; 62.5 ns カウントクロック設定,
                STH       r1,@r3    ; コンペアクリア割込み許可,
                                   ; コンペアクリア割込みフラグビットのクリア,
                                   ; 割込みマスクを禁止,
                                   ; タイマクリア, 動作許可
                ST ILM     #14H     ; PS 中の ILM をレベル 20 に設定
                OR CCR     #10H     ; 割込み許可
                LOOP      LDI       #00H,r0 ; 無限ループ

```

```

LDI      #01H,r1
BRA      LOOP      ;
; ----- 割込みプログラム -----
WARI     LDI      #0100H,r1
          ANDH     r1,@r3      ; 割込み要求フラグをクリア
;
;      ユーザ処理
;
;      RETI      : 割込みから復帰

; ----- ベクタ設定 -----
VECT     .ORG      FFFF8H
          .DATA.W   WARI      ; 割込みルーチンの設定
          .ORG      FFFF8H
          .DATA.W   0x07000000 ; シングルチップモードの設定
          .DATA.W   START     ; リセットベクタの設定
          .END

```

■ 16 ビットアウトプットコンペアのプログラム例

● 処理

- 16 ビットフリーランタイマのカウント値がアウトプットコンペア用と一致したときアウトプットコンペア一致が発生します。
- 16 ビットフリーランタイマがアップダウンカウントモード時に使用します。
- 16 MHz は、マシクロック用で、62.5 ns は 16 ビットフリーランタイマ 0 のカウントクロック用です。

● コーディング例

```

ICR44    EQU      00046CH      ; アウトプットコンペア 0/1 割込みレジスタ
TCCSH    EQU      0000A8H      ; タイマ制御状態レジスタ
CPCLRBH  EQU      0000A4H      ; コンペアクリアバッファレジスタ
OCCPBH0  EQU      000090H      ; アウトプットコンペアバッファレジスタ 0
OCCPBH1  EQU      000092H      ; アウトプットコンペアバッファレジスタ 1
OCSH1    EQU      00009CH      ; コンペア制御レジスタ
; ----- メインプログラム -----
START:
;      :      ; スタックポインタ (SP) が既に初期化されて
;      :      ; いるものと仮定
          AND CCR #0EFH      ; 割込み禁止
          LDI      #ICR44,r0
          LDI      #00H,r1

```

```

STB      r1,@r0      ; 割込みレベル 16 ( 最強 )
LDI      #CPCLRBH,r0 ; 16 ビットフリーランタイムのコンペアクリア
LDI      #0FFFFH,r1  ; バッファレジスタに設定
STH      r1,@r0

LDI      #OCCPBH0,r0 ; アウトプットコンペアレジスタ 0 を設定
LDI      #0BFFFH,r1
STH      r1,@r0
LDI      #OCCPBH1,r0 ; アウトプットコンペアレジスタ 1 を設定
LDI      #07FFFH,r1
STH      r1,@r0
LDI      #OCOSH1,r3   ; アウトプットコンペア出力を有効
LDI      #6C33H,r2    ; コンペア一致割込み 0/1 を許可
STH      r2,@r3       ; 割込みフラグビットのクリア

LDI      #TCCSH,r0    ; アップカウントダウンモード ,
LDI      #0010H,r1    ; タイマクリア , 動作許可
STH      r1,@r0
ST ILM    #14H        ; PS 中の ILM をレベル 20 に設定
OR CCR    #10H        ; 割込み許可
LOOP      LDI      #00H,r0 ; 無限ループ
          LDI      #01H,r1
          BRA      LOOP    ;

; ----- 割込みプログラム -----
WARI :
          ANDH     r2,@r3   ; 割込みレジスタフラグをクリア
          ;
          ; ユーザ処理
          ;
          RETI          : 割込みから復帰

; ----- ベクタ設定 -----
VECT      .ORG      FFFF8H
          .DATA.W   WARI    ; 割込みルーチンの設定
          .ORG      FFFF8H
          .DATA.W   0x07000000 ; シングルチップモードの設定
          .DATA.W   START   ; リセットベクタの設定
          .END

```

第13章

U-TIMER

(UART ボーレート発生用 16 ビットタイマ)

U-TIMER の概要，レジスタの構成 / 機能および動作について説明します。

13.1 概要

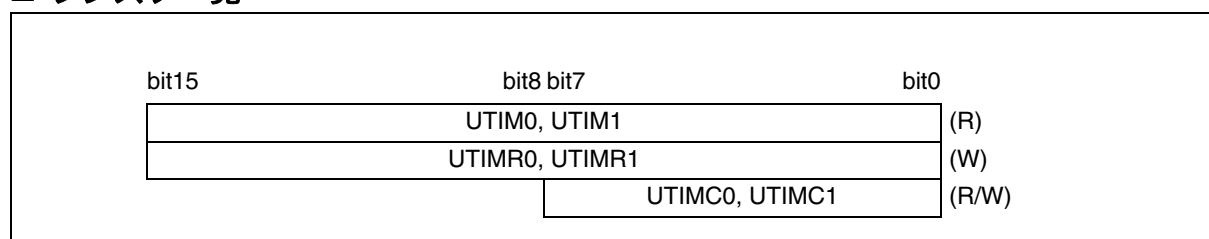
13.2 動作説明

13.1 概要

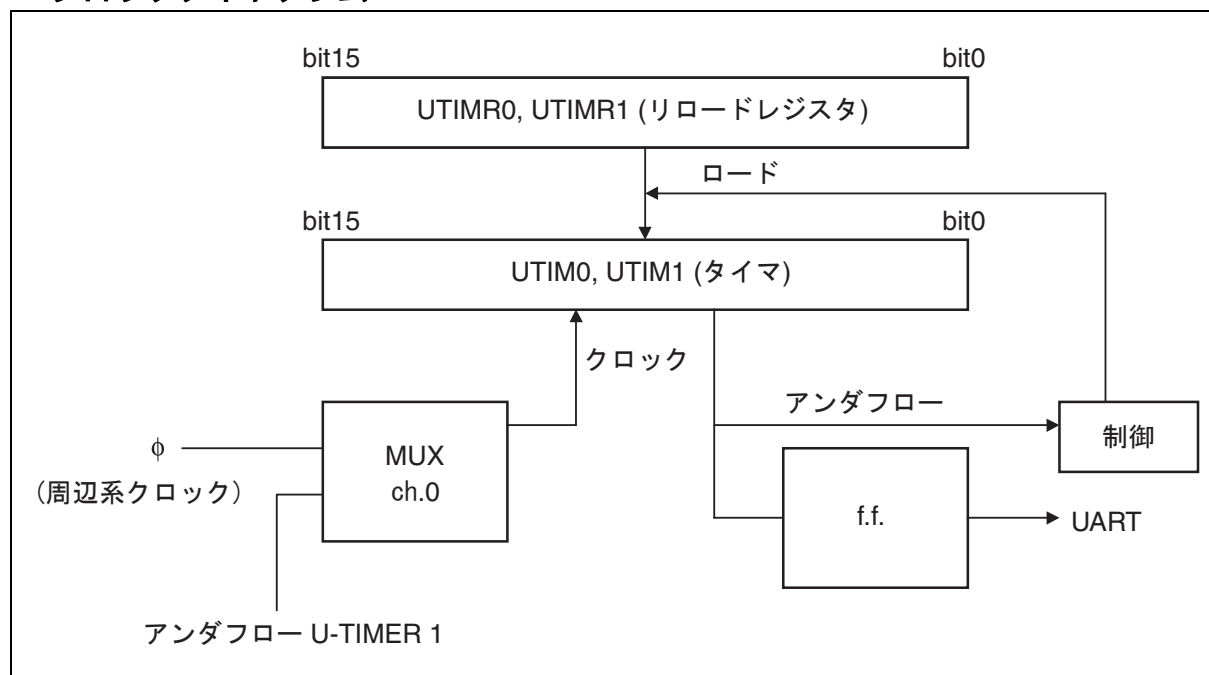
U-TIMER は、UART のポーレートを発生するための 16 ビットタイマです。チップの動作周波数と、U-TIMER のリロード値の組合せで任意のポーレートを設定できます。また、カウントアンダフローで割込みを発生するので、インターバルタイマとしても使用可能です。

MB91265A シリーズは、本タイマを 2 チャンネル内蔵しています。インターバルタイマとして使用する場合 2 組の U-TIMER をカスケードして使用でき、最大 $2^{32} \times \phi$ のインターバルをカウントできます。

■ レジスタ一覧



■ ブロックダイアグラム



■ レジスタ説明

● U-TIMER (UTIM: UTIM0, UTIM1)

UTIM (上位)									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000064 _H	b15	b14	b13	b12	b11	b10	b9	b8	00000000 _B
ch.1 :00006C _H	R	R	R	R	R	R	R	R	
UTIM (下位)									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000064 _H	b7	b6	b5	b4	b3	b2	b1	b0	00000000 _B
ch.1 :00006C _H	R	R	R	R	R	R	R	R	
R : リードオンリ									

UTIM は , タイマの値を示します。16 ビット転送命令でアクセスしてください。

● Reload register (UTIMR: UTIMR0, UTIMR1)

UTIMR (上位)									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
ch.0 :000064 _H	b15	b14	b13	b12	b11	b10	b9	b8	00000000 _B
ch.1 :00006C _H	W	W	W	W	W	W	W	W	
UTIMR (下位)									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000064 _H	b7	b6	b5	b4	b3	b2	b1	b0	00000000 _B
ch.1 :00006C _H	W	W	W	W	W	W	W	W	
W : ライトオンリ									

UTIMR は , UTIM がアンダフローしたときに UTIM にリロードされる値を格納するレジスタです。

このレジスタは必ず 16 ビット転送命令でアクセスしてください。

● U-Timer Control register (UTIMC: UTIMC0, UTIMC1)

UTIMC									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000067 _H	UCC1	-	-	UTIE	UNDR	CLKS	UTST	UTCR	0--00000 _B
ch.1 :00006F _H	R/W	-	-	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									
- : 未定義									

UTIMC は , U-TIMER の動作を制御します。

このレジスタは , 必ずバイト転送命令でアクセスしてください。

[bit7] UCC1 (U-timer Count Control 1)

UCC1 ビットは, U-TIMER のカウント方法を制御します。

UCC1	動作
0	通常動作 $= 2n + 2$ [初期値]
1	+ 1 モード $= 2n + 3$

n : UTIMR の設定値

: UART に対する出力クロックの周期

U-TIMER は, UART に対して通常の $2(n + 1)$ の周期のクロックのほかに奇数分周を設定できます。

UCC1 を "1" に設定すると $2n + 3$ の周期を発生します。

設定例 : 1. UTIMR=5, UCC1=0 : 発生周期 $= 2n + 2 = 12$ サイクル

2. UTIMR=25, UCC1=1 : 発生周期 $= 2n + 3 = 53$ サイクル

3. UTIMR=60, UCC1=0 : 発生周期 $= 2n + 2 = 122$ サイクル

U-TIMER をインターバルタイマとして使用する場合は, UCC1 の値を "0" に設定してください。

[bit6, bit5] - : 未定義ビット

[bit4] UTIE (U-Timer Interrupt Enable)

UTIE は, U-TIMER のアンダフローによる割込み許可ビットです。

0 : 割込み禁止 [初期値]

1 : 許可

[bit3] UNDR (UNDeR flow flag)

UNDR は, アンダフローが発生したことを示すフラグです。UTIE が "1" で UNDR がセットされるとアンダフロー割込みが発生します。UNDR はリセット, または "0" 書込みによりクリアされます。

リードモディファイライト(RMW)系命令の読出し時は, 常に "1" が読み出されます。

また, UNDR に対する "1" 書込みは無効です。

[bit2] CLKS (CLock Select)

CLKS は, U-TIMER の ch.0 と ch.1 のカスケード指定ビットです。

0: クロックソースは, 周辺系クロック (ϕ) [初期値]

1: U-TIMER ソースクロックタイミングとして, ch.0 のアンダフロー信号を使用
(ブロックダイアグラムの f.f)

CLKS は ch.1 でのみ有効, ch.0 では常に "0" にしてください。

(注意事項) ϕ (周辺系クロック = CLKP) は, ギアの設定により周期が変わります。

[bit1] UTST (U-Timer STart)

U-TIMER の動作許可ビットです。

0: 停止。動作中でも "0" 書込みで停止します [初期値]。

1: 動作。動作中に "1" を書き込んでも動作は続行されます。

[bit0] UTCR (U-Timer Clear)

UTCR に "0" を書き込むと U-TIMER は "0000_H" にクリアされます (f.f. も "0" にクリア)。

常に "1" が読み出されます。

< 注意事項 >

- ストップ状態からスタートビット UTST をアサート (スタート) すると自動的にリロードします。
 - ストップ状態からクリアビット UTCR とスタートビット UTST を同時にアサートすると、カウンタを "0" クリアして、直後のカウントダウンでアンダフローが発生します。
 - 動作中にクリアビット UTCR をアサートすると、カウンタも "0" クリアされます。このため、出力波形にヒゲ状の短いパルスが出力される場合があります。UART やカスケードモードの上位側の U-TIMER が誤動作する可能性があります。出力クロックを使用している場合には、動作中にクリアビットによるクリアを行わないでください。
 - カスケードモードで、下位側の UTIMR (リロードレジスタ) に "0" または "1" をセットすると正しくカウントされません。
 - U-TIMER コントロールレジスタの bit1 (U-TIMER スタートビット: UTST) および bit0 (U-TIMER のクリアビット: UTCR) をタイマ停止状態で同時にアサートすると、クリアした後のカウンタロードのタイミングで同レジスタの bit3 (アンダフローフラグ: UNDR) がセットします。また、内部ボーレートクロックは同タイミングで "H" レベルになります。
 - 割込み要求セットタイミングとクリアタイミングが重複した場合には、フラグセットが優先され、クリア動作は、無効となります。
 - ch.0 をカスケードモードで使用しない場合や本モジュールを単にタイマ機能として使用する場合は、U-TIMER コントロールレジスタの bit2 (基準クロック選択ビット: CLKS) には常に "0" を書き込んでください。また、"CLKS" の設定変更は、本モジュールが動作停止の状態で行ってください。
 - U-TIMER リロードレジスタへの書き込みとリロードのタイミングが重複した場合には、旧データがカウンタにロードされ、新データがカウンタにロードされるのは、次のリロードのタイミングとなります。
 - タイマクリアとタイマカウント / リロードのタイミングが重複した場合には、タイマクリアが最優先されます。
 - クロック同期モードを使用する場合、UTIMR のリロード値は "3" 以上に設定してください。
-

13.2 動作説明

UART のボーレートの計算とカスケードモードについて説明します。

■ ボーレートの計算

UART は、対応する U-TIMER (U-TIMER0 UART0, U-TIMER1 UART1) のアンダフローフリップフロップ (図中 f.f.) をボーレート用クロックソースとして使用します。

● 非同期 (調歩同期) モード

UART は、U-TIMER の出力を 16 分周して使用します。

$$\text{bps} = \frac{\phi}{(2n + 2) \times 16} \dots\dots\dots \text{UCC1=0 時} \quad \begin{array}{l} n \dots \text{UTIMR (リロード値)} \\ \phi \dots \text{周辺系マシクロック周波数} \\ \quad \text{(ギアにより変動)} \end{array}$$

$$\text{bps} = \frac{\phi}{(2n + 3) \times 16} \dots\dots\dots \text{UCC1=1 時}$$

最大 bps : クロック 33 MHz 時 1031250bps

● クロック同期モード

$$\text{bps} = \frac{\phi}{(2n + 2)} \dots\dots\dots \text{UCC1=0 時} \quad \begin{array}{l} n \dots \text{UTIMR (リロード値)} \\ \quad \text{(注意事項) } n \text{ は "3" 以上としてください} \\ \phi \dots \text{周辺系マシクロック周波数} \\ \quad \text{(ギアにより変動)} \end{array}$$

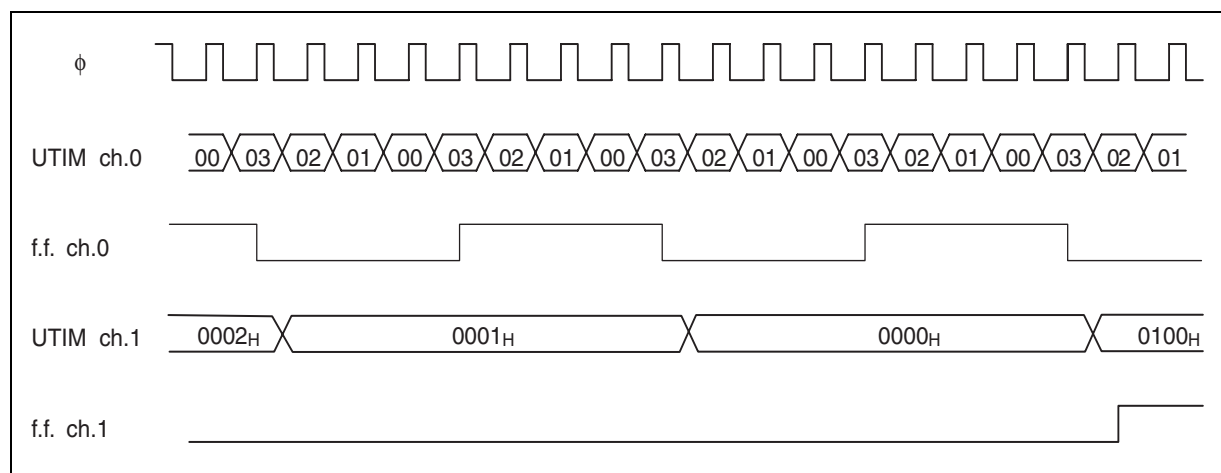
$$\text{bps} = \frac{\phi}{(2n + 3)} \dots\dots\dots \text{UCC1=1 時}$$

最大 bps : クロック 33 MHz 時 4125000 bps

■ カスケードモード

U-TIMER ch.0, ch.1 はカスケードモードで使用できます。

例) UTIMR ch.0 を "0003_H" に、UTIMR ch.1 を "0100_H" に設定した場合:



第14章

UART

UART の概要，レジスタの構成 / 機能および動作について説明します。

- 14.1 概要
- 14.2 レジスタ詳細説明
- 14.3 UART の動作
- 14.4 応用例
- 14.5 ボーレートと U-TIMER のリロード値の設定例

14.1 概要

UART は、非同期（調歩同期）通信、またはクロック同期通信を行うためのシリアル I/O ポートです。MB91265A シリーズは、UART を 2 チャンネル内蔵します。

■ UART の特長

- 全二重ダブルバッファ
- 非同期（調歩同期）、クロック同期通信が可能
- マルチプロセッサモードのサポート
- 完全プログラマブルボーレート
内蔵タイマにより任意のボーレートを設定可能（「第 13 章 U-TIMER（UART ボーレート発生用 16 ビットタイマ）」を参照）
- 外部クロックによる自由なボーレートの設定が可能
- エラー検出機能（パリティ、フレーミング、オーバラン）
- 転送信号は NRZ 符号
- 割込みによる DMA 転送の起動が可能

■ レジスタ一覧

図 14.1-1 UART のレジスタ一覧

bit15				bit8 bit7				bit0			
SCR				SMR				(R/W, W)			
SSR				SIDR(R)/SODR(W)				(R/W, R, W)			
8 ビット				8 ビット							

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	シリアルインプットデータレジスタ シリアルアウトプットデータレジスタ (SIDR/SODR)
D7	D6	D5	D4	D3	D2	D1	D0	

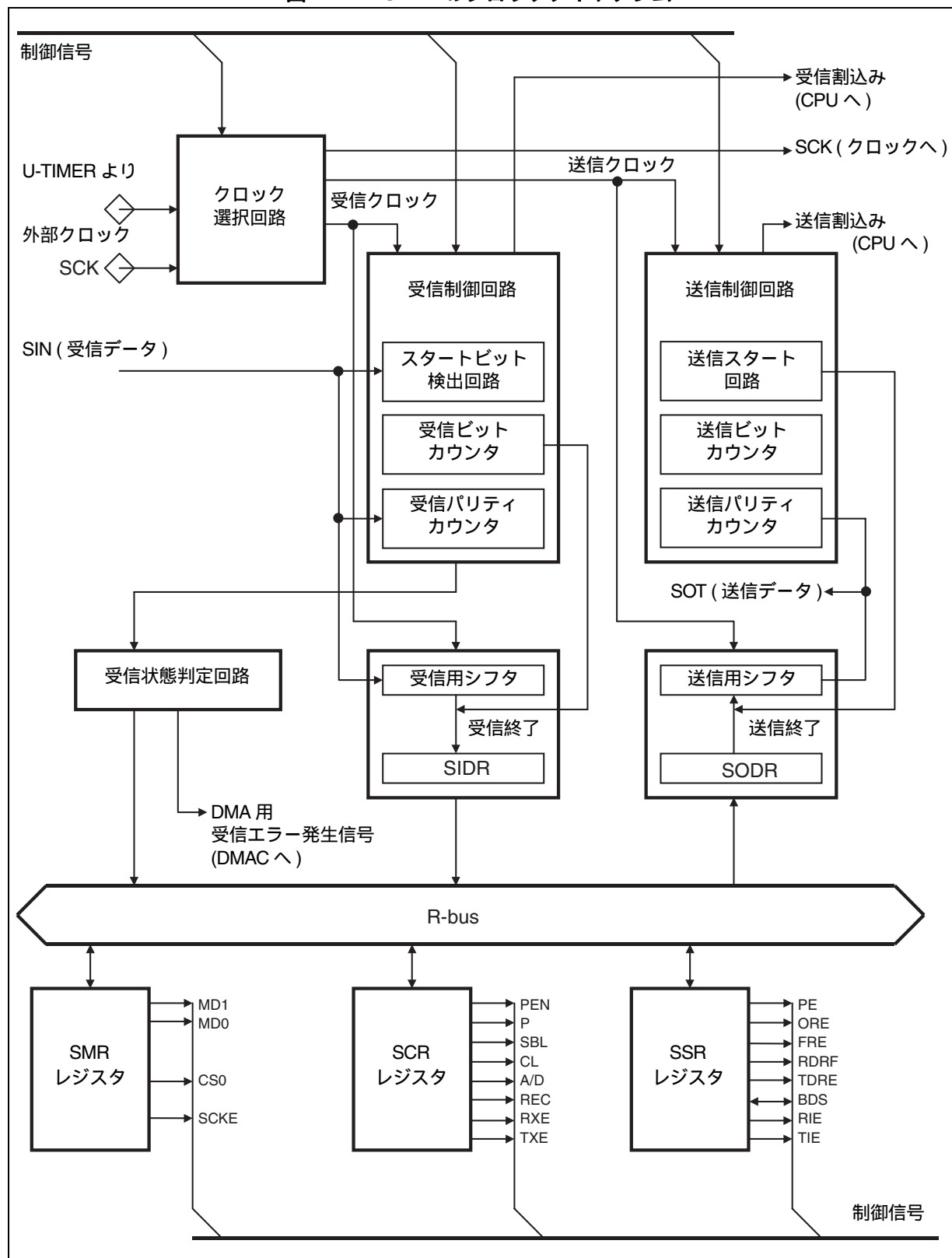
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	シリアルステータスレジスタ (SSR)
PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	シリアルモードレジスタ (SMR)
MD1	MD0	-	-	CS0	-	SCKE	-	

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	シリアルコントロールレジスタ (SCR)
PEN	P	SBL	CL	A/D	REC	RXE	TXE	

■ ブロックダイアグラム

図 14.1-2 UART のブロックダイアグラム



14.2 レジスタ詳細説明

UART で使用するレジスタの詳細について説明します。

■ シリアルモードレジスタ (SMR: SMR0, SMR1)

SMR									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000063 _H	MD1	MD0	-	-	CS0	-	SCKE	-	00--0-0-B
ch.1 :00006B _H	R/W	R/W	R/W	R/W	W	R/W	R/W		
R/W : リード/ライト可能									
W : ライトオンリ									
- : 予約									

SMR は, UART の動作モードを指定します。動作モードの設定は動作停止中に行い, 動作中にこのレジスタへの書込みは行わないでください。

[bit7, bit6] MD1, MD0 : (MoDe select)

UART の動作モードを選択します。

表 14.2-1 動作モード選択

モード	MD1	MD0	動作モード
0	0	0	非同期 (調歩同期) : ノーマルモード [初期値]
1	0	1	非同期 (調歩同期) : マルチプロセッサモード
2	1	0	クロック同期モード
-	1	1	設定禁止

< 注意事項 >

モード 1 のクロック非同期モード (マルチプロセッサ) とは, 1 台のホスト CPU に数台のスレーブ CPU が接続される使用法です。本リソースでは, 受信データのデータ形式を判別できません。したがって, マルチプロセッサモード のマスタのみをサポートします。
また, パリティチェック機能は使用できませんので SCR レジスタの PEN は "0" に設定してください。

[bit5, bit4] - : 予約ビット

常に "1" を書き込んでください。

[bit3] CS0(Clock Select)

UART の動作クロックを選択します。

0 : 内蔵タイマ (U-TIMER) [初期値]

1 : 外部クロック

[bit2] - : 予約ビット

常に "0" を書き込んでください。

[bit1] SCKE : (SCLK Enable)

クロック同期モード (モード 2) で通信を行う場合, SCK 端子をクロック入力端子にするか, クロック出力端子として使うかを指定します。

クロック非同期モード時または外部クロックモード時では "0" に設定してください。

0: クロック入力端子として機能します [初期値]。

1: クロック出力端子として機能します。

< 注意事項 >

クロック入力端子として使用するには, CS0 ビットを "1" にして外部クロックを選択しておく必要があります。

[bit0] - : 未定義ビット

未使用ビットです。

■ シリアルコントロールレジスタ (SCR: SCR0, SCR1)

SCR									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000062 _H	PEN	P	SBL	CL	A/D	REC	RXE	TXE	00000100 _B
ch.1 :00006A _H	R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	
R/W : リード / ライト可能									
W : ライトオンリ									

SCR はシリアル通信を行う場合の転送プロトコルを制御します。

[bit7] PEN (Parity Enable)

シリアル通信において、パリティを付加してデータ通信を行うかどうかを指定します。

0	パリティなし [初期値]
1	パリティあり

< 注意事項 >

パリティを付加できるのは、非同期 (調歩同期) 通信モードのノーマルモード (モード 0) のみです。マルチプロセッサモード (モード 1) およびクロック同期通信 (モード 2) では、パリティを付加することはできません。

[bit6] P (Parity)

パリティを付加してデータ通信を行うとき、偶数 / 奇数パリティを指定します。

0	偶数パリティ [初期値]
1	奇数パリティ

[bit5] SBL (Stop Bit Length)

非同期 (調歩同期) 通信を行うときのフレームエンドマークである、ストップビットのビット長を指定します。

0	1 ストップビット [初期値]
1	2 ストップビット

[bit4] CL (Character Length)

送受信する 1 フレームのデータ長を指定します。

0	7 ビットデータ [初期値]
1	8 ビットデータ

< 注意事項 >

7 ビットデータを扱えるのは、非同期（調歩同期）通信のうちのノーマルモード（モード 0）のみです。マルチプロセッサモード（モード 1）およびクロック同期通信（モード 2）では、8 ビットデータとしてください。

[bit3] A/D (Address/Data)

非同期（調歩同期）通信のマルチプロセッサモード（モード 1）において、送受信するフレームのデータ形式を指定します。

0	データフレーム [初期値]
1	アドレスフレーム

[bit2] REC (Receiver Error Clear)

0	"0" を書き込むことで、SSR レジスタのエラーフラグ (PE, ORE, FRE) をクリアします。
1	"1" 書込みは無効であり、読出し値は常に "1" になります。

[bit1] RXE (Receiver Enable)

UART の受信動作を制御します。

0	受信動作を禁止します [初期値]
1	受信動作を許可します

< 注意事項 >

受信中（受信シフトレジスタにデータが入力されているとき）に受信動作を禁止した場合には、そのフレームの受信を完了し受信データバッファ SISR レジスタに受信データをストアしたところで受信動作を停止します。

[bit0] TXE : (Transmitter Enable)

UART の送信動作を制御します。

0	送信動作を禁止します [初期値]
1	送信動作を許可します

< 注意事項 >

送信中（送信レジスタからデータが出力されているとき）に送信動作を禁止した場合は、送信データバッファ SODR レジスタにデータがなくなった後に送信動作を停止します。

■ シリアルインプットデータレジスタ (SIDR: SIDR0, SIDR1) /
シリアルアウトプットデータレジスタ (SODR: SODR0, SODR1)

SIDR								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ch.0 :000061 _H	D7	D6	D5	D4	D3	D2	D1	D0
ch.1 :000069 _H	R	R	R	R	R	R	R	R
初期値 XXXXXXXX _B								
SODR								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ch.0 :000061 _H	D7	D6	D5	D4	D3	D2	D1	D0
ch.1 :000069 _H	W	W	W	W	W	W	W	W
初期値 XXXXXXXX _B								
R : リードオンリ								
W : ライトオンリ								
X : 不定								

本レジスタは、受信 / 送信用のデータバッファレジスタです。

データ長が 7 ビットの場合 SIDR, SODR の bit7 (D7) は無効データとなります。BDS=1
のときは、SIDR と SODR のアクセス時に、バス上のデータの上位側と下位側を入れ替
えるため、見かけ上 bit0 (D0) が無視されたように見えます。

SODR レジスタへの書込みは、SSR レジスタの TDRE が "1" のときに書き込んでくださ
い。

< 注意事項 >

このアドレスへの書込みは SODR レジスタへの書込みを、読出しは SIDR レジスタの読
出しを意味します。

■ シリアルステータスレジスタ (SSR: SSR0, SSR1)

SSR									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
ch.0 :000060 _H	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	00001000 _B
ch.1 :000068 _H	R	R	R	R	R	R/W	R/W	R/W	
R/W : リード / ライト可能									
R : リードオンリ									

SSR は UART の動作状態を表すフラグで構成されています。

[bit7] PE (Parity Error)

受信時にパリティエラーが発生したときにセットされる割込み要求フラグです。
一度セットされたフラグをクリアするには、SCR レジスタの REC ビット (bit10) に "0" を書き込みます。

本ビットがセットされた場合には、SIDR のデータは無効データとなります。

0	パリティなし [初期値]
1	パリティあり

[bit6] ORE (Over Run Error)

受信時にオーバランエラーが発生したときにセットされる割込み要求フラグです。
一度セットされたフラグをクリアするには、SCR レジスタの REC ビットに "0" を書き込みます。

本ビットがセットされた場合には、SIDR のデータは無効データとなります。

0	オーバランエラーなし [初期値]
1	オーバランエラーあり

[bit5] FRE (FRaming Error)

受信時にフレーミングエラーが発生したときにセットされる割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 ビットに "0" を書き込みます。

本ビットがセットされた場合には、SIDR のデータは無効データとなります。

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あり

< 注意事項 >

- シリアルモードレジスタの bit3 による内部/外部ボーレートクロックの切換えは書込み後、すぐに反映されるので、UART が動作停止状態のときに行ってください。
 - シリアルモードレジスタの bit3 は書込みのみ可能です。
-

[bit4] RDRF (Receiver Data Register Full)

SIDR レジスタに受信データがあることを示す割込み要求フラグです。

SIDR レジスタに受信データがロードされるとセットされ、SIDR レジスタを読み出すと自動的にクリアされます。

0	受信データなし [初期値]
1	受信データあり

[bit3] TDRE (Transmitter Data Register Empty)

SODR に送信データを書き込めることを示す割込み要求フラグです。

SODR レジスタに送信データを書き込むとクリアされます。書き込んだデータが送信用シフトにロードされて、転送が開始されると再びセットされ、次の送信データを書き込めることを表します。

0	送信データの書き込み禁止
1	送信データの書き込み許可 [初期値]

[bit2] BDS(Bit Direction Select)

転送方向選択ビットです。

0	最下位ビット (LSB) 側から転送します [初期値]
1	最上位ビット (MSB) 側から転送します

< 注意事項 >

シリアルデータレジスタへの読出し、書き込み時にデータの上位側と下位側を入れ替えるため、SODR レジスタへデータを書き込んだ後、このビットを書き換えると、そのデータは無効になります。

ハーフワード (16 ビット) にて、SODR レジスタと BDS を同時に書き換えた場合は、書き込み前の BDS の値に従い、SODR レジスタへの書き込みが行われます。

[bit1] RIE (Receiver Interrupt Enable)

受信割込みを制御します。

0	割込みを禁止します [初期値]
1	割込みを許可します

< 注意事項 >

受信割込み要因は、PE, ORE, FRE によるエラー発生のほか、RDRF による正常受信があります。

[bit0] TIE (Transmitter Interrupt Enable)

送信割込みを制御します。

0	割込みを禁止します [初期値]
1	割込みを許可します

< 注意事項 >

送信割込み要因は、TDRE による送信要求があります。

14.3 UART の動作

UART の各動作モードにおける動作について説明します。UART には、非同期 (調歩同期) モードとクロック同期モードがあり、さらに、非同期モードには、ノーマルモードとマルチプロセッサモードがあります。

■ 動作モード

UART は、表 14.3-1 に示す動作モードを持ち、SMR レジスタ、SCR レジスタに値を設定することによりモードを切り換えることができます。

表 14.3-1 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	あり / なし	7	非同期 (調歩同期) : ノーマルモード	1 ビットまたは 2 ビット
	あり / なし	8		
1	なし	8 + 1	非同期 (調歩同期) : マルチプロセッサ モード	
2	なし	8	クロック同期モード	なし

ただし、非同期 (調歩同期) モードでのストップビット長については送信動作のみ指定が可能です。受信動作については常に 1 ビット長となります。上記モード以外では動作しませんので、設定しないでください。

■ UART のクロック選択

● 内部タイマ

CS0 を "0" に設定して U-TIMER を選択した場合は、U-TIMER に設定したリロード値でボーレートが決まります。このときのボーレートの算出式は次のとおりです。

$$\text{非同期 (調歩同期)} \quad \phi / (16 \times \beta)$$

$$\text{クロック同期} \quad \phi / \beta$$

ϕ : 周辺系マシンのクロック周波数

β : U-TIMER で設定した周期 ($2n + 2$ または $2n + 3$, n はリロード値 [$n - 3$])

非同期 (調歩同期) モードのボーレートは、設定したボーレートの $-1\% \sim +1\%$ までの範囲で転送が可能です。

● 外部クロック

CS0 を "1" に設定して外部クロックを選択した場合のボーレートは外部クロックの周波数を f とすると次のようになります。

$$\text{非同期 (調歩同期)} \quad f/16$$

$$\text{クロック同期} \quad f$$

ただし、 f は最大 3.125 MHz までです。

■ 非同期 (調歩同期) モード

● 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 14.3-1 にデータフォーマットを示します。

図 14.3-1 転送データフォーマット (モード 0, モード 1)

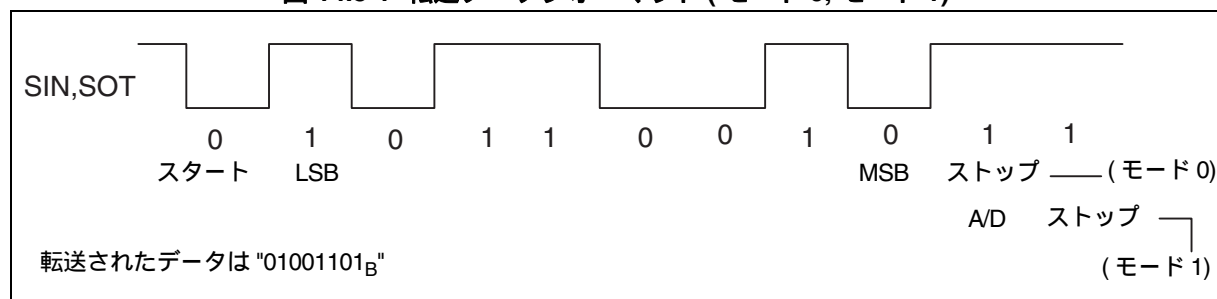


図 14.3-1 に示すように、転送データは必ずスタートビット ("L" レベルデータ) より始まり、LSB ファーストで指定されたデータビット長の転送が行われ、ストップビット ("H" レベルデータ) で終了します。外部クロックを選択している場合は、常にクロックを入力してください。

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 ビットに設定できますが、マルチプロセッサモード (モード 1) では 8 ビットでなければなりません。また、マルチプロセッサモードではパリティを付加することはできません。そのかわり、A/D ビットが必ず付加されます。

● 受信動作

SCR レジスタの RXE ビット (bit1) が "1" ならば、常に受信動作が行われています。

受信ラインにスタートビットが現れると、SCR レジスタで決められたデータフォーマットに従い 1 フレームデータの受信が行われます。1 フレームの受信が終わると、エラーが発生した場合には、エラーフラグのセットが行われた後 RDRF フラグ (SSR レジスタ bit4) がセットされます。このとき同じ SSR レジスタの RIE ビット (bit1) が "1" にセットされていれば CPU に対して受信割り込みが発生します。SSR レジスタの各フラグを調べ、正常受信なら SIDR レジスタを読み出して、エラーが発生していれば必要な処理を行うようにしてください。

RDRF フラグは、SIDR レジスタを読み出すとクリアされます。

● 送信動作

SSR レジスタの TDRE フラグ (bit3) が "1" のとき、SODR レジスタに送信データを書き込みます。ここで、SCR レジスタの TXE ビット (bit0) が "1" なら送信が行われます。

SODR レジスタにセットしたデータが送信用シフトレジスタにロードされて送信が開始されると TDRE フラグが再びセットされ、次の送信データをセットできるようになります。このとき、同じ SSR レジスタの TIE ビット (bit0) が "1" にセットされていれば CPU に対して送信割り込みが発生して、SODR レジスタに送信データをセットするように要求します。

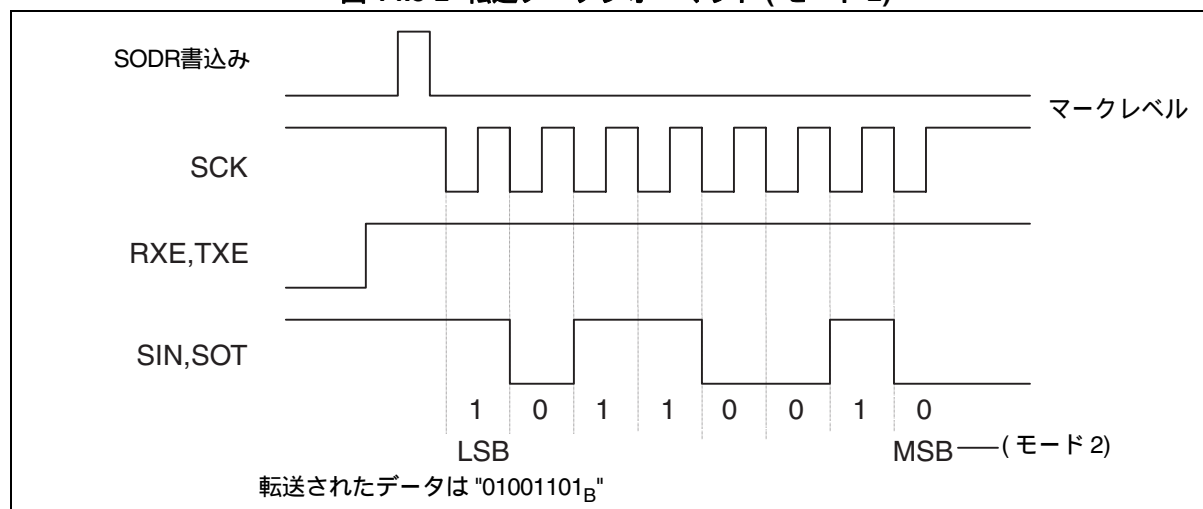
TDRE フラグは、SODR レジスタにデータをセットするといったんクリアされます。

■ クロック同期モード

● 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 14.3-2 に送受信クロックとデータとの関係を示します。

図 14.3-2 転送データフォーマット (モード 2)



内部クロック (U-TIMER) を選択している場合は、データを送信するとデータ受信用同期クロックが自動的に生成されます。

また、外部クロックを選択している場合は、送信側 UART の送信用データバッファ SODR レジスタにデータがあること (TDRE フラグが "0") を確かめた後、正確に 1 バイト分のクロックを供給する必要があります。また、送信開始前と終了後は必ずマークレベルにしてください。

データ長は 8 ビットのみとなり、パリティを付加することはできません。また、スタート/ストップビットがないのでオーバランエラー以外のエラー検出は行われません。

● 初期化

クロック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。

SMR レジスタ

MD1, MD0 : "10_B"

CS0 : クロック入力を指定

SCKE : 内部タイマの場合 "1", 外部クロックの場合 "0"

SCR レジスタ

PEN : "0"

P, SBL, A/D : これらのビットは意味を持ちません

CL : "1"

REC : "0" (初期化するため)

RXE, TXE : 少なくとも、どちらか一方を "1"

SSR レジスタ

RIE : 割込みを使用する場合 "1", 割込みを使用しない場合 "0"

TIE : "0"

● 通信開始

SODR レジスタへの書き込みによって通信を開始します。受信のみの場合でも、必ず仮の送信データを SODR レジスタに書き込む必要があります。

● 通信終了

SSR レジスタの RDRF フラグが "1" に変化したことにより確認できます。SSR レジスタの ORE ビットによって、通信が正常に行われたかを判断してください。

■ 割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 つの割込み要因を持ちます。

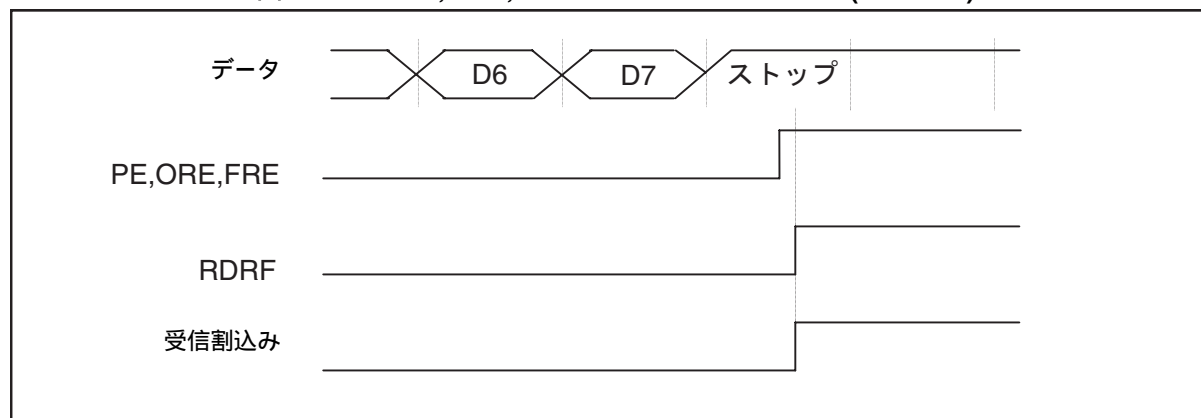
5 つのフラグは、PE/ORE/FRE/RDRF/TDRE です。PE はパリティエラー、ORE はオーバーランエラー、FRE はフレーミングエラーのことで、受信時エラーが発生したときにセットされ、SCR レジスタの REC に "0" を書き込むとクリアされます。RDRF は受信データが SIDR レジスタにロードされるとセットされ、SIDR レジスタを読み出すことでクリアされます。ただし、モード 1 ではパリティ検出機能、モード 2 ではパリティ検出機能とフレーミングエラー検出機能はありません。TDRE は、SODR レジスタが空になり書き込み可能な状態になるとセットされ、SODR レジスタへ書き込むとクリアされます。

2 つの割込み要因は、受信用のものと送信用のものです。受信時は、PE/ORE/FRE/RDRF により割込みを要求します。送信時は、TDRE により割込みを要求します。各動作モードによる割込みフラグのセットタイミングを以下に示します。

● モード 0 の受信動作時

PE, ORE, FRE, RDRF は受信転送が終了し最後のストップビットを検出するときにフラグがセットされ、CPU への割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。

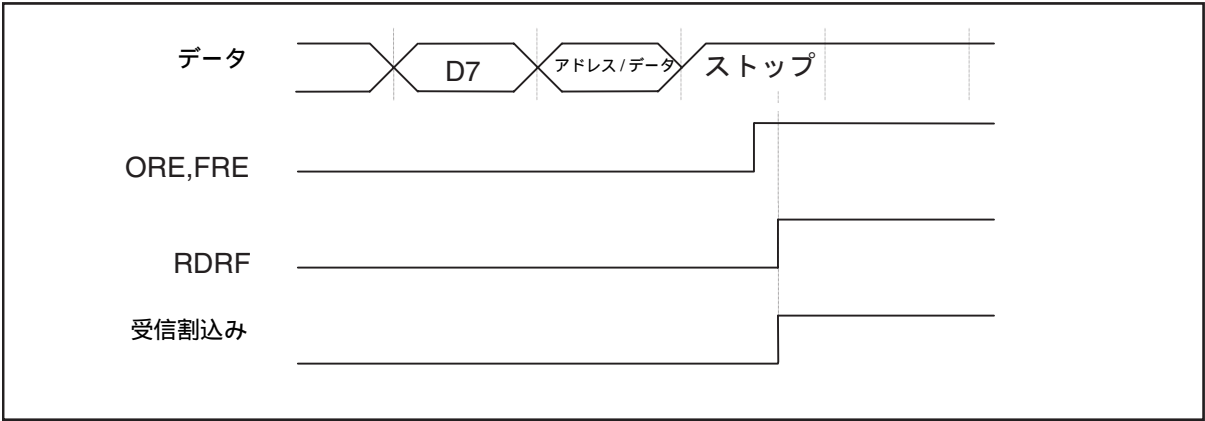
図 14.3-3 ORE, FRE, RDRF のセットタイミング (モード 0)



● モード 1 の受信動作時

ORE, FRE, RDRF は受信転送が終了し、最後のストップビットを検出するときにフラグがセットされ、CPU への割込み要求が発生します。また、受信可能なデータ長は 8 ビットのため、最後の 9 ビット目のアドレス / データを示すデータは無効データとなります。ORE, FRE がアクティブ時は、SIDR のデータは無効データとなります。

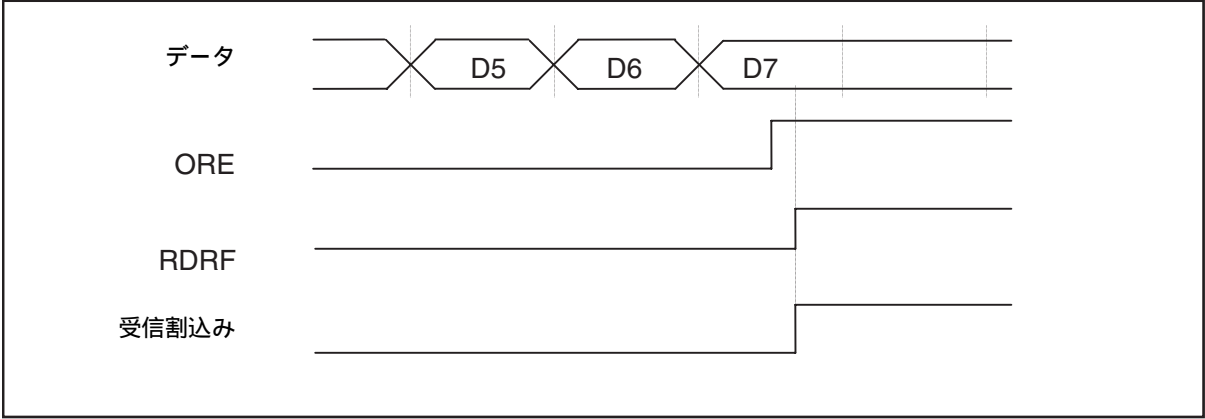
図 14.3-4 ORE, FRE, RDRF のセットタイミング (モード 1)



● モード 2 の受信動作時

ORE, RDRF は受信転送が終了し、最後のデータ (D7) を検出するときにフラグがセットされ、CPU への割込み要求が発生します。ORE がアクティブ時は、SIDR のデータは無効データとなります。

図 14.3-5 ORE, RDRF のセットタイミング (モード 2)



● モード 0, モード 1, モード 2 の送信動作時

TDRE は SODR レジスタへ書き込まれるとクリアされ、内部のシフトレジスタに転送され、次のデータ書込みが可能な状態になるとセットされ、CPU への割込み要求が発生します。送信動作中に SCR レジスタの TXE に "0" (モード 2 のときは RXE も含む) を書き込むと、SSR レジスタの TDRE が "1" となり、送信用のシフトレジスタが停止した後 UART の送信動作を禁止します。送信動作中に SCR レジスタの TXE に "0" (モード 2 のときは RXE も含む) を書き込んだ後、送信が停止する前に SODR レジスタへ書き込まれたデータは送信されます。

図 14.3-6 TDRE のセットタイミング (モード 0, モード 1)

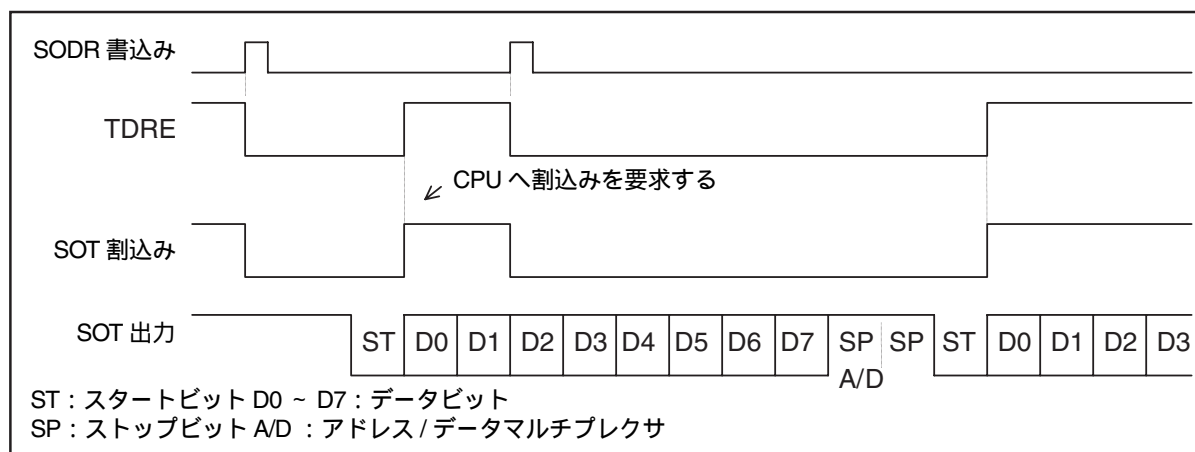
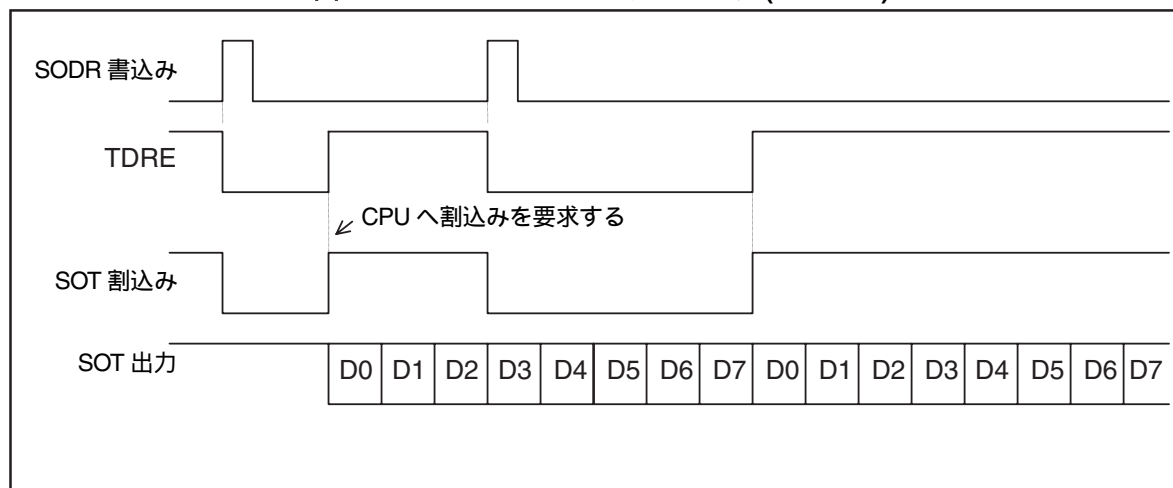


図 14.3-7 TDRE のセットタイミング (モード 2)



■ 使用上の注意

SODR レジスタへの書込みによって通信を開始します。受信のみの場合でも、必ず仮の送信データを SODR レジスタに書き込む必要があります。

通信モードの設定は、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。

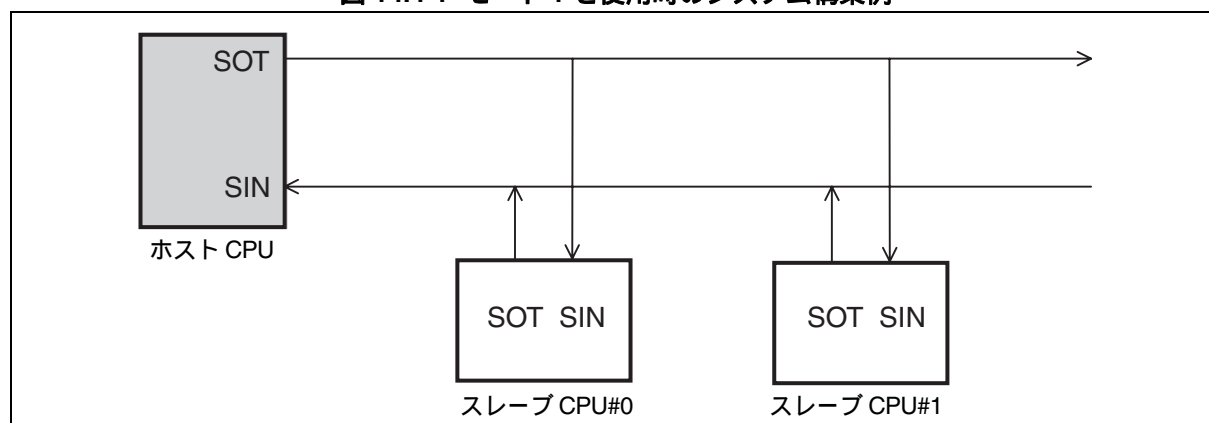
14.4 応用例

UART の応用例を示します。UART のモード 1 における応用例について説明します。

■ 応用例

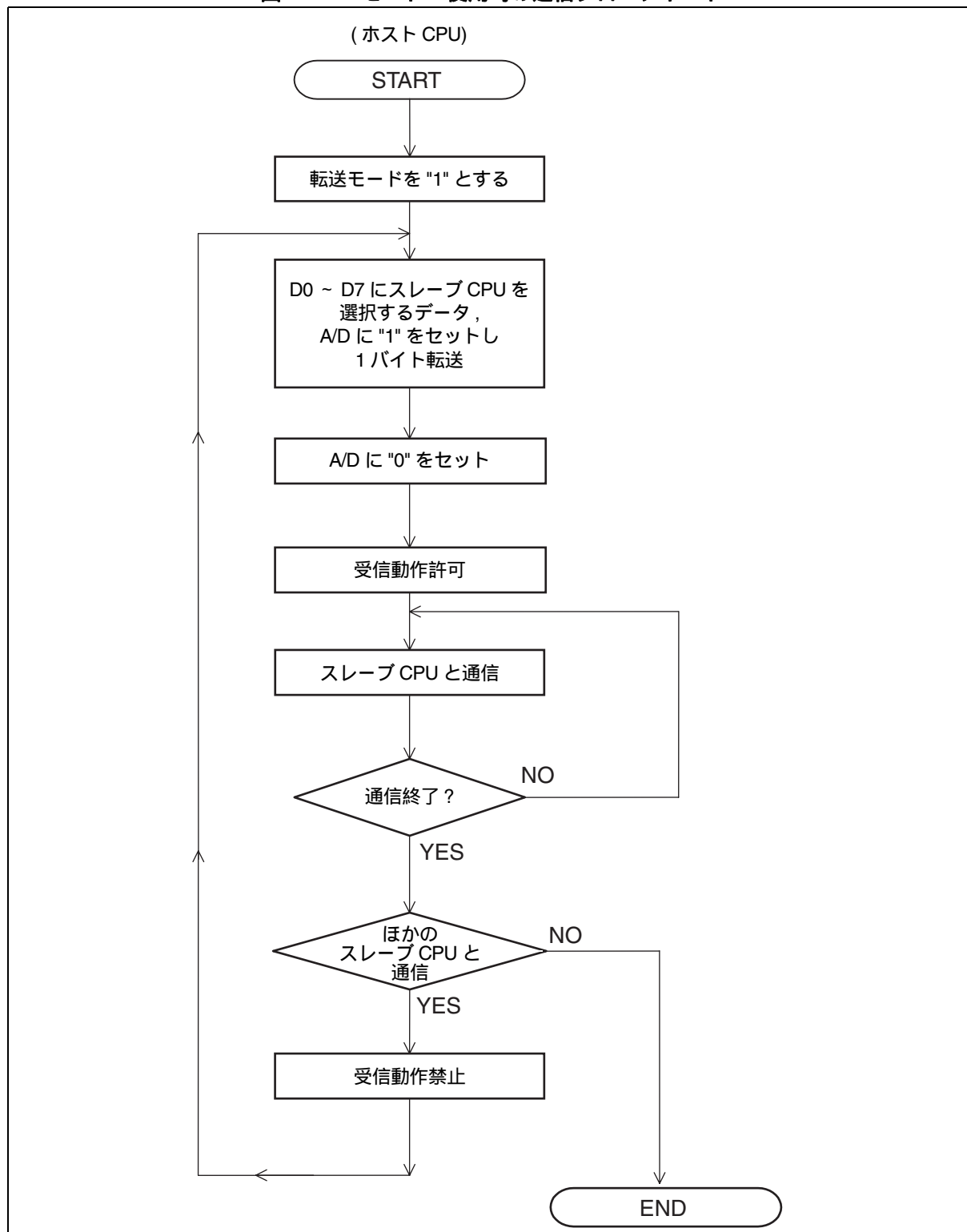
モード 1 は、1 台のホスト CPU に数台のスレーブ CPU が接続されるような場合に使用されます (図 14.4-1 を参照)。このリソースでは、ホスト側の通信インタフェースのみサポートします。

図 14.4-1 モード 1 を使用時のシステム構築例



通信はホスト CPU がアドレスデータを転送することによって始まります。アドレスデータとは、SCR レジスタの A/D が "1" のときのデータで、それにより通信先となるスレーブ CPU が選択され、ホスト CPU との通信が可能になります。通常データは、SCR レジスタの A/D が "0" のときのデータです。図 14.4-2 にそのフローチャートを示します。このモードにおいては、パリティチェック機能は使用できませんので SCR レジスタの PEN ビットは "0" としてください。

図 14.4-2 モード 1 使用時の通信フローチャート



14.5 ボーレートと U-TIMER のリロード値の設定例

ボーレートと U-TIMER のリロード値の設定例を示します。

表中の周波数は、周辺系マシンのクロック周波数を表します。また、UCC1 は、U-TIMER の UTIMC レジスタ UCC1 ビットに設定する値です。

表中の " - " 部は、誤差が $\pm 1\%$ を超えてしまうため使用できないことを示します。

表 14.5-1 非同期 (調歩同期) モード

ボーレート	μs	33 MHz	20 MHz	16.5 MHz	10 MHz
1200	833.33	858(UCC1=0)	520(UCC1=0)	428(UCC1=1)	259(UCC1=1)
2400	416.67	428(UCC1=1)	259(UCC1=1)	214(UCC1=0)	129(UCC1=0)
4800	208.33	214(UCC1=0)	129(UCC1=0)	106(UCC1=0)	64(UCC1=0)
9600	104.17	106(UCC1=1)	64(UCC1=0)	52(UCC1=1)	31(UCC1=1)
19200	52.08	52(UCC1=1)	31(UCC1=1)	26(UCC1=0)	
38400	26.04	26(UCC1=0)		12(UCC1=1)	
57600	17.36	17(UCC1=0)		8(UCC1=0)	
10400	96.15	98(UCC1=0)	59(UCC1=0)	48(UCC1=1)	29(UCC1=0)
31250	32.00	32(UCC1=0)	19(UCC1=0)	15(UCC1=1)	9(UCC1=0)
62500	16.00	15(UCC1=1)	9(UCC1=0)		4(UCC1=0)

表 14.5-2 クロック同期モード

ボーレート	μs	33 MHz	20 MHz	16.5 MHz	10 MHz
250K	4.00	65(UCC1=0)	39(UCC1=0)	32(UCC1=0)	19(UCC1=0)
500K	2.00	32(UCC1=0)	19(UCC1=0)	15(UCC1=1)	9(UCC1=0)
1M	1.00	15(UCC1=1)	9(UCC1=0)	7(UCC1=0)*	4(UCC1=0)

*: $\pm 1\%$ 以上の誤差あり

第15章

C-CAN

C-CAN の機能と動作について説明します。

15.1 C-CAN の特長

15.2 C-CAN のブロックダイアグラム

15.3 C-CAN のレジスタ

15.4 C-CAN レジスタ機能

15.5 C-CAN 機能

15.1 C-CAN の特長

C-CAN は、シリアル通信用の標準プロトコルである CAN プロトコル ver2.0A/B に準拠しており、自動車や FA などの工業分野に広く使用されています。

■ C-CAN の特長

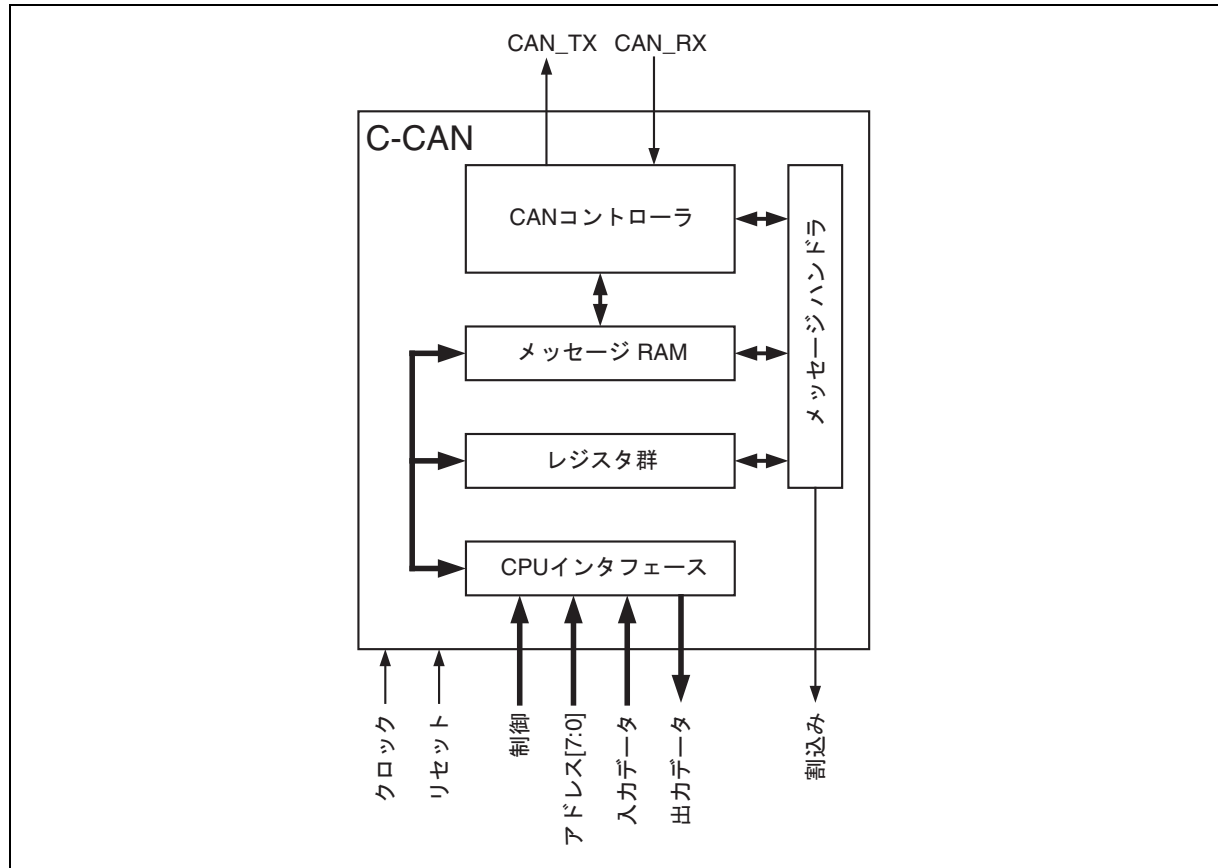
C-CAN には以下の特長があります。

- CAN プロトコル ver2.0A/B をサポート
- 1Mbps までのビットレートをサポート
- メッセージオブジェクトごとの識別マスク
- プログラマブル FIFO モードをサポート
- マスク可能な割込み
- 自己テスト動作用プログラマブルループバックモードをサポート
- インタフェースレジスタを使用してメッセージバッファへの読出し / 書込み

15.2 C-CAN のブロックダイアグラム

図 15.2-1 に、C-CAN のブロックダイアグラムを示します。

図 15.2-1 C-CAN ブロックダイアグラム



■ CAN コントローラ

CAN プロトコルと送受信メッセージ転送のためのシリアル / パラレル変換用のシリアルレジスタを制御します。

■ メッセージ RAM

メッセージオブジェクトを格納します。

■ レジスタ群

C-CAN で使用されるすべてのレジスタです。

■ メッセージハンドラ

メッセージ RAM と CAN コントローラを制御します。

■ CPU インタフェース

FR 内部バスのインタフェースを制御します。

15.3 C-CAN のレジスタ

C-CAN には、以下のレジスタがあります。

- CAN 制御レジスタ (CTRLR)
 - CAN ステータスレジスタ (STATR)
 - CAN エラーカウンタ (ERRCNT)
 - CAN ビットタイミングレジスタ (BTR)
 - CAN 割込みレジスタ (INTR)
 - CAN テストレジスタ (TESTR)
 - CAN プリスケアラ拡張レジスタ (BRPER)
 - IFx コマンド要求レジスタ (IFxCREQ)
 - IFx コマンドマスクレジスタ (IFxCMSK)
 - IFx マスクレジスタ 1, 2(IFxMSK1, IFxMSK2)
 - IFx アービトレーション 1, 2(IFxARB1, IFxARB2)
 - IFx メッセージ制御レジスタ (IFxMCTR)
 - IFx データレジスタ A1, A2, B1, B2(IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)
 - CAN 送信要求レジスタ 1, 2(TREQR1, TREQR2)
 - CAN New Data レジスタ 1, 2(NEWDT1, NEWDT2)
 - CAN 割込みペンディングレジスタ 1, 2(INTPND1, INTPND2)
 - CAN メッセージ有効レジスタ 1, 2(MSGVAL1, MSGVAL2)
 - CAN クロックプリスケアラレジスタ (CANPRE)
-

■ 全体コントロールレジスタ一覧

表 15.3-1 全体コントロールレジスタ一覧

アドレス	レジスタ				備考
	+0	+1	+2	+3	
Base-addr + 00 _H	CAN 制御レジスタ (CTRLR)		CAN ステータスレジスタ (STATR)		STAR: BOff, EWarn,EPass= 読出しのみ RxOk,TxOk,LEC= 読出し / 書込み
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	予約ビット	「15.4.1.1 CAN 制御レジスタ (CTRLR)」参照	予約ビット	「15.4.1.2 CAN ステータスレジスタ (STATR)」参照	
	Reset: 00 _H	Reset: 01 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 04 _H	CAN エラーカウンタ (ERRCNT)		CAN ビットタイミングレジスタ (BTR)		ERRCNT: 読出しのみ BTR: Init(CTRLR)=CCE (CTRLR)=1 時書込み可能
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	RP, REC[6:0]	TEC[7:0]	TSeg2[2:0], TSeg1[3:0]	SJW[1:0], BRP[5:0]	
	Reset: 00 _H	Reset: 00 _H	Reset: 23 _H	Reset: 01 _H	
Base-addr + 08 _H	CAN 割込みレジスタ (INTR)		CAN テストレジスタ (TESTR)		INTR: 読出しのみ TESTR: Test(CTRLR)=1 時, 書込み可能. "Rx" は CAN_RX 端子のレベル値を示す
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	IntId[15:8]	IntId[7:0]	予約ビット	「15.4.1.6 CAN テストレジスタ (TESTR)」参照	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H & 0br0000000	
Base-addr + 0C _H	CAN プリスケール拡張レジスタ (BRPER)		予約ビット		BRP: CCE(CTRLR)=1 時書込み可能
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	予約ビット	BRP[3:0]	-	-	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	

■ メッセージインタフェースレジスタ一覧

表 15.3-2 メッセージインタフェースレジスタ一覧 (1 / 3)

アドレス	レジスタ				備考
	+0	+1	+2	+3	
Base-addr + 10	IF 1 コマンド要求レジスタ (IF1CREQ)		IF1 コマンドマスクレジスタ (IF1CMSK)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Busy	Mess. No. [5:0]	予約ビット	「15.4.2.2 IFx コマンドマスクレジスタ (IFxCMSK)」参照	
	Reset: 00 _H	Reset: 01 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 14 _H	IF1 マスクレジスタ 2 (IF1CMSK2)		IF1 マスクレジスタ 1 (IF1CMSK1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MXtd. MDir, Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset: FF _H	Reset: FF _H	Reset: FF _H	Reset: FF _H	
Base-addr + 18 _H	IF1 アービトレーションレジスタ 2 (IF1ARB2)		IF1 アービトレーションレジスタ 1 (IF1ARB1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal, Xtd, Dir, ID[28:24]	ID[23:16]	ID[15:8]	ID[7:0]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 1C _H	IF1 メッセージコントロールレジスタ (IF1MCTR)		予約ビット		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	「15.4.2.5 IFx メッセージ制御レジスタ (IFxMCTR)」参照	「15.4.2.5 IFx メッセージ制御レジスタ (IFxMCTR)」参照	-	-	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 20 _H	IF1 データ A レジスタ 1 (IF1DTA1)		IF1 データ A レジスタ 2 (IF1DTA2)		バイト配列順序 : ビッグエンディアン
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 24 _H	IF1 データ B レジスタ 1 (IF1DTB1)		IF1 データ B レジスタ 2 (IF1DTB2)		バイト配列順序 : ビッグエンディアン
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 30 _H	IF1 データ A レジスタ 2 (IF1DTA2)		IF1 データ A レジスタ 1 (IF1DTA1)		バイト配列順序 : リトルエンディアン
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	

表 15.3-2 メッセージインタフェースレジスタ一覧 (2 / 3)

アドレス	レジスタ				備考
	+0	+1	+2	+3	
Base-addr + 34 _H	IF1 データ B レジスタ 2 (IF1DTB2)		IF1 データ B レジスタ 1 (IF1DTB1)		バイト配列順序 : リトルエンディ アン
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 40 _H	IF2 コマンド要求レジスタ (IF2CREQ)		IF2 コマンドマスクレジスタ (IF2CMSK)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Busy	Mess. No. [5:0]	予約ビット	「15.4.2.2 IFx コマ ンドマスクレジ スタ (IFxCMSK)」 参照	
	Reset: 00 _H	Reset: 01 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 44 _H	IF2 マスクレジスタ 2 (IF2CMSK2)		IF2 マスクレジスタ 1 (IF2CMSK1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MXtd. MDir, Msk[28:24]	Msk[23:16]	Msk[15:8]	Msk[7:0]	
	Reset: FF _H	Reset: FF _H	Reset: FF _H	Reset: FF _H	
Base-addr + 48 _H	IF2 アービトレーションレジスタ 2 (IF2ARB2)		IF2 アービトレーションレジスタ 1 (IF2ARB1)		
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal, Xtd, Dir,ID[28:24]	ID[23:16]	ID[15:8]	ID[7:0]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 4C _H	IF2 メッセージコントロールレジスタ (IF2MCTR)		予約ビット		
	bit[15:8]	bit[7:0]	bit[7:0]	bit[15:8]	
	「15.4.2.5 IFx メッ セージ制御レジ スタ (IFxMCTR)」 参照	「15.4.2.5 IFx メッ セージ制御レジ スタ (IFxMCTR)」 参照	-	-	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 50	IF2 データ A レジスタ 1 (IF2DTA1)		IF2 データ A レジスタ 2 (IF2DTA2)		バイト配列順序 : ビッグエンディ アン
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[0]	Data[1]	Data[2]	Data[3]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 54 _H	IF2 データ B レジスタ 1 (IF2DTB1)		IF2 データ B レジスタ 2 (IF2DTB2)		バイト配列順序 : ビッグエンディ アン
	bit[7:0]	bit[15:8]	bit[7:0]	bit[15:8]	
	Data[4]	Data[5]	Data[6]	Data[7]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	

表 15.3-2 メッセージインタフェースレジスタ一覧 (3 / 3)

アドレス	レジスタ				備考
	+0	+1	+2	+3	
Base-addr + 60 _H	IF2 データ A レジスタ 2 (IF2DTA2)		IF2 データ A レジスタ 1 (IF2DTA1)		バイト配列順序 : リトルエンディ アン
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[3]	Data[2]	Data[1]	Data[0]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 64 _H	IF2 データ B レジスタ 2 (IF2DTB2)		IF2 データ B レジスタ 1 (IF2DTB1)		バイト配列順序 : リトルエンディ アン
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	Data[7]	Data[6]	Data[5]	Data[4]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	

■ メッセージハンドラレジスタ一覧

表 15.3-3 メッセージハンドラレジスタ一覧

アドレス	レジスタ				備考
	+0	+1	+2	+3	
Base-addr + 80 _H	CAN 送信要求レジスタ 2 (TREQR2)		CAN 送信要求レジスタ 1 (TREQR1)		INTR1,2: 読出し のみ
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	TxRqst[32:25]	TxRqst[24:17]	TxRqst[16:9]	TxRqst[8:1]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + 90 _H	CAN New Data レジスタ 2 (NEWDT2)		CAN New Data レジスタ 1 (NEWDT1)		NEWDT1,2: 読出し のみ
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	NewDat[32:25]	NewDat[24:17]	NewData[16:9]	NewData[8:1]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + A0 _H	CAN 割込みペンディングレジスタ 2 (INTPND2)		CAN 割込みペンディングレジスタ 1 (INTPND1)		INTPND1,2: 読出し のみ
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	IntPnd[32:25]	IntPnd[24:17]	IntPnd[16:9]	IntPnd[8:1]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	
Base-addr + B0 _H	CAN メッセージ有効レジスタ 2 (MSGVAL2)		CAN メッセージ有効レジスタ 1 (MSGVAL1)		MSGVAL1,2: 読出し のみ
	bit[15:8]	bit[7:0]	bit[15:8]	bit[7:0]	
	MsgVal[32:25]	MsgVal[24:17]	MsgVal[16:9]	MsgVal[8:1]	
	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	Reset: 00 _H	

■ クロックプリスケータレジスタ

表 15.3-4 クロックプリスケータレジスタ

アドレス	レジスタ				備考
	+0	+1	+2	+3	
0001A8 _H	CANPRE	-	-	-	CAN プリスケータ
	bit[3:0]	-	-	-	
	CANPRE[3:0]	-	-	-	
	Reset: 00 _H	-	-	-	

15.4 C-CAN レジスタ機能

CAN レジスタは、256 バイト (64 ワード) のアドレス空間が割り当てられ、バイトもしくはワードのアクセスが可能です。メッセージ RAM への CPU のアクセスは、メッセージインタフェースレジスタを介して行います。CAN レジスタを掲載し、それぞれのレジスタの詳細機能を記載します。

■ 全体コントロールレジスタ

・ CAN 制御レジスタ	(CTRLR)
・ CAN ステータスレジスタ	(STATR)
・ CAN エラーカウンタ	(ERRCNT)
・ CAN ビットタイミングレジスタ	(BTR)
・ CAN 割込みレジスタ	(INTR)
・ CAN テストレジスタ (TESTR)	
・ CAN プリスケーラ拡張レジスタ	(BRPER)

■ メッセージインタフェースレジスタ

・ IFx コマンド要求レジスタ	(IFxCREQ)
・ IFx コマンドマスクレジスタ	(IFxCMSK)
・ IFx マスクレジスタ 1, 2	(IFxMSK1, IFxMSK2)
・ IFx アービトレーションレジスタ 1, 2	(IFxARB1, IFxARB2)
・ IFx メッセージ制御レジスタ	(IFxMCTR)
・ IFx データレジスタ A1, A2, B1, B2	(IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

■ メッセージハンドラレジスタ

・ CAN 送信要求レジスタ 1, 2	(TREQR1, TREQR2)
・ CAN データ更新レジスタ 1, 2	(NEWDT1, NEWDT2)
・ CAN 割込みペンディングレジスタ 1, 2	(INTPND1, INTPND2)
・ CAN メッセージ有効レジスタ 1, 2	(MSGVAL1, MSGVAL2)

■ プリスケーラレジスタ

CAN クロックプリスケーラレジスタ	(CANPRE)
--------------------	----------

15.4.1 全体コントロールレジスタ

全体コントロールレジスタは、CAN プロトコル制御および動作モードを制御し、ステータス情報を提供します。

■ 全体コントロールレジスタ

- ・ CAN 制御レジスタ (CTRLR)
- ・ CAN ステータスレジスタ (STATR)
- ・ CAN エラーカウンタ (ERRCNT)
- ・ CAN ビットタイミングレジスタ (BTR)
- ・ CAN 割込みレジスタ (INTR)
- ・ CAN テストレジスタ (TESTR)
- ・ CAN プリスケアラ拡張レジスタ (BRPER)

15.4.1.1 CAN 制御レジスタ (CTRLR)

本レジスタは、CAN コントローラの動作モードを制御します。

■ CAN 制御レジスタ (CTRLR)

CAN 制御レジスタ (上位)								
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Base+00 _H	-	-	-	-	-	-	-	-
	R	R	R	R	R	R	R	R
初期値 00000000 _B								
CAN 制御レジスタ (下位)								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Base+01 _H	Test	CCE	DAR	-	EIE	SIE	IE	Init
	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
初期値 00000000 _B								
R/W : リード/ライト可能								
R : リードオンリ								

■ レジスタ機能

[bit15 ~ bit8] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

[bit7] Test: テストモード許可ビット

Test	機能
0	通常動作 [初期値]
1	テストモード

[bit6] CCE: ビットタイミングレジスタ書込み許可ビット

CCE	機能
0	CAN ビットタイミングレジスタおよび CAN プリスケール拡張レジスタへの書込みを禁止します。[初期値]
1	CAN ビットタイミングレジスタおよび CAN プリスケール拡張レジスタへの書込みを許可します。Init ビットが "1" のときに有効です。

[bit5] DAR: 自動再送禁止ビット

DAR	機能
0	調停負けまたはエラー検出時のメッセージの自動再送を許可します。 [初期値]
1	自動再送を禁止します。

CAN 仕様 (ISO11898, 6.3.3 リカバリ処理 参照) より, CAN コントローラは調停負けあるいは転送中のエラー検出によりフレームの自動再送を行います。自動再送する場合は, DAR ビットを "0" にリセットします。CAN を Time Triggered CAN(TTCAN, ISO11898-1 参照) 環境で動作させるためには, DAR ビットに "1" を設定する必要があります。

< 注意事項 >

DAR ビットに "1" を設定したモードでは, メッセージオブジェクト (メッセージオブジェクトについては「15.4.3 メッセージオブジェクト」を参照してください) の TxRqst ビットと NewDat ビットの動作が異なります。

- フレーム送信を開始したとき, メッセージオブジェクトの TxRqst が "0" にリセットされますが, NewDat ビットはセットされたままです。
- フレーム送信が正常終了すると NewDat は "0" にリセットされます。

送信が調停負けもしくはエラー検出すると, NewDat はセットされたままです。送信を再開するためには, CPU により TxRqst に "1" を設定する必要があります。

[bit4] - : 予約ビット

予約ビットは, "0" が読み出され, 書込みの場合 "0" を設定してください。

[bit3] EIE: エラー割込みコード許可ビット

EIE	機能
0	CAN ステータスレジスタの BOff または EWarn ビットの変化により, CAN 割込みレジスタへの割込みコードの設定を禁止します。 [初期値]
1	CAN ステータスレジスタの BOff または EWarn ビットの変化により, CAN 割込みレジスタへのステータス割込みコードの設定を許可します。

[bit2] SIE: ステータス割込みコード許可ビット

SIE	機能
0	CAN ステータスレジスタの TxOk,RxOk または LEC ビットの変化により, CAN 割込みレジスタへの割込みコードの設定を禁止します。 [初期値]
1	CAN ステータスレジスタの TxOk,RxOk または LEC ビットの変化により, CAN 割込みレジスタへのステータス割込みコードの設定を許可します。CPU からの書込みによって発生した TxOk, RxOk, LEC ビットの変化は CAN 割込みレジスタには設定されません。

[bit1] IE: 割込み許可ビット

IE	機能
0	割込みの発生を禁止します。[初期値]
1	割込みの発生を許可します。

[bit0] Init: 初期化ビット

Init	機能
0	CAN コントローラ動作可能
1	初期化 [初期値]

< 注意事項 >

- バスオフリカバリシーケンス (CAN 仕様 Rev. 2.0 参照) は, Init ビットの設定 / 解除によって短縮はできません。デバイスがバスオフになると, CAN コントローラ自身が Init ビットを "1" にセットし, すべてのバス動作を停止します。バスオフ状態から Init ビットを "0" にクリアすると, バスアイドルが連続的に 129 回 (11 ビットのレセツシブを 1 回とする) 発生するまでバス動作を停止状態にします。バスオフリカバリシーケンス実行後, エラーカウンタをリセットします。
- CAN ビットタイミングレジスタへの書込みは, Init ビットおよび CCE ビットに "1" を設定してください。
- 低消費電力モード (停止モード, クロックモード) へ遷移する前に Init ビットに "1" を書き込んで CAN コントローラの初期化を行ってください。
- CAN プリスケーラレジスタにより, CAN インタフェースに供給するクロックの分周比を変更する場合は, Init ビットに "1" を設定後に CAN プリスケーラレジスタの変更を行ってください。

15.4.1.2 CAN ステータスレジスタ (STATR)

本レジスタは、CAN ステータスおよび CAN バス状態を表示します。

■ CAN ステータスレジスタ (STATR)

CAN ステータスレジスタ (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+02 _H	-	-	-	-	-	-	-	-	00000000 _B
	R	R	R	R	R	R	R	R	

CAN ステータスレジスタ (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+03 _H	BOff	EWarn	EPass	RxOk	TxOk	LEC			00000000 _B
	R	R	R	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

R : リードオンリ

■ レジスタ機能

[bit15 ~ bit8] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

[bit7] BOff: バスオフビット

BOff	機能
0	CAN コントローラはバスオフ状態でないことを示します。 [初期値]
1	CAN コントローラはバスオフ状態であることを示します。

[bit6] EWarn: ワーニングビット

EWarn	機能
0	送信と受信カウンタがともに 96 未満であることを示します。 [初期値]
1	送信または受信カウンタが 96 以上であることを示します。

[bit5] EPass: エラーパッシブビット

EPass	機能
0	送信と受信カウンタがともに 128 未満 (エラーアクティブ状態) であることを示します。 [初期値]
1	受信カウンタは RP ビット =1, 送信カウンタが 128 以上 (エラーパッシブ状態) であることを示します。

[bit4] RxOk: メッセージ正常受信ビット

RxOk	機能
0	CAN バス上で正常なメッセージ通信が行われていない またはバスアイドル状態であることを示します。 [初期値]
1	CAN バス上で正常なメッセージ通信が行われたことを示します。

[bit3] TxOk: メッセージ正常送信ビット

TxOk	機能
0	バスアイドル状態もしくは正常なメッセージ送信が行われていないことを示します。 [初期値]
1	正常なメッセージ送信が行われたことを示します。

< 注意事項 >

RxOk および TxOk ビットは, CPU によってのみリセットされます。

[bit2 ~ bit0] LEC: ラストエラーコードビット

LEC	状態	機能
0	正常	正常に送信または受信されたことを示します。〔初期値〕
1	Stuff エラー	メッセージ内において 6 ビット以上連続してドミナントまたはレセッシブを検出したことを示します
2	Form エラー	受信されたフレームの固定フォーマット部が間違っ検出したことを示します。
3	Ack エラー	送信メッセージに対し、ほかのノードからアクノリッジされなかったことを示します。
4	Bit1 エラー	調停フィールドを除くメッセージの送信データにおいて、レセッシブを送信したにもかかわらずドミナントを検出したことを示します。
5	Bit0 エラー	メッセージの送信データにおいて、ドミナントを送信したにもかかわらずレセッシブを検出したことを示します。バスリカバリ中には、11 ビットのレセッシブを検出するごとにセットされます。このビットを読み出すことによりバスリカバリシーケンスを監視できます。
6	CRC エラー	受信したメッセージの CRC データと計算した CRC の結果が一致しなかったことを示します。
7	未検出	CPU によって LEC ビットに "7" の書込みを行ってから、LEC 値が "7" を読み出した場合、その期間は送受信しなかったことを示します (バスアイドル状態)。

LEC ビットは CAN バス上で発生した最後のエラーを示すコードを保持します。メッセージがエラーなしで転送 (受信 / 送信) 完了すると "0" にクリアされます。未検出コード "7" は、コード更新をチェックするために CPU によって設定されます。

< 注意事項 >

- ステータス割込みコード (8000_H) は、EIE ビットが "1" のときに BOff または EWarn ビットが変化した場合、もしくは SIE ビットが "1" のときに RxOk, TxOk または LEC ビットが変化した場合、CAN 割込みレジスタに設定されます。
- RxOk, TxOk ビットは CPU の書込みによって更新されますので、CAN コントローラによってセットされた RxOk, TxOk ビットが消えてしまいます。もし RxOk, TxOk ビットを使用する場合には、RxOk または TxOk ビットが "1" にセットされてから (45 × BT) 時間以内にクリアしてください。BT は 1 ビットタイムを示します。
- SIE ビットが "1" の時、LEC ビットの変化による割込みが発生した場合には CAN ステータスレジスタに書き込まないでください。
- EPass ビットの変化あるいは RxOk, TxOk および LEC ビットへの CPU 書込み動作では発生しません。
- BOff ビットまたは EPass ビットが "1" になっても EWarn ビットは "1" にセットされています。
- 本レジスタを読み出すことにより、CAN 割込みレジスタのステータス割込み (8000_H) はクリアされます。

15.4.1.3 CAN エラーカウンタ (ERRCNT)

本レジスタは、受信エラーパッシブ表示および受信エラーカウンタ、送信エラーカウンタを示します。

■ CAN エラーカウンタ (ERRCNT)

CAN エラーカウンタレジスタ (上位)								
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Base+04 _H	RP	REC6 ~ REC0						初期値
	R	R	R	R	R	R	R	00000000 _B
CAN エラーカウンタレジスタ (下位)								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Base+05 _H	TEC7 ~ TEC0							初期値
	R	R	R	R	R	R	R	00000000 _B
R : リードオンリ								

■ レジスタ機能

[bit15] RP: 受信エラーパッシブ表示

RP	機能
0	受信エラーカウンタはエラーパッシブ状態でないことを示します。 [初期値]
1	受信エラーカウンタは CAN 仕様で定義されているエラーパッシブ状態に到達したことを示します。

[bit14 ~ bit8] REC6 ~ REC0: 受信エラーカウンタ

受信エラーカウンタ値。受信エラーカウンタ値の範囲は 0 ~ 127 です。

[bit7 ~ bit0] TEC7 ~ TEC0: 送信エラーカウンタ

送信エラーカウンタ値。送信エラーカウンタ値の範囲は 0 ~ 255 です。

15.4.1.4 CAN ビットタイミングレジスタ (BTR)

本レジスタは、プリスケアラおよびビットタイミングを設定します。

■ CAN ビットタイミングレジスタ (BTR)

CAN ビットタイミングレジスタ (上位)								
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Base+06 _H	-	TSeg2			TSeg1			
	R	R/W	R/W	R/W	R	R	R	R
初期値								
00100011 _B								
CAN ビットタイミングレジスタ (下位)								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Base+07 _H	SJW		BRP					
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初期値								
00000001 _B								
R/W : リード/ライト可能								
R : リードオンリ								

■ レジスタ機能

[bit15] - : 予約ビット

予約ビットは, "0" が読み出され, 書込みの場合 "0" を設定してください。

[bit14 ~ bit12] TSeg2: タイムセグメント 2 設定ビット

有効設定値は 0 ~ 7 です。TSeg2+1 の値がタイムセグメント 2 になります。

タイムセグメント 2 は, CAN 仕様のフェーズバッファセグメント (PHASE_SEG2) に相当します。

[bit11 ~ bit8] TSeg1: タイムセグメント 1 設定ビット

有効設定値は 1 ~ 15 です。"0" の設定は禁止です。TSeg1+1 の値がタイムセグメント 1 になります。

タイムセグメント 1 は, CAN 仕様のプロパゲーションセグメント (PROP_SEG) + フェーズバッファセグメント 1 (PHASE_SEG1) に相当します。

[bit7, bit6] SJW: 再同期化ジャンプ幅設定ビット

有効設定値は 0 ~ 3 です。SJW+1 の値が再同期ジャンプ幅となります。

[bit5 ~ bit0] BRP: ボーレートプリスケアラ設定ビット

有効設定値は 0 ~ 63 です。BRP+1 の値がボーレートプリスケアラになります。

システムクロック (f_{sys}) を分周して, CAN コントローラの基本単位時間 (t_q) を決定します。

< 注意事項 >

CAN 制御レジスタの CCE ビットと Init ビットが "1" にセットされているときに, CAN ビットタイミングレジスタおよび CAN プリスケアラ拡張レジスタを設定してください。

15.4.1.5 CAN 割込みレジスタ (INTR)

本レジスタは、メッセージ割込みコードおよびステータス割込みコードを表示します。

■ CAN 割込みレジスタ (INTR)

CAN 割込みレジスタ (上位)								
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Base+08 _H	IntId15 ~ IntId8							初期値 00000000 _B
	R	R	R	R	R	R	R	R
CAN 割込みレジスタ (下位)								
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Base+09 _H	IntId7 ~ IntId0							初期値 00000000 _B
	R	R	R	R	R	R	R	R
R : リードオンリ								

■ レジスタ機能

IntId	機能
0000 _H	割込みなし
0001 _H ~ 0020 _H	割込み要因はメッセージオブジェクトの番号を示します (メッセージ割込みコード)
0021 _H ~ 7FFF _H	未使用
8000 _H	CAN ステータスレジスタの変化による割込みを示します (ステータス割込みコード)
8001 _H ~ FFFF _H	未使用

複数の割込みコードが保留中である場合、CAN 割込みレジスタは優先度の高い割込みコードを示します。割込みコードが CAN 割込みレジスタにセットされていても優先度の高い割込みコードが発生した場合には、CAN 割込みレジスタは優先度の高い割込みコードに更新されます。

優先度の高い割込みコードは、ステータス割込みコード (8000_H)、メッセージ割込み (0001_H, 0002_H, 0003_H,, 0020_H) の順になります。

IntId ビットが 0000_H 以外で、CAN 制御レジスタの IE ビットが "1" にセットされると、CPU への割込み信号がアクティブになります。IntId の値が 0000_H になる (割込み要因がリセットされる) もしくは CAN 制御レジスタの IE ビットが "0" にリセットされると、割込み信号はインアクティブになります。

対象となるメッセージオブジェクト (メッセージオブジェクトについては「15.4.3 メッセージオブジェクト」を参照してください) の IntPnd ビットを "0" にクリアすることでメッセージ割込みコードはクリアされます。

ステータス割込みコードは CAN ステータスレジスタを読み出すことでクリアされます。

15.4.1.6 CAN テストレジスタ (TESTR)

本レジスタは、テストモードの設定および RXO 端子のモニタを行います。動作については、「15.5.7 テストモード」を参照してください。

■ CAN テストレジスタ (TESTR)

CAN テストレジスタ (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+0A _H	-	-	-	-	-	-	-	-	00000000 _B
	R	R	R	R	R	R	R	R	

CAN テストレジスタ (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+0B _H	Rx	Tx1	Tx0	LBack	Silent	Basic	-	-	00000000 _B
	R	R/W	R/W	R/W	R/W	R/W	R	R	

R/W : リード/ライト可能

R : リードオンリ

bit7 の Rx の初期値 R は、CAN バス上のレベルが表示されます。

■ レジスタ機能

[bit15 ~ bit8] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

[bit7] Rx: RXO 端子モニタビット

Rx	機能
0	CAN バスはドミナントであることを示します。
1	CAN バスはレセッシブであることを示します。

[bit6, bit5] Tx1, Tx0: TXO 端子コントロールビット

Tx1-0	機能
00	通常動作 [初期値]
01	サンプリングポイントが Tx 端子に出力されます。
10	TX 端子にドミナントを出力します。
11	TX 端子にレセッシブを出力します。

[bit4] LBack: ループバックモード

LBack	機能
0	ループバックモードを禁止します。 [初期値]
1	ループバックモードを許可します。

[bit3] Silent: サイレントモード

Silent	機能
0	サイレントモードを禁止します。 [初期値]
1	サイレントモードを許可します。

[bit2] Basic: ベーシックモード

Basic	機能
0	ベーシックモードを禁止します。 [初期値]
1	ベーシックモードを許可します。 IF1 レジスタは送信メッセージとして、IF2 レジスタは受信メッセージとして使用されます。

[bit1 ~ bit0] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

< 注意事項 >

- CAN 制御レジスタの Test ビットを "1" にセットした後、本レジスタへ書き込んでください。テストモードが有効となるのは、CAN 制御レジスタの Test ビットが "1" のときです。途中で CAN 制御レジスタの Test ビットを "0" にするとテストモードから通常モードになります。
- Tx ビットを "00_B" 以外に設定した場合、メッセージを送信することができません。

15.4.1.7 CAN プリスケーラ拡張レジスタ (BRPER)

本レジスタは、CAN ビットタイミングで設定したプリスケーラと組み合わせることにより、CAN コントローラで使用するプリスケーラを拡張します。

■ CAN プリスケーラ拡張レジスタ (BRPER)

CAN プリスケーラ拡張レジスタ (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+0C _H	-	-	-	-	-	-	-	-	00000000 _B
	R	R	R	R	R	R	R	R	

CAN プリスケーラ拡張レジスタ (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+0D _H	-	-	-	-	BRPE				00000000 _B
	R	R	R	R	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

R : リードオンリ

■ レジスタ機能

[bit15 ~ bit4] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

[bit3 ~ bit0] BRPE: ポーレートプリスケーラ拡張ビット

CAN ビットタイミングレジスタの BRP と BRPE を組み合わせることにより、1023 までポーレートプリスケーラを拡張できます。

{BRPE(MSB:4 ビット),BRP(LSB:6 ビット)} + 1 の値が CAN コントローラのプリスケーラとなります。

15.4.2 メッセージインタフェースレジスタ

CPU からメッセージ RAM へのアクセスを制御するために 2 組のメッセージインタフェースレジスタを提供します。

メッセージ RAM への CPU アクセスを制御するために使用される 2 組のメッセージインタフェースレジスタがあります。この 2 組のレジスタは、転送された (する) データ (メッセージオブジェクト) をバッファすることで、メッセージ RAM への CPU アクセスと CAN コントローラからのアクセスの競合を回避します。メッセージオブジェクト (メッセージオブジェクトについては、「15.4.3 メッセージオブジェクト」を参照してください) は、メッセージインタフェースレジスタとメッセージ RAM 間を一度に転送します。

テストベシックモードを除き、2 組のメッセージインタフェースレジスタの機能は同一で、独立して動作可能です。例えば、IF1 のメッセージインタフェースレジスタをメッセージ RAM への書き込み動作中に、IF2 のメッセージインタフェースレジスタをメッセージ RAM からの読出しに使用することも可能です。表 15.4-1 に 2 組のメッセージインタフェースレジスタを示します。

メッセージインタフェースレジスタは、コマンドレジスタ (コマンド要求、コマンドマスクレジスタ) と、このコマンドレジスタによって制御されるメッセージバッファレジスタ (マスク、アービトレーション、メッセージ制御、データレジスタ) から構成されます。コマンドマスクレジスタは、データ転送の方向とメッセージオブジェクトのどの部分が転送されるのかを示します。コマンド要求レジスタは、メッセージ番号の選択と、コマンドマスクレジスタに設定された動作を行います。

表 15.4-1 IF1, IF2 メッセージインタフェースレジスタ

アドレス	IF1 レジスタセット	アドレス	IF2 レジスタセット
Base + 10 _H	IF1 コマンド要求	Base + 40 _H	IF2 コマンド要求
Base + 12 _H	IF1 コマンドマスク	Base + 42 _H	IF2 コマンドマスク
Base + 14 _H	IF1 マスク 2	Base + 44 _H	IF2 マスク 2
Base + 16 _H	IF1 マスク 1	Base + 46 _H	IF2 マスク 1
Base + 18 _H	IF1 アービトレーション 2	Base + 48 _H	IF2 アービトレーション 2
Base + 1A _H	IF1 アービトレーション 1	Base + 4A _H	IF2 アービトレーション 1
Base + 1C _H	IF1 メッセージ制御	Base + 4C _H	IF2 メッセージ制御
Base + 20 _H	IF1 データ A1	Base + 50 _H	IF2 データ A1
Base + 22 _H	IF1 データ A2	Base + 52 _H	IF2 データ A2
Base + 24 _H	IF1 データ B1	Base + 54 _H	IF2 データ B1
Base + 26 _H	IF1 データ B2	Base + 56 _H	IF2 データ B2

15.4.2.1 IFx コマンド要求レジスタ (IFxCREQ)

本レジスタは、メッセージ RAM のメッセージ番号の選択とメッセージ RAM とメッセージバッファレジスタ間の転送を行います。また、テストのベーシックモードでは、IF1 を送信制御用に IF2 を受信制御用として使用します。

■ IFx コマンド要求レジスタ (IFxCREQ)

IFx コマンド要求レジスタ (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+10 _H	BUSY	-	-	-	-	-	-	-	00000000 _B
Base+40 _H	R/W	R	R	R	R	R	R	R	

IFx コマンド要求レジスタ (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+11 _H	-	-	Message Number						00000000 _B
Base+41 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能

R : リードオンリ

■ レジスタ機能

IFx コマンド要求レジスタへメッセージ番号を書き込むとすぐにメッセージ RAM とメッセージバッファレジスタ (マスク、アービトレーション、メッセージ制御、データレジスタ) とのメッセージ転送が開始されます。この書込み動作で、BUSY ビットが "1" にセットされ転送処理中であることを示します。その転送が終了すると、BUSY ビットが "0" にリセットされます。

BUSY ビットが "1" のときに、CPU からメッセージインタフェースレジスタへアクセスが発生すると、BUSY ビットが "0" になるまで (コマンド要求レジスタ書き込み後、Clock で 3 ~ 6 サイクル期間)、CPU をウェイトさせます。

テストのベーシックモードでは、BUSY ビットの使用方法が異なります。IF1 コマンド要求レジスタは、送信メッセージとして使用され、BUSY ビットに "1" をセットすることによりメッセージ送信開始を指示します。メッセージ転送が正常終了すると、BUSY ビットは "0" にリセットされます。また、BUSY ビットを "0" にリセットすることにより、いつでもメッセージ転送を中断させることが可能です。

IF2 コマンド要求レジスタは、受信メッセージとして使用され、BUSY ビットを "1" にセットすることにより、受信したメッセージを IF2 メッセージインタフェースレジスタに格納します。

[bit15] BUSY: ビジーフラグビット

- テストベーシックモード以外

BUSY	機能
0	メッセージインタフェースレジスタとメッセージ RAM 間でデータ転送処理を行っていないことを示します。 [初期値]
1	メッセージインタフェースレジスタとメッセージ RAM 間でデータ転送処理中であることを示します。

- テストベーシックモード

IF1 コマンド要求レジスタ

BUSY	機能
0	メッセージ送信を禁止します。
1	メッセージ送信を許可します。

IF2 コマンド要求レジスタ

BUSY	機能
0	メッセージ受信を禁止します。
1	メッセージ受信を許可します。

[bit14 ~ bit6] - : 予約ビット

予約ビットは, "0" が読み出され, 書込みの場合 "0" を設定してください。

[bit5 ~ bit0] Message Number: メッセージ番号 (32 message buffer CAN 用)

Message Number	機能
00 _H	設定禁止です。設定した場合, 20 _H として解釈され, 20 _H が読み出されます。
01 _H ~ 20 _H	処理を行うメッセージ番号を設定します。
21 _H ~ 3F _H	設定禁止です。設定した場合, 01 _H ~ 1F _H として解釈され, 解釈された値が読み出されます。

< 注意事項 >

BUSY ビットは, 読出し / 書込みが可能です。テストのベーシックモード時以外は, このビットに何を書いても動作に影響しません (ベーシックモードについては「15.5.7 テストモード」を参照してください)。

15.4.2.2 IFx コマンドマスクレジスタ (IFxCMSK)

本レジスタは、メッセージインタフェースレジスタとメッセージ RAM 間の転送方向を制御し、どのデータを更新するかを設定します。また、テストのベーシックモードでは本レジスタは無効となります。

■ IFx コマンドマスクレジスタ (IFxCMSK)

IFx コマンドマスクレジスタ (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+12 _H	-	-	-	-	-	-	-	-	00000000 _B
Base+42 _H	R	R	R	R	R	R	R	R	

IFx コマンドマスクレジスタ (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+13 _H	WR/RD	Mask	Arb	Control	CIP	TxRqst/ NewDat	Data A	Data B	00000000 _B
Base+43 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

R : リードオンリ

■ レジスタ機能

[bit15 ~ bit8] - : 予約ビット

予約ビットは、"0" が読み出され、書込みの場合 "0" を設定してください。

[bit7] WR/RD: ライト / リード制御ビット

WR/RD	機能
0	メッセージ RAM からデータを読み出すことを示します。メッセージ RAM からの読出しは IFx コマンド要求レジスタへの書込みによって実行されます。メッセージ RAM から読み出すデータは、Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, Data B ビットの設定に依存します。 [初期値]
1	メッセージ RAM へデータを書き込むことを示します。メッセージ RAM への書込みは IFx コマンド要求レジスタへの書込みによって実行されます。メッセージ RAM への書込みデータは、Mask, Arb, Control, CIP, TxRqst/NewDat, Data A, Data B ビットの設定に依存します。

< 注意事項 >

リセット後、メッセージ RAM のデータは不定です。メッセージ RAM のデータが不定状態でメッセージ RAM のデータを読み出すことは禁止です。

IFx コマンドマスクレジスタの bit6 ~ bit0 は、転送方向 (WR/RD ビット) の設定により違った意味になります。

● 転送方向が書込みの場合 (WR/RD=1)

[bit6] Mask: マスクデータ更新ビット

Mask	機能
0	メッセージオブジェクト* のマスクデータ (ID マスク + MDir + MXtd) を更新しないことを示します。 [初期値]
1	メッセージオブジェクト* のマスクデータ (ID マスク + MDir + MXtd) を更新することを示します。

[bit5] Arb: アービトレーションデータ更新ビット

Arb	機能
0	メッセージオブジェクト* のアービトレーションデータ (ID + Dir + Xtd + MsgVal) を更新しないことを示します。 [初期値]
1	メッセージオブジェクト* のアービトレーションデータ (ID + Dir + Xtd + MsgVal) を更新することを示します。

[bit4] Control: 制御データ更新ビット

Control	機能
0	メッセージオブジェクト* の制御データ (IFx メッセージ制御レジスタ) を更新しないことを示します。 [初期値]
1	メッセージオブジェクト* の制御データ (IFx メッセージ制御レジスタ) を更新することを示します。

[bit3] CIP: 割込みクリアビット

本ビットに "0" または "1" を設定しても CAN コントローラへの動作に影響を与えません。

[bit2] TxRqst/NewDat: メッセージ送信要求ビット

TxRqst/NewDat	機能
0	メッセージオブジェクト* および CAN 送信要求レジスタの TxRqst ビットに "0" を設定することを示します。 [初期値]
1	メッセージオブジェクト* および CAN 送信要求レジスタの TxRqst ビットに "1" を設定 (送信要求) することを示します。

*: 「15.4.3 メッセージオブジェクト」を参照してください。

[bit1] Data A: データ 0 ~ 3 更新ビット

Data A	機能
0	メッセージオブジェクト* のデータ 0 ~ 3 を更新しないことを示します。 [初期値]
1	メッセージオブジェクト* のデータ 0 ~ 3 を更新することを示します。

[bit0] Data B: データ 4 ~ 7 更新ビット

Data B	機能
0	メッセージオブジェクト* のデータ 4 ~ 7 を更新しないことを示します。 [初期値]
1	メッセージオブジェクト* のデータ 4 ~ 7 を更新することを示します

*: 「15.4.3 メッセージオブジェクト」を参照してください。

< 注意事項 >

- IFx コマンドマスクレジスタの TxRqst/NewDat ビットが "1" に設定されると、IFx メッセージ制御レジスタの TxRqst ビットの設定は無効となります。
- テストのベーシックモードでは本レジスタは無効となります。

● 転送方向が読出しの場合 (WR/RD=0)

[bit6] Mask: マスクデータ更新ビット

Mask	機能
0	メッセージオブジェクト* から IFx マスクレジスタ 1,2 ヘデータ (ID マスク + MDir + MXtd) を転送しないことを示します。 [初期値]
1	メッセージオブジェクト* から IFx マスクレジスタ 1,2 ヘデータ (ID マスク + MDir + MXtd) を転送することを示します。

[bit5] Arb: アービトレーションデータ更新ビット

Arb	機能
0	メッセージオブジェクト* から IFx アービトレーション 1,2 ヘデータ (ID+ Dir + Xtd + MsgVal) を転送しないことを示します。 [初期値]
1	メッセージオブジェクト* から IFx アービトレーション 1,2 ヘデータ (ID+ Dir + Xtd + MsgVal) を転送することを示します。

[bit4] Control: 制御データ更新ビット

Control	機能
0	メッセージオブジェクト* から IFx メッセージ制御レジスタヘデータを転送しないことを示します。 [初期値]
1	メッセージオブジェクト* から IFx メッセージ制御レジスタヘデータを転送することを示します。

[bit3] CIP: 割込みクリアビット

CIP	機能
0	メッセージオブジェクト* および CAN 割込みペンディングレジスタの IntPnd ビットを保持することを示します。 [初期値]
1	メッセージオブジェクト* および CAN 割込みペンディングレジスタの IntPnd ビットを "0" にクリアすることを示します。

[bit2] TxRqst/NewDat: データ更新ビット

TxRqst/NewDat	機能
0	メッセージオブジェクト* および CAN データ更新レジスタの NewDat ビットを保持することを示します。 [初期値]
1	メッセージオブジェクト* および CAN データ更新レジスタの NewDat ビットを "0" にクリアすることを示します。

*: 「15.4.3 メッセージオブジェクト」を参照してください。

[bit1] Data A: データ 0 ~ 3 更新ビット

Data A	機能
0	メッセージオブジェクト* および CAN データレジスタ A1, A2 のデータを保持することを示します。 [初期値]
1	メッセージオブジェクト* および CAN データレジスタ A1, A2 のデータを更新することを示します。

[bit0] Data B: データ 4 ~ 7 更新ビット

Data B	機能
0	メッセージオブジェクト* および CAN データレジスタ B1, B2 のデータを保持することを示します。 [初期値]
1	メッセージオブジェクト* および CAN データレジスタ B1, B2 のデータを更新することを示します。

* : 「15.4.3 メッセージオブジェクト」を参照してください。

< 注意事項 >

- メッセージオブジェクトへの読出しアクセスにより, IntPnd および NewDat ビットは "0" にリセットすることが可能です。ただし, IFx メッセージ制御レジスタの IntPnd および NewDat ビットには, 読出しアクセスによりリセットされる前の IntPnd, NewDat ビットが格納されます。
- テストのベーシックモードでは無効となります。

15.4.2.3 IFx マスクレジスタ 1,2 (IFxMSK1, IFxMSK2)

本レジスタは、メッセージ RAM のメッセージオブジェクトマスクデータを読み出し / 書き込みするために用いられます。また、テストのベーシックモードでは、設定されているマスクデータは無効となります。
各ビットの機能については、「15.4.3 メッセージオブジェクト」に記述されています。

■ IFx マスクレジスタ 1,2 (IFxMSK1, IFxMSK2)

IFx マスクレジスタ 2 (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+14 _H	MXtd	MDir	-	Msk28 ~ Msk24					11111111 _B
Base+44 _H	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	

IFx マスクレジスタ 2 (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+15 _H	Msk23 ~ Msk16								11111111 _B
Base+45 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IFx マスクレジスタ 1 (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+16 _H	Msk15 ~ Msk8								11111111 _B
Base+46 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IFx マスクレジスタ 1 (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+17 _H	Msk7 ~ Msk0								11111111 _B
Base+47 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

R : リードオンリ

本レジスタのビット説明については「15.4.3 メッセージオブジェクト」を参照してください。

本レジスタの予約ビット (IFx マスクレジスタ 2 の bit13) は "1" が読み出され、書き込み時は "1" を書き込んでください。

15.4.2.4 IFx アービトレーションレジスタ 1,2 (IFxARB1, IFxARB2)

本レジスタは、メッセージ RAM のメッセージオブジェクトアービトレーションデータを読み出し / 書込みするために用いられます。また、テストのベーシックモードでは無効となります。

各ビットの機能については、「15.4.3 メッセージオブジェクト」に記述されています。

■ IFx アービトレーションレジスタ 1,2 (IFxARB1, IFxARB2)

IFx アービトレーションレジスタ 2 (上位バイト)									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+18 _H	MsgVal	Xtd	Dir	ID28 ~ ID24					00000000 _B
Base+48 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
IFx アービトレーションレジスタ 2 (下位バイト)									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+19 _H	ID23 ~ ID16								00000000 _B
Base+49 _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
IFx アービトレーションレジスタ 1 (上位バイト)									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+1A _H	ID15 ~ ID8								00000000 _B
Base+4A _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
IFx アービトレーションレジスタ 1 (下位バイト)									
アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+1B _H	ID7 ~ ID0								00000000 _B
Base+4B _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

本レジスタのビット説明については「15.4.3 メッセージオブジェクト」を参照してください。

< 注意事項 >

送信途中でメッセージオブジェクトの MsgVal ビットを "0" にクリアした場合、送信が完了した時点で CAN ステータスレジスタの TxOk ビットは "1" になりますが、メッセージオブジェクトおよび CAN 送信要求レジスタの TxRqst ビットは "0" にクリアされませんのでメッセージインタフェースレジスタによって TxRqst ビットを "0" にクリアしてください。

15.4.2.5 IFx メッセージ制御レジスタ (IFxMCTR)

本レジスタは、メッセージ RAM のメッセージオブジェクト制御データを読み出し/書き込みするために用いられます。また、テストのベーシックモードでは、IF1 メッセージ制御レジスタは無効となります。IF2 メッセージ制御レジスタの NewDat と MsgLst は通常の動作を行い、DLC ビットは受信したメッセージの DLC を表示します。その他の制御ビットは無効 ("0") として動作します。各ビットの機能については、「15.4.3 メッセージオブジェクト」に記述されています。

■ IFx メッセージ制御レジスタ (IFxMCTR)

IFx メッセージ制御レジスタ (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+1C _H	NewDat	MsgLst	IntPnd	UMask	TxE	RxE	RmtEn	TxRqst	00000000 _B
Base+4C _H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

IFx メッセージ制御レジスタ (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+1D _H	EoB	-	-	-	DLC3 ~ DLC0				00000000 _B
Base+4D _H	R/W	R	R	R	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能

R : リードオンリ

本レジスタのビット説明については「15.4.3 メッセージオブジェクト」を参照してください。

< 注意事項 >

TxRqst, NewDat および IntPnd ビットは、IFx コマンドマスクレジスタの WR/RD ビットの設定によって以下の動作になります。

- 転送方向が書き込みの場合 (IFx コマンドマスクレジスタ : WR/RD=1)
 - IFx コマンドマスクレジスタの TxRqst/NewDat が "0" に設定されている場合のみ、本レジスタの TxRqst ビットが有効となります。
- 転送方向が読み出しの場合 (IFx コマンドマスクレジスタ : WR/RD=0)
 - IFx コマンドマスクレジスタの CIP ビットを "1", IFx コマンド要求レジスタへの書き込みによりメッセージオブジェクトおよび CAN 割込みペンディングレジスタの IntPnd ビットをリセットした場合、本レジスタにはリセットされる前の IntPnd ビットが格納されます。
 - IFx コマンドマスクレジスタの TxRqst/NewDat ビットを "1", IFx コマンド要求レジスタへの書き込みによりメッセージオブジェクトおよび CAN データ更新レジスタの NewDat ビットをリセットした場合、本レジスタにはリセットする前の NewDat ビットが格納されます。

15.4.2.6 IFx データレジスタ A1,A2,B1,B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

本レジスタは、メッセージ RAM のメッセージオブジェクト送受信データを読み出し / 書き込みするために用いられます。データフレームの送受信のみ使用され、リモートフレームの送受信には使用されません。

■ IFx データレジスタ A1,A2,B1,B2 (IFxDTA1, IFxDTA2, IFxDTB1, IFxDTB2)

	addr+0	addr+1	addr+2	addr+3
IFx Message Data A1 (アドレス 20 _H & 50 _H)	Data(0)	Data(1)	–	–
IFx Message Data A2 (アドレス 22 _H & 52 _H)	–	–	Data(2)	Data(3)
IFx Message Data B1 (アドレス 24 _H & 54 _H)	Data(4)	Data(5)	–	–
IFx Message Data B2 (アドレス 26 _H & 56 _H)	–	–	Data(6)	Data(7)
IFx Message Data A2 (アドレス 30 _H & 60 _H)	Data(3)	Data(2)	–	–
IFx Message Data A1 (アドレス 32 _H & 62 _H)	–	–	Data(1)	Data(0)
IFx Message Data B2 (アドレス 34 _H & 64 _H)	Data(7)	Data(6)	–	–
IFx Message Data B1 (アドレス 36 _H & 66 _H)	–	–	Data(5)	Data(4)

図 15.4-1 IFx データレジスタ

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
	Data								00000000 _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									

■ レジスタ機能

● 送信メッセージデータの設定

設定したデータは , MSB (bit7,bit15) より開始して Data(0), Data(1), ..., Data(7) の順で送信されます。

● 受信メッセージデータ

受信メッセージデータは , MSB (bit7,bit15) より開始して Data(0), Data(1), ..., Data(7) の順で格納されます。

< 注意事項 >

- 受信メッセージデータが8バイトより少ない場合は, データレジスタの残りのバイトには不定データが書き込まれます。
 - メッセージオブジェクトへの転送は, Data A もしくは Data B の 4 バイト単位で行われますので, 4 バイトのうちある一部のデータだけを更新することはできません。
-

15.4.3 メッセージオブジェクト

メッセージ RAM には、32 のメッセージオブジェクトがあります。メッセージ RAM への CPU アクセスと CAN コントローラからのアクセスの競合を回避するために、CPU はメッセージオブジェクトへ直接、アクセスはできません。これらのアクセスは、IFx メッセージインタフェースレジスタ経由で行います。本節では、メッセージオブジェクトの構成と機能を説明します。

■ メッセージオブジェクトの構成

メッセージオブジェクト

UMask	Msk28 ~ Msk0	MXtd	MDir	EoB	New Dat		MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID28 ~ ID0	Xtd	Dir	DLC3 ~ DLC0	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7

< 注意事項 >

メッセージオブジェクトは、CAN 制御レジスタの Init ビット、ハードウェアリセットでは初期化されません。ハードウェアリセットの場合、ハードウェアリセット解除後、CPU によってメッセージ RAM を初期化するかメッセージ RAM の MsgVal を "0" にしてください。

■ メッセージオブジェクトの機能

ID28 ~ ID0, Xtd, Dir ビットは、メッセージを送信する場合、ID とメッセージの種類に使用されます。メッセージを受信する場合、Msk28 ~ Msk0, MXtd, MDir ビットとともに受容フィルタで使用されます。

受容フィルタを通過したデータフレームもしくはリモートフレームは、メッセージオブジェクトに格納されます。Xtd は拡張フレームか標準フレームかを示し、Xtd が "1" の場合、29 ビット ID (拡張フレーム) を、Xtd が "0" の場合は、11 ビット ID (標準フレーム) を受信することになります。

受信したデータフレームもしくはリモートフレームが 1 つ以上のメッセージオブジェクトと一致した場合は、一致した最小のメッセージ番号に格納されます。詳細は、「15.5.3 メッセージ受信動作」の受信メッセージの受容フィルタを参照してください。

MsgVal: 有効メッセージビット

MsgVal	機能
0	メッセージオブジェクトは無効です。 メッセージの送受信は行いません。
1	メッセージオブジェクトは有効です。 メッセージの送受信が可能となります。

< 注意事項 >

- CAN 制御レジスタ (CTRLR) の Init ビットを "0" にリセットする前と, ID28 ~ ID0, Xtd, Dir, DLC3 ~ DLC0 を変更する前には, メッセージオブジェクトの MsgVal ビットを必ず初期化してください。
- 送信中に MsgVal ビットを "0" にすると送信が完了した時点で CAN ステータスレジスタ (STATR) の TxOk ビットは "1" になりますが, メッセージオブジェクトおよび CAN 送信要求レジスタ (TREQR) の TxRqst ビットは "0" にクリアされませんのでメッセージインタフェースレジスタによって TxRqst ビットを "0" にクリアしてください。

UMask: 受容マスク許可ビット

UMask	機能
0	Msk28 ~ Msk0, MXtd, MDir を使用しません。
1	Msk28 ~ Msk0, MXtd, MDir を使用します。

< 注意事項 >

- CAN 制御レジスタの Init ビットが "1" のとき, あるいは MsgVal ビットが "0" のときに UMask ビットを変更してください。
- Dir ビットが "1" かつ RmtEn ビットが "0" のとき, UMask の設定により動作が異なります。
 - UMask が "1" の場合は, 受容フィルタを通過しリモートフレームを受信したとき, TxRqst ビットを "0" にリセットします。そのとき, 受信した ID, IDE, RTR, DLC はメッセージオブジェクトに格納し, NewDat ビットは "1" にセットされ, データは変更しません (データフレームのように取り扱います)。
 - UMask が "0" の場合は, リモートフレーム受信に対し, TxRqst ビットはそのまま保持し, リモートフレームを無視します。

ID28 ~ ID0: メッセージ ID

ID	機能
ID28 ~ ID0	29 ビット ID (拡張フレーム) を指示します。
ID28 ~ ID18	11 ビット ID (標準フレーム) を指示します。

Msk28 ~ Msk0: ID マスク

Msk	機能
0	メッセージオブジェクトの ID と対応するビットをマスクします。
1	メッセージオブジェクトの ID と対応するビットをマスクしません。

Xtd: 拡張 ID 許可ビット

Xtd	機能
0	メッセージオブジェクトは 11 ビット ID (標準フレーム) が使用されます。
1	メッセージオブジェクトは 29 ビット ID (拡張フレーム) が使用されます。

MXtd: 拡張 ID マスクビット

MXtd	機能
0	受容フィルタで拡張 ID ビット (IDE) をマスクします。
1	受容フィルタで拡張 ID ビット (IDE) をマスクしません。

< 注意事項 >

11 ビット ID (標準フレーム) がメッセージオブジェクトに設定されると、受信したデータフレームの ID は、ID28 ~ ID18 へ書き込まれます。ID マスクは、Msk28 ~ Msk18 が使用されます。

Dir: メッセージ方向ビット

Dir	機能
0	受信方向を示します。 TxRqst が "1" にセットされると、リモートフレームの送信を行い、TxRqst が "0" のときは、受容フィルタを通過したデータフレームを受信します。
1	送信方向を示します。 TxRqst が "1" にセットされると、データフレームを送信し、TxRqst が "0" で RmtEn が "1" にセットされている場合、受容フィルタを通過したリモートフレームの受信によって、CAN コントローラ自身が TxRqst を "1" にセットします。

MDir: メッセージ方向マスクビット

MDir	機能
0	受容フィルタでメッセージ方向ビット (Dir) をマスクします。
1	受容フィルタでメッセージ方向ビット (Dir) をマスクしません。

< 注意事項 >

MDir ビットは常に "1" を設定してください。

EoB: エンドオブバッファビット (詳細は , 「 15.5.4 FIFO バッファ機能」 参照)

EoB	機能
0	メッセージオブジェクトは FIFO バッファとして使用され , 最終メッセージでないことを示します。
1	単一メッセージオブジェクトまたは FIFO バッファの最終メッセージオブジェクトを示します。

< 注意事項 >

- EoB ビットは , 2 ~ 32 メッセージの FIFO バッファを構成するために使用します。
- 単一メッセージオブジェクト (FIFO を使用しない場合) は , 必ず EoB ビットに "1" を設定してください。

NewDat: データ更新ビット

NewDat	機能
0	有効なデータがないことを示します。
1	有効なデータがあることを示します。

MsgLst: メッセージロスト

MsgLst	機能
0	メッセージロストは発生していません。
1	メッセージロストが発生しています。

< 注意事項 >

MsgLst ビットは Dir ビットが "0" のとき (受信方向) のみ有効です。

RxIE: 受信割り込みフラグイネーブルビット

RxIE	機能
0	フレーム受信成功後 , IntPnd は変更されません。
1	フレーム受信成功後 , IntPnd が "1" にセットされます。

TxIE: 送信割込みフラグイネーブルビット

TxIE	機能
0	フレーム送信成功後, IntPnd は変更されません。
1	フレーム送信成功後, IntPnd が "1" にセットされます。

IntPnd: 割込みペンディングビット

IntPnd	機能
0	割込み要因がありません。
1	割込み要因があります。 ほかに優先度の高い割込みがない場合は, CAN 割込みレジスタの IntId ビットはこのメッセージオブジェクトを示します。

RmtEn: リモートイネーブル

RmtEn	機能
0	リモートフレームの受信で, TxRqst は変更されません。
1	Dir ビットが "1" でリモートフレームを受信すると, TxRqst が "1" にセットされます。

< 注意事項 >

Dir ビットが "1" かつ RmtEn ビットが "0" のとき, UMask の設定により動作が異なります。

- UMask が "1" の場合は, 受容フィルタを通過しリモートフレームを受信したとき, TxRqst ビットを "0" にリセットします。そのとき, 受信した ID, IDE, RTR, DLC はメッセージオブジェクトに格納し, NewDat ビットは "1" にセットされ, データは変更しません (データフレームのように扱います)。
- UMask が "0" の場合は, リモートフレーム受信に対し, TxRqst ビットはそのまま保持し, リモートフレームを無視します。

TxRqst: 送信要求ビット

TxRqst	機能
0	送信アイドル状態 (送信中でもないし, 送信待ち状態でもない) を示します。
1	送信中または, 送信待ちであることを示します。

DLC3 ~ DLC0: データ長コード

DLC3	機能
0 ~ 8	データフレーム長は 0 ~ 8 バイトです。
9 ~ 15	設定禁止です。 設定された場合は, 8 バイト長となります。

< 注意事項 >

データフレームを受信すると DLC ビットには, 受信した DLC が格納されます。

Data 0 ~ Data 7: データ 0 ~ データ 7

Data	機能
Data 0	CAN データフレームの最初のデータバイト
Data 1	CAN データフレームの 2 番目のデータバイト
Data 2	CAN データフレームの 3 番目のデータバイト
Data 3	CAN データフレームの 4 番目のデータバイト
Data 4	CAN データフレームの 5 番目のデータバイト
Data 5	CAN データフレームの 6 番目のデータバイト
Data 6	CAN データフレームの 7 番目のデータバイト
Data 7	CAN データフレームの 8 番目のデータバイト

< 注意事項 >

- CAN バスへのシリアル出力は, MSB (bit7 もしくは bit15) より出力します。
- 受信メッセージデータが 8 バイトより少ない場合は, データレジスタの残りのバイトには不定が書き込まれます。
- メッセージオブジェクトへの転送は, Data A もしくは Data B の 4 バイト単位で行われますので, 4 バイトのうちある一部のデータだけを更新することはできません。

15.4.4 メッセージハンドラレジスタ

メッセージハンドラレジスタは、すべて読出し専用です。メッセージオブジェクトの TxRqst, NewDat, IntPnd, MsgVal ビットと IntId ビットは、ステータスを表示します。

■ メッセージハンドラレジスタ

- ・ CAN 送信要求レジスタ 1, 2 (TREQR1,TREQR2)
- ・ CAN データ更新レジスタ 1, 2 (NEWDT1,NEWDT2)
- ・ CAN 割込みペンディングレジスタ 1, 2 (INTPND1,INTPND2)
- ・ CAN メッセージ有効レジスタ 1, 2 (MSGVAL1,MSGVAL2)

15.4.4.1 CAN 送信要求レジスタ (TREQR1, TREQR2)

本レジスタは、全メッセージオブジェクトの TxRqst ビットを表示します。TxRqst ビットを読み出すことにより、どのメッセージオブジェクトの送信要求がペンディング中であるかをチェックすることができます。

■ CAN 送信要求レジスタ (TREQR1, TREQR2)

CAN 送信要求レジスタ 2 (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+80 _H	TxRqst32 ~ TxRqst25								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 送信要求レジスタ 2 (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+81 _H	TxRqst24 ~ TxRqst17								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 送信要求レジスタ 1 (上位バイト)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+82 _H	TxRqst16 ~ TxRqst9								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 送信要求レジスタ 1 (下位バイト)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+83 _H	TxRqst8 ~ TxRqst1								00000000 _B
	R	R	R	R	R	R	R	R	

R : リードオンリ

■ レジスタ機能

TxRqst32 ~ TxRqst1: 送信要求ビット

TxRqst	機能
0	送信アイドル状態 (送信中でもないし, 送信待ち状態でもない) を示します。
1	送信中または, 送信待ちであることを示します。

TxRqst ビットのセット / リセット条件を以下に示します。

- セット条件
 - IFx コマンドマスキングレジスタの WR/RD に "1", TxRqst に "1" を設定し, IFx コマンド要求レジスタへの書き込みにより特定オブジェクトの TxRqst にセットできます。
 - IFx コマンドマスキングレジスタの WR/RD に "1", TxRqst に "0", IFx メッセージ制御レジスタの TxRqst に "1" を設定した場合, IFx コマンド要求レジスタへの書き込みにより特定オブジェクトの TxRqst にセットできます。
 - Dir ビットが "1", RmtEn ビットが "1" に設定し, 受容フィルタを通過したリモートフレームの受信によりセットされます。
- リセット条件
 - IFx コマンドマスキングレジスタの WR/RD に "1", TxRqst に "0", IFx メッセージ制御レジスタの TxRqst に "0" を設定した場合, IFx コマンド要求レジスタへの書き込みにより特定オブジェクトの TxRqst をリセットできます。
 - フレームの送信が正常終了すると, リセットされます。
 - Dir が "1", RmtEn が "0", UMask が "1" の場合, 受容フィルタを通過したリモートフレームの受信によりリセットされます。

15.4.4.2 CAN データ更新レジスタ (NEWDT1, NEWDT2)

本レジスタは、全メッセージオブジェクトの NewDat ビットを表示します。NewDat ビットを読み出すことにより、どのメッセージオブジェクトのデータが更新されたかチェックすることができます。

■ CAN データ更新レジスタ (NEWDT1, NEWDT2)

CAN データ更新レジスタ 2 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+90 _H	NewDat32 ~ NewDat25								00000000 _B
	R	R	R	R	R	R	R	R	

CAN データ更新レジスタ 2 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+91 _H	NewDat24 ~ NewDat17								00000000 _B
	R	R	R	R	R	R	R	R	

CAN データ更新レジスタ 1 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+92 _H	NewDat16 ~ NewDat9								00000000 _B
	R	R	R	R	R	R	R	R	

CAN データ更新レジスタ 1 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+93 _H	NewDat8 ~ NewDat1								00000000 _B
	R	R	R	R	R	R	R	R	

R : リードオンリ

■ レジスタ機能

NewDat32 ~ NewDat1: データ更新ビット

NewDat	機能
0	有効なデータがないことを示します。
1	有効なデータがあることを示します。

NewDat ビットのセット / リセット条件を以下に示します。

- セット条件
 - IFx コマンドマスクレジスタの WR/RD に "1", IFx メッセージ制御レジスタの NewDat に "1" を設定して, IFx コマンド要求レジスタの書込みにより特定オブジェクトにセットできます。
 - 受容フィルタを通過したデータフレームの受信によりセットされます。
 - Dir が "1", RmtEn が "0", UMask が "1" の場合, 受容フィルタを通過したリモートフレームの受信によりセットされます。
- リセット条件
 - IFx コマンドマスクレジスタの WR/RD に "0", NewDat に "1" を設定した場合, IFx コマンド要求レジスタの書込みにより特定オブジェクトの NewDat をリセットできます。
 - IFx コマンドマスクレジスタの WR/RD に "1", IFx メッセージ制御レジスタの NewDat を "0" に設定した場合, IFx コマンド要求レジスタの書込みにより特定オブジェクトの NewDat をリセットできます。
 - 送信用シフトレジスタ(内部レジスタ)へデータを転送終了後, リセットされます。

15.4.4.3 CAN 割込みペンディングレジスタ (INTPND1, INTPND2)

本レジスタは、全メッセージオブジェクトの IntPnd ビットを表示します。IntPnd ビットを読み出すことにより、どのメッセージオブジェクトが割込みペンディング中であるかチェックすることができます。

■ CAN 割込みペンディングレジスタ (INTPND1, INTPND2)

CAN 割込みペンディングレジスタ 2 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+A0 _H	IntPnd32 ~ IntPnd25								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 割込みペンディングレジスタ 2 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+A1 _H	IntPnd24 ~ IntPnd17								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 割込みペンディングレジスタ 1 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+A2 _H	IntPnd16 ~ IntPnd9								00000000 _B
	R	R	R	R	R	R	R	R	

CAN 割込みペンディングレジスタ 1 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+A3 _H	IntPnd8 ~ IntPnd1								00000000 _B
	R	R	R	R	R	R	R	R	

R : リードオンリ

■ レジスタ機能

IntPnd32 ~ IntPnd1: 割込みペンディングビット

IntPnd	機能
0	割込み要因がありません。
1	割込み要因があります。

IntPnd ビットのセット / リセット条件を以下に示します。

- セット条件
 - TxIE が "1" にセットされている場合 , フレームの正常送信完了によりセットされます。
 - RxIE が "1" にセットされている場合 , 受容フィルタを通過したフレームの正常受信完了によりセットされます。
- リセット条件
 - IFx コマンドマスキレジスタが WR/RD に "1", IntPnd に "1" を設定した場合 , IFx コマンド要求レジスタの書込みにより特定オブジェクトの IntPnd をリセットできます。

15.4.4.4 CAN メッセージ有効レジスタ (MSGVAL1, MSGVAL2)

本レジスタは、全メッセージオブジェクトの MsgVal ビットを表示します。MsgVal ビットを読み出すことにより、どのメッセージオブジェクトが有効であるかチェックすることができます。

■ CAN メッセージ有効レジスタ (MSGVAL1, MSGVAL2)

CAN メッセージ有効レジスタ 2 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+B0 _H	MsgVal32 ~ MsgVal25								00000000 _B
	R	R	R	R	R	R	R	R	

CAN メッセージ有効レジスタ 2 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+B1 _H	MsgVal24 ~ MsgVal17								00000000 _B
	R	R	R	R	R	R	R	R	

CAN メッセージ有効レジスタ 1 (上位)

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
Base+B2 _H	MsgVal16 ~ MsgVal9								00000000 _B
	R	R	R	R	R	R	R	R	

CAN メッセージ有効レジスタ 1 (下位)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
Base+B3 _H	MsgVal8 ~ MsgVal1								00000000 _B
	R	R	R	R	R	R	R	R	

R : リードオンリ

■ レジスタ機能

MsgVal32 ~ MsgVal1: メッセージ有効ビット

MsgVal	機能
0	メッセージオブジェクトは無効です。 メッセージの送受信は行いません。
1	メッセージオブジェクトは有効です。 メッセージの送受信が可能となります。

MsgVal ビットのセット / リセット条件を以下に示します。

- セット条件
 - IFx アビトレーションレジスタ 2 の MsgVal に "1" を設定し, IFx コマンド要求レジスタへの書込みにより特定オブジェクトの MsgVal をセットできます。
- リセット条件
 - IFx アビトレーションレジスタ 2 の MsgVal に "0" を設定し, IFx コマンド要求レジスタへの書込みにより特定オブジェクトの MsgVal をリセットできます。

15.4.5 CAN クロックプリスケアラレジスタ

本レジスタは、CAN インタフェースに供給するクロックの分周比を定義するものです。本レジスタの値を変更する場合は、CAN 制御レジスタ (CTRLR) の初期化ビット (Init) を "1" にセットし、すべてのバス動作を停止してください。

■ CAN クロックプリスケアラレジスタ

CAN クロックプリスケアラレジスタ									
アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
01A8 _H	-	-	-	-	CANPRE3	CANPRE2	CANPRE1	CANPRE0	00000000 _B
	R	R	R	R	R/W	R/W	R/W	R/W	
R/W : リード / ライト可能									
R : リードオンリ									

■ レジスタ機能

[bit15 ~ bit12] - : 予約ビット

本ビットは "0" が読み出され、書込みはレジスタに反映されません。

[bit11 ~ bit8] CAN クロックプリスケアラ設定ビット

CANPRE[3:0]	機能
0000 _B	CAN クロックとしてシステムクロックの 1/2 周期が選択されます。 (初期値 : CANPRE[3:0]=0000 _B)
0001 _B	
001x _B	CAN クロックとしてシステムクロックの 1/4 周期が選択されます。
01xx _B	CAN クロックとしてシステムクロックの 1/8 周期が選択されます。
1000 _B	CAN クロックとしてシステムクロックの 2/3 周期が選択されます。 クロックの Duty は 67% となります。 [設定禁止]
1001 _B	CAN クロックとしてシステムクロックの 1/3 周期が選択されます。
101x _B	CAN クロックとしてシステムクロックの 1/6 周期が選択されます。
11xx _B	CAN クロックとしてシステムクロックの 1/12 周期が選択されます。

< 注意事項 >

- CAN クロックプリスケアラ設定ビットの変更は、CAN 制御レジスタの初期化ビットを "1" にセットし、すべてのバス動作を停止させた後に行ってください。
- 本レジスタの設定により CAN インタフェースに供給するクロックは 16MHz 以下としてください。
- CANPRE[3:0]=1000_B は、設定禁止です。

15.5 C-CAN 機能

CAN コントローラの動作と機能について説明します。

以下の機能について説明します。

- メッセージオブジェクト
- メッセージ送信動作
- メッセージ受信動作
- FIFO バッファ機能
- 割込み機能
- ビットタイミング
- テストモード
- ソフトウェア初期化
- CAN クロックプリスケアラ

15.5.1 メッセージオブジェクト

メッセージ RAM のメッセージオブジェクトとインタフェースについて説明します。

■ メッセージオブジェクト

メッセージ RAM のメッセージオブジェクト設定 (MsgVal, NewDat, IntPnd, TxRqst ビットを除く) は、ハードウェアリセットによって初期化されません。そのためメッセージオブジェクトを CPU で初期化するか、MsgVal ビットを無効 (MsgVal=0) に設定してください。また、CAN ビットタイミングレジスタの設定は CAN 制御レジスタの Init ビットが "0" のとき行ってください。

メッセージオブジェクトの設定は、メッセージインタフェースレジスタ (IFx マスクレジスタ, IFx アービトレーションレジスタ, IFx メッセージ制御レジスタ, IFx データレジスタ) に設定した後、IFx コマンド要求レジスタへのメッセージ番号の書込みにより、そのインタフェースレジスタのデータが指定されたメッセージオブジェクトへ転送されます。

CAN 制御レジスタの Init ビットが "0" にクリアされると CAN コントローラは動作を開始します。受容フィルタを通過した受信メッセージは、メッセージ RAM へ格納されます。送信要求が保留されているメッセージは、メッセージ RAM から CAN コントローラのシフトレジスタへ転送され、CAN バスへの送信が行われます。

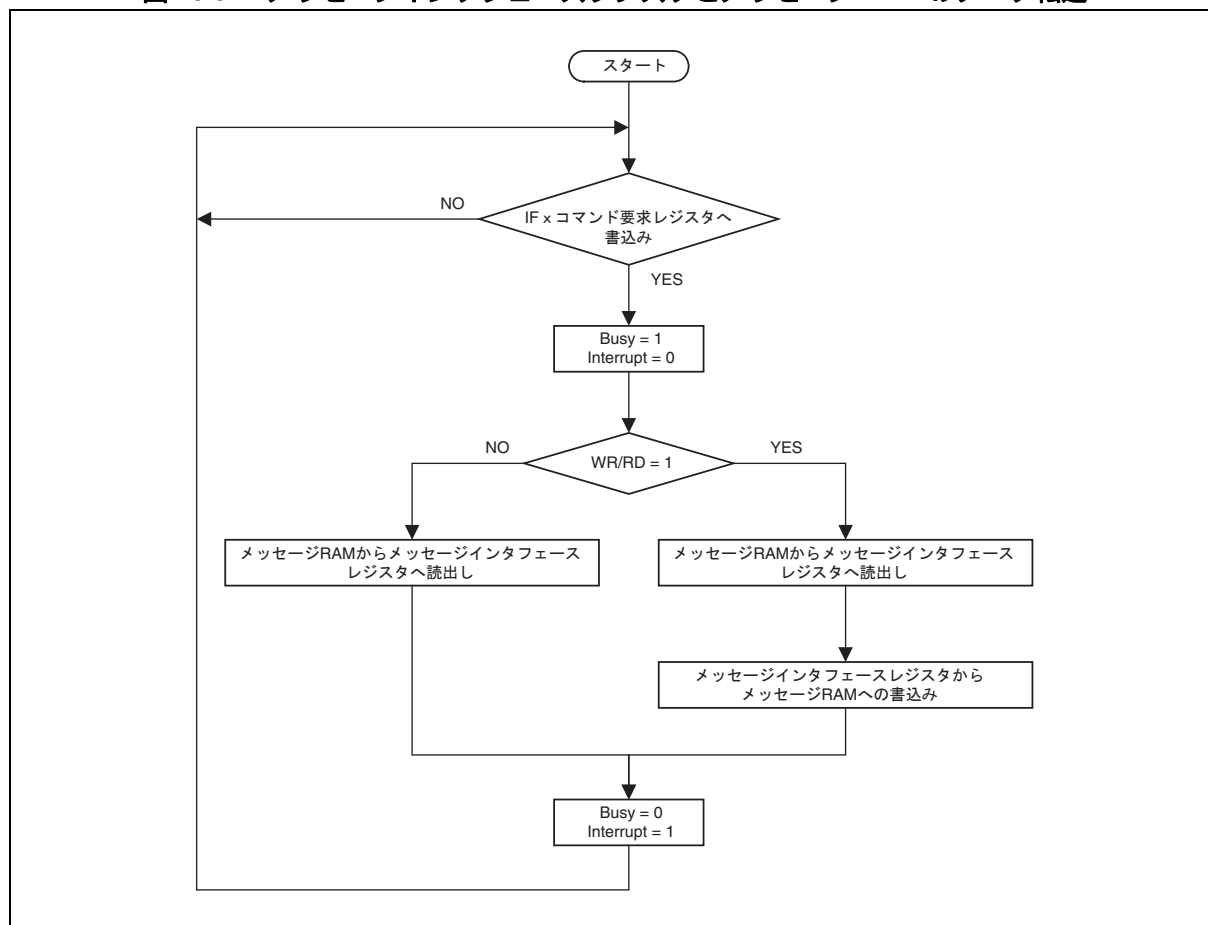
CPU は、メッセージインタフェースレジスタ経由で受信メッセージの読出しおよび、送信メッセージの更新を行います。また、CAN 制御レジスタおよび、IFx メッセージ制御レジスタ (メッセージオブジェクト) の設定に応じて、CPU への割込みが行われます。

■ メッセージ RAM とのデータ送受信

メッセージインタフェースレジスタとメッセージ RAM とのデータ転送を開始すると、IFx コマンド要求レジスタの BUSY ビットを "1" にセットします。転送完了後、BUSY ビットは "0" にクリアされます (図 15.5-1 参照)。

IFx コマンドマスクレジスタは、一つのメッセージオブジェクトの全データ転送か、データの部分転送を行うかを設定します。メッセージ RAM の構造上、メッセージオブジェクトの単一ビット/バイトの書込みは不可能となっており、常に一つのメッセージオブジェクトの全データをメッセージ RAM へ書き込みます。したがって、メッセージインタフェースレジスタからメッセージ RAM へのデータ転送は、リードモディファイライトサイクルを実行しています。

図 15.5-1 メッセージインタフェースレジスタとメッセージ RAM のデータ転送



15.5.2 メッセージ送信動作

送信メッセージオブジェクトの設定方法および送信動作について説明します。

■ メッセージ送信

メッセージインタフェースレジスタとメッセージ RAM 間でデータ転送がない場合、CAN メッセージ有効レジスタの MsgVal ビットと CAN 送信要求レジスタの TxRqst ビットを評価します。送信要求を保留している中で、最高優先度の有効であるメッセージオブジェクトが、送信用のシフトレジスタへ転送が行われます。そのときメッセージオブジェクトの NewDat ビットは "0" にリセットされます。

正常に送信が完了したとき、メッセージオブジェクトに新たなデータがない(NewDat=0)場合は、TxRqst ビットは "0" にリセットされます。TxIE が "1" にセットされている場合は、送信成功後に IntPnd ビットが "1" にセットされます。CAN コントローラが CAN バス上で調停に負けた場合、あるいは転送中にエラーが発生した場合は、CAN バスがアイドルになり次第、直ちにメッセージの再送信が行われます。

■ 送信優先度

メッセージオブジェクトの送信優先度は、メッセージ番号によって決定します。メッセージオブジェクト 1 が最高の優先度で、メッセージオブジェクト 32 (搭載している最大メッセージオブジェクト番号) が最低優先度となります。したがって、2 つ以上の送信要求が保留されていると、対応するメッセージオブジェクトの小さい番号順に転送が行われます。

■ 送信メッセージオブジェクトの設定

表 15.5-1 に送信オブジェクトの初期化方法を示します。

表 15.5-1 送信メッセージオブジェクトの初期化

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

IFx アービトレーションレジスタ (ID28 ~ ID0 と Xtd ビット) は、アプリケーションで与えられ、送信メッセージの ID およびメッセージの種類を定義します。

標準フレーム (11 ビット ID) を設定した場合は、ID28 ~ ID18 を使用し、ID17 ~ ID0 は無効となります。拡張フレーム (29 ビット ID) を設定した場合は、ID28 ~ ID0 を使用します。

TxIE ビットに "1" をセットすると、メッセージオブジェクトの送信成功後に IntPnd ビットが "1" にセットされます。

RmtEn ビットに "1" をセットすると、一致するリモートフレームを受信後、TxRqst ビットに "1" をセットし、データフレームを自動的に送信します。

データレジスタ (DLC3 ~ DLC0, Data7 ~ Data) の設定は、アプリケーションで与えられます。

UMask=1 のとき、IFx マスクレジスタ (Msk28 ~ Msk0, UMask, MXtd, MDir ビット) は、マスク設定によりグループ化された ID を持つリモートフレームを受信し、その後、送信を許可 (TxRqst ビットに "1" をセット) するために使用されます。

詳細は、「15.5.3 メッセージ受信動作」のリモートフレームを参照してください。

< 注意事項 >

IFx マスクレジスタの Dir ビットをマスク許可に設定することは禁止です。

■ 送信メッセージオブジェクトの更新

CPU は、送信メッセージオブジェクトのデータをメッセージインタフェースレジスタ経由で、更新することができます。

送信メッセージオブジェクトのデータは、対応する IFx データレジスタの 4 バイト単位 (IFx データレジスタ A, IFx データレジスタ B 単位) でデータ書込みが行われます。そのため 1 バイトだけ送信メッセージオブジェクトを変更することはできません。

8 バイトのデータのみを更新するときは、まず IFx コマンドマスクレジスタへ 0087_B の書込みを行います。そして、IFx コマンド要求レジスタへメッセージ番号の書込みにより、送信メッセージオブジェクトのデータ更新 (8 バイトデータ) と TxRqst ビットへの "1" 書込みが同時に行われます。

送信中のメッセージ番号に連続して送信させたい場合、TxRqst ビットと NewDat に "1" を設定してください。TxRqst ビットは "0" にリセットされずに連続送信が可能となります。

NewDat ビットが "1", TxRqst ビットがともに "1" の場合、送信を開始すると NewDat ビットは "0" にリセットされます。

< 注意事項 >

- データを更新する場合は、IFx データレジスタ A もしくは IFx データレジスタ B の 4 バイト単位で行ってください。
 - データのみを更新する場合は、NewDat ビットと TxRqst ビットに "1" を設定してください。
-

15.5.3 メッセージ受信動作

受信メッセージオブジェクトの設定方法および受信動作について説明します。

■ 受信メッセージの受容フィルタ

メッセージのアービトレーション/コントロールフィールド (ID + IDE + RTR + DLC) が CAN コントローラ受信用シフトレジスタへ完全にシフトされると、有効メッセージオブジェクトとの一致比較のためにメッセージ RAM のスキャンを開始します。

このとき、メッセージ RAM のメッセージオブジェクトから調停フィールドとマスクデータ (MsgVal, UMask, NewDat, EoB を含む) がロードされ、メッセージオブジェクトとシフトレジスタの調停フィールドがマスクデータを含んで比較されます。

この動作は、“メッセージオブジェクトとシフトレジスタの調停フィールドが一致検出されるまで”、または“メッセージ RAM の最終ワードに到達するまで”、繰り返し実行します。一致が検出されると、メッセージ RAM のスキャンは停止され、受信フレームのタイプ (データフレームまたはリモートフレーム) により、CAN コントローラは処理を行います。

■ 受信優先度

メッセージオブジェクトの受信優先度は、メッセージ番号によって決定します。メッセージオブジェクト 1 が最高の優先度で、メッセージオブジェクト 32 (搭載している最大メッセージオブジェクト番号) が最低優先度となります。したがって、受容フィルタで2つ以上一致した場合は、メッセージ番号の小さい番号が受信メッセージオブジェクトとなります。

■ データフレーム受信

CAN コントローラは、受容フィルタで一致したメッセージオブジェクトのメッセージ RAM へ、シフトレジスタから受信メッセージを転送し格納します。この格納するデータは、データバイトだけではなく、すべてのアービトレーションフィールドおよびデータ長コードも格納します。これは、IFx マスクレジスタがマスク設定されている場合でも実行されます (ID とデータバイトを保持するために格納されます)。

NewDat ビットは、新たなデータが受信されると "1" にセットされます。CPU がメッセージオブジェクトを読み出したときには、NewDat ビットを "0" にリセットしてください。メッセージ受信時に、既に NewDat ビットが "1" にセットされている場合は、その前のデータが失われたことになり、MsgLst が "1" にセットされます。

RxIE ビットが "1" にセットされている場合、メッセージバッファを受信すると CAN 割込みペンディングレジスタの IntPnd ビットに "1" をセットします。そのとき、そのメッセージオブジェクトの TxRqst ビットは "0" にリセットされます。これは、リモートフレーム送信処理中に、要求データフレームを受信した場合、送信処理を防ぐために行われます。

■ リモートフレーム

リモートフレーム受信時の動作は、下記の 3 つの処理があります。一致するメッセージオブジェクトの設定より、リモートフレーム受信時の処理が選択されます。

1) Dir=1 (送信方向), RmtEn=1, UMask=1 または UMask=0

一致したリモートフレームの受信を行い、このメッセージオブジェクトの TxRqst ビットのみ "1" にセットされ、リモートフレームに対するデータフレームの自動返信 (送信) を行います (TxRqst ビット以外のメッセージオブジェクトは変更されません)。

2) Dir=1 (送信方向), RmtEn=0, UMask=0

受信したリモートフレームがメッセージオブジェクトと一致しても受信しないでリモートフレームを無効にします (このメッセージオブジェクトの TxRqst ビットは変更されません)。

3) Dir=1 (送信方向), RmtEn=0, UMask=1

受信したリモートフレームがメッセージオブジェクトと一致した場合、このメッセージオブジェクトの TxRqst ビットは "0" にリセットされ、リモートフレームは受信データフレームのように処理されます。受信したアービトレーションフィールドとコントロールフィールド (ID + IDE + RTR + DLC) は、メッセージ RAM のメッセージオブジェクトへ格納され、このメッセージオブジェクトの NewDat ビットが "1" にセットされます。メッセージオブジェクトのデータフィールドは変更されません。

■ 受信メッセージオブジェクトの設定

表 15.5-2 に受信メッセージオブジェクトの初期化方法を示します。

表 15.5-2 受信メッセージオブジェクトの初期化

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

IFx アービトレーションレジスタ (ID28 ~ ID0, Xtd ビット) は、アプリケーションによって与えられ、受容フィルタに用いられる受信メッセージ ID とメッセージの種類を定義します。

標準フレーム (11 ビット ID) を設定した場合は、ID28 ~ ID18 を使用し、ID17 ~ ID0 は無効となります。また、標準フレームが受信されると、ID17 ~ ID0 は "0" にリセットされます。拡張フレーム (29 ビット ID) を設定した場合は、ID28 ~ ID0 を使用します。

RxIE ビットが "1" にセットされた場合、受信データフレームがメッセージオブジェクトへ格納されると IntPnd ビットが "1" にセットされます。

データ長コード (DLC3 ~ DLC0) は、アプリケーションによって与えられます。CAN コントローラが、受信データフレームをメッセージオブジェクトへ格納するとき、受信データ長コードと 8 バイトのデータを格納します。データ長コードが 8 未満の場合は、メッセージオブジェクトの残りのデータは不定データが書き込まれます。

UMask=1 のとき、IFx マスクレジスタ (Msk28 ~ Msk0, UMask, MXtd, MDir ビット) は、マスク設定によりグループ化された ID を持つデータフレームの受信を許可するために使用します。詳細は、「15.5.3 メッセージ受信動作」のデータフレーム受信を参照してください。

< 注意事項 >

IFx マスクレジスタの Dir ビットのマスク設定は禁止です。

■ 受信メッセージの処理

CPU は、メッセージインタフェースレジスタを介して、受信メッセージをいつでも読み出すことが可能です。

通常、IFx コマンドマスクレジスタに "007F_B" を書き込みます。次にメッセージオブジェクトのメッセージ番号を IFx コマンド要求レジスタに書き込みます。この手順によって、指定されたメッセージ番号の受信メッセージをメッセージ RAM からメッセージインタフェースレジスタに転送します。このとき、IFx コマンドマスクレジスタの設定により、メッセージオブジェクトの NewDat ビットと IntPnd ビットを "0" にクリアすることが可能です。

受信メッセージの処理は、受容フィルタにより一致した場合、メッセージを受信します。メッセージオブジェクトで受容フィルタのマスクを使用している場合は、マスク設定されたデータが受容フィルタから除外され、メッセージを受信するか判断します。

NewDat ビットは、メッセージオブジェクトが最後に読み出されてから、新しいメッセージが受信されたかを示します。

MsgLst ビットは、受信したデータがメッセージオブジェクトから読み出されない状態で次の受信データを受信したために前のデータを失ってしまったことを示します。MsgLst ビットは自動的にリセットされません。

リモートフレーム送信処理中に、受容フィルタにより一致するデータフレームが受信された場合には、TxRqst ビットは自動的に "0" にリセットされます。

15.5.4 FIFO バッファ機能

受信メッセージ処理におけるメッセージオブジェクトの FIFO バッファの構成および動作について説明します。

■ FIFO バッファの構成

FIFO バッファの受信メッセージオブジェクトの構成は、EoB ビットを除いて、受信メッセージオブジェクトの構成と同じです（「15.5.3 メッセージ受信動作」の受信メッセージオブジェクトの設定を参照してください）。

FIFO バッファは、2 つ以上の受信メッセージオブジェクトを連結して使用します。この FIFO バッファへ受信メッセージを格納するためには、受信メッセージオブジェクトの ID とマスクを使用する場合はそれらの設定を一致させなければなりません。

FIFO バッファの最初の受信メッセージオブジェクトは、優先順位の高いメッセージ番号の小さい番号となります。FIFO バッファの最後の受信メッセージオブジェクトは、EoB ビットに "1" をセットし、FIFO バッファブロックの終わりを示す必要があります（FIFO バッファ構成を使用するメッセージオブジェクトの最終メッセージオブジェクト以外は、EoB ビットに "0" を設定してください）。

< 注意事項 >

- FIFO バッファで使用するメッセージオブジェクトの ID とマスク設定は必ず同じ設定にしてください。
- FIFO バッファを使用しない場合は、必ず EoB ビットに "1" を設定してください。

■ FIFO バッファによるメッセージ受信

受信メッセージが、FIFO バッファの ID と一致した場合は、最小メッセージ番号の FIFO バッファの受信メッセージオブジェクトへ格納されます。

FIFO バッファの受信メッセージオブジェクトへメッセージが格納されると、この受信メッセージオブジェクトの NewDat ビットが "1" にセットされます。EoB ビットが "0" の受信メッセージオブジェクトへ NewDat ビットをセットすると、CAN コントローラによる FIFO バッファ書込みは、最後の受信メッセージオブジェクト（EoB ビット = 1）に到達するまで、受信メッセージオブジェクトは保護され、書込みが行われません。

最終FIFOバッファまで有効なデータが格納された状態で受信メッセージオブジェクトの NewDat ビットに "0" 書込み（書込み保護の解除）が行われないと次に受信されたメッセージが最終メッセージオブジェクトへ書き込まれ、メッセージは上書きされてしまいます。

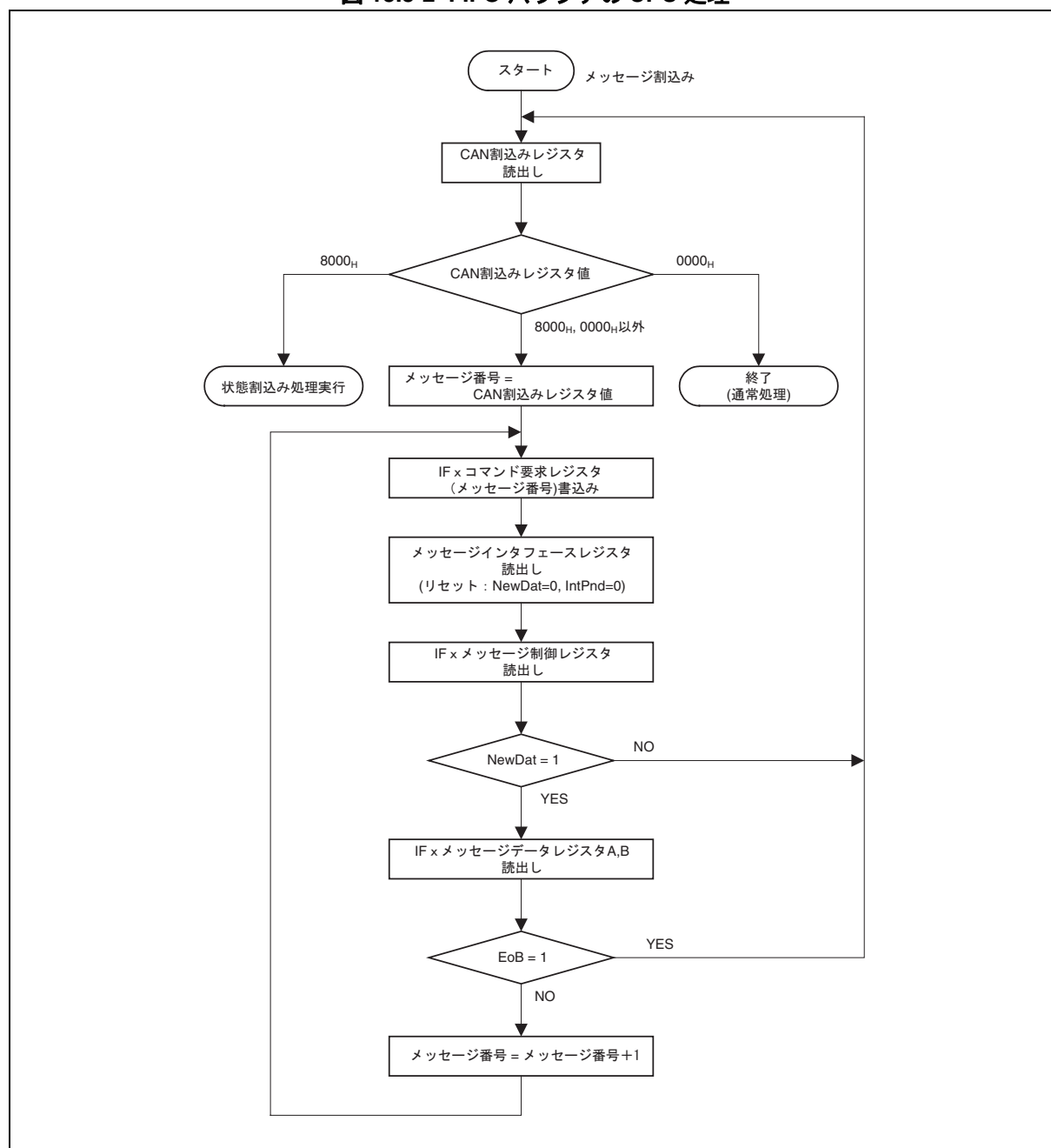
■ FIFO バッファからの読出し

CPU が受信メッセージオブジェクトの内容を読み出すには、IF_x コマンド要求レジスタへ受信メッセージ番号を書き込むことによって、メッセージインタフェースレジスタに転送され読み出すことができます。このとき、IF_x コマンドマスクレジスタの WR/RD を "0" (読出し) および、TxRqst/NewDat=1, IntPnd=1 に設定し、NewDat ビットと IntPnd ビットを "0" にリセットしてください。

FIFO バッファの機能を保証するために、FIFO バッファの受信メッセージオブジェクトは、必ず最小のメッセージ番号から読み出してください。

図 15.5-2 に FIFO バッファで連結されるメッセージオブジェクトの CPU の処理方法を示します。

図 15.5-2 FIFO バッファの CPU 処理



15.5.5 割込み機能

ステータス割込み (IntId=8000_H) およびメッセージ割込み (IntId= メッセージ番号) による割込み処理について説明します。

複数の割込みが保留中である場合、CAN 割込みレジスタは、保留中の最高優先度の割込みコードを示します。割込みコードの設定された時間順は無視され、常に優先順位の高い割込みコードが表示されます。CPU がクリアするまで割込みコードは保持されます。

ステータス割込み (IntId ビットの 8000_H) は、最高優先度となります。

メッセージ割込みの優先度は、メッセージ番号の小さいメッセージが高く、大きいメッセージが低くなります。

メッセージ割込みは、メッセージオブジェクトの IntPnd ビットのクリアによってクリアされます。ステータス割込みは、CAN ステータスレジスタの読出しでクリアされます。

CAN 割込みペンディングレジスタの IntPnd ビットは、割込みの有無を示します。保留中の割込みがないときは、IntPnd ビットは "0" を示します。

CAN 制御レジスタの IE ビットおよび、IFx メッセージ制御レジスタの TxIE ビット、RxIE ビットに "1" をセットしている状態で、IntPnd ビットが "1" になると CPU への割込み信号がアクティブになります。割込み信号は、CAN 割込みペンディングレジスタが "0" にクリアされる（割込み要因リセット）か、CAN 制御レジスタの IE ビットが "0" にリセットされるまでアクティブ状態を保持します。

CAN 割込みレジスタの 8000_H は、CAN コントローラによって CAN ステータスレジスタが更新されたことを示し、この割込みが最高優先度となります。CAN ステータスレジスタの更新による割込みは、CAN 制御レジスタの EIE ビットと SIE ビットにより CAN 割込みレジスタへの設定許可または禁止の制御ができます。また、CPU への割込み信号の制御は、CAN 制御レジスタの IE ビットで行うことができます。

CAN ステータスレジスタの RxOk ビット、TxOk ビット、LEC ビットは、CPU からの書込みにより更新（リセット）することができますが、その書込みにより割込みのセットまたはリセットを行うことはできません。

CAN 割込みレジスタの 8000_H、0000_H 以外は、メッセージ割込みが保留中であることを示し、優先度の高い保留中のメッセージ割込みを示します。

CAN 割込みレジスタは、IE がリセットされた場合でも更新されます。

CPU へのメッセージ割込みの原因は、CAN 割込みレジスタまたは CAN 割込みペンディングレジスタで確認することができます（「15.4.4 メッセージハンドラレジスタ」参照）。メッセージ割込みをクリアする場合、同時にメッセージデータを読み出すことは可能であり、CAN 割込みレジスタで示されているメッセージ割込みをクリアすると次に優先度の高い割込みが CAN 割込みレジスタにセットされ、次の割込み処理を待つことになります。割込みがない場合には、CAN 割込みレジスタは 0000_H を示します。

< 注意事項 >

- ステータス割込み (IntId=8000_H) は, CAN ステータスレジスタの読出しアクセスにより割込みクリアされます。
 - CAN ステータスレジスタの書込みアクセスによる, ステータス割込み (IntId=8000_H) は発生しません。
-

15.5.6 ビットタイミング

ビットタイミングについての概要と CAN コントローラにおけるビットタイミングについて説明します。

CAN ネットワークの各 CAN ノードは、それぞれクロック発振器（通常は水晶発振器）を持っています。ビットタイムのタイムパラメータは、CAN ノードごとに個別に構成できます。CAN ノードの発振周期 (f_{osc}) が異なっても、共通のビットレートを作り出せます。

これらの発振器の周波数は、温度や電圧の変化、コンポーネントの悪化により少し異なります。その変動が発振器の許容範囲 (df) 内である限りは、CAN ノードはビットストリームへ再同期化することで異なるビットレートを補償できます。

CAN 仕様に準じて、ビットタイムは 4 つの区分に分けられ（図 15.5-3 参照）、同期化部 (Sync_Seg)、伝送時間部 (Prop_Seg)、フェーズバッファ部 1 (Phase_Seg1)、フェーズバッファ部 2 (Phase_Seg2) で構成されます。それぞれの区分は、プログラマブルな時間量（表 15.5-3 参照）からなります。ビットタイムの基本単位時間 (t_q) は、CAN コントローラのシステムクロック f_{sys} とボーレートプリスケアラ (BRP) で定義されます。

$$t_q = BRP / f_{sys}$$

CAN のシステムクロック f_{sys} は、Clock 入力の周波数（図 15.2-1 参照）になります。同期化部の Sync_Seg は、CAN バスのエッジを期待するビットタイム内のタイミングとなります。伝送時間部の Prop_Seg は、CAN ネットワーク内の物理的遅延時間を補償します。フェーズバッファ部の Phase_Seg1、Phase_Seg2 は、サンプリングポイントを指定します。再同期化ジャンプ幅 (SJW) は、エッジフェーズエラーを補償するために再同期化時のサンプリングポイントの移動幅を定義します。

図 15.5-3 ビットタイミング

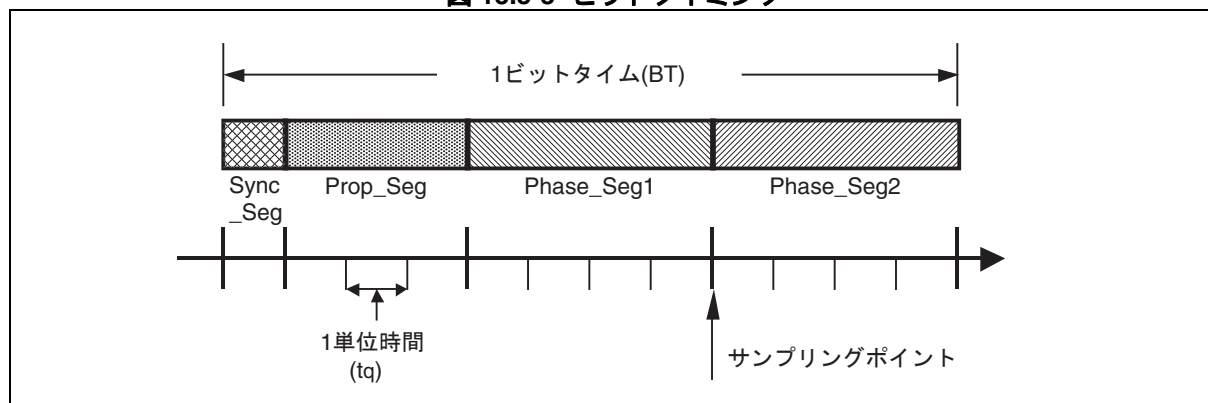


表 15.5-3 CAN ビットタイムのパラメータ

パラメータ	レンジ	機能
BRP	[1-32]	時間量の長さ t_q の定義
Sync_Seg	1 t_q	固定長 システムクロックへの同期化
Prop_Seg	[1-8] t_q	物理遅延時間の補償
Phase_Seg1	[1-8] t_q	サンプルポイント前のエッジフェーズエラー保証 同期化により一時的に長くされる可能性があります。
Phase_Seg2	[1-8] t_q	サンプルポイント後のエッジフェーズエラー保証 同期化により一時的に短くされる可能性があります。
SJW	[1-4] t_q	再同期化ジャンプ幅 どちらかのフェーズバッファ部より長くなることはありません。

次に CAN コントローラにおけるビットタイミングを示します。

図 15.5-4 CAN コントローラにおけるビットタイミング

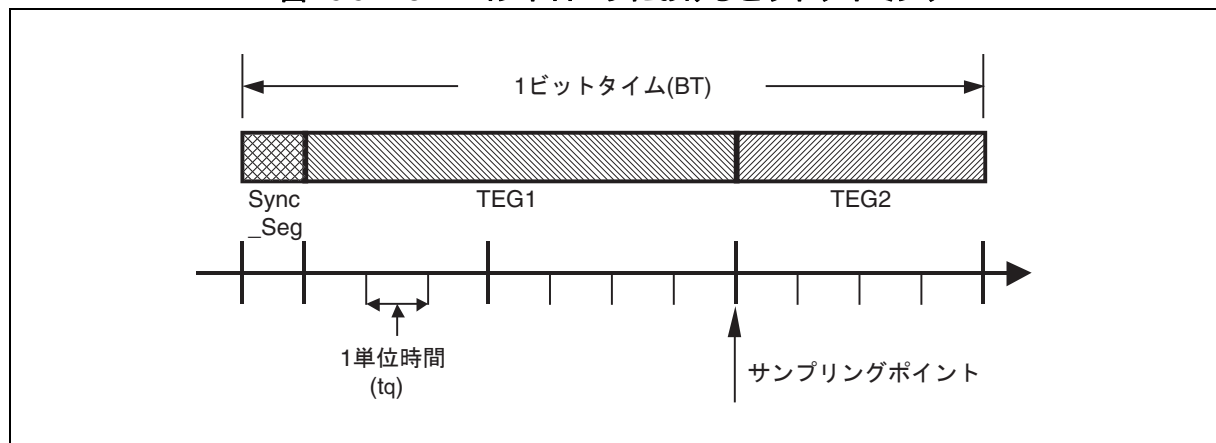


表 15.5-4 CAN コントローラのパラメータ

パラメータ	レンジ	機能
BRPE,BRP	[0-1023]	時間量の長さ t_q の定義 ビットタイミングレジスタおよびプレスケーラ拡張レジスタにより最大 1024 までプリスケーラを拡張できます。
Sync_Seg	1 t_q	システムクロックへの同期化 固定長
TSEG1	[1-15] t_q	サンプリングポイント前のタイムセグメントです。 Prop_Seg と Phase_Seg1 に相当します。 ビットタイミングレジスタにより制御可能です。
TSEG2	[0-7] t_q	サンプリングポイント後のタイムセグメントです。 Phase_Seg2 に相当します。 ビットタイミングレジスタにより制御可能です。
SJW	[0-3] t_q	再同期化ジャンプ幅です。 ビットタイミングレジスタにより制御可能です。

各パラメータの関係を以下に示します。

$$t_q = ([BRPE, BRP] + 1) / f_{sys}$$

$$BT = SYNC_SEG + TEG1 + TEG2$$

$$= (1 + (TSEG1 + 1) + (TSEG2 + 1)) \times t_q$$

$$= (3 + TSEG1 + TSEG2) \times t_q$$

15.5.7 テストモード

テストモードの設定方法および動作について説明します。

■ テストモード設定

CAN 制御レジスタの Test ビットに "1" をセットすることでテストモードになります。テストモードに設定すると、CAN テストレジスタのビット Tx1, Tx0, LBack, Silent, Basic ビットが有効となります。

CAN 制御レジスタの Test ビットを "0" にリセットすることにより、すべてのテストレジスタ機能を無効にします。

■ サイレントモード

CAN テストレジスタの Silent ビットを "1" にセットすることにより、CAN コントローラをサイレントモードに設定できます。

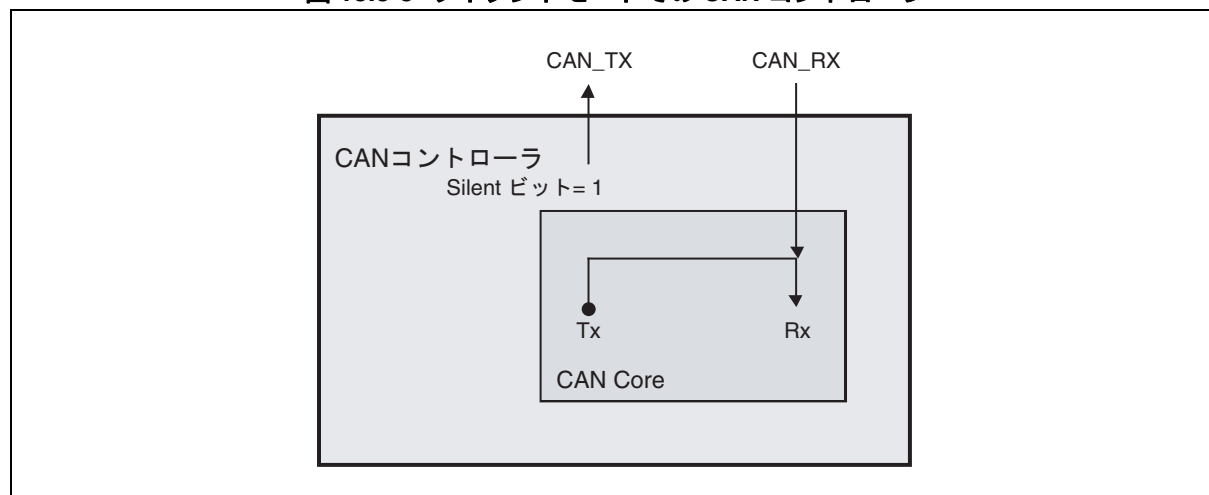
サイレントモードでは、データフレームとリモートフレームを受信可能であるが、CAN バス上にはレセッシブのみ出力し、メッセージおよび ACK の送信を行いません。

CAN コントローラがドミナントビット (ACK ビット、オーバーロードフラグ、アクティブエラーフラグ) の送信を要求された場合、CAN コントローラ内部の折り返し回路で RX 側に送信されます。この動作では、CAN バス上においてレセッシブ状態であっても、受信側では CAN コントローラ内部で折り返し送信されたドミナントビットを受信します。

サイレントモードでは、ドミナントビット (ACK ビット、エラーフラグ) 送信による影響がない状態で、CAN バス上のトラフィック解析ができます。

図 15.5-5 にサイレントモードでの信号 CAN_TX と CAN_RX の CAN コントローラへの接続を示します。

図 15.5-5 サイレントモードでの CAN コントローラ



■ ループバックモード

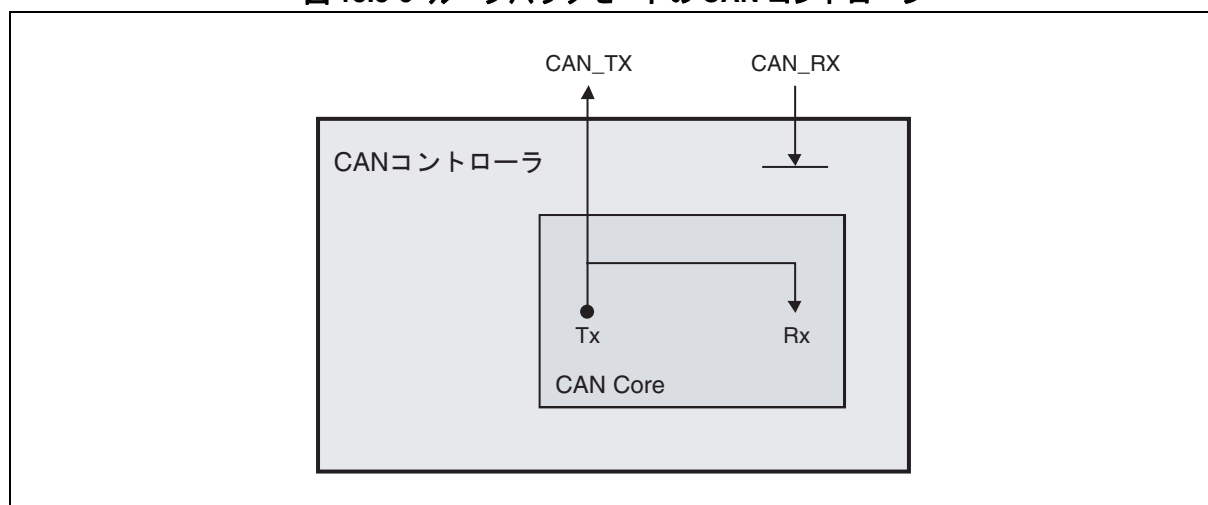
CAN テストレジスタの LBack ビットを "1" にセットすることにより, CAN コントローラをループバックモードに設定できます。

ループバックモードは, 自己診断機能に使用できます。

ループバックモードでは, CAN コントローラ内部で TX 側と RX 側が接続され, CAN コントローラが送信したメッセージを, RX 側で受信したメッセージとして扱い, 受容フィルタを通過したメッセージは, 受信バッファに格納します。

図 15.5-6 にループバックモードでの信号 CAN_TX と CAN_RX の CAN コントローラへの接続を示します。

図 15.5-6 ループバックモードの CAN コントローラ



< 注意事項 >

外部信号から独立するため, データ / リモートフレームのアクノリッジスロットでのドミナントビットはサンプリングされません。そのため通常, CAN コントローラはアクノリッジエラーを発生しますが, 本テストモードではアクノリッジエラーを発生しないようにしています。

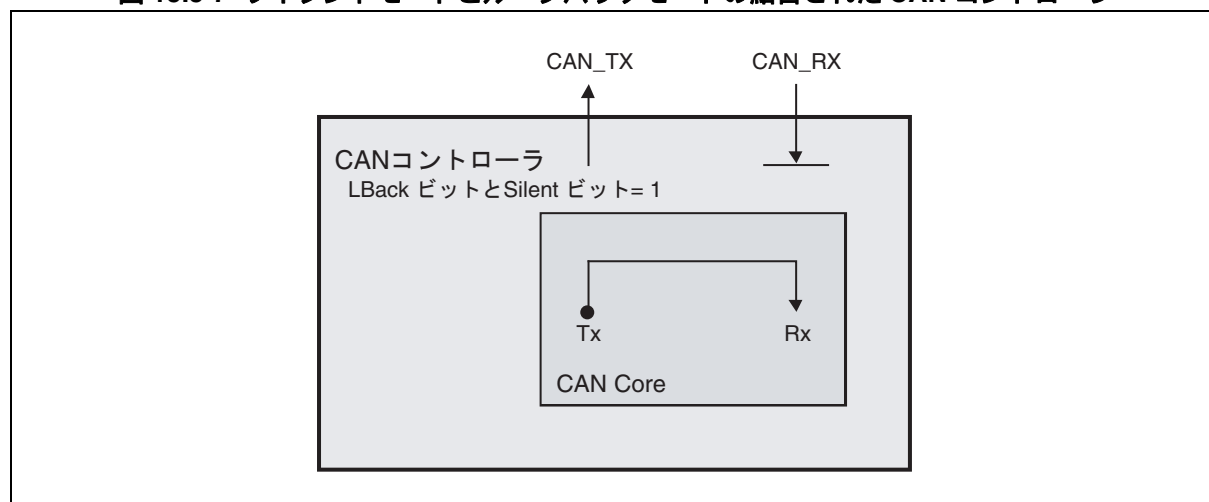
■ サイレントモードとループバックモードの結合

CAN テストレジスタの LBack ビットと Silent ビットを同時に "1" にセットすることにより、ループバックモードとサイレントモードを結合させ動作することが可能です。

このモードは、「ホットセルフテスト」用に使えます。「ホットセルフテスト」とは、CAN コントローラがループバックモードでテストしたときに、CAN_TX 端子にはレセシブの固定出力、CAN_RX 端子からの入力は無効となりますので、CAN システムの動作に影響がないことを意味しています。

図 15.5-7 にサイレントモードとループバックモードの結合したときの信号 CAN_TX と CAN_RX の CAN コントローラへの接続を示します。

図 15.5-7 サイレントモードとループバックモードの結合された CAN コントローラ



■ ベーシックモード

CAN テストレジスタの Basic ビットを "1" にセットすることにより、CAN コントローラをベーシックモードに設定できます。

ベーシックモードでは、CAN コントローラは、メッセージ RAM を使用せずに動作します。

IF1 メッセージインタフェースレジスタは、送信制御用として使用されます。

メッセージ送信を行う場合、まず、IF1 メッセージインタフェースレジスタに送信する内容を設定します。次に、IF1 コマンド要求レジスタの Busy ビットに "1" をセットすることで送信要求します。Busy ビットが "1" にセットされている間、IF1 メッセージインタフェースレジスタをロックしている、または送信が保留されていることを示します。

Busy ビットに "1" がセットされると CAN コントローラは以下の動作を行います。

CANバスがバスアイドルになるとすぐに、IF1 メッセージインタフェースレジスタの内容を、送信用シフトレジスタへロードし、送信を開始します。正常に送信完了すると、Busy ビットが "0" にリセットされ、ロックされていた IF1 メッセージインタフェースレジスタを開放します。

送信が保留されているときに、IF1 コマンド要求レジスタの Busy ビットを "0" にリセットすることでいつでも中断できます。また、送信中に Busy ビットを "0" にリセットすると、調停負けやエラーなどの場合に行われる再送信を停止します。

IF2 メッセージインタフェースレジスタは、受信制御用として使用されます。

メッセージの受信は、受容フィルタを使用せずにすべて受信します。IF2 コマンド要求レジスタの Busy ビットを "1" にセットすることにより、受信したメッセージの内容を読み出すことが可能となります。

Busy ビットに "1" がセットされると CAN コントローラは以下の動作を行います。

- 受信したメッセージ（受信用のシフトレジスタの内容）を受容フィルタなしで、IF2 メッセージインタフェースレジスタへ格納します。

新しいメッセージが IF2 メッセージインタフェースレジスタに格納された場合、CAN コントローラが NewDat ビットを "1" にセットします。また、NewDat ビットが "1" のときに、さらに新しいメッセージを受信した場合は、CAN コントローラが MsgLst を "1" にセットします。

< 注意事項 >

- ベーシックモードでは、制御/状態ビットに関係するすべてのメッセージオブジェクトと IFx コマンドマスクレジスタの制御モード設定は無効となります。
- コマンド要求レジスタのメッセージ番号は無効です。
- IF2 メッセージ制御レジスタの NewDat ビットと MsgLst ビットは通常時と同様に動作し、DLC3 ~ DLC0 は受信された DLC を示し、ほかの制御ビットは "0" として読み出されます。

■ 端子 CAN_TX のソフトウェア制御

CAN 送信端子である CAN_TX は、4 つの出力機能があります。

- シリアルデータ出力（通常出力）
- CAN コントローラのビットタイミングをモニタするための、CAN サンプリングポイント信号出力
- ドミナント固定出力
- レセッシブ固定出力

ドミナントおよびレセッシブの固定出力は、CAN 受信端子の CAN_RX モニタ機能と共に、CAN バスの物理層のチェックに使用することができます。

CAN_TX 端子の出力モードは、CAN テストレジスタの Tx1 と Tx0 ビットにより制御可能です。

< 注意事項 >

CAN メッセージ送信もしくはループバックモード、サイレントモード、ベーシックモードを使用する際は、CAN_TX をシリアルデータ出力に設定する必要があります。

15.5.8 ソフトウェア初期化

ソフトウェアによる初期化について説明します。

ソフトウェアでの初期化要因を以下に示します。

- ハードウェアリセット
- CAN 制御レジスタの Init ビットの設定
- バスオフ状態への遷移

ハードウェアによるリセットは、メッセージ RAM (MsgVal, NewDat, IntPnd, TxRqst ビットを除く) 以外すべて初期化されます。メッセージ RAM はハードウェアによるリセット後、CPU によって初期化するかメッセージ RAM の MsgVal を "0" にしてください。また、ビットタイミングレジスタを設定する場合には、CAN 制御レジスタの Init ビットを "0" にクリアする前に設定してください。

CAN 制御レジスタの Init ビットは、以下の条件で "1" にセットされます。

- CPU からの "1" 書込み
- ハードウェアリセット
- バスオフ

Init ビットが "1" にセットされると、CAN バスの全メッセージ送受信は停止され、CAN バス出力の CAN_TX 端子はレセッシブ出力となります (CAN_TX テストモードは除く)。Init ビットに "1" をセットすると、エラーカウンタは変化しませんし、レジスタも変更されません。

CAN 制御レジスタの Init ビットと CCE ビットが "1" にセットされると、ボーレート制御用のビットタイミングレジスタとプリスケアラ拡張レジスタへの設定が可能となります。

Init ビットを "0" にリセットすることによりソフトウェア初期化を終了します。また、Init ビットを "0" にすることは、CPU からのアクセスでしか実行できません。

Init ビットが "0" にリセットされてから、連続した 11 ビットのレセッシブの発生 (= バスアイドル) を待つことにより、CAN バス上のデータ転送と同期化した後、メッセージの転送が行われます。

通常動作中にメッセージオブジェクトのマスク、ID、XTd、EoB、RmtEn を変更する場合、MsgVal を無効に設定してから変更してください。

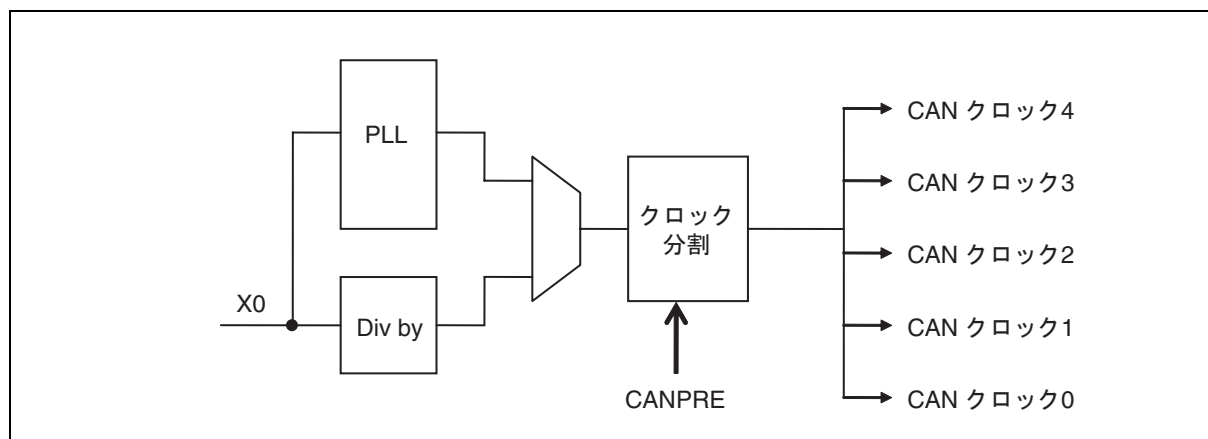
15.5.9 CAN クロックプリスケーラ

PLL 動作中の CAN クロック切換えについて説明します。

■ ブロック図

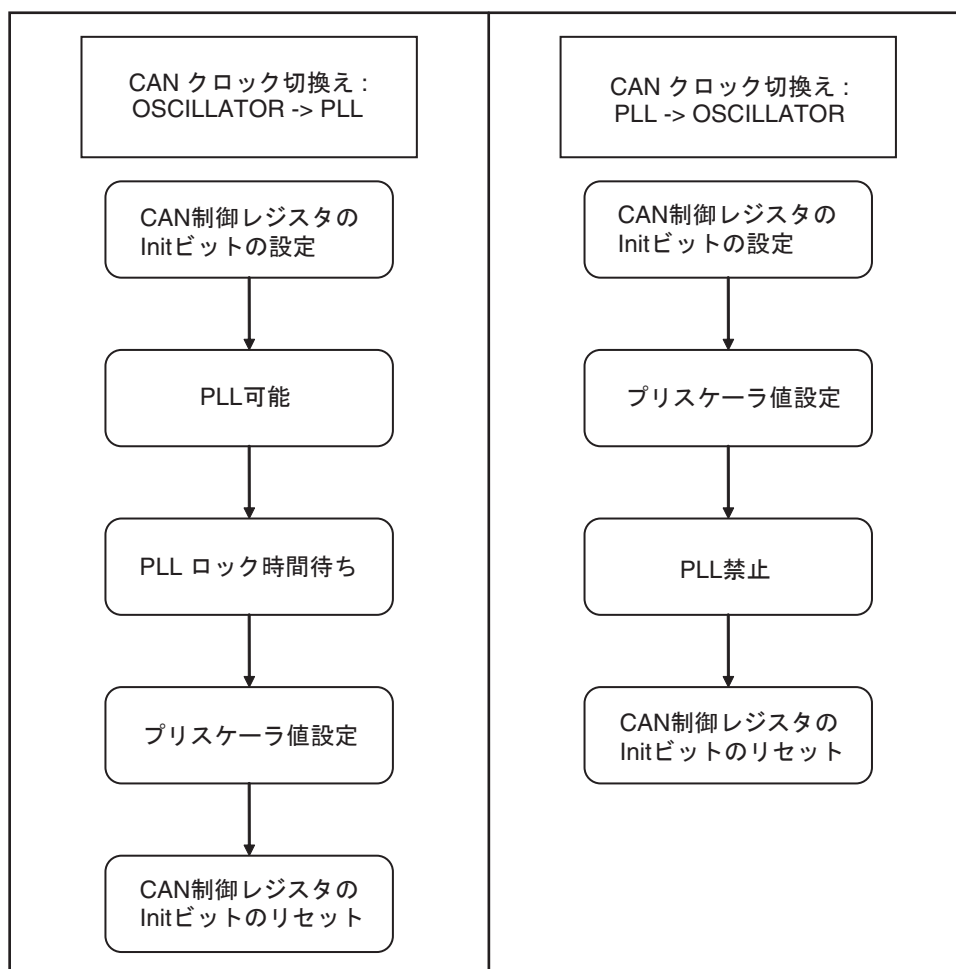
CAN クロックプリスケーラの概要について以下のブロック図に示します。

CAN クロックプリスケーラレジスタのCANPREビットの設定に従いCANインタフェースに供給されるクロックの分周比が決定されます。



■ クロック切換え手順

CAN クロックプリスケラを使用したクロックの切換え方法については、以下に示される手順を推奨します。



■ CAN クロックプリスケアラ設定

CAN クロックプリスケアラに設定可能な値を示します。

CAN インタフェースに供給されるクロックは、システムクロックを CAN クロックプリスケアラの設定値に従い分周されたものとなります。

CANPRE[3:0]	機能	システムクロック :32 MHz 時
0000 _B	CAN クロックとしてシステムクロックの 1/2 周期が選択されます。 (初期値 : CANPRE[3:0]=0000 _B)	16MHz
0001 _B		
001x _B	CAN クロックとしてシステムクロックの 1/4 周期が選択されます。	8MHz
01xx _B	CAN クロックとしてシステムクロックの 1/8 周期が選択されます。	4MHz
1000 _B	CAN クロックとしてシステムクロックの 2/3 周期が選択されます。 クロックの Duty は 67% となります。	21.33MHz (設定禁止)
1001 _B	CAN クロックとしてシステムクロックの 1/3 周期が選択されます。	10.67MHz
101x _B	CAN クロックとしてシステムクロックの 1/6 周期が選択されます。	5.33MHz
11xx _B	CAN クロックとしてシステムクロックの 1/12 周期が選択されます。	2.67MHz

< 注意事項 >

- CAN クロックプリスケアラ設定ビットの変更は、CAN 制御レジスタの初期化ビットを "1" にセットし、すべてのバス動作を停止させた後に行ってください。
- 本レジスタの設定により CAN インタフェースに供給するクロックは 16MHz 以下としてください。
- CANPRE[3:0]=1000_B は、設定禁止です。

第16章

8/10 ビット A/D コンバータ

A/D コンバータの概要，レジスタの構成 / 機能および動作について説明します。

- 16.1 概要
- 16.2 構成
- 16.3 端子
- 16.4 レジスタ
- 16.5 割込み
- 16.6 動作説明
- 16.7 A/D 変換データ保護機能
- 16.8 使用上の注意

16.1 概要

8/10 ビット A/D コンバータには、RC 逐次比較変換方式でアナログ入力電圧を 10 ビットもしくは 8 ビットのデジタル値に変換する機能があります。入力信号は、各アナログ入力端子から選択し、変換起動はソフトウェア、内部クロック、外部端子トリガの 3 種類から選択できます。

■ 8/10 ビット A/D コンバータの概要

アナログ入力端子に入力されたアナログ電圧（入力電圧）をデジタル値に A/D 変換する機能があり、次の特長をもっています。

- 変換時間は、最小 1.2 μ s(マシクロック 33 MHz 時、サンプリング時間を含む)です。
- 変換方式は、サンプルホールド回路付き RC 逐次変換比較方式です。
- 10 ビットまたは 8 ビットの分解能が選択できます。
- アナログ入力端子はプログラムで選択できます。
- A/D 変換終了時に割込み要求を発生できます。
- 割込み許可の状態では変換データ保護機能が働くため連続変換してもデータの欠落がありません。
- 変換の起動要因はソフトウェア、16 ビットリロードタイマ 1 もしくは多機能タイマ（立上りエッジ）、外部端子トリガ（立下りエッジ）から選択できます。

変換モードは 3 種類あります。

表 16.1-1 8/10 ビット A/D コンバータの変換モード

変換モード	シングル変換動作	スキャン変換動作
単発変換モード	指定したチャンネル (1 チャンネルのみ) を 1 回変換して終了	連続した複数のチャンネル (最大 7 チャンネルまで指定可能) を 1 回変換して終了
連続変換モード	指定したチャンネル (1 チャンネルのみ) を繰り返し変換	連続した複数のチャンネル (最大 7 チャンネルまで指定可能) を繰り返し変換
停止変換モード	指定したチャンネル (1 チャンネルのみ) を 1 回変換したら一時停止し、次の起動がかかるまで待機	連続した複数のチャンネル (最大 7 チャンネルまで指定可能) を変換 ただし、1 チャンネル変換ごとに一時停止し、次の起動がかかるまで待機

< 注意事項 >

- 本品種では、4 チャンネルのアナログ入力をもつユニット 1 と、7 チャンネルのアナログ入力をもつユニット 2 の 2 ユニットを搭載しています。
- ユニット 2 の起動要因で、多機能タイマによる起動許可をしたときは、リロードタイマ 1 による起動はできません。

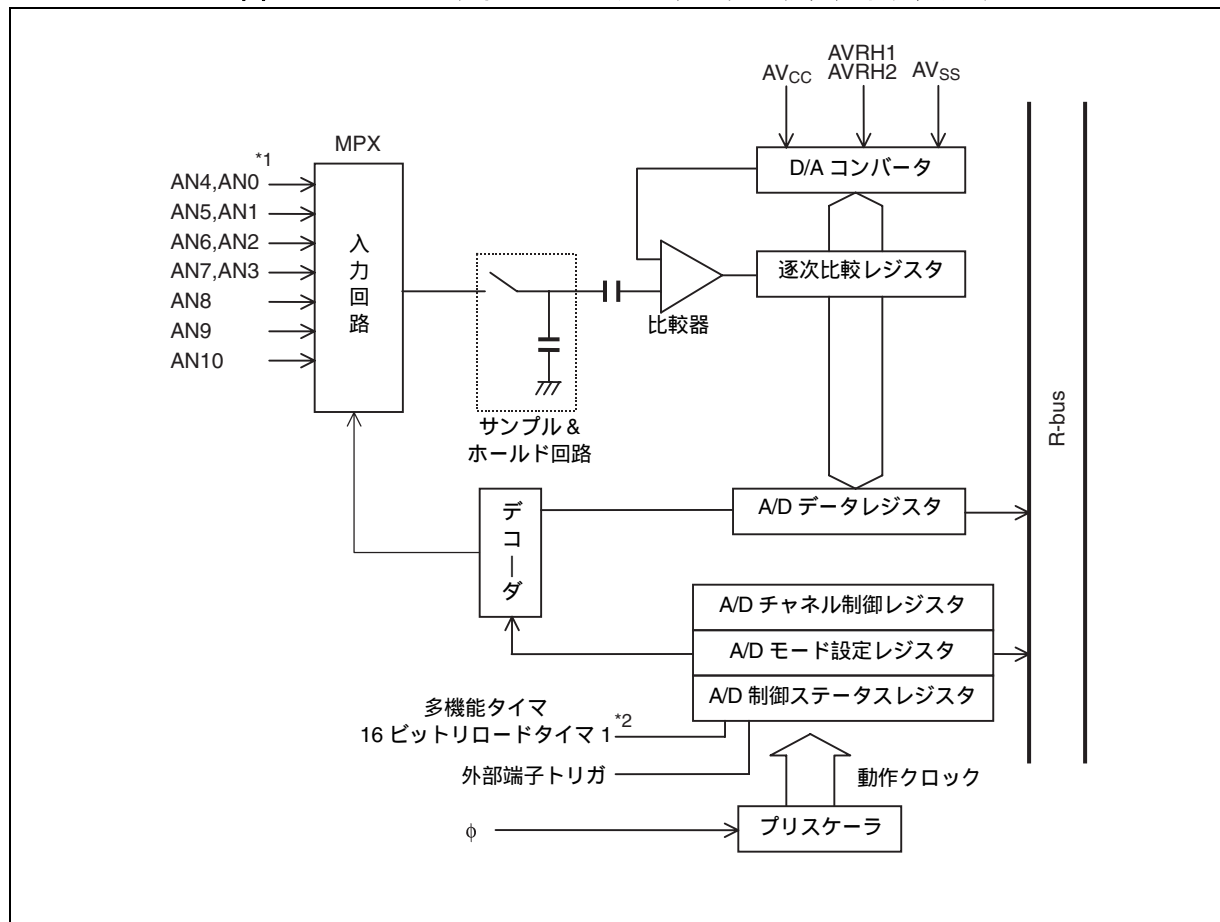
16.2 構成

8/10 ビット A/D コンバータは、次の 11 のブロックで構成されています。

- A/D 制御ステータスレジスタ (ADCS)
- A/D チャンネル制御レジスタ (ADCH)
- A/D モード設定レジスタ (ADMD)
- A/D データレジスタ (ADCD)
- クロックセクタ (A/D 変換起動用入力クロックセクタ)
- デコーダ
- アナログチャンネルセクタ
- サンプルホールド回路
- D/A コンバータ
- コンパレータ
- コントロール回路

■ 8/10 ビット A/D コンバータのブロックダイアグラム

図 16.2-1 8/10 ビット A/D コンバータのブロックダイアグラム



*1: ユニット 1 : AN0, AN1, AN2, AN3 の 4 チャンネル

ユニット 2 : AN4 ~ AN10 の 7 チャンネル

*2: ユニット 1 : 多機能タイマ

ユニット 2 : 多機能タイマまたは 16 ビットリロードタイマ 1

(多機能タイマによる起動が優先されます。)

● A/D 制御ステータスレジスタ (ADCS)

一時停止および変換確認, 割込み要求の許可 / 禁止, 割込み要求の状態の確認, A/D 変換の分解能を選択する機能があります。

● A/D チャンネル制御レジスタ (ADCH)

A/D チャンネルを選択する機能があります。

● A/D モード設定レジスタ (ADMD)

変換モードの選択と, A/D 変換のコンペア時間やサンプリング時間を設定する機能があります。

● A/D データレジスタ (ADCD)

A/D 変換結果を格納するレジスタです。

● クロックセクタ

A/D 変換起動クロックを選択するセクタです。起動クロックには, 16 ビットリロードタイマ ch.1 出力, 多機能タイマまたは外部端子トリガが選択できます。

- ユニット 2 は, 16 ビットリロードタイマ ch.1 出力または多機能タイマにより起動可能
- ユニット 1 は, 多機能タイマにより起動可能

● デコーダ

A/D チャンネル制御レジスタ (ADCH) の ANE0 ~ ANE2, ANS0 ~ ANS2 ビットの設定から, 使用するアナログ入力端子を選択する回路です。

● アナログチャンネルセクタ

アナログ入力端子の中から使用する端子を選択する回路です。

- ユニット 1 は, 4 本のアナログ入力
- ユニット 2 は, 7 本のアナログ入力

● サンプルホールド回路

アナログチャンネルセクタで選択された入力電圧を保持する回路です。A/D 変換を起動した直後の入力電圧をサンプルホールドすることで, A/D 変換中 (比較中) の入力電圧の変動の影響を受けずに変換できます。

● D/A コンバータ

サンプルホールドされた入力電圧と比較するための基準電圧を発生します。

● コンパレータ

サンプルホールドされた入力電圧と D/A コンバータの出力電圧を比較し, 大小を判定します。

● コントロール回路

コンパレータからの大小信号で A/D 変換値を決定します。A/D 変換の終了後、変換結果を A/D データレジスタ (ADCD) に格納し、割込み要求を発生します。

16.3 端子

8/10 ビット A/D コンバータの端子および端子のブロックダイアグラムを示します。

■ 8/10 ビット A/D コンバータの端子

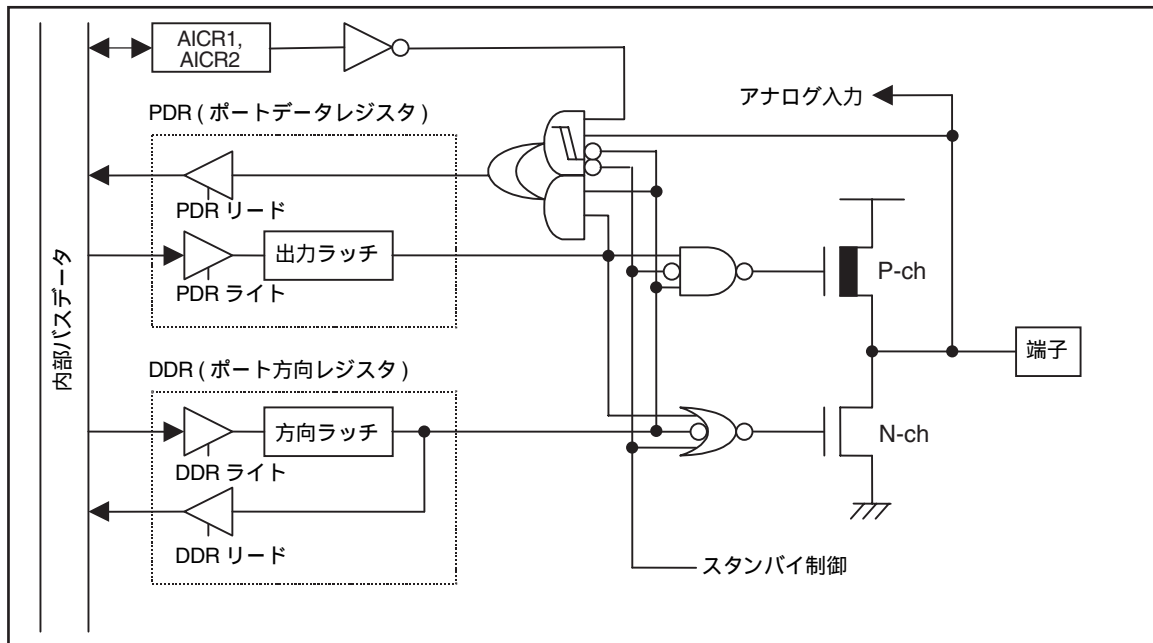
A/D コンバータの端子は汎用ポートと兼用になっています。表 16.3-1 に端子の機能、入出力形式、8/10 ビット A/D コンバータ使用時の設定などを示します。

表 16.3-1 8/10 ビット A/D コンバータの端子

機能	端子名	端子機能	入出力形式	プルアップ 設定	スタンバイ 制御	端子の使用に必要な I/O ポートの設定		
ch.0	P50/AN0	ポート 5 入出力 / アナログ入力	CMOS 出力 / CMOS ヒステ リシス入力 または アナログ入力	なし	あり	ポート 5 を入力設定 (DDR5:bit0 ~ bit7=0) アナログ入力に設定 (AICR1 : bit0 ~ bit3=1)		
ch.1	P51/AN1							
ch.2	P52/AN2							
ch.3	P53/AN3							
ch.4	P54/AN4					ポート 5 を入力設定 (DDR5:bit0 ~ bit7=0) アナログ入力に設定 (AICR2 : bit0 ~ bit3=1)		
ch.5	P55/AN5							
ch.6	P56/AN6							
ch.7	P57/AN7							
ch.8	P44/AN8	ポート 4 入出力 / アナログ入力						ポート 4 を入力設定 (DDR4:bit4 ~ bit6=0) アナログ入力に設定 (AICR2 : bit4 ~ bit6=1)
ch.9	P45/AN9							
ch.10	P46/AN10							

■ 8/10 ビット A/D コンバータの端子のブロックダイヤグラム

図 16.3-1 AN0 ~ AN10 端子のブロックダイヤグラム



< 注意事項 >

- 入力ポートとして使用する端子は、対応する DDR レジスタのビットを "0" に設定し、かつ外部端子にプルアップ抵抗を付加してください。また AICR レジスタの対応するビットを "0" に設定してください。
- アナログ入力端子として使用する端子は、対応する AICR レジスタのビットを "1" に設定してください。このときの PDR レジスタの読出し値は "0" になります。

16.4 レジスタ

8/10 ビット A/D コンバータのレジスタ一覧を示します。

■ 8/10 ビット A/D コンバータのレジスタ一覧

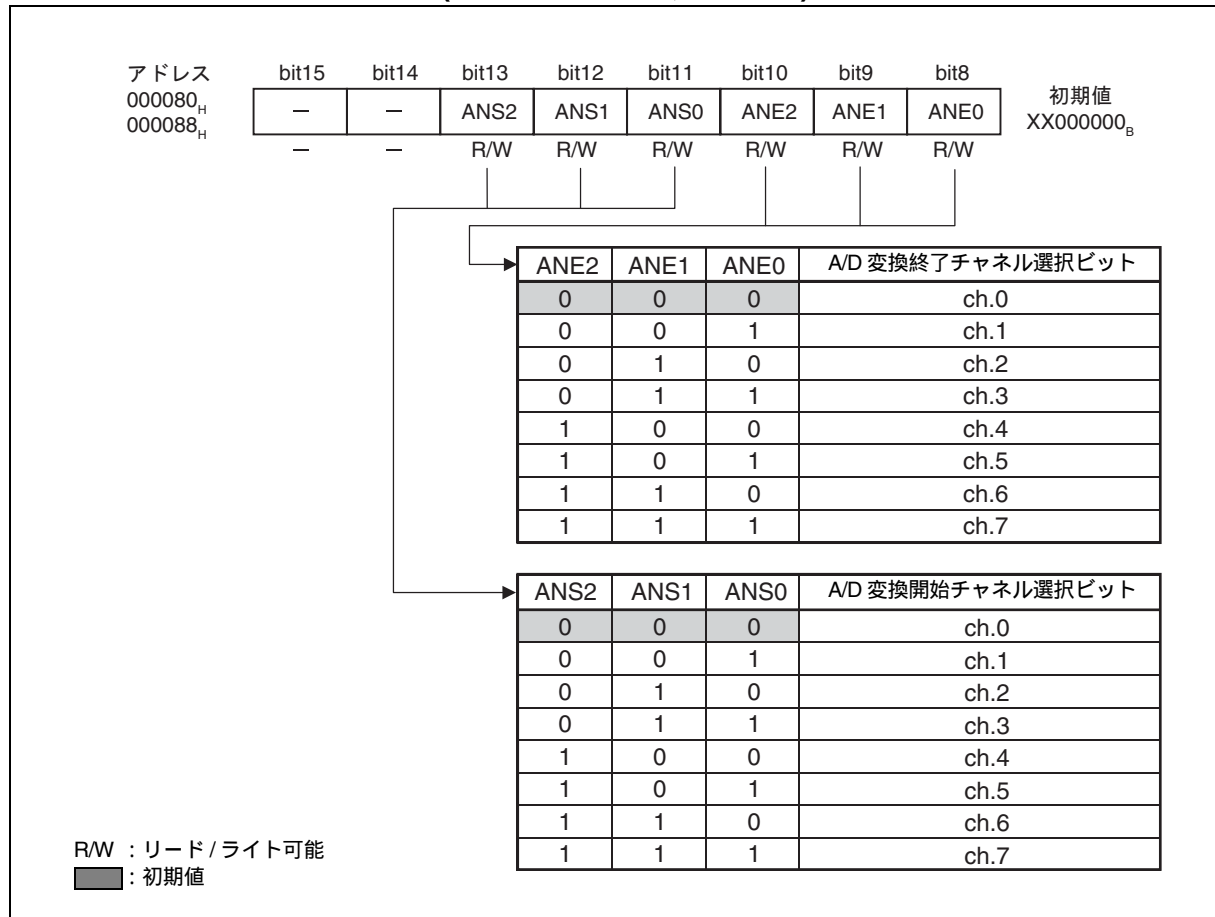
図 16.4-1 8/10 ビット A/D コンバータのレジスタ一覧

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
000086 _H	AICR1															
00008E _H	AICR2															
000080 _H	ADCH1								ADMD1							
000084 _H	ADCS1															
000082 _H	ADCD11								ADCD10							
000088 _H	ADCH2								ADMD2							
00008C _H	ADCS2															
00008A _H	ADCD21								ADCD20							

16.4.1 A/D チャンネル制御レジスタ (ADCH)

A/D チャンネル制御レジスタは、A/D 変換チャンネルの選択をする機能があります。

■ A/D チャンネル制御レジスタ (ADCH: ADCH1, ADCH2)



< 注意事項 >

- A/D ユニット 1 は、ch.0 ~ ch.3 = AN0 ~ AN3 (ch.4 ~ ch.7 は空き)
- A/D ユニット 2 は、ch.0 ~ ch.6 = AN4 ~ AN10 (ch.7 は空き)
- A/D ユニット 1 の ANE2, ANS2 には、必ず、"0" を書き込んでください。
- A/D ユニット 2 の ANE0 ~ ANE2=1, ANS0 ~ ANS2=1 は、設定禁止です。
- A/D ユニット 1, 2 では、必ず、"ANS0 ANE0" となるように設定してください。

表 16.4-1 A/D チャンネル制御レジスタ (ADCH) の各ビットの機能説明

ビット名		機能
bit15, bit14	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは、動作に影響しません。
bit13 ~ bit11	ANS2, ANS1, ANS0: A/D 変換開始 チャンネル選択 ビット	<ul style="list-style-type: none"> A/D 変換の開始チャンネルの設定および変換中チャンネル番号の確認を行うビットです。 A/D 変換を起動すると、これらのビットに書き込まれたチャンネルから A/D 変換を開始します。 A/D 変換中は、変換中のチャンネル番号が読めます。停止変換モードでの一時停止中は、直前に変換したチャンネルの番号が読み出せます。
bit10 ~ bit8	ANE2, ANE1, ANE0: A/D 変換終了 チャンネル選択 ビット	<ul style="list-style-type: none"> A/D 変換の終了チャンネルの設定を行うビットです。 A/D 変換を起動すると、これらのビットに書き込まれたチャンネルまで A/D 変換を行います。 ANS2 ~ ANS0 と同じチャンネルを設定すると、そのチャンネルのみ変換を行います。 また、連続変換モードまたは停止変換モードを設定しているときは、これらのビットで設定されたチャンネルまでの変換が終わると、ANS2 ~ ANS0 で設定された開始チャンネルに戻ります。このとき設定チャンネルは、必ず開始チャンネル 終了チャンネルとなるように設定してください。 A/D 変換開始チャンネル選択ビット (ANS2, ANS1, ANS0) に開始チャンネルを設定した後に、A/D 変換モード選択ビット (MD1, MD0) および A/D 変換終了チャンネル選択ビット (ANE2, ANE1, ANE0) をリードモディファイライト系命令で設定しないでください。 ANS2, ANS1, ANS0 ビットは A/D 変換動作が開始するまでは前回の 変換チャンネルが読み出されるため、ANS2, ANS1, ANS0 ビットに開始 チャンネルを設定した後に、MD1, MD0 ビットおよび ANE2, ANE1, ANE0 ビットをリードモディファイライト系命令で設定した場合、 ANE2, ANE1, ANE0 ビットの値が書き換わる可能性があります。

16.4.2 A/D モード設定レジスタ (ADMD)

A/D モード設定レジスタは、変換モードの選択と、A/D 変換のコンペア時間やサンプリング時間を設定する機能があります。

■ A/D モード設定レジスタ (ADMD: ADMD1, ADMD2)

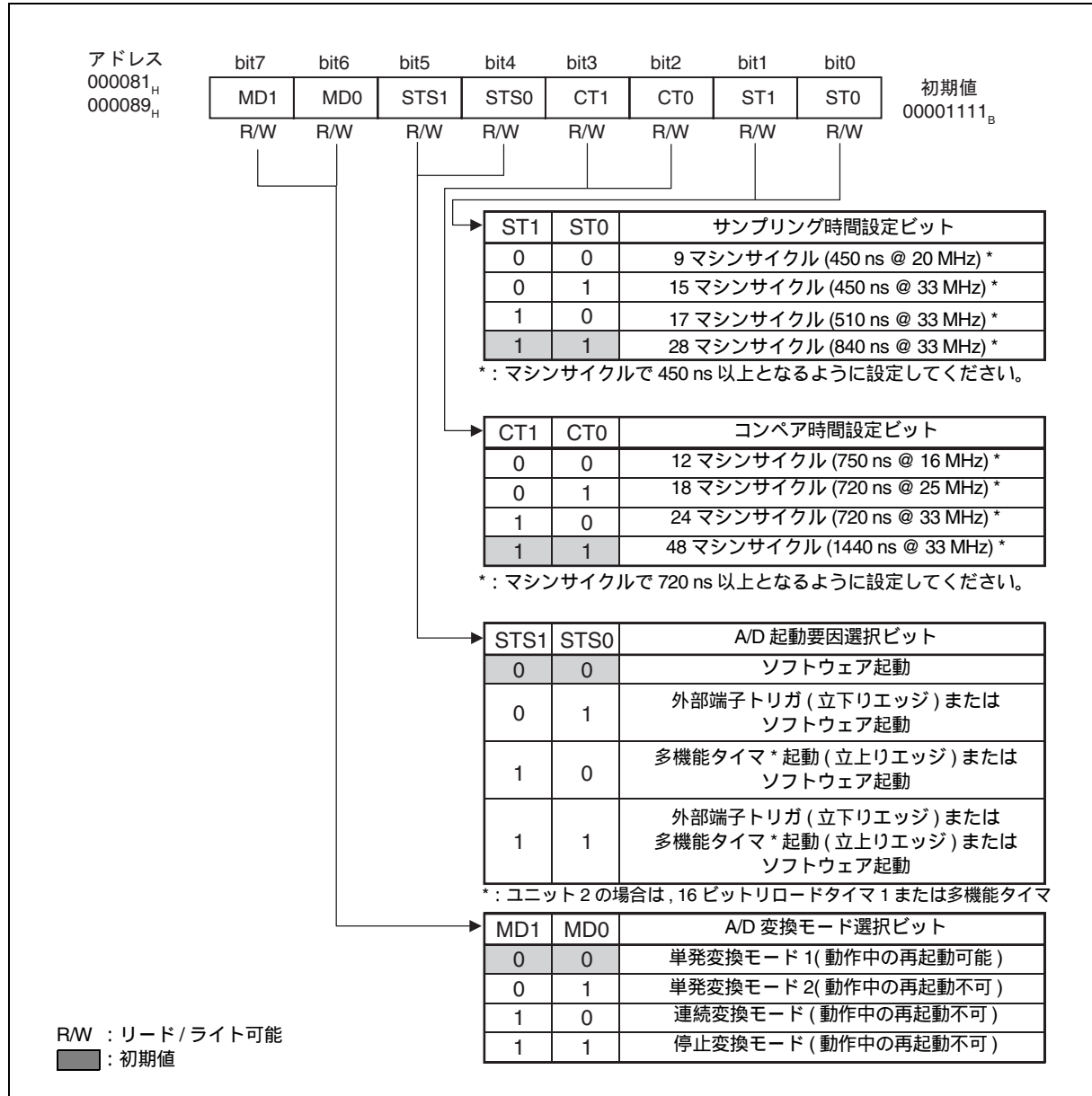


表 16.4-2 A/D モード設定レジスタ (ADMD) の各ビットの機能説明 (1 / 2)

ビット名	機能
bit7, bit6	<p>MD1, MD0 : A/D 変換モード選択ビット</p> <ul style="list-style-type: none"> • A/D 変換機能時の変換モードを選択するビットです。 • MD1, MD0 の 2 ビット値によって, 単発変換モード 1, 単発変換モード 2, 連続変換モード, 停止変換モードのいずれかが選択されます。 • それぞれのモードの意味は, 以下のとおりです。 <p>単発変換モード 1 : ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまでの A/D 変換を連続して, 1 度だけします。動作中の再起動が可能です。</p> <p>単発変換モード 2 : ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまでの A/D 変換を連続して, 1 度だけします。動作中の再起動はできません。</p> <p>連続変換モード : ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまでの A/D 変換を連続して, BUSY ビットで強制停止するまで繰り返し行います。動作中の再起動はできません。</p> <p>停止変換モード : ANS2 ~ ANS0 の設定チャンネルから ANE2 ~ ANE0 の設定チャンネルまでの A/D 変換を 1 チャンネルごとに一時停止しながら, BUSY ビットで強制停止するまで繰り返します。動作中の再起動はできません。一時停止中の再起動は, STS1, STS0 ビットで選択した起動要因の発生によります。</p> <p>(注意事項)</p> <ul style="list-style-type: none"> • 単発, 連続, 停止の各変換モードの再起動不可はタイマ, 外部トリガ, ソフトすべての起動に適用されます。 • ビットを書換えは, 必ず変換動作前の A/D 動作が停止している状態で行ってください。 • A/D 変換モード選択ビット (MD1, MD0) を "00_B" に設定した場合は, A/D 変換中の再起動ができます。このモードではソフトウェア起動 (STS1, STS0=00_B) のみ設定可能です。再起動は下記の手順で行ってください。 <ol style="list-style-type: none"> 1. INT ビットを "0" にクリアする 2. START ビットに "1" を, INT ビットに "0" を同時に書き込む <p>A/D 変換モード選択ビット (MD1, MD0) を "01_B" に設定した場合は, A/D 変換中の再起動はできません。</p> <p>A/D 変換の再起動と終了が同時に発生した場合, 再起動は行われずに A/D 変換は終了し, データレジスタ (ADCR1/ADCR0) には "300_H" が格納されます。したがって, A/D 変換再起動と終了が同時に発生しないように再起動をかけてください。</p>

表 16.4-2 A/D モード設定レジスタ (ADMD) の各ビットの機能説明 (2 / 2)

ビット名		機能
bit5, bit4	STS1, STS0 : A/D 起動要因 選択ビット	<ul style="list-style-type: none"> A/D 変換の起動要因の選択を行います。 起動要因が兼用になっている場合には , 最初に発生した起動要因で起動します。 <p>(注意事項)</p> <ul style="list-style-type: none"> 起動要因は , 書換えと同時に変更されますので , A/D 変換動作中に書き換える場合には , 目的とする起動要因がない状態で切り換えてください。 STS1, STS0=11_B のとき , 外部トリガ入力が "L" のとき , タイマ起動はできません。また , タイマが "H" のとき , 外部トリガ起動はできません。
bit3, bit2	CT1, CT0 : コンペア時間 設定ビット	<ul style="list-style-type: none"> A/D 変換時のコンペア時間を選択するビットです。 アナログ入力を取り込まれた (サンプリング時間経過) 後 , このビットに設定された時間後に変換結果のデータが確定し , A/D コントロールステータスレジスタ (ADCD) に格納されます。 <p>(注意事項)</p> <ul style="list-style-type: none"> CT1, CT0=00_B, 10_B, 11_B のとき , コンペア時間は 720 ns 以上とし , CT1, CT0=01_B のとき , コンペア時間は 900 ns 以上となるように設定しなければ , 正常なアナログ変換値が得られない場合があります。 ビットの書換えは , 必ず変換動作前の A/D 動作が停止している状態で行ってください。
bit1, bit0	ST1, ST0 : サンプリング 時間設定 ビット	<ul style="list-style-type: none"> A/D 変換時のサンプリング時間を選択するビットです。 A/D が起動されると , このビットに設定された時間 , アナログ入力を取り込まれます。 <p>(注意事項)</p> <ul style="list-style-type: none"> サンプリング時間は 450 ns 以上となるように設定しなければ , 正常なアナログ変換値が得られない場合があります。 ビットの書換えは , 必ず変換動作前の A/D 動作が停止している状態で行ってください。

16.4.3 A/D 制御ステータスレジスタ (ADCS)

A/D 制御ステータスレジスタは、一時停止および変換確認、割込み要求の許可 / 禁止、割込み要求の状態の確認、A/D 変換の分解能を選択する機能があります。

■ A/D 制御ステータスレジスタ (ADCS: ADCS1, ADCS2)

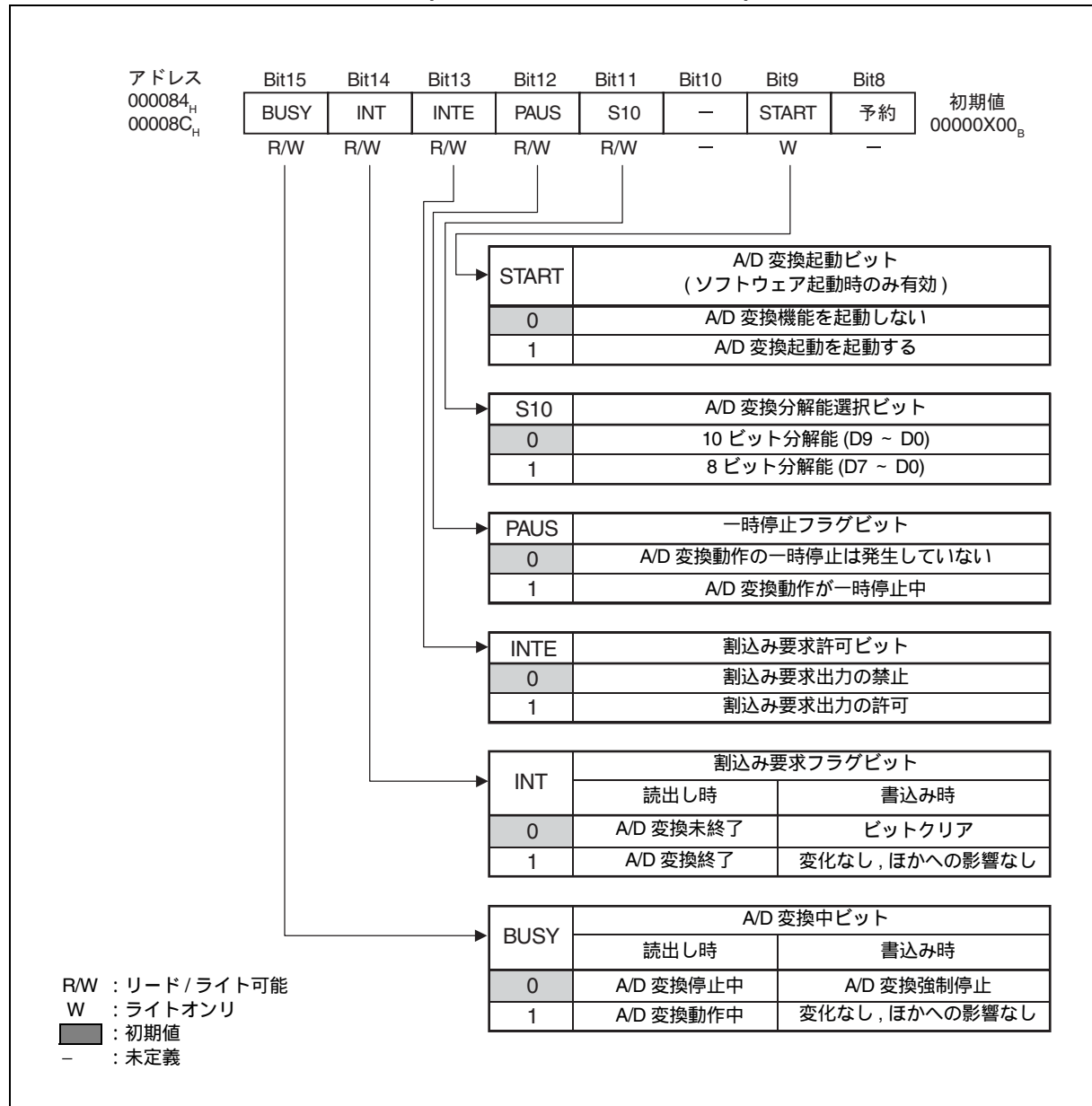


表 16.4-3 A/D 制御ステータスレジスタ (ADCS) の各ビットの機能説明 (1 / 2)

ビット名		機能
bit15	BUSY : A/D 変換中ビット	<ul style="list-style-type: none"> • A/D コンバータの動作表示ビットです。 • 読出し時, このビットが "0" であれば A/D 変換停止中であることを示し, "1" であれば A/D 変換動作中であることを示します。 • 書込み時, このビットへの "0" の書込みによって A/D 変換動作は強制的に停止されます。"1" の書込みでは, 変化せずほかへの影響はありません。 • リードモディファイライト (RMW) 時の読出しは, "1" が読み出されます。 <p>(注意事項)</p> <p>強制停止とソフトウェア起動 (BUSY=0, START=1) を同時にしないでください。</p>
bit14	INT : 割込み要求フラグビット	<ul style="list-style-type: none"> • A/D 変換によって A/D データレジスタにデータがセットされれば, このビットは "1" にセットされます。 • このビットと割込み要求許可ビット (ADCS : INTE) が "1" のとき, 割込み要求を発生します。 • 書込み時は, "0" でこのビットがクリアされ, "1" では変化せずほかへの影響はありません。 • リードモディファイライト (RMW) 時の読出しは, "1" が読み出されます。 <p>(注意事項)</p> <p>このビットへの "0" 書込みによるクリアは, A/D 停止中に行ってください。</p>
bit13	INTE : 割込み要求許可ビット	<ul style="list-style-type: none"> • CPU への割込み出力の許可 / 禁止をするビットです。 • このビットと, 割込み要求フラグビット (ADCS : INT) が "1" のとき, 割込み要求を発生します。

表 16.4-3 A/D 制御ステータスレジスタ (ADCS) の各ビットの機能説明 (2 / 2)

ビット名		機能
bit12	PAUS : 一時停止フラグ ビット	<ul style="list-style-type: none"> A/D 変換動作が一時停止したときに "1" にセットされます。 この A/D コンバータには、A/D データレジスタが 1 つしかないため、連続変換モードを使用したときに旧変換結果の CPU による読出しが完了していなければ、新しい変換結果の書込みにより旧変換データは失われてしまいます。したがって、連続変換モードを使用するときは、基本的には変換終了ごとに変換結果をメモリに転送するように設定しておく必要があります。ただし、多重割込みなどで変換データの転送が次の変換に間に合わない場合が想定できます。このビットはそのときの対処として考えられた機能で、変換終了後にデータレジスタの内容を転送するまでの間、このビットを "1" にセットし、その間は A/D 変換は停止し、次の変換データを格納しないようになっています。 このフラグのクリアはレジスタへの "0" 書込みのみです。 リードモディファイライト (RMW) 時の読出しは、"1" が読み出されます。
bit11	S10 : A/D 変換分解能 選択ビット	<ul style="list-style-type: none"> A/D 変換の分解能を選択するビットです。 このビットに "0" を書き込むと 10 ビット分解能が選択され、"1" を書き込むと 8 ビット分解能が選択されます。 <p>(注意事項)</p> <ul style="list-style-type: none"> 分解能によって、使用されるデータビットが異なります。 ビットの書換えは、必ず変換動作前の A/D 動作が停止している状態で行ってください。
bit10	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは、動作に影響しません。
bit9	START : A/D 変換起動 ビット (ソフトウェア 起動時のみ有効)	<ul style="list-style-type: none"> A/D 変換動作をソフトウェア的に起動するビットです。 このビットに "1" を書き込むと A/D 変換が起動します。 停止変換モード時は、このビットによる再起動はかかりません。 <p>(注意事項)</p> <p>強制停止とソフトウェア起動 (BUSY=0, START=1) を同時にしないでください。</p>
bit8	予約ビット	必ず、"0" を書き込んでください。

16.4.4 A/D データレジスタ (ADCD)

A/D データレジスタは , A/D 変換結果を格納するレジスタです。

■ A/D データレジスタ (ADCD: ADCD10, ADCD11, ADCD20, ADCD21)

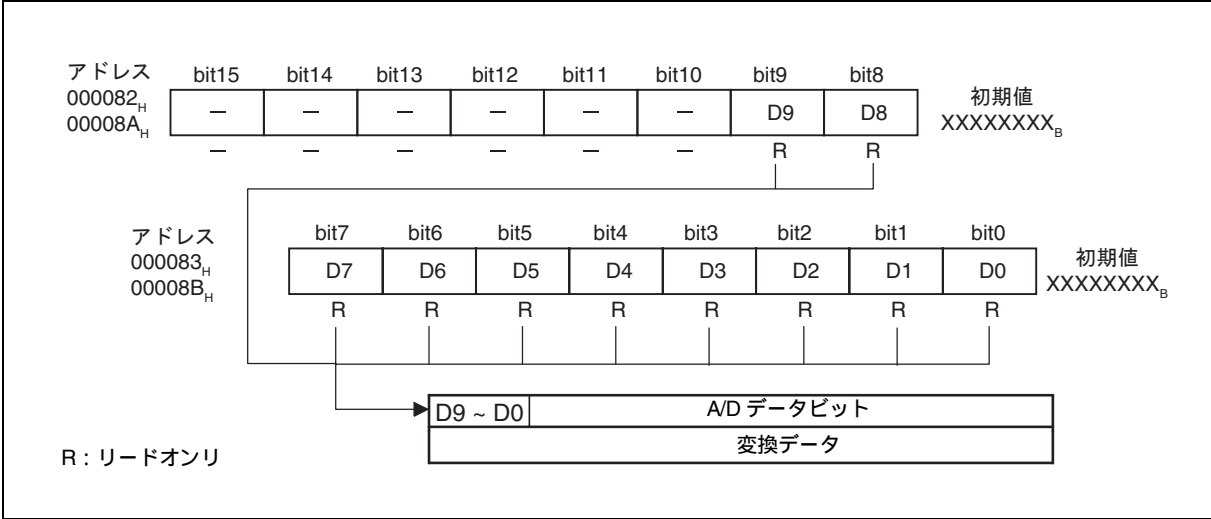


表 16.4-4 A/D データレジスタ (ADCD) の各ビットの機能説明

ビット名		機能
bit15 ~ bit10	未定義ビット	<ul style="list-style-type: none">読出し値は不定です。このビットへの書込みは , 動作に影響しません。
bit9 ~ bit0	D9 ~ D0 : A/D データビット	<ul style="list-style-type: none">A/D 変換の結果が格納され , レジスタは 1 回の変換終了ごとに書き換えられます。通常は , 最終変換値が格納されます。本レジスタの初期値は不定です。 (注意事項) <ul style="list-style-type: none">変換データ保護機能があります。A/D 変換中に本ビットにデータを書き込まないようにしてください。D9, D8 は , 8 ビット分解能を選択したときは , "0" が読み出されます。

16.4.5 アナログ入力制御レジスタ (AICR)

アナログ入力制御レジスタは、アナログ入力を制御するレジスタです。

■ アナログ入力制御レジスタ (AICR: AICR1, AICR2)

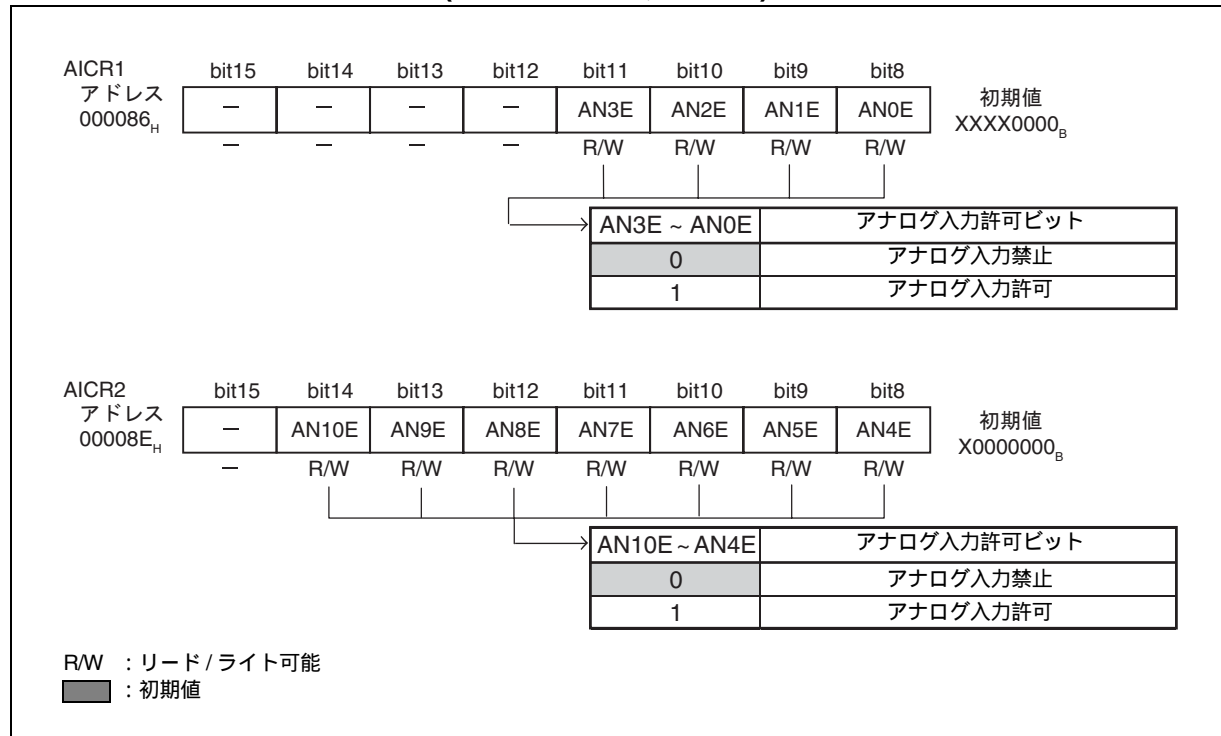


表 16.4-5 アナログ入力制御レジスタ (AICR) の各ビットの機能説明

ビット名		機能
(AICR1) bit11 ~ bit8 (AICR2) bit14 ~ bit8	AN3E ~ AN0E, AN10E ~ AN4E : アナログ入力許可 ビット	<ul style="list-style-type: none"> このビットが "0" の場合、アナログ入力は禁止されます。 このビットが "1" の場合、アナログ入力が許可されます。 アナログ入力端子として使用する端子は、対応する AICR レジスタのビットを "1" に設定してください。このときの PDR レジスタの読出し値は "0" になります。
(AICR1) bit15 ~ bit12 (AICR2) bit15	未定義ビット	<ul style="list-style-type: none"> 読出し値は不定です。 このビットへの書込みは、動作に影響しません。

16.5 割込み

8/10 ビット A/D コンバータは、A/D 変換で A/D データレジスタにデータがセットされることで、割込み要求を発生させることができます。

■ 8/10 ビット A/D コンバータの割込み

8/10 ビット A/D コンバータの割込み制御ビットと割込み要因は、表 16.5-1 のようになっています。

表 16.5-1 8/10 ビット A/D コンバータの割込み制御ビットと割込み要因

	8/10 ビット A/D コンバータ
割込み要求フラグビット	ADCS: INT
割込み要求許可ビット	ADCS: INTE
割込み要因	A/D 変換結果の A/D データレジスタへの書込み

A/D 変換動作が起動され、A/D 変換結果が A/D データレジスタ (ADCD) にセットされると、A/D 制御ステータスレジスタ (ADCS) の INT ビットが "1" にセットされます。このとき、割込み要求が許可 (ADCS: INTE=1) されていると、割込みコントローラに割込み要求を出力します。

16.6 動作説明

8/10 ビット A/D コンバータには、単発変換モード、連続変換モード、停止変換モードの 3 種類のモードがあります。各モードでの動作について説明します。

■ 単発変換モードの動作

単発変換モードは、ANS ビットと ANE ビットで設定されたアナログ入力を順次変換していき、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 変換は停止します。開始チャンネルと終了チャンネルが同じ (ANS=ANE) ときは ANS ビットで指定した 1 チャンネルだけの変換となります。単発変換モードで動作させるには、図 16.6-1 の設定が必要です。

図 16.6-1 単発変換モードでの設定

図 10.0-1 半角変換レジスタの設定

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADCH/ ADMD	-	-	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	MD1	MD0	STS1	STS0	CT1	CT0	ST1	ST0
ADCS	BUSY	INT	INTE	PAUS	S10	-	START	予約	0							
ADCD	-	-	-	-	-	-	変換データを格納									
AICR																

：使用ビット

：使用する端子の対応するビットに "1" を設定

0 ： "0" を設定

< 参考 >

単発変換モードでの変換順序の例を以下に示します。

ANS=000_B, ANE=011_B のとき : AN0 AN1 AN2 AN3 終了

ANS=011_B, ANE=011_B のとき : AN3 終了

< 注意事項 >

- A/D ユニット 1 は, AN0 ~ AN3 の 4 チャンネル, A/D ユニット 2 は, AN4 ~ AN10 の 7 チャンネルとなっております。
- A/D ユニット 1 の ANE2, ANS2 には, 必ず, "0" を書き込んでください。
- A/D ユニット 2 の ANE0 ~ ANE2=1, ANS0 ~ ANS2=1 は, 設定禁止です。
- A/D ユニット 1, 2 では, 必ず, "ANS0 ANE0" となるように設定してください。
- A/D 変換モード選択ビット (MD1, MD0) を "00_B" に設定した場合は, A/D 変換中の再起動ができます。このモードではソフトウェア起動 (STS1, STS0=00_B) のみ設定可能です。再起動は下記の手順で行ってください。

1. INT ビットを "0" にクリアする

2. START ビットに "1" を, INT ビットに "0" を同時に書き込む

A/D 変換モード選択ビット (MD1, MD0) を "01_B" に設定した場合は, A/D 変換中の再起動はできません。

A/D 変換の再起動と終了が同時に発生した場合, 再起動は行われずに A/D 変換は終了し, データレジスタ (ADCR1/ADCR0) には "300_H" が格納されます。したがって, A/D 変換再起動と終了が同時に発生しないように再起動をかけてください。

■ 連続変換モードの動作

連続変換モードは、ANS ビットと ANE ビットで設定されたアナログ入力を順次変換していき、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 ビットで設定されたアナログ入力に戻り、A/D 変換動作を続けます。開始チャンネルと終了チャンネルが同じとき (ANS=ANE) は ANS で指定したチャンネルだけの変換を繰り返します。連続変換モードで動作させるには、図 16.6-2 に示す設定が必要です。

図 16.6-2 連続変換モードでの設定

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADCH/ ADMD	-	-	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	MD1	MD0	STS1	STS0	CT1	CT0	ST1	ST0
									1	0						
ADCS	BUSY	INT	INTE	PAUS	S10	-	START	予約								
																0
ADCD	-	-	-	-	-	-	変換データを格納									
AICR																

: 使用ビット
 : 使用する端子の対応するビットに "1" を設定
 0 : "0" を設定
 1 : "1" を設定

< 参考 >

連続変換モードでの変換順序の例を以下に示します。

ANS=000_B, ANE=011_B のとき : AN0 AN1 AN2 AN3 AN0 繰返し

ANS=011_B, ANE=011_B のとき : AN3 AN3 繰返し

< 注意事項 >

- A/D ユニット 1 は、AN0 ~ AN3 の 4 チャンネル、A/D ユニット 2 は、AN4 ~ AN10 の 7 チャンネルとなっています。
- A/D ユニット 1 の ANE2, ANS2 には、必ず、"0" を書き込んでください。
- A/D ユニット 2 の ANE0 ~ ANE2=1, ANS0 ~ ANS2=1 は、設定禁止です。
- A/D ユニット 1, 2 では、必ず、"ANS0 ANE0" となるように設定してください。

■ 停止変換モードの動作

停止変換モードは、ANS ビットと ANE ビットで設定されたアナログ入力を 1 チャネルごとに一時停止しながら変換していき、ANE ビットで設定された終了チャネルまで変換が終わると ANS ビットで設定されたアナログ入力に戻り、A/D 変換と一時停止の動作を続けます。開始チャネルと終了チャネルが同じとき (ANS=ANE) は ANS ビットで指定したチャネルだけの変換を繰り返します。一時停止時の変換の再起動は、STS1、STS0 ビットで指定した起動要因を発生します。停止変換モードで動作させるには、図 16.6-3 の設定が必要です。

図 16.6-3 停止変換モードでの設定

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0

ADCH/ ADMD	-	-	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0	MD1	MD0	STS1	STS0	CT1	CT0	ST1	ST0
---------------	---	---	------	------	------	------	------	------	-----	-----	------	------	-----	-----	-----	-----

1 1

ADCS

BUSY	INT	INTE	PAUS	S10	-	START	予約
------	-----	------	------	-----	---	-------	----

0

ADCD

-	-	-	-	-	-	変換データを格納										
---	---	---	---	---	---	----------	--	--	--	--	--	--	--	--	--	--

AICR

--	--	--	--	--	--	--	--

：使用ビット
 ：使用する端子の対応するビットに "1" を設定
 0 ： "0" を設定
 1 ： "1" を設定

< 参考 >

停止変換モードでの変換順序の例を以下に示します。

ANS=000_B, ANE=011_B のとき: AN0 一時停止 AN1 一時停止 AN2 一時停止
AN0 繰返し

ANS=011_B, ANE=011_B のとき : AN3 一時停止 AN3 一時停止 繰返し

< 注意事項 >

- A/D ユニット 1 は, AN0 ~ AN3 の 4 チャンネル, A/D ユニット 2 は, AN4 ~ AN10 の 7 チャンネルとなっています。
- A/D ユニット 1 の ANE2, ANS2 には, 必ず, "0" を書き込んでください。
- A/D ユニット 2 の ANE0 ~ ANE2=1, ANS0 ~ ANS2=1 は, 設定禁止です。
- A/D ユニット 1, 2 では, 必ず, "ANS0 ANE0" となるように設定してください。

16.7 A/D 変換データ保護機能

割込み許可状態で A/D 変換を実行すると、変換データ保護機能が働きます。

■ A/D 変換データ保護機能

A/D コンバータでは、変換データ格納用のデータレジスタが 1 つしかないので、A/D 変換をすると、変換終了時にデータレジスタ内の格納データを書き換えます。そのため、変換データのメモリへの転送が間に合わないと同前のデータが一部欠落します。この対策として、割込み許可 (INTE=1) のときは、以下のようにデータ保護機能が働くようになっています。

変換データが A/D データレジスタ (ADCD) に格納されると、A/D 制御ステータスレジスタ (ADCS) の INT ビットが "1" にセットされます。この INT ビットが "1" の間、次の変換終了後も ADCD への変換データの格納は行われず、PAUS ビットを セットし、A/D は一時停止状態になります。一時停止中、直前の変換データを保持しています。一時停止状態の解除は、INT ビットのクリアにより行われます。一時停止状態解除後、保持していた変換データを ADCD に格納し、次の動作に移行します。

< 注意事項 >

- 変換データ保護機能は、割込み許可 (ADCS : INTE=1) 状態でしか動作しません。
 - 一時停止中に再起動をかけると待機データが壊れます。
-

16.8 使用上の注意

8/10 ビット A/D コンバータを使用するときの注意点を示します。

■ 8/10 ビット A/D コンバータ使用上の注意

● アナログ入力端子

A/D 入力端子はポートの入出力端子と兼用になっており、ポート方向レジスタ (DDR) とアナログ入力許可レジスタ (AICR) で切り換えて使用するようになっています。アナログ入力として使用する端子では、DDR の対応するビットに "0" を書き込んでポート設定を入力にしたうえで、AICR レジスタでアナログ入力モード (AICRx=1) に設定して、ポート側の入力ゲートを固定してください。ポート入力モード (AICRx=0) の状態では中間レベルの信号が入力されると、ゲートに入力リーク電流が流れます。

● 内部タイマで使用する時の注意

A/D コンバータを内部タイマで起動するとき、A/D 制御ステータスレジスタ (ADMD) の STS1, STS0 ビットで設定しますが、このとき内部タイマの入力値は、インアクティブ側 (内部タイマのときは "L") にしておいてください。アクティブ側にしておくと、ADMD レジスタへの書き込みと同時に動作し始めるときがあります。

● A/D コンバータの電源・アナログ入力の投入順序

A/D コンバータの電源 (AVcc, AVRH1, AVRH2, AVss)、アナログ入力 (AN0 ~ AN10) への印加は、必ずデジタル電源 (Vcc) の投入後、または同時に投入してください。また、電源切断時は、A/D コンバータの電源およびアナログ入力の切断後にデジタル電源 (Vcc) を切断、または同時に切断してください。

● A/D コンバータの電源電圧について

ラッチアップ防止のため、A/D コンバータの電源 (AVcc) は、デジタル電源 (Vcc) の電圧を超えないようにしてください。

● A/D ユニット 1, 2 の設定について

- A/D ユニット 1 は、AN0 ~ AN3 の 4 チャンネル、A/D ユニット 2 は、AN4 ~ AN10 の 7 チャンネルとなっております。
- A/D ユニット 1 の ANE2, ANS2 には、必ず、"0" を書き込んでください。
- A/D ユニット 2 の ANE0 ~ ANE2=1, ANS0 ~ ANS2=1 は、設定禁止です。
- A/D ユニット 1, 2 では、必ず、"ANS0 ANE0" となるように設定してください。

● A/D 変換の再起動について

A/D 変換モード選択ビット (MD1, MD0) を "00_B" に設定した場合は, A/D 変換中の再起動ができます。このモードではソフトウェア起動 (STS1, STS0=00_B) のみ設定可能です。再起動は下記の手順で行ってください。

1. INT ビットを "0" にクリアする
2. START ビットに "1" を, INT ビットに "0" を同時に書き込む

A/D 変換モード選択ビット (MD1, MD0) を "01_B" に設定した場合は, A/D 変換中の再起動はできません。

A/D 変換の再起動と終了が同時に発生した場合, 再起動は行われずに A/D 変換は終了し, データレジスタ (ADCR1/ADCR0) には "300_H" が格納されます。したがって, A/D 変換再起動と終了が同時に発生しないように再起動をかけてください。

第17章

積和演算回路

積和演算回路の概要，レジスタの構成 / 機能および動作について説明します。

- 17.1 特長
- 17.2 命令定義
- 17.3 レジスタ説明
- 17.4 動作説明
- 17.5 命令詳細説明

17.1 特長

積和演算回路の特長，レジスタ一覧およびブロックダイアグラムについて説明します。

■ 特長

- 高速積和演算 (1 システムクロックサイクル)
- データ形式 : 16 ビット固定小数点 (16 × 16 + 40 ビット)
- 命令領域 : 256 × 6 ビット
- データ領域 : 64 × 16 ビット × 2 組
- 丸め処理可能
- 飽和処理可能
- 加算項目数 : 最大 64 項
- 命令 : MAC 命令 , STR 命令 , JMP 命令
- 遅延処理 : 64 × 16 ビット内で自由に転送可能
- 固定小数点方式 : Q8 ~ Q15 のうちから選択可能
- プログラム実行制御 : 8 種類の命令を選択可能
- 変数モニタ : 8 × 16 ビットまでの計算結果をプログラム停止させずにモニタ可能

■ レジスタ一覧

図 17.1-1 積和演算回路のレジスタ一覧

	bit15	bit8	bit7	bit0	
アドレス :00039E _H	予約領域			アクセス禁止	
アドレス :0003A0 _H	DSP-PC (プログラムカウンタ)			DSP-CSR (コントロール / ステータス)	R/W,R,W
アドレス :0003A2 _H	DSP-LY(遅延レジスタ) 上位			DSP-LY(遅延レジスタ) 下位	R/W
アドレス :0003A4 _H	DSP-OT0(出力キュー 0) 上位			DSP-OT0(出力キュー 0) 下位	R
アドレス :0003A6 _H	DSP-OT1(出力キュー 1) 上位			DSP-OT1(出力キュー 1) 下位	R
アドレス :0003A8 _H	DSP-OT2(出力キュー 2) 上位			DSP-OT2(出力キュー 2) 下位	R
アドレス :0003AA _H	DSP-OT3(出力キュー 3) 上位			DSP-OT3(出力キュー 3) 下位	R
アドレス :0003AC _H	予約領域			予約領域	アクセス禁止
アドレス :0003AE _H	予約領域			予約領域	アクセス禁止
アドレス :0003B0 _H	DSP-OT4(出力キュー 4) 上位			DSP-OT4(出力キュー 4) 下位	R
アドレス :0003B2 _H	DSP-OT5(出力キュー 5) 上位			DSP-OT5(出力キュー 5) 下位	R
アドレス :0003B4 _H	DSP-OT6(出力キュー 6) 上位			DSP-OT6(出力キュー 6) 下位	R
アドレス :0003B6 _H	DSP-OT7(出力キュー 7) 上位			DSP-OT7(出力キュー 7) 下位	R

アドレス:		積和演算 マクロ	アクセス
00C000 _H	X-RAM (係数 RAM) ... 64 × 16 ビット	00 _H	R/W
:		:	
00C07E _H		3F _H	
アドレス:		積和演算 マクロ	アクセス
00C080 _H	Y-RAM (変数 RAM) ... 64 × 16 ビット	00 _H	R/W
:		:	
00C0FE _H		3F _H	
アドレス:		積和演算 マクロ	アクセス
00C100 _H	I-RAM (命令 RAM) ... 256 × 16 ビット	00 _H	R/W
:		:	
00C2FE _H		FF _H	

< 注意事項 >

- CPU から上記レジスタ / RAM 領域への書込みは、必ず偶数番地へのハーフワード (またはワード) 転送命令で行ってください。
- RAM 領域への書込み、読出しそのものは、バイト転送命令でも CPU より可能ですが、積和演算回路として使用時は 1 項目目の注意事項を守ってください。

■ ブロックダイアグラム

図 17.1-2 積和演算回路のブロックダイアグラム

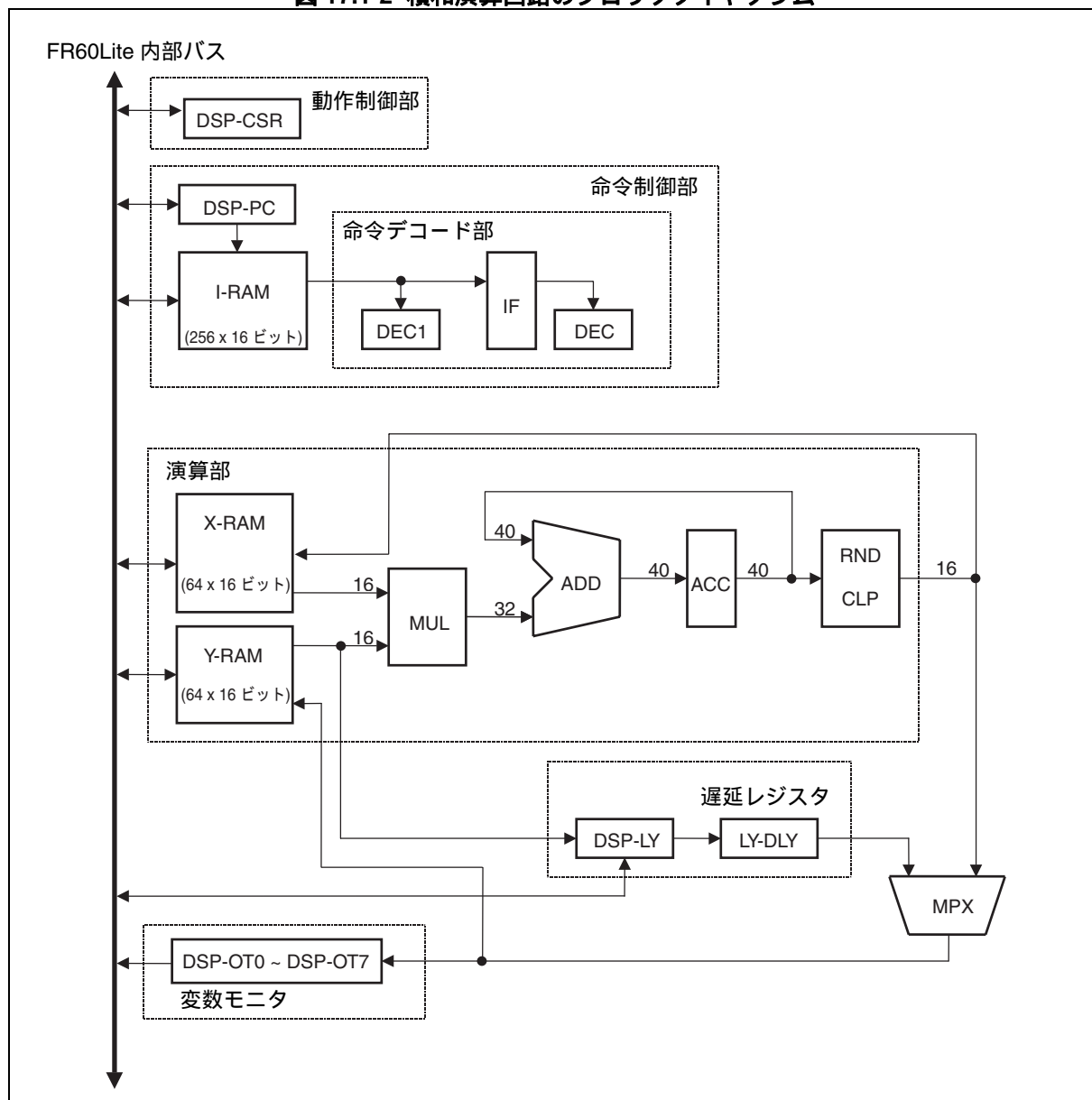


表 17.1-1 ブロックダイアグラム概略説明

ブロック	レジスタ	機能
動作制御	DSP-CSR	積和演算マクロの動作制御レジスタ。CPU および servo block から以下の動作を制御します。 <ul style="list-style-type: none"> ・ 計算開始 / 終了指示 ・ 割込み制御 ・ プログラムフロー制御 (積和演算マクロの条件付き分岐命令に使用)
命令制御	DSP-PC	プログラムカウンタ CPU から設定した先頭番地からプログラム実行を開始します。
	I-RAM	256 × 16 ビットの命令 RAM です。 積和演算マクロが計算停止中, CPU から読出しおよび書込みができます。 計算開始前に CPU から命令コードをロードしてください。
	IF	命令フェッチレジスタ
	DEC1* DEC*	命令デコーダ
演算部	X-RAM	64 × 16 ビットのデータ RAM です。 積和演算マクロが計算停止時, CPU から読出しおよび書込みができます。 計算開始前に CPU から係数をロードしてください。
	Y-RAM	64 × 16 ビットのデータ RAM です。 積和演算マクロが計算停止時, CPU から読出しおよび書込みができます。 計算開始前に CPU から変数をロードしてください。
	MUL*	16 × 16 32 ビット乗算器です。
	ADD*	32 + 40 40 ビット加算器です。
	ACC*	40 ビットのアキュムレータです。
	CLP* RND* SLQ*	40 16 ビット転送時, 範囲外の 16 ビットデータ時に最大値に飽和します。 40 16 ビット転送時, 下位ビットの丸め処理を行います。 40 16 ビット転送時, 転送ビットを選択します。
遅延レジスタ	DSP-LY LY-DLY*	遅延レジスタ 積和演算時, 変数値を保持し, Y-RAM に書き戻すことができます。
変数モニタ出力	DSP-OT0 ~ DSP-OT7	変数モニタ出力レジスタ。 Y-RAM の 0 ~ 7 番地の内容と同じ値を保持しています。 計算中 (Y-RAM のアクセス禁止中) に Y-RAM の 0 ~ 7 番地の値をモニタできます。

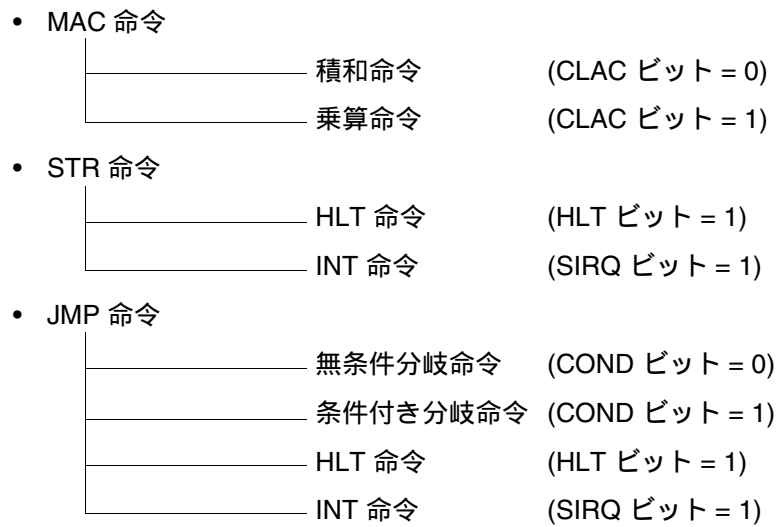
*: CPU からアクセス不可

17.2 命令定義

積和演算マクロが持っている命令は、大きく分けて 3 種類 (MAC/STR/JMP 命令) あります。

■ 命令定義

積和演算マクロの 3 種類以外の命令を表記上使用していますが、その場合下記に示す命令の階層構成となります。



17.3 レジスタ説明

積和演算回路で使用するレジスタの構成および機能について説明します。

■ DSP コントロール / ステータスレジスタ (DSP-CSR)

コントロール / ステータスレジスタは 8 ビット長のレジスタであり、積和演算マクロのステート切換え / CPU への割込みのコントロールおよび積和演算マクロの状態を示す各種フラグより構成されています。また、積和演算マクロの条件付き分岐命令の条件設定も本レジスタで設定を行います。

- 8 ビットレジスタで外部より常に読出しおよび書込みが可能

● コントロール機能

- 積和演算マクロのステート (計算の開始 / 停止) の遷移 (GoDSP and HltDSP)
- CPU に対する割込みマスク (IeDSP)
- 積和演算マクロの条件付き分岐命令の条件設定 (USR2, USR1, USR0)

● ステータス機能

- 積和演算マクロの現在のステート取得フラグ (RunDSP)
- 割込み要求フラグ (IrqDSP)
- 飽和处理フラグ (SatDSP)

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0003A1 _H	-	USR2	USR1	USR0	IrqDSP	IeDSP	HltDSP	GoDSP	00000000 _B
	SatDSP	USR2	USR1	USR0	IrqDSP	IeDSP	-	RunDSP	
	R	R/W	R/W	R/W	R/W	R/W	W	R/W	

R/W : リード / ライト可能
 R : リードオンリ
 W : ライトオンリ
 - : 未定義

[bit7] **SatDSP** (飽和处理フラグ) : Read only

- 計算途中で飽和处理を行ったことを保持するステータスフラグです。
- STR 命令で飽和处理を指示し (CLP = 1), 実際に飽和处理が行われた場合にセットされます。また計算中一度セットされると次の計算開始まで値を保持しています。
- 本ビットは計算開始によりクリアされます。
 セット要因 : 計算途中で STR 命令による飽和处理が行われた場合セット
 クリア要因 : 計算開始によりクリア [初期値]
- リセット時 : "0" に初期化されます (飽和处理なし)。
- 読出しのみ可能です。書込みしてもビット値は変化しません。

[bit6, bit5, bit4] **USR2, USR1, USR0** (ジャンプ条件設定ビット) : Read/Write

- 本ビットは、積和演算マクロの条件付き分岐命令 (COND ビット = 1 時) で参照され、本ビット値と条件付き分岐命令の UBP フラグが一致 (条件成立) するとジャンプします。すなわち、この条件付き分岐命令と計算命令を組み合わせることで 8 種類の計算ルーチンを CPU 側から切換え可能です。
- リセット時 : 000_B に初期化されます。
- 読出しおよび書込みが可能です。

[bit3] **IrqDSP** (割込み要求フラグ) : Read/Write

- 積和演算マクロのソフトウェア割込み要求が発生したことを示すフラグです。割込み要求が許可されているとき (IeDSP = 1) に、本ビットがセットされると CPU に対して割込み要求が発生します。
- 積和演算マクロの割込み要求は STR 命令 / JMP 命令の SIRQ ビットを "1" にすることで、ソフトウェア的に発生させます。

セット要因 : 積和演算マクロのソフトウェア割込み (STR 命令 / JMP 命令) 発生により
リセット

クリア要因 : "0" 書込みによりクリア [初期値]

- リセット時 : "0" に初期化されます (割込み要求なし)。
- 読出しおよび書込みが可能です。ただし、書込みは "0" のみ可能で "1" を書き込んでもビット値は変化しません。
- リードモディファイライト (RMW) 命令における読出し値は、ビット値にかかわらず常に "1" です。

[bit2] **IeDSP** (割込み要求許可ビット) : Read/Write

CPU に対する割込み要求 (IrqDSP = 1) を以下のように制御します。

"0": 割込み要求出力禁止 (IrqDSP がセットされても割込み発生せず) [初期値]

"1": 割込み要求出力許可 (IrqDSP がセットされると割込み発生する)

- リセット時 : "0" に初期化されます (割込み要求出力禁止)。
- 読出しおよび書込みが可能です。

[bit1] **HitDSP** (計算停止) : Write only

- 計算の強制停止ビットです。
- 本ビットに "1" を書き込むと、計算実行中 (RunDSP = 1) であれば実行中の命令が終了後 (2 サイクルであれば 2 サイクル後) 計算を停止し RunDSP フラグがクリアされます。
- 計算停止中の場合、何も影響を与えません。
- また、本ビットにより強制停止させた場合、DSP-PC は停止した命令の次の命令アドレスを指していますので、命令の継続実行が可能です。
 - "1" : 書込みにより強制停止
 - "0" : 書込みは無効。読出しは常に "0"
- リセット時 : "0" に初期化されます。

[bit0] **GoDSP** (計算開始) : Write only

RunDSP (計算実行中フラグ) : Read only

- GoDSP ビットに "1" を書き込むことにより, 計算の開始を指示します。計算停止中 (RunDSP = 0) であれば, 計算の起動となり, RunDSP フラグがセットされます。既に計算実行中 (RunDSP = 1) であれば, 何も影響を 与えません。
- RunDSP フラグは, 計算を実行中であることを示します。計算の開始でセットされ, HltDSP ビットへの "1" 書込み, または 積和演算マクロの HLT 命令実行でクリアされます。
- 計算実行中 (RunDSP = 1) は, DSP-PC, DSP-LY, X-RAM, Y-RAM, I-RAM は CPU からアクセスできません。DSP-CSR, DSP-OT0 ~ DSP-OT7 のみがモニタできます。
- 計算を開始するには, 起動と同時またはそれ以前に DSP-PC へ計算ルーチンの先頭アドレスを格納する必要があります。
 - 書込み時機能 (GoDSP: 計算開始)
 - "0": 機能なし / 動作に影響なし
 - "1": 計算停止中の場合 計算開始
計算実行中の場合 何も影響がありません
 - 読出し時機能 (RunDSP: 計算実行中フラグ)
 - "0": 計算を停止中である [初期値]
クリア要因 HltDSP"1" 書込みおよび HLT 命令実行
 - "1": 計算を実行中である
セット要因 計算開始
- リセット時 : 0 に初期化されます (計算停止中)。
- 読出しおよび書込みが可能です。ただし, 書込み時と読出し時では, 上記のように意味が異なります。
- リードモディファイライト (RMW) 命令における読出し値は, ビット値にかかわらず常に "0" です。

■ DSP プログラム・カウンタ (DSP-PC)

プログラム・カウンタは 8 ビット長のカウンタであり、積和演算マクロで実行する命令コードが格納されているメモリ (I-RAM) アドレスを示しています。プログラムカウンタは命令の実行により自動的に更新されますが、積和演算マクロの JMP により書き換えることができます。

また、計算停止中でのみ CPU よりアクセス (R/W) 可能であり、計算開始と同時にまたはそれ以前に DSP-PC へ計算ルーチンの先頭アドレスを格納する必要があります。

HLT 命令実行後、または DSP-CSR の HltDSP へ "1" 書込み後、DSP-PC は停止した命令の次のアドレスを指しており、再度 GoDSP をセットすることにより継続してプログラムを実行できます。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	初期値
0003A0 _H	-	-	-	-	-	-	-	-	XXXXXXXX _B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード / ライト可能
 X : 不定
 - : 未定義

- リセット時：不定となります。
- 読出しおよび書込みが可能です。積和演算マクロが計算停止中 (DSP-CSR:RunDSP=0) でのみアクセス可能です。
- 計算実行中 (DSP-CSR:RunDSP = 1) では、バスから切り離されるため CPU からのアクセスはできなくなります。

■ DSP 遅延レジスタ (DSP-LY)

DSP-LY は 16 ビット長のレジスタであり、積和演算マクロの MAC 命令の遅延書込みビット (LDLY) が "1" のときに使用されます。計算中 (DSP-CSR : RunDSP = 1) はアクセスができません。

- MAC 命令の LDLY ビットが "1" のときは、次の 2 つの動作を順次実行します。

DSP-LY レジスタの内容を、LY-DLY レジスタに転送する。

MAC 命令で選択した Y-RAM の読出しデータを DSP-LY レジスタに格納する。

- MAC 命令の STLY ビットが "1" のときは、MAC 命令実行後に LY-DLY レジスタの値を MAC 命令で選択した Y-RAM のアドレスへ書き込みます。このとき、実行時間は 2 サイクルになります。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
0003A2 _H																	XXXXXXXX _B
	R/W		R/W		R/W		R/W		R/W		R/W		R/W		R/W		
	X		X		X		X		X		X		X		X		

R/W : リード / ライト可能
 X : 不定
 - : 未定義

- リセット時：不定となります。
- 読出しおよび書込みが可能です。ただし、DSP-LY が計算停止中 (DSP-CSR : RunDSP = 0) のみアクセスが可能です。計算実行中 (DSP-CSR : RunDSP = 1) では、バスから切り離されるため CPU からのアクセスはできなくなります。

■ DSP 変数モニタレジスタ (DSP-OT0 ~ DSP-OT7)

変数モニタレジスタとして、16 ビットのレジスタが 8 本あります。電源投入時の初期状態を除き、Y-RAM の 0 ~ 7 番地と同じ内容を保持しています。CPU より常に読出しのみ可能であり、計算中であっても Y-RAM の 0 ~ 7 番地の内容をモニタできます。

アドレス	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
DSP-OT0 0003A4 _H																
DSP-OT1 0003A6 _H	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
DSP-OT2 0003A8 _H																
DSP-OT3 0003AA _H																
DSP-OT4 0003B0 _H																
DSP-OT5 0003B2 _H																
DSP-OT6 0003B4 _H																
DSP-OT7 0003B6 _H																

初期値 : XXXXXXXX XXXXXXXX_B

R : リードオンリ
X : 不定

- リセット時：不定となります。
- 常に読出しのみ可能です。積和演算マクロがプログラム実行中であっても読出し可能です。

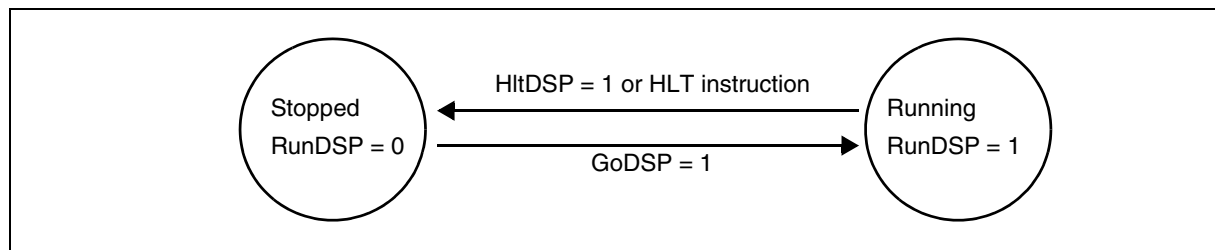
17.4 動作説明

積和演算回路の動作および機能について説明します。

■ 動作モード

積和演算マクロの動作は、DSP-CSR レジスタの操作により制御されます。

積和演算マクロの状態としては、以下の 2 状態があり、停止状態で GoDSP ビットに "1" を書き込むか Servo Block から GODSPSV 信号を入力することにより、積和演算マクロはプログラム実行を開始します。また、停止中と計算中とでは CPU からアクセス可能なレジスタおよびメモリに違いがあります。



各状態について説明します。

- 停止中： 積和演算マクロは停止状態です。
CPU から命令 RAM (I-RAM)、データ RAM (X-RAM, Y-RAM) および、積和演算マクロの全レジスタがアクセスできます。
HltDSP に "1" を書き込むか、HLT 命令の実行により、本状態に遷移します。
また、システムリセット時は本状態に初期化されます。
- 計算中： 積和演算マクロは計算状態です。
停止状態から GoDSP ビットに "1" を書き込むと、本状態に遷移し現在の DSP-PC (プログラムカウンタ) からプログラム実行を開始します。
HltDSP ビットに "1" を書き込むか、HLT 命令が実行されると、停止状態に遷移しプログラム実行を停止します。
CPU からは、DSP-CSR、DSP-OT0 ~ DSP-OT7 レジスタのみアクセスできます (その他のレジスタおよび RAM はアクセス禁止*)。

※: アクセス禁止ですが、R/W した場合は以下ようになります。

書込み時 何も影響ありません (書き込みされません)。
読出し時 不定です。

■ 命令動作

DSP-CSR レジスタの GoDSP ビットに "1" を書き込むと、積和演算マクロは現在の DSP-PC (プログラムカウンタ) から命令実行を開始します (CPU の動作と並列して動作します)。

実行に先立って、I-RAM および DSP-PC の値を設定してください (DSP-CSR と DSP-PC は同時設定が可能です)。

積和演算マクロの命令実行が開始されると、以下の動作制御が行われます。

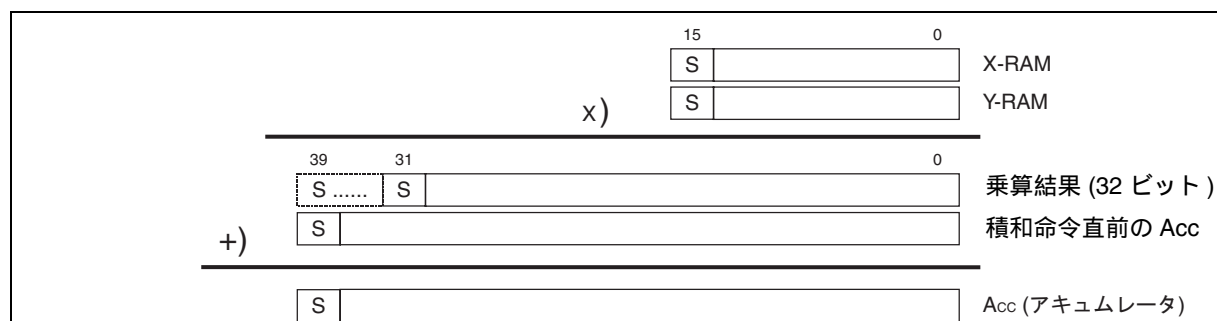
- 積和演算マクロで HLT 命令*を実行すると、その命令実行が終了後、停止状態に遷移します。
このとき DSP-PC は HLT 命令の次のアドレスを指した状態で停止します。

- JMP 命令や STR 命令実行時, CPU に割込み要求を発生させることができます (割込みマスク可能です)。
- DSP-CSR の USR0 ~ USR2 ビットを参照する条件付き分岐命令を使用し, プログラムフローの切換えを行います。

*: HLT 命令とは, JMP, STR 命令の HLT ビットが "1" である命令のことです。

■ 演算機能

積和演算マクロは, 2 組の 16 ビットデータ RAM (X-RAM and Y-RAM) を持ち, 積和 (および乗算) 命令実行時, 各組の RAM データを読み込んで符号付き積和 (および乗算) 演算を行い, 40 ビットのアキュムレータに格納します。以下にデータ形式を示します。



(注意事項) ・ 乗算命令の場合は, 乗算結果を 40 ビットに符号拡張した値がアキュムレータに格納されます (直前のアキュムレータの内容はゼロクリアされます)。

- ・ "S" は符号ビットを示します。

積和命令を数多く繰り返して, アキュムレータがオーバーフローした場合は結果が保証されません。

積和命令は連続して 512 回以上行わないでください。

■ 遅延書込み機能

積和 (および乗算) 命令実行時, 以下の転送動作も併せて行うことができます。本転送と演算処理を併用することで, デジタルフィルタにおけるデータの遅延処理が容易に実現できます。

- Y-RAM からの読出し値を DSP-LY レジスタに格納
- 命令実行前の DSP-LY レジスタ値を, LY-DLY レジスタを経由して Y-RAM のリードアドレスに遅延書込み

■ 演算結果の転送処理

アキュムレータに格納された計算結果を X-RAM /Y-RAM へ転送する場合, 16 ビット幅で転送されますが, その際, 以下に示す位取り処理が行われて転送されます。

- 出力ビット選択

40 ビットのアキュムレータのうち, 以下に示すビット幅を選択できます。

bit27 ~ bit12	(Q12 フォーマット)
bit28 ~ bit13	(Q13 フォーマット)
bit29 ~ bit14	(Q14 フォーマット)
bit30 ~ bit15	(Q15 フォーマット)
bit23 ~ bit8	(Q8 フォーマット)
bit24 ~ bit9	(Q9 フォーマット)
bit25 ~ bit10	(Q10 フォーマット)
bit26 ~ bit11	(Q11 フォーマット)

- 丸め処理

選択した出力ビットの LSB の直下のビット値を "0" 捨 "1" 入します。

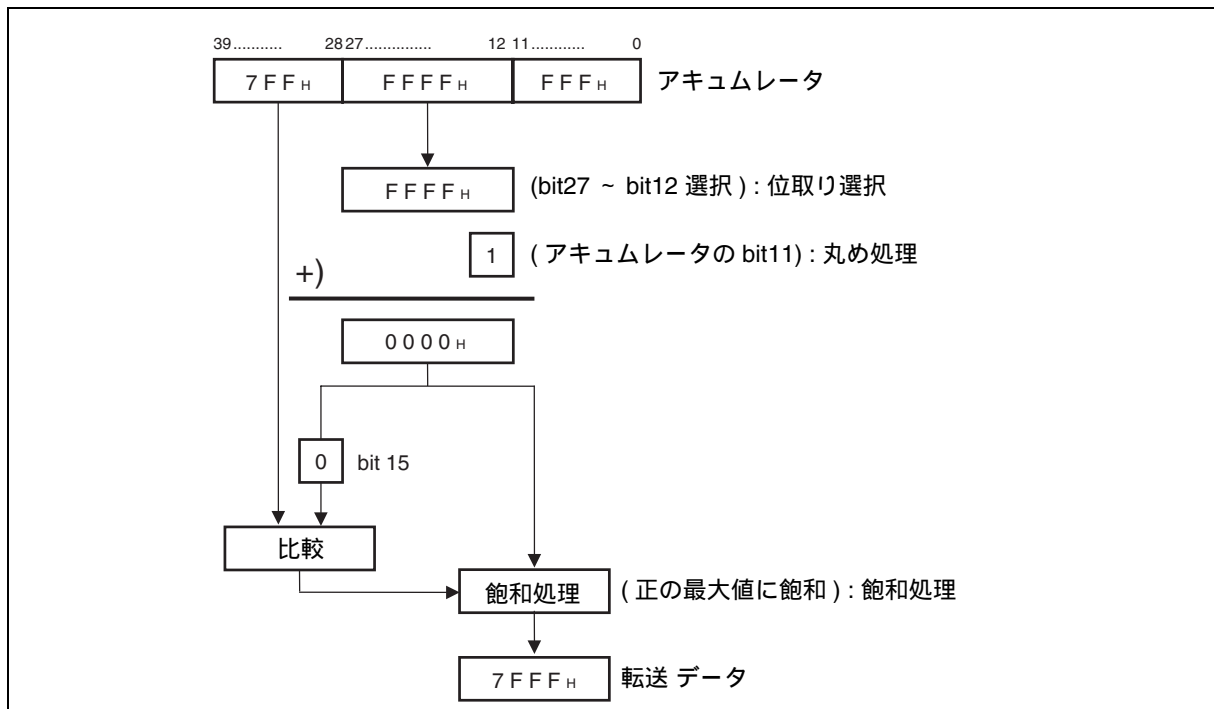
- 飽和処理

丸め後の 16 ビットデータの符号ビット (MSB) と、アキュムレータ内の上位ビットの比較をして異なるビットがある場合、飽和処理が行われます。飽和結果は、アキュムレータの符号ビット (MSB) によって以下の値になります。

アキュムレータの符号が "0" の場合 正の最大値 "7FFF_H" に飽和されます。

アキュムレータの符号が "1" の場合 負の最大値 "8000_H" に飽和されます。

以下に例を示します。



■ 変数モニタ出力

積和演算マクロは、Y-RAM の 0 ~ 7 番地の内容を常に保持しているレジスタ (DSP-OT0 ~ DSP-OT7) を持っています。Y-RAM の 0 ~ 7 番地にデータ書込みが行われた際 (CPU からの書込み, STR 命令による書込み, 遅延書込み), DSP-OT0 ~ DSP-OT7 レジスタにも同じ値が格納されます。

計算中は CPU からの Y-RAM へのアクセスが禁止されますが、CPU で参照したい計算結果を STR 命令で Y-RAM の 0 ~ 7 番地へ格納することによって CPU から常に計算結果を参照することができます。

< 注意事項 >

積和演算マクロにおいて DMA 転送を使用する場合は、

- CPU クロックは周辺クロックに対して同じか、もしくは速い設定にしてください。
- CPU クロックが周辺クロックよりも遅い場合、DMA 転送が正しく動作しません。

17.5 命令詳細説明

積和演算回路で使用する MAC 命令, STR 命令, JMP 命令の詳細について説明します。

■ MAC 命令

動作	: ACC	ACC + X data × Y data
	LY-DLY	DSP-LY
	DSP-LY	Y data (LDLY = 1)
	Y-RAM	LY-DLY (STLY = 1)
説明	: アキュムレータに X-RAM の X データと Y-RAM の Y データの乗算値を加算します。 同時に DSP-LY レジスタの内容を LY-DLY レジスタへ転送します。	
Word 数	: 1 word (16 ビット幅)	
サイクル 数	: 1 システムクロック サイクル (STLY = 1 のとき 2 サイクルになります。)	
動作コード	:	

bit15	bit14	bit13	bit12	bit11.....bit6	bit5.....bit0
1	CLAC	STLY	LDLY	X-Addr	Y-Addr

[bit14] CLAC (Clear Acc)

本ビットをセットすることにより乗算命令として機能します。

"0": Acc Acc + X data × Y data [積和命令]

"1": Acc 0 + X data × Y data [乗算命令]

[bit13] STLY (Store LY)

- ビットが "1" のときに以下の動作を行います。"0" の場合は演算のみ実行されません。
- 演算後, さらに LY-DLY レジスタの内容を, Y-RAM の Y-Addr 番地へ格納します。
- 本ビットをセットした場合のみ実行時間が 2 サイクルとなります。

[bit12] LDLY (Load LY)

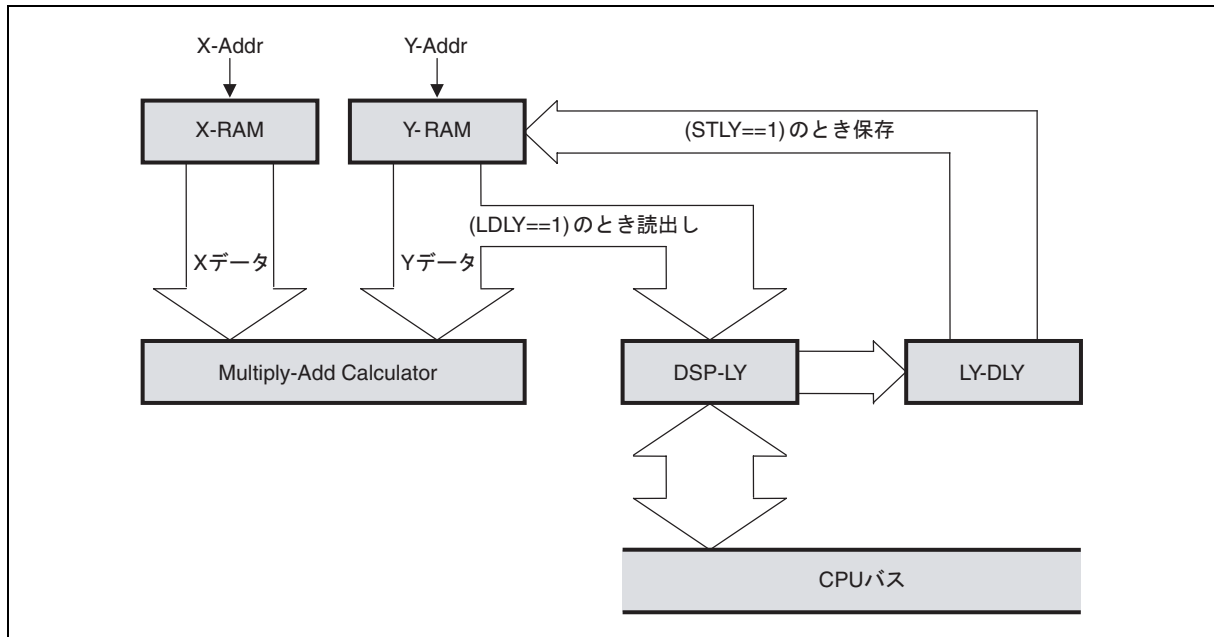
- ビットが "1" のときに以下の動作を行います。"0" の場合は演算のみ実行されません。
- 演算時に Y-RAM の Y-Addr 番地の内容を DSP-LY レジスタにも格納します。

[bit11 ~ bit6] X-Addr (X-RAM Address)

X-RAM 内の X データを指定するためのアドレス指定ビットです。

[bit5 ~ bit0] Y-Addr (Y-RAM Address)

- Y-RAM 内の Y データを指定するためのアドレス指定ビットです。



■ STR 命令 (転送命令)

動作 : Data RAM アキュムレータ

説明 : アキュムレータの 40 ビットデータを RND/CLP/SLQ フラグの指示により 16 ビットデータに変換し, SLY フラグと X/Y-Addr の指示するデータ RAM へ格納します。

Word 数 : 1 word (16 ビット幅)

サイクル数 : 1 システムクロック サイクル

動作コード :

bit15	bit14	bit13	bit12	bit11	bit10	bit9.....bit7	6	bit5.....bit0
0	1	HLT	SIRQ	RND	CLP	SLQ	SLY	X/Y-Addr

[bit13] HLT (HLT 命令指示フラグ)

本ビットをセットすることにより, 命令実行後に積和演算マクロはプログラム実行を停止します。

DSP-CSR レジスタの RunDSP フラグはクリアされます。

[bit12] SIRQ (INT 命令指示フラグ)

本ビットをセットすることにより, 命令実行後に CPU に対する割り込み要求を発生させ, DSP-CSR レジスタの IrqDSP フラグをセットします。

[bit11] RND (Rounding)

SLQ ビット指示された 16 ビットデータに対し, 丸め処理を指示するビットです。

丸め処理は 16 ビットデータの LSB より 1 つ下位のビットを 0 捨 1 入します。

[bit10] CLP (Clipping)

アキュムレータの計算結果が, SLQ ビットで指示された 16 ビットデータに対してオーバフローした値である場合, 16 ビットデータに対して飽和処理を指示するビットです。

実際にはアキュムレータの MSB(39 ビット) から 16 ビットデータの MSB(SLQ 指示) が同じ値でない場合に飽和処理が行われます。丸め処理の指定があった場合は, 丸め処理の結果に対して比較されます。

丸め処理前のアキュムレータの値が正のときは正の最大値 ($7FFF_H$), 負の場合は負の最大値 (8000_H) が転送されます。

丸め処理および飽和処理によりアキュムレータの符号は反転することなく, 符号は保存されます。

[bit9 ~ bit7] SLQ

アキュムレータからデータ RAM へ転送するビット位置を指定します。

SLQ ビット	オーバフロー判定ビット	転送 16 ビットデータ	丸めビット	固定小数点方式
0 0 0	bit39 ~ bit27	bit27 ~ bit12	bit11	Q12
0 0 1	bit39 ~ bit28	bit28 ~ bit13	bit12	Q13
0 1 0	bit39 ~ bit29	bit29 ~ bit14	bit13	Q14
0 1 1	bit39 ~ bit30	bit30 ~ bit15	bit14	Q15
1 0 0	bit39 ~ bit23	bit23 ~ bit8	bit7	Q8
1 0 1	bit39 ~ bit24	bit24 ~ bit9	bit8	Q9
1 1 0	bit39 ~ bit25	bit25 ~ bit10	bit9	Q10
1 1 1	bit39 ~ bit26	bit26 ~ bit11	bit10	Q11

[bit6] SLY

転送先を指定します。

"0": X-RAM

"1": Y-RAM

[bit5 ~ bit0] X/Y Addr (RAM Address)

データ RAM の直接アドレスを指定します。

■ JMP 命令 (分岐命令)

動作: [条件成立時] DSP-PC J-Addr8
 [条件不成立時] DSP-PC DSP-PC + 1

説明: 条件が成立したときに分岐し, 成立しなければ何も実行しません。

Word 数: 1 word (16 ビット幅)

サイクル数: 1 システムクロックサイクル

動作コード:

bit15	bit14	bit13	bit12	bit11	bit10...bit8	bit7.....bit0
0	0	HLT	SIRQ	COND	UBP2 ~ UBP0	J-Addr8

[bit13] HLT (HLT 命令指示フラグ)

本ビットをセットすることにより, 命令実行後に積和演算マクロはプログラム実行を停止します。

DSP-CSR レジスタの RunDSP フラグはクリアされます。

[bit12] SIRQ (INT 命令指示フラグ)

本ビットをセットすることにより, 命令実行後に CPU に対する割込み要求を発生させます。

DSP-CSR レジスタの IrqDSP フラグをセットします。

[bit11] COND (CONDition)

"0": 無条件分岐

"1": 条件付き分岐

[bit10 ~ bit8] UBP2 ~ UBP0 (条件指定)

条件付き分岐時の条件を設定します。DSP-CSR レジスタの USR2, USR1, USR0 ビットと本ビットが一致すると条件成立となります。

無条件分岐のときは, "000_B" に設定する必要があります。

[bit7 ~ bit0] J-Addr8 (Jump Address)

分岐先のアドレスを指定します。

第18章

DMAC

(DMA コントローラ)

DMAC の概要，レジスタの構成 / 機能および動作について説明します。

- 18.1 概要
- 18.2 レジスタ詳細説明
- 18.3 DMAC (DMA コントローラ) の動作
- 18.4 動作フローチャート
- 18.5 データパス

18.1 概要

DMAC は、DMA (Direct Memory Access) 転送を実現するためのモジュールです。DMA 転送により、CPU を介さずに各種データ転送を高速に行うことが可能となり、システムのパフォーマンスを向上させます。

■ ハードウェア構成

本モジュールは、主に以下の回路およびレジスタにより構成されます。

- 独立した DMA チャンネル × 5 チャンネル
- 5 チャンネル独立アクセス制御回路
- 20 ビットアドレスレジスタ (リロード指定可能: ch.0 ~ ch.3)
- 24 ビットアドレスレジスタ (リロード指定可能: ch.4)
- 16 ビット転送回数レジスタ (リロード指定可能: 各チャンネル 1 本)
- 4 ビットブロック回数レジスタ (各チャンネル 1 本)
- 2 サイクル転送

■ 主要機能

本モジュールによるデータ転送には以下の機能があります。

- 複数チャンネルの独立したデータ転送が可能 (5 チャンネル)
 1. 優先順位 (ch.0>ch.1>ch.2>ch.3>ch.4)
 2. ch.0 - ch.1 間にて順位回転が可能
 3. DMAC 起動要因
 - 内蔵周辺要求 (割込み要求を共用: 外部割込みを含む)
 - ソフトウェア要求 (レジスタ書込み)
 4. 転送モード
 - バースト転送 / ステップ転送 / ブロック転送
 - アドレッシングモード 20 ビット (24 ビット) アドレス指定 (増加 / 減少 / 固定)
(アドレス増減幅は ± 1, ± 2, ± 4 固定)
 - データの種類 バイト / ハーフワード / ワード長
 - シングルショット / リロード選択可能

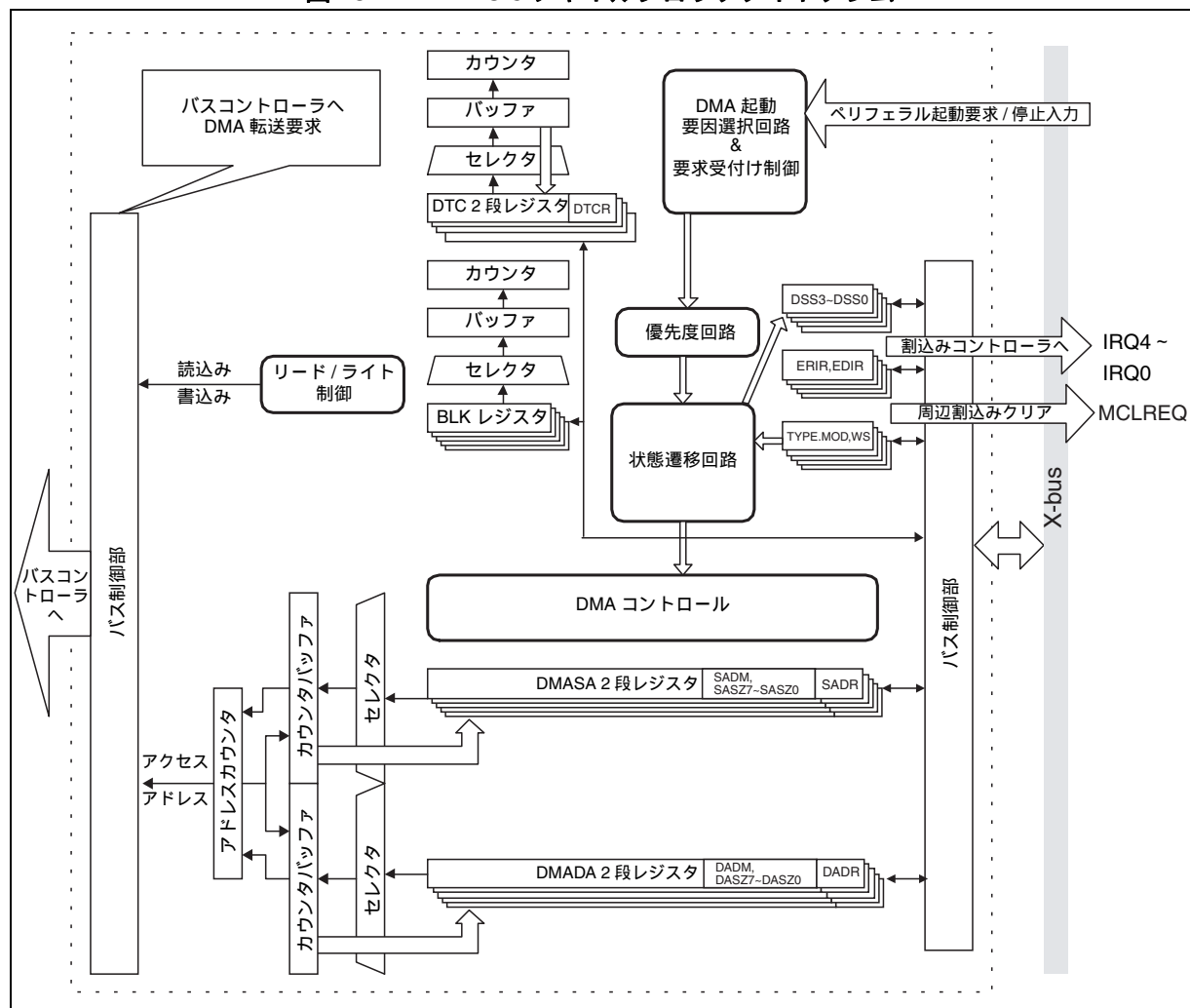
■ レジスタ概要

図 18.1-1 DMAC のレジスタ一覧

		bit 31 24 23 16 15 8 7 0
ch.0 コントロール / ステータスレジスタ A	DMACA0 000200 _H	
ch.0 コントロール / ステータスレジスタ B	DMACB0 000204 _H	
ch.1 コントロール / ステータスレジスタ A	DMACA1 000208 _H	
ch.1 コントロール / ステータスレジスタ B	DMACB1 00020C _H	
ch.2 コントロール / ステータスレジスタ A	DMACA2 000210 _H	
ch.2 コントロール / ステータスレジスタ B	DMACB2 000214 _H	
ch.3 コントロール / ステータスレジスタ A	DMACA3 000218 _H	
ch.3 コントロール / ステータスレジスタ B	DMACB3 00021C _H	
ch.4 コントロール / ステータスレジスタ A	DMACA4 000220 _H	
ch.4 コントロール / ステータスレジスタ B	DMACB4 000224 _H	
全体制御レジスタ	DMACR 000240 _H	
		bit 31 24 23 16 15 8 7 0
ch.0 転送元アドレスレジスタ	DMASA0 001000 _H	
ch.0 転送先アドレスレジスタ	DMADA0 001004 _H	
ch.1 転送元アドレスレジスタ	DMASA1 001008 _H	
ch.1 転送先アドレスレジスタ	DMADA1 00100C _H	
ch.2 転送元アドレスレジスタ	DMASA2 001010 _H	
ch.2 転送先アドレスレジスタ	DMADA2 001014 _H	
ch.3 転送元アドレスレジスタ	DMASA3 001018 _H	
ch.3 転送先アドレスレジスタ	DMADA3 00101C _H	
		bit 31 24 23 16 15 8 7 0
ch.4 転送元アドレスレジスタ	DMASA4 001020 _H	
ch.4 転送先アドレスレジスタ	DMADA4 001024 _H	

■ ブロックダイアグラム

図 18.1-2 DMAC 5 チャンネルブロックダイアグラム



18.2 レジスタ詳細説明

DMA コントローラで使用するレジスタの設定時の注意事項とレジスタの詳細について説明します。

■ レジスタ設定時の注意

本 DMAC の設定をする場合、DMA が停止しているときに行う必要があるビットがあります。動作中（転送中）に設定した場合には正常な動作は保証されません。

* マークは DMAC 転送中に設定すると動作に影響するビットです。このビットの書換えは DMAC 転送停止中（起動禁止状態または一時停止状態）に行ってください。

DMA 転送の起動禁止状態 (DMACR:DMAE=0 または DMACA:DENB=0 のとき) で設定した場合は、起動許可後に設定は有効になります。

DMA 転送の一時停止状態 (DMACR:DMAH3 ~ DMAH0] 0000 または DMACA:PAUS=1 のとき) で設定した場合は、一時停止解除後に設定は有効になります。

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 コントロール / ステータスレジスタ A [DMACA: DMACA0 ~ DMACA4]

DMAC 各チャネルの動作制御を行うレジスタで、チャネルごとに独立して存在します。各ビット機能は以下に示すとおりです。

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0 :000200 _H	DENB	PAUS	STRG	IS4~IS0				-				BLK3~BLK0				
ch.1 :000208 _H	R/W	R/W	R/W	R/W								R/W				
ch.2 :000210 _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ch.3 :000218 _H	DTC15~DTC0															
ch.4 :000220 _H	R/W															

(初期値 : 00000000 00000000 00000000 00000000_B)

[bit31] DENB (Dma ENaBle) : DMA 動作許可ビット

各転送チャネルに対応し、DMA 転送の起動許可 / 禁止を行います。

起動されたチャネルは、転送要求が発生し受け付けられると DMA 転送を開始します。

起動許可されていないチャネルに対して発生した転送要求は、すべて無効となります。

起動されたチャネルの転送が指定回数分すべて終了した場合、本ビットは "0" になり、転送は停止します。

本ビットに "0" を書き込むと強制停止しますが、必ず PUAS ビット [DMACA: bit30] で DMA を一時停止状態した後強制停止 ("0" 書込み) してください。一時停止しないで強制停止した場合は、DMA は停止しますが転送データは保証されません。停止の確認は DSS2 ~ DSS0 ビット (DMACB:bit18 ~ bit16) で行ってください。

DENB	機能
0	対応チャネル DMA 動作禁止 (初期値)
1	対応チャネル DMA 動作許可

- ・ リセット時、停止要求が受け付けられた場合："0" に初期化されます。
- ・ 読出しおよび書込みが可能です。
- ・ DMAC 全体制御レジスタ DMACR の bit15:DMAE により、全チャネルの動作が禁止されている場合、本ビットへの "1" 書込みは無効となり、停止状態を維持します。また、本ビットにより動作が許可されている状態にて前記ビットにより動作が禁止された場合、本ビットは "0" となり、転送は中断します（強制停止）。

[bit30] PAUS (PAUSE)：一時停止指示

対応するチャネルの DMA 転送の一時停止制御を行います。本ビットがセットされると、再び本ビットがクリアされるまでの間は、DMA 転送を行いません（DMA が停止中は DSS ビットが "1xx" になります）。

起動前に本ビットをセットしてから起動した場合、一時停止状態のままとなります。

本ビットがセットされている間に新たに発生した転送要求は受け付けられますが、本ビットをクリアしないと転送は開始しません（「18.3.3 DMA 転送全般」の「■ 転送要求の受け付けと転送」を参照）。

PAUS	機能
0	対応チャネル DMA 動作許可（初期値）
1	対応チャネル DMA 一時停止

- ・ リセット時："0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit29] STRG (Software TRiGger)：転送要求

対応するチャネルの DMA 転送要求を生成します。本ビットに "1" を書き込むと、レジスタへの書込みが終了した時点より転送要求が発生し、対応チャネルの転送を開始します。

ただし、対応するチャネルが起動されていない場合は、本ビットへの操作は無効となります。

（注意事項） DMAE ビットの書込みによる起動と本ビットによる転送要求が同時の場合、転送要求は有効となり、転送を開始します。また、PAUS ビットへの "1" 書込みと同時である場合、転送要求は有効となりますが、PAUS ビットを "0" に戻すまで DMA 転送は開始しません。

STRG	機能
0	無効
1	DMA 起動要求

- ・ リセット時："0" に初期化されます。
- ・ 読出し値は常に "0" となります。
- ・ 書込み値は "1" のみ有効で、"0" は動作に影響を与えません。

[bit28 ~ bit24] IS4 ~ IS0 (Input Select) * : 転送要因選択

転送要求の要因を以下のように選択します。ただし, STRG ビット機能によるソフトウェア転送要求は本設定にかかわらず有効となります。

IS	機能	転送停止要求
00000 _B	ソフトウェア転送要求のみ	なし
00001 _B	設定禁止	
01111 _B	設定禁止	
10000 _B	UART0 (受信完了)	あり
10001 _B	UART1 (受信完了)	
10010 _B	システム予約	
10011 _B	UART0 (送信完了)	なし
10100 _B	UART1 (送信完了)	
10101 _B	システム予約	
10110 _B	外部割込み 0	
10111 _B	外部割込み 1	
11000 _B	リロードタイマ 0	
11001 _B	リロードタイマ 1	
11010 _B	リロードタイマ 2	
11011 _B	積和マクロ	
11100 _B	PPG0	
11101 _B	PPG1	
11110 _B	PPG2	
11111 _B	PPG4	

- ・ リセット時: "00000_B" に初期化されます。
- ・ 読出しおよび書込みが可能です。

(注意事項) ・ 周辺機能の割込みによる DMA 起動を設定した場合 (IS=1xxxx_B), 選択した機能は, ICR レジスタで割込みを禁止状態にしてください。

- ・ 周辺機能の割込みによる DMA 起動を設定した状態で, ソフトウェア転送要求により DMA 転送を起動すると, 転送終了後, 該当する周辺に対して要因クリアを行います。このため, 本来の転送要求をクリアしてしまう可能性がありますので, 周辺機能の割込みによる DMA 起動を設定した状態では, ソフトウェア転送要求による起動を行わないでください。

[bit23 ~ bit20] - : 未定義ビット

読出し値は "0000_B" 固定です。書込みは無効となります。

[bit19 ~ bit16] BLK3 ~ BLK0 (BLoCK size) : ブロックサイズ指定

対応するチャンネルのブロック転送時のブロックサイズを指定します。本ビットに設定した値が 1 回の転送単位におけるワード数 (正確には、データ幅設定の繰返し回数) となります。ブロック転送を行わない場合は 01_H (サイズ1) を設定してください。

BLK3~BLK0	機能
XXXX _B	対応チャンネルのブロックサイズ指定

- リセット時: "0000_B" に初期化されます。
- 読出しおよび書込みが可能です。
- 全ビット "0" を指定した場合、ブロックサイズは 16 ワードとなります。
- 読出し時は、常にブロックサイズ (リロード値) が読み出されます。

[bit15 ~ bit0] DTC15 ~ DTC0 (Dma Terminal Count register) *: 転送回数レジスタ

転送回数を格納するレジスタです。各レジスタは 16 ビット長で構成されています。すべてのレジスタは、専用のリロードレジスタを持っています。転送回数レジスタのリロードを許可しているチャンネルに使用した場合、転送終了時に自動的に初期設定値をレジスタに戻します。

DTC15 ~ DTC0	機能
XXXX _H	対応チャンネルの転送回数指定

DMA 転送が起動すると、本レジスタのデータを DMA 専用転送回数カウンタのカウントバッファに格納して 1 転送単位ごとに - 1 カウントします (減算)。DMA の転送終了時にカウンタバッファの内容を本レジスタにライトバックして DMA は終了します。よって DMA 動作中の転送回数指定値を読み出すことはできません。

- リセット時: "00000000 00000000_B" に初期化されます。
- 読出しおよび書込みが可能です。DTC のアクセスは、必ずハーフワード長またはワード長にてアクセスしてください。
- 読出し時の値は、カウント値となります。リロード値の読出しはできません。

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 コントロール / ステータスレジスタ B [DMACB: DMACB0 ~ DMACB4]

DMAC 各チャネルの動作制御を行うレジスタで、チャネルごとに独立して存在します。
各ビット機能は以下に示すとおりです。

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
ch.0 :000204 _H	TYPE1,TYPE0		MOD1,MOD0		WS1,WS0		SADM	DADM	DTCR	SADR	DADR	ERIE	EDIE	DSS2 ~ DSS0		
ch.1 :00020C _H	R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ch.2 :000214 _H	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ch.3 :00021C _H	SASZ7 ~ SASZ0								DASZ7 ~ DASZ0							
ch.4 :000224 _H	R/W								R/W							

(初期値 : 00000000 00000000 00000000 00000000_B)

* マークは DMAC 転送中に設定すると動作に影響するビットです。このビットの書換えは DMAC 転送停止中 (起動禁止状態または一時停止状態) に行ってください。

[bit31, bit30] TYPE1, TYPE0 (TYPE) * : 転送タイプ設定

対応チャネルの動作タイプを以下のように設定します。

2 サイクル転送モード : 転送元アドレス (DMASA) と転送先アドレス (DMADA) を設定して読出し動作と書込み動作を転送回数分繰り返して転送するモードです。

TYPE1, TYPE0	機能
00 _B	2 サイクル転送 (初期値)
01 _B	設定禁止
10 _B	設定禁止
11 _B	設定禁止

- リセット時 : "00_B" に初期化されます。
- 読出しおよび書込みが可能です。
- 必ず "00_B" に設定してください。

[bit29, bit28] MOD1, MOD0 (MODE) * : 転送モード設定

対応チャネルの動作モードを以下のように設定します。

MOD1, MOD0	機能
00 _B	ブロック / ステップ転送モード (初期値)
01 _B	バースト転送モード
10 _B	設定禁止
11 _B	設定禁止

- リセット時 : "00_B" に初期化されます。
- 読出しおよび書込みが可能です。

[bit27, bit26] WS1, WS0 (Word Size) : 転送データ幅選択

対応チャネルの転送データ幅を選択します。本レジスタに設定したデータ幅単位で指定回数分の転送を行います。

WS1, WS0	機能
00 _B	バイト単位で転送 (初期値)
01 _B	ハーフワード単位で転送
10 _B	ワード幅単位で転送
11 _B	設定禁止

- リセット時: "00_B" に初期化されます。
- 読出しおよび書込みが可能です。

[bit25] SADM (Source-ADdr. count-Mode select) *: 転送元アドレスカウントモード指定
対応するチャネルの転送元アドレスの 1 転送ごとのアドレス処理を指定します。

アドレス増加 / 減少は, 設定された転送元アドレスカウント幅 (SASZ) に従って, 1 転送後に加算 / 減算され転送終了時に次回アクセス用アドレスが対応するアドレスレジスタ (DMASA) に書き込まれます。

したがって, DMA 転送が終了するまで転送元アドレスレジスタは更新されません。

アドレス固定にする場合は, 本ビットを "0" または "1" に指定してアドレスカウント幅 (SASZ, DASZ) を "0" としてください。

SADM	機能
0	転送元アドレスは増加します (初期値)。
1	転送元アドレスは減少します。

- リセット時: "0" に初期化されます。
- 読出しおよび書込みが可能です。

[bit24] DADM (Destination-ADdr. Count-Mode select) * : 転送先アドレスカウントモード指定

対応するチャンネルの転送先アドレスの 1 転送ごとのアドレス処理を指定します。

アドレス増加 / 減少は、設定された転送先アドレスカウント幅 (DASZ) に従って、1 転送後に加算 / 減算され転送終了時に次回アクセス用アドレスが対応するアドレスレジスタ (DMADA) に書き込まれます。

したがって、DMA 転送が終了するまで転送先アドレスレジスタは更新されません。

アドレス固定にする場合は、本ビットを "0" または "1" に指定してアドレスカウント幅 (SASZ, DASZ) を "0" としてください。

DADM	機能
0	転送先アドレスは増加します (初期値)。
1	転送先アドレスは減少します。

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit23] DTCCR (DTC-reg. Reload) * : 転送回数レジスタリロード指定

対応するチャンネルの転送回数レジスタのリロード機能を制御します。

本ビットによりリロード動作が許可されている場合、転送終了後に回数レジスタ値を初期設定値に戻して停止し、転送要求 (STRG, または IS 設定による起動要求) 待ち状態になります (本ビットが "1" の場合、DENB ビットはクリアされません)。

DENB=0, または DMAE=0 を設定すると強制停止します。

回数カウンタのリロード動作を禁止している場合は、アドレスレジスタにリロード指定をしても、転送終了にて停止するシングルショット動作となります。この場合 DENB ビットはクリアされます。

DTCCR	機能
0	転送回数レジスタリロードを禁止 (初期値)
1	転送回数レジスタリロードを許可

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit22] SADR (Source-ADdr.-reg. Reload) * : 転送元アドレスレジスタリロード指定

対応するチャンネルの転送元アドレスレジスタのリロード機能を制御します。

本ビットによりリロード動作が許可されている場合、転送終了後に転送元アドレスレジスタ値を初期設定値に戻します。

回数カウンタのリロード動作を禁止している場合は、アドレスレジスタにリロード指定をしても、転送終了にて停止するシングルショット動作となります。この場合、アドレスレジスタ値は初期設定値がリロードされた状態で停止します。

本ビットによりリロード動作が禁止されている場合は、転送終了時のアドレスレジスタ値は最終アドレスの次のアクセスアドレスとなります (アドレス増加を指定している場合は、増加したアドレスとなります)。

SADR	機能
0	転送元アドレスレジスタリロード禁止 (初期値)
1	転送元アドレスレジスタリロード許可

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit21] DADR (Dest.-ADdr.-reg. Reload) * : 転送先アドレスレジスタリロード指定

対応するチャンネルの転送先アドレスレジスタのリロード機能を制御します。

本ビットによりリロード動作が許可されている場合、転送終了後に転送先アドレスレジスタ値を初期設定値に戻します。

その他、機能の詳細は bit22:SADR の内容と同等になります。

DADR	機能
0	転送先アドレスレジスタリロード禁止 (初期値)
1	転送先アドレスレジスタリロード許可

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit20] ERIE (ERror Interrupt Enable)* : エラー割込み出力許可

エラー発生による終了時の割込み発生を制御します。発生したエラーの内容は DSS2 ~ DSS0 にて示されます。すべての終了要因で本割込みが発生するのではなく、特定の終了要因の際のみ割込みが発生することに注意してください (DSS2 ~ DSS0 ビット説明を参照)。

ERIE	機能
0	エラー割込み要求出力禁止 (初期値)
1	エラー割込み要求出力許可

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit19] EDIE (EnD Interrupt Enable) * : 終了割込み出力許可

正常終了時の割込み発生を制御します。

EDIE	機能
0	終了割込み要求出力禁止 (初期値)
1	終了割込み要求出力許可

- ・ リセット時: "0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit18 ~ bit16] DSS2 ~ DSS0 (Dma Stop Status) * : 転送停止要因表示

対応するチャンネルの DMA 転送停止 / 終了の要因を示す 3 ビットのコード (終了コード) を表示します。終了コードの内容は以下のとおりです。

DSS2	機能	割込み発生
0	初期値	なし
1	DMA 一時停止中 (DMAH, PAUS ビット, 割込みなど)	なし

DSS1, DSS0	機能	割込み発生
00 _B	初期値	なし
01 _B	-	なし
10 _B	転送停止要求	エラー
11 _B	正常終了	終了

転送停止要求は、周辺回路からの要求を使用した場合のみセットされます。

(注意事項) 「割込み発生」欄は、発生可能な割込み要求の種類を示します。

- ・ リセット時: "000_B" に初期化されます。
- ・ "000_B" を書き込むことにより、クリアされます。
- ・ 読出しおよび書込みが可能です。本ビットへの書込みは "000_B" のみ有効となります。

[bit15 ~ bit8] SASZ7 ~ SASZ0 (Source Addr count SiZe) * :

転送元アドレスカウントサイズ指定

対応するチャンネルの 1 転送ごとの転送元アドレス (DMASA) の増減幅を指定します。本ビットに設定した値が 1 回の転送単位におけるアドレス増減幅となります。アドレスの増減は転送元アドレスカウントモード (SADM) の指定に従います。

SASZ7 ~ SASZ0	機能
00 _H	アドレス固定
01 _H	バイト単位で転送
02 _H	ハーフワード単位で転送
04 _H	ワード単位で転送
上記以外	設定禁止

- リセット時 : "00000000_B" に初期化されます。
- 読出しおよび書込みが可能です。
- アドレス固定以外に設定する場合 , 転送データ幅 (WS) と同じ転送単位を設定してください。

[bit7 ~ bit0] DASZ7 ~ DASZ0 (Des Addr count SiZe) * :

転送先アドレスカウントサイズ指定

対応するチャンネルの 1 転送ごとの転送先アドレス (DMADA) の増減幅を指定します。本ビットに設定した値が 1 回の転送単位におけるアドレス増減幅となります。アドレスの増減は転送先アドレスカウントモード (DADM) の指定に従います。

DASZ7 ~ DASZ0	機能
00 _H	アドレス固定
01 _H	バイト単位で転送
02 _H	ハーフワード単位で転送
04 _H	ワード単位で転送
上記以外	設定禁止

- リセット時 : "00000000_B" に初期化されます。
- 読出しおよび書込みが可能です。
- アドレス固定以外に設定する場合 , 転送データ幅 (WS) と同じ転送単位を設定してください。

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 転送元 / 転送先アドレス設定レジスタ [(DMASA/DMADA):DMASA0 ~ DMASA4/DMADA0 ~ DMADA4]

DMAC 各チャネルの動作制御を行うレジスタで、チャネルごとに独立して存在します。
各ビット機能は以下に示すとおりです。

図 18.2-1 ch.0 ~ ch.3

アドレス 001000 _H ~ 00101C _H	bit31 bit30 bit29 bit28 bit27 bit26 bit25 bit24 bit23 bit22 bit21 bit20 bit19 bit18 bit17 bit16
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	R/W
	DMASA0 ~ DMASA3[19:16]
	bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	DMASA0 ~ DMASA3[15:0]
	R/W
	(初期値 : 00000000 00000000 00000000 00000000 _B)
	bit31 bit30 bit29 bit28 bit27 bit26 bit25 bit24 bit23 bit22 bit21 bit20 bit19 bit18 bit17 bit16
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	R/W
	DMADA0 ~ DMADA3[19:16]
	bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	DMADA0 ~ DMADA3[15:0]
	R/W
	(初期値 : 00000000 00000000 00000000 00000000 _B)

図 18.2-2 ch.4

アドレス 001020 _H 001024 _H	bit31 bit30 bit29 bit28 bit27 bit26 bit25 bit24 bit23 bit22 bit21 bit20 bit19 bit18 bit17 bit16
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	R/W
	DMASA4[23:16]
	bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	DMASA4[15:0]
	R/W
	(初期値 : 00000000 00000000 00000000 00000000 _B)
	bit31 bit30 bit29 bit28 bit27 bit26 bit25 bit24 bit23 bit22 bit21 bit20 bit19 bit18 bit17 bit16
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	R/W
	DMADA4[23:16]
	bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
	<div style="border: 1px solid black; width: 100%; height: 1.2em;"></div>
	DMADA4[15:0]
	R/W
	(初期値 : 00000000 00000000 00000000 00000000 _B)

転送元 / 先アドレスを格納するレジスタ群です。ch.0 ~ ch.3 は 20 ビット長、ch.4 は 24 ビット長で構成されています。

* マークは DMAC 転送中に設定すると動作に影響するビットです。このビットの書換えは DMAC 転送停止中（起動禁止状態または一時停止状態）に行ってください。

[bit19 ~ bit0] DMASA (DMA Source Addr) * :

ch.0 ~ ch.3 : DMASA19 ~ DMASA0 : 転送元アドレス設定

[bit23 ~ bit0] DMASA (DMA Source Addr) * :

ch.4 : DMASA23 ~ DMASA0 : 転送元アドレス設定

転送元アドレスの設定を行います。

[bit19 ~ bit0] DMADA (DMA Destination Addr) * :

ch.0 ~ ch.3 : DMADA19 ~ DMADA0 : 転送先アドレス設定

[bit23 ~ bit0] DMADA (DMA Destination Addr) * :

ch.4 : DMADA23 ~ DMADA0 : 転送先アドレス設定

転送先アドレスの設定を行います。

DMA 転送が起動すると、本レジスタのデータを DMA 専用アドレスカウンタのカウントバッファに格納して、1 転送ごとに設定に従いアドレスカウントします。DMA の転送終了時にカウントバッファの内容を本レジスタにライトバックして DMA は終了します。したがって DMA 動作中のアドレスカウンタ値を読み出すことはできません。

すべてのレジスタは、専用のリロードレジスタを持っています。転送元 / 転送先アドレスレジスタのリロードを許可しているチャンネルに使用した場合、転送終了時に自動的に初期設定値をレジスタに戻します。この際、ほかのアドレスレジスタには影響を与えません。

- リセット時："00000000 00000000 00000000 00000000_B" に初期化されます。
- 読出しおよび書込みが可能です。本レジスタは、必ず 32 ビットデータでアクセスしてください。
- 読出し時の値は、転送中は転送前のアドレス値、転送終了時には次のアクセスアドレス値となります。リロード値の読出しはできません。したがって、転送アドレスをリアルタイムで読出しすることはできません。
- 存在しない上位ビットには、"0" を設定してください。

< 注意事項 >

本レジスタにて DMAC 自身のレジスタを設定しないでください。DMAC 自身のレジスタに DMA 転送を行うことはできません。

■ DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 DMAC 全体制御レジスタ [DMACR]

DMAC の 5 チャンネル分全体の動作制御を行うレジスタです。本レジスタは必ずバイト長でアクセスしてください。

各ビット機能は以下に示すとおりです。

図 18.2-3 DMACR

アドレス	bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24	bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
000240 _H	DMAE	-	-	PM01	DMAH3 ~ DMAH0			-	-	-	-	-	-	-	-	-
	R/W			R/W	R/W	R/W	R/W	R/W								
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(初期値：0XX00000 XXXXXXXX XXXXXXXX XXXXXXXX_B)

* マークは DMAC 転送中に設定すると動作に影響するビットです。このビットの書換えは DMAC 転送停止中（起動禁止状態または一時停止状態）に行ってください。

[bit31] DMAE (DMA Enable) : DMA 動作許可

DMA 全チャンネルの動作制御を行います。

本ビットにより DMA 動作が禁止されている場合、チャンネルごとの起動 / 停止の設定や動作状態にかかわらず、全チャンネルの転送動作が禁止されます。転送中であったチャンネルは要求を取り下げ、ブロック境界にて転送を停止します。禁止状態にて、各チャンネルに対して行われる起動操作は、すべて無効となります。

本ビットにより DMA 動作が許可されている場合、チャンネルごとに起動 / 停止操作が有効となります。本ビットで DMA 動作許可を行ったのみでは、各チャンネルに対する起動は行われません。

本ビットに "0" を書き込むと強制停止しますが、必ず DMAH3 ~ DMAH0 ビット (DMACR:bit27 ~ bit24) で DMA を一時停止状態にした後強制停止 ("0" 書込み) してください。一時停止しないで強制停止した場合は DMA は停止しますが転送データは保証されません。停止の確認は DSS2 ~ DSS0 ビット (DMACB:bit18 ~ bit16) で行ってください。

DMAE	機能
0	全チャンネル DMA 動作禁止 (初期値)
1	全チャンネル DMA 動作許可

- ・ リセット時："0" に初期化されます。
- ・ 読出しおよび書込みが可能です。

[bit28] PM01 (Priority Mode ch.0, ch.1 robin) : チャネル優先度回転

ch.0, ch.1 の優先度を転送ごとに順位を回転させるときに設定します。

PM01	機能
0	優先順位固定 (ch.0 > ch.1) (初期値)
1	優先順位回転 (ch.1 > ch.0)

- リセット時 : "0" に初期化されます。
- 読出しおよび書込みが可能です。

[bit27 ~ bit24] DMAH3 ~ DMAH0 (DMA Halt) : DMA 一時停止

DMA 全チャネルの一時停止制御を行います。本ビットがセットされると、再び本ビットがクリアされるまでの間は、全チャネルの DMA 転送を行いません。

起動前に本ビットをセットした後起動した場合、全チャネルが一時停止のままとなります。

本ビットがセットされている間に DMA 転送が許可 (DENB=1) されているチャネルに発生した転送要求はすべて有効となり、本ビットをクリアすることにより転送を開始します。

DMAH3 ~ DMAH0	機能
0000 _B	全チャネル DMA 動作許可 (初期値)
0000 _B 以外	全チャネル DMA 一時停止

- リセット時 : "0" に初期化されます。
- 読出しおよび書込みが可能です。

[bit30, bit29, bit23 ~ bit0] - : 未定義ビット

読出し値は不定です。

18.3 DMAC (DMA コントローラ) の動作

DMA コントローラの動作について説明します。

■ 主要動作

- 各転送チャネルは、独立に各種機能を設定します。
- 各チャネルは起動許可後、設定した転送要求を検出するまでは転送動作を行いません。
- 転送要求検出により、バスコントローラに対し DMA 転送要求を出力し、バスコントローラの制御によりバス権を取得して転送を開始します。
- 転送はチャネルごとに独立に設定されたモード設定に従ったシーケンスで行われます。

■ 転送モード

DMA の各チャネルは、それぞれの DMACB レジスタの MOD1, MOD0 ビットで設定された転送モードに従って転送動作を行います。

● ブロック / ステップ転送

1 回の転送要求にて 1 ブロック転送単位のみ転送を行い、その後、次の転送要求が受け付けられるまでは DMA はバスコントローラに対し転送要求を停止します。

1 ブロック転送単位：設定されたブロックサイズ分 DMACA:BLK3 ~ BLK0

● バースト転送

1 回の転送要求にて指定転送回数終了まで連続して転送を行います。

指定転送回数：ブロックサイズ分 × 転送回数分

(DMACA:BLK3 ~ BLK0 × DMACA:DTC15 ~ DTC0)

■ 転送タイプ

● 2 サイクル転送 (通常転送)

DMA コントローラの動作は、読出し動作と書込み動作を 1 つの単位として動作を行います。

転送元レジスタのアドレスからデータを読み出して転送先レジスタのアドレスへデータを書き込みします。

■ 転送アドレス

アドレッシングは、チャンネル転送元 / 転送先ごとに独立に設定します。

● 2 サイクル転送でのアドレスの指定

あらかじめアドレスを設定してあるレジスタ (DMASA, DMADA) から読み出した値をアドレスとしてアクセスします。

転送要求を受け付けた後、DMA はレジスタからアドレスを一時記憶バッファに格納して転送を開始します。

1 回の転送 (アクセス) ごとにアドレスカウンタにて次回アクセスアドレスを生成 (加算 / 減算 / 固定選択可能) して、一時記憶バッファの内容は 1 ブロック転送単位終了ごとにレジスタ (DMASA, DMADA) ヘライトバックされます。

したがって、アドレスレジスタ (DMASA, DMADA) 値は、1 ブロック転送単位ごとにしか更新されませんので転送中のアドレスをリアルタイムに知ることはできません。

■ 転送回数と転送終了

● 転送回数

1 ブロック転送単位終了ごとに、転送回数レジスタをデクリメント (- 1) します。転送回数レジスタが "0" になると指定転送回数終了となり、終了コードを表示して停止または再起動します。

転送回数レジスタ値はアドレスレジスタと同様に 1 ブロック転送単位ごとにしか更新されません。

転送回数レジスタリロード禁止に設定している場合は転送を終了します。許可されている場合はレジスタ値を初期化して転送待ち状態になります (DMACB:DTCR)。

● 転送終了

転送終了要因には、以下のものがあり、終了時は終了コードとして要因が表示されます。(DMACB:DSS2 ~ DSS0)

- 指定転送回数の終了 (DMACA:BLK3 ~ BLK0 × DMACA:DTC15 ~ DTC0) 正常終了
- 周辺回路からの転送停止要求の発生 エラー
- リセットの発生 リセット

各終了要因に対応して、転送停止要因表示 (DSS) され転送終了割込み / エラー割込みを発生可能です。

18.3.1 転送要求の設定

DMA 転送を起動する転送要求には内蔵周辺要求とソフトウェア要求の 2 種類があります。ソフトウェア要求については、ほかの要求の設定にかかわらず常に使用することができます。

■ 内蔵周辺要求

内蔵周辺回路の割込み発生により、転送要求を発生します。

チャンネルごとに、どの周辺の割込みにより転送要求を発生するかを設定します (DMACA:IS4 ~ IS0=1xxxxB)。

(注意事項) 転送要求に使用した割込み要求は、CPU への割込み要求としても見えますので割込みコントローラの設定を割込み禁止に設定してください (ICR レジスタ)。

■ ソフトウェア要求

レジスタのトリガビットへの書込みにより、転送要求を発生します (DMACA:STRG)。

上記の転送要求とは独立で、常に使用することができます。

起動 (転送許可) と同時にソフトウェア要求を行った場合、直後にバスコントローラに対し DMA 転送要求を出力し転送を開始します。

(注意事項) 内蔵周辺要求を設定したチャンネルに対してソフトウェア要求を行うと、転送終了後、該当する周辺に対して要因クリアを行います。このため、本来の転送要求をクリアしてしまう可能性がありますので、ソフトウェア要求を行わないでください。

18.3.2 転送シーケンス

チャンネルごとに、DMA 転送起動後の動作シーケンスなどを決定する転送タイプと転送モードを独立して設定することができます (DMACB:TYPE1, TYPE0, MOD1, MOD0 の設定)。

■ 転送シーケンスの選択

レジスタの設定により以下のシーケンスが選択可能です。

- バースト 2 サイクル転送
- ブロック / ステップ 2 サイクル転送

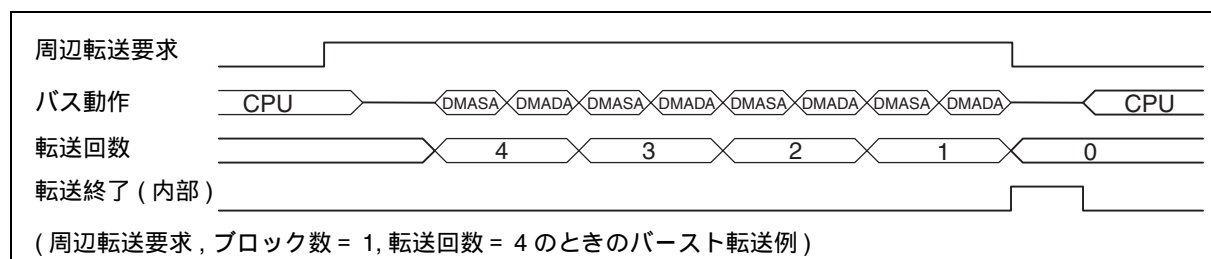
■ バースト 2 サイクル転送

1 回の転送要因にて指定転送回数の転送を続けて行います。2 サイクル転送の場合の転送元 / 転送先アドレスは ch.0 ~ ch.3 では 20 ビット、ch.4 では 24 ビット指定可能です。

転送要因は、周辺転送要求 / ソフトウェア転送要求を選択できます。

● バースト転送の特長

- 転送要求を 1 回受け付けると、転送回数レジスタが "0" になるまで連続して転送を行います。
- 転送回数はブロックサイズ分 × 転送回数分になります (DMACA:BLK3 ~ BLK0 × DMACA:DTC15 ~ DTC0)。
- 転送中に再度要求が発生した場合、要求は無視されます。
- 転送回数レジスタのリロード機能が有効である場合、転送終了後より次の転送要求を受け付けます。
- 転送中により高い優先順位の他チャンネルの転送要求を受け付けた場合、ブロック転送単位の境目でチャンネルを切り換え、そのチャンネルの転送要求がクリアされるまで復帰しません。



■ ステップ/ブロック転送 2 サイクル転送

ステップ/ブロック転送 (1 転送要求ごとに指定ブロック回数分のみ転送を行う) の場合の転送元/転送先アドレスは、ch.0 ~ ch.3 では 20 ビット、ch.4 では 24 ビット指定可能です。

● ステップ転送

ブロックサイズに "1" を設定すると、ステップ転送シーケンスとなります。

[ステップ転送の特長]

- 転送要求を 1 回受け付けると、1 回の転送を行った後、転送要求をクリアして転送を停止します (バスコントローラに対し DMA 転送要求を取り下げる)。
- 転送中に再度要求が発生した場合、要求は無視されます。
- 転送中により高い優先順位のお他チャネルの転送要求を受け付けた場合、転送停止後に、チャネルを切り換えて続けて転送を開始します。ステップ転送における優先順位は、転送要求が同時に発生した場合のみ意味を持ちます。

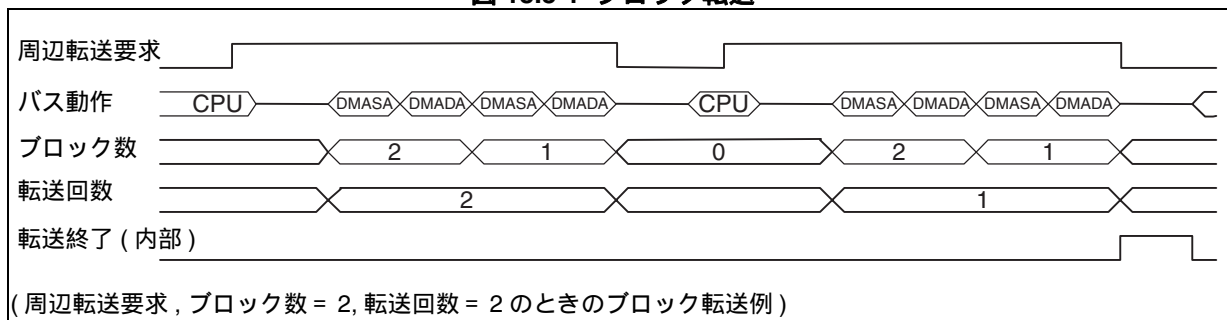
● ブロック転送

ブロックサイズに "1" 以外を設定すると、ブロック転送シーケンスとなります。

[ブロック転送の特長]

- 1 転送単位が複数回 (ブロック数) の転送サイクルから構成されること以外は、ステップ転送と全く同じ動作となります。

図 18.3-1 ブロック転送



18.3.3 DMA 転送全般

DMA 転送のブロックサイズとリロード動作について説明します。

■ ブロックサイズ

- 転送データの 1 転送単位はブロックサイズ指定レジスタに設定した数 (×データ幅) のデータの集合となります。
- 1 転送サイクルにて転送されるデータは、データ幅指定の値に固定されるため、1 転送単位はブロックサイズ指定値分の転送サイクル数より構成されることとなります。
- 転送中に、より高位の優先順位の転送要求が受け付けられた場合、または転送の一時停止要求が発生した場合において、ブロック転送時においても 1 転送単位の境界にならないと停止しません。これにより、分割・一時停止を希望しないデータブロックのデータ保護が可能ですが、ブロックサイズが大きい場合はレスポンスを低下させる原因ともなります。
- リセット発生の場合のみ即時に停止しますが、転送中であったデータの内容などは保証されません。

■ リロード動作

本モジュールでは、チャンネルごとに以下の 3 種類のリロード機能の設定が可能です。

● 転送回数レジスタリロード機能

指定回数の転送が終了した後、転送回数レジスタに初期設定値を再設定して起動受け待ちします。

全転送シーケンスを繰り返し行う際に設定します。

リロード指定をしない場合、指定回数の転送終了後は回数レジスタ値が "0" のままとなり、以降の転送は行われません。

● 転送元アドレスレジスタリロード機能

指定回数の転送が終了した後、転送元アドレスレジスタに初期設定値を再設定します。

転送元アドレス領域内で固定領域から繰り返し転送する場合に設定します。

リロード指定をしない場合、指定回数の転送終了後は転送元アドレスレジスタ値が終了時の次のアドレスとなります。アドレス領域を固定しない場合に使用します。

● 転送先アドレスレジスタリロード機能

指定回数の転送が終了した後、転送先アドレスレジスタに初期設定値を再設定します。

転送先アドレス領域内で固定領域へ繰り返し転送する場合に設定します。

リロード指定をしない場合、指定回数の転送終了後は転送元アドレスレジスタ値が終了時の次のアドレスとなります。アドレス領域を固定しない場合に使用します。

- 転送元 / 転送先レジスタのリロード機能を有効にしたのみでは、指定回数転送終了後の再起動は行われず、各アドレスレジスタ値が再設定されるのみとなります。

< 注意事項 >

動作モードとリロード動作の特殊な例

- 転送終了にていったん停止し、再度入力検出から行いたい場合は、リロード指定を行わないようにしてください。
 - バースト / ブロック / ステップ転送モードで転送している場合では、転送終了となるとリロード後いったん転送を中断し、改めて転送要求入力検出されるまで転送を行いません。
-

■ アドレッシングモード

各転送チャンネルの転送先および転送元アドレスは、それぞれ独立に指定します。

指定方法には下記の方法があります。転送シーケンスによって設定してください。

● アドレスレジスタ指定

2 サイクル転送モードでは、転送元アドレス設定レジスタ (DMASA) には転送元アドレスを、転送先アドレス設定レジスタ (DMADA) には転送先アドレスを設定してください。

● アドレスレジスタの特長

ch.0 ~ ch.3 は 20 ビット、ch.4 は 24 ビットのレジスタです。

● アドレスレジスタの機能

- 毎アクセス時に読み出され、アドレスバスへ放出されます。
 - 同時にアドレスカウンタにて次回アクセス時のアドレス計算が行われ、計算結果のアドレスにてアドレスレジスタを更新します。
 - アドレス計算は、各チャンネル / 転送先 / 転送元それぞれ独立に、加算 / 減算より選択します。アドレスの増減幅はアドレスカウントサイズ指定レジスタ値によります (DMACB:SASZ, DASZ)。
 - 転送終了時のアドレスレジスタには、リロード機能を有効にしていない場合、最終アドレスにアドレス計算をした結果のアドレスが残されます。
 - リロード機能を有効にしている場合、アドレスの初期値がリロードされます。
-

< 注意事項 >

- 20 ビットまたは 24 ビット長フルアドレス計算の結果、オーバフロー / アンダフローが発生した場合でも、そのチャンネルの転送は継続されます。オーバフロー / アンダフローが発生しないように各チャンネルを設定してください。
 - アドレスレジスタに DMAC 自身のレジスタのアドレスを設定しないでください。
-

■ データの種類など

1 回の転送で転送されるデータ長 (データ幅) は、バイト、ハーフワード、ワードから選択します。

● バイト、ハーフワード、ワード

DMA 転送においても、ワードバウンダリ仕様が守られるため、転送先 / 転送元アドレス指定においてデータ長と食い違うアドレスが設定された場合、異なる下位ビットは無視されます。

- ワード ...実際のアクセスアドレスは下位 2 ビットが "00_B" から始まる 4 バイトとなります。
- ハーフワード ...実際のアクセスアドレスは下位 1 ビットが "0" から始まる 2 バイトとなります。
- バイト ...実際のアクセスアドレスとアドレス指定が一致します。

転送元アドレスと転送先アドレスの下位ビットが食い違っている場合、内部アドレスバス上には設定そのままのアドレスが出力されますが、バス上の各転送対象において、上記に従ってアドレスが修正されてアクセスが行われます。

■ 転送回数制御

転送回数は、最大で 16 ビット長の範囲内 (1 回 ~ 65536 回) で指定します。転送回数指定値は転送回数レジスタ (DMACA:DTC) に設定します。

レジスタ値は転送開始時に一時記憶バッファへ格納され、転送回数カウンタにより減算されます。このカウンタ値が "0" となったとき、指定回数の転送終了として検出され、そのチャンネルの転送停止または再起動受け待ち (リロード指定時) が行われます。

● 転送回数レジスタ群の特長

- 各レジスタ 16 ビットです。
- すべてのレジスタはそれぞれ専用リロードレジスタを持ちます。
- レジスタ値が "0" のときに起動すると、65536 回の転送を行います。

● リロード動作

- リロード機能をもつレジスタで、リロード機能が許可されている場合にのみ有効です。
- 転送起動時に、回数レジスタの初期値をリロードレジスタに退避します。
- 転送回数カウンタにてカウントを行った際 "0" になると、転送終了を通知すると共に、リロードレジスタより初期値を読み出し、回数レジスタに書き込みます。

■ CPU 制御

DMA 転送要求が受け付けられると、DMA はバスコントローラに対し転送要求を発行します。

バスコントローラはバス動作の切れ目にて内部バス使用权を DMA に明け渡し、DMA 転送が開始されます。

● DMA 転送と割り込み

- DMA 転送中は、基本的には転送終了まで割り込みの受け付けは停止します。
- また割り込み処理動作中に DMA 転送要求があった場合、転送要求は受け付けられ、転送終了まで割り込み処理動作は停止します。
- 例外として、NMI 要求または割り込みコントローラにて設定されたホールド抑止レベルより高いレベルの割り込み要求が発生した場合には、DMAC は転送単位 (1 ブロック) の境界にてバスコントローラへの転送要求を一時取り下げ、割り込み要求がクリアされるまでは転送を一時停止状態にします。この間、転送要求は内部で保持されます。割り込み要求がクリアされた後に再度 DMAC はバスコントローラへ転送要求を発行してバス使用权を取得し、DMA 転送を再開します。

● DMA 抑止

- DMA 転送中に、より優先度の高い割り込み要因が発生すると DMA 転送を中断して該当割り込みルーチンへ分岐します。この機構は割り込み要求がある限り有効ですが、割り込み要因をクリアすると抑止機構が働かなくなり、割り込み処理ルーチン内で DMA 転送を再開します。このため、DMA 転送を中断するレベルの割り込み要因の処理ルーチン内で、割り込み要因クリア後の DMA 再転送開始を抑止したいときは、DMA 抑止機能を使用します。DMA 抑止機能は、DMA 全体制御レジスタの DMAH3 ~ DMAH0 ビットに "0" 以外の値を書くことで起動し、"0" を書くことで停止します。
- 本機能は主として割り込み処理ルーチン内で使用します。割り込み処理ルーチンで割り込み要因をクリアする前に DMA 抑止レジスタの内容を 1 つ増加させます。このようにすると、以降、DMA 転送は行いません。割り込み処理への対応を行った後、復帰する前に DMAH3 ~ DMAH0 ビットの内容を 1 つ減少させます。もし、多重割り込みであれば DMAH3 ~ DMAH0 ビットの内容は、まだ "0" にならないため、引き続いて DMA 転送は抑止されます。また、多重割り込みでなければ、DMAH3 ~ DMAH0 ビットの内容は "0" になるため、その後すぐに DMA 要求を有効にします。

< 注意事項 >

- レジスタのビット数は 4 ビットですので、15 レベルを超える多重割り込みで本機能を使用することができません。
 - DMA タスクの優先順位は必ずほかの割り込みレベルより 15 レベルは上に置いてください。
-

■ 動作開始

DMA 転送の開始は、チャンネルごとに独立に制御しますが、その前に全チャンネルの動作を許可しておく必要があります。

● 全チャンネル動作許可

DMAC 各チャンネルに起動を行う前に、DMA 動作許可ビット (DMACR:DMAE) にてあらかじめ全チャンネルの動作を許可する必要があります。許可されていない状態にて行った起動設定、および発生した転送要求はすべて無効となります。

● 転送起動

各チャンネルの制御レジスタにある動作許可ビットにて、転送動作を起動します。起動されたチャンネルに対する転送要求が受け付けられると、設定されたモードにて DMA 転送動作が開始されます。

● 一時停止状態よりの起動

各チャンネルまたは全チャンネル制御にて、起動前に一時停止状態にしてあった場合、転送動作を起動しても一時停止状態を維持しています。この間に転送要求が発生した場合は、要求を受け付け、保持します。一時停止を解除した時点から転送を開始します。

■ 転送要求の受け付けと転送

DMA 転送の転送要求の受け付けと転送について説明します。

● 転送要求の受け付けと転送

- 起動後、各チャンネルに対して設定した転送要求のサンプリングが開始されます。
- 周辺割込み起動を選択した場合は、転送要求がクリアされるまで DMAC は転送を続けますがクリアされたら 1 転送単位で転送を停止します (周辺割込み起動)。周辺割込みはレベル検出扱いとなりますので、割込みは DMA による割込みクリアで行うようにしてください。
- 転送要求は、他チャンネルの要求が受け付けられて転送が行われている間でも、常に受け付けられており、優先順位の判断を行って、1 転送単位ごとに転送するチャンネルを決定しています。

■ DMA による周辺割込みクリア

本 DMA には周辺割込みをクリアする機能があります。この機能は DMA 起動要因に周辺割込みを選択したときに機能します (IS4 ~ IS0=1xxxx_B のとき)。

周辺割込みのクリアは設定された起動要因にのみ行われます。つまり IS4 ~ IS0 で設定された周辺機能のみクリアされます。

● 割込みクリアの発生タイミング

転送モードにより発生するタイミングが違います (動作フローを参照)。

• ブロック / ステップ転送

ブロック転送を選択した場合、1 ブロック (ステップ) 転送ごとにクリア信号を発生します。

• バースト転送

バースト転送を選択した場合は、指定転送回数がすべて終了した後、クリア信号を発生します。

■ 一時停止

DMA 転送は、以下の場合に一時停止します。

- 制御レジスタへの書込みによる一時停止の設定 (各チャネル独立または全チャネル同時に設定)

一時停止ビットにより一時停止を設定すると、一時停止解除設定を再度設定するまでの間、対応するチャネルの転送は停止します。一時停止の確認は DSS ビットで行ってください。

一時停止を解除すると、転送を再開します。

- NMI/ ホールド抑止レベル割込み処理中

NMI 要求、またはホールド抑止レベルより高いレベルの割込み要求が発生した場合、転送中のチャネルはすべて転送単位の境界にて一時停止し、バス権を開放して NMI/ 割込み処理を優先させます。また、NMI/ 割込み処理中に受け付けられた転送要求は、そのまま保持され、NMI 処理の終了を待ちます。

要求が保持されたチャネルは、NMI/ 割込み処理が終了した後に転送を再開します。

■ 動作終了 / 停止

DMA 転送の終了は、チャネルごとに独立に制御しますが、全チャネルの動作を禁止することも可能です。

- 転送終了

リロード動作が有効でない場合、転送回数レジスタが "0" になると転送を停止し、終了コードにて『正常終了』を表示した後、以降の転送要求は無効となります (DMACA: DENB ビットをクリアする)。

リロード動作が有効である場合、転送回数レジスタが "0" になると初期値をリロードし、終了コードにて『正常終了』を表示した後、再度転送要求待ちとなります (DMACA: DENB ビットをクリアしない)。

- 全チャネル動作禁止

DMA 動作許可ビット DMAE にて全チャネルの動作を禁止すると、動作中であったチャネルも含め DMAC のすべての動作が停止します。以降、再度全チャネル DMA 動作を許可しても、チャネルごとに起動し直さないと、転送は行われません。この場合、割込みは一切発生しません。

■ エラーによる停止

指定回数の転送終了による正常終了以外の要因で停止する場合として、各種エラー発生による停止および強制停止があります。

● 周辺回路よりの転送停止要求の発生

転送要求を出力する周辺回路によっては、異常を検出した際などに転送停止要求を発生するものがあります（例：通信系周辺における受信 / 送信エラーなど）。

この転送停止要求を受け取った DMAC は、終了コードにて『転送停止要求』を表示して対応するチャンネルの転送を停止します。

< 注意事項 >

- 周辺回路の転送停止要求の有無については、DMACA レジスタの bit28 ~ bit24 (IS4 ~ IS0) 転送要因選択ビットの説明を参照してください。
 - 各転送停止要求の発生条件についての詳細は、各周辺回路の説明を参照してください。
-

■ DMAC 割込み制御

転送要求となる周辺割込みとは独立に、DMAC チャンネルごとに以下の割込みを出力することが可能です。

- 転送終了割込み...正常終了した場合のみ発生する
- エラー割込み.....周辺回路からの転送停止要求（周辺に起因するエラー）

これら割込みはすべて終了コードの内容に従って出力されます。

割込み要求のクリアは、DMACS の DSS2 ~ DSS0（終了コード）に "000_B" を書き込むことにより行います。なお、終了コードは再起動する際には必ず "000_B" を書き込んでクリアしてください。

リロード動作が有効である場合は自動的に再起動しますが、この際には終了コードはクリアされず、次の転送終了による新しい終了コードの書き込み時まで保持されます。

終了コードにて表示できる終了要因は 1 種類のみですので、複数の要因が同時に発生した場合は優先判断を行った結果を表示します。この際に発生する割込みは、表示されている終了コードに従います。

終了コードの表示の優先順位を以下に示します（上から、優先度の高い順）。

- リセット
- "000_B" 書き込みによるクリア
- 周辺停止要求
- 正常終了
- チャンネル選択と制御

■ スリープ中の DMA 転送

DMAC は、スリープモード中でも動作させることができます。

スリープモードでの動作を期待する場合、次のことに注意する必要があります。

- CPU は停止状態なので DMAC のレジスタを書き換えることはできません。スリープモードに入る前に設定は済ませておいてください。
- スリープモードは、割り込みで解除されますので、DMAC 起動要因で周辺での割り込みを選択した場合は、割り込みコントローラで割り込みを禁止する必要があります。

同様に、DMAC 終了割り込みでスリープモードを解除したくない場合は割り込みを禁止にしてください。

■ チャンネル選択と制御

転送チャンネルの数は 5 チャンネルまで同時設定が可能です。各チャンネルは基本的に各機能を独立に設定することが可能です。

● チャンネル間優先順位

DMA 転送は同時に 1 チャンネルのみ可能なため、各チャンネル間には優先順位が設定されます。

順位設定には、固定 / 回転の 2 モードがあり、チャンネルグループ (後述) ごとに選択します。

(1) 固定モード

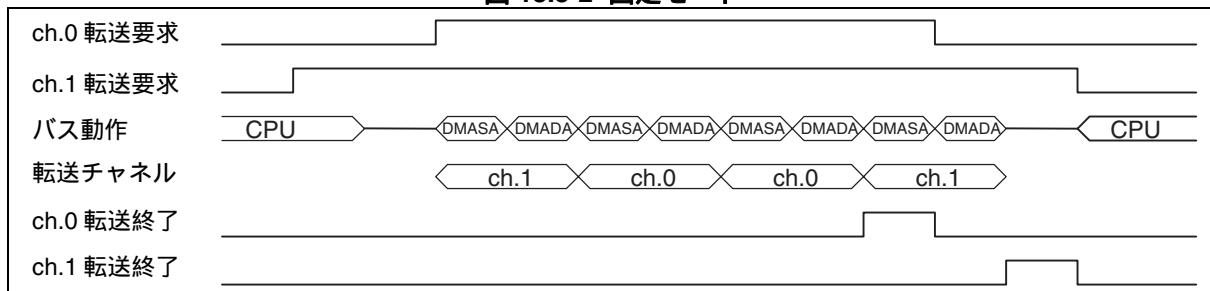
チャンネル番号の小さい順に固定されます。

(ch.0 > ch.1 > ch.2 > ch.3 > ch.4)

転送中により高い優先度の転送要求が受け付けられた場合、1 転送単位 (ブロックサイズ指定レジスタに設定した数 × データ幅) の転送が終了した時点で、転送チャンネルが高優先度側に切り換わります。

高優先度側の転送が終了すると、元のチャンネルの転送を再開します。

図 18.3-2 固定モード

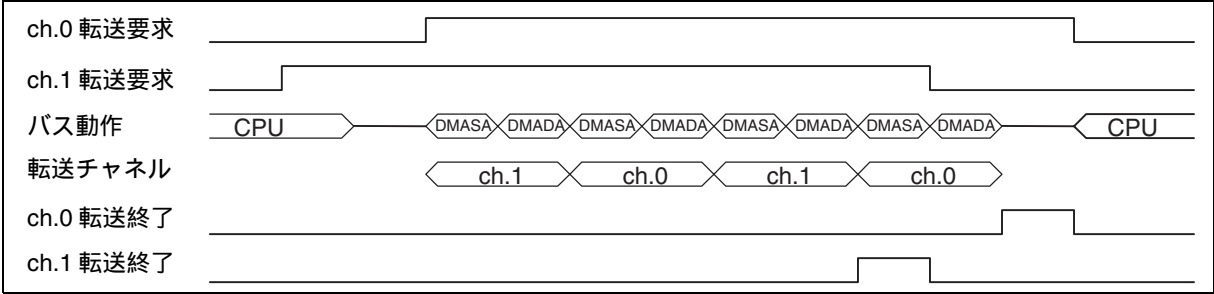


(2) 回転モード (ch.0 - ch.1 間のみ)

動作許可後の初期状態は (1) と同じ順位に設定されますが , 1 転送終了ごとにそのチャネルの優先度は逆転します。したがって , 同時に転送要求が出力されている場合 , 1 転送単位ごとにチャネルが切り換わります。

連続 / バースト転送を設定した場合に効果のあるモードです。

図 18.3-3 回転モード



● チャンネルグループ

優先順位の選択は , 以下の単位で設定します。

モード	優先度	備考
固定	ch.0>ch.1	
回転	ch.0>ch.1	初期状態は上側の順位です。 上側が転送されると反転します。
	ch.0<ch.1	

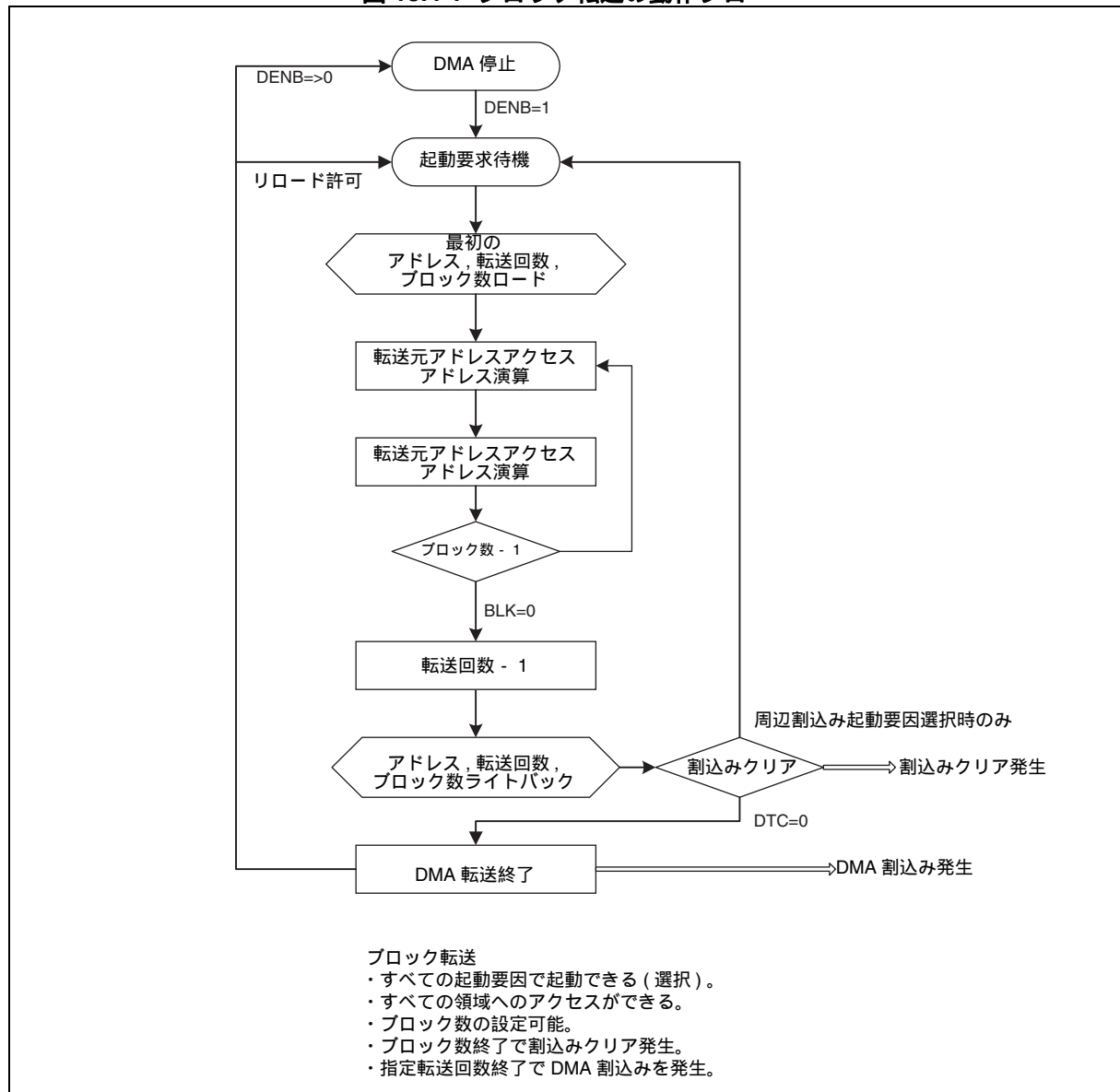
18.4 動作フローチャート

次の各転送モードにおける動作フローチャートを示します。

- ・ブロック転送
- ・バースト転送

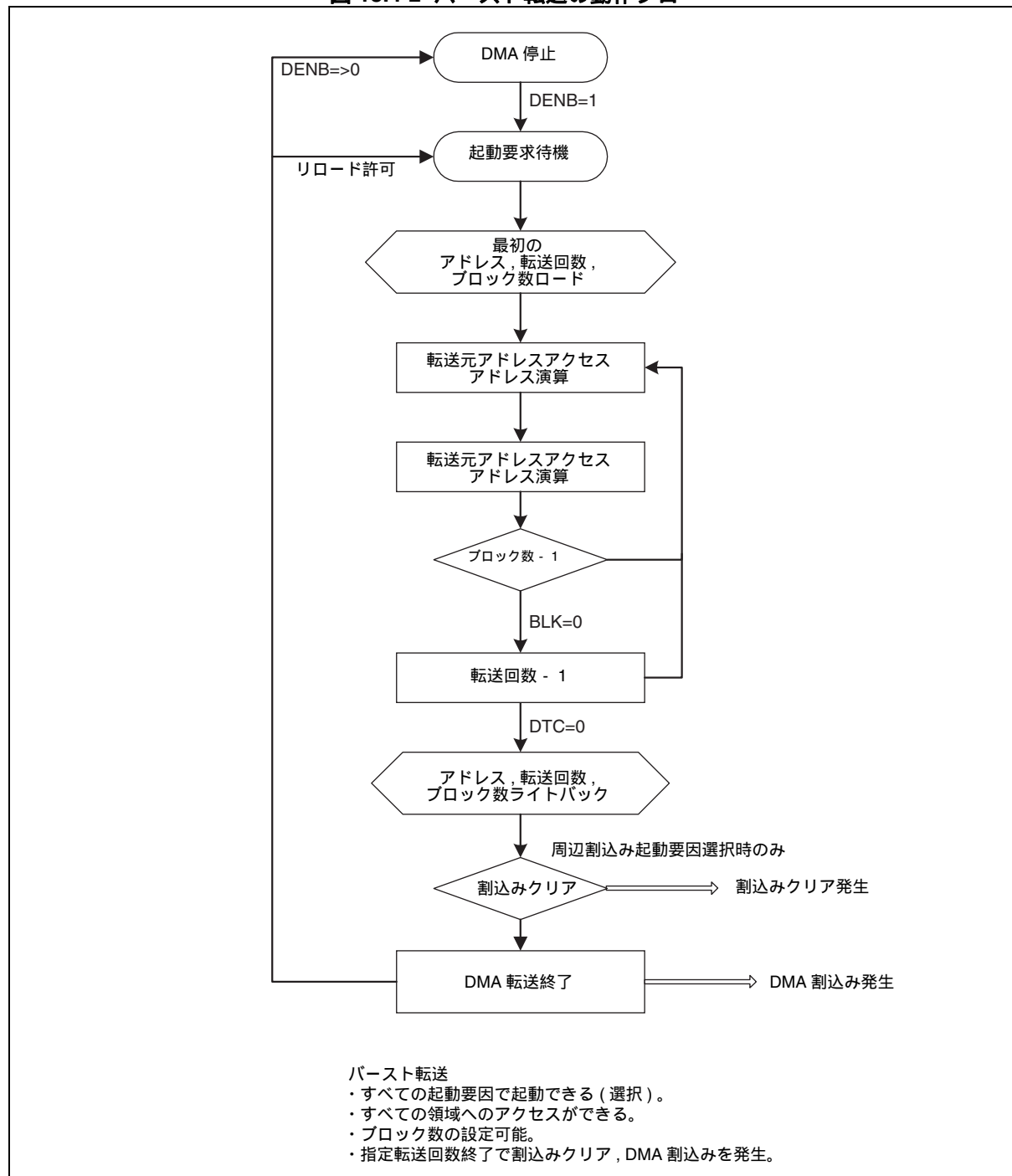
■ ブロック転送

図 18.4-1 ブロック転送の動作フロー



■ バースト転送

図 18.4-2 バースト転送の動作フロー



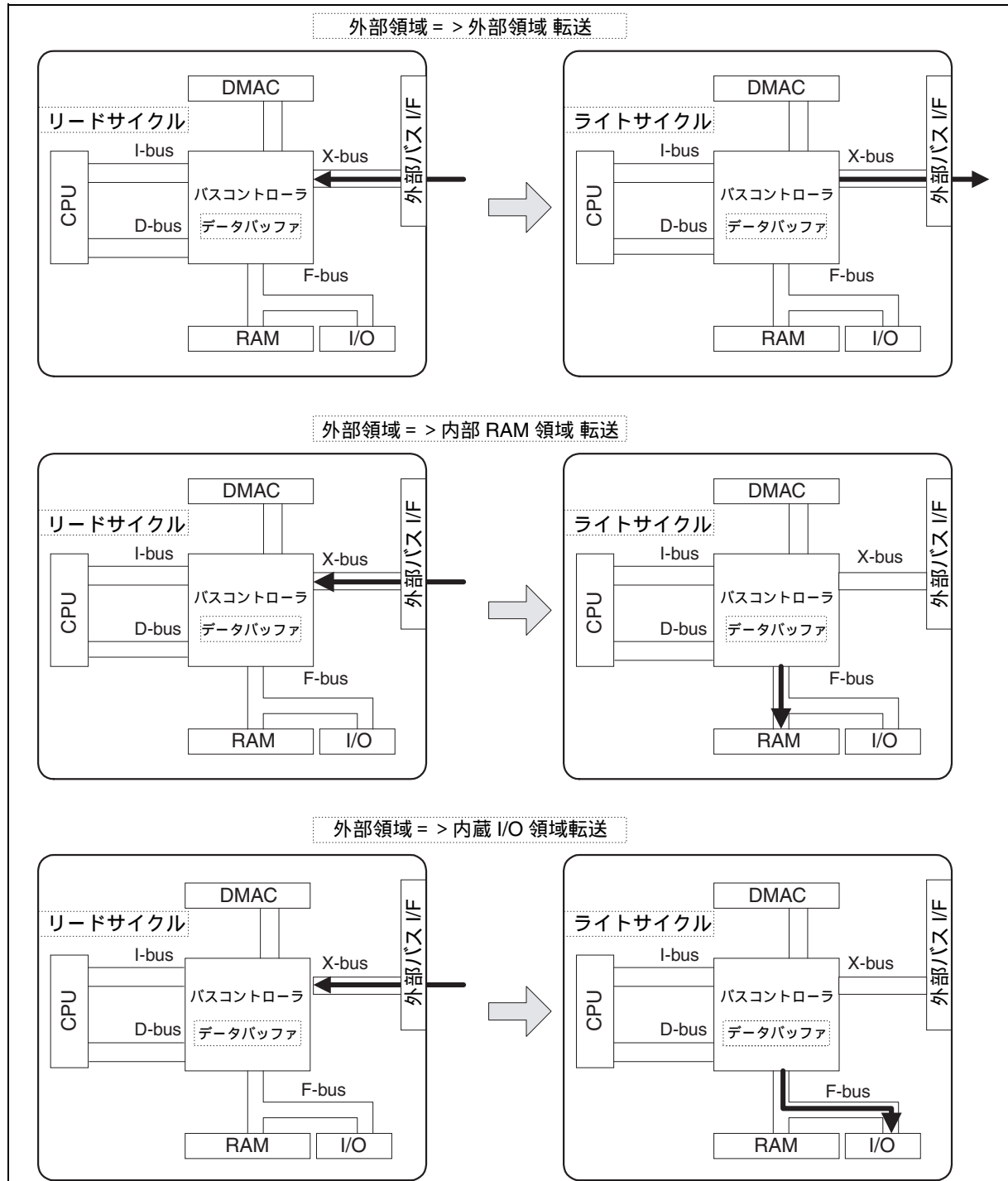
18.5 データバス

2 サイクル転送時のデータの動きを示します。

■ 2 サイクル転送時のデータの動き

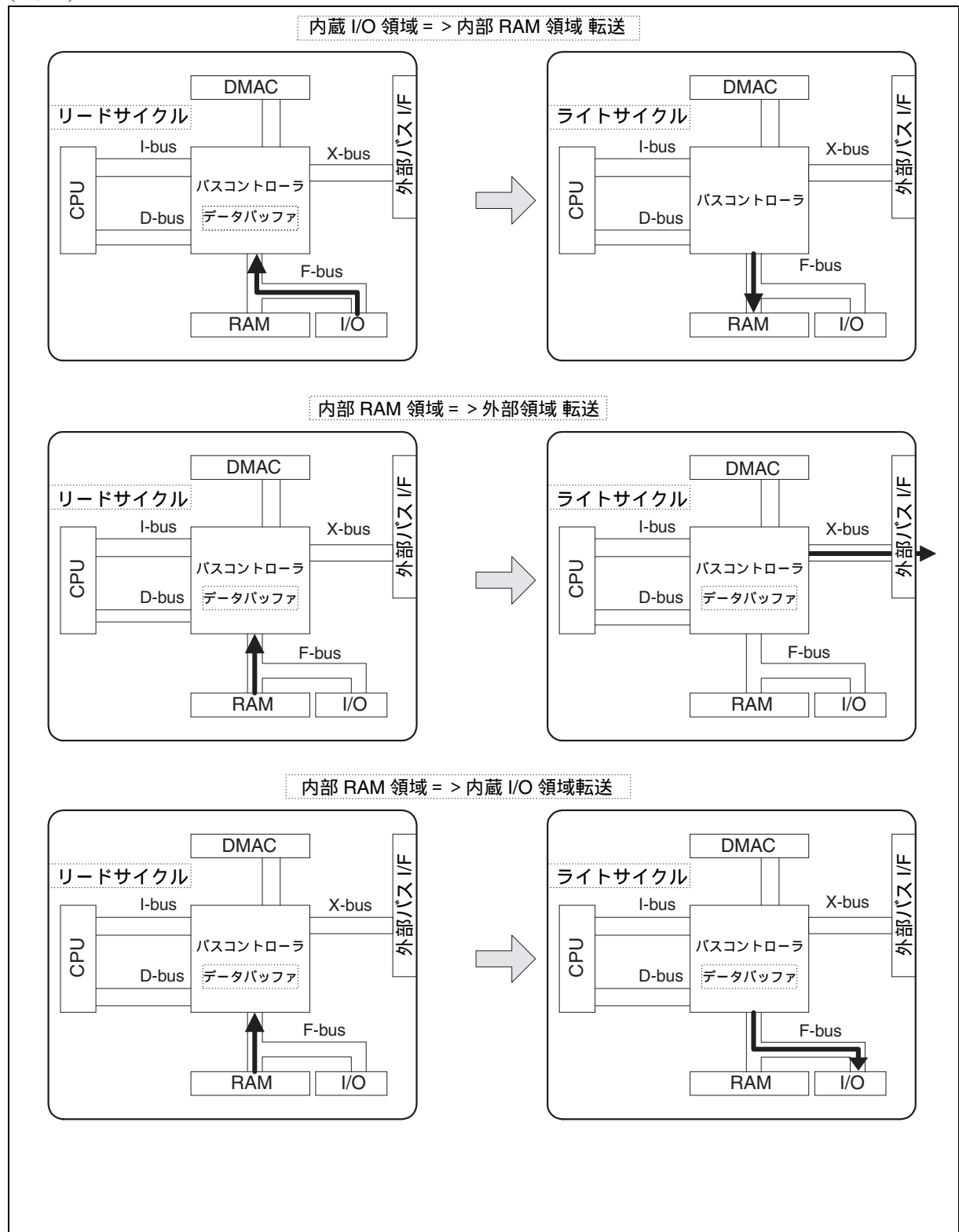
転送例の 6 種類を図示します (ほかの組合せは省略)。

図 18.5-1 2 サイクル転送例



(続く)

(続き)



第19章

フラッシュメモリ

フラッシュメモリの概要，レジスタの構成 / 機能および動作について説明します。

- 19.1 概要
- 19.2 レジスタ
- 19.3 フラッシュメモリのアクセスモード
- 19.4 自動アルゴリズム起動方法
- 19.5 自動アルゴリズム実行状態
- 19.6 デュアルオペレーション動作

19.1 概要

MB91F267A/MB91F267NA には、128KB (1 M ビット) の容量で、+ 3.3 V 単一電源による全セクタ括消去およびセクタ単位での消去が可能で、FR-CPU によるワードおよびバイト (16 ビット / 8 ビット) 単位での書込みが可能なフラッシュメモリを内蔵しています。

■ フラッシュメモリの概要

このフラッシュメモリは、内蔵の 3.3V 動作 128KB フラッシュメモリです。当社製 4 M ビット (512KB × 8/256KB × 16) フラッシュメモリ MBM29LV400TC と同じもので (容量、一部セクタ構成を除く) あり、ROM ライタによるデバイス外部からの書込みも可能です。

また、MBM29LV400TC 相当の機能に加え、FR-CPU の内蔵 ROM として使用する場合には、ワード (32 ビット) 単位での命令 / データ読出しが可能であり、デバイスの高速動作を実現できます。

本書と併せて MBM29LV400TC データシートも参照してください。

フラッシュメモリマクロと FR-CPU インタフェース回路の組合せにより、以下の機能を実現します。

- CPU のプログラム / データ格納用メモリとして機能
 - ROM として使用する際は 32 ビットバス幅でアクセス可能
 - CPU による読出し / 書込み / 消去 (自動プログラムアルゴリズム *) が可能
- 単体フラッシュメモリ製品の MBM29LV400TC 相当の機能
 - ROM ライタによる読出し / 書込み / 消去 (自動プログラムアルゴリズム *) が可能

*: 自動プログラムアルゴリズム = Embedded Algorithm™

ここでは、FR-CPU からこのフラッシュメモリを使用する場合について説明しています。

ROM ライタからこのフラッシュメモリを使用する場合の詳細につきましては、別途 ROM ライタの取扱説明書を参照してください。

■ 自動アルゴリズム実行状態

CPU プログラミングモードにて自動アルゴリズムを起動した場合には、ハードウェアシーケンスフラグにより自動アルゴリズムの動作状態を知ることができます。

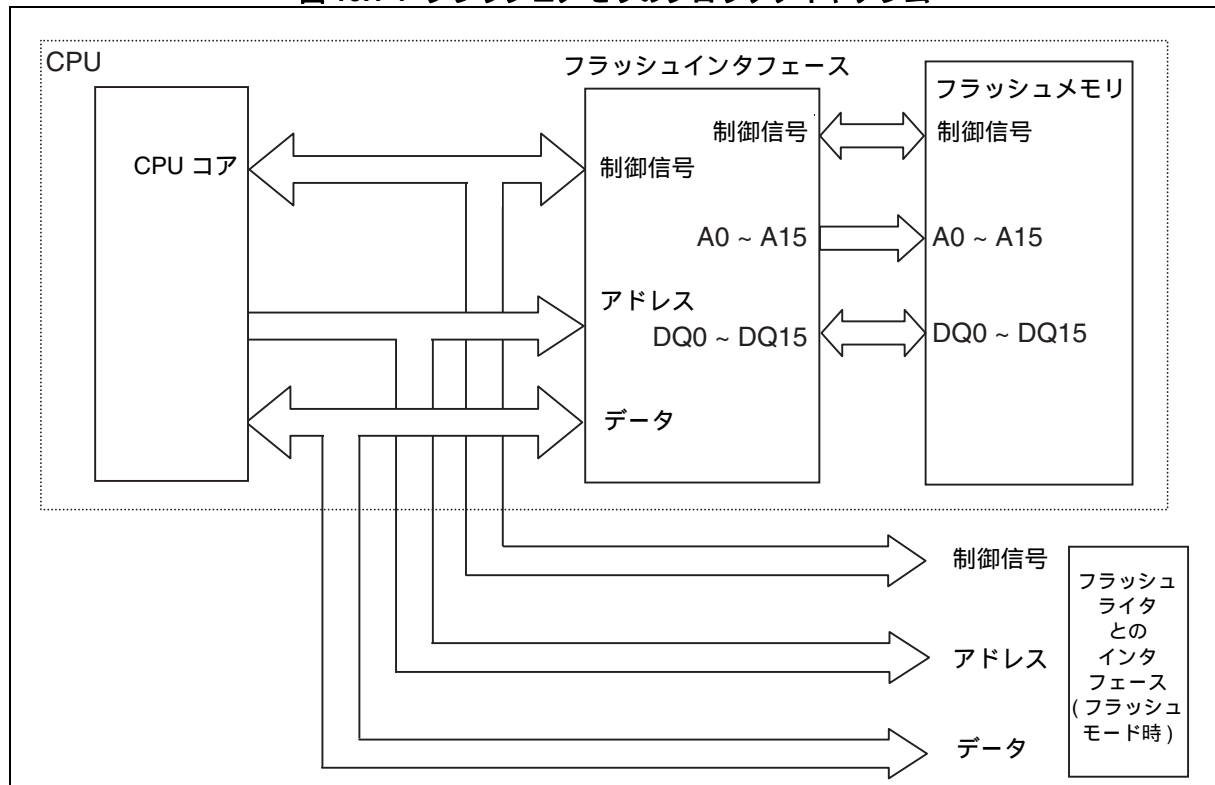
ハードウェアシーケンスフラグについては「19.4 自動アルゴリズム起動方法」を参照してください。

■ ROM ライタによる書込み

このフラッシュメモリは、ROM ライタを用いてデバイス外部より書込みを行うことができます。この状態では、単体フラッシュメモリ製品の MBM29LV400TC 相当の端子機能がデバイス外部端子に割り当てられ、FR-CPU は動作を停止します。CPU モードとはアドレス線の接続が変更され、メモリ領域内のマッピングが変化します。詳細については『対応 ROM ライタの仕様書』を参照してください。

■ フラッシュメモリのブロックダイアグラム

図 19.1-1 フラッシュメモリのブロックダイアグラム



■ フラッシュメモリのセクタ構成

フラッシュメモリのアドレスマッピングは、FR-CPU からのアクセス時と ROM ライタでのアクセス時とは異なります。図 19.1-2 と表 19.1-1 に FR-CPU からのアクセス時におけるマッピングを、図 19.1-3 に ROM ライタからのアクセス時におけるマッピングを示します。

● FR-CPU からのアクセス時におけるマッピング

図 19.1-2 FR-CPU からのアクセス時におけるマッピング

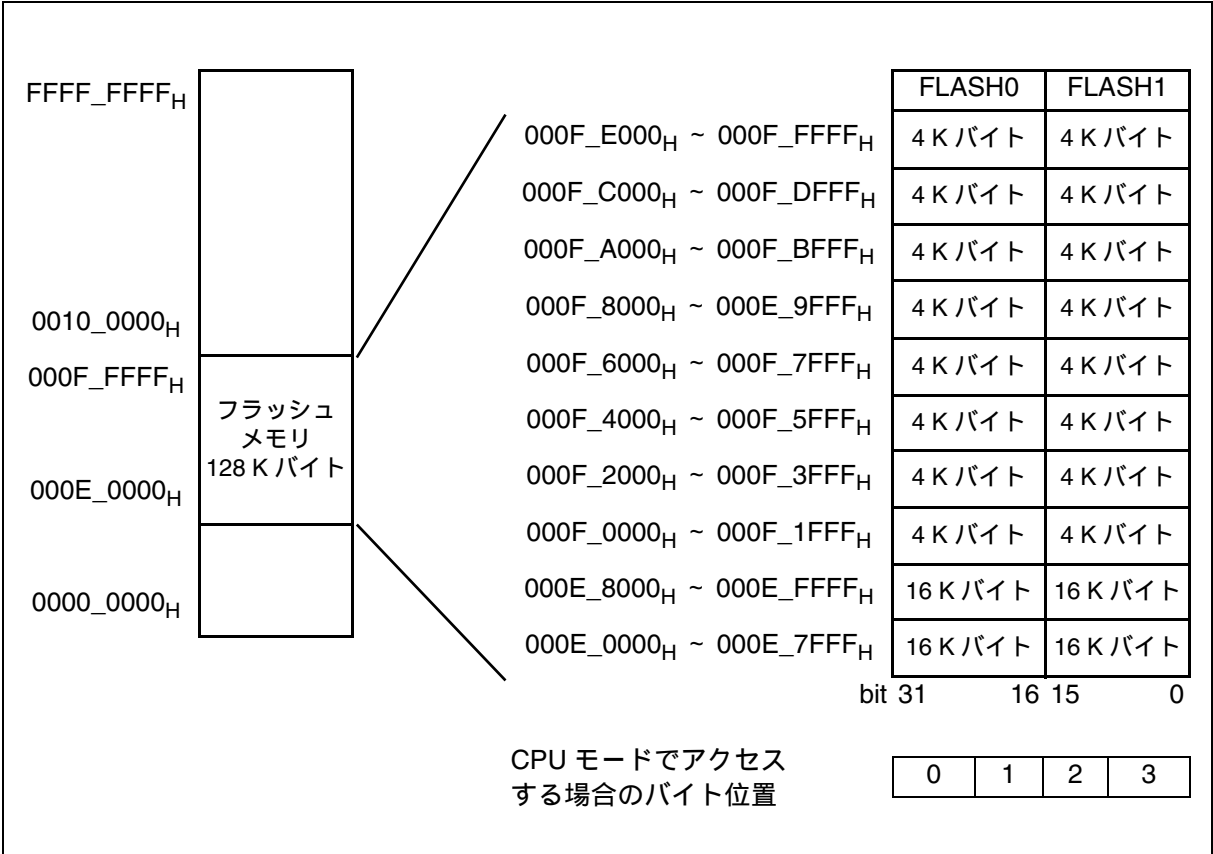


表 19.1-1 セクタアドレス一覧 (FR-CPU からのアクセス時)

フラッシュ	アドレス範囲	対応ビット位置	セクタ容量
FLASH1	F_E002 _H , F_E003 _H ~ F_FFFE _H , F_FFFF _H	bit15 ~ bit0	4 K バイト
	F_C002 _H , F_C003 _H ~ F_DFFE _H , F_DFFF _H	bit15 ~ bit0	4 K バイト
	F_A002 _H , F_A003 _H ~ F_BFFE _H , F_BFFF _H	bit15 ~ bit0	4 K バイト
	F_8002 _H , F_8003 _H ~ F_9FFE _H , F_9FFF _H	bit15 ~ bit0	4 K バイト
	F_6002 _H , F_6003 _H ~ F_7FFE _H , F_7FFF _H	bit15 ~ bit0	4 K バイト
	F_4002 _H , F_4003 _H ~ F_5FFE _H , F_5FFF _H	bit15 ~ bit0	4 K バイト
	F_2002 _H , F_2003 _H ~ F_3FFE _H , F_3FFF _H	bit15 ~ bit0	4 K バイト
	F_0002 _H , F_0003 _H ~ F_1FFE _H , F_1FFF _H	bit15 ~ bit0	4 K バイト
	E_8002 _H , E_8003 _H ~ E_FFFE _H , E_FFFF _H	bit15 ~ bit0	16 K バイト
	E_0002 _H , E_0003 _H ~ E_7FFE _H , E_7FFF _H	bit15 ~ bit0	16 K バイト
FLASH0	F_E000 _H , F_E001 _H ~ F_FFFC _H , F_FFFD _H	bit31 ~ bit16	4 K バイト
	F_C000 _H , F_C001 _H ~ F_DFFC _H , F_DFFD _H	bit31 ~ bit16	4 K バイト
	F_A000 _H , F_A001 _H ~ F_BFFC _H , F_BFFD _H	bit31 ~ bit16	4 K バイト
	F_8000 _H , F_8001 _H ~ F_9FFC _H , F_9FFD _H	bit31 ~ bit16	4 K バイト
	F_6000 _H , F_6001 _H ~ F_7FFC _H , F_7FFD _H	bit31 ~ bit16	4 K バイト
	F_4000 _H , F_4001 _H ~ F_5FFC _H , F_5FFD _H	bit31 ~ bit16	4 K バイト
	F_2000 _H , F_2001 _H ~ F_3FFC _H , F_3FFD _H	bit31 ~ bit16	4 K バイト
	F_0000 _H , F_0001 _H ~ F_1FFC _H , F_1FFD _H	bit31 ~ bit16	4 K バイト
	E_8000 _H , E_8001 _H ~ E_FFFC _H , E_FFFD _H	bit31 ~ bit16	16 K バイト
	E_0000 _H , E_0001 _H ~ E_7FFC _H , E_7FFD _H	bit31 ~ bit16	16 K バイト

● ROM ライタからのアクセス時におけるマッピング

図 19.1-3 ROM ライタからのアクセス時におけるアドレスマッピング

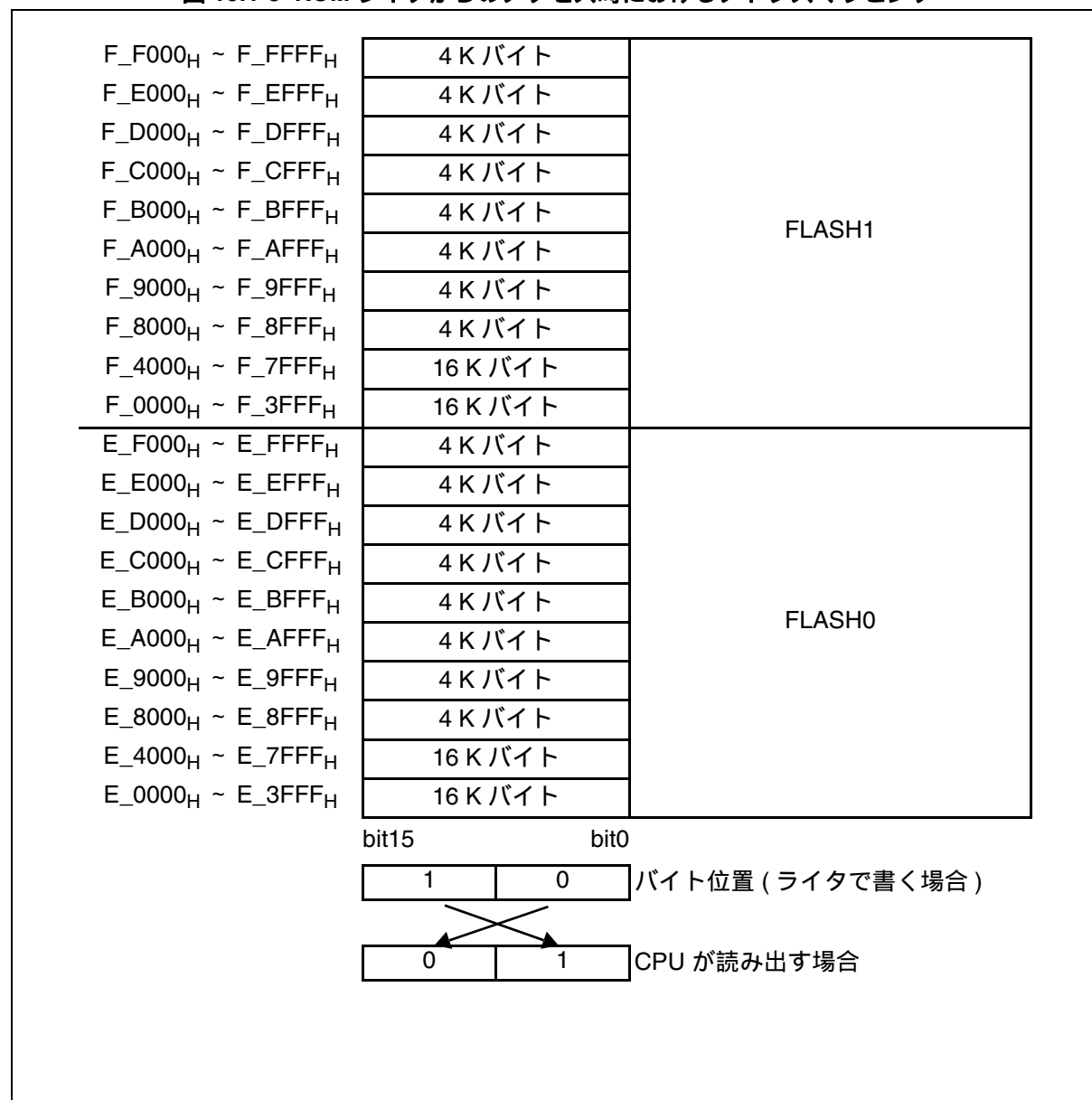


表 19.1-2 セクタアドレス一覧 (ROM ライタからのアクセス時)

フラッシュ	アドレス範囲	対応ビット位置	セクタ容量
FLASH1	F_F000 _H ~ F_FFFF _H	bit15 ~ bit0	4 K バイト
	F_E000 _H ~ F_EFFF _H	bit15 ~ bit0	4 K バイト
	F_D000 _H ~ F_DFFF _H	bit15 ~ bit0	4 K バイト
	F_C000 _H ~ F_CFFF _H	bit15 ~ bit0	4 K バイト
	F_B000 _H ~ F_BFFF _H	bit15 ~ bit0	4 K バイト
	F_A000 _H ~ F_AFFF _H	bit15 ~ bit0	4 K バイト
	F_9000 _H ~ F_9FFF _H	bit15 ~ bit0	4 K バイト
	F_8000 _H ~ F_8FFF _H	bit15 ~ bit0	4 K バイト
	E_4000 _H ~ E_7FFF _H	bit15 ~ bit0	16 K バイト
	E_0000 _H ~ E_3FFF _H	bit15 ~ bit0	16 K バイト
FLASH0	F_F000 _H ~ F_FFFF _H	bit15 ~ bit0	4 K バイト
	F_E000 _H ~ F_EFFF _H	bit15 ~ bit0	4 K バイト
	F_D000 _H ~ F_DFFF _H	bit15 ~ bit0	4 K バイト
	F_C000 _H ~ F_CFFF _H	bit15 ~ bit0	4 K バイト
	F_B000 _H ~ F_BFFF _H	bit15 ~ bit0	4 K バイト
	F_A000 _H ~ F_AFFF _H	bit15 ~ bit0	4 K バイト
	F_9000 _H ~ F_9FFF _H	bit15 ~ bit0	4 K バイト
	F_8000 _H ~ F_8FFF _H	bit15 ~ bit0	4 K バイト
	E_4000 _H ~ E_7FFF _H	bit15 ~ bit0	16 K バイト
	E_0000 _H ~ E_3FFF _H	bit15 ~ bit0	16 K バイト

19.2 レジスタ

フラッシュメモリにはフラッシュメモリステータスレジスタ (FLCR) とフラッシュメモリウェイトレジスタ (FLWC) の 2 つのレジスタがあります。

■ フラッシュメモリのレジスタ一覧

図 19.2-1 に、フラッシュメモリのレジスタ一覧を示します。

図 19.2-1 フラッシュメモリのレジスタ一覧

bit7	bit0	フラッシュメモリステータスレジスタ (FLCR)
bit7	bit0	フラッシュメモリウェイトレジスタ (FLWC)

表 19.2-1 アドレスマップ

アドレス	オフセット				ブロック
	+ 0	+ 1	+ 2	+ 3	
FLCR[R/W] B 007000 _H	01101000	-	-	-	フラッシュ インタフェース
FLWC[R/W] B 007004 _H	00000011	-	-	-	

19.2.1 フラッシュメモリステータスレジスタ (FLCR)

フラッシュメモリステータスレジスタ (FLCR) は、フラッシュメモリの動作状態を示すレジスタです。

■ フラッシュメモリステータスレジスタ (FLCR)

このレジスタは、CPU に対する割込みの制御や、フラッシュメモリへの書込みの制御を行います。CPU でのみアクセスすることが可能です。ライター搭載時はアクセスできません。

このレジスタは、リードモディファイライト (RMW) 系命令ではアクセスしないでください。

図 19.2-2 に、フラッシュメモリステータスレジスタ (FLCR) のビット構成を示します。

図 19.2-2 フラッシュメモリステータスレジスタ (FLCR) のビット構成

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
007000 _H	-	-	-	-	-	-	WE	-	01101000 _B
	R/W	R/W	R/W	R	R	R/W	R/W	R/W	
R/W : リード/ライト可能									
R : リードオンリ									

以下に、フラッシュメモリステータスレジスタ (FLCR) の各ビットの機能を説明します。

[bit7] - : 予約ビット

ビットは予約ビットです。このビットは常に "0" を書き込んでください。

[bit6, bit5] - : 予約ビット

ビットは予約ビットです。このビットは常に "1" を書き込んでください。

[bit4, bit3] - : 予約ビット

ビットは予約ビットです。このビットは読出しのみで書込みは動作に影響を与えません。

[bit2] - : 予約ビット

ビットは予約ビットです。このビットは常に "0" を書き込んでください。

[bit1] WE : CPU モードにおけるフラッシュメモリへのデータおよびコマンドの書込みを制御します。

本ビットが "0" の間は、フラッシュメモリへのデータおよびコマンドの書込みは、すべて無効になります。

本ビットが "1" の間は、フラッシュメモリへのデータおよびコマンドの書込みが有効となり、自動アルゴリズムの起動が可能となります。

ただし、フラッシュメモリからのデータの書込みは 16 ビットおよび 8 ビットアクセスになります。フラッシュメモリへの書込みアクセスは 16 ビットおよび 8 ビットアクセスのみを使用してください。32 ビットのアクセスは禁止です。

本ビットの書換えは、必ずハードウェアシーケンスフラグにより自動アルゴリズムが停止していることを確認した後に行ってください。

WE	機能
0	フラッシュメモリへの書込み禁止
1	フラッシュメモリへの書込み許可

● 制限事項

本レジスタの WE ビットを書き換える場合には必ず下記の命令シーケンスを実行してください。

また、本レジスタ書換え時には、DMA、割込み、スタンバイ動作を行わないでください。

命令シーケンス：

```

1 : LDI  #(_FLWC),  R0
2 : LDI  #0x01,     R1
3 : STB  R1,         @R0          //set FLWC(WTC=1)

4 : LDI  #(_FLCR),  R0
5 : LDI  #0x62,     R1
6 : STB  r1,         @R0          //set FLCR(WE=1)

7 : LDI  #(_FLWC),  R0
8 : LDI  #0x03,     R1
9 : STB  R1,         @R0          //set FLWC(WTC=3)

```

[bit0] - : 予約ビット

このビットは予約ビットです。このビットは常に "0" を書き込んでください。

19.2.2 フラッシュウェイトレジスタ (FLWC)

フラッシュウェイトレジスタ (FLWC) は、CPU モードにおいてフラッシュメモリのウェイト制御を行います。

■ フラッシュウェイトレジスタ (FLWC)

図 19.2-3 に、フラッシュウェイトレジスタ (FLWC) のビット構成を示します。

図 19.2-3 フラッシュウェイトレジスタ (FLWC) のビット構成

アドレス	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	初期値
007004 _H	-	-	FAC1	FAC0	-	WTC2	WTC1	WTC0	00000011 _B
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : リード/ライト可能
R : リードオンリ

以下に、フラッシュウェイトレジスタ (FLWC) の各ビットの機能を説明します。

[bit7, bit6] - : 予約ビット

これらのビットは予約ビットです。これらのビットは常に "0" を書き込んでください。

[bit5, bit4] FAC1, FAC0 : 内部書込み信号のパルス幅制御ビット

FAC1	FAC0	ATDIN	EQIN	(初期値)
0	0	0.5 クロック	1 クロック	
0	1	1 クロック	1.5 クロック	
1	0	1.5 クロック	2 クロック	
1	1	2 クロック	2.5 クロック	

(注意事項) ・ ATDIN, EQIN は内部書込み信号です。通常は初期値設定で使用してください。

・ マスク ROM 品は常に "00_B" を書き込んでください。

[bit3] - : 予約ビット

ビットは予約ビットです。このビットは常に "0" を書き込んでください。

[bit2 ~ bit0] WTC2 ~ WTC0 : ウェイトサイクル制御ビット

本ビットはフラッシュメモリアクセス時のウェイトサイクル数を制御します。

WTC2	WTC1	WTC0	ウェイト サイクル	読出し時	書込み時	
0	0	0	-	設定禁止	設定禁止	
0	0	1	1	33 MHz まで動作可能	設定禁止	
0	1	0	2	33 MHz まで動作可能	設定禁止	
0	1	1	3	33 MHz まで動作可能	33 MHz まで動作可能	(初期値)
1	0	0	4	設定禁止	設定禁止	
1	0	1	5	設定禁止	設定禁止	
1	1	0	6	設定禁止	設定禁止	
1	1	1	7	設定禁止	設定禁止	

- (注意事項) ・ FAC1, FAC0 で設定したサイクル以上になるように設定してください。
- ・ 初期値は書込み用の設定になっています。読出しのみを行う場合 (FLCR WE=0 設定の場合) ウェイトサイクル 1 (WTC2, WTC1, WTC0 = 001_B) と設定することで最高速で読出しが可能になります。
 - ・ MASK 品の初期値は , ウェイトサイクル 3 (WTC2, WTC1, WTC0 = 011_B) で読み出されます。ウェイトサイクル 1 (WTC2, WTC1, WTC0 = 001_B) と設定することで最高速で読出しが可能になります。

19.3 フラッシュメモリのアクセスモード

FR-CPU によりアクセスする場合，以下の 2 種類のアクセスモードが存在します。

- ROM モード

ワード (32 ビット) 長データを一括で読み出すことができますが，書込みはできません。

- プログラミングモード

ワード (32 ビット) 長アクセスは禁止されていますが，ハーフワード (16 ビット)，バイト (8 ビット) 長での書込みが可能になります。

■ FR-CPU ROM モード (32/16/8 ビット，読出しのみ)

FR-CPU の内蔵 ROM として機能するモードです。ワード (32 ビット) 長データの一括読出しが可能となりますが，フラッシュメモリへの書込み，自動アルゴリズムの起動はできません。

- モードの指定方法

- フラッシュメモリステータスレジスタの "WE" ビットが "0" のとき，このモードとなります。
- CPU 動作時でのリセット解除後は常にこのモードになります。
- CPU 動作時以外ではこのモードにすることができません。

- 動作内容

フラッシュメモリ領域の読出し時，メモリよりワード (32 ビット) 長データを一括で読出し可能です。

読出しにかかるサイクル数は最短で 2 サイクル/1 ワード (1 ウェイト) です。これにより FR-CPU に対し，ウェイトなしで命令を供給することができます。

- 制限事項

ROM ライタ書込み時とはアドレスの与え方およびエンディアンが異なります。また，このモードではフラッシュメモリへの書込み，自動アルゴリズムの起動はできません。

■ FR-CPU プログラミングモード (16/8 ビット, 読出し / 書込み)

データの消去 / 書込みが可能になるモードです。ただし、データの消去 / 書込みを実行しているバンク上でのプログラム実行はできません。

● モード指定の方法

- フラッシュメモリステータスレジスタ (FLCR) の "WE" ビットが "1" のとき、このモードになります。
- CPU 動作時でのリセット解除後は "WE" ビットは "0" になっています。このモードを指定する場合には "1" を書き込んでください。再度 "0" を書き込むか、リセットの発生により "WE" ビットが "0" になると、ROM モードに戻ります。

● 動作内容

- フラッシュメモリ領域の読出し時、フラッシュメモリからワード (32 ビット) 長データを一括で読出し可能です。ただし、読出しにかかるサイクル数はフラッシュメモリステータスレジスタ (FLCR) の書込み時の設定 4 サイクル / ハーフワード (3 ウェイト) になります。
- フラッシュメモリへのコマンド書込みにより、自動アルゴリズムを起動することができます。自動アルゴリズムの起動により、フラッシュメモリの消去 / 書込みが可能となります。自動アルゴリズムの詳細については、「19.4 自動アルゴリズム起動方法」および「19.5 自動アルゴリズム実行状態」を参照してください。

● 制限事項

ROM ライタ書込み時とはアドレスの与え方およびエンディアンが異なります。

プログラミングアクセスモード切換えで、WE ビットを書き換える際には「19.2.1 フラッシュメモリステータスレジスタ (FLCR)」の「制限事項」に従って書き換えてください。

19.4 自動アルゴリズム起動方法

フラッシュメモリに対する書込み / 消去は、フラッシュメモリ自身が持つ自動アルゴリズムを起動することで行います。

■ コマンドシーケンス

自動アルゴリズムを起動するには、フラッシュメモリに対し、1 回 ~ 6 回のハーフワード (16 ビット) データの連続書込みを実行します。これをコマンドとよびます。不正なアドレスとデータを書き込んだり、誤った順番でアドレスとデータを書き込んだりすると、フラッシュメモリは読出しモードにリセットされます。

表 19.4-1、表 19.4-2 に、フラッシュメモリの書込み / 消去時に使用するコマンドの一覧を示します。

表 19.4-1 CPU モードでの FLASH0 書込みコマンドシーケンス

コマンド シーケンス	バス ライト サイクル	1st バスライト サイクル		2nd バスライト サイクル		3rd バスライト サイクル		4th バスライト / ライトサイクル		5th バスライト サイクル		6th バスライト サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し / リセット	1	XXXXXX _H	F0 _H	-	-	-	-	-	-	-	-	-	-
読出し / リセット	4	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	F0 _H	RA	RD	-	-	-	-
書込み	4	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	A0 _H	PA	PD	-	-	-	-
チップ消去	6	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	80 _H	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	10 _H
セクタ消去	6	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	80 _H	F5554 _H	AA _H	EAAA8 _H	55 _H	SA	30 _H
セクタ消去一時停止		アドレス = "XXXXXX _H ", データ = "B0 _H " の入力で、セクタ消去一時停止											
セクタ消去再開		アドレス = "XXXXXX _H ", データ = "30 _H " の入力で、セクタ消去再開											
Auto Select	3	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	90 _H	-	-	-	-	-	-
連続モード	3	F5554 _H	AA _H	EAAA8 _H	55 _H	F5554 _H	20 _H	-	-	-	-	-	-
連続書込み	2	XXXXXX _H	A0 _H	PA	PD	-	-	-	-	-	-	-	-
連続モード リセット	2	XXXXXX _H	90 _H	XXXXXX _H	F0 _H or 00 _H	-	-	-	-	-	-	-	-

表 19.4-2 CPU モードでの FLASH1 書き込みコマンドシーケンス

コマンド シーケンス	バス ライト サイクル	1st バスライト サイクル		2nd バスライト サイクル		3rd バスライト サイクル		4th バスリード/ ライトサイクル		5th バスライト サイクル		6th バスライト サイクル	
		アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ	アドレス	データ
読出し/ リセット	1	XXXXXX _H	F0 _H	-	-	-	-	-	-	-	-	-	-
読出し/ リセット	4	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	F0 _H	RA	RD	-	-	-	-
書き込み	4	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	A0 _H	PA	PD	-	-	-	-
チップ消去	6	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	80 _H	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	10 _H
セクタ消去	6	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	80 _H	F5556 _H	AA _H	EAAAA _H	55 _H	SA	30 _H
セクタ消去一時停止		アドレス="XXXXXX _H ", データ="B0 _H " の入力で, セクタ消去一時停止											
セクタ消去再開		アドレス="XXXXXX _H ", データ="30 _H " の入力で, セクタ消去再開											
Auto Select	3	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	90 _H	-	-	-	-	-	-
連続モード	3	F5556 _H	AA _H	EAAAA _H	55 _H	F5556 _H	20 _H	-	-	-	-	-	-
連続書き込み	2	XXXXXX _H	A0 _H	PA	PD	-	-	-	-	-	-	-	-
連続モード リセット	2	XXXXXX _H	90 _H	XXXXXX _H	F0 _H or 00 _H	-	-	-	-	-	-	-	-

ワードモード/バイトモード共にコマンドは同じで, 表記されていないビットのデータは任意となります。

RA : 読出しアドレス

PA : 書き込みアドレス

SA : セクタアドレス (セクタ内の任意の 1 アドレスを指定。)

RD : 読出しデータ

PD : 書き込みデータ

● リード (読出し) / リセットコマンド

タイミングリミット超過から読出しモードへ復帰するには, 読出し/リセットコマンドシーケンスを発行することで行います。読出しサイクルでフラッシュメモリからデータを読み出します。フラッシュメモリは, ほかのコマンドが入力されるまで, 読出し状態を保ちます。

フラッシュメモリは, 電源投入時自動的に読出し/リセットにセットされます。この場合は, データ読出しにコマンドは必要ありません。

● プログラム (書き込み)

CPU プログラミングモードでは, ハーフワード単位を基本として書き込みを行います。書き込みは 4 回のバス動作で行われます。コマンドシーケンスには 2 つの " アンロック " サイクルがあり, 書き込みセットアップコマンドと書き込みデータサイクルが続きます。そして, 最後の書き込みサイクルでメモリへの書き込みが開始されます。

自動書き込みアルゴリズムコマンドシーケンス実行後は, フラッシュメモリはそれ以上の外部からの制御を必要としません。フラッシュメモリは, 自動的に内部で作られた適切な書き込みパルスを発生し, 書き込まれたセルのマージンを検証します。自動書き込み動作は, データポーリング機能により, bit7 のデータがこのビットに書き込んだデータに一致したとき終了し (「 19.5 自動アルゴリズム実行状態 」 の 「 ■ ハードウェアシーケンスフラグ 」 を参照), このときをもって読出しモードに戻り, これ以上書き込みアドレス

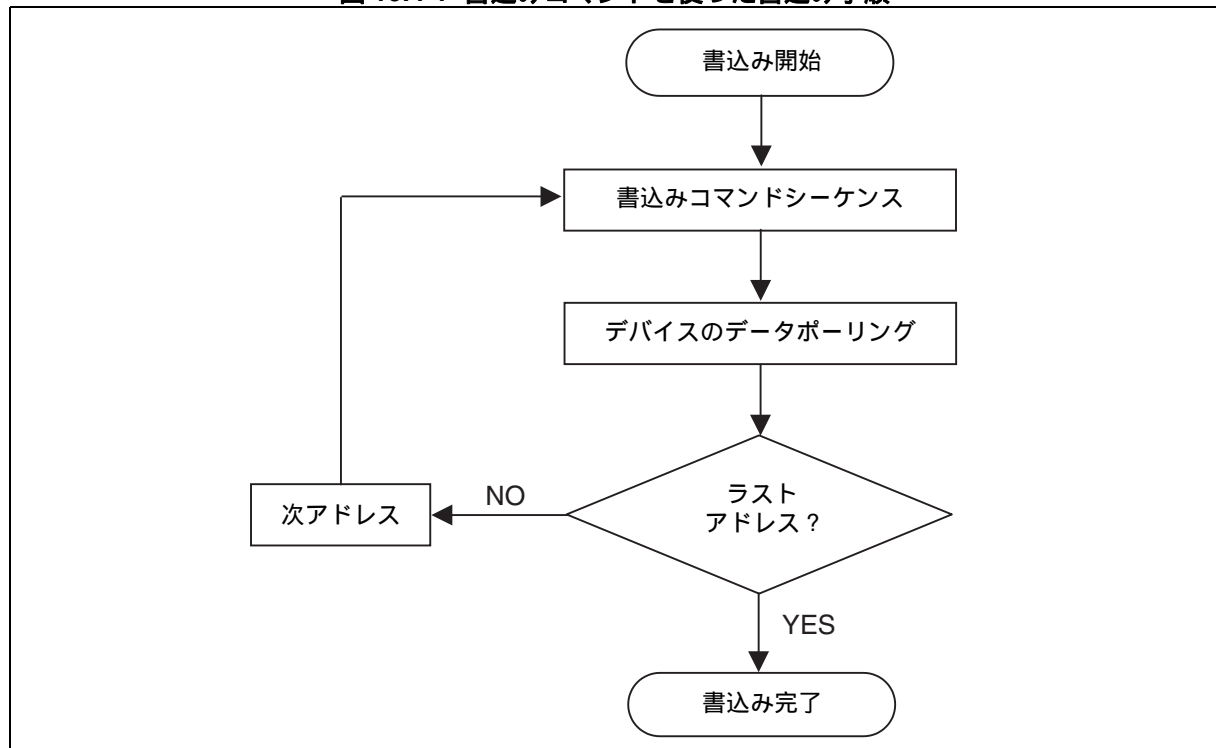
は受け付けなくなります。この結果、フラッシュメモリはこの時点で次の有効アドレスを要求します。このようにデータポーリングは、メモリが書き込み中であることを示します。

書き込み中はフラッシュメモリに書き込まれたすべてのコマンドが無視されます。もし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。

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。書き込みによって、データ "0" をデータ "1" に戻すことはできません。データ "0" にデータ "1" を書き込むと、データポーリングアルゴリズムにより、素子が不良と判定されるか、または見かけ上データ "1" が書き込まれたように見えるかのどちらかです。しかし、リセット / 読出しモード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" データにすることができます。

書き込みコマンドを使った書き込み手順を図 19.4-1 に示します。

図 19.4-1 書き込みコマンドを使った書き込み手順



● チップ消去

チップ消去（全セクタの一括消去）は、6 回のアクセスで行われます。最初に、2 つの "アンロック" サイクルがあり、引き続き "セットアップ" コマンドが書き込まれます。チップ消去コマンドまでに、さらに 2 つの "アンロック" サイクルが続けられます。

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き込みを行う必要はありません。自動消去アルゴリズム実行中には、フラッシュメモリは自動的にすべてのセルを消去する前に "0" のパターンに書き込んで検証します（プリプログラム）。この動作中には、フラッシュメモリは外部からの制御は必要としません。

自動消去はコマンドシーケンス中の書き込みで開始され、bit7 が "1" になったときに終了し、このときフラッシュメモリは読出しモードに戻ります。チップ消去時間は「セクタ消去時間 × 全セクタ数 + チップ書き込み時間（プリプログラム）」となります。

チップ消去コマンドを使ったチップ消去手順を図 19.4-2 に示します。

● セクタ消去

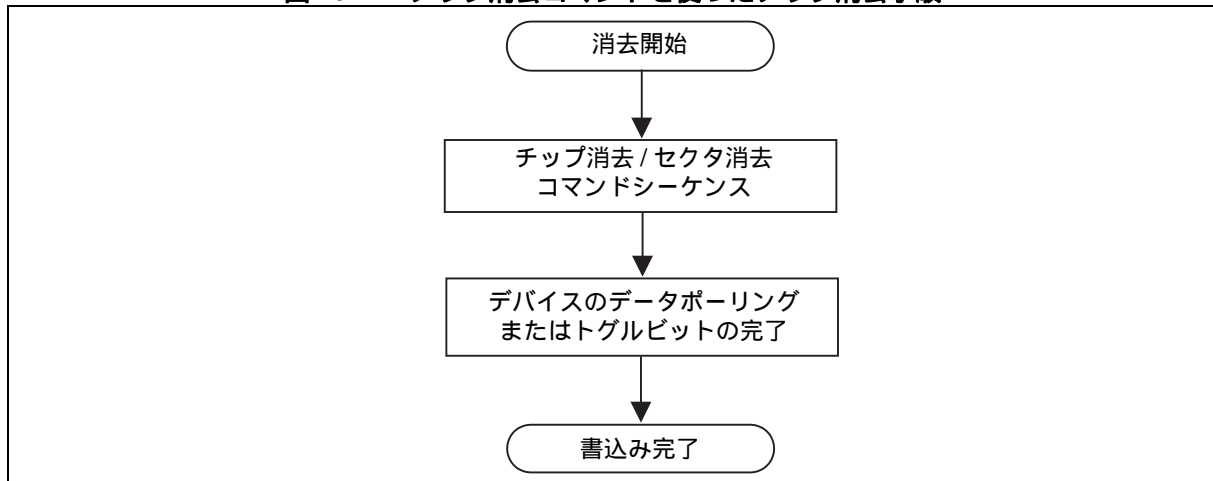
セクタ消去は、6 回のアクセスで行われます。2 つの "アンロック" サイクルがあり、引き続き "セットアップ" コマンドを書き込んで、その後さらに 2 つの "アンロック" サイクルが続き、6 サイクル目にセクタ消去コマンドを入力することによりセクタ消去が始まります。最後のセクタ消去コマンドの書き込みから 50 μs のタイムアウト期間中、次のセクタ消去コマンドの受け付けが可能です。

複数のセクタ消去は前述したような 6 つのバスサイクルを書き込むことで同時に受け付け可能となります。このシーケンスは同時に消去するセクタのアドレスにセクタ消去コマンド (30_H) を引き続き書き込むことで行います。最後のセクタ消去コマンドの書き込みから 50 μs のタイムアウト期間終了により、セクタ消去が開始されます。すなわち、複数のセクタを同時に消去する場合には、次の消去セクタをそれぞれ 50 μs 以内に入力する必要があります。それ以降ではコマンド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マンドが有効かどうかは bit3 にてモニタ可能です（「19.5 自動アルゴリズム実行状態」の「ハードウェアシーケンスフラグ」を参照）。

タイムアウト中のセクタ消去コマンドまたは消去一時停止以外のいかなるコマンドも読出しにリセットし、その前のコマンドシーケンスは無視します。この場合は、そのセクタを再度消去することにより消去が完了します。セクタ消去バッファへのセクタアドレス入力は、セクタのどのような組合せや数 (0 ~ 6) からでも実行可能です。セクタ消去では、消去前にユーザがフラッシュメモリに書き込みを行う必要はありません。フラッシュメモリは自動的に消去されるセクタ内のすべてのセルに書き込みを行います（プリプログラム）。また、セクタ消去中はほかの消去されないセクタは何の影響も受けません。これらの動作中は、フラッシュメモリは外部からの制御は必要としません。

自動セクタ消去は、最後のセクタ消去コマンドの書き込みから 50 μs のタイムアウト期間の後に開始され、bit7 のデータが "1" になったとき（「19.5 自動アルゴリズム実行状態」の「ハードウェアシーケンスフラグ」を参照）終了し、フラッシュメモリは読出しモードに戻ります。ほかのコマンドは無視されます。データポーリングは消去されたセクタ内のどのアドレスでも働きます。複数セクタ消去時間は「（セクタ消去時間 + セクタ書き込み時間（プリプログラム））× 消去セクタ数」となります。

図 19.4-2 チップ消去コマンドを使ったチップ消去手順



● 消去一時停止

消去一時停止コマンドは、ユーザがセクタ消去中にフラッシュメモリの自動アルゴリズムを一時停止して、消去中でないセクタからのデータ読出しと書き込みを可能にするものです。このコマンドはセクタ消去中のみ有効で、チップ消去や書き込みの動作中は無視されます。消去一時停止コマンド (B0_H) はセクタ消去コマンド (30_H) 後のセクタ消去タイムアウト期間を含むセクタ消去動作中にのみ有効です。このコマンドがタイムアウト期間中に入力されると直ちにタイムアウトを終了し、消去動作を中断します。消去再開コマンドが書き込まれると消去動作が再開されます。消去一時停止、消去再開コマンドの入力の際のアドレスは任意であってかまいません。

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、フラッシュメモリが消去動作を停止するために最大 20 μs の時間がかかります。フラッシュメモリが消去一時停止モードに入ると、レディ / ビジー出力と bit7 が "1" を出力し、bit6 がトグル動作をやめます。消去しているセクタのアドレスを入力し、bit6 と bit7 の読出し値をモニタすることによって、消去動作を停止しているかどうかを確かめられます。さらに、消去一時停止コマンドの書き込みは無視されます。消去動作が停止したとき、フラッシュメモリは消去一時停止読出しモードになります。このモードでのデータの読出しはデータが消去一時停止していないセクタに有効となりますが、それ以外は標準的な読出しと同じです。消去一時停止読出し中、その消去一時停止したセクタからの連続的な読出しに対しては、bit2 はトグル動作をします（「19.5 自動アルゴリズム実行状態」の「ハードウェアシーケンスフラグ」を参照）。

消去一時停止読出しモードに入った後、ユーザは書き込みのコマンドシーケンスを書き込むことによりフラッシュメモリに書き込みができます。この書き込みモードは、消去一時停止書き込みモードとなります。このモードでの書き込みは、データが消去一時停止していないセクタに有効となりますが、それ以外は通常のバイト書き込みと同じです。消去一時停止書き込みモード中、その消去一時停止したセクタからの連続的な読出しに対しては、bit2 はトグル動作をします。消去一時停止ビット (bit6) によって検出できます。

< 注意事項 >

bit6 はどのようなアドレスに対しても読出し可能ですが、bit7 は書き込みアドレスに対して読出しを行わなければなりません。セクタ消去動作を再開するためには、再開コマンド (30_H) を入力する必要があります。この時点でさらに再開コマンドを入力しても無視されます。他方、消去一時停止コマンドはフラッシュメモリが消去再開した後に入力することができます。

19.5 自動アルゴリズム実行状態

このフラッシュメモリでは、書込み / 消去のフローを自動アルゴリズムで行うため、フラッシュメモリ内部の動作状態や動作完了したことをフラッシュメモリ外部に知らせるハードウェアを持ちます。

■ 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、自動アルゴリズム実行中にフラッシュメモリの任意のアドレス (バイトアクセス時は奇数アドレス) を読み出すことによりデータとして得られます。データのうち、有効ビットは 5 ビットで、それぞれが自動アルゴリズムの状態を示します。

ハードウェアシーケンスフラグの構成を図 19.5-1 に示します。

なお、ハードウェアシーケンスフラグ状態一覧を表 19.5-1 に示します。

図 19.5-1 ハードウェアシーケンスフラグの構成

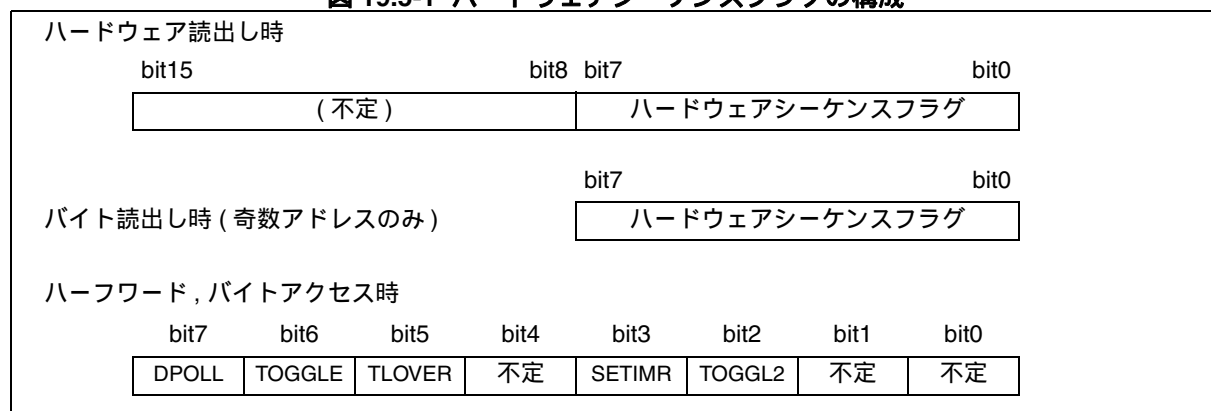


表 19.5-1 ハードウェアシーケンスフラグ状態一覧

状態		DPOLL (bit7)	TOGGLE (bit6)	TLOVER (bit5)	SETIMR (bit3)	TOGGL2 (bit2)
実行中	自動書込み動作	反転データ	トグル	0	0	1
	自動消去時の書込み / 消去動作	0	トグル	0	1	トグル
	消去一時停止	1	1	0	0	トグル
	読出し (消去中のセクタ)	データ	データ	データ	データ	データ
	書込み (消去していないセクタ)	反転データ	トグル	0	0	1 ^{*1}
タイムリミット超過	自動書込み動作	反転データ	トグル	1	0	1
	自動消去時の書込み / 消去動作	0	トグル	1	1	*2

*1：消去一時停止書込み中、その書込みされているアドレスの読出しに対し、bit2 は論理 "1" を出力します。しかし、消去一時停止しているセクタからの連続的な読出しに対し bit2 はトグル動作をします。

*2：bit5 が "1" のとき (タイムリミット超過)、書込み / 消去中セクタへの連続的な読出しに対しては bit2 はトグル動作をし、ほかのセクタへの読出しに対してはトグル動作しません。

以下に、各ビットの説明をします。

[bit7] DPOLL: データポーリング

- 自動書込み動作中
自動書込みアルゴリズム実行中に読出しを行うと、フラッシュメモリは bit7 に最後に書き込まれたデータの反転データを出力します。自動書込みアルゴリズム終了時に読出しアクセスを行うと、フラッシュメモリはアドレス信号の指し示す番地の読出しデータの bit7 を出力します。
- 自動消去動作中
自動消去アルゴリズム実行中に読出しを行うと、フラッシュメモリはアドレス信号の指し示す番地によらず "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中
セクタ消去一時停止中に読出しを行うと、フラッシュメモリはアドレス信号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、アドレス信号の指し示す番地の読出し値の bit7 を出力します。後述の bit6 のトグルビットとともに参照することで、現在セクタ消去一時停止状態にあるか否か、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 >

自動アルゴリズムの動作が終了に近づくとき、bit7(データポーリング) は読出し動作中、非同期的に変化します。これはフラッシュメモリが動作状態の情報を bit7 に送り出し、そして確定したデータをその次に送り出すことを意味します。フラッシュメモリが自動アルゴリズムを終了したとき、また bit7 が確定データを出力しているときでも、ほかのビットはまだ不確定です。ほかのビットの確定データは、連続した読出しの実行によって読み出されます。

[bit6] TOGGLE: トグルビット

- 自動書込み / 消去動作中
自動書込みまたは消去アルゴリズム実行中に連続した読出しを行うと、フラッシュメモリは "1" と "0" をトグルする結果を bit6 に出力します。自動書込みまたは消去アルゴリズムが終了すると、連続した読出しに対して bit6 のトグル動作をやめ、有効なデータを出力します。
トグルビットは各コマンドシーケンスの最後の書込みサイクルの後から有効になります。
なお、書込みの際、書き込もうとしたセクタが書換え保証されているセクタの場合は、約 2 μ s の間トグル動作した後、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、もし選択されたすべてのセクタが書込み保証されている場合には、トグルビットは約 100 μ s トグル動作をし、その後データを書き換えしないで読出しモードに戻ります。
- セクタ消去一時停止中
セクタ消去一時停止中に読出しを行うと、フラッシュメモリはアドレス信号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、アドレス信号の指し示す番地の読出し値の bit6 を出力します。

● 制限事項

トグルビットの読出しを行う場合には、必ず下記の命令シーケンスを実行してください。

命令シーケンス：			
1：	LDI	#(読出しアドレス),	R0
2：	NOP		//requisite nop
3：	LDUH	@R0,	R1 //read a toggle data

[bit5] TLOVER: タイミングリミット超過

- 自動書込み / 消去動作中

bit5 は自動アルゴリズムの実行がフラッシュメモリ内部に規定された時間 (内部パルス回数) を超えてしまったことを示します。この状態において bit5 は "1" を出力します。すなわち自動アルゴリズムが動作中で、このフラグが "1" を出力した場合は、書込みまたは消去が失敗したことを示します。

また、bit5 は消去することなく非ブランクの部分に書き込もうとすると、フェイルとなります。この場合、bit7 (データポーリング) から確定データを読むことができず、bit6 (トグルビット) はトグルリングしたままとなります。この状態でタイムリミットを超えると、bit5 に "1" が出力されます。この場合は、フラッシュメモリが不良なのではなく、正しく使用されなかったということを表しています。もし、この状態が発生したときは、リセットコマンドを実行してください。

[bit3] SETIMR: セクタ消去タイマ

- セクタ消去動作中

最初のセクタ消去コマンドシーケンス実行後、セクタ消去ウェイト期間中になります。bit3 は、この期間中の場合は "0" を、セクタ消去ウェイト期間を超えてしまっている場合は "1" を出力します。データポーリングとトグルビットは最初のセクタ消去コマンドシーケンスの実行後から有効となります。

データポーリング機能やトグルビット機能により、消去アルゴリズムが実行中を示している場合にこのフラグが "1" であれば、内部で制御される消去が始まっており、続けてのコマンドライトはデータポーリングかトグルビットが消去の終了を示すまで無視されます (消去一時停止コードの入力のみ受け付けます)。このフラグが "0" の場合、フラッシュメモリは追加のセクタ消去コードの書込みを受け付けます。このことを確認するために、引き続きセクタ消去コードの書込みに先立ってソフトウェアでこのフラグの状態をチェックすることを推奨します。もし、2 回目の状態チェックで "1" であったなら、追加セクタ消去コードは受け付けられていない可能性があります。セクタ消去一時停止中に読出しを行うと、フラッシュメモリはアドレス信号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、アドレス信号の指し示す番地の読出し値の bit3 を出力します。

[bit2] TOGGL2: トグルビット 2

• セクタ消去動作中

このトグルビットは ,bit6 のトグルビットに加えて ,フラッシュメモリが自動消去動作中であるか ,消去一時停止中であるかを検出することに使われます。自動消去中に消去しているセクタから連続して読出しを行うと bit2 がトグル動作をします。フラッシュメモリが消去一時停止読出しモードならば ,消去一時停止しているセクタから連続して読出しを行うと bit2 はトグル動作をします。

フラッシュメモリが消去一時停止書込みモードのときは ,消去一時停止していないセクタからアドレスを連続して読み出すと bit2 は "1" が読み出されます。bit6 は bit2 と違い ,通常の手書き ,消去または消去一時停止書込み動作中にのみトグル動作をします。

例えば ,bit2 と bit6 は ,消去一時停止読出しモードを検出するために一緒に使われます (bit2 はトグル動作をするが ,bit6 はトグル動作をしない)。

さらに ,bit2 は消去しているセクタの検出にも使われます。フラッシュメモリが消去動作のときは ,bit2 は消去しているセクタからの読出しならばトグル動作をします。

■ハードウェアシーケンスフラグの使用例

ハードウェアシーケンスフラグを用いることで、フラッシュメモリ内部の自動アルゴリズムの状態判定が可能です。例として書き込み / 消去判定のフローチャートを、それぞれデータポーリング機能を用いた場合と、トグルビット機能を用いた場合について図 19.5-2、図 19.5-3 に示します。

図 19.5-2 データポーリング機能を用いた書き込み / 消去判定のフローチャート

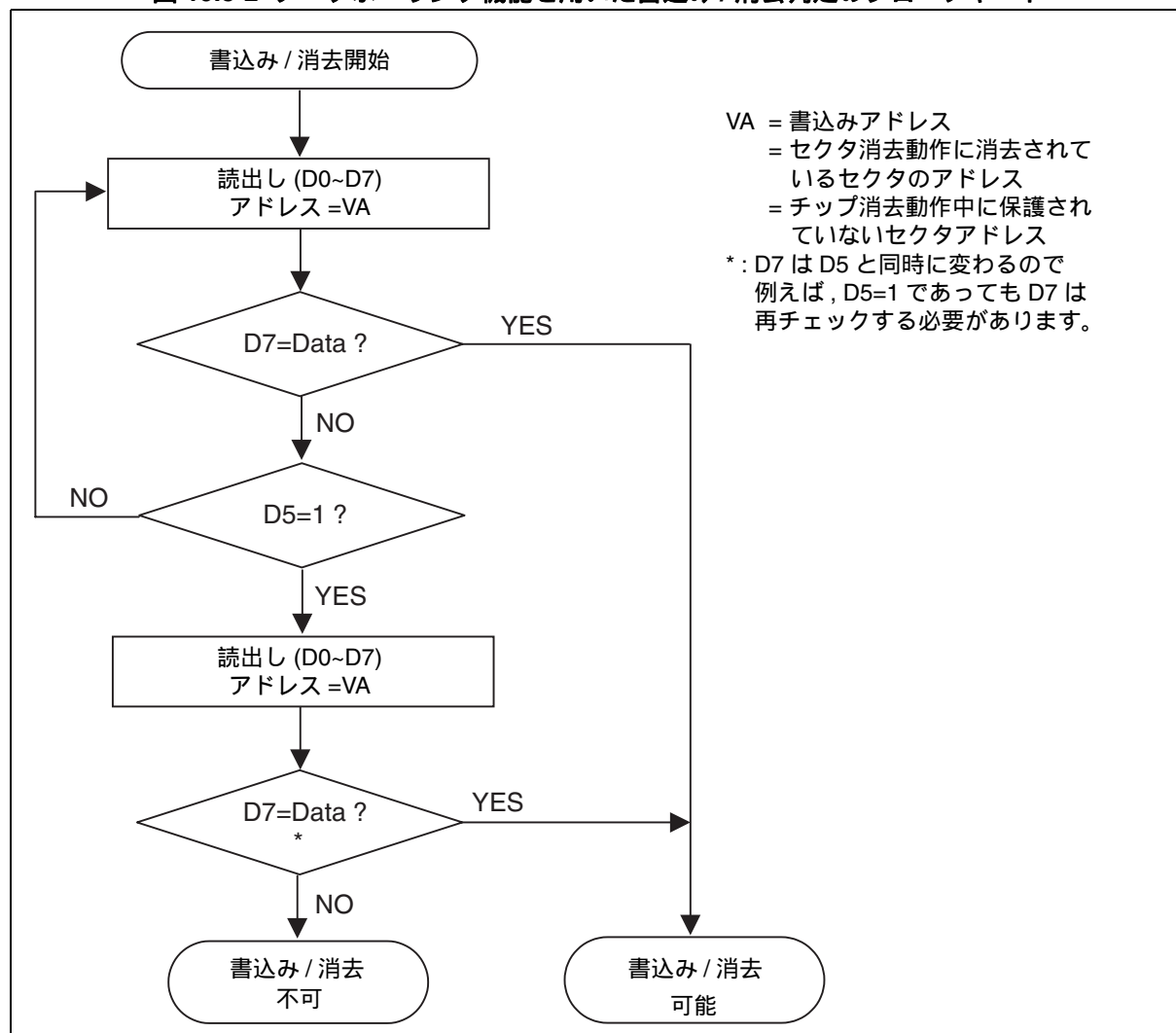
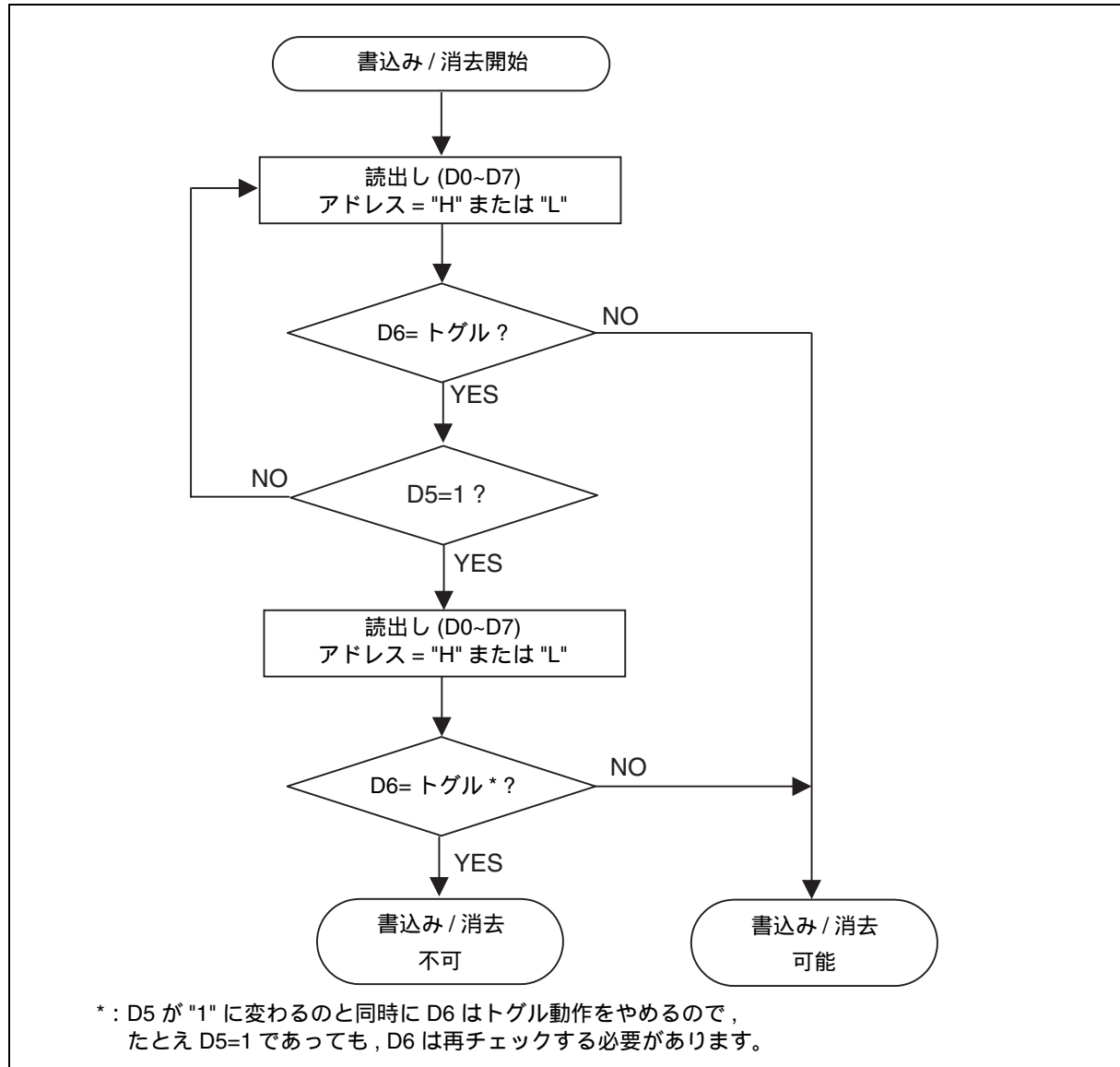


図 19.5-3 トグルビット機能を用いた書き込み / 消去判定のフローチャート



19.6 デュアルオペレーション動作

MB91F267A/MB91F267NA のフラッシュメモリはデュアルオペレーションに対応しています。

上位バンク (32K バイト × 2 + 8K バイト × 4) と下位バンク (8K バイト × 4) において、従来のフラッシュ品では行えなかったバンクごとの消去 / 書込みと読出しとの同時実行が可能です。

■ デュアルオペレーションフラッシュの特長

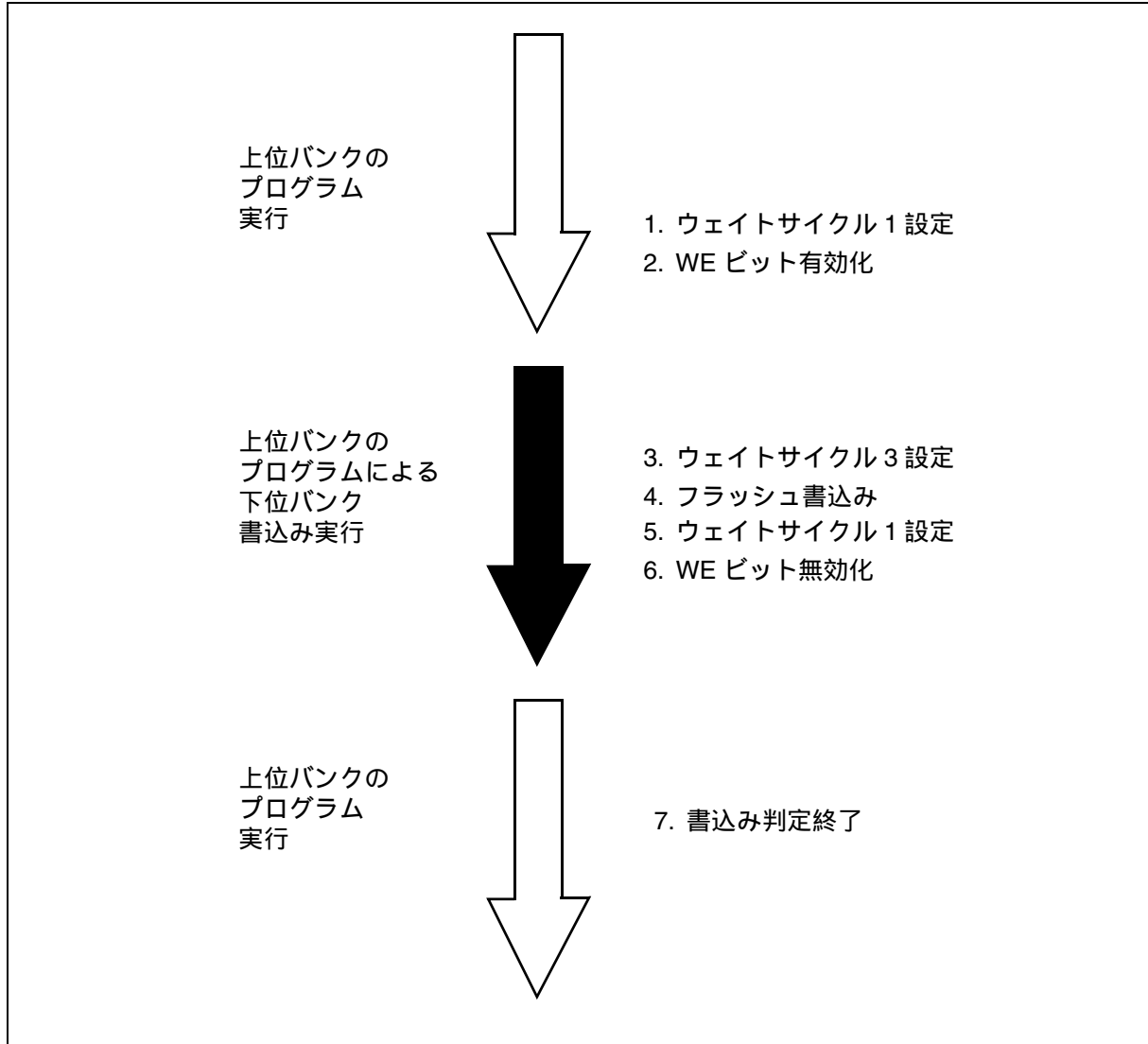
- 2 バンク構成による消去 / 書込みと読出しとの同時実行が可能です。

	FLASH0	FLASH1
000F_E000 _H ~ 000F_FFFF _H	4 K バイト	4 K バイト
000F_C000 _H ~ 000F_DFFF _H	4 K バイト	4 K バイト
000F_A000 _H ~ 000F_BFFF _H	4 K バイト	4 K バイト
000F_8000 _H ~ 000E_9FFF _H	4 K バイト	4 K バイト
000F_6000 _H ~ 000F_7FFF _H	4 K バイト	4 K バイト
000F_4000 _H ~ 000F_5FFF _H	4 K バイト	4 K バイト
000F_2000 _H ~ 000F_3FFF _H	4 K バイト	4 K バイト
000F_0000 _H ~ 000F_1FFF _H	4 K バイト	4 K バイト
000E_8000 _H ~ 000E_FFFF _H	16 K バイト	16 K バイト
000E_0000 _H ~ 000E_7FFF _H	16 K バイト	16 K バイト
	bit 31	16 15 0

- 次の 4 種類の組合せが可能です。

	上位バンク	下位バンク
1	読出し	読出し
2	読出し	書込み / セクタ消去
3	書込み / セクタ消去	読出し
4	チップ消去	チップ消去

図 19.6-1 書き込みシーケンス (例)



第20章

シリアル書込み接続

シリアル書込み基本構成，シリアルオンボード書込みに使用する端子，シリアル書込み接続例およびフラッシュマイコンプログラマシステム構成について説明します。

20.1 概要

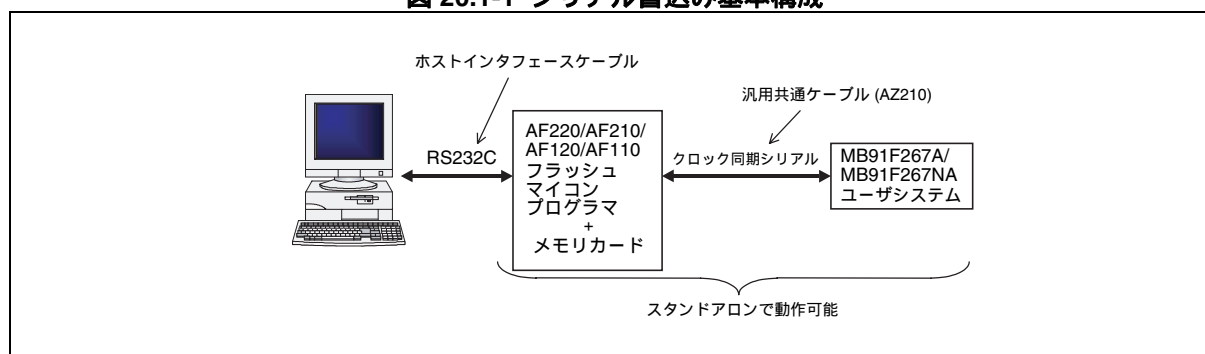
20.1 概要

MB91F267A/MB91F267NA ではフラッシュメモリのシリアルオンボード書込み (富士通標準) をサポートしています。その仕様について説明します。

■ シリアル書込み基本構成

富士通標準シリアルオンボード書込みには、横河デジタルコンピュータ製 AF220/AF210/AD120/AF110 フラッシュマイコンプログラマを使用します。シングルチップモードで動作するプログラムで書き込むことができます。図 20.1-1 に MB91F267A/MB91F267NA シリアル書込み接続の基本構成について示します。

図 20.1-1 シリアル書込み基本構成



< 注意事項 >

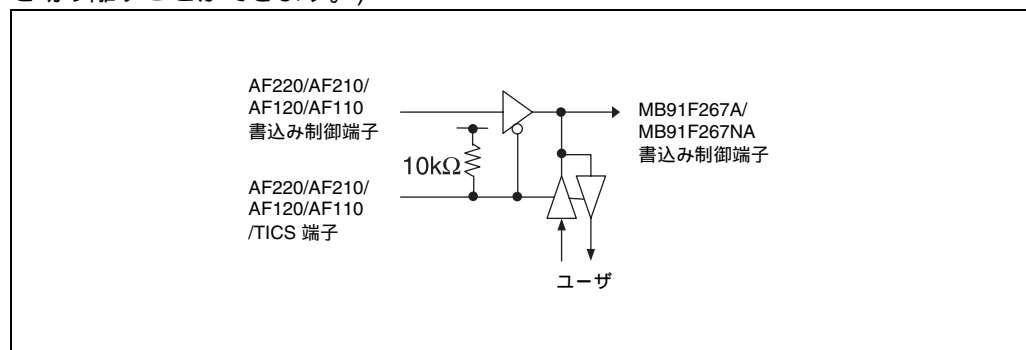
AF210 フラッシュマイコンプログラマの機能・操作方法および接続用汎用共通ケーブル (AZ210)、コネクタにつきましては、横河デジタルコンピュータ株式会社にお問い合わせください。

■ 富士通標準シリアルオンボード書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	書き込みモードに制御します。 フラッシュシリアル書き込みモード：MD2, MD1, MD0=1, 0, 0 参考：シングルチップモード：MD2, MD1, MD0=0, 0, 0
P30, P31	書き込みプログラム起動端子	P30 に "L" レベル, P31 に "H" レベルを入力してください。
$\overline{\text{INIT}}$	リセット端子	-
SIN0	シリアルデータ入力端子	UART の ch.0 リソースをクロック同期モードとして使用します。
SOT0	シリアルデータ出力端子	
SCK0	シリアルクロック入力端子	
Vcc	電源電圧供給端子	書き込み電圧をユーザシステムから供給してください。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ムの GND と共通にします。

< 注意事項 >

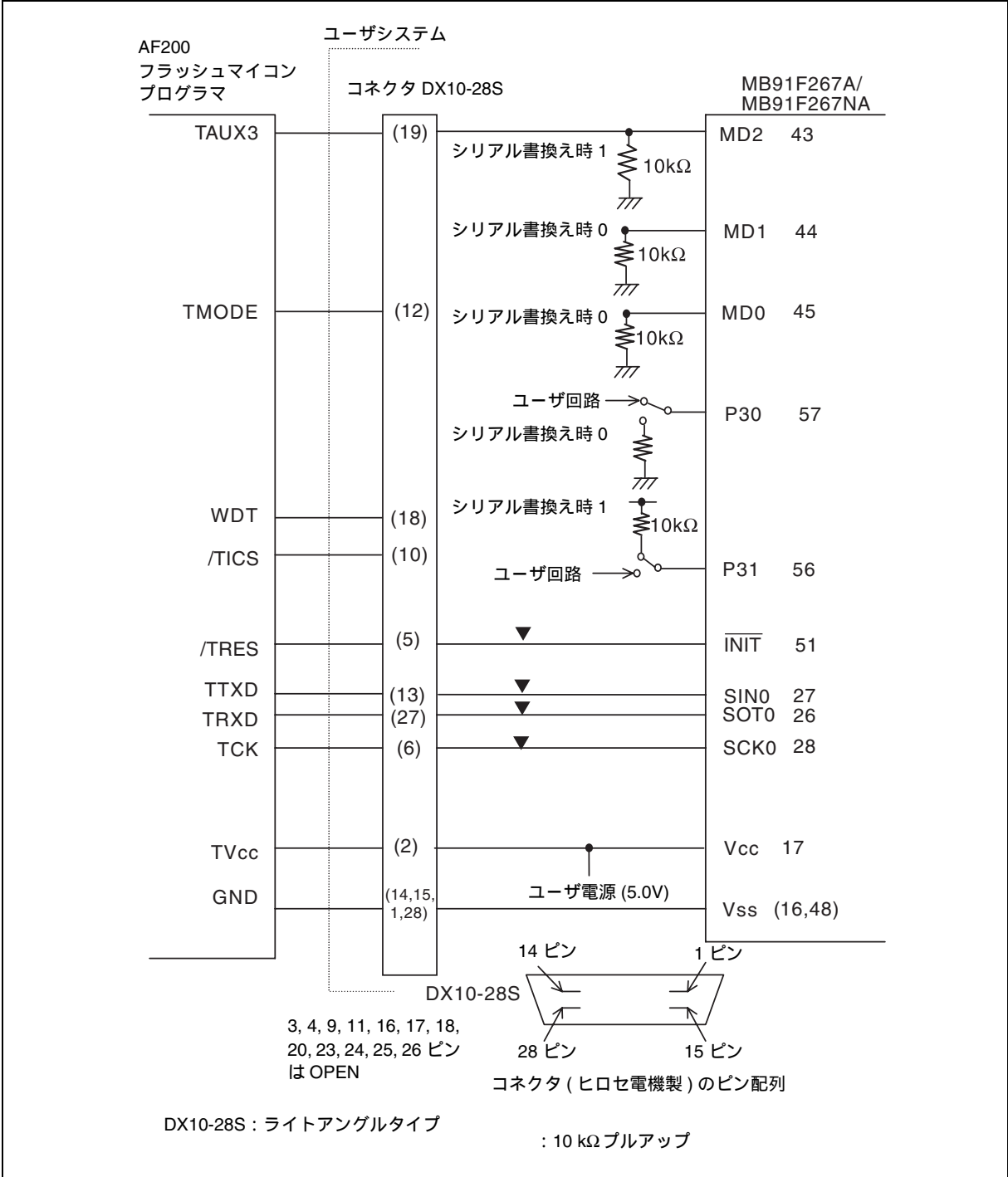
- P30, P31, SIN0, SOT0, SCK0 端子をユーザシステムでも使用する場合には、下図の制御回路が必要となります。
(フラッシュマイコンプログラムの /TICS 信号により、シリアル書き込み中はユーザ回路を切り離すことができます。)



- AF200 との接続はユーザ電源が OFF の状態で行ってください。

■ シリアル書き込み接続例

図 20.1-2 MB91F267A/MB91F267NA シリアル書き込み接続例



■ AF200 フラッシュマイコンプログラマ システム構成 (横河デジタルコンピュータ製)

型格		機能
本体	AF220 /AC4P	イーサネットインタフェースモデル /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 : 1 m
FF201		富士通製 FR フラッシュマイコン用コントロールモジュール
AZ290		リモートコントローラ
/P4		4 M バイト PC Card (Option) フラッシュメモリ容量 512 K バイトまで

問い合わせ先 : 横河デジタルコンピュータ株式会社

電話 : 042-333-6224

■ 原発振クロック周波数について

フラッシュメモリ書込み時に使用可能な原発振クロックは 4.0 MHz となっています。

■ その他の注意事項

シリアルライターを用いた場合のフラッシュメモリ書込み時のポート状態は、書込みに使用している端子を除き、リセット状態と同じです。

各 CPU ステートにおける端子状態, リトル・エンディアン領域を利用する際の注意事項, FR ファミリの命令一覧および MB91265A シリーズを使用する際の注意事項について説明します。

付録 A I/O マップ

付録 B 割込みベクタ

付録 C 各 CPU ステートにおける端子状態

付録 D リトル・エンディアン領域を利用する際の注意事項

付録 E 命令一覧表

付録 F 使用上の注意

付録 A I/O マップ

メモリ空間領域と周辺リソースの各レジスタの対応を示します。

■ I/O マップ

図 A-1 表の見方

アドレス	レジスタ				ブロック
	+0	+1	+2	+3	
000000 _H	PDR0 [R/W]B XXXXXXXX	PDR1 [R/W]B XXXXXXXX	PDR2 [R/W]B XXXXXXXX	PDR3 [R/W]B XXXXXXXX	T-unit ポートデータレジスタ

リード/ライト 属性, アクセス単位
 (B: バイト, H: ハーフワード, W: ワード)

リセット後のレジスタ初期値

レジスタ名 (1 コラムのレジスタが 4n 番地, 2 コラムが 4n + 1 番地...)

最左のレジスタ番地 (ワードでアクセスした際は, 1 コラム目のレジスタがデータの MSB 側となる。)

< 注意事項 >

レジスタのビット値は, 以下のように初期値を表します。

"1": 初期値 "1"

"0": 初期値 "0"

"X": 初期値不定

"-": その位置に物理的にレジスタが存在しない

記述されていないデータアクセス属性によるアクセスは禁止です。

付表 A-1 I/O マップ (1 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
000000 _H	PDR0 [R/W] B,H,W XXXXXXXXXX	PDR1 [R/W] B,H,W XXXXXXXXXX	PDR2 [R/W] B,H,W XXXXXXXXXX	PDR3 [R/W] B,H,W XXXXXXXXXX	ポートデータ レジスタ
000004 _H	PDR4 [R/W] B,H,W XXXXXXXXXX	PDR5 [R/W] B,H,W XXXXXXXXXX	——	——	
000008 _H	——	——	——	——	
00000C _H	——	——	——	——	
000010 _H	PDRG [R/W] B,H,W -----X-	——	——	——	
000014 _H ~ 00003C _H	——				予約
000040 _H	EIRR0 [R/W] B,H,W 00000000	ENIR0 [R/W] B,H,W 00000000	ELVR0 [R/W] B,H,W 00000000 00000000		外部割込み (INT0 ~ INT7)
000044 _H	DICR [R/W] B,H,W -----0	HRCL [R/W,R] B,H,W 0--11111	——	——	遅延割込み / ホールド リクエスト
000048 _H	TMRLR0 [W] H,W XXXXXXXXXX XXXXXXXXXX		TMR0 [R] H,W XXXXXXXXXX XXXXXXXXXX		リロードタイマ 0
00004C _H	——	——	TMCSR0 [R/W,R] B,H,W ---00000 00000000		
000050 _H	TMRLR1 [W] H,W XXXXXXXXXX XXXXXXXXXX		TMR1 [R] H,W XXXXXXXXXX XXXXXXXXXX		リロードタイマ 1
000054 _H	——	——	TMCSR1 [R/W,R] B,H,W ---00000 00000000		
000058 _H	TMRLR2 [W] H,W XXXXXXXXXX XXXXXXXXXX		TMR2 [R] H,W XXXXXXXXXX XXXXXXXXXX		リロードタイマ 2
00005C _H	——		TMCSR2 [R/W,R] B,H,W ---00000 00000000		
000060 _H	SSR0 [R/W,R] B,H,W 00001000	SIDR0[R]/SODR0[W] B,H,W XXXXXXXXXX	SCR0 [R/W,W] B,H,W 00000100	SMR0 [R/W,W] B,H,W 00--0-0-	UART0
000064 _H	UTIM0 [R] H / UTIMR0 [W] H 00000000 00000000		DRCL0 -----*1	UTIMC0 [R/W] B 0--00001	U-Timer0
000068 _H	SSR1 [R/W,R] B,H,W 00001000	SIDR1[R]/SODR1[W] XXXXXXXXXX	SCR1 [R/W,W] B,H,W 00000100	SMR1 [R/W] B,H,W 00--0-0-	UART1
00006C _H	UTIM1 [R] H / UTIMR1 [W] H 00000000 00000000		DRCL1 -----*1	UTIMC1 [R/W] B 0--00001	U-Timer1
000070 _H	——	——	——	——	予約
000074 _H	——	——	——	——	
000078 _H	——	——	——	——	
00007C _H	——	——	——	——	
000080 _H	ADCH1 [R/W] B,H,W XX000000	ADMD1 [R/W] B,H,W 00001111	ADCD11 [R] B,H,W XXXXXXXXXX	ADCD10 [R] B,H,W XXXXXXXXXX	A/D コンバータ 1 アナログ入力 制御 1
000084 _H	ADCS1[R/W,W] B,H,W 00000X00	——	AICR1 [R/W] B,H,W ---0000	——	
000088 _H	ADCH2 [R/W] B,H,W XX000000	ADMD2 [R/W] B,H,W 00001111	ADCD21 [R] B,H,W XXXXXXXXXX	ADCD20 [R] B,H,W XXXXXXXXXX	A/D コンバータ 2 アナログ入力 制御 2
00008C _H	ADCS2[R/W,W] B,H,W 00000X00	——	AICR2 [R/W] B,H,W -0000000	——	

付表 A-1 I/O マップ (2 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
000090 _H	OCCPBH0, OCCPBL0[W]/ OCCPH0, OCCPL0 [R] H,W 00000000 00000000		OCCPBH1, OCCPBL1[W]/ OCCPH1, OCCPL1 [R] H,W 00000000 00000000		16 ビット OCU
000094 _H	OCCPBH2, OCCPBL2[W]/ OCCPH2, OCCPL2 [R] H,W 00000000 00000000		OCCPBH3, OCCPBL3[W]/ OCCPH3, OCCPL3 [R] H,W 00000000 00000000		
000098 _H	OCCPBH4, OCCPBL4[W]/ OCCPH4, OCCPL4 [R] H,W 00000000 00000000		OCCPBH5, OCCPBL5[W]/ OCCPH5, OCCPL5 [R] H,W 00000000 00000000		
00009C _H	OCSH1 [R/W] B,H,W X1100000	OCSL0 [R/W] B,H,W 00001100	OCSH3 [R/W] B,H,W X1100000	OCSL2 [R/W] B,H,W 00001100	
0000A0 _H	OCSH5 [R/W] B,H,W X1100000	OCSL4 [R/W] B,H,W 00001100	OCMOD [R/W] B,H,W XX000000	——	
0000A4 _H	CPCLRBH0, CPCLRBL0[W]/ CPCLRH0, CPCLRL0[R] H,W 11111111 11111111		TCDTH0, TCDTL0 [R/W] H,W 00000000 00000000		16 ビット フリーラン タイマ 0
0000A8 _H	TCCSH0 [R/W] B,H,W 00000000	TCCSL0 [R/W] B,H,W 01000000	——	ADTRGC [R/W] B,H,W XXXX0000	
0000AC _H	IPCPH0, IPCPL0 [R] H,W XXXXXXXX XXXXXXXX		IPCPH1, IPCPL1 [R] H,W XXXXXXXX XXXXXXXX		16 ビット ICU
0000B0 _H	IPCPH2, IPCPL2 [R] H,W XXXXXXXX XXXXXXXX		IPCPH3, IPCPL3 [R] H,W XXXXXXXX XXXXXXXX		
0000B4 _H	PICSH01 [W] B,H,W 000000--	PICSL01 [R/W] B,H,W 00000000	ICSH23 [R] B,H,W XXXXXX00	ICSL23 [R/W] B,H,W 00000000	
0000B8 _H	——	——	——	——	
0000BC _H	TMRRH0, TMRRL0 [R/W] H,W XXXXXXXX XXXXXXXX		TMRRH1, TMRRL1 [R/W] H,W XXXXXXXX XXXXXXXX		波形ジェネレータ
0000C0 _H	TMRRH2, TMRRL2 [R/W] H,W XXXXXXXX XXXXXXXX		——	——	
0000C4 _H	DTCCR0 [R/W] B,H,W 00000000	DTCCR1 [R/W] B,H,W 00000000	DTCCR2 [R/W] B,H,W 00000000	——	
0000C8 _H	——	SIGCR1 [R/W] B,H,W 00000000	——	SIGCR2 [R/W] B,H,W XXXXXXX1	
0000CC _H	——	——	ADCOMP1 [R/W] H,W 00000000 00000000		
0000D0 _H	ADCOMP2 [R/W] H,W 00000000 00000000		ADCOMP2 [R/W] B,H,W XX0000XX	ADCOMP1 [R/W] B,H,W XXXXXX00X	A/D COMP
0000D4 _H	——	——	——	——	予約
0000D8 _H	——	——	——	——	
0000DC _H	——	——	——	——	
0000E0 _H	PWCSR0 [R/W,R] B,H,W 00000000 00000000		PWCR0 [R] H,W 00000000 00000000		16 ビット PWC タイマ
0000E4 _H	——	——	——	——	
0000E8 _H	——	PDIVR0 [R/W] B,H,W XXXXXX000	——	——	
0000EC _H	——	——	——	——	予約
0000F0 _H	——	——	——	——	
0000F4 _H ~ 0000EC _H	——	——	——	——	

付表 A-1 I/O マップ (3 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
000100 _H	PRLH0 [R/W] B,H,W XXXXXXXXXX	PRLL0 [R/W] B,H,W XXXXXXXXXX	PRLH1 [R/W] B,H,W XXXXXXXXXX	PRLL1 [R/W] B,H,W XXXXXXXXXX	8/16 ビット PPG タイマ 0 ~ 7
000104 _H	PRLH2 [R/W] B,H,W XXXXXXXXXX	PRLL2 [R/W] B,H,W XXXXXXXXXX	PRLH3 [R/W] B,H,W XXXXXXXXXX	PRLL3 [R/W] B,H,W XXXXXXXXXX	
000108 _H	PPGC0 [R/W] B,H,W 0000000X	PPGC1 [R/W] B,H,W 0000000X	PPGC2 [R/W] B,H,W 0000000X	PPGC3 [R/W] B,H,W 0000000X	
00010C _H	PRLH4 [R/W] B,H,W XXXXXXXXXX	PRLL4 [R/W] B,H,W XXXXXXXXXX	PRLH5 [R/W] B,H,W XXXXXXXXXX	PRLL5 [R/W] B,H,W XXXXXXXXXX	
000110 _H	PRLH6 [R/W] B,H,W XXXXXXXXXX	PRLL6 [R/W] B,H,W XXXXXXXXXX	PRLH7 [R/W] B,H,W XXXXXXXXXX	PRLL7 [R/W] B,H,W XXXXXXXXXX	
000114 _H	PPGC4 [R/W] B,H,W 0000000X	PPGC5 [R/W] B,H,W 0000000X	PPGC6 [R/W] B,H,W 0000000X	PPGC7 [R/W] B,H,W 0000000X	
000118 _H ~ 00012C _H	——	——	——	——	予約
000130 _H	TRG [R/W] B,H,W 00000000 00000000		——	GATEC [R/W] B,H,W XXXXXXXX00	8/16 ビット PPG タイマ 0 ~ 7
000134 _H	REVC [R/W] B,H,W 00000000 00000000		——	——	
000138 _H	——	——	——	——	予約
00013C _H	——	——	——	——	
000140 _H	——	——	——	——	予約
000144 _H	TTCR [R/W] B,H,W 11110000	——	——	TSTPR0 [R] B,H,W 00000000	タイミング ジェネレータ
000148 _H	COMP0 [R/W] B,H,W 00000000	COMP2 [R/W] B,H,W 00000000	COMP4 [R/W] B,H,W 00000000	COMP6 [R/W] B,H,W 00000000	
00014C _H	——	——	——	——	
000150 _H	——	——	——	——	
000154 _H	CPCLRBH1, CPCLRBL1[W]/ CPCLRH1, CPCLRL1[R] H,W 11111111 11111111		TCDTH1, TCDTL1 [R/W] H,W 00000000 00000000		16 ビット フリーラン タイマ 1
000158 _H	TCCSH1 [R/W] B,H,W 00000000	TCCSL1 [R/W] B,H,W 01000000	——	——	
00015C _H	CPCLRBH2, CPCLRBL2[W]/ CPCLRH2, CPCLRL2[R] H,W 11111111 11111111		TCDTH2, TCDTL2 [R/W] H,W 00000000 00000000		16 ビット フリーラン タイマ 2
000160 _H	TCCSH2 [R/W] B,H,W 00000000	TCCSL2 [R/W] B,H,W 01000000	——	——	
000164 _H	——	——	——	——	予約
000168 _H	——	FSR2 [R/W] B,H,W 00000000	FSR1 [R/W] B,H,W XXXXX0000	FSR0 [R/W] B,H,W 00000000	FRT セレクタ
00016C _H ~ 0001A4 _H	——				予約
0001A8 _H	CANPRE [R,R/W] B,H,W 00000000	——	——	——	C-CAN ^{*2} プリスケラ
0001AC _H ~ 0001FC _H	——				予約

付表 A-1 I/O マップ (4 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
000200 _H	DMACA0 [R/W] B,H,W ^{*3} 00000000 00000000 00000000 00000000				DMAC
000204 _H	DMACB0 [R/W] B,H,W 00000000 00000000 00000000 00000000				
000208 _H	DMACA1 [R/W] B,H,W ^{*3} 00000000 00000000 00000000 00000000				
00020C _H	DMACB1 [R/W] B,H,W 00000000 00000000 00000000 00000000				
000210 _H	DMACA2 [R/W] B,H,W ^{*3} 00000000 00000000 00000000 00000000				
000214 _H	DMACB2 [R/W] B,H,W 00000000 00000000 00000000 00000000				
000218 _H	DMACA3 [R/W] B,H,W ^{*3} 00000000 00000000 00000000 00000000				
00021C _H	DMACB3 [R/W] B,H,W 00000000 00000000 00000000 00000000				
000220 _H	DMACA4 [R/W] B,H,W ^{*3} 00000000 00000000 00000000 00000000				
000224 _H	DMACB4 [R/W] B,H,W 00000000 00000000 00000000 00000000				
000228 _H ~ 00023C _H	_____				予約
000240 _H	DMACR [R/W] B 0XX00000 XXXXXXXX XXXXXXXX XXXXXXXX				DMAC
000244 _H ~ 00024C _H	_____				予約
000250 _H	_____	_____	_____	_____	予約
000254 _H ~ 000398 _H	_____				予約
00039C _H	_____	_____	_____	_____	積和演算回路
0003A0 _H	DSP-PC [R/W] XXXXXXXXXX	DSP-CSR [R/W,R,W] 00000000	DSP-LY [R/W] XXXXXXXXXX XXXXXXXX		
0003A4 _H	DSP-OT0 [R] XXXXXXXXXX XXXXXXXX		DSP-OT1 [R] XXXXXXXXXX XXXXXXXX		
0003A8 _H	DSP-OT2 [R] XXXXXXXXXX XXXXXXXX		DSP-OT3 [R] XXXXXXXXXX XXXXXXXX		
0003AC _H	_____	_____	_____	_____	
0003B0 _H	DSP-OT4 [R] XXXXXXXXXX XXXXXXXX		DSP-OT5 [R] XXXXXXXXXX XXXXXXXX		
0003B4 _H	DSP-OT6 [R] XXXXXXXXXX XXXXXXXX		DSP-OT7 [R] XXXXXXXXXX XXXXXXXX		
0003B8 _H	_____	_____	_____	_____	
0003BC _H ~ 0003EC _H	_____				予約

付表 A-1 I/O マップ (5 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
0003F0 _H	BSD0 [W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				ビットサーチ モジュール
0003F4 _H	BSD1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003F8 _H	BSDC [W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0003FC _H	BSRR [R] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000400 _H	DDR0 [R/W] B,H,W 00000000	DDR1 [R/W] B,H,W 00000000	DDR2 [R/W] B,H,W 00000000	DDR3 [R/W] B,H,W 00000000	データ方向 レジスタ
000404 _H	DDR4 [R/W] B,H,W -0000000	DDR5 [R/W] B,H,W 00000000	_____	_____	
000408 _H	_____	_____	_____	_____	
00040C _H	_____	_____	_____	_____	
000410 _H	DDRG [R/W] B,H,W -----0-	_____	_____	_____	
000414 _H ~ 00041C _H	_____				予約
000420 _H	PFR0 [R/W] B,H,W 00-----	PFR1 [R/W] B,H,W --0-00-0	_____	_____	ポート機能 レジスタ
000424 _H	_____	_____	_____	_____	
000428 _H	_____	_____	_____	_____	
00042C _H	_____	_____	_____	_____	
000430 _H	_____	_____	_____	PTFR0 [R/W] B,H,W 00000000	
000434 _H ~ 00043C _H	_____				予約
000440 _H	ICR00 [R/W,R] B,H,W ---11111	ICR01 [R/W,R] B,H,W ---11111	ICR02 [R/W,R] B,H,W ---11111	ICR03 [R/W,R] B,H,W ---11111	割込み制御 ユニット
000444 _H	ICR04 [R/W,R] B,H,W ---11111	ICR05 [R/W,R] B,H,W ---11111	ICR06 [R/W,R] B,H,W ---11111	ICR07 [R/W,R] B,H,W ---11111	
000448 _H	ICR08 [R/W,R] B,H,W ---11111	ICR09 [R/W,R] B,H,W ---11111	ICR10 [R/W,R] B,H,W ---11111	ICR11 [R/W,R] B,H,W ---11111	
00044C _H	ICR12 [R/W,R] B,H,W ---11111	ICR13 [R/W,R] B,H,W ---11111	ICR14 [R/W,R] B,H,W ---11111	ICR15 [R/W,R] B,H,W ---11111	
000450 _H	ICR16 [R/W,R] B,H,W ---11111	ICR17 [R/W,R] B,H,W ---11111	ICR18 [R/W,R] B,H,W ---11111	ICR19 [R/W,R] B,H,W ---11111	
000454 _H	ICR20 [R/W,R] B,H,W ---11111	ICR21 [R/W,R] B,H,W ---11111	ICR22 [R/W,R] B,H,W ---11111	ICR23 [R/W,R] B,H,W ---11111	
000458 _H	ICR24 [R/W,R] B,H,W ---11111	ICR25 [R/W,R] B,H,W ---11111	ICR26 [R/W,R] B,H,W ---11111	ICR27 [R/W,R] B,H,W ---11111	
00045C _H	ICR28 [R/W,R] B,H,W ---11111	ICR29 [R/W,R] B,H,W ---11111	ICR30 [R/W,R] B,H,W ---11111	ICR31 [R/W,R] B,H,W ---11111	
000460 _H	ICR32 [R/W,R] B,H,W ---11111	ICR33 [R/W,R] B,H,W ---11111	ICR34 [R/W,R] B,H,W ---11111	ICR35 [R/W,R] B,H,W ---11111	
000464 _H	ICR36 [R/W,R] B,H,W ---11111	ICR37 [R/W,R] B,H,W ---11111	ICR38 [R/W,R] B,H,W ---11111	ICR39 [R/W,R] B,H,W ---11111	
000468 _H	ICR40 [R/W,R] B,H,W ---11111	ICR41 [R/W,R] B,H,W ---11111	ICR42 [R/W,R] B,H,W ---11111	ICR43 [R/W,R] B,H,W ---11111	
00046C _H	ICR44 [R/W,R] B,H,W ---11111	ICR45 [R/W,R] B,H,W ---11111	ICR46 [R/W,R] B,H,W ---11111	ICR47 [R/W,R] B,H,W ---11111	

付表 A-1 I/O マップ (6 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
000470 _H ~ 00047C _H	_____				予約
000480 _H	RSRR [R,R/W] B,H,W 10000000	STCR [R/W] B,H,W 00110011	TBCR [R/W] B,H,W 00XXXX00	CTBR [W] B,H,W XXXXXXXXXX	クロック制御 ユニット
000484 _H	CLKR [R/W] B,H,W 00000000	WPR _____ *1	DIVR0 [R/W] B,H,W 00000011	DIVR1 [R/W] B,H,W 00000000	
000488 _H	_____	_____	_____	_____	
00048C _H	_____	_____	_____	_____	
000490 _H	_____	_____	_____	_____	
000494 _H ~ 0005FC _H	_____				予約
000600 _H	PCR0 [R/W] B,H,W 00000000	PCR1 [R/W] B,H,W 00000000	PCR2 [R/W] B,H,W 00000000	PCR3 [R/W] B,H,W 00-----	ブルアップ制御
000604 _H	PCR4 [R/W] B,H,W ----0000	_____	_____	_____	
000608 _H	_____	_____	_____	_____	
00060C _H	_____	_____	_____	_____	
000610 _H	PCRG [R/W] B,H,W -----0-	_____	_____	_____	
000614 _H ~ 00063C _H	_____				予約
001000 _H	DMASA0 [R/W] W 00000000 00000000 00000000 00000000				DMAC
001004 _H	DMADA0 [R/W] W 00000000 00000000 00000000 00000000				
001008 _H	DMASA1 [R/W] W 00000000 00000000 00000000 00000000				
00100C _H	DMADA1 [R/W] W 00000000 00000000 00000000 00000000				
001010 _H	DMASA2 [R/W] W 00000000 00000000 00000000 00000000				
001014 _H	DMADA2 [R/W] W 00000000 00000000 00000000 00000000				
001018 _H	DMASA3 [R/W] W 00000000 00000000 00000000 00000000				
00101C _H	DMADA3 [R/W] W 00000000 00000000 00000000 00000000				
001020 _H	DMASA4 [R/W] W 00000000 00000000 00000000 00000000				
001024 _H	DMADA4 [R/W] W 00000000 00000000 00000000 00000000				
001028 _H ~ 006FFC _H	_____				予約

付表 A-1 I/O マップ (7 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
007000 _H	FLCR [R/W,R] B 01101000	——	——	——	FLASH
007004 _H	FLWC [R/W,R] B 00000011	——	——	——	
007008 _H	——	——	——	——	
00700C _H	——	——	——	——	
007010 _H	——	——	——	——	
007014 _H ~ 00BFFC _H	——				予約
00C000 _H ~ 00C07C _H	X-RAM (係数 RAM) [R/W] 64 × 16 ビット				積和演算回路
00C080 _H ~ 00C0FC _H	Y-RAM (変数 RAM) [R/W] 64 × 16 ビット				
00C100 _H ~ 00C2FC _H	I-RAM (命令 RAM) [R/W] 256 × 16 ビット				
00C300 _H ~ 00FFFC _H	——				予約
020000 _H	CTRLR0 [R,R/W] 00000000 00000001		STATR0 [R,R/W] 00000000 00000000		C-CAN ^{*2}
020004 _H	ERRCNT0 [R] 00000000 00000000		BTR0 [R,R/W] 00100011 00000001		
020008 _H	INTR0 [R] 00000000 00000000		TESTR0 [R,R/W] 00000000 X0000000		
02000C _H	BRPER0 [R,R/W] 00000000 00000000		——	——	
020010 _H	IF1CREQ0 [R,R/W] 00000000 00000000		IF1CMSK0 [R,R/W] 00000000 00000000		
020014 _H	IF1MSK20 [R,R/W] 11111111 11111111		IF1MSK10 [R/W] 11111111 11111111		
020018 _H	IF1ARB20 [R/W] 00000000 00000000		IF1ARB10 [R/W] 00000000 00000000		
02001C _H	IF1MCTR0 [R,R/W] 00000000 00000000		——	——	
020020 _H	IF1DTA10 [R/W] 00000000 00000000		IF1DTA20 [R/W] 00000000 00000000		
020024 _H	IF1DTB10 [R/W] 00000000 00000000		IF1DTB20 [R/W] 00000000 00000000		
020030 _H	予約 (IF1 data mirror, little endian byte ordering)				
020040 _H	IF2CREQ0 [R,R/W] 00000000 00000000		IF2CMSK0 [R,R/W] 00000000 00000000		
020044 _H	IF2MSK20 [R,R/W] 11111111 11111111		IF2MSK10 [R/W] 11111111 11111111		
020048 _H	IF2ARB20 [R/W] 00000000 00000000		IF2ARB10 [R/W] 00000000 00000000		
02004C _H	IF2MCTR0 [R,R/W] 00000000 00000000		——	——	

付表 A-1 I/O マップ (8 / 8)

アドレス	レジスタ				ブロック
	+ 0	+ 1	+ 2	+ 3	
020050 _H	IF2DTA10 [R/W] 00000000 00000000		IF2DTA20 [R/W] 00000000 00000000		C-CAN ^{*2}
020054 _H	IF2DTB10 [R/W] 00000000 00000000		IF2DTB20 [R/W] 00000000 00000000		
020060 _H	予約 (IF2 data mirror, little endian byte ordering)				
020080 _H	TREQR20 [R] 00000000 00000000		TREQR10 [R] 00000000 00000000		
020090 _H	NEWDT20 [R] 00000000 00000000		NEWDT10 [R] 00000000 00000000		
0200A0 _H	INTPND20 [R] 00000000 00000000		INTPND10 [R] 00000000 00000000		
0200B0 _H	MESVAL20 [R] 00000000 00000000		MESVAL10 [R] 00000000 00000000		

*1 : 予約レジスタです。アクセス禁止です。

*2 : C-CAN は , MB91F267NA/MB91267NA に搭載されています。

*3 : DMACA0 ~ DMACA4 の下位 16 ビット (DTC15 ~ DTC0) は , バイトでのアクセスはできません。
 (注意事項) ・ FLWC(7004_H) の初期値は , 評価用品ツール上は "00010011_B" です。評価用品にて ,
 "00000011_B" を書き込みしても動作に変わりはありません。

- ・ 書き込みのみのビットのあるレジスタに対してリードモディファイライト (RMW) 系命令を行わないでください。
- ・ 予約または (-) の領域のデータは不定です。

付録 B 割込みベクタ

MB91265A シリーズのベクタテーブルを示します。

■ 割込みベクタ

付表 B-1 ベクタテーブル (1 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
リセット	0	00	-	3FC _H	000FFFC _H
モードベクタ	1	01	-	3F8 _H	000FFF8 _H
システム予約	2	02	-	3F4 _H	000FFF4 _H
システム予約	3	03	-	3F0 _H	000FFF0 _H
システム予約	4	04	-	3EC _H	000FFFE _C
システム予約	5	05	-	3E8 _H	000FFFE8 _H
システム予約	6	06	-	3E4 _H	000FFFE4 _H
コプロセッサ不在トラップ	7	07	-	3E0 _H	000FFFE0 _H
コプロセッサエラートラップ	8	08	-	3DC _H	000FFFD _C
INTE 命令	9	09	-	3D8 _H	000FFFD8 _H
システム予約	10	0A	-	3D4 _H	000FFFD4 _H
システム予約	11	0B	-	3D0 _H	000FFFD0 _H
ステップトレーストラップ	12	0C	-	3CC _H	000FFFC _C
NMI 要求 (tool)	13	0D	-	3C8 _H	000FFFC8 _H
未定義命令例外	14	0E	-	3C4 _H	000FFFC4 _H
NMI 要求	15	0F	15(F _H) 固定	3C0 _H	000FFFC0 _H
外部割込み 0	16	10	ICR00	3BC _H	000FFFB _C
外部割込み 1	17	11	ICR01	3B8 _H	000FFFB8 _H
外部割込み 2	18	12	ICR02	3B4 _H	000FFFB4 _H
外部割込み 3	19	13	ICR03	3B0 _H	000FFFB0 _H
外部割込み 4	20	14	ICR04	3AC _H	000FFFA _C
外部割込み 5	21	15	ICR05	3A8 _H	000FFFA8 _H
外部割込み 6/C-CAN ウェイクアップ*	22	16	ICR06	3A4 _H	000FFFA4 _H
外部割込み 7	23	17	ICR07	3A0 _H	000FFFA0 _H
リロードタイマ 0	24	18	ICR08	39C _H	000FFF9 _C
リロードタイマ 1	25	19	ICR09	398 _H	000FFF98 _H
リロードタイマ 2	26	1A	ICR10	394 _H	000FFF94 _H
UART0 (受信完了)	27	1B	ICR11	390 _H	000FFF90 _H
UART0 (送信完了)	28	1C	ICR12	38C _H	000FFF8 _C
DTTI 端子	29	1D	ICR13	388 _H	000FFF88 _H
DMAC0 (終了, エラー)	30	1E	ICR14	384 _H	000FFF84 _H
DMAC1 (終了, エラー)	31	1F	ICR15	380 _H	000FFF80 _H
DMAC2/DMAC3/DMAC4 (終了, エラー)	32	20	ICR16	37C _H	000FFF7 _C

付表 B-1 ベクタテーブル (2 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
UART1 (受信完了)	33	21	ICR17	378 _H	000FFF78 _H
UART1 (送信完了)	34	22	ICR18	374 _H	000FFF74 _H
C-CAN 0*	35	23	ICR19	370 _H	000FFF70 _H
システム予約	36	24	ICR20	36C _H	000FFF6C _H
積和	37	25	ICR21	368 _H	000FFF68 _H
PPG0/PPG1	38	26	ICR22	364 _H	000FFF64 _H
PPG2/PPG3	39	27	ICR23	360 _H	000FFF60 _H
PPG4/PPG5/PPG6/PPG7	40	28	ICR24	35C _H	000FFF5C _H
システム予約	41	29	ICR25	358 _H	000FFF58 _H
波形 (アンダフロー) 0/1/2	42	2A	ICR26	354 _H	000FFF54 _H
フリーランタイム 1(コンペアクリア)	43	2B	ICR27	350 _H	000FFF50 _H
フリーランタイム 1(0 検出)	44	2C	ICR28	34C _H	000FFF4C _H
フリーランタイム 2(コンペアクリア)	45	2D	ICR29	348 _H	000FFF48 _H
フリーランタイム 2(0 検出)	46	2E	ICR30	344 _H	000FFF44 _H
タイムベースタイムオーバーフロー	47	2F	ICR31	340 _H	000FFF40 _H
フリーランタイム 0(コンペアクリア)	48	30	ICR32	33C _H	000FFF3C _H
フリーランタイム 0(0 検出)	49	31	ICR33	338 _H	000FFF38 _H
システム予約	50	32	ICR34	334 _H	000FFF34 _H
A/D コンバータ 1	51	33	ICR35	330 _H	000FFF30 _H
A/D コンバータ 2	52	34	ICR36	32C _H	000FFF2C _H
PWC0 (測定完了)	53	35	ICR37	328 _H	000FFF28 _H
システム予約	54	36	ICR38	324 _H	000FFF24 _H
PWC0 (オーバフロー)	55	37	ICR39	320 _H	000FFF20 _H
システム予約	56	38	ICR40	31C _H	000FFF1C _H
ICU 0 (取込み)	57	39	ICR41	318 _H	000FFF18 _H
ICU 1 (取込み)	58	3A	ICR42	314 _H	000FFF14 _H
ICU2/ICU3 (取込み)	59	3B	ICR43	310 _H	000FFF10 _H
OCU0/OCU1 (一致)	60	3C	ICR44	30C _H	000FFF0C _H
OCU2/OCU3 (一致)	61	3D	ICR45	308 _H	000FFF08 _H
OCU4/OCU5 (一致)	62	3E	ICR46	304 _H	000FFF04 _H
遅延割込み要因ビット	63	3F	ICR47	300 _H	000FFF00 _H
システム予約 (REALOS にて使用)	64	40	-	2FC _H	000FFEFC _H
システム予約 (REALOS にて使用)	65	41	-	2F8 _H	000FFE8 _H
システム予約	66	42	-	2F4 _H	000FFE4 _H
システム予約	67	43	-	2F0 _H	000FFE0 _H
システム予約	68	44	-	2EC _H	000FEEC _H
システム予約	69	45	-	2E8 _H	000FEE8 _H
システム予約	70	46	-	2E4 _H	000FEE4 _H
システム予約	71	47	-	2E0 _H	000FEE0 _H
システム予約	72	48	-	2DC _H	000FFEDC _H
システム予約	73	49	-	2D8 _H	000FFED8 _H

付表 B-1 ベクタテーブル (3 / 3)

割込み要因	割込み番号		割込みレベル	オフセット	TBR デフォルト のアドレス
	10 進	16 進			
システム予約	74	4A	-	2D4 _H	000FFED4 _H
システム予約	75	4B	-	2D0 _H	000FFED0 _H
システム予約	76	4C	-	2CC _H	000FFECC _H
システム予約	77	4D	-	2C8 _H	000FFEC8 _H
システム予約	78	4E	-	2C4 _H	000FFEC4 _H
システム予約	79	4F	-	2C0 _H	000FFEC0 _H
INT 命令で使用	80	50	-	2BC _H	000FFEB0 _H
	~	~		~	~
	255	FF		000 _H	000FFC00 _H

* : C-CAN の割込みは , MB91F267NA/MB91267NA に搭載されている機能です。

付録 C 各 CPU ステートにおける端子状態

端子の状態に対する語句は以下の意味を持ちます。

■ 各 CPU ステートにおける端子状態

- 入力可能
入力機能を使用可能な状態であることを意味する。
- 入力 "0" 固定
端子からすぐの入力ゲートで外部入力を遮断して内部へ "0" を伝えている状態。
- 出力 Hi-Z
端子駆動用トランジスタを駆動禁止状態にし、端子をハイインピーダンスにすることを意味する。
- 出力保持
本モードになる直前に出力していた状態をそのまま出力することを意味する。
つまり、出力のある内蔵周辺が動作中であればその内蔵周辺に従い出力を行い、ポートなどとして出力している場合にはその出力を保持する。
- 直前の状態を保持
本モードになる直前に出力していた状態をそのまま出力、あるいは入力であれば入力可能を意味する。

表 C-1 シングルチップモード

Pin No. (LQFP)	端子名	機能	イニシャライズ時		スリープ時	ストップ時	
			INIT=L*1	INIT=H*2		HIZ=0	HIZ=1
3 ~ 10	P50 ~ P57	AN0 ~ AN7	出力 Hi-Z/ 入力不可	出力 Hi-Z/ 入力可能	直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
11 ~ 13	P44 ~ P46	AN8 ~ AN10					
14	NMI	NMI	入力可能	入力可能	入力可能	入力可能	入力可能
18	P00	PPG1/INT4	出力 Hi-Z/ 入力不可	出力 Hi-Z/ 入力可能	直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
19	P01	PPG2			入力可能	入力可能	入力可能
20	P02	PPG3/INT5			直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
21 ~ 23	P03 ~ P05	TIN0 ~ TIN2			入力可能	入力可能	入力可能
24, 25	P06, P07	TOT1, TOT2			直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
26	P10	SOT0					
27	P11	SIN0					
28	P12	SCK0					
29	P13	SOT1					
30	P14	SIN1					
31	P15	SCK1					
32	P16	PPG5/INT6/ RX0*3			入力可能	入力可能	入力可能
33	P17	PPG6/TX0*3			直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
34	P20	ADTG1/IC2					
35	P21	ADTG2/IC3					
36	P22	PWI0					
37	P23	DTTI					
38	P24	CKI					
39	P25	IC0					
40	P26	IC1					
41	P27	ポート					
42	PG1	PPG0			入力可能	入力可能	入力可能
49	P37	PPG4					
50	P36	PPG7/INT7			直前状態保持	直前状態保持	出力 Hi-Z/ 入力 0 固定
52 ~ 57	P35 ~ P30	RTO5 ~ RTO0			入力可能	入力可能	入力可能
58 ~ 61	P40 ~ P43	INT0 ~ INT3					

*1: INIT=L : INIT が "L" の期間の端子状態を示す。

*2: INIT=H : INIT が, "L" から "H" へ遷移した直後の端子状態を示す。

*3: C-CAN の端子は, MB91F267NA/MB91267NA に搭載されています。

付録 D リトル・エンディアン領域を利用する際の注意事項

リトルエンディアン領域を利用する際の注意事項を以下の項目ごとに説明します。

- C コンパイラ
 - アセンブラ
 - リンカ
 - デバッグ
-

■ C コンパイラ (fcc911)

C言語でプログラミングをするにあたって、リトルエンディアン領域に対して次のような操作を行ったときは、動作が保証できませんのでご注意ください。

- 初期値付き変数の配置
- 構造体代入
- 文字列操作関数を使った文字型配列以外の操作
- 文字列操作関数使用時の -K lib オプションの指定
- double 型, long double 型の利用
- スタックのリトルエンディアン領域への配置

● 初期値付き変数の配置

リトルエンディアン領域に、初期値付きの変数を配置することはできません。

コンパイラは、リトルエンディアンの初期値を生成する機能を持っていません。リトルエンディアン領域に変数を配置することはできますが、初期値を設定することはできません。

プログラムの先頭で、初期値を設定する処理を行ってください。

[例] リトルエンディアン領域の変数 `little_data` に初期値を設定する場合

```
extern int little_data;

void little_init(void) {
    little_data = 初期値;
}

void main(void) {
    little_init();
    ...
}
```

● 構造体代入

構造体どうしの代入を行うとき、コンパイラは最適な転送方法を選択してバイト、ハーフワード、ワードごとの転送を行います。このため、通常の領域に割り当てられた構造体変数とリトルエンディアンの領域に割り当てられた構造体変数の間で構造体代入を行うと、正しい結果が得られません。

構造体のメンバをそれぞれ代入してください。

[例] リトルエンディアン領域の構造体変数 `little_st` に構造体代入する場合

```
struct tag { char c; int i; } normal_st;
extern struct tag little_st;

#define STRMOVE(DEST, SRC) DEST.c=SRC.c; DEST.i=SRC.i;

void main(void) {
    STRMOVE(little_st, normal_st);
}
```

また、構造体のメンバの配置はコンパイラごとに違うので、ほかのコンパイラでコンパイルされた構造体とは、メンバの配置が違っていることが考えられます。このようなときには、前述の方法においても正しい結果が得られません。

構造体のメンバの配置が一致しないときは、リトルエンディアン領域に構造体変数を配置しないでください。

● 文字列操作関数を使った文字型配列以外の操作

標準ライブラリとして用意されている文字列操作関数はバイト単位での処理を行います。このためリトルエンディアン領域に配置された `char`, `unsigned char`, `signed char` 型以外の型を持つ領域に対し、文字列操作関数を使った処理を行うと、正しい結果が得られません。

このような処理は行わないでください。

[不具合例] `memcpy` でのワードデータの転送

```
int big = 0x01020304;    /* ビッグエンディアン領域 */
extern int little;        /* リトルエンディアン領域 */
memcpy(&little, &big, 4); /* memcpy による転送 */
```

上記の実行結果は

(ビッグエンディアン領域)					(リトルエンディアン領域)			
01	02	03	04	memcpy	01	02	03	04

となり、ワードデータの転送結果としては誤りになる。

(正しい結果)	04	03	02	01
-----------	----	----	----	----

● 文字列操作関数使用時の -K lib オプションの指定

-K lib オプションを指定すると、コンパイラはいくつかの文字列操作関数に対し、インライン展開を行います。このとき、最適な処理を選択するためハーフワードまたはワードごとの処理に変更される場合があります。

このためリトルエンディアン領域に対する処理が正しく実行されません。

リトルエンディアン領域に対し文字列操作関数を使った処理を行っているときは、-K lib オプションを指定しないでください。

-K lib オプションを包含する -O 4 オプションや -K speed オプションも同様に指定しないでください。

● double 型, long double 型の利用

double 型および long double 型へのアクセスは、上位 1 ワード、下位 1 ワードをそれぞれアクセスする方法で行われます。このため、リトルエンディアン領域に配置された double 型, long double 型変数に対するアクセスは、正しい結果が得られません。

リトルエンディアン領域に割り当てられた同一型の変数どうしの代入は可能ですが、最適化の結果これらの代入が定数の代入に置き換えられるときもあります。

double 型および long double 型変数をリトルエンディアン領域に配置しないでください。

[不具合例] double 型データの転送

```
double big = 1.0;      /* ビッグエンディアン領域 */
extern int little;     /* リトルエンディアン領域 */
little = big;          /* double 型データの転送 */
```

上記の実行結果は

(ビッグエンディアン領域)								(リトルエンディアン領域)							
3f	f0	00	00	00	00	00	00	00	00	f0	3f	00	00	00	00

となり、double 型データの転送結果としては誤りになります。

(正しい結果)

00	00	00	00	00	00	00	f0	3f
----	----	----	----	----	----	----	----	----

● スタックのリトルエンディアン領域への配置

リトルエンディアン領域にスタックの一部あるいは全部が配置された場合、動作を保証しません。

■ アセンブラ (fasm911)

FR のアセンブラ言語でプログラミングをするにあたって、リトルエンディアン領域に関して注意項目を以下に示します。

● セクションについて

リトルエンディアン領域は、主にリトルエンディアン系 CPU とデータ交換を行うことを目的としています。そのため、リトルエンディアン領域は初期値のないデータセクションとして定義してください。

もし、リトルエンディアン領域にコードやスタック、初期値付きデータセクションなどの指定をした場合には、本品種でのアクセス動作は保証できなくなります。

[例]

```
/* 正しいリトルエンディアン領域のセクション定義 */
```

```
.SECTION Little_Area, DATA, ALIGN=4
```

```
Little_Word:
```

```
.RES.W 1
```

```
Little_Half:
```

```
.RES.H 1
```

```
Little_Byte:
```

```
.RES.B 1
```

● データのアクセスについて

リトルエンディアン領域へのデータのアクセスを行う場合、そのデータの値は、エンディアンを意識せずにコーディングできます。ただし、リトルエンディアン領域のデータへのアクセスは、必ずデータサイズと同じサイズでアクセスしてください。

[例]

```
LDI    #0x01020304, r0
```

```
LDI    #Little_Word, r1
```

```
LDI    #0x0102, r2
```

```
LDI    #Little_Half, r3
```

```
LDI    #0x01, r4
```

```
LDI    #Little_Byte, r5
```

```
/* 32 ビットデータは、ST 命令 ( や LD 命令など ) でアクセスします。 */
```

```
ST      r0, @r1
```

```
/* 16 ビットデータは、STH 命令 ( や LDH 命令など ) でアクセスします。 */
```

```
STH     r2, @r3
```

/* 8 ビットデータは , STB 命令 (や LDB 命令など) でアクセスします。 */

STB r4, @r5

本品種でデータサイズと異なるサイズでアクセスした場合には , その値の保証はできません。例えば , 連続する 2 つの 16 ビットデータを , 32 ビットアクセス命令を使って一度にアクセスした場合にはデータの値の保証はできません。

■ リンカ (flnk911)

リトルエンディアン領域を使用するプログラムの作成において、リンク時のセクション配置で注意項目を以下に示します。

● セクション種別の制限

リトルエンディアン領域には、初期値なしデータセクションのみ配置することができます。

リトルエンディアン領域に初期値付きデータセクション、スタックセクションおよびコードセクションを配置した場合、リンカの内部では、ビッグエンディアンでアドレス解決などの演算処理を行っていますので、プログラム動作は保証できません。

● エラーの未検出

リンカは、リトルエンディアン領域の認識をしていませんので、上記制限事項に違反した配置が行われても、エラーメッセージを通知することはありません。リトルエンディアン領域に配置したセクションの内容を十分にご確認のうえご使用ください。

■ デバッガ (sim911, eml911, mon911)

● シミュレータデバッガ

リトルエンディアン領域を示すような、メモリ空間指定コマンドはありません。

したがって、メモリ操作コマンドや、メモリを操作する命令実行は、ビッグエンディアンとして扱われます。

● エミュレータデバッガ、モニタデバッガ

以下のコマンドでリトルエンディアン領域をアクセスした場合に、正常な値として扱われませんので注意してください。

- set memory/show memory/enter/examine/set watch コマンド

浮動小数点 (single/double) のデータを扱った場合、指定した値が設定・表示ともにできません。

- search memory コマンド

ハーフワード、ワードのデータの検索を行った場合、指定した値で検索が行われません。

- 行 / 逆アセンブル (ソースウィンドウの逆アセンブル表示を含む)

正常な命令コードが設定・表示ともにできません。

(リトルエンディアン領域には、命令コードを配置しないようにしてください。)

- call/show call コマンド

スタック領域が、リトルエンディアン領域に置かれた場合、正常に動作しません。

(リトルエンディアン領域にスタック領域を配置しないでください。)

付録 E 命令一覧表

FR ファミリの命令一覧表です。

■ 命令一覧表

図 E-1 命令一覧表の読み方

ニーモニック	型	OP	CYC	NZVC	動作	備考
ADD Rj, Rj	A	AG	1	CCCC	Ri + Rj -> Rj	
* ADD #s5, Rj	C	A4	1	CCCC	Ri + s5 -> Ri	
,	,	,	,	,	,	
,	,	,	,	,	,	

(1) (2) (3) (4) (5) (6) (7)

(1) 命令名が示されています。

* 印は、CPU 仕様ではなくアセンブラで命令を拡張または追加した拡張命令です。

(2) オペランドに指定可能なアドレッシングモードを記号で示されています。

記号の意味は、「● アドレッシングモードの記号」を参照してください。

(3) 命令フォーマットが示されています。

(4) 命令コードが 16 進数表示されています。

(5) マシンサイクル数を表しています。

a: メモリアクセスサイクルであり、Ready 機能により延びる可能性があります。

b: メモリアクセスサイクルであり、Ready 機能により延びる可能性があります。
ただし、LD動作の対象となるレジスタを直後の命令が参照する場合には、インタロックがかかり、実行サイクル数が + 1 増加します。

c: 直後の命令が、R15 または SSP または USP に対し、読出しあるいは書込みを行う命令であるとき、あるいは命令フォーマット A の命令であるとき、インタロックがかかり、実行サイクル数は + 1 増加して 2 となります。

d: 直後の命令が MDH/MDL を参照する場合、インタロックがかかり、実行サイクル数は増加して 2 となります。

a, b, c, d とも最小は 1 サイクルです。

(6) フラグ変化を表しています。

フラグ変化
C … 変化する
- … 変化しない
0 … クリア
1 … セット

フラグの意味
N… ネガティブフラグ
Z… ゼロフラグ
V… オーバフラグ
C… キャリフラグ

(7) 命令動作が表記されています。

● アドレッシングモードの記号

Ri	: レジスタ直接 (R0 ~ R15, AC, FP, SP)
Rj	: レジスタ直接 (R0 ~ R15, AC, FP, SP)
R13	: レジスタ直接 (R13, AC)
Ps	: レジスタ直接 (プログラムステータスレジスタ)
Rs	: レジスタ直接 (TBR, RP, SSP, USP, MDH, MDL)
Cri	: レジスタ直接 (CR0 ~ CR15)
CRj	: レジスタ直接 (CR0 ~ CR15)
#i8	: 符号なし 8 ビット即値 (- 128 ~ + 255) (注意事項) - 128 ~ - 1 は , + 128 ~ + 255 として扱います。
#i20	: 符号なし 20 ビット即値 (- 0X80000 ~ 0XFFFFF) (注意事項) - 0X7FFFF ~ - 1 は , 0X7FFFF ~ 0XFFFFFF として扱います。
#i32	: 符号なし 32 ビット即値 (- 0X80000000 ~ 0XFFFFFFFF) (注意事項) - 0X80000000 ~ - 1 は , 0X80000000 ~ 0XFFFFFFFF として扱います。
#s5	: 符号付き 5 ビット即値 (- 16 ~ + 15)
#s10	: 符号付き 10 ビット即値 (- 512 ~ + 508, 4 の倍数のみ)
#u4	: 符号なし 4 ビット即値 (0 ~ 15)
#u5	: 符号なし 5 ビット即値 (0 ~ 31)
#u8	: 符号なし 8 ビット即値 (0 ~ 255)
#u10	: 符号なし 10 ビット即値 (0 ~ 1020, 4 の倍数のみ)
@dir8	: 符号なし 8 ビット直接アドレス (0 ~ 0XFF)
@dir9	: 符号なし 9 ビット直接アドレス (0 ~ 0X1FE, 2 の倍数のみ)
@dir10	: 符号なし 10 ビット直接アドレス (0 ~ 0X3FC, 4 の倍数のみ)
label9	: 符号付き 9 ビット分岐アドレス (- 0X100 ~ + 0XFC, 2 の倍数のみ)
label12	: 符号付き 12 ビット分岐アドレス (- 0X800 ~ + 0X7FC, 2 の倍数のみ)
label20	: 符号付き 20 ビット分岐アドレス (- 0X80000 ~ + 0X7FFFF)
label32	: 符号付き 32 ビット分岐アドレス (- 0X80000000 ~ + 0X7FFFFFFF)
@Ri	: レジスタ間接 (R0 ~ R15, AC, FP, SP)
@Rj	: レジスタ間接 (R0 ~ R15, AC, FP, SP)
@(R13,Rj)	: レジスタ相対間接 (Rj: R0 ~ R15, AC, FP, SP)
@(R14,disp10)	: レジスタ相対間接 (disp10: - 0X200 ~ 0X1FC 4 の倍数のみ)
@(R14,disp9)	: レジスタ相対間接 (disp9: - 0X100 ~ 0XFE 2 の倍数のみ)
@(R14,disp8)	: レジスタ相対間接 (disp8: - 0X80 ~ 0X7F)
@(R15,udisp6)	: レジスタ相対間接 (udisp6: 0 ~ 60, 4 の倍数のみ)
@Ri+	: ポストインクリメント付きレジスタ間接 (R0 ~ R15, AC, FP, SP)
@R13+	: ポストインクリメント付きレジスタ間接 (R13, AC)
@SP+	: スタックポップ
@-SP	: スタックプッシュ
(reglist)	: レジスタリスト

● 命令フォーマット

	MSB	16bit	LSB
A	OP	Rj	Ri
	8	4	4
B	OP	i8/O8	Ri
	4	8	4
C	OP	u4/m4	Ri
	8	4	4
*C'	OP	s5/u5	Ri
	7	5	4
D	OP	u8/rel8/dir/ reglist	
	8	8	
E	OP	SUB-OP	Ri
	8	4	4
F	OP	rel11	
	5	11	

ADD, ADDN, CMP, LSL, LSR, ASR 命令のみ

表 E-1 加減算

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
ADD Rj, Ri	A	A6	1	CCCC	Ri+Rj->Ri	
*ADD #s5, Ri	C'	A4	1	CCCC	Ri+s5->Ri	アセンブラでは上位 1 ビットを符号と見る
ADD #u4, Ri	C	A4	1	CCCC	Ri+extu(i4)->Ri	ゼロ拡張
ADD2 #u4, Ri	C	A5	1	CCCC	Ri+extu(i4)->Ri	マイナス拡張
ADDN Rj, Ri	A	A7	1	CCCC	Ri+Rj+c->Ri	キャリ付き加算
ADDN Rj, Ri	A	A2	1	----	Ri+Rj->Ri	
*ADDN #s5, Ri	C'	A0	1	----	Ri+s5->Ri	アセンブラでは上位 1 ビットを符号と見る
ADDN #u4, Ri	C	A0	1	----	Ri+extu(i4)->Ri	ゼロ拡張
ADDN2 #u4, Ri	C	A1	1	----	Ri+extu(i4)->Ri	マイナス拡張
SUB Rj, Ri	A	AC	1	CCCC	Ri-Rj->Ri	
SUBC Rj, Ri	A	AD	1	CCCC	Ri-Rj-c->Ri	キャリ付き減算
SUBN Rj, Ri	A	AE	1	----	Ri-Rj->Ri	

表 E-2 比較演算

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
CMP Rj, Ri	A	AA	1	CCCC	Ri-Rj	
*CMP #s5, Ri	C'	A8	1	CCCC	Ri-s5	アセンブラでは上位 1 ビットを符号と見る
CMP #u4, Ri	C	A8	1	CCCC	Ri-extu(i4)	ゼロ拡張
CMP2 #u4, Ri	C	A9	1	CCCC	Ri-extu(i4)	マイナス拡張

表 E-3 論理演算

ニーモニク	型	OP	CYCLE	NZVC	動作	RMW	備考
AND Rj, Ri	A	82	1	CC--	Ri &= Rj	—	ワード
AND Rj, @Ri	A	84	1+2a	CC--	(Ri) &= Rj	—	ワード
ANDH Rj, @Ri	A	85	1+2a	CC--	(Ri) &= Rj	—	ハーフワード
ANDB Rj, @Ri	A	86	1+2a	CC--	(Ri) &= Rj	—	バイト
OR Rj, Ri	A	92	1	CC--	Ri = Rj	—	ワード
OR Rj, @Ri	A	94	1+2a	CC--	(Ri) = Rj	—	ワード
ORH Rj, @Ri	A	95	1+2a	CC--	(Ri) = Rj	—	ハーフワード
ORB Rj, @Ri	A	96	1+2a	CC--	(Ri) = Rj	—	バイト
EOR Rj, Ri	A	9A	1	CC--	Ri ^= Rj	—	ワード
EOR Rj, @Ri	A	9C	1+2a	CC--	(Ri) ^= Rj	—	ワード
EORH Rj, @Ri	A	9D	1+2a	CC--	(Ri) ^= Rj	—	ハーフワード
EORB Rj, @Ri	A	9E	1+2a	CC--	(Ri) ^= Rj	—	バイト

表 E-4 ビット操作命令

ニーモニク	型	OP	CYCLE	NZVC	動作	RMW	備考
BANDL #u4, @Ri	C	80	1+2a	----	(Ri)&=(0xF0+u4)	—	下位 4 ビットを操作
BANDH #u4, @Ri	C	81	1+2a	----	(Ri)&=((u4<<4)+0xF)	—	上位 4 ビットを操作
*BAND #u8, @Ri *1				----	(Ri)&=u8	—	
BORL #u4, @Ri	C	90	1+2a	----	(Ri) = u4	—	下位 4 ビットを操作
BORH #u4, @Ri	C	91	1+2a	----	(Ri) = (u4<<4)	—	上位 4 ビットを操作
*BOR #u8, @Ri *2				----	(Ri) = u8	—	
BEORL #u4, @Ri	C	98	1+2a	----	(Ri) ^= u4	—	下位 4 ビットを操作
BEORH #u4, @Ri	C	99	1+2a	----	(Ri) ^= (u4<<4)	—	上位 4 ビットを操作
*BEOR #u8, @Ri *3				----	(Ri) ^= u8	—	
BTSTL #u4, @Ri	C	88	2+a	0C--	(Ri) & u4	—	下位 4 ビットをテスト
BTSTH #u4, @Ri	C	89	2+a	CC--	(Ri) & (u4<<4)	—	上位 4 ビットをテスト

*1: アセンブラは、u8&0F_H でビットが立っていれば、BANDL を生成し、u8&F0_H でビットが立っていれば、BANDH を生成します。BANDL, BANDH 両方生成する場合があります。

*2: アセンブラは、u8&0F_H でビットが立っていれば、BORL を生成し、u8&F0_H でビットが立っていれば、BORH を生成します。BORL, BORH 両方生成する場合があります。

*3: アセンブラは、u8&0F_H でビットが立っていれば、BEORL を生成し、u8&F0_H でビットが立っていれば、BEORH を生成します。BEORL, BEORH 両方生成する場合があります。

表 E-5 乗除算

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
MUL Rj,Ri	A	AF	5	CCC-	Ri * Rj -> MDH,MDL	32bit × 32bit=64bit
MULU Rj,Ri	A	AB	5	CCC-	Ri * Rj -> MDH,MDL	符号なし
MULH Rj,Ri	A	BF	3	CC--	Ri * Rj -> MDL	16bit × 16bit=32bit
MULUH Rj,Ri	A	BB	3	CC--	Ri * Rj -> MDL	符号なし
DIV0S Ri	E	97-4	1	----		ステップ演算
DIV0U Ri	E	97-5	1	----		32bit/32bit=32bit
DIV1 Ri	E	97-6	d	-C-C		
DIV2 Ri	E	97-7	1	-C-C		
DIV3	E	9F-6	1	----		
DIV4S	E	9F-7	1	----		
*DIV Ri * ¹			36	-C-C	MDL / Ri -> MDL , MDL % Ri -> MDH	
*DIVU Ri * ²				-C-C	MDL / Ri -> MDL , MDL % Ri -> MDH	

表 E-6 シフト

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
LSL Rj, Ri	A	B6	1	CC-C	Ri << Rj -> Ri	論理シフト
*LSL #u5, Ri(u5:0 ~ 31)	C'	B4	1	CC-C	Ri << u5 -> Ri	
LSL #u4, Ri	C	B4	1	CC-C	Ri << u4 -> Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	Ri <<(u4+16) -> Ri	
LSR Rj, Ri	A	B2	1	CC-C	Ri >> Rj -> Ri	論理シフト
*LSR #u5, Ri(u5:0 ~ 31)	C'	B0	1	CC-C	Ri >> u5 -> Ri	
LSR #u4, Ri	C	B0	1	CC-C	Ri >> u4 -> Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	Ri >>(u4+16) -> Ri	
ASR Rj, Ri	A	BA	1	CC-C	Ri >> Rj -> Ri	算術シフト
*ASR #u5, Ri (u5:0 ~ 31)	C'	B8	1	CC-C	Ri >> u5 -> Ri	
ASR #u4, Ri	C	B8	1	CC-C	Ri >> u4 -> Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	Ri >>(u4+16) -> Ri	

表 E-7 即値セット /16 ビット /32 ビット即値転送命令

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
LDI:32 #i32, Ri	E	9F-8	3	----	i32 -> Ri	上位 12 ビットはゼロ拡張
LDI:20 #i20, Ri	C	9B	2	----	i20 -> Ri	上位 24 ビットはゼロ拡張
LDI:8 #i8, Ri	B	C0	1	----	i8 -> Ri	
*LDI # {i8 i20 i32} ,Ri * ₃					{i8 i20 i32} -> Ri	

*1: DIV0S, DIV1 × 32, DIV2, DIV3, DIV4S を生成する。命令コード長は、72 バイトとなります。

*2: DIV0U, DIV1 × 32 を生成する。命令コード長は、66 バイトとなります。

*3: 即値が絶対値の場合は、i8, i20, i32 の選択はアセンブラが自動的に行います。

即値が、相対値または外部参照シンボルを含む場合は、i32 が選択されます。

表 E-8 メモリロード

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
LD @Rj, Ri	A	04	b	----	(Rj)->Ri	Rs : 特殊レジスタ *
LD @(R13,Rj), Ri	A	00	b	----	(R13+Rj)->Ri	
LD @(R14,disp10), Ri	B	2	b	----	(R14+disp10)->Ri	
LD @(R15,udisp6), Ri	C	03	b	----	(R15+udisp6)->Ri	
LD @R15+, Ri	E	07-0	b	----	(R15)->Ri, R15+=4	
LD @R15+, Rs	E	07-8	b	----	(R15)->Rs, R15+=4	
LD @R15+, PS	E	07-9	1+a+b	CCCC	(R15)->PS, R15+=4	
LDUH @Rj, Ri	A	05	b	----	(Rj)->Ri	ゼロ拡張
LDUH @(R13,Rj), Ri	A	01	b	----	(R13+Rj)->Ri	ゼロ拡張
LDUH @(R14,disp9), Ri	B	4	b	----	(R14+disp9)->Ri	ゼロ拡張
LDUB @Rj, Ri	A	06	b	----	(Rj)->Ri	ゼロ拡張
LDUB @(R13,Rj), Ri	A	02	b	----	(R13+Rj)->Ri	ゼロ拡張
LDUB @(R14,disp8), Ri	B	6	b	----	(R14+disp8)->Ri	ゼロ拡張

※: ハード仕様の o8, o4 のフィールドには, 次のようにアセンブラが計算して値を設定します。

disp10/4 o8, disp9/2 o8, disp8 o8, disp10, disp9, disp8 は符号付き, udisp6/4 o4 udisp6 は符号なし。

表 E-9 メモリストア

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
ST Ri, @Rj	A	14	a	----	Ri->(Rj)	ワード
ST Ri, @(R13,Rj)	A	10	a	----	Ri->(R13+Rj)	ワード
ST Ri, @(R14,disp10)	B	3	a	----	Ri->(R14+disp10)	ワード
ST Ri, @(R15,udisp6)	C	13	a	----	Ri->(R15+udisp6)	Rs 特殊レジスタ *
ST Ri, @-R15	E	17-0	a	----	R15-=4, Ri->(R15)	
ST Rs, @-R15	E	17-8	a	----	R15-=4, Rs->(R15)	
ST PS, @-R15	E	17-9	a	----	R15-=4, PS->(R15)	
STH Ri, @Rj	A	15	a	----	Ri->(Rj)	ハーフワード
STH Ri, @(R13,Rj)	A	11	a	----	Ri->(R13+Rj)	ハーフワード
STH Ri, @(R14,disp9)	B	5	a	----	Ri->(R14+disp9)	ハーフワード
STB Ri, @Rj	A	16	a	----	Ri->(Rj)	バイト
STB Ri, @(R13,Rj)	A	12	a	----	Ri->(R13+Rj)	バイト
STB Ri, @(R14,disp8)	B	7	a	----	Ri->(R14+disp8)	バイト

※: ハード仕様の o8, o4 のフィールドには, 次のようにアセンブラが計算して値を設定します。

disp10/4 o8, disp9/2 o8, disp8 o8, disp10, disp9, disp8 は符号付き, udisp6/4 o4 udisp6 は, 符号なし。

表 E-10 レジスタ間転送

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
MOV Rj, Ri	A	8B	1	----	Rj -> Ri	汎用レジスタ間転送 Rs : 特殊レジスタ Rs : 特殊レジスタ *
MOV Rs, Ri	A	B7	1	----	Rs -> Ri	
MOV Ri, Rs	E	B3	1	----	Ri -> Rs	
MOV PS, Ri	E	17-1	1	----	PS -> Ri	
MOV Ri, PS	E	07-1	c	CCCC	Ri -> PS	

※: 特殊レジスタ Rs : TBR, RP, USP, SSP, MDH, MDL

表 E-11 通常分岐 (遅延なし)

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
JMP @Ri	E	97-0	2	----	Ri -> PC	
CALL label12	E	D0	2	----	PC+2->RP, PC+2+(label12-PC-2)->PC	
CALL @Ri	F	97-1	2	----	PC+2->RP, Ri->PC	
RET	E	97-2	2	----	RP -> PC	リターン
INT #u8	D	1F	3+3a	----	SSP-=4, PS->(SSP), SSP-=4, PC+2->(SSP), 0->I フラグ, 0->S フラグ, (TBR+0x3FC-u8 × 4)->PC	
INTE	E	9F-3	3+3a	----	SSP-=4, PS->(SSP), SSP-=4, PC+2->(SSP), 0->S フラグ, (TBR+10x3D8)->PC	
RETI	E	97-3	2+2A	CCCC	(R15)->PC, R15-=4, (R15)->PS, R15-=4	エミュレータ用
BRA label9	D	E0	2	----	PC+2+(label9-PC-2)->PC	
BNO label9	D	E1	1	----	非分岐	
BEQ label9	D	E2	2/1	----	if(Z==1) then PC+2+(label9-PC-2)->PC	
BNE label9	D	E3	2/1	----	s/Z==0	
BC label9	D	E4	2/1	----	s/C==1	
BNC label9	D	E5	2/1	----	s/C==0	
BN label9	D	E6	2/1	----	s/N==1	
BP label9	D	E7	2/1	----	s/N==0	
BV label9	D	E8	2/1	----	s/V==1	
BNV label9	D	E9	2/1	----	s/V==0	
BLT label9	D	EA	2/1	----	s/V xor N==1	
BGE label9	D	EB	2/1	----	s/V xor N==0	
BLE label9	D	EC	2/1	----	s/(V xor N) or Z==1	
BGT label9	D	ED	2/1	----	s/(V xor N) or Z==0	
BLS label9	D	EE	2/1	----	s/C or Z==1	
BHI label9	D	EF	2/1	----	s/C or Z==0	

- (注意事項) ・ CYCLE 数の「2/1」は、分岐するとき 2、分岐しないとき 1であることを示します。
- ハード仕様の rel11, rel8 のフィールドには、次のようにアセンブラが計算して値を設定します。
(label12-PC-2)/2 rel11, (label9-PC-2)/2 rel8, label12, label9 は符号付き。
 - RETI 命令を実行するときは S フラグが "0" であることが必要です。

表 E-12 遅延分岐

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
JMP:D @Ri	E	9F-0	1	----	Ri -> PC	
CALL:D label12	F	D8	1	----	PC+4->RP, PC+2+(label12-PC-2)->PC	
CALL:D @Ri	E	9F-1	1	----	PC+4->RP, Ri->PC	
RET:D	E	9F-2	1	----	RP -> PC	リターン
BRA:D label9	D	F0	1	----	PC+2+(label9-PC-2)->PC	
BNO:D label9	D	F1	1	----	非分岐	
BEQ:D label9	D	F2	1	----	if(Z==1) then PC+2+(label9-PC-2)->PC	
BNE:D label9	D	F3	1	----	s/Z==0	
BC:D label9	D	F4	1	----	s/C==1	
BNC:D label9	D	F5	1	----	s/C==0	
BN:D label9	D	F6	1	----	s/N==1	
BP:D label9	D	F7	1	----	s/N==0	
BV:D label9	D	F8	1	----	s/V==1	
BNV:D label9	D	F9	1	----	s/V==0	
BLT:D label9	D	FA	1	----	s/V xor N==1	
BGE:D label9	D	FB	1	----	s/V xor N==0	
BLE:D label9	D	FC	1	----	s/(V xor N) or Z==1	
BGT:D label9	D	FD	1	----	s/(V xor N) or Z==0	
BLS:D label9	D	FE	1	----	s/C or Z==1	
BHI:D label9	D	FF	1	----	s/C or Z==0	

(注意事項) ・ ハード仕様の rel11, rel8 のフィールドには、次のようにアセンブラが計算して値を設定します。

(label12-PC-2)/2 rel11, (label9-PC-2)/2 rel8, label12, label9 は符号付き。

- ・ 遅延分岐は必ず次の命令 (遅延スロット) を実行してから分岐します。
- ・ 遅延スロットに置くことができる命令は、すべての1サイクル, a, b, c, dサイクル命令です。複数サイクル命令は置けません。

表 E-13 その他

ニーモニック	型	OP	CYCLE	NZVC	動作	RMW	備考
NOP	E	9F-A	1	----	何も変化しない	—	
ANDCCR #u8	D	83	c	CCCC	CCR and u8 -> CCR	—	
ORCCR #u8	D	93	c	CCCC	CCR or u8 -> CCR	—	
STILM #u8	D	87	1	----	i8 -> ILM	—	ILM 即値セット
ADDSP #s10 ^{*1}	D	A3	1	----	R15 += s10	—	ADD SP 命令
EXTSB Ri	E	97-8	1	----	符号拡張 8->32bit	—	
EXTUB Ri	E	97-9	1	----	ゼロ拡張 8->32bit	—	
EXTSH Ri	E	97-A	1	----	符号拡張 16->32bit	—	
EXTUH Ri	E	97-B	1	----	ゼロ拡張 16->32bit	—	
LDM0 (reglist)	D	8C		----	(R15)->reglist, R15 インクリメント	—	ロードマルチ R0-R7
LDM1 (reglist)	D	8D		----	(R15)->reglist, R15 インクリメント	—	ロードマルチ R8-R15
*LDM (reglist) ^{*2}				----	(R15)->reglist, R15 インクリメント	—	ロードマルチ R0-R15
STM0 (reglist)	D	8E		----	R15 デクリメント, reglist->(R15)	—	ストアマルチ R0-R7
STM1 (reglist)	D	8F		----	R15 デクリメント, reglist->(R15)	—	ストアマルチ R8-R15
*STM (reglist) ^{*3}				----	R15 デクリメント, reglist->(R15)	—	ストアマルチ R0-R15
ENTER #u10 ^{*4}	D	0F	1+a	----	R14 -> (R15 - 4), R15 - 4 -> R14, R15 - u10 -> R15	—	関数の入口処理
LEAVE	E	9F-9	b	----	R14 + 4 -> R15, (R15 - 4) -> R14	—	関数の出口処理
XCHB @Rj, Ri	A	8A	2a	----	Ri -> TEMP (Rj) -> Ri TEMP -> (Rj)		セマフォ管理用 バイトデータ

*1: s10 はアセンブラが, s10/4 を計算して s8 にして値を設定します。s10 は符号付き。

*2: reglist で, R0 ~ R7 のいずれかの指定があれば, LDM0 を生成し, R8 ~ R15 のいずれかの指定があれば, LDM1 を生成します。LDM0, LDM1 両方生成する場合があります。

*3: reglist で, R0 ~ R7 のいずれかの指定があれば, STM0 を生成し, R8 ~ R15 のいずれかの指定があれば, STM1 を生成します。STM1, STM0 両方生成する場合があります。

*4: u10 はアセンブラが, u10/4 を計算して u8 にして値を設定します。u10 は符号なし。

(注意事項) ・ LDM0 (reglist), LDM1 (reglist) の実行サイクル数は, 指定されたレジスタ数が n のとき,
 $a \times (n - 1) + b + 1$ サイクルとなります。
 ・ STM0 (reglist), STM1 (reglist) の実行サイクル数は, 指定されたレジスタ数が n のとき,
 $a \times n + 1$ サイクルとなります。

表 E-14 20 ビット通常分岐マクロ命令

ニーモニック	動作	備考
*CALL20 label20,Ri	次の命令のアドレス ->RP, label20->PC	Ri: テンポラリレジスタ 参考 1 参照
*BRA20 label20,Ri	label20->PC	Ri: テンポラリレジスタ 参考 2
*BEQ20 label20,Ri	if(Z==1) then label20->PC	Ri: テンポラリレジスタ 参考 3
*BNE20 label20,Ri	s/Z==0	
*BC20 label20,Ri	s/C==1	
*BNC20 label20,Ri	s/C==0	
*BN20 label20,Ri	s/N==1	
*BP20 label20,Ri	s/N==0	
*BV20 label20,Ri	s/V==1	
*BNV20 label20,Ri	s/V==0	
*BLT20 label20,Ri	s/V xor N==1	
*BGE20 label20,Ri	s/V xor N==0	
*BLE20 label20,Ri	s/(V xor N) or Z==1	
*BGT20 label20,Ri	s/(V xor N) or Z==0	
*BLS20 label20,Ri	s/C or Z==1	
*BHI20 label20,Ri	s/C or Z==0	

[参考 1] CALL20

(1) label20-PC-2 が , - 0x800 ~ + 0x7fe の場合は , 次のように命令を生成します。

```
CALL label12
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:20 #label20,Ri
```

```
CALL @Ri
```

[参考 2] BRA20

(1) label20-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
BRA label9
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:20 #label20,Ri
```

```
JMP @Ri
```

[参考 3] Bcc20

(1) label20-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
Bcc label9
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
Bxcc false xcc は cc の背反条件
```

```
LDI:20 #label20,Ri
```

```
JMP @Ri
```

```
false:
```


表 E-15 20 ビット遅延分岐マクロ命令

ニーモニク	動作	備考
*CALL20:D label20,Ri	次の命令のアドレス +2->RP, label20->PC	Ri: テンポラリレジスタ 参考 1 参照
*BRA20:D label20,Ri	label20->PC	Ri: テンポラリレジスタ 参考 2 参照
*BEQ20:D label20,Ri	if(Z==1) then label20->PC	Ri: テンポラリレジスタ 参考 3 参照
*BNE20:D label20,Ri	s/Z==0	
*BC20:D label20,Ri	s/C==1	
*BNC20:D label20,Ri	s/C==0	
*BN20:D label20,Ri	s/N==1	
*BP20:D label20,Ri	s/N==0	
*BV20:D label20,Ri	s/V==1	
*BNV20:D label20,Ri	s/V==0	
*BLT20:D label20,Ri	s/V xor N==1	
*BGE20:D label20,Ri	s/V xor N==0	
*BLE20:D label20,Ri	s/(V xor N) or Z==1	
*BGT20:D label20,Ri	s/(V xor N) or Z==0	
*BLS20:D label20,Ri	s/C or Z==1	
*BHI20:D label20,Ri	s/C or Z==0	

[参考 1] CALL20:D

(1) label20-PC-2 が , - 0x800 ~ + 0x7fe の場合は , 次のように命令を生成します。

```
CALL:D label12
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:20 #label20,Ri
```

```
CALL:D @Ri
```

[参考 2] BRA20:D

(1) label20-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
BRA:D label9
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:20 #label20,Ri
```

```
JMP:D @Ri
```

[参考 3] Bcc20:D

(1) label20-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
Bcc:D label9
```

(2) label20-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
Bxcc false xcc は , cc の背反条件
```

```
LDI:20 #label20,Ri
```

```
JMP:D @Ri
```

```
false:
```

表 E-16 32 ビット通常分岐マクロ命令

ニーモニック	動作	備考
*CALL32 label32,Ri	次の命令のアドレス ->RP, label32->PC	Ri: テンポラリレジスタ 参考 1 参照
*BRA32 label32,Ri	label32->PC	Ri: テンポラリレジスタ 参考 2 参照
*BEQ32 label32,Ri	if(Z==1) then label32->PC	Ri: テンポラリレジスタ 参考 3 参照
*BNE32 label32,Ri	s/Z==0	
*BC32 label32,Ri	s/C==1	
*BNC32 label32,Ri	s/C==0	
*BN32 label32,Ri	s/N==1	
*BP32 label32,Ri	s/N==0	
*BV32 label32,Ri	s/V==1	
*BNV32 label32,Ri	s/V==0	
*BLT32 label32,Ri	s/V xor N==1	
*BGE32 label32,Ri	s/V xor N==0	
*BLE32 label32,Ri	s/(V xor N) or Z==1	
*BGT32 label32,Ri	s/(V xor N) or Z==0	
*BLS32 label32,Ri	s/C or Z==1	
*BHI32 label32,Ri	s/C or Z==0	

[参考 1] CALL32

(1) label32-PC-2 が , - 0x800 ~ + 0x7fe の場合は , 次のように命令を生成します。

```
CALL label12
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:32 #label32,Ri
CALL @Ri
```

[参考 2] BRA32

(1) label32-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
BRA label9
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:32 #label32,Ri
JMP @Ri
```

[参考 3] Bcc32

(1) label32-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
Bcc label9
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
Bxcc false xcc は , cc の背反条件
LDI:32 #label32,Ri
JMP @Ri
false:
```

表 E-17 32 ビット遅延分岐マクロ命令

ニーモニック	動作	備考
*CALL32D label32,Ri	次の命令のアドレス +2->RP, label32->PC	Ri: テンポラリレジスタ 参考 1 参照
*BRA32:D label32,Ri	label32->PC	Ri: テンポラリレジスタ 参考 2 参照
*BEQ32:D label32,Ri	if(Z==1) then label32->PC	Ri: テンポラリレジスタ 参考 3 参照
*BNE32:D label32,Ri	s/Z==0	
*BC32:D label32,Ri	s/C==1	
*BNC32:D label32,Ri	s/C==0	
*BN32:D label32,Ri	s/N==1	
*BP32:D label32,Ri	s/N==0	
*BV32:D label32,Ri	s/V==1	
*BNV32:D label32,Ri	s/V==0	
*BLT32:D label32,Ri	s/V xor N==1	
*BGE32:D label32,Ri	s/V xor N==0	
*BLE32:D label32,Ri	s/(V xor N) or Z==1	
*BGT32:D label32,Ri	s/(V xor N) or Z==0	
*BLS32:D label32,Ri	s/C or Z==1	
*BHI32:D label32,Ri	s/C or Z==0	

[参考 1] CALL32:D

(1) label32-PC-2 が , - 0x800 ~ + 0x7fe の場合は , 次のように命令を生成します。

```
CALL:D label12
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:32 #label32,Ri
```

```
CALL:D @Ri
```

[参考 2] BRA32:D

(1) label32-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
BRA:D label9
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
LDI:32 #label32,Ri
```

```
JMP:D @Ri
```

[参考 3] Bcc32:D

(1) label32-PC-2 が , - 0x100 ~ + 0xfe の場合は , 次のように命令を生成します。

```
Bcc:D label9
```

(2) label32-PC-2 が , (1) の範囲外の場合および外部参照シンボルを含む場合は , 次のように命令を生成します。

```
Bxcc false xcc は , cc の背反条件
```

```
LDI:32 #label32,Ri
```

```
JMP:D @Ri
```

```
false:
```

表 E-18 ダイレクトアドレッシング

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
DMOV @dir10, R13	D	08	b	----	(dir10)-> R13	ワード
DMOV R13, @dir10	D	18	a	----	R13 ->(dir10)	ワード
DMOV @dir10, @R13+	D	0C	2a	----	(dir10)->(R13),R13+=4	ワード
DMOV @R13+, @dir10	D	1C	2a	----	(R13)->(dir10),R13+=4	ワード
DMOV @dir10, @-R15	D	0B	2a	----	R15-=4, (R15)->(dir10)	ワード
DMOV @R15+, @dir10	D	1B	2a	----	(R15)->(dir10),R15+=4	ワード
DMOVH @dir9, R13	D	09	b	----	(dir9)-> R13	ハーフワード
DMOVH R13, @dir9	D	19	a	----	R13 ->(dir9)	ハーフワード
DMOVH @dir9, @R13+	D	0D	2a	----	(dir9)->(R13),R13+=2	ハーフワード
DMOVH @R13+, @dir9	D	1D	2a	----	(R13)->(dir9),R13+=2	ハーフワード
DMOV B @dir8, R13	D	0A	b	----	(dir8)-> R13	バイト
DMOV B R13, @dir8	D	1A	a	----	R13 ->(dir8)	バイト
DMOV B @dir8, @R13+	D	0E	2a	----	(dir8)->(R13),R13++	バイト
DMOV B @R13+, @dir8	D	1E	2a	----	(R13)->(dir8),R13++	バイト

(注意事項) dir8, dir9, dir10 フィールドには , 次のようにアセンブラが計算して値を設定します。

dir8 dir, dir9/2 dir, dir10/4 dir dir8, dir9, dir10 は符号なし。

表 E-19 リソース命令

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
LDRES @Ri+, #u4	C	BC	a	----	(Ri)->u4 のリソース Ri+=4	u4: チャンネル番号
STRES #u4, @Ri+	C	BD	a	----	u4 のリソース ->(Ri) Ri+=4	u4: チャンネル番号

(注意事項) 本品種では , チャンネル番号を持つリソースを搭載していないので , 使用できません。

表 E-20 コプロセッサ制御命令

{CRi | CRj} := CR0 | CR1 | CR2 | CR3 | CR4 | CR5 | CR6 | CR7 | CR8 | CR9 | CR10 |
CR11 | CR12 | CR13 | CR14 | CR15

u4: := チャンネル指定

u8: := コマンド指定

ニーモニック	型	OP	CYCLE	NZVC	動作	備考
COPOP #u4, #u8, CRj, Cri	E	9F-C	2+a	----	演算指示	エラートラップなし
COPLD #u4, #u8, Rj, Cri	E	9F-D	1+2a	----	Rj -> CRi	
COPST #u4, #u8, CRj, Ri	E	9F-E	1+2a	----	CRj -> Ri	
COPSV #u4, #u8, CRj, Ri	E	9F-F	1+2a	----	CRj -> Ri	

(注意事項) 本品種では , コプロセッサを搭載していないため使用できません。

付録 F 使用上の注意

MB91265A シリーズを使用する際の注意事項を説明します。

■ 共通事項

● クロック制御部

$\overline{\text{INIT}}$ への "L" 入力時には、発振安定待ち時間を確保してください。

● 兼用ポートの機能切換え

PORT と兼用端子の切換えは、PFR (ポートファンクションレジスタ) で行います。
ただし、バス端子は外バスの設定により切り換わります。

● D-bus メモリ

コード領域を、D-bus 上のメモリに設定しないでください。

D-bus へは命令フェッチを行いませんので、D-bus 領域へ命令フェッチを行った場合、誤ったデータをコードとして解釈するため、暴走する可能性があります。

● 低消費電力モード

- (1) スタンバイモードに入れる場合は、同期スタンバイモード (TBCR: タイムベースカウンタ制御レジスタの bit8 SYNC5 ビットにて設定します) を使用した上で、以下のシーケンスを必ず使用してください。

```
/* STCR ライト */
ldi    #_STCR, R0          ; STCR レジスタ (0x0481)
ldi    #_Val_of_Stby, r1    ; Val_of_Stby は、STCR へのライトデータ
stb     r1, @r0             ; STCR へのライト

/* STBR ライト */
ldi     #_CTBR, r2          ; CTBR レジスタ (0x0483)
ldi     #0xA5, r1           ; クリアコマンド (1)
stb     r1, @r2             ; CTBR への A5 ライト
ldi     #0xA5, r1           ; クリアコマンド (2)
stb     r1, @r2             ; CTBR への A5 ライト

/* ここでタイムベースカウンタクリア */
ldub    @r0, r1             ; STCR リード

/* 同期スタンバイ遷移開始 */
ldub    @r0, r1             ; STCR ダミーリード
nop                                           ; タイミング調整用の NOP × 5
nop
nop
nop
nop
nop
```

- (2) モニタデバッガを使用される場合は、以下のことを行わないでください。

- 上記命令列に対して、ブレークポイントを設定しないでください。
- また、上記命令列に対して、ステップ実行を行わないでください。

● PS レジスタに関する注意事項

一部の命令で PS レジスタを先行処理しているため、下記の例外動作により、デバッグ使用時に割り込み処理ルーチンでブレークしたり、PS フラグの表示内容が更新されたりする場合があります。

いずれの場合も、EIT から復帰後以降に、正しく再処理を行うように設計されていますので、EIT 前後の動作は仕様どおりの処理を行います。

- DIV0U/DIV0S 命令の直前の命令で、(a) ユーザ割り込み・NMI を受け付けた場合、(b) ステップ実行を行った場合、(c) データイベントまたはエミュレータメニューにてブレークした場合、以下の動作を行う場合があります。
 - (1) D0, D1 フラグが、先行して更新されます。
 - (2) EIT 処理ルーチン (ユーザ割り込み・NMI, またはエミュレータ) を実行します。
 - (3) EIT から復帰後、DIV0U/DIV0S 命令が実行され、D0/D1 フラグが (1) と同じ値に更新されます。
- ユーザ割り込み・NMI 要因が発生している状態で、割り込みを許可するために OR CCR, ST ILM, MOV Ri, PS の各命令が実行されると、以下の動作を行います。
 - (1) PS レジスタが、先行して更新されます。
 - (2) EIT 処理ルーチン (ユーザ割り込み・NMI) を実行します。
 - (3) EIT から復帰後、上記命令が実行され、PS レジスタが (1) と同じ値に更新されます。

● ウォッチドッグタイマ機能について

本品種が備えているウォッチドッグタイマ機能は、プログラムが一定時間内にリセット延期動作を行うことを監視し、プログラムの暴走によりリセット延期動作が行われなかったときに、CPU をリセットするための機能です。そのため、いったんウォッチドッグタイマ機能を有効にすると、リセットするまで動作を続けます。

例外として、CPU のプログラム実行が停止する条件では自動的にリセット延期を行います。この例外にあてはまる条件については、「3.11.8 クロック制御部が持つ周辺回路 タイムベースカウンタ ウォッチドッグタイマ」のウォッチドッグタイマの一時停止 (自動発生延期) の項を参照してください。

■ デバッグ関連の注意事項

● RETI 命令のステップ実行

ステップ実行する際、割込みが頻繁に発生する環境下では、RETI をステップ実行後に該当割込み処理ルーチンだけを繰り返して実行します。その結果、メインルーチンや割込みレベルの低いプログラムの実行が行われなくなります。

回避のために、RETI 命令をステップ実行しないでください。

または、該当割込みルーチンのデバッグが不要になった段階で、該当割込みを禁止してデバッグを行ってください。

● オペランドブレイク

システムスタックポインタのアドレスを含む領域に対するアクセスを データイベントブレイクの対象に設定しないでください。

● フラッシュメモリの未使用領域実行

フラッシュメモリの未使用領域 (データが "FFFF_H") を誤って実行してしまうと、ブレイクを受け付けない状態になってしまいます。これを回避するために、デバッグのコードイベントのアドレスマスク機能を使用して、未使用領域の命令アクセス時にブレイクさせることを推奨致します。

● パワーオンデバッグ

パワーオンデバッグで電源をオフにするときは、次の3条件をすべて満たす状態で行ってください。

- (1) ユーザ電源が 0.9V_{cc} から 0.5V_{cc} まで低下する時間が 25 μ s 以上
(注意事項) 2 電源の場合は、V_{cc} は外部 I/O 電源電圧のことを指します。
- (2) CPU 動作周波数が 1 MHz 以上
- (3) ユーザプログラム実行中

● NMI 要求 (tool) に対する割込みハンドラ

ICE 未接続の状態において、ICE からのブレイク要求のみでセットされる要因フラグが、DSU 端子へのノイズなどの影響で誤って立ってしまった場合の誤動作を防ぐため、次のプログラムを割込みハンドラに追加してください。なお、このプログラムを追加した状態で ICE を使用しても問題ありません。

追加場所

次の割込みハンドラ

割込み要因	: NMI 要求 (tool)
割込み番号	: 13 (10 進) , 0D (16 進)
オフセット	: 3C8 _H
TBR がデフォルトのアドレス	: 000FFFC8 _H

追加プログラム

```

STM    (R0, R1)
LDI    #B00H, R0    ; B00H は DSU のブレイク要因レジスタのアドレス
LDI    #0, R1
STB    R1, @R0      ; ブレイク要因レジスタをクリア
LDM    (R0, R1)
RETI
```

索引

Numerics

16/8 ビット	
FR-CPU プログラミングモード	
(16/8 ビット, 読出し / 書込み)	514
16 ビット	
16 ビットアウトプットコンペア	
タイミング	288
16 ビットアウトプットコンペアとフリーラン	
タイマの動作について	289
16 ビットアウトプットコンペアの使用上の	
注意	310
16 ビットアウトプットコンペアの動作	
(セット / リセットモード, MOD1x=1)	287
16 ビットアウトプットコンペアの動作	
(反転モード, MOD1x=0)	284
16 ビットアウトプットコンペアの	
プログラム例	313
16 ビットアウトプットコンペアの	
レジスタ	220
16 ビットアウトプットコンペア用フリーラン	
タイマの選択	283
16 ビットアウトプットコンペア割込み	273
16 ビットインプットキャプチャ	
入力タイミング	295
16 ビットインプットキャプチャの使用上の注意	310
16 ビットインプットキャプチャの動作	295
16 ビットインプットキャプチャの	
ブロックダイアグラム	213
16 ビットインプットキャプチャのレジスタ	
	221
16 ビットインプットキャプチャ用フリーラン	
タイマの選択	294
16 ビットインプットキャプチャ割込み	273
16 ビットデッドタイム制御レジスタ, 上位	
(DTCR0)	256
16 ビットデッドタイム制御レジスタ, 上位	
(DTCR2)	262
16 ビットデッドタイムレジスタ (TMRRH0 ~	
TMRRH2, TMRRL0 ~ TMRRL2)	255
16 ビットフリーランタイムの使用上の注意	310
16 ビットフリーランタイムのプログラム例	
	312
16 ビットフリーランタイムのレジスタ	218
16 ビットフリーランタイム割込み	272
32 ビット 16 ビットバスコンバータ	28
TMRLR レジスタ (16 ビットリロードレジスタ)	155
TMR レジスタ (16 ビットタイマレジスタ)	155
16 ビットデッドタイム制御レジスタ	
16 ビットデッドタイム制御レジスタ, 下位	
(DTCR1)	259

2 サイクル転送	
ステップ / ブロック転送 2 サイクル転送	487
2 サイクル転送時のデータの動き	499
32/16/8 ビット	
FR-CPU ROM モード	
(32/16/8 ビット, 読出しのみ)	513
32 ビット	
32 ビット 16 ビットバスコンバータ	28
8/10 ビット	
8/10 ビット A/D コンバータ使用上の注意	443
8/10 ビット A/D コンバータの概要	420
8/10 ビット A/D コンバータの端子	424
8/10 ビット A/D コンバータの端子の	
ブロックダイアグラム	112, 425
8/10 ビット A/D コンバータの	
ブロックダイアグラム	421
8/10 ビット A/D コンバータのレジスタ一覧	
	426
8/10 ビット A/D コンバータの割込み	437
8 ビット	
8 ビット PPG ch.0, ch.2, ch.4, ch.6 の	
ブロックダイアグラム	171
8 ビット PPG ch.1, ch.5 のブロックダイアグラム	172
8 ビット PPG ch.3, ch.7 のブロックダイアグラム	173
MB91F267A/MB91F267NA/MB91267NA/MB91267A	
MB91F267A/MB91F267NA/MB91267NA/	
MB91267A のメモリマップ	25

A

A/D	
A/D 起動	308
フリーランタイム 0 による A/D 起動	282
A/D 起動コンペア	
A/D 起動コンペアのブロックダイアグラム	215
A/D 起動コンペアのレジスタ	223
A/D コンバータ	
8/10 ビット A/D コンバータ使用上の注意	443
8/10 ビット A/D コンバータの概要	420
8/10 ビット A/D コンバータの端子	424
8/10 ビット A/D コンバータの端子の ブロックダイアグラム	112, 425
8/10 ビット A/D コンバータの ブロックダイアグラム	421
8/10 ビット A/D コンバータのレジスター一覧	426
8/10 ビット A/D コンバータの割込み	437
A/D コンペア	
A/D コンペア起動許可	308
A/D コンペア起動モード	308
A/D 制御ステータスレジスタ	
A/D 制御ステータスレジスタ (ADCS:ADCS1,ADCS2)	432
A/D チャネル制御レジスタ	
A/D チャネル制御レジスタ (ADCH:ADCH1,ADCH2)	427
A/D データレジスタ	
A/D データレジスタ (ADCD:ADCD10,ADCD11,ADCD20, ADCD21)	435
A/D トリガ制御レジスタ	
A/D トリガ制御レジスタ (ADTRGC)	233
A/D 変換	
A/D 変換データ保護機能	442
A/D モード設定レジスタ	
A/D モード設定レジスタ (ADMD:ADMD1,ADMD2)	429
ADCD	
A/D データレジスタ (ADCD:ADCD10,ADCD11,ADCD20, ADCD21)	435
ADCH	
A/D チャネル制御レジスタ (ADCH:ADCH1,ADCH2)	427
ADCOMP	
コンペアレジスタ 1,2 (ADCOMP1,ADCOMP2)	268
ADCOMP_C	
制御レジスタ 1,2 (ADCOMP_C1,ADCOMP_C2)	269
ADCS	
A/D 制御ステータスレジスタ (ADCS:ADCS1,ADCS2)	432
ADMD	
A/D モード設定レジスタ (ADMD:ADMD1,ADMD2)	429
ADTRGC	
A/D トリガ制御レジスタ (ADTRGC)	233

AF200

AF200 フラッシュマイコンプログラマシステム 構成 (横河デジタルコンピュータ製)	533
--	-----

AICR

Analog Input Control Register (AICR:AICR1,AICR2)	113
アナログ入力制御レジスタ (AICR:AICR1,AICR2)	436
Analog Input Control Register Analog Input Control Register (AICR:AICR1,AICR2)	113

C

CAN

CAN クロックプリスケアラ設定	417
Cancel Request	
ホールドリクエスト取下げ要求 (Hold Request Cancel Request)	126

CAN_TX

端子 CAN_TX のソフトウェア制御	413
---------------------	-----

CAN コントローラ

CAN コントローラ	345
------------	-----

C-CAN

C-CAN の特長	344
-----------	-----

CCR

CCR (Condition Code Register)	33
-------------------------------	----

CLKB

CPU クロック (CLKB)	75
-----------------	----

CLKP

周辺クロック (CLKP)	75
---------------	----

CLKR

CLKR : クロックソース制御レジスタ	87
----------------------	----

CLKT

外部バスクロック (CLKT)	75
-----------------	----

Condition Code Register

CCR (Condition Code Register)	33
-------------------------------	----

CPCLRBH

コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2,CPCLRBL0 ~ CPCLRBL2)	224
---	-----

CPCLRBL

コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2,CPCLRBL0 ~ CPCLRBL2)	224
---	-----

CPCLRH

コンペアクリアレジスタ (CPCLRH0 ~ CPCLRH2,CPCLRL0 ~ CPCLRL2)	225
---	-----

CPCLRL

コンペアクリアレジスタ (CPCLRH0 ~ CPCLRH2,CPCLRL0 ~ CPCLRL2)	225
---	-----

CPU

CPU	28
CPU クロック (CLKB)	75
CPU 制御	491
FR-CPU ROM モード (32/16/8 ビット, 読出しのみ)	513
FR-CPU プログラミングモード (16/8 ビット, 読出し / 書込み)	514

各 CPU ステートにおける端子状態	548
CPU インタフェース	
CPU インタフェース	345
CTBR	
CTBR : タイムベースカウンタクリア レジスタ	86
C コンパイラ	
C コンパイラ (fcc911)	550
D	
Data Direction Register	
Data Direction Register (DDR: DDR0 ~ DDR5,DDRG)	107
DDR	
Data Direction Register (DDR: DDR0 ~ DDR5,DDRG)	107
DDRG	
Data Direction Register (DDR: DDR0 ~ DDR5,DDRG)	107
DIVR	
DIVR0: 基本クロック分周設定レジスタ 0	89
DIVR1: 基本クロック分周設定レジスタ 1	91
DMA	
DMA による周辺割込みクリア	492
DMAC	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 DMAC 全体制御 レジスタ [DMACR]	481
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 コントロール/ ステータスレジスタ A[DMACA:DMACA0 ~ DMACA4]	469
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 コントロール/ ステータスレジスタ B[DMACB:DMACB0 ~ DMACB4]	473
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 転送元 / 転送先 アドレス設定レジスタ [(DMASA/DMADA):DMASA0 ~ DMASA4/ DMADA0 ~ DMADA4]	479
DMAC 割込み制御	494
DMACR	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 DMAC 全体制御 レジスタ [DMACR]	481
DMAC 全体制御レジスタ	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 DMAC 全体制御 レジスタ [DMACR]	481
DMADA	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 転送元 / 転送先 アドレス設定レジスタ [(DMASA/DMADA):DMASA0 ~ DMASA4/ DMADA0 ~ DMADA4]	479
DMASA	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 転送元 / 転送先 アドレス設定レジスタ [(DMASA/DMADA):DMASA0 ~ DMASA4/ DMADA0 ~ DMADA4]	479
DMA 転送	
スリープ中の DMA 転送	495
DSP-CSR	
DSP コントロール / ステータスレジスタ (DSP-CSR)	451
DSP-LY	
DSP 遅延レジスタ (DSP-LY)	455

DSP-OT	
DSP 変数モニタレジスタ (DSP-OT0 ~ DSP-OT7)	456
DSP-PC	
DSP プログラム・カウンタ (DSP-PC)	454
DSP コントロール / ステータスレジスタ	
DSP コントロール / ステータスレジスタ (DSP-CSR)	451
DSP 遅延レジスタ	
DSP 遅延レジスタ (DSP-LY)	455
DSP 変数モニタレジスタ	
DSP 変数モニタレジスタ (DSP-OT0 ~ DSP-OT7)	456
DTCR	
16 ビットデッドタイム制御レジスタ, 下位 (DTCR1)	259
16 ビットデッドタイム制御レジスタ, 上位 (DTCR0)	256
16 ビットデッドタイム制御レジスタ, 上位 (DTCR2)	262
DTTI	
DTTI 端子入力の動作	306
DTTI 端子ノイズキャンセル機能	307
DTTI 端子割込み	307
波形制御レジスタ 2 (SIGCR2) の DTTI ビットの 動作	307
E	
EIT	
EIT からの復帰	45
EIT の動作	55
EIT ベクタテーブル	50
EIT 要因	45
EIT 要因の優先度	53
多重 EIT 処理	53
eml911	
デバッグ (sim911,eml911,mon911)	556
F	
fasm911	
アセンブラ (fasm911)	553
fcc911	
C コンパイラ (fcc911)	550
FIFO バッファ	
FIFO バッファからの読出し	404
FIFO バッファによるメッセージ受信	403
FIFO バッファの構成	403
FLCR	
フラッシュメモリステータスレジスタ (FLCR)	509
flnk911	
リンカ (flnk911)	555
FLWC	
フラッシュウェイトレジスタ (FLWC)	511
FR-CPU	
FR-CPU ROM モード (32/16/8 ビット, 読出しのみ)	513
FR-CPU プログラミングモード (16/8 ビット, 読出し / 書込み)	514

FSR	
フリーランタイムセレクトレジスタ	
(FSR0 ~ FSR2)	270

G

GATE	
RTO0 ~ RTO5 と GATE の出力状態	296
GATEC	
GATE 機能制御レジスタ (GATEC)	180
GATE 機能制御レジスタ	
GATE 機能制御レジスタ (GATEC)	180

H

Hold Request	
ホールドリクエスト取下げ要求	
(Hold Request Cancel Request)	126
Hold Request Cancel Level register	
Hold Request Cancel Level register (HRCL)	
.....	121
HRCL	
Hold Request Cancel Level register (HRCL)	
.....	121
HRCLR	
ホールドリクエスト取下げ要求機能 (HRCLR) の	
使用例	127

I

I/O ポート	
I/O ポートのモード	105
I/O マップ	
I/O マップ	536
ICR	
ICR (Interrupt Control Register)	48
Interrupt Control Register(ICR)	120
ICSH	
インプットキャプチャ状態制御レジスタ	
(ch.2,ch.3), 上位 (ICSH23)	247
ICSL	
インプットキャプチャ状態制御レジスタ	
(ch.2,ch.3), 下位 (ICSL23)	249
ILM	
ILM	35, 47
INIT	
$\overline{\text{INIT}}$ 端子入力 (設定初期化リセット端子)	
.....	64
設定初期化リセット (INIT)	63
設定初期化リセット (INIT) 解除シーケンス	
.....	66
Interrupt Control Register	
ICR (Interrupt Control Register)	48
Interrupt Control Register(ICR)	120
IPCPH	
インプットキャプチャデータレジスタ	
(IPCPH0 ~ IPCPH3,IPCPL0 ~ IPCPL3)	
.....	246

IPCPL	
インプットキャプチャデータレジスタ	
(IPCPH0 ~ IPCPH3,IPCPL0 ~ IPCPL3)	
.....	246

I フラグ

I フラグ	47
-------------	----

J

JMP	
JMP 命令 (分岐命令)	464

L

LQFP	
LQFP-64 (MB91F267A/MB91F267NA/MB91267A/	
MB91267NA)	5

M

MAC	
MAC 命令	460
MB91267A	
LQFP-64 (MB91F267A/MB91F267NA/MB91267A/	
MB91267NA)	5
MB91267NA	
LQFP-64 (MB91F267A/MB91F267NA/MB91267A/	
MB91267NA)	5
MB91F267A	
LQFP-64 (MB91F267A/MB91F267NA/MB91267A/	
MB91267NA)	5
MB91F267NA	
LQFP-64 (MB91F267A/MB91F267NA/MB91267A/	
MB91267NA)	5
MOD	
16 ビットアウトプットコンペアの動作	
(セット / リセットモード ,MOD1x=1)	
.....	287
16 ビットアウトプットコンペアの動作	
(反転モード ,MOD1x=0)	284
mon911	
デバッグ (sim911,eml911,mon911)	556
Multiply & Divide register	
乗除算レジスタ (Multiply & Divide register)	
.....	37

N

NMI	
NMI (Non Maskable Interrupt)	125
割込み・NMI に対するレベルマスク	47
Non Maskable Interrupt	
NMI (Non Maskable Interrupt)	125

O

OCCPBH	
アウトプットコンペアバッファレジスタ	
(OCCPBH0 ~ OCCPBH5,OCCPBL0 ~	
OCCPBL5)	235

OCCPBL	
アウトプットコンペアバッファレジスタ	
(OCCPBH0 ~ OCCPBH5,OCCPBL0 ~	
OCCPBL5)	235
OCCPH	
アウトプットコンペアレジスタ	
(OCCPH0 ~ OCCPH5,OCCPL0 ~ OCCPL5)	
.....	236
OCCPL	
アウトプットコンペアレジスタ	
(OCCPH0 ~ OCCPH5,OCCPL0 ~ OCCPL5)	
.....	236
OCMOD	
コンペアモード制御レジスタ (OCMOD)	
.....	244
OCSH	
コンペア制御レジスタ, 上位	
(OCSH1,OCSH3,OCSH5)	237
OCSL	
コンペア制御レジスタ, 下位	
(OCSL0,OCSL2,OCSL4)	241
P	
PC	
PC (Program Counter)	36
PCR	
Pull-up Resistance Control Register	
(PCR: PCR0 ~ PCR4,PCRG)	108
PCRG	
Pull-up Resistance Control Register	
(PCR: PCR0 ~ PCR4,PCRG)	108
PDIVR	
分周比制御レジスタ (PDIVR:PDIVR0)	196
PDR	
Port Data Register	
(PDR: PDR0 ~ PDR5,PDRG)	106
PDRG	
Port Data Register	
(PDR: PDR0 ~ PDR5,PDRG)	106
PFR	
Port Function Register (PFR:PFR0,PFR1,PTFR0)	
.....	109
PICSH	
PPG 出力制御レジスタ, 上位 (PICSH01)	251
PICSL	
インプットキャプチャ状態制御レジスタ	
(ch.0,ch.1), 下位 (PICSL01)	253
PLL	
PLL 通倍率	72
PLL 通倍率変更後の待ち時間	73
PLL 動作許可	72
PLL 動作許可後の待ち時間	73
Port Data Register	
Port Data Register	
(PDR: PDR0 ~ PDR5,PDRG)	106
Port Function Register	
Port Function Register (PFR:PFR0,PFR1,PTFR0)	
.....	109
PPG	
8ビット PPG ch.0,ch.2,ch.4,ch.6 の	
ブロックダイアグラム	171

8ビット PPG ch.1,ch.5 のブロックダイアグラム	
.....	172
8ビット PPG ch.3,ch.7 のブロックダイアグラム	
.....	173
PPG0 出力制御	298
PPG の機能	168
ゲートトリガされた PPG0 出力	298
PPGCn レジスタ	
PPGCn レジスタ (PPGn 動作モード制御レジスタ)	
n=0,1,2,3,4,5,6,7	175
PPGn 動作モード制御レジスタ	
PPGCn レジスタ (PPGn 動作モード制御レジスタ)	
n=0,1,2,3,4,5,6,7	175
PPG 起動レジスタ	
PPG 起動レジスタ (TRG)	179
PPG 出力制御レジスタ	
PPG 出力制御レジスタ, 上位 (PICSH01)	
.....	251
PRLH	
PRL/PRLH レジスタ (リロードレジスタ:	
PRL0 ~ PRL7/PRLH0 ~ PRLH7)	178
PRL	
PRL/PRLH レジスタ (リロードレジスタ:	
PRL0 ~ PRL7/PRLH0 ~ PRLH7)	178
Program Counter	
PC (Program Counter)	36
Program Status	
PS(Program Status)	32
PS	
PS(Program Status)	32
PTFR	
Port Function Register (PFR:PFR0,PFR1,PTFR0)	
.....	109
Pull-up Resistance Control Register	
Pull-up Resistance Control Register	
(PCR: PCR0 ~ PCR4,PCRG)	108
PWC	
PWC の機能	188
PWC のブロックダイアグラム	189
PWCR	
PWC データバッファレジスタ (PWCR:PWCR0)	
.....	195
PWCSR	
PWC コントロール / ステータスレジスタ	
(PWCSR:PWCSR0)	190
PWC コントロール / ステータスレジスタ	
PWC コントロール / ステータスレジスタ	
(PWCSR:PWCSR0)	190
PWC データバッファレジスタ	
PWC データバッファレジスタ	
(PWCR:PWCR0)	195
R	
RAM	
メッセージ RAM	345
メッセージ RAM とのデータ送受信	396
Return Pointer	
RP (Return Pointer)	36
REVC	
出力反転レジスタ (REVC)	179

ROM	
FR-CPU ROM モード	
(32/16/8 ビット, 読出しのみ)	513
ROM ライタ	
ROM ライタによる書込み	503
RP	
RP (Return Pointer)	36
RSRR	
RSRR : リセット要因レジスタ / ウォッチドッグ	
タイマ制御レジスタ	79
RST	
動作初期化リセット (RST)	63
動作初期化リセット (RST) 解除シーケンス	
.....	66
RTO	
RTO0 ~ RTO5 と GATE の出力状態	296
S	
SCR	
SCR (System Condition Code Register)	34
シリアルコントロールレジスタ	
(SCR:SCR0,SCR1)	327
SIDR	
シリアルインプットデータレジスタ	
(SIDR:SIDR0,SIDR1)/	
シリアルアウトプットデータレジスタ	
(SODR:SODR0, SODR1)	329
SIGCR	
波形制御レジスタ 1 (SIGCR1)	265
波形制御レジスタ 2 (SIGCR2)	267
波形制御レジスタ 2 (SIGCR2) の DTTI ビットの	
動作	307
sim911	
デバツガ (sim911,eml911,mon911)	556
SMR	
シリアルモードレジスタ (SMR:SMR0,SMR1)	
.....	325
SODR	
シリアルインプットデータレジスタ	
(SIDR:SIDR0,SIDR1)/	
シリアルアウトプットデータレジスタ	
(SODR:SODR0, SODR1)	329
SRST	
STCR:SRST ビット書込み	
(ソフトウェアリセット)	64
SSP	
SSP (System Stack Pointer)	36, 49
SSR	
シリアルステータスレジスタ	
(SSR:SSR0,SSR1)	330
STCR	
STCR : スタンバイ制御レジスタ	81
STCR:SRST ビット書込み	
(ソフトウェアリセット)	64
STOP 状態	
STOP 状態からの復帰動作について	137
外部割込みを使用した STOP 状態からの復帰時に	
おける注意事項	136
STR	
STR 命令 (転送命令)	462

System Stack Pointer	
SSP (System Stack Pointer)	36, 49
System Condition Code Register	
SCR (System Condition Code Register)	34

T

Table Base Register	
TBR (Table Base Register)	36, 49
TBCR	
TBCR : タイムベースカウンタ制御レジスタ	
.....	84
TBR	
TBR (Table Base Register)	36, 49
TCCSH	
タイマ状態制御レジスタ, 上位	
(TCCSH0 ~ TCCSH2)	227
TCCSL	
タイマ制御レジスタ, 下位	
(TCCSL0 ~ TCCSL2)	230
TCDTH	
タイマデータレジスタ	
(TCDTH0 ~ TCDTH2,TCDTL0 ~ TCDTL2)	
.....	226
TCDTL	
タイマデータレジスタ	
(TCDTH0 ~ TCDTH2,TCDTL0 ~ TCDTL2)	
.....	226
TMCSR	
コントロールステータスレジスタ	
(TMCSR:TMCSR0 ~ TMCSR2)	152
TMRLR レジスタ	
TMRLR レジスタ	
(16 ビットリロードレジスタ)	155
TMRRH	
16 ビットデッドタイムレジスタ	
(TMRRH0 ~ TMRRH2,TMRRL0 ~ TMRRL2)	
.....	255
TMRRL	
16 ビットデッドタイムレジスタ	
(TMRRH0 ~ TMRRH2,TMRRL0 ~ TMRRL2)	
.....	255
TMR レジスタ	
TMR レジスタ (16 ビットタイムレジスタ)	
.....	155
TRG	
PPG 起動レジスタ (TRG)	179

U

UART	
UART のクロック選択	333
UART の特長	322
User Stack Pointer	
USP (User Stack Pointer)	37
USP	
USP (User Stack Pointer)	37

あ

アーキテクチャ	
内部アーキテクチャ	27
アウトプットコンペア	
16 ビットアウトプットコンペアとフリーラン タイマの動作について	289
16 ビットアウトプットコンペアの使用上の注意	310
16 ビットアウトプットコンペアの動作 (セット/リセットモード,MOD1x=1)	287
16 ビットアウトプットコンペアの動作 (反転モード,MOD1x=0)	284
16 ビットアウトプットコンペアのプログラム例	313
16 ビットアウトプットコンペアのレジスタ	220
16 ビットアウトプットコンペア用フリーラン タイマの選択	283
16 ビットアウトプットコンペア割込み	273
アウトプットコンペアタイミング	
16 ビットアウトプットコンペアタイミング	288
アウトプットコンペアバッファレジスタ	
アウトプットコンペアバッファレジスタ (OCCPBH0 ~ OCCPBH5,OCCPBL0 ~ OCCPBL5)	235
アウトプットコンペアレジスタ	
アウトプットコンペアレジスタ (OCCPH0 ~ OCCPH5,OCCPL0 ~ OCCPL5)	236
アセンブラ	
アセンブラ (fasm911)	553
アドレッシング	
ダイレクトアドレッシング領域	24, 40
アドレッシングモード	
アドレッシングモード	489
アナログ入力制御レジスタ	
アナログ入力制御レジスタ (AICR:AICR1,AICR2)	436
アルゴリズム	
自動アルゴリズム実行状態	502
アンドフロー	
アンドフロー動作	157

い

インタフェース	
CPU インタフェース	345
メッセージインタフェースレジスター一覧	348
インプットキャプチャ	
16 ビットインプットキャプチャ入力タイミング	295
16 ビットインプットキャプチャの使用上の注意	310
16 ビットインプットキャプチャの動作	295
16 ビットインプットキャプチャの ブロックダイアグラム	213
16 ビットインプットキャプチャのレジスタ	221

16 ビットインプットキャプチャ用フリーラン タイマの選択	294
16 ビットインプットキャプチャ割込み	273
インプットキャプチャ状態制御レジスタ	
インプットキャプチャ状態制御レジスタ (ch.0,ch.1), 下位 (PICSL01)	253
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 下位 (ICSL23)	249
インプットキャプチャ状態制御レジスタ (ch.2,ch.3), 上位 (ICSH23)	247
インプットキャプチャデータレジスタ	
インプットキャプチャデータレジスタ (IPCPH0 ~ IPCPH3, IPCPL0 ~ IPCPL3)	246

う

ウォッチドッグタイマ制御レジスタ	
RSRR: リセット要因レジスタ/ウォッチドッグ タイマ制御レジスタ	79
ウォッチドッグリセット	
ウォッチドッグリセット	65

え

エラー	
エラーによる停止	494
演算機能	
演算機能	458
演算結果	
演算結果の転送処理	458

お

応用例	
応用例	339
オーダリング	
バイトオーダリング	38
ビットオーダリング	38

か

外形寸法図	
パッケージ外形寸法図	6
解除シーケンス	
設定初期化リセット (INIT) 解除シーケンス	66
動作初期化リセット (RST) 解除シーケンス	66
外部	
外部バスクロック (CLKT)	75
外部カウンタクロック	
選択された外部カウンタクロック	281
外部割込み	
外部割込みを使用した STOP 状態からの復帰時に おける注意事項	136
カウンタ	
カウンタのクリア	200
カウンタの動作状態	159

カウントクロック	
カウントクロックの選択	198
選択された外部カウントクロック	281
カスケードモード	
カスケードモード	320

き

基本クロック分周設定レジスタ	
DIVR0: 基本クロック分周設定レジスタ 0	89
DIVR1: 基本クロック分周設定レジスタ 1	91
基本構成	
シリアル書込み基本構成	530
基本プログラミングモデル	
基本プログラミングモデル	31
共通事項	
共通事項	572

く

クロック	
CPU クロック (CLKB)	75
UART のクロック選択	333
クロック切り替え手順	416
クロック分周	77
原発振クロック周波数について	533
周辺クロック (CLKP)	75
内部クロック動作	156
クロック切り替え	
クロック切り替え手順	416
クロック生成制御部	
クロック生成制御部のブロックダイアグラム	78
クロックソース制御レジスタ	
CLKR: クロックソース制御レジスタ	87
クロック同期	
クロック同期モード	335
クロックプリスケアラ	
CAN クロックプリスケアラ設定	417
クロックプリスケアラレジスタ	
クロックプリスケアラレジスタ	351

け

ゲート機能	
ゲート機能のブロックダイアグラム	174
ゲートトリガ	
ゲートトリガされた PPG0 出力	298
原発振	
原発振クロック周波数について	533

こ

コマンドシーケンス	
コマンドシーケンス	515
コントロール/ステータスレジスタ	
DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 コントロール/ ステータスレジスタ A[DMACA:DMACA0 ~ DMACA4]	469

DMAC-ch.0,ch.1,ch.2,ch.3,ch.4 コントロール/ ステータスレジスタ B[DMACB:DMACB0 ~ DMACB4]	473
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 (TMCSR:TMCSR0 ~ TMCSR2)	152
コンパイラ	
C コンパイラ (fcc911)	550
コンペア	
A/D コンペア起動許可	308
A/D コンペア起動モード	308
コンペアクリアバッファ	
コンペアクリアバッファ	277
コンペアクリアバッファレジスタ	
コンペアクリアバッファレジスタ (CPCLRBH0 ~ CPCLRBH2,CPCLRBL0 ~ CPCLRBL2)	224
コンペアクリアレジスタ	
コンペアクリアレジスタ (CPCLRH0 ~ CPCLRH2,CPCLRL0 ~ CPCLRL2)	225
コンペア制御レジスタ	
コンペア制御レジスタ, 下位 (OCSL0,OCSL2,OCSL4)	241
コンペア制御レジスタ, 上位 (OCSH1,OCSH3,OCSH5)	237
コンペアモード制御レジスタ	
コンペアモード制御レジスタ (OCMOD)	244
コンペアレジスタ	
コンペアレジスタ 1,2 (ADCOMP1,ADCOMP2)	268

さ

サイレントモード	
サイレントモード	410
サイレントモードとループバックモードの 結合	412

し

シーケンス	
コマンドシーケンス	515
設定初期化リセット (INIT) 解除シーケンス	66
転送シーケンスの選択	486
動作初期化リセット (RST) 解除シーケンス	66
自動アルゴリズム	
自動アルゴリズム実行状態	502
周辺クロック	
周辺クロック (CLKP)	75
周辺割込み	
DMA による周辺割込みクリア	492
受信	
FIFO バッファによるメッセージ受信	403
受信メッセージの処理	402
データフレーム受信	400
受信メッセージ	
受信メッセージの受容フィルタ	400

受信メッセージオブジェクト	
受信メッセージオブジェクトの設定	401
受信優先度	
受信優先度	400
出力状態	
RTO0 ~ RTO5 と GATE の出力状態	296
出力制御	
PPG0 出力制御	298
出力端子	
出力端子機能	158
出力反転レジスタ	
出力反転レジスタ (REVC)	179
主要機能	
主要機能	116
受容フィルタ	
受信メッセージの受容フィルタ	400
乗除算レジスタ	
乗除算レジスタ (Multiply & Divide register)	37
シリアルアウトプットデータレジスタ	
シリアルインプットデータレジスタ	
(SIDR:SIDR0, SIDR1)/	
シリアルアウトプットデータレジスタ	
(SODR:SODR0, SODR1)	329
シリアルインプットデータレジスタ	
シリアルインプットデータレジスタ	
(SIDR:SIDR0, SIDR1)/	
シリアルアウトプットデータレジスタ	
(SODR:SODR0, SODR1)	329
シリアルオンボード書込み	
富士通標準シリアルオンボード書込みに使用する	
端子	531
シリアル書込み	
シリアル書込み基本構成	530
シリアル書込み接続例	532
シリアルコントロールレジスタ	
シリアルコントロールレジスタ	
(SCR:SCR0, SCR1)	327
シリアルステータスレジスタ	
シリアルステータスレジスタ (SSR:SSR0, SSR1)	330
シリアルモードレジスタ	
シリアルモードレジスタ (SMR:SMR0, SMR1)	325

す

スタック	
割込みスタック	49
スタンバイ制御レジスタ	
STCR: スタンバイ制御レジスタ	81
スタンバイモード	
スタンバイモード (ストップ/スリープ) からの	
復帰	126
ステップ/ブロック転送	
ステップ/ブロック転送 2 サイクル転送	487
ストップ	
スタンバイモード (ストップ/スリープ) からの	
復帰	126
ストップモード	
ストップモード復帰後の待ち時間	74

スリープ	
スタンバイモード (ストップ/スリープ) からの	
復帰	126
スリープ中の DMA 転送	495

せ

制御レジスタ	
制御レジスタ 1,2	
(ADCOMPC1, ADCOMPC2)	269
セクタ	
フラッシュメモリのセクタ構成	504
接続例	
シリアル書込み接続例	532
設定初期化後	
設定初期化後の待ち時間	73
設定初期化	
INIT 端子入力 (設定初期化リセット端子)	64
設定初期化リセット (INIT)	63
設定初期化リセット (INIT) 解除シーケンス	66
セット/リセットモード	
16 ビットアウトプットコンペアの動作	
(セット/リセットモード, MOD1x=1)	287
セットタイミング	
割込み発生およびフラグの	
セットタイミング	336
全体コントロールレジスタ	
全体コントロールレジスタ	352, 353
全体コントロールレジスタ一覧	347

そ

送受信	
メッセージ RAM とのデータ送受信	396
送信	
メッセージ送信	398
送信メッセージオブジェクト	
送信メッセージオブジェクトの更新	399
送信メッセージオブジェクトの設定	398
送信優先度	
送信優先度	398
ソースクロック	
ソースクロックの選択	71
ソフトウェア	
ソフトウェア要求	485
端子 CAN_TX のソフトウェア制御	413
ソフトウェアリセット	
STCR:SRST ビット書込み	
(ソフトウェアリセット)	64

た

退避	
退避・復帰の処理	147
タイマ	
タイマ割込み	278
タイマクリア	
タイマクリア	276

タイマ状態制御レジスタ	
タイマ状態制御レジスタ, 上位	
(TCCSH0 ~ TCCSH2)	227
タイマ制御レジスタ	
タイマ制御レジスタ, 下位 (TCCSL0 ~ TCCSL2)	
.....	230
タイマデータレジスタ	
タイマデータレジスタ (TCDTH0 ~	
TCDTH2, TCDTL0 ~ TCDTL2)	226
タイマモード	
タイマモード	276
タイマモードの動作	300
タイマレジスタ	
TMR レジスタ (16 ビットタイマレジスタ)	
.....	155
タイミング	
16 ビットインプットキャプチャ入力タイミング	
.....	295
タイミングジェネレータ	
タイミングジェネレータの動作概要	165
タイムベースカウンタ	
タイムベースカウンタ	93
タイムベースカウンタクリアレジスタ	
CTBR: タイムベースカウンタクリアレジスタ	
.....	86
タイムベースカウンタ制御レジスタ	
TBCR: タイムベースカウンタ制御	
レジスタ	84
ダイレクトアドレッシング	
ダイレクトアドレッシング領域	24, 40
多機能タイマ	
多機能タイマの構成	208
多機能タイマの端子	217
多機能タイマの動作	275
多機能タイマのブロックダイアグラム	210
多重 EIT	
多重 EIT 処理	53
端子	
8/10 ビット A/D コンバータの端子の	
ブロックダイアグラム	112
端子機能一覧	7
端子状態	
各 CPU ステートにおける端子状態	548
単発変換	
単発変換モードの動作	438

ち

遅延書込み	
遅延書込み機能	458
遅延スロット	
遅延スロット付き動作	42
遅延スロットなし動作	44
チャネル	
チャネル選択と制御	495
調歩同期	
非同期 (調歩同期) モード	334

つ

通常リセット	
通常リセット動作	69

て

停止変換	
停止変換モードの動作	441
低消費電力モード	
低消費電力モード	99
通倍率	
PLL 通倍率	72
通倍率変更後	
PLL 通倍率変更後の待ち時間	73
データ	
A/D 変換データ保護機能	442
データフレーム	
データフレーム受信	400
テストモード	
テストモード設定	410
デッドタイム制御レジスタ	
16 ビットデッドタイム制御レジスタ, 上位	
(DTCR0)	256
16 ビットデッドタイム制御レジスタ, 上位	
(DTCR2)	262
デッドタイムレジスタ	
16 ビットデッドタイムレジスタ (TMRRH0 ~	
TMRRH2, TMRRL0 ~ TMRRL2)	255
デッドタイムタイマモード	
デッドタイムタイマモード時の動作	302
デバイス	
デバイスの取扱いについて	20
デバイス状態	
デバイス状態と各遷移	96
デバッグ	
デバッグ (sim911, eml911, mon911)	556
デバッグ	
デバッグ関連の注意事項	574
デュアルオペレーションフラッシュ	
デュアルオペレーションフラッシュの特長	
.....	526
電源投入後	
電源投入後の待ち時間	73
転送	
スリープ中の DMA 転送	495
転送アドレス	484
転送シーケンスの選択	486
転送タイプ	483
転送要求の受け付けと転送	492
バースト 2 サイクル転送	486
バースト転送	498
ブロック転送	497
転送回数	
転送回数と転送終了	484
転送回数制御	
転送回数制御	490
転送終了	
転送回数と転送終了	484
転送処理	
演算結果の転送処理	458
転送命令	
STR 命令 (転送命令)	462
転送モード	
転送モード	483

転送元 / 転送先アドレス設定レジスタ	
DMAC-ch.0, ch.1, ch.2, ch.3, ch.4 転送元 / 転送先	
アドレス設定レジスタ	
[(DMASA/DMADA):DMASA0 ~ DMASA4/	
DMADA0 ~ DMADA4]	479
転送要求	
転送要求の受け付けと転送	492

と

同期	
クロック同期モード	335
同期リセット	
同期リセット動作	69
動作初期化	
動作初期化リセット (RST)	63
動作初期化リセット (RST) 解除シーケンス	66
動作モード	
動作モード	59, 333, 457
動作モードの選択	198
特長	
特長	2, 26, 45, 446
取下げ要求	
ホールドリクエスト取下げ要求	
(Hold Request Cancel Request)	126
ホールドリクエスト取下げ要求機能 (HRCR) の	
使用例	127

な

内蔵周辺要求	
内蔵周辺要求	485
内部アーキテクチャ	
内部アーキテクチャ	27
内部クロック	
内部クロック動作	156

に

入出力回路形式	
入出力回路形式	15
入力タイミング	
16 ビットインプットキャプチャ入力タイミング	295

の

ノイズキャンセル	
DTTI 端子ノイズキャンセル機能	307

は

バースト 2 サイクル転送	
バースト 2 サイクル転送	486
バースト転送	
バースト転送	498
ハードウェア	
ハードウェア構成	116, 466
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	520

ハードウェアシーケンスフラグの使用例	524
ハーバード	
ハーバード プリンストンバスコンバータ	29
バイトオーダーリング	
バイトオーダーリング	38
波形ジェネレータ	
波形ジェネレータの使用上の注意	311
波形ジェネレータ割込み	274
波形制御レジスタ	
波形制御レジスタ 1 (SIGCR1)	265
波形制御レジスタ 2 (SIGCR2)	267
波形制御レジスタ 2 (SIGCR2) の DTTI ビットの	
動作	307
バスクロック	
外部バスクロック (CLKT)	75
バスコンバータ	
32 ビット 16 ビットバスコンバータ	28
バスモード	
バスモード	59
パッケージ	
パッケージ外形寸法図	6
発振安定待ち	
発振安定待ち発生要因	67
発振安定待ち時間	
発振安定待ち時間の選択	68
バッファ	
FIFO バッファからの読み出し	404
FIFO バッファによるメッセージ受信	403
FIFO バッファの構成	403
パルス	
パルス幅測定動作詳細	200
パルス幅測定の起動と停止	199
パルス幅測定	
パルス幅測定の起動と停止	199
反転モード	
16 ビットアウトプットコンペアの動作	
(反転モード, MOD1x=0)	284
汎用レジスタ	
汎用レジスタ	32

ひ

ビットオーダーリング	
ビットオーダーリング	38
非同期	
非同期 (調歩同期) モード	334

ふ

フィルタ	
受信メッセージの受容フィルタ	400
富士通標準	
富士通標準シリアルオンボード書込みに使用する	
端子	531
復帰	
退避・復帰の処理	147
フラグ	
I フラグ	47
割込み発生およびフラグのセットタイミング	336

フラッシュウェイトレジスタ	
フラッシュウェイトレジスタ (FLWC)	511
フラッシュマイコンプログラマシステム	
AF200 フラッシュマイコンプログラマシステム	
構成 (横河デジタルコンピュータ製)	533
フラッシュメモリ	
フラッシュメモリの概要	502
フラッシュメモリのセクタ構成	504
フラッシュメモリのブロックダイアグラム	503
フラッシュメモリのレジスタ一覧	508
フラッシュメモリステータスレジスタ	
フラッシュメモリステータスレジスタ (FLCR)	509
フリーランタイム	
16 ビットアウトプットコンペアとフリーラン	
タイマの動作について	289
16 ビットアウトプットコンペア用フリーラン	
タイマの選択	283
16 ビットインプットキャプチャ用フリーラン	
タイマの選択	294
16 ビットフリーランタイムの使用上の注意	310
16 ビットフリーランタイムのプログラム例	312
16 ビットフリーランタイムのレジスタ	218
16 ビットフリーランタイム割込み	272
フリーランタイム 0 による A/D 起動	282
フリーランタイムセクタ	
フリーランタイムセクタの	
ブロックダイアグラム	216
フリーランタイムセクタレジスタ	
フリーランタイムセクタレジスタ	223
フリーランタイムセクタレジスタ	
(FSR0 ~ FSR2)	270
プリスケアラレジスタ	
プリスケアラレジスタ	352
プリンストンバスコンバータ	
ハーバード プリンストンバスコンバータ	29
プログラミングモード	
FR-CPU プログラミングモード	
(16/8 ビット, 読出し / 書込み)	514
プログラミングモデル	
基本プログラミングモデル	31
プログラム例	
16 ビットアウトプットコンペアのプログラム例	313
16 ビットフリーランタイムのプログラム例	312
プログラム・カウンタ	
DSP プログラム・カウンタ (DSP-PC)	454
ブロックサイズ	
ブロックサイズ	488
ブロック図	
ブロック図	415
ブロックダイアグラム	
16 ビットインプットキャプチャの	
ブロックダイアグラム	213
8/10 ビット A/D コンバータの端子の	
ブロックダイアグラム	112, 425

8/10 ビット A/D コンバータの	
ブロックダイアグラム	421
8 ビット PPG ch.0, ch.2, ch.4, ch.6 の	
ブロックダイアグラム	171
8 ビット PPG ch.1, ch.5 のブロックダイアグラム	172
8 ビット PPG ch.3, ch.7 のブロックダイアグラム	173
A/D 起動コンペアのブロックダイアグラム	215
PWC のブロックダイアグラム	189
クロック生成制御部のブロックダイアグラム	78
ゲート機能のブロックダイアグラム	174
多機能タイマのブロックダイアグラム	210
フラッシュメモリのブロックダイアグラム	503
フリーランタイムセクタの	
ブロックダイアグラム	216
ブロックダイアグラム	4, 119, 131, 140, 142, 163, 316, 324, 448, 468
ポート基本ブロックダイアグラム	104
リロードタイマのブロックダイアグラム	151
ブロック転送	
ブロック転送	497
分岐命令	
JMP 命令 (分岐命令)	464
分周	
クロック分周	77
分周比制御レジスタ	
分周比制御レジスタ (PDIVR:PDIVR0)	196

へ

ベーシックモード	
ベーシックモード	412
ベクタテーブル	
EIT ベクタテーブル	50
ベクタテーブル初期領域	41
変数モニタ	
変数モニタ出力	459

ほ

ポート	
ポート基本ブロックダイアグラム	104
ホールドリクエスト	
ホールドリクエスト取下げ要求	
(Hold Request Cancel Request)	126
ホールドリクエスト取下げ要求機能 (HRCR) の	
使用例	127
ポーレート	
ポーレートの計算	320
保護機能	
A/D 変換データ保護機能	442

ま

マスク

割込みマスク機能	279
待ち時間	
PLL 通倍率変更後の待ち時間	73
PLL 動作許可後の待ち時間	73
ストップモード復帰後の待ち時間	74
設定初期化後の待ち時間	73
電源投入後の待ち時間	73

め

命令

JMP 命令 (分岐命令)	464
MAC 命令	460
STR 命令 (転送命令)	462
命令一覧表	557
命令概要	29
命令定義	450
命令動作	457

メッセージ

FIFO バッファによるメッセージ受信	403
受信メッセージの受容フィルタ	400
受信メッセージの処理	402
メッセージ RAM	345

メッセージ RAM

メッセージ RAM とのデータ送受信	396
--------------------------	-----

メッセージインタフェース

メッセージインタフェースレジスタ一覧	348
--------------------------	-----

メッセージインタフェースレジスタ

メッセージインタフェースレジスタ	352
------------------------	-----

メッセージオブジェクト

受信メッセージオブジェクトの設定	401
送信メッセージオブジェクトの更新	399
送信メッセージオブジェクトの設定	398
メッセージオブジェクト	396
メッセージオブジェクトの機能	379
メッセージオブジェクトの構成	379

メッセージ送信

メッセージ送信	398
---------------	-----

メッセージハンドラ

メッセージハンドラ	345
メッセージハンドラレジスタ一覧	350

メッセージハンドラレジスタ

メッセージハンドラレジスタ	352, 385
---------------------	----------

メモリマップ

91F267A/MB91F267NA/MB91267NA/MB91267A のメモリマップ	25
メモリマップ	40

も

モード

16 ビットアウトプットコンペアの動作 (セット / リセットモード, MOD1x=1)	287
16 ビットアウトプットコンペアの動作 (反転モード, MOD1x=0)	284
A/D コンペア起動モード	308

FR-CPU ROM モード

(32/16/8 ビット, 読出しのみ)	513
----------------------------	-----

FR-CPU プログラミングモード

(16/8 ビット, 読出し / 書込み)	514
-----------------------------	-----

I/O ポートのモード

アドレッシングモード	489
------------------	-----

カスケードモード

クロック同期モード	335
-----------------	-----

サイレントモード

サイレントモードとループバックモードの結合	412
-----------------------------	-----

スタンバイモード (ストップ / スリープ) からの

復帰	126
----------	-----

タイマモード

タイマモードの動作	300
-----------------	-----

単発変換モードの動作

停止変換モードの動作	441
------------------	-----

テストモード設定

デッドタイムタイマモード時の動作	302
------------------------	-----

転送モード

動作モード	333, 457
-------------	----------

非同期 (調歩同期) モード

ベーシックモード	412
----------------	-----

モード設定

ループバックモード	411
-----------------	-----

連続変換モードの動作

変数モニタ出力	459
---------------	-----

モニタ

変数モニタ出力	459
---------------	-----

ゆ

優先順位判定

優先順位判定	122
--------------	-----

優先度

EIT 要因の優先度	53
------------------	----

り

リセット

INIT 端子入力 (設定初期化リセット端子)	64
-------------------------------	----

STCR:SRST ビット書込み

(ソフトウェアリセット)	64
--------------------	----

ウォッチドッグリセット

設定初期化リセット (INIT)	63
------------------------	----

設定初期化リセット (INIT) 解除

シーケンス	66
-------------	----

通常リセット動作

同期リセット動作	69
----------------	----

動作初期化リセット (RST)

動作初期化リセット (RST) 解除シーケンス	66
-------------------------------	----

リセット要因

リセット要因レジスタ	
------------	--

RSRR: リセット要因レジスタ / ウォッチドッグ

タイマ制御レジスタ	79
-----------------	----

リモートフレーム

リモートフレーム	401
----------------	-----

リロードタイマ

リロードタイマのブロックダイアグラム	151
--------------------------	-----

リロード動作	
リロード動作	488
リロードレジスタ	
PRL/PRH レジスタ (リロードレジスタ :	
PRL0 ~ PRL7/PRH0 ~ PRH7)	178
TMRLR レジスタ (16 ビットリロードレジスタ)	
.....	155
リンカ	
リンカ (flnk911)	555

る

ループバックモード	
サイレントモードとループバックモードの結合	
.....	412
ループバックモード	411

れ

レジスタ群	
レジスタ群	345
レジスタ設定	
レジスタ設定時の注意	469
レベルマスク	
割込み・NMI に対するレベルマスク	47
連続変換	
連続変換モードの動作	440

わ

ワードアライメント	
ワードアライメント	39
割込み	
16 ビットアウトプットコンペア割込み	
.....	273
16 ビットインプットキャプチャ割込み	
.....	273
16 ビットフリーランタイム割込み	272
8/10 ビット A/D コンバータの割込み	437
DMAC 割込み制御	494
DMA による周辺割込みクリア	492
DTTI 端子割込み	307
タイマ割込み	278
波形ジェネレータ割込み	274
割込みスタック	49
割込み発生およびフラグの	
セットタイミング	336
割込み・NMI に対するレベルマスク	47
割込みベクタ	
割込みベクタ	545
割込みマスク	
割込みマスク機能	279
割込みレベル	
割込みレベル	46

CM71-10130-4

富士通半導体デバイス・CONTROLLER MANUAL

FR60Lite

32 ビット・マイクロコントローラ

MB91265A Series

ハードウェアマニュアル

2007 年 11 月 第 4 版発行

発行 **富士通株式会社** 電子デバイス事業本部

編集 マーケティング統括部 販売戦略部
