

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

アプリケーション・ノート

μPD784038,784038Yサブシリーズ

16/8 ビット・シングルチップ・マイクロコンピュータ

ハードウェア基礎編

μPD784031	μPD784031Y
μPD784035	μPD784035Y
μPD784036	μPD784036Y
μPD784037	μPD784037Y
μPD784038	μPD784038Y
μPD78P4038	μPD78P4038Y
μPD784031(A)	
μPD784035(A)	
μPD784036(A)	

(メ モ)

目 次 要 約

第1章	概 説	...	17
第2章	タイマ/カウンタのプログラム例	...	23
第3章	シリアル・インタフェースのプログラム例	...	125
第4章	割り込み処理のプログラム例	...	229
第5章	A/Dコンバータのプログラム例	...	315
第6章	D/Aコンバータのプログラム例	...	331
第7章	PWM出力ユニットのプログラム例	...	341
第8章	イニシャライズ処理	...	349

CMOSデバイスの一般的注意事項

静電気対策（MOS全般）

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレイやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

未使用入力の処理（CMOS特有）

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性（タイミングは規定しません）を考慮すると、個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

初期化以前の状態（MOS全般）

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

EEPROMは、日本電気株式会社の商標です。

本製品のうち、外国為替および外国貿易管理法の規定により規制貨物等（または役務）に該当するものについては、日本国外に輸出する際に、同法に基づき日本国政府の輸出許可が必要です。

本資料に掲載の応用回路および回路定数は、例示的に示したものであり、量産設計を対象とするものではありません。

注意： μ PD784038YサブシリーズはI²Cバス・インタフェース回路を内蔵しています。

I²Cバス・インタフェースを使用される場合には、カスタム・コードをご発注いただく時に、事前にその旨ご申告下さい。申告に基づき、以下の特典が受けられます。

日本電気株式会社のI²Cバス対応部品をご購入いただくことにより、これらの部品をI²Cシステムに使用する実施権がフィリップス社I²C特許に基づき許諾されることとなります。ただし、これらのI²Cシステムはフィリップス社によって設定されたI²C標準規格に合致しているものとします。

Purchase of NEC I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

● **本資料の内容は予告なく変更することがありますので、最新のものであることをご確認の上ご使用ください。**

● 文書による当社の承諾なしに本資料の転載複製を禁じます。

● 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。

● 本資料に記載された回路、ソフトウェア、及びこれらに付随する情報は、半導体製品の動作例、応用例を説明するためのものです。従って、これら回路・ソフトウェア・情報をお客様の機器に使用される場合には、お客様の責任において機器設計をしてください。これらの使用に起因するお客様もしくは第三者の損害に対して、当社は一切その責を負いません。

● 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。

● 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。

標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット

特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災/防犯装置、各種安全装置、生命維持を直接の目的としない医療機器

特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等

当社製品のデータ・シート/データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

はじめに

対象者 このアプリケーション・ノートは、 μ PD784038, 784038Yサブシリーズ製品の機能を理解し、 μ PD784038, 784038Yサブシリーズ製品を用いたアプリケーション・プログラムを設計するユーザのエンジニアを対象とします。

- ・ μ PD784038サブシリーズ : μ PD784031, 784035, 784037, 784038, 78P4038, 784031(A), 784035(A), 784036(A)
- ・ μ PD784038Yサブシリーズ : μ PD784031Y, 784035Y, 784037Y, 784038Y, 78P4038Y

目的 このアプリケーション・ノートは、 μ PD784038, 784038Yサブシリーズ製品の基礎的な機能について、応用プログラムを用いてユーザに理解していただくことを目的とします。
なお、掲載のプログラムおよびハードウェア構成は例示的に示したものであり、量産設計を対象とするものではありません。

構成 このアプリケーション・ノートでは、基本的な数値演算プログラムなどについて説明しています。

応用分野 μ PD784038, 784038Yサブシリーズは1 Mバイトのプログラム・メモリ空間や高速命令実行、低電圧動作などが可能なため、次に示すような広い範囲の応用分野に適応できます。

μ PD784038サブシリーズ

- ・ LBP
- ・ オートフォーカス・カメラ
- ・ PPC
- ・ プリンタ
- ・ 電子タイプライタ
- ・ エアコン
- ・ 電子楽器
- ・ 携帯電話 など

μ PD784038Yサブシリーズ

- ・ 携帯電話
- ・ コードレス電話
- ・ AV機器 など

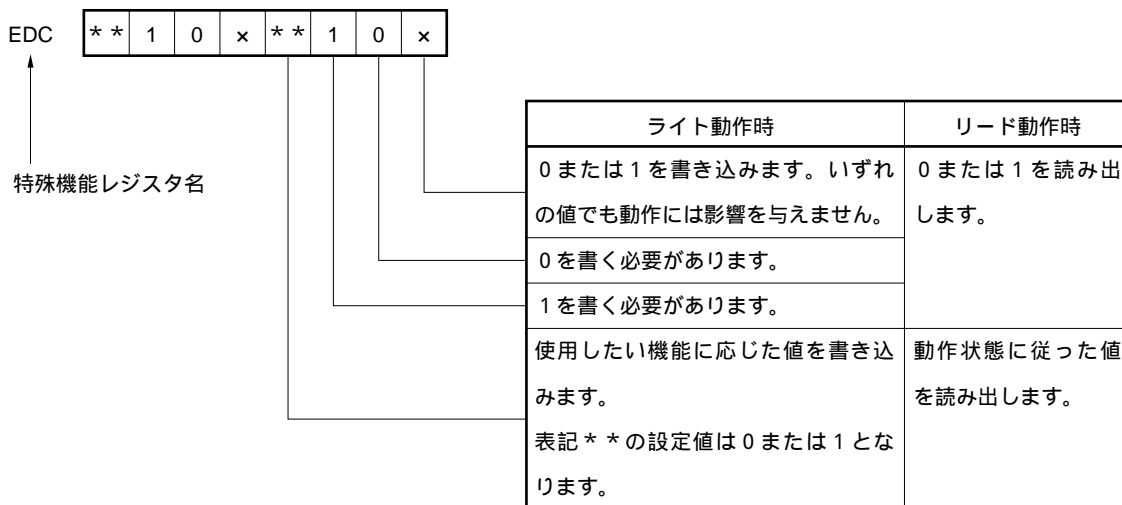
μ PD784038サブシリーズと μ PD784038Yサブシリーズの違い

μ PD784038サブシリーズと μ PD784038Yサブシリーズで機能が違うのは、クロック同期式シリアル・インタフェースだけです。その他の機能については、2つのサブシリーズは共通です。

凡 例 本アプリケーションの本文中に使用している記号，表記は次の内容を示します。

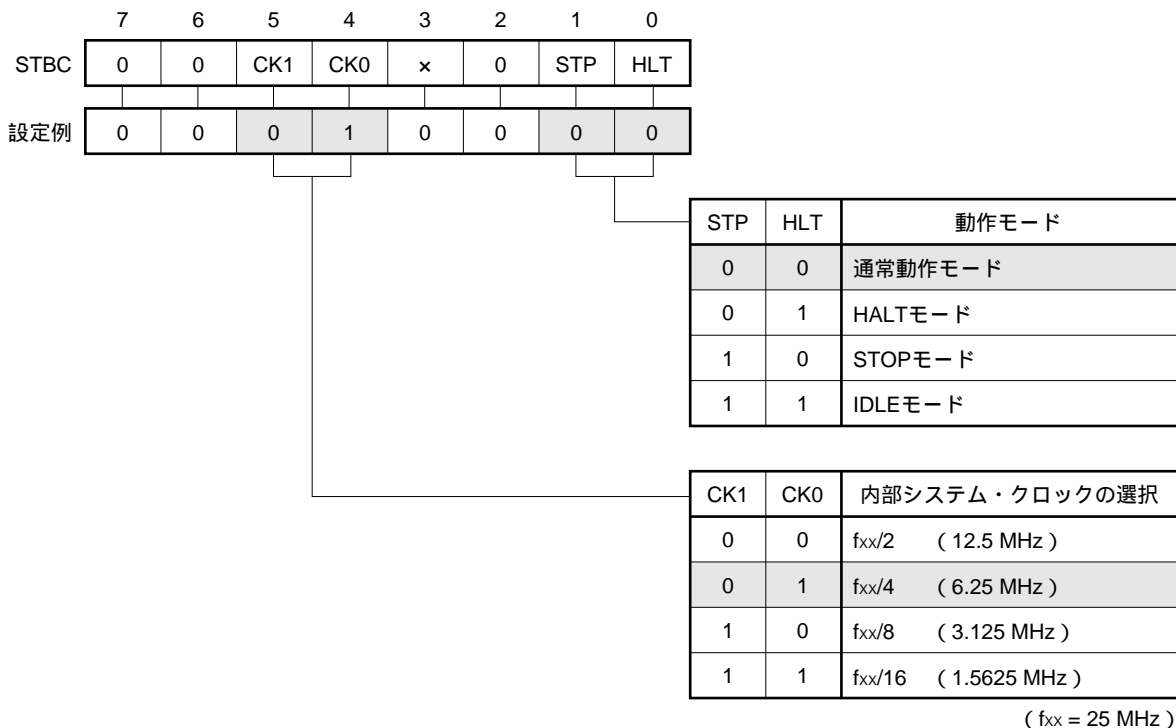
データ表記の重み	: 左側が上位桁，右側が下位桁
アクティブ・ロウの表記	: $\overline{\text{xxx}}$ (端子，信号名称に上線)
注	: 本文に付けた注の説明
注意	: 特に気をつけていただきたい内容
備考	: 本文の補足説明
数の表記	: 2進数・・・ $\text{xxx} \times \text{xx} \times \text{B}$
	: 10進数・・・ $\text{xxx} \times \text{x}$
	: 16進数・・・ $\text{xxx} \times \text{H}$
まぎらわしい文字	: 0 (ゼロ)，O (オー)
	: 1 (イチ)，l (エル)，I (アイ)

特殊機能レジスタ (SFR) 表記



本文中のレジスタ表記に『設定禁止』と書いてあるコードの組み合わせは、絶対に書き込まないでください。

レジスタ表記例



備考 このアプリケーション・ノートでは、設定するレジスタのビットに網掛け をしています。利用される時は、設定例を参考にしてください。

関連資料 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

78K/ シリーズ共通資料

資料名	資料番号	
	和文	英文
ユーザーズ・マニュアル 命令編	U10905J	U10905E
アプリケーション・ノート ソフトウェア基礎編	U10095J	U10095E
インストラクション活用表	U10594J	-
インストラクション・セット	U10595J	-
78Kシリーズ セレクション・ガイド	U10930J	U10930E
シングルチップ・マイクロコンピュータ 開発ツール セレクション・ガイド	U11069J	U11069E

個別資料

μPD784038サブシリーズ

資料名	資料番号	
	和文	英文
μPD784031 データ・シート	U11507J	U11507E
μPD784035, 784036, 784037, 784038 データ・シート	U10847J	U10847E
μPD78P4038 データ・シート	U10848J	U10848E
μPD784038サブシリーズ 特殊機能レジスタ活用表	U11090J	-
μPD784038, 784038Yサブシリーズ ユーザーズ・マニュアル ハードウェア編	U11316J	U11316E
μPD784038, 784038Yサブシリーズ アプリケーション・ノート ハードウェア基礎編	このマニュアル	作成予定
μPD784031(A) データ・シート	U13009J	U13009E
μPD784035(A), 784036(A) データ・シート	U13010J	U13010E

μPD784038Yサブシリーズ

資料名	資料番号	
	和文	英文
μPD784031Y データ・シート	U11504J	U11504E
μPD784035Y, 784036Y, 784037Y, 784038Y データ・シート	U10741J	U10741E
μPD78P4038Y データ・シート	U10742J	U10742E
μPD784038Yサブシリーズ 特殊機能レジスタ活用表	U11091J	-
μPD784038, 784038Yサブシリーズ ユーザーズ・マニュアル ハードウェア編	U11316J	U11316E
μPD784038, 784038Yサブシリーズ アプリケーション・ノート ハードウェア基礎編	このマニュアル	作成予定

注意 上記関連資料は、予告なしに内容を変更することがあります。設計などには、必ず最新の資料をご使用ください。

目 次

第1章 概 説 ...	17
1.1 本書の見方 ...	17
1.2 アプリケーション・プログラムの利用方法 ...	19
1.3 μ PD784038, 784038Yサブシリーズ製品の特徴 ...	21
第2章 タイマ/カウンタのプログラム例 ...	23
2.1 タイマ/カウンタ機能概要 ...	23
2.2 内部インターバル・タイマの応用 ...	24
2.3 プログラマブル矩形波出力の応用 ...	33
2.4 フリーランニング・インターバル・タイマの応用 ...	42
2.5 PWM出力の応用 ...	57
2.6 PPG出力の応用 ...	69
2.7 ソフト・トリガド・ワンショット・パルス出力の応用 ...	82
2.8 ワンショット・タイマ機能応用 ...	93
2.9 パルス周期測定 ...	103
2.10 外部イベント・カウンタ ...	115
第3章 シリアル・インタフェースのプログラム例 ...	125
3.1 シリアル・インタフェース機能概要 ...	125
3.2 アシンクロナス・シリアル・インタフェース・モード ...	129
3.3 アシンクロナス・シリアル・インタフェースのプログラム例 ...	134
3.4 クロック同期式シリアル・インタフェースのプログラム例 ...	149
3.4.1 3線式シリアルI/Oモードのプログラム例 ...	154
3.4.2 I ² Cバス・モードのプログラム例 (μ PD784038Yサブシリーズのみ) ...	179
第4章 割り込み処理のプログラム例 ...	229
4.1 割り込み機能概要 ...	229
4.1.1 ベクタ割り込み ...	235
4.1.2 コンテキスト・スイッチング機能 ...	236
4.1.3 マクロ・サービス機能 ...	239
4.2 ベクタ割り込みとコンテキスト・スイッチングのプログラム例 ...	260
4.3 マクロ・サービスのプログラム例 ...	273
4.3.1 マクロ・サービス・タイプAのプログラム例 ...	273
4.3.2 マクロ・サービス・タイプBのプログラム例 ...	285
4.3.3 マクロ・サービス・タイプCのプログラム例 ...	296

第5章	A/Dコンバータのプログラム例	...	315
5.1	A/Dコンバータ機能概要	...	315
5.2	A/Dコンバータのプログラム例	...	316
第6章	D/Aコンバータのプログラム例	...	331
6.1	D/Aコンバータ機能概要	...	331
6.2	D/Aコンバータのプログラム例	...	332
第7章	PWM出力ユニットのプログラム例	...	341
7.1	PWM出力ユニットの機能概要	...	341
7.2	PWM出力ユニットのプログラム例	...	344
第8章	イニシャライズ処理	...	349
8.1	割り込みベクタ・アドレスの設定	...	350
8.2	SP (スタック・ポインタ) / RB (レジスタ・バンク) / STBC (CPUクロック) のイニシャライズ	...	351
8.3	使用しない周辺機能のイニシャライズ処理	...	352
8.4	RAMクリア処理	...	353
8.5	タイマのイニシャライズ処理	...	354
8.6	ウォッチドッグ・タイマの設定	...	355
8.7	コンテキスト・スイッチングのイニシャライズ処理	...	355
8.8	割り込みのイニシャライズ処理	...	356

図の目次 (1/3)

図番号	タイトル, ページ
1 - 1	クロック発生回路のブロック図 ... 19
1 - 2	電源電圧およびクロック・サイクル・タイム (参考) ... 20
2 - 1	INTC01割り込み要求を発生させるインターバル・タイマ ... 25
2 - 2	INTC01割り込み要求を発生させるインターバル・タイマのタイミング・チャート ... 25
2 - 3	TO1端子からの矩形波出力 ... 34
2 - 4	TO1端子からの矩形波出力のタイミング・チャート ... 34
2 - 5	TO0, TO1端子からのタイマ出力 ... 43
2 - 6	TO0, TO1端子からのタイマ出力のタイミング・チャート ... 44
2 - 7	TO0端子からのPWM出力 ... 57
2 - 8	TO0端子からのPWM出力のタイミング・チャート ... 58
2 - 9	TO0端子からのPPG出力 ... 69
2 - 10	TO0端子からのPPG出力のタイミング・チャート ... 70
2 - 11	PPG出力タイミング (TM0) ... 78
2 - 12	TO1端子からの矩形波出力 ... 83
2 - 13	TO0端子からのワンショット・パルス出力例 ... 83
2 - 14	ワンショット・タイマ動作 ... 94
2 - 15	パルス周期測定のブロック図 ... 104
2 - 16	INTP2入力のパルス幅測定のタイミング・チャート ... 105
2 - 17	サンプリング・クロック選択レジスタ (SCS0) の設定 ... 116
2 - 18	INTP0端子から入力されるクロックをカウントする外部イベント・カウンタ ... 116
2 - 19	タイマ / カウンタ 1 の外部イベント・カウント・タイミング ... 117
3 - 1	シリアル・インタフェース例 ... 126
3 - 2	アシンクロナス・シリアル・インタフェース・モードのシステム構成例 ... 129
3 - 3	アシンクロナス・シリアル・インタフェースの送受信データのフォーマット ... 130
3 - 4	送信時の動作タイミング (ストップ・ビット長 = 1) ... 131
3 - 5	受信時の動作タイミング ... 131
3 - 6	受信エラーのタイミング ... 133
3 - 7	μPD784038とパソコンとの接続 ... 134
3 - 8	3線式シリアルI/Oモードのシステム構成例 ... 149
3 - 9	3線式シリアルI/Oモード (送信時) のタイミング ... 150
3 - 10	3線式シリアルI/Oモード (受信時) のタイミング ... 151
3 - 11	3線式シリアルI/Oモード (送受信時) のタイミング ... 152
3 - 12	マスタ, スレーブとの接続図 ... 154
3 - 13	マスタ側から見たタイミング・チャート ... 155

図の目次 (2/3)

図番号	タイトル, ページ
3 - 14	I ² Cバスのシリアル・データ転送タイミング ... 179
3 - 15	スタート・コンディション ... 180
3 - 16	アドレス ... 180
3 - 17	転送方向指定 ... 181
3 - 18	アクノリッジ信号 ... 182
3 - 19	ストップ・コンディション ... 183
3 - 20	ウエイト信号 ... 184
3 - 21	マスタとスレーブの接続図 ... 188
3 - 22	I ² Cバス・モードのプログラム例のタイミング・チャート ... 189
3 - 23	I ² Cバス・モードのプログラム例のタイミング・チャート (スタート・コンディション アドレスの送受信) ... 190
3 - 24	I ² Cバス・モードのプログラム例のタイミング・チャート (データの送受信: マスタ スレーブ) ... 191
3 - 25	I ² Cバス・モードのプログラム例のタイミング・チャート (データの送受信: スタート・コンディション) ... 192
3 - 26	I ² Cバス・モードのプログラム例のタイミング・チャート (アドレスの送受信: マスタ受信時) ... 193
3 - 27	I ² Cバス・モードのプログラム例のタイミング・チャート (データの送受信: スレーブ マスタ) ... 194
3 - 28	I ² Cバス・モードのプログラム例のタイミング・チャート (データの送受信 ストップ・コンディション) ... 195
4 - 1	割り込み優先順位指定フラグの指定方法 ... 233
4 - 2	割り込み処理中に他の割り込みが発生した場合の処理例 ... 234
4 - 3	割り込み要求の発生によるコンテキスト・スイッチング動作 ... 237
4 - 4	RETCS命令によるコンテキスト・スイッチング機能を使用した割り込みからの復帰動作 ... 238
4 - 5	マクロ・サービスの概要図 ... 239
4 - 6	マクロ・サービス・コントロール・ワードのアドレス (LOCATION 0の場合) ... 242
4 - 7	マクロ・サービス・モード・レジスタのフォーマット ... 244
4 - 8	マクロ・サービス・タイプAのマクロ・サービス・チャンネルの配置 ... 247
4 - 9	マクロ・サービス・タイプBのマクロ・サービス・チャンネルの配置 ... 248
4 - 10	マクロ・サービス・タイプCのマクロ・サービス・チャンネルの配置 ... 250
4 - 11	VCIEフラグが0の場合のマクロ・サービス終了時の動作 (アシンクロナス・シリアル・インタフェースによるデータの受信例) ... 253
4 - 12	VCIEフラグが1の場合のマクロ・サービス終了時の動作 (アシンクロナス・シリアル・インタフェースによるデータの送信例) ... 254

図の目次 (3/3)

図番号	タイトル, ページ
4 - 13	マクロ・サービスの基本処理シーケンス ... 255
4 - 14	マクロ・サービス・タイプAのデータ転送処理フロー ... 256
4 - 15	マクロ・サービス・タイプBのデータ転送処理フロー ... 257
4 - 16	マクロ・サービス・タイプCのデータ転送処理フロー ... 258
4 - 17	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図(マクロ・サービス・タイプA) ... 275
4 - 18	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図(マクロ・サービス・タイプB) ... 286
4 - 19	本プログラムのタイミング・チャート(1相励磁) ... 297
4 - 20	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図(マクロ・サービス・タイプC) ... 299
5 - 1	A/Dコンバータのブロック図 ... 315
5 - 2	A/Dコンバータのプログラム例の使用RAM領域 ... 317
6 - 1	D/Aコンバータのブロック図 ... 331
6 - 2	サイン出力波形 ... 332
6 - 3	サイン波形変換回路 ... 334
7 - 1	PWM出力機能のブロック図 ... 341
7 - 2	PWM出力の基本動作 ... 342
7 - 3	PWM出力タイミング例(PWMパルス幅書き換え周期 $2^{12}/f_{PwMC}$) ... 343

表の目次

表番号	タイトル, ページ
2 - 1	タイマ / カウンタの動作 ... 23
2 - 2	TM0によるPWM出力のプログラムで用いるワーク / フラグ ... 59
2 - 3	TM0によるPPG出力のプログラムで用いるワーク / フラグ ... 71
2 - 4	パルス周期測定 of プログラムで用いるワーク・エリア ... 106
2 - 5	外部イベント・カウンタ機能のプログラムで用いるRAM ... 117
3 - 1	μPD784038, 784038Yサブシリーズのシリアル・インタフェース機能一覧 ... 125
3 - 2	UART/IOE1とUART2/IOE2の名称の違い ... 128
3 - 3	受信エラーの要因 ... 133
3 - 4	UARTのプログラム例の仕様 ... 135
3 - 5	アシンクロナス・シリアル・インタフェースのプログラム例で使用するRAM/フラグ ... 135
3 - 6	3線式シリアル・インタフェースのプログラム例 (マスタ側) で使用するRAM/フラグ ... 155
3 - 7	3線式シリアル・インタフェースのプログラム例 (スレーブ側) で使用するRAM/フラグ ... 156
3 - 8	INTCSI発生タイミングおよびウエイト制御 ... 186
3 - 9	I ² Cバスのプログラム例 (マスタ側) で使用するRAM/フラグ ... 196
3 - 10	I ² Cバスのプログラム例 (スレーブ側) で使用するRAM/フラグ ... 196
4 - 1	割り込み要求の処理形態 ... 229
4 - 2	割り込み要求タイプ別の割り込み機能の特徴 ... 230
4 - 3	μPD784038, 784038Yサブシリーズのマスカブル割り込みのデフォルト優先順位 ... 232
4 - 4	μPD784038, 784038Yサブシリーズの各割り込みに対するベクタ・テーブル・アドレス ... 235
4 - 5	マクロ・サービス・タイプ別の機能の特徴 ... 241
4 - 6	マクロ・サービス・タイプAのマクロ・サービス・チャンネル構成 ... 246
4 - 7	マクロ・サービス・タイプBのマクロ・サービス・チャンネル構成 ... 248
4 - 8	マクロ・サービス・タイプCのマクロ・サービス・チャンネル構成 ... 249
4 - 9	ベクタ割り込みとコンテキスト・スイッチング機能使用時のプログラム処理の違い ... 260
4 - 10	ベクタ割り込み, コンテキスト・スイッチング機能使用時のプログラムで用いるRAM/フラグ ... 261
4 - 11	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定 ... 274
4 - 12	マクロ・サービス・タイプAのプログラムで用いるRAM領域 ... 276
4 - 13	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定 ... 285
4 - 14	マクロ・サービス・タイプBのプログラムで用いるRAM領域 ... 287
4 - 15	本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定 ... 298
5 - 1	A/Dコンバータのプログラム例で使用するRAM領域 ... 319
6 - 1	サイン出力波形の電圧, 設定値 (AV _{REF2} = 5 V, AV _{REF3} = 0 V) ... 333
6 - 2	D/Aコンバータのプログラム例で使用するRAM領域 ... 335

第1章 概 説

1.1 本書の見方

本書では、 μ PD784038, 784038Yサブシリーズに標準的に内蔵されている周辺ハードウェア（タイマ、割り込み機能、A/Dコンバータ、スタンバイ機能など）を各章に分類し、各々の章で基本的なプログラムを使用例として示し、ユーザ・プログラムとして参考活用していただけるようにアルゴリズムの解説をしています。項目は次のとおりです。

機能概要

内蔵周辺機能の機能概要を、ブロック図などを使用し説明します。

プログラム例

（1）処理概要

- （a）仕様 : プログラム処理の仕様を説明します。
- （b）使用周辺機能説明 : 仕様を実現させるために使用する周辺機能の説明をします。

（2）使用RAM領域

プログラム中で使用しているRAM領域を説明します。
ただし、使用RAM領域がワーク・エリアの場合は、プログラム実行後の内容は不定となります。

（3）使用レジスタ

- （a）汎用レジスタ : プログラム中で使用している汎用レジスタを示します。すでに別のプログラムで使用しているレジスタの内容を破壊したくない場合は、このプログラムを実行する前にレジスタ・バンク切り替えなどであらかじめレジスタの内容を退避する必要があります。
- （b）特殊機能レジスタ^注 : 特殊機能レジスタ（SFR）の設定例を示します。

注 特殊機能レジスタ : 内蔵周辺ハードウェアのモード・レジスタ、コントロール・レジスタなどの特別な機能が割り付けられたレジスタ

(4) 入力方法

内蔵周辺ハードウェアを動作させる際、必要となる入力の引数を説明します。

プログラム例は、リロケータブル・プログラムで掲載しています。ユーザ・プログラムで利用される場合は、EQU疑似命令でパラメータの定義と、パブリック宣言をしてください。

(5) 出力方法

プログラムを実行したあとの出力の引数を説明します。

(6) プログラム説明

プログラムのアルゴリズムを示します。

フロー・チャートおよびプログラム・リストをあわせて参照してください。

(7) フロー・チャート

プログラムのアルゴリズムをフロー・チャートで示します。

(8) プログラム・リスト

プログラム・リストを示します。

プログラム・リストはすべてソース・プログラムで記載してあります。実際に配置されるアドレスはリンク条件によって異なります。

<p>このアプリケーション・ノートのプログラムは例題であり、そのプログラムの動作を保障するものではありません。</p>

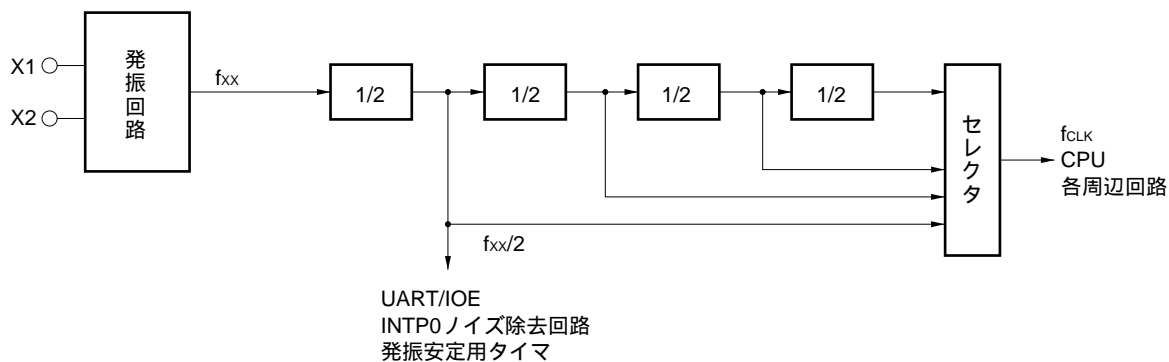
1.2 アプリケーション・プログラムの利用方法

μPD784038, 784038Yサブシリーズでは、内部クロック周波数 (f_{xx}) を4 ~ 32 MHzの範囲で選択し、供給することが可能です (発振周波数、または外部クロック周波数)。

UART/IOE, ノイズ除去回路, タイマ/カウンタ, 発振安定用タイマには内部クロック周波数 (f_{xx}) を1/2分周したクロック周波数 ($f_{xx}/2$) が基準クロックとして供給されます。

また、CPUおよびその他の周辺ハードウェアには、クロック周波数 (f_{xx}) を分周回路にて1/2, 1/4, 1/8, 1/16に分周したクロックをセレクトにより内部システム・クロック周波数 (f_{CLK}) として選択します。

図1 - 1 クロック発生回路のブロック図



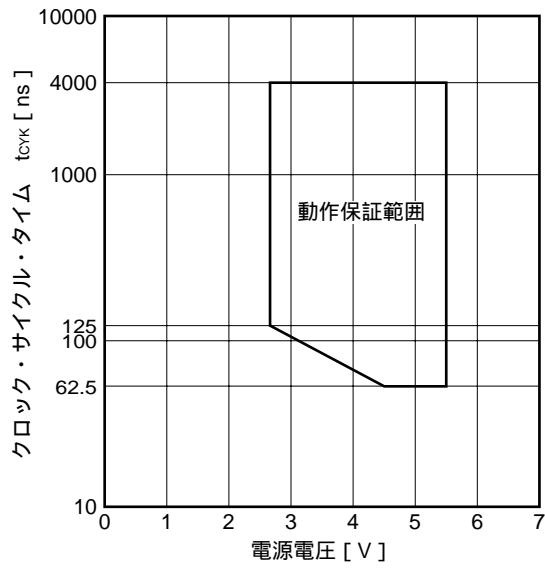
備考 f_{xx} : 発振周波数または外部クロック入力

f_{CLK} : 内部システム・クロック周波数

本編で示すアプリケーション・プログラム例は、内部システム・クロック周波数 (f_{CLK}) を12.5 MHz、動作電圧を5 Vにて設計しています。

実際にプログラムを設計する場合は、システムの動作電圧を考慮し、内部システム・クロック周波数を選択してください。

図1 - 2 電源電圧およびクロック・サイクル・タイム (参考)



例

- ・ 供給クロック32 MHz, 1/2分周を選択した場合
 $1 / (32 * (1/2)) * 1000 = 62.5 (ns)$
- ・ 供給クロック 4 MHz, 1/16分周を選択した場合
 $1 / (4 * (1/16)) * 1000 = 4000 (ns)$

1.3 μ PD784038, 784038Yサブシリーズ製品の特徴

μ PD78234サブシリーズ, μ PD784026サブシリーズとピン・コンパチブル
 μ PD78234サブシリーズ, μ PD784026サブシリーズの内部メモリ容量を拡張
 高速命令実行

・最小命令実行時間 (32 MHz動作時) : 125 ns/250 ns/500 ns/1000 ns

制御用途に適する命令セット

データ・メモリ拡張機能 (メモリ空間1 Mバイト : バンク指定ポインタ \times 2)

割り込みコントローラ (4レベル・プライオリティ)

・ベクタ割り込み処理 / マクロ・サービス / コンテキスト・スイッチング

スタンバイ機能

・HALT/STOP/IDLEモード

内部メモリとI/Oポート

項目		品名	μ PD784031	μ PD784035	μ PD784036	μ PD784037	μ PD784038	μ PD78P4038
			μ PD784031(A) μ PD784031Y	μ PD784035(A) μ PD784035Y	μ PD784036(A) μ PD784036Y			
内部ROM		なし	48 Kバイト (マスクROM)	64 Kバイト (マスクROM)	96 Kバイト (マスクROM)	128 Kバイト (マスクROM)	128 Kバイト (ワンタイムPROM, またはEPROM)	
内部RAM		2048バイト			3584バイト	4352バイト		
I/Oポート	合計	46本	64本					
	入力	8本	8本					
	入出力	34本	56本					
	出力	4本	0本					
付加機能注 付き端子	プルアップ抵抗付き端子	32本	54本					
	LEDダイレクト・ドライブ出力	8本	24本					
	トランジスタ・ダイレクト・ドライブ	8本	8本					

注 付加機能付き端子は、I/O端子の中に含まれています。

備考 μ PD784031(A), 784035(A), 784036(A) は、 μ PD784031, 784035, 784036 (標準品) に比べてより厳しい品質保証プログラムを適用している製品です (NECではこれらを品質水準の分類において、特別水準と称しています)。

シリアル・インタフェース

- ・ UART/IOE (ボー・レート・ジェネレータ内蔵) : 2チャンネル
- ・ CSI (3線式シリアルI/O, 2線式シリアルI/O, I²Cバス^注) : 1チャンネル

注 μ PD784038Yサブシリーズのみ

リアルタイム出力ポート (タイマ/カウンタと組み合わせることにより, 2系統のステッピング・モータの独立制御が可能)

8ビットA/Dコンバータ (アナログ8入力)

8ビットD/Aコンバータ (アナログ2出力)

12ビットPWM出力 (2出力)

高機能タイマ/カウンタ

- ・ タイマ/カウンタ (16ビット) \times 3ユニット
- ・ タイマ (16ビット) \times 1ユニット

ウォッチドッグ・タイマ : 1チャンネル

クロック出力機能 : fCLK, fCLK/2, fCLK/4, fCLK/8, fCLK/16から選択 (μ PD784031, 784031Y以外)

第2章 タイマ/カウンタのプログラム例

2.1 タイマ/カウンタ機能概要

μPD784038, 784038Yサブシリーズは、タイマ/カウンタを3ユニット、タイマを1ユニット内蔵しています。

表2 - 1 タイマ/カウンタの動作

項 目		名 称	タイマ/カウンタ0	タイマ/カウンタ1	タイマ/カウンタ2	タイマ3
カウント幅	8ビット		-			
	16ビット					
動作モード	インターバル・タイマ		2ch	2ch	2ch	1ch
	外部イベント・カウンタ					-
	ワンショット・タイマ		-	-		-
機 能	タイマ出力		2ch	-	2ch	-
	トグル出力			-		-
	PWM/PPG出力			-		-
	ワンショット・パルス出力 ^注			-	-	-
	リアルタイム出力		-		-	-
	パルス幅測定		1入力	1入力	2入力	-
	割り込み要求数		2	2	2	1

注 ワンショット・パルス出力機能とは、ソフトウェアによりパルス出力レベルをアクティブにし、ハードウェア（割り込み要求信号）によりインアクティブにする機能です。

この機能は、タイマ/カウンタ2のワンショット・タイマ機能とは性質が異なります。

本章では、これらのタイマ/カウンタ機能の説明として次に示すプログラム例を紹介します。

- ・ 2.2 内部インターバル・タイマの応用
- ・ 2.3 プログラマブル矩形波出力の応用
- ・ 2.4 フリーランニング・インターバル・タイマの応用
- ・ 2.5 PWM出力の応用
- ・ 2.6 PPG出力の応用
- ・ 2.7 ソフト・トリガド・ワンショット・パルス出力の応用
- ・ 2.8 ワンショット・タイマ機能応用
- ・ 2.9 パルス周期測定
- ・ 2.10 外部イベント・カウンタ

2.2 内部インターバル・タイマの応用

インターバル・タイマは、タイマ/カウンタのコンペア動作の応用により実現します。

(1) 処理概要

(a) 仕様

インターバル時間を10 msとして、タイマ/カウンタ0のタイマ割り込み (INTC01) を発生させる場合のプログラム例を紹介します。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32
- ・ 入力パラメータ (INTVL1) : 下記計算式参照

$$\begin{aligned}
 \text{INTVL1} &= \frac{\text{インターバル時間}}{\text{カウント・クロック}} \\
 &= \frac{10 \times 10^{-3}}{32 / (25 \times 10^6)} \\
 &= 7813 \text{ (小数点以下四捨五入)}
 \end{aligned}$$

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

タイマ/カウンタ0は、コンペア・レジスタの設定値とタイマ・レジスタのカウント値とを比較するコンペア動作を行います。

あらかじめ設定されたコンペア・レジスタ (CR00, CR01) の値と、タイマ・レジスタ0 (TM0) のカウント値が一致すると、出力制御回路に一致信号を送るとともに、同時に割り込み要求 (INTC00, INTC01) を発生します。また、コンペア・レジスタ (CR01) の設定値とタイマ・レジスタ0 (TM0) のカウント値の一致時にカウント値をクリアすることができ、コンペア・レジスタ (CR01) の設定値を繰り返しカウントするインターバル・タイマとして動作します。

図2 - 1に、インターバル・タイマ機能のブロック図を示します。

図2 - 1 INTC01割り込み要求を発生させるインターバル・タイマ

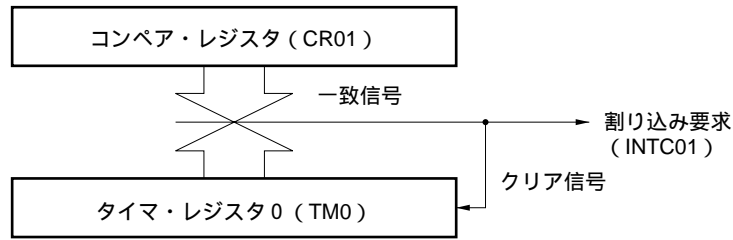
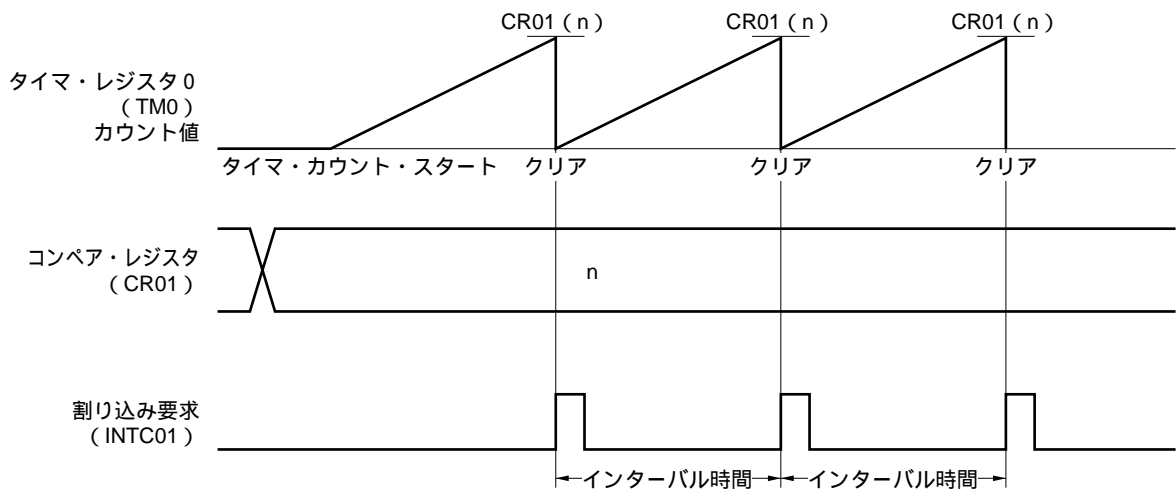


図2 - 2 INTC01割り込み要求を発生させるインターバル・タイマのタイミング・チャート



備考 インターバル時間 = $(n + 1) \times x / f_{xx}$ { n: 0 n FFFFH }

(f_{xx} = クリスタル / セラミック発振周波数)

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル / セラミック発振周波数の分周比)

INTC01割り込み要求はタイマ・レジスタ0 (TM0) のカウント値とコンペア・レジスタ (CR01) の設定値の一致によって発生します。インターバル・タイマとして、INTC01割り込み要求を発生させるために、一致時にタイマ・レジスタ0 (TM0) が、自動的にクリアされるように設定します。

また、タイマ/カウンタ0には2つのコンペア・レジスタ (CR00, CR01) が存在しますが、タイマ・レジスタ0 (TM0) のクリア機能はCR01にしかありません。したがって、任意のインターバル時間を設定するためには、通常のコンペア・レジスタ (CR01) にインターバル時間を設定すると、タイマ・レジスタ0 (TM0) のクリア動作も可能です。

(2) 使用RAM領域

なし

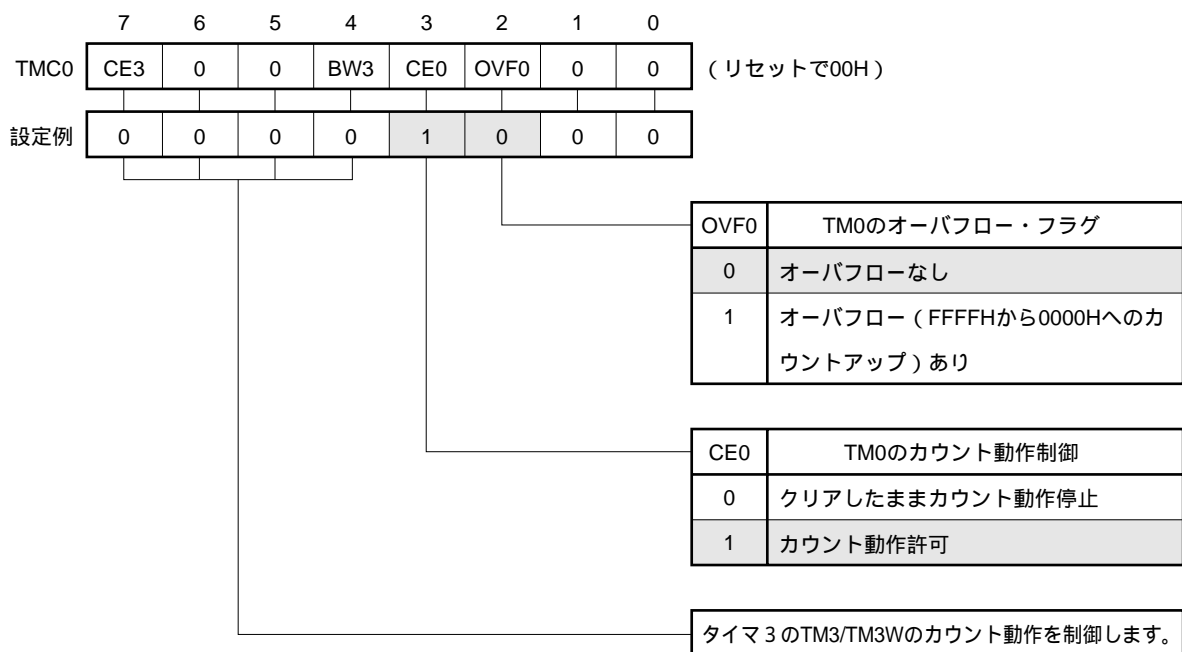
(3) 使用レジスタ

(a) 汎用レジスタ

なし

(b) 特殊機能レジスタ

タイマ・コントロール・レジスタ0 (TMC0)

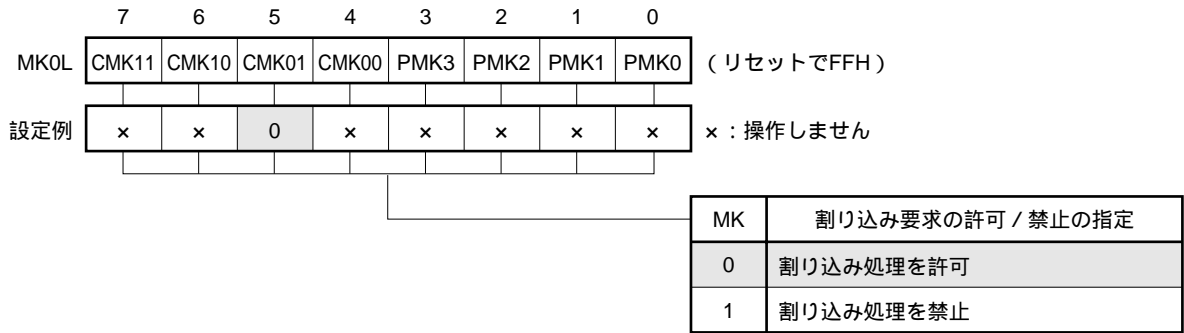


備考 OVF0ビットはソフトウェアでのみリセットされます。

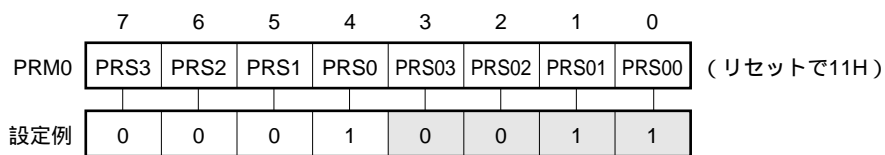
キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)



割り込みマスク・レジスタL (MK0L)



プリスケアラ・モード・レジスタ0 (PRM0)



($f_{xx} = 25 \text{ MHz}$)

PRS03	PRS02	PRS01	PRS00	タイマ/カウンタ0のTM0のカウンタ・クロック指定	
				カウンタ・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	$f_{xx}/8$	0.32 μs
0	0	1	0	$f_{xx}/16$	0.64 μs
0	0	1	1	$f_{xx}/32$	1.28 μs
0	1	0	0	$f_{xx}/64$	2.56 μs
0	1	0	1	$f_{xx}/128$	5.12 μs
0	1	1	0	$f_{xx}/256$	10.2 μs
0	1	1	1	$f_{xx}/512$	20.5 μs
1	0	0	0	$f_{xx}/1024$	41.0 μs
1	0	0	1	$f_{xx}/2048$	81.9 μs
1	1	1	1	外部クロック (INTP3)	-
上記以外				設定禁止	

タイマ3のTM3/TM3Wのカウンタ・クロックを指定します。

備考 f_{xx} : X1入力周波数, または発振周波数

(4) 入力方法

下記のレーベルに次の内容を定義します。

INTVL1 : INTC01割り込み要求を発生させるインターバル時間を定義します。

定義した値は、コンペア・レジスタ (CR01) の設定値を示します。

タイマ/カウンタ0のカウント・クロックは $f_{xx}/8$ から $f_{xx}/2048$ までプリスケアラ・モード・レジスタ0 (PRM0) により指定します。コンペア・レジスタ (CR01) への設定値 (INTVL1) であるインターバル時間は次式で求められます。

$$\text{インターバル時間} = \text{INTVL1} \times (8/f_{xx} \sim 2048/f_{xx})$$

{INTVL1 : 0 INTVL1 FFFFH}

(f_{xx} = クリスタル/セラミック発振周波数)

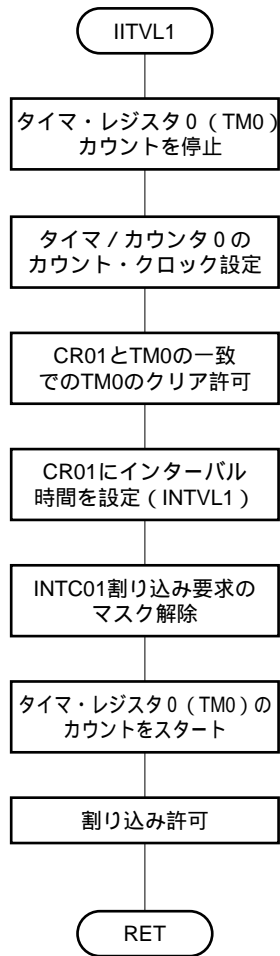
(5) 出力方法

なし

(6) プログラム説明 [レーベル名称 : IITVL1]

- (a) タイマ・レジスタ0 (TM0) のカウント動作を停止します。
- (b) タイマ/カウンタ0のカウント・クロックを $f_{xx}/32$ に指定します。
- (c) タイマ・レジスタ0 (TM0) とコンペア・レジスタ (CR01) の一致によるタイマ・レジスタ0 (TM0) のクリアを許可します。
- (d) コンペア・レジスタ (CR01) にインターバル時間 (INTVL1) を設定します。
- (e) INTC01割り込み要求のマスクを解除します。
- (f) タイマ・レジスタ0 (TM0) のカウント動作を許可します。
- (g) 割り込みを許可 (EI) します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

INTVL1 : INTC01割り込み要求を発生させるインターバル時間を定義

メイン・ルーチンのプログラム・リスト記述例

```

PUBLIC INTVL1      ; PARAMETER
EXTRN IITVL1      ; PACKAGE
.
.
INTVL1 EQU 7813    ; PARAMETER FOR INTERVAL
.
.
CALL! IITVL1
.
.

```

備考 上記記述例のようにRAM領域 (INTVL1) にインターバル時間を定義し、サブルーチンを呼んでください。

本アプリケーション・ルーチンのプログラム・リスト

```

NAME IITV1M
;
;*****
; * 16bit-Timer / Counter Unit *
; * internal interval timer *
;*****
;
;
PUBLIC IITVL1
EXTRN INTVL1 ; Compare data for interval timer
;
;
CSEG
IITVL1:
MOV TMC0,#00000000B ; timer stop
MOV PRM0,#00010011B ; TM0 prescaler fxx/32
MOV CRC0,#00011000B ; clear enable TM0 by CR01
MOVW CR01,#INTVL1-1 ; set interval time
CLR1 CMK01 ; open INTC01 mask
MOV TMC0,#00001000B ; timer start
EI
;
RET
;
END

```

タイマ/カウンタを8ビット動作モードで使用する場合

タイマ/カウンタ1, 2, およびタイマ3は, タイマ機能を8ビット動作モードで使用することができます。

8ビット動作モードの使用方法は, 基本的に16ビット動作モードと同じです。

次に, 8ビット動作モードの使用方法を示します。

- ・タイマ・コントロール・レジスタ (TMC0, TMC1) で8ビット/16ビット動作モードを指定します。



- 注 0 : 8ビット動作モード
- 1 : 16ビット動作モード

- ・コンペア・レジスタ (CR01) の下位8ビットにインターバル時間を設定します。

インターバル時間 = (n+1) × x / f_{xx} { n : 0 n FFH }

(f_{xx} = クリスタル / セラミック発振周波数)

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル / セラミック発振周波数の分周比)

2.3 プログラマブル矩形波出力の応用

プログラマブル矩形波出力は、基本的にインターバル・タイマ機能を用い、割り込み要求信号によって外部への出力信号を反転させるものです。したがって、デューティ比は50 %となります。

(1) 処理概要

(a) 仕様

タイマ/カウンタ0を用いて、TO1端子から周波数1 kHzの矩形波を出力するプログラム例を示します。

この場合、INTC01割り込み要求の発生タイミングは500 μsごと(2 kHz)となります。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32
- ・ 入力パラメータ (INTVL3) : 下記計算式参照

$$\begin{aligned}
 \text{INTVL3} &= \frac{\text{割り込み要求タイミング}}{\text{カウント・クロック}} \\
 &= \frac{500 \mu\text{s}}{32/f_{xx}^{\text{注}}} \\
 &= 391 \text{ (小数点以下四捨五入)}
 \end{aligned}$$

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

プログラマブル矩形波出力は、コンペア・レジスタ (CR00, CR01) の設定値がタイマ・レジスタ0 (TM0) のカウント値と一致するたびにタイマ出力端子 (TO0, TO1) の出力レベルを反転させるモードです。TO0端子はコンペア・レジスタ (CR00) とタイマ・レジスタ0 (TM0) の一致によって出力レベルが反転し、TO1端子はコンペア・レジスタ (CR01) とタイマ・レジスタ0 (TM0) の一致によって出力レベルが反転します。

図2 - 3に、TO1端子からの矩形波出力の機能のブロック図を示します。

図2 - 3 TO1端子からの矩形波出力

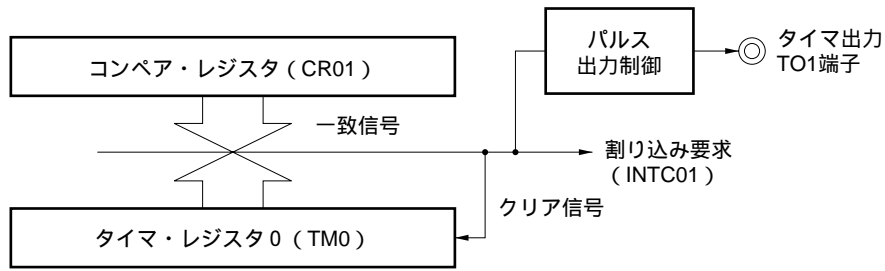
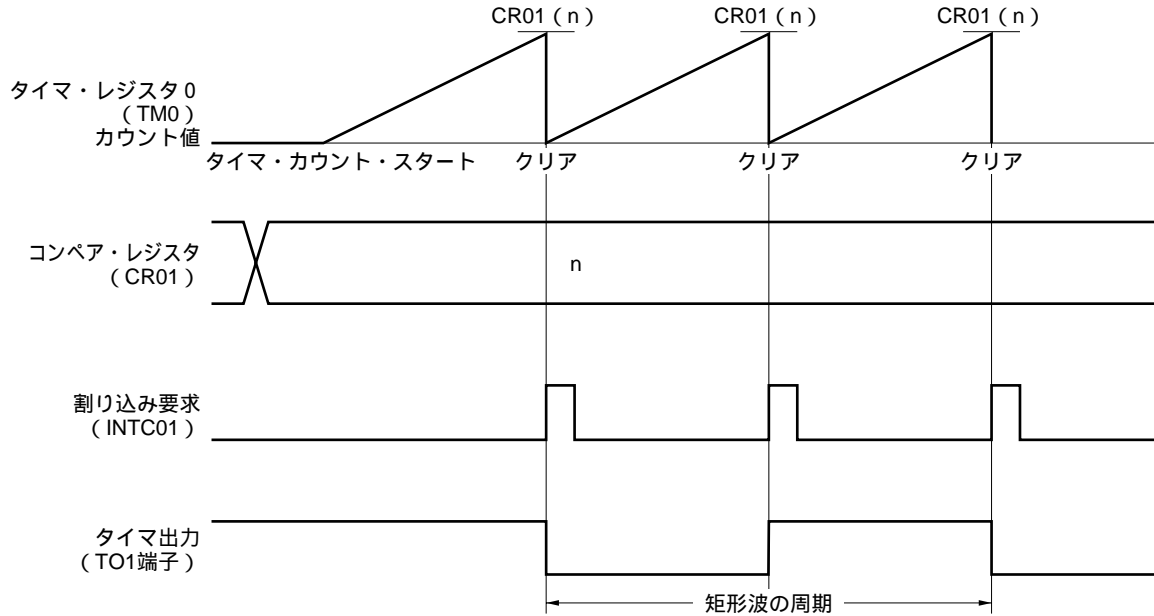


図2 - 4 TO1端子からの矩形波出力のタイミング・チャート



備考 矩形波の周期 = $(n+1) \times 2 \times x / f_{xx}$ { n:0 n FFFFH }

(f_{xx} = クリスタル/セラミック発振周波数)

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル/セラミック発振周波数の分周比)

INTC01割り込み要求はタイマ・レジスタ0 (TM0) のカウント値とコンペア・レジスタ (CR01) の設定値の一致によって発生します。インターバル・タイマとして、INTC01割り込み要求を発生させるために、一致時にタイマ・レジスタ0 (TM0) が自動的にクリアされるように設定します。

タイマ出力 (TO1) は、INTC01割り込み要求によって反転されるために、周期は2.2 内部インターバル・タイマの応用でとりあげたインターバル・タイマによるインターバル時間の2倍になります。

(2) 使用RAM領域

なし

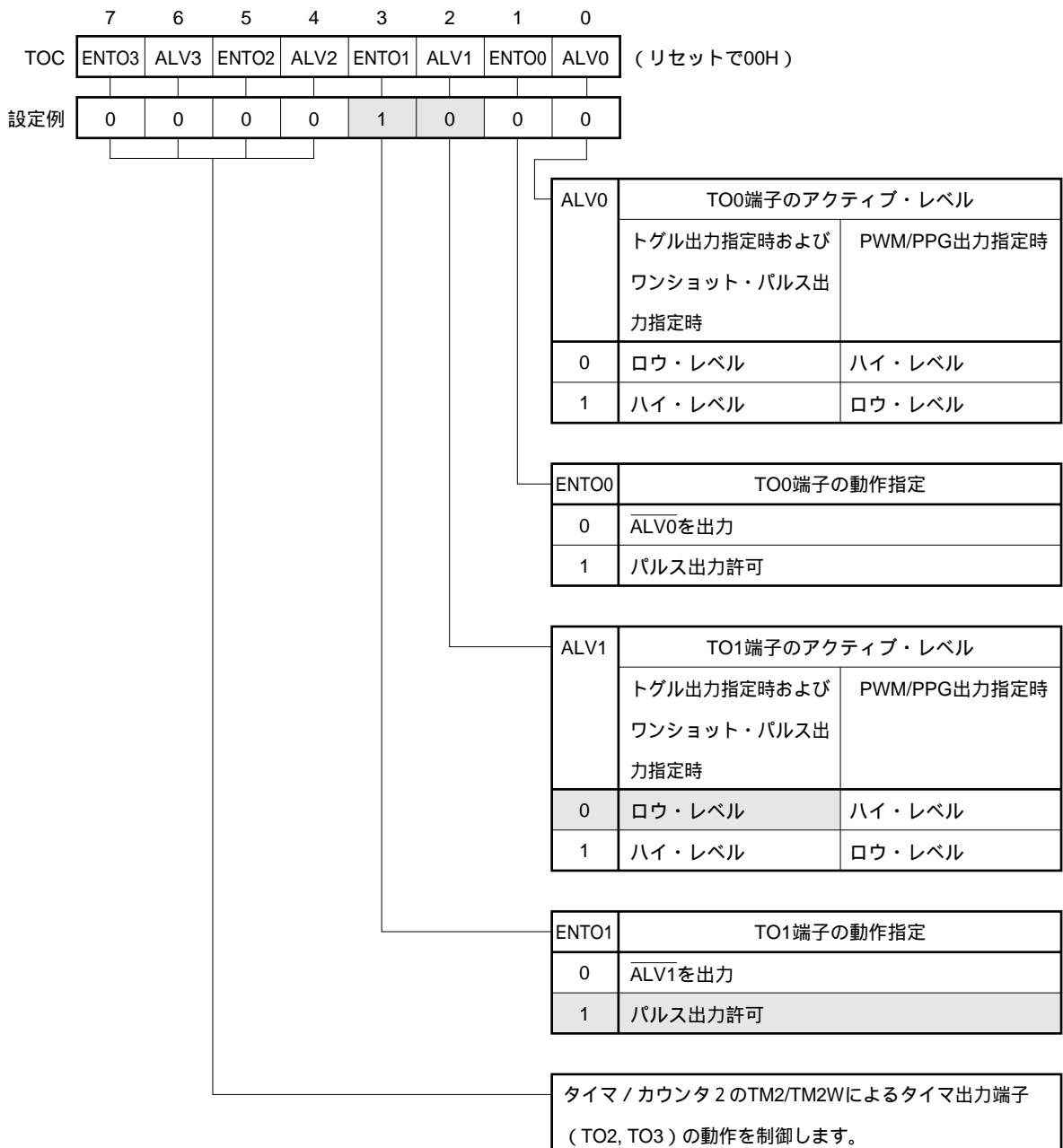
(3) 使用レジスタ

(a) 汎用レジスタ

なし

(b) 特殊機能レジスタ

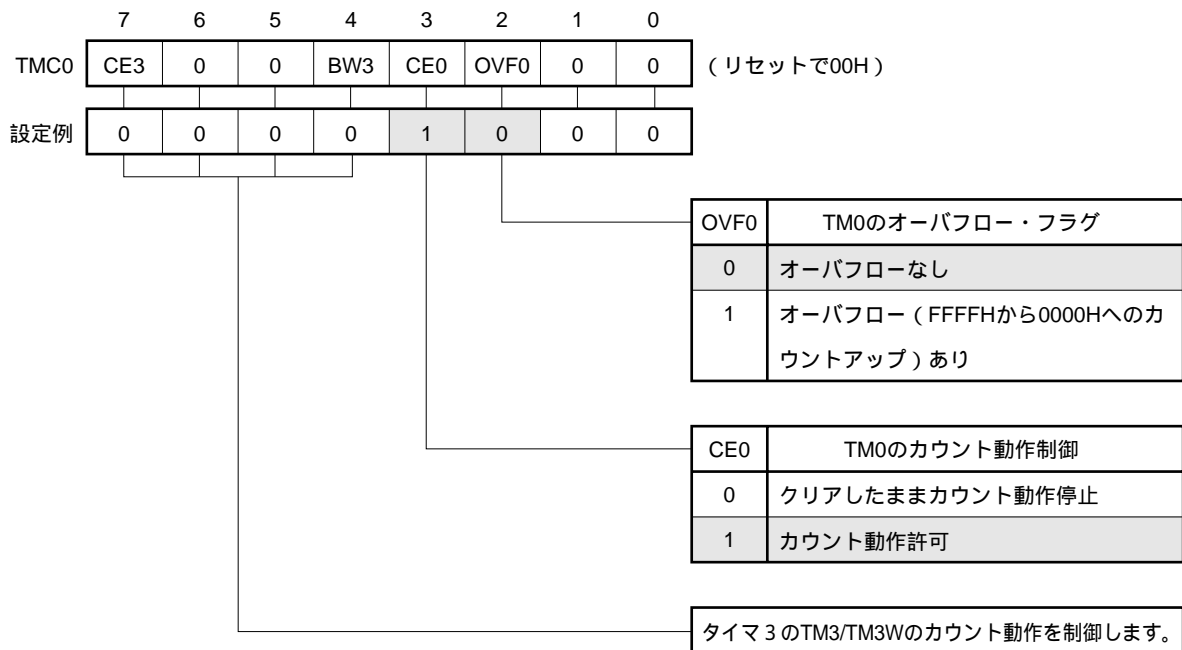
タイマ出力コントロール・レジスタ (TOC)



キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)



タイマ・コントロール・レジスタ0 (TMC0)



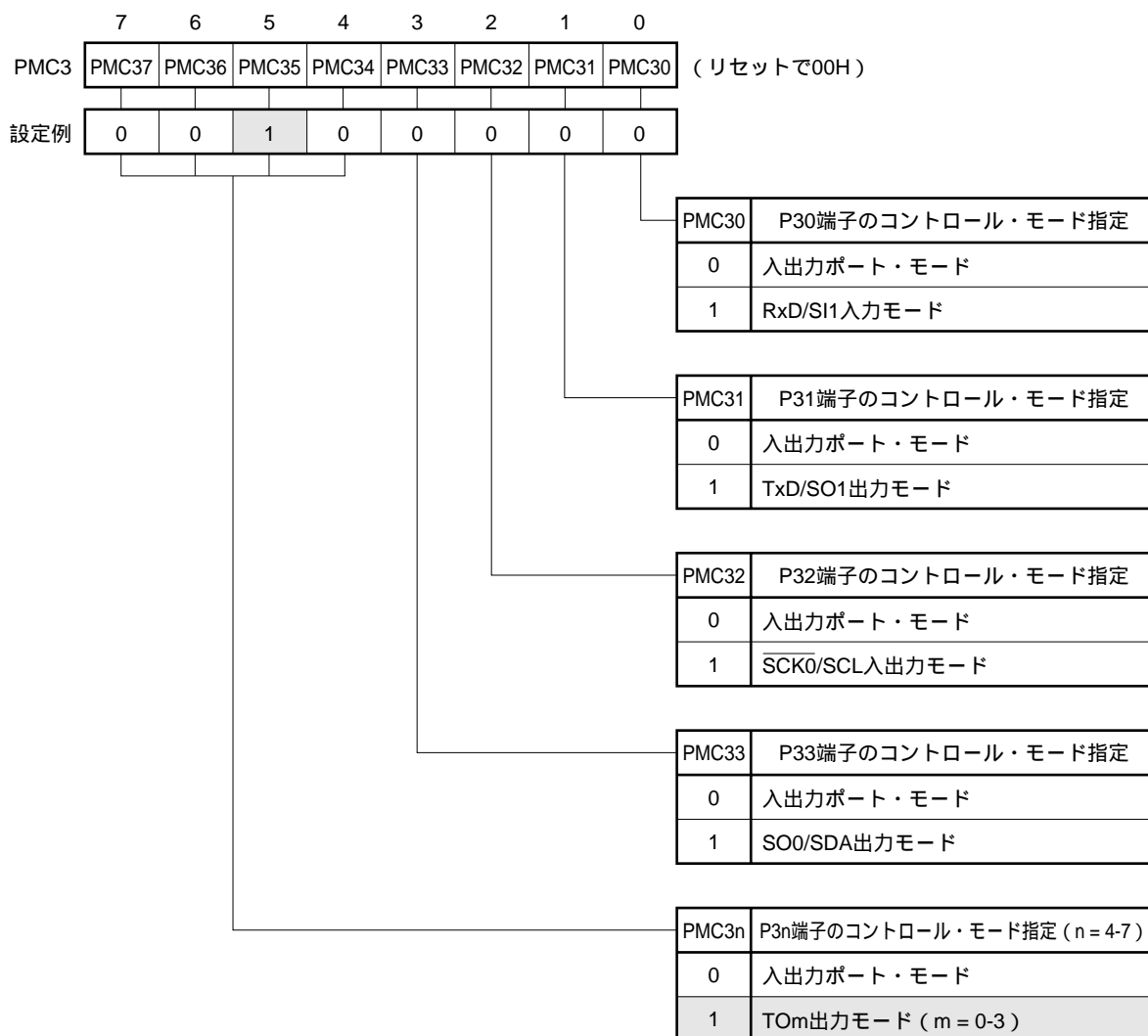
備考 OVF0ビットはソフトウェアでのみリセットされます。

プリスケアラ・モード・レジスタ0 (PRM0)



備考 f_{xx} : クリスタル/セラミック発振周波数

ポート3モード・コントロール・レジスタ (PMC3)



(4) 入力方法

下記に示すレーベルに次の内容を定義します。

INTVL3 : 矩形波の周期の値を定義します。

定義した値は、コンペア・レジスタ (CR01) の設定値を示します。

タイマ/カウンタ0のカウンタ・クロックは $f_{xx}/8$ から $f_{xx}/2048$ までプリスケアラ・モード・レジスタ0 (PRM0) により指定します。矩形波の周期は、次式で求められます。

$$\begin{aligned} \text{矩形波の周期} &= \text{INTVL3} \times 2 \times (8/f_{xx} \sim 2048/f_{xx}) \\ &\{ \text{INTVL3} : 0 \quad \text{INTVL3} \quad \text{FFFFH} \} \\ &(\text{f}_{xx} = \text{クリスタル/セラミック発振周波数}) \end{aligned}$$

(5) 出力方法

なし

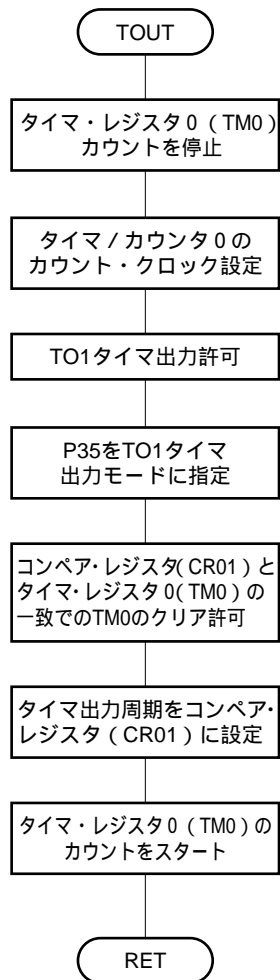
(6) プログラム説明 [レーベル名称: TOUT]

- (a) タイマ・レジスタ0 (TM0) のカウンタ動作を停止します。
- (b) タイマ/カウンタ0のカウンタ・クロックを $f_{xx}/32$ に指定します。
- (c) タイマ出力コントロール・レジスタ (TOC) で、タイマ出力 (TO1端子) のアクティブ・レベルをハイ・レベルに設定し、パルス出力許可にします。
- (d) P35をタイマ出力TO1端子として用いるために、ポート3モード・コントロール・レジスタ (PMC3) をTO1出力モードに指定します。

注意 P35端子を、ポート3モード・コントロール・レジスタ (PMC3) でTO1出力モードに切り替える場合、出力信号にレベル差があるとTO1を出力したときに不定なレベルが出力され、エッジ割り込みが発生するなどの誤動作をする場合が考えられます。したがって必ず先にTO1端子のアクティブ・レベルをタイマ出力コントロール・レジスタ (TOC) で設定したあと、ポート3モード・コントロール・レジスタ (PMC3) でTO1出力モードに切り替えてください。

- (e) タイマ・レジスタ0 (TM0) とコンペア・レジスタ (CR01) の一致時の、タイマ・レジスタ0 (TM0) のクリア動作を許可します。
- (f) コンペア・レジスタ (CR01) に矩形波の周期を設定します。
- (g) タイマ・レジスタ0 (TM0) のカウンタ動作を許可します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

INTVL3：矩形波の周期の値を定義

メイン・ルーチンのプログラム・リスト記述例

```

PUBLIC INTVL3      ; PARAMETER
EXTRN  TOUT        ; PACKAGE
.
.
INTVL3 EQU 391      ; PARAMETER FOR INTERVAL
.
.
TUT:
CALL  !TOUT
.
.

```

備考 上記記述例のようにRAM領域 (INTVL3) に矩形波の周期の値を定義し、サブルーチンを呼んでください。

本アプリケーション・ルーチンのプログラム・リスト

```

NAME  TOUTM
;
;*****
;* programmable pulse output *
;*****
;
;
PUBLIC TOUT
EXTRN  INTVL3      ; timer output frequency
;
CSEG
TOUT:
MOV    TMC0,#0000000B ; timer stop
MOV    PRM0,#00010011B ; TMO prescaler fxx/32
MOV    TOC,#00001000B ; enable T01 timer output
MOV    OSPC,#10000000B ; TOGGLE OUTPUT
MOV    PMC3,#00100000B ; P35=control port T01 OUT
MOV    CRC0,#00011000B ; clear enable TMO by CR01
MOVW   CR01,#INTVL3-1 ; set interval time
MOV    TMC0,#00001000B ; timer start

RET
;
END

```

2.4 フリーランニング・インターバル・タイマの応用

フリーランニング・インターバル・タイマはタイマを任意の値でクリアせず、フリーランニングさせ、割り込み要求処理中にコンペア・レジスタ (CR00, CR01) に一定の値を加算することにより、インターバル・タイマを作るものです。

この方法を用いると、1つのタイマ/カウンタで2種類のインターバル・タイマを作ることができます。

(1) 処理概要

(a) 仕様

タイマ/カウンタ0を用いて、1つのタイマ・レジスタに対し、2本のコンペア・レジスタ (CR00, CR01) を使用して、2種類のインターバル・タイマを作り (INTC00とINTC01割り込み要求発生)、さらに、そのインターバル・タイマによってTO0端子から周波数500 Hz, TO1端子から周波数1 kHzのタイマ出力を行うプログラム例を示します (タイマ出力のデューティは50%です)。

この場合、INTC00割り込み要求を1 msごとに、INTC01割り込み要求を500 μsごとにそれぞれ発生させます。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32
- ・ 入力パラメータ (INTVL4) : 下記計算式参照
- (INTVL5) : "

$$\begin{aligned}
 \text{INTVL4} &= \frac{1 \text{ ms}}{\text{カウント・クロック}} \\
 &= \frac{1 \text{ ms}}{32/f_{xx}^{\text{注}}} \\
 &= 781 \text{ (小数点以下四捨五入)} \\
 \\
 \text{INTVL5} &= \frac{500 \mu\text{s}}{\text{カウント・クロック}} \\
 &= \frac{500 \mu\text{s}}{32/f_{xx}^{\text{注}}} \\
 &= 391 \text{ (小数点以下四捨五入)}
 \end{aligned}$$

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

図2 - 5に、TO0, TO1端子からのタイマ出力の機能を示します。

図2 - 5 TO0, TO1端子からのタイマ出力

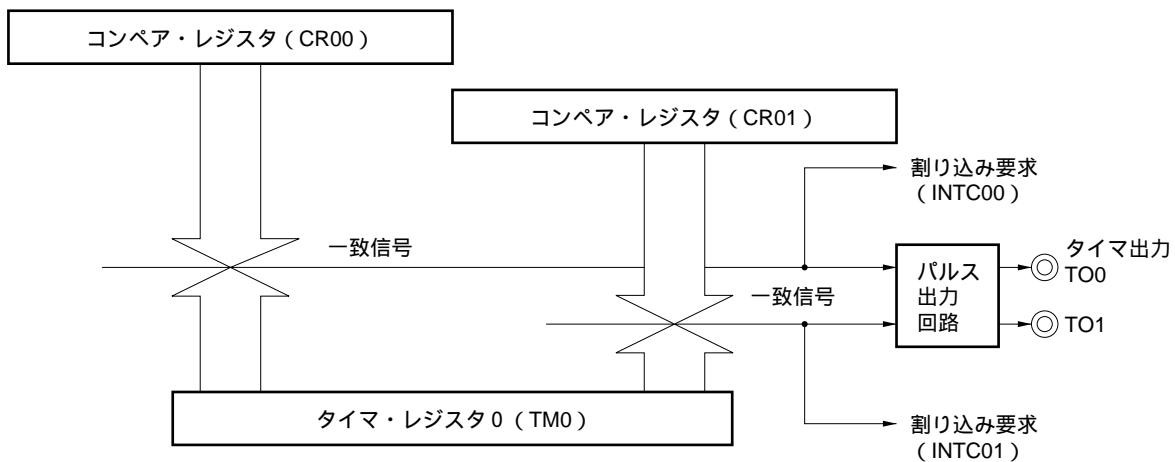
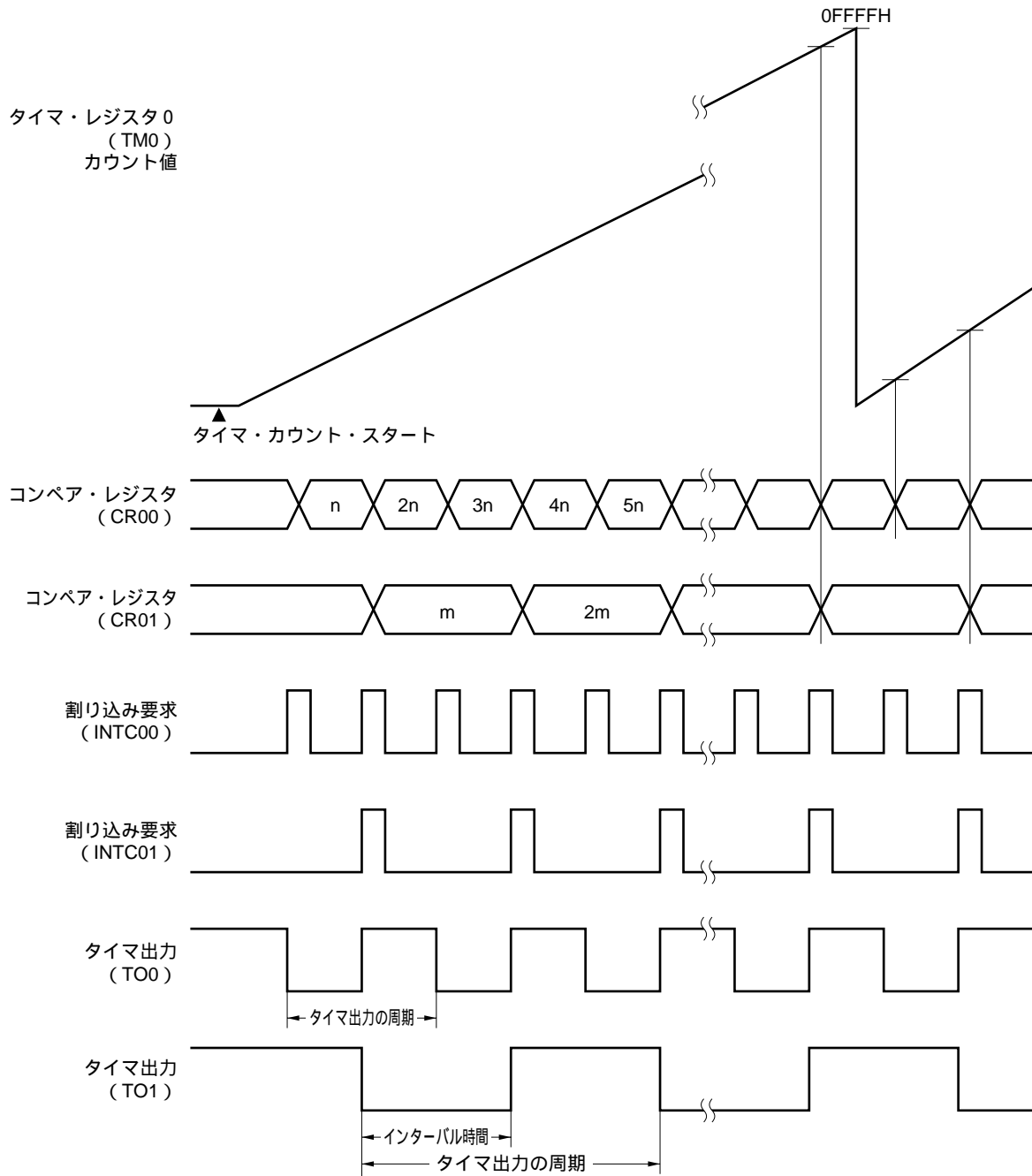


図2 - 6 TO0, TO1端子からのタイマ出力のタイミング・チャート



備考 タイマ出力の周期 = $n(m) \times 2x/f_{xx}$

{ $n(m) : 1 \quad n(m) \quad FFFFH$ }

(f_{xx} = クリスタル/セラミック発振周波数)

$\left\{ \begin{array}{l} n = \text{TM0のコンペア・レジスタ設定値} \\ m = \text{TM1のコンペア・レジスタ設定値} \end{array} \right.$

$x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048$

(x = クリスタル/セラミック発振周波数の分周比)

2種類のインターバル・タイマを作るには、まず、タイマ・レジスタ0 (TM0) をフリーランニングさせてください。

その後、INTC00, INTC01割り込み要求発生時に、現在のコンペア・レジスタ (CR00, CR01) の設定値にそれぞれのインターバル時間分の設定値を加算することによって、一定周期のタイマ出力ができます。

(2) 使用RAM領域

なし

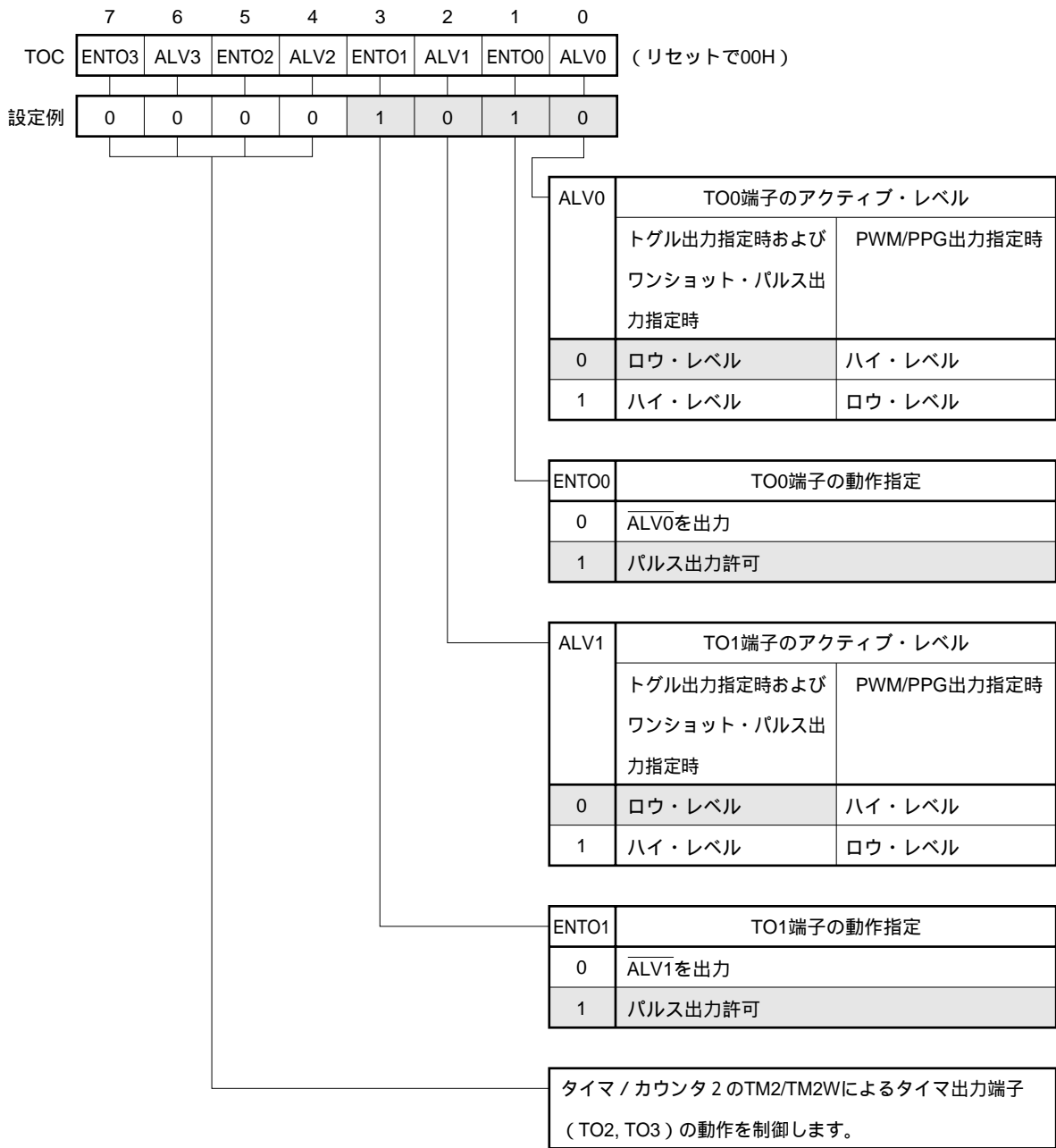
(3) 使用レジスタ

(a) 汎用レジスタ

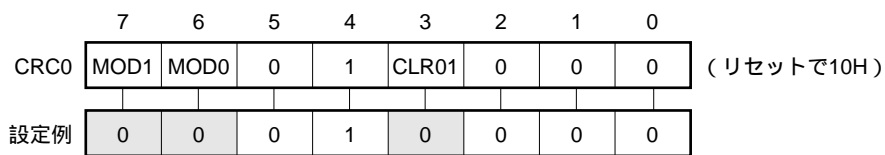
AXレジスタ

(b) 特殊機能レジスタ

タイマ出力コントロール・レジスタ (TOC)

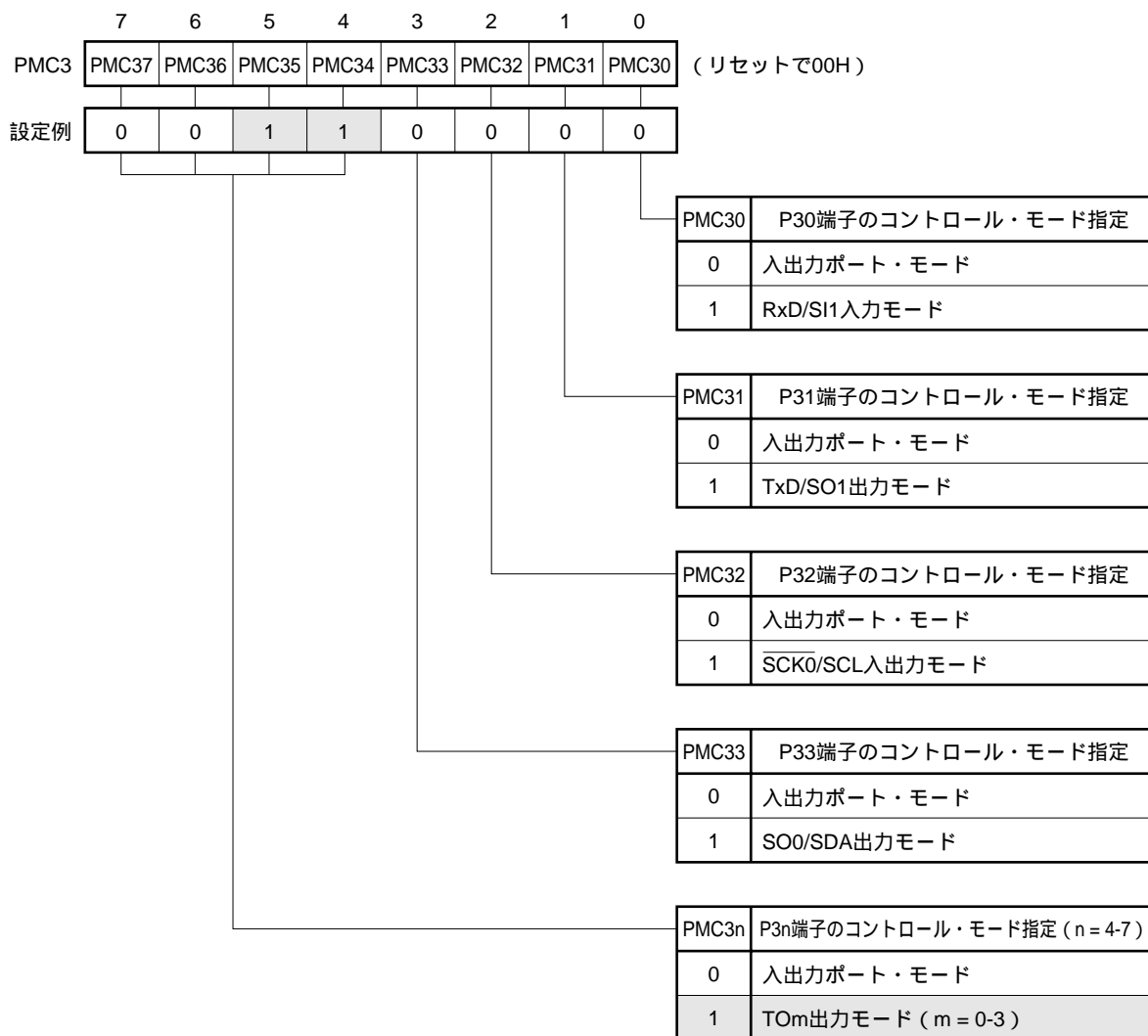


キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)

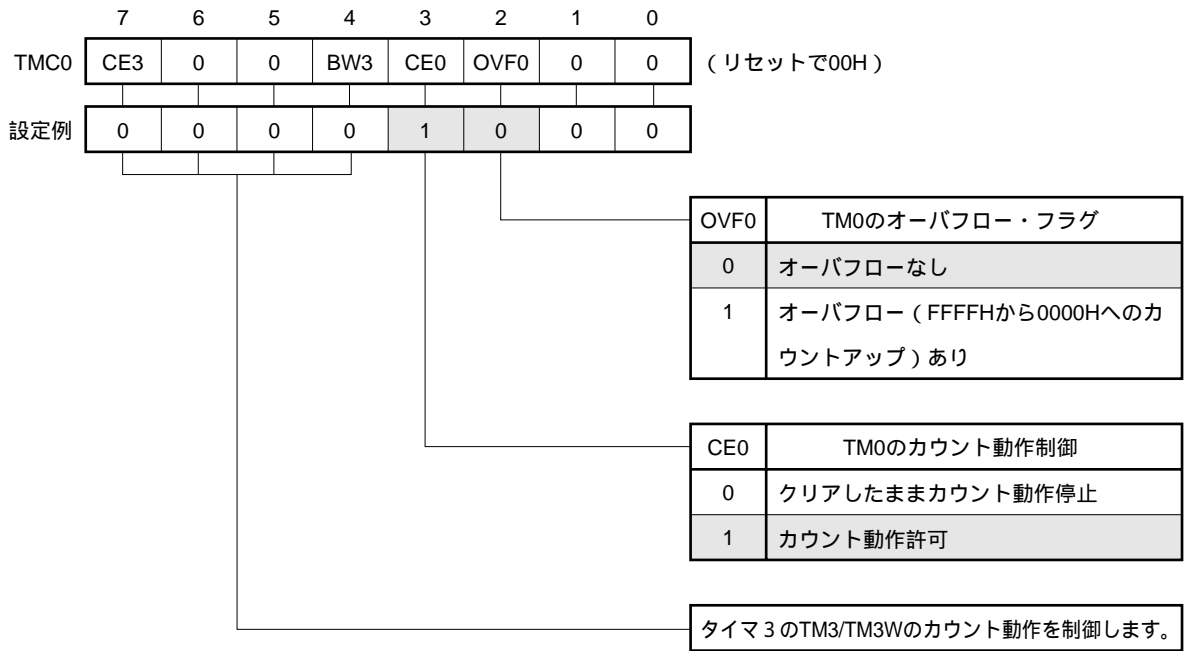


MOD1	MOD0	CLR01	タイマ出力モード指定		TM0 = CR01時の TM0のクリア動作
			TO0	TO1	
0	0	0	トグル出力	トグル出力	禁止
0	0	1	トグル出力	トグル出力	許可
0	1	0	PWM出力	トグル出力	禁止
0	1	1	設定禁止		
1	0	0	PWM出力	PWM出力	禁止
1	0	1	設定禁止		
1	1	0	設定禁止		
1	1	1	PPG出力	トグル出力	許可

ポート3モード・コントロール・レジスタ (PMC3)

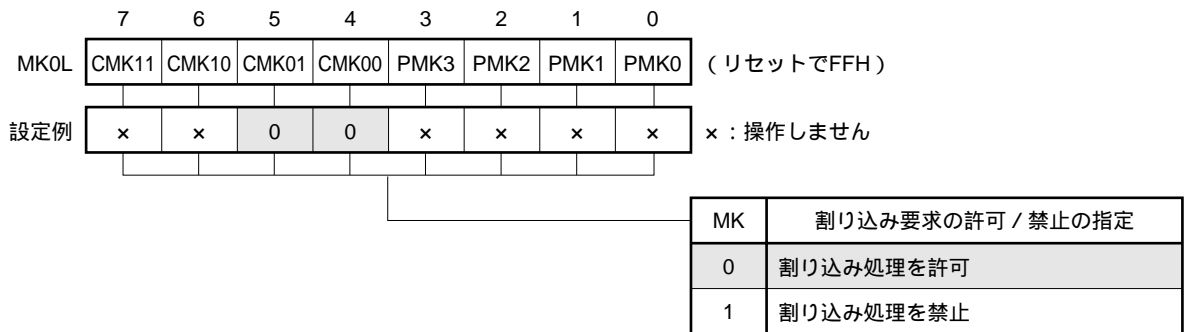


タイマ・コントロール・レジスタ0 (TMC0)

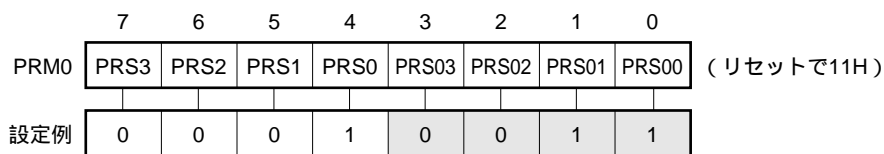


備考 OVF0ビットはソフトウェアでのみリセットされます。

割り込みマスク・レジスタL (MK0L)



プリスケアラ・モード・レジスタ0 (PRM0)



($f_{xx} = 25 \text{ MHz}$)

PRS03	PRS02	PRS01	PRS00	タイマ/カウンタ0のTM0のカウント・クロック指定	
				カウント・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	$f_{xx}/8$	0.32 μs
0	0	1	0	$f_{xx}/16$	0.64 μs
0	0	1	1	$f_{xx}/32$	1.28 μs
0	1	0	0	$f_{xx}/64$	2.56 μs
0	1	0	1	$f_{xx}/128$	5.12 μs
0	1	1	0	$f_{xx}/256$	10.2 μs
0	1	1	1	$f_{xx}/512$	20.5 μs
1	0	0	0	$f_{xx}/1024$	41.0 μs
1	0	0	1	$f_{xx}/2048$	81.9 μs
1	1	1	1	外部クロック (INTP3)	-
上記以外				設定禁止	

タイマ3のTM3/TM3Wのカウント・クロックを指定します。

備考 f_{xx} : クリスタル/セラミック発振周波数

(4) 入力方法

下記に示すレーベルに次の内容を定義します。

INTVL4 : TO0端子からのタイマ出力の周期を定義します。

定義した設定値は、一回INTC00割り込み要求が発生するごとに、割り込み処理で、現在のコンペア・レジスタ (CR00) の設定値にこの設定値を加算します。

INTVL5 : TO1端子からのタイマ出力の周期を定義します。

定義した設定値は、一回INTC01割り込み要求が発生するごとに、割り込み処理で、現在のコンペア・レジスタ (CR01) の設定値にこの設定値を加算します。

タイマ/カウンタ0のカウント・クロックは $f_{xx}/8$ から $f_{xx}/2048$ までプリスケアラ・モード・レジスタ0 (PRM0) により指定します。タイマ出力の周期は次式で求められます。

$$\begin{aligned} \text{タイマ出力の周期} &= \text{INTVL4 (INTVL5)} \times 2 \times (8/f_{xx} \sim 2048/f_{xx}) \\ &\quad \{ \text{INTVL4 (INTVL5)} : 0 \quad \text{INTVL4 (INTVL5)} \quad \text{FFFFH} \} \\ &\quad (f_{xx} = \text{クリスタル/セラミック発振周波数}) \end{aligned}$$

(5) 出力方法

なし

(6) プログラム説明

イニシャライズ処理 [レーベル名称: FRUN0]

- (a) タイマ・レジスタ0 (TM0) のカウント動作を停止します。
- (b) タイマ出力コントロール・レジスタ (TOC) で、タイマ出力TO0端子、TO1端子のアクティブ・レベルをロウ・レベルに設定し、パルス出力許可に設定します。
- (c) P34, P35をTO0, TO1端子として用いるためにポート3モード・コントロール・レジスタ (PMC3) で、P34, P35をTO0, TO1出力モードに指定します。

注意 P34, P35端子を、ポート3モード・コントロール・レジスタでTO0, TO1出力モードに切り替える場合、出力信号にレベル差があるとTO0, TO1出力モードに設定したときに不定なレベルが出力され、エッジ割り込みが発生するなどの誤動作をする場合が考えられます。したがって必ず先にTO1端子の出力アクティブ・レベルをタイマ出力コントロール・レジスタ (TOC) で設定したのち、ポート3モード・コントロール・レジスタ (PMC3) でTO0, TO1出力モードに切り替えてください。

- (d) タイマ・レジスタ0 (TM0) とコンペア・レジスタ (CR00, CR01) の一致時の、タイマ・レジスタ0 (TM0) のクリア動作を禁止します (フリーランニング・モード)。

- (e) コンペア・レジスタ (CR00, CR01) にそれぞれのタイマ出力の周期を設定します。
- (f) INTC00, INTC01割り込み要求のマスクを解除します。
- (g) タイマ・レジスタ 0 (TM0) のカウント動作を許可します。

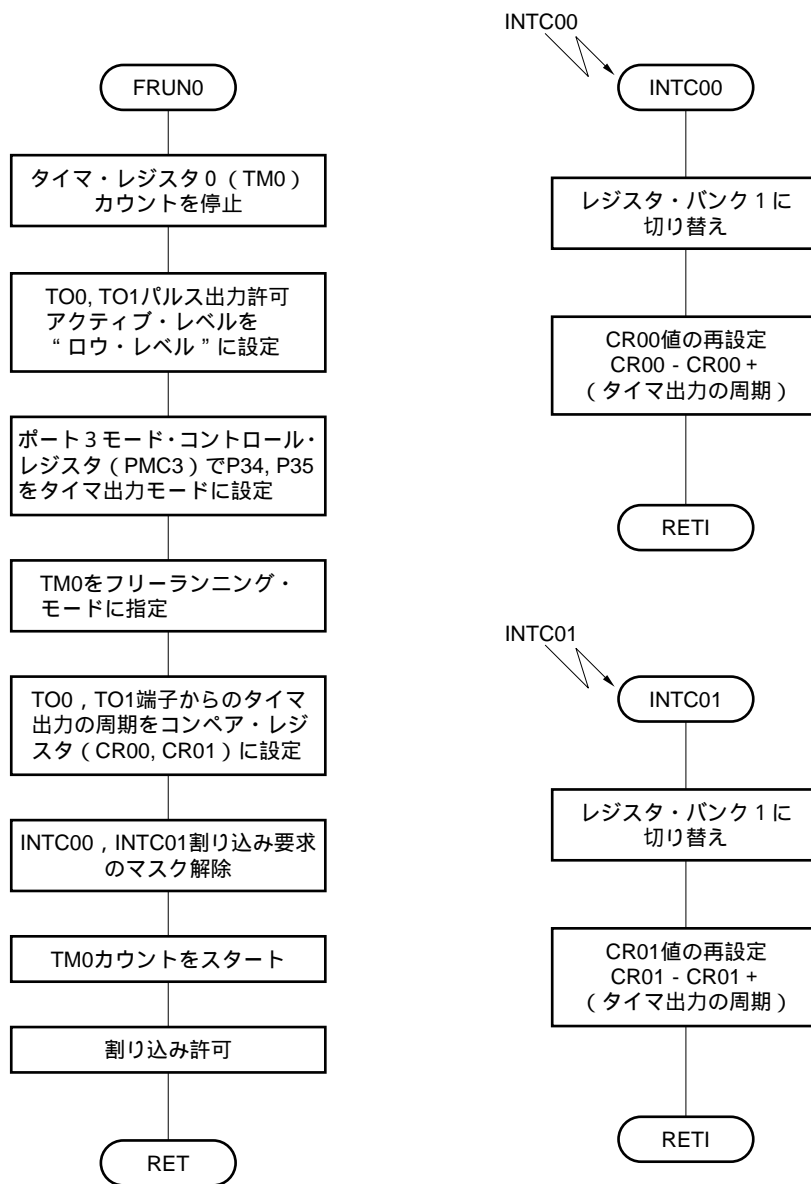
INTC00割り込み処理 [レーベル名称 : INTC00]

- (a) レジスタ・バンク 1 に切り替えます。
- (b) 現在のコンペア・レジスタ (CR00) の設定値にタイマ出力の周期の値を加算します。加算値のオーバーフローは無視します。
- (c) 加算値をコンペア・レジスタ (CR00) に設定します。

INTC01割り込み処理 [レーベル名称 : INTC01]

- (a) レジスタ・バンク 1 に切り替えます。
- (b) 現在のコンペア・レジスタ (CR01) の設定値にタイマ出力の周期の値を加算します。加算値のオーバーフローは無視します。
- (c) 加算値をコンペア・レジスタ (CR01) に設定します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

INTVL4 : TO0端子からのタイマ出力の周期を定義

INTVL5 : TO1端子からのタイマ出力の周期を定義

メイン・ルーチンのプログラム・リスト記述例

```
PUBLIC  INTVL4,INTVL5      ; PARAMETER
EXTRN  FRUNO              ; PACKAGE
      .
      .
INTVL4 EQU    781         ; free running timer INTC00 compare data
INTVL5 EQU    391         ; free running timer INTC01 compare data
      .
      .
      CALL    !FRUNO
      .
      .
```

備考 上記記述例のようにRAM領域 (INTVL4, INTVL5) にTO0, TO1端子からのタイマ出力の周期を定義し, サブルーチンを呼んでください。

本アプリケーション・ルーチンのプログラム・リスト

```

NAME      F_RUN0
;
;*****
;
;* 16bit-Timer / Counter                                     *
;*      free running interval timer                         *
;*****
;
;
PUBLIC    FRUN0
EXTRN    INTVL4,INTVL5

INTC00VT CSEG AT 000EH
        DW      INTC00      ; INTC00
INTC01VT CSEG AT 0010H
        DW      INTC01      ; INTC01
;
;
CSEG
FRUN0:
MOV      TMC0,#00000000B ; timer stop
MOV      TOC,#00001010B ; timer output,active level low
MOV      PMC3,#00110000B ; P3 control port
MOV      CRC0,#00010000B ; set timer free running mode
MOV      PRM0,#00010011B ; TMO prescaler fxx/32
MOVW     CR00,#INTVL4     ; set interval time
MOVW     CR01,#INTVL5     ; set interval time
CLR1     CMK00             ; open INTC00 mask
CLR1     CMK01             ; open INTC01 mask
MOV      TMC0,#00001000B ; start timer
EI                          ; interrupt enable

RET

;
;*****
;
;*                      INTC00 interrupt routine          *
;*****
INTC00:
        ADDW    CR00,#INTVL4 ; count INTC00 compare data
        RETI

;
;*****
;
;*                      INTC01 interrupt routine          *
;*****
INTC01:
        ADDW    CR01,#INTVL5 ; count INTC01 compare data
        RETI

;
END

```

タイマ/カウンタを8ビット動作モードで使用する場合

タイマ/カウンタ1, 2は, タイマ機能を8ビット動作モードで使用することができます。

8ビット動作モードの使用方法は, 基本的に16ビット動作モードと同じです。

次に, 8ビット動作モードの使用方法を示します。

- ・タイマ・コントロール・レジスタ (TMC0, TMC1) で8ビット/16ビット動作モードを指定します。



注 0 : 8ビット動作モード
 1 : 16ビット動作モード

- ・2種類の周期のインターバル・タイマを作るには, まず, タイマ・レジスタ1 (TM1) またはタイマ・レジスタ2 (TM2) をフリーランニングさせてください。その後, 割り込み要求発生時ごとに, 現在のコンペア・レジスタ (CR10, CR11, CR20, CR21) の設定値にそれぞれのインターバル時間分の設定値を加算することによって, 一定周期の2つのタイマ出力ができます。
- ・コンペア・レジスタ (CR10, CR11, CR20, CR21) の下位8ビットにインターバル時間を設定します。

タイマ出力の周期 = $n \times 2^x \times x / f_{xx}$

{ n : 1 n FFH }

(f_{xx} = クリスタル/セラミック発振周波数)

n = コンペア・レジスタ設定値

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル/セラミック発振周波数の分周比)

2.5 PWM出力の応用

PWM出力は、タイマのフルカウント期間を1周期とし、コンペア動作による割り込み要求により、パルス幅を制御します。

(1) 処理概要

(a) 仕様

TO0端子から、INTC00割り込み要求によってパルス幅を決定するPWM信号を出力するプログラム例を紹介します。なお、パルス幅変更要求処理のプログラム例において、次のPWM出力のパルス幅は、A/Dコンバータ変換結果レジスタ(ADCR)の格納値をコンペア・レジスタ(CR00)に設定しています。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

タイマ・レジスタ0(TM0)がフルカウントする時間を1周期とするPWM信号を出力するモードです。TO0端子から出力されるPWM信号のパルス幅は、コンペア・レジスタ(CR00)の設定値によって決定されます。この機能を使用する場合は、キャプチャ/コンペア・コントロール・レジスタ0(CRC0)のCLR01ビットを0にする必要があります。

図2-7に、TO0端子からのPWM出力の機能を示します。

図2-7 TO0端子からのPWM出力

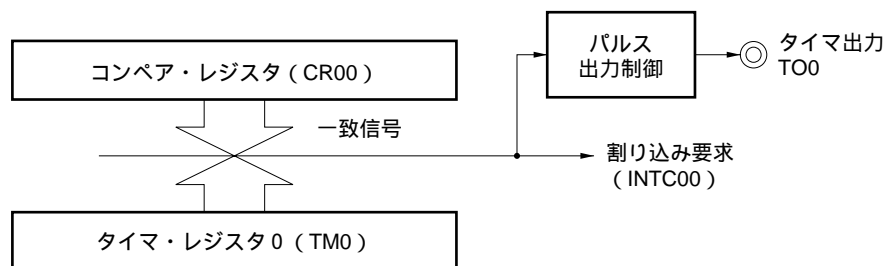
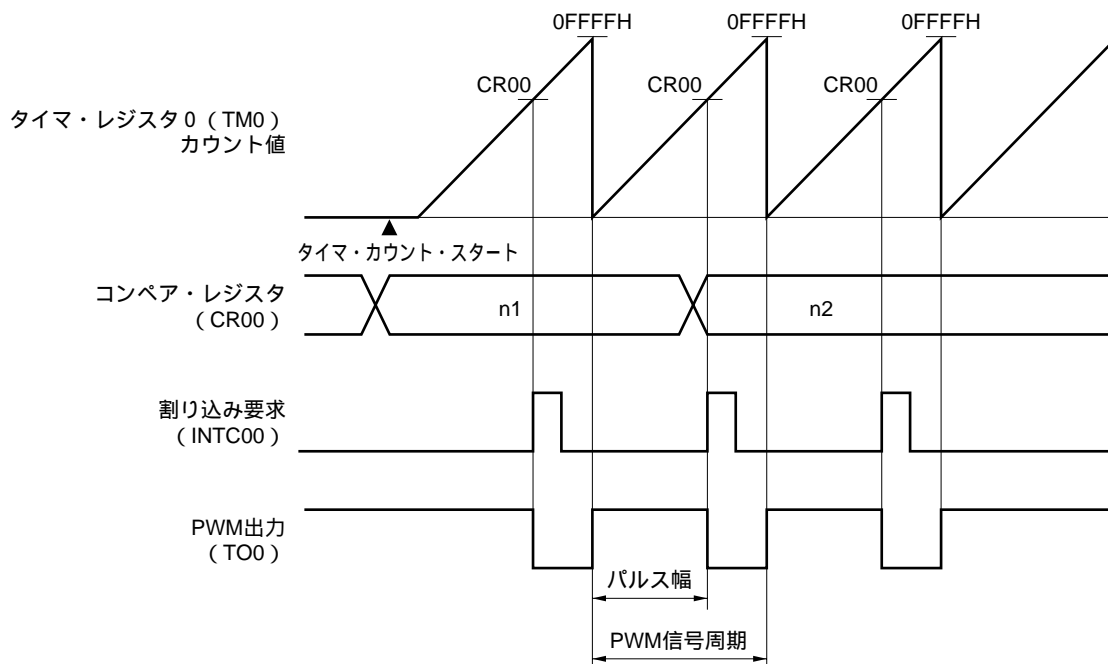


図2 - 8 TO0端子からのPWM出力のタイミング・チャート



備考 PWM信号周期 = $65536 \times x / f_{xx}$

パルス幅 = $n \times x / f_{xx}$

{ n : 1 n FFFFH }

(f_{xx} = クリスタル/セラミック発振周波数)

n = TM0のコンペア・レジスタ設定値

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル/セラミック発振周波数の分周比)

PWM信号の1周期は、タイマ・レジスタ0 (TM0) がフルカウント (FFFFH) する期間になりますので、タイマはフリーランニングさせてください。

なお、タイマ/カウンタ0のカウント・クロックは $f_{xx}/8$ から $f_{xx}/2048$ でプリスケアラ・モード・レジスタ0により指定します。

1周期は、 $f_{xx} = 25$ MHzの場合、20 msから5.37 sとなります。

また、パルス幅を変更する場合は、タイマ・レジスタ0 (TM0) とコンペア・レジスタ (CR00) の設定値の一致によるINTC00割り込み要求発生時に、割り込み処理内で、パルス幅を決定するコンペア・レジスタ (CR00) の設定値を書き換えます。

備考 TO0, TO1端子より同時に2種類のPWM出力を行う場合は、コンペア・レジスタ (CR00, CR01) にそれぞれのパルス幅を決定する値を設定しておきます。コンペア・レジスタ (CR00, CR01) とタイマ・レジスタ0 (TM0) の一致によって発生する2本の割り込み要求 (INTC00, INTC01) によって、同時に2種類のPWM出力ができます。

(2) 使用RAM領域

本プログラムでは、表2 - 2に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表2 - 2 TM0によるPWM出力のプログラムで用いるワーク/フラグ

RAM/フラグ名称	用 途	バイト数	初期値
DUTY1	パルス幅 (CR00) に設定するカウンタ値を設定するRAM領域	2	7FFFH
CNGFLAG	PWMのパルス幅変更判別フラグ CNGFLAG = 0...変更要求なし CNGFLAG = 1...変更要求あり	1ビット	0

(3) 使用レジスタ

(a) 汎用レジスタ

イニシャライズ処理 [レーベル名称 : PWM00]

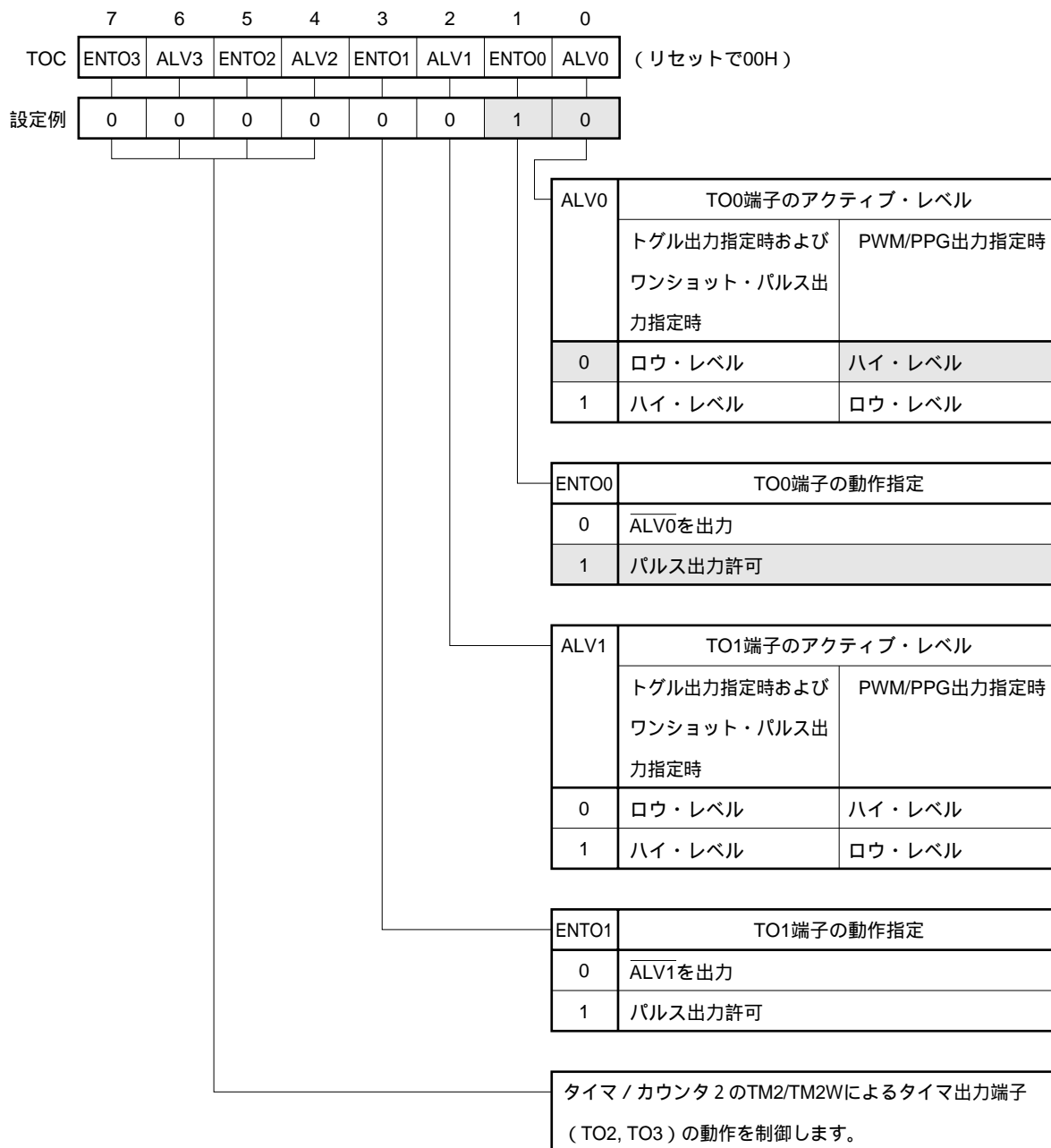
AXレジスタ

INTC00割り込み処理 [レーベル名称 : INTC00]

AXレジスタ

(b) 特殊機能レジスタ

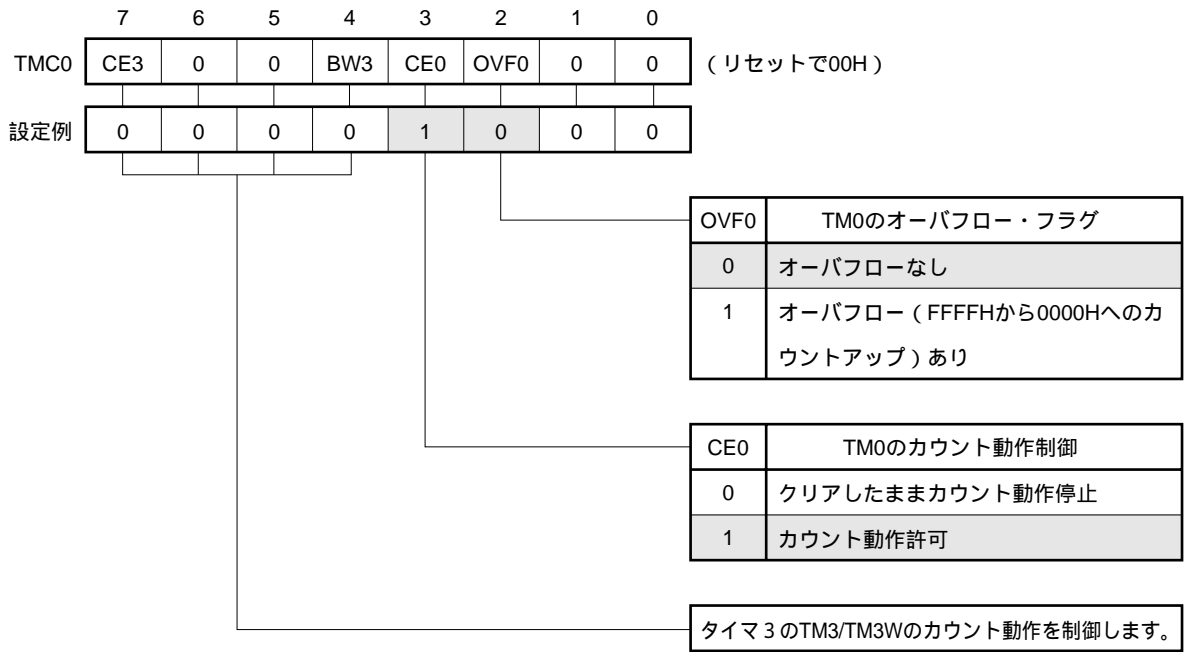
タイマ出力コントロール・レジスタ (TOC)



キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)



タイマ・コントロール・レジスタ0 (TMC0)



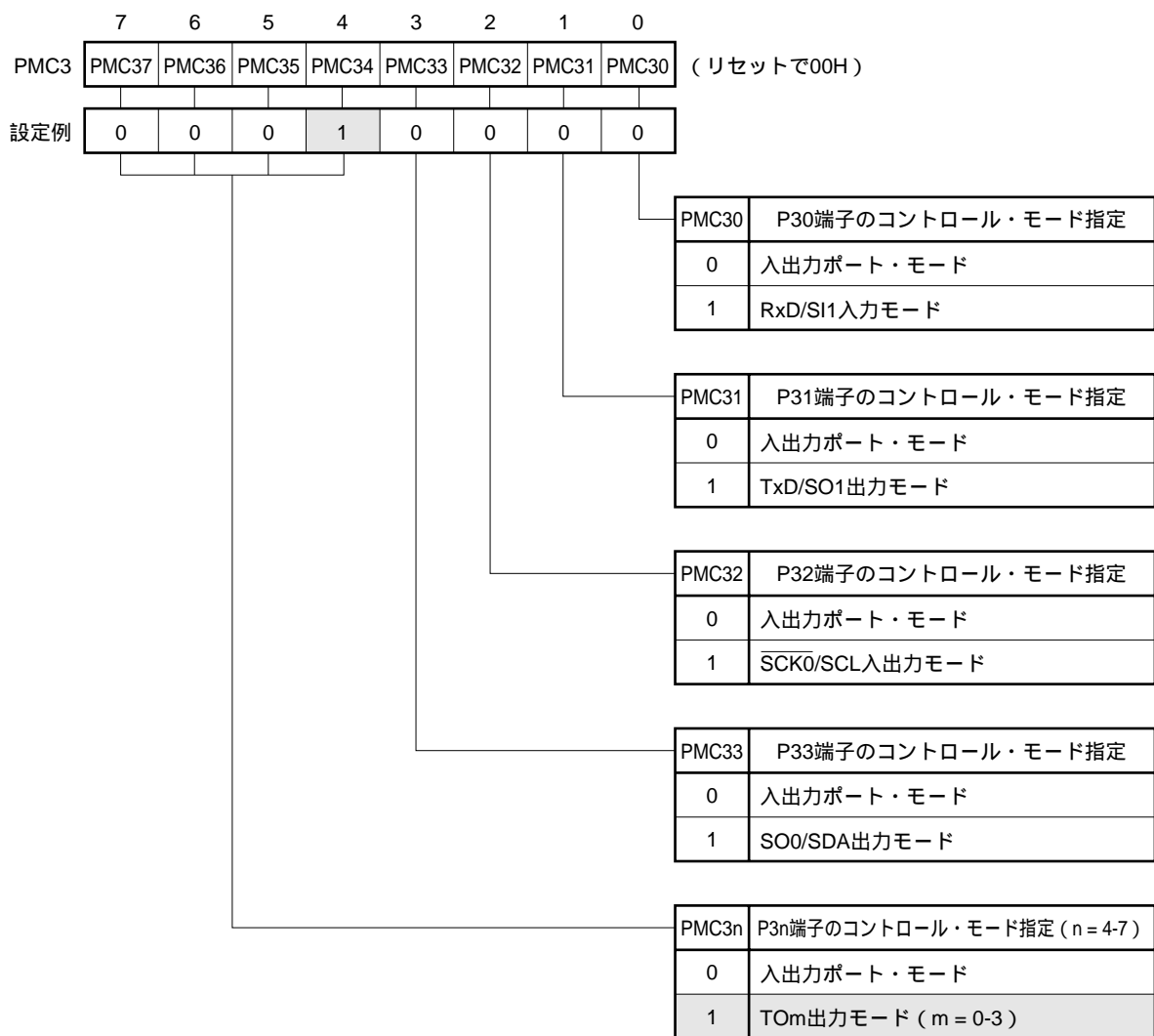
備考 OVF0ビットはソフトウェアでのみリセットされます。

プリスケアラ・モード・レジスタ0 (PRM0)



備考 f_{xx} : クリスタル/セラミック発振周波数

ポート3モード・コントロール・レジスタ (PMC3)



(4) 入力方法

RAM領域に定義した2バイトのワーク領域に次の内容を設定します。

DUTY1：パルス幅を決定する設定値を定義します。コンペア・レジスタ (CR00) の設定値を示します。

(5) 出力方法

なし

(6) プログラム説明

イニシャライズ処理 [レーベル名称：PWM00]

- (a) タイマ・レジスタ0 (TM0) のカウント動作を停止します。
- (b) タイマ出力コントロール・レジスタ (TOC) でタイマ出力TO0端子のアクティブ・レベルをハイ・レベルに設定し、パルス出力を許可にします。

注意 PWM/PPG出力では、タイマ・コントロール・レジスタ (TOC) のALV0フラグを0に設定した場合は、アクティブ・レベルはハイ・レベルとなります。

- (c) P34をTO0出力端子として用いるためにポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに指定します。

注意 P34端子を、ポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替える場合、出力信号にレベル差があるとTO0出力モード設定時に不定なレベルが出力され、エッジ割り込みが発生するなどの誤動作をする場合が考えられます。したがって必ず先にTO0端子の出力アクティブ・レベルをタイマ出力コントロール・レジスタ (TOC) で設定したのち、ポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替えてください。

- (d) タイマ・レジスタ0 (TM0) とコンペア・レジスタ (CR00, CR01) の一致による、タイマ・レジスタ0 (TM0) のクリアを禁止 (フリーランニング・モード) します。また、TO0端子をPWM出力に設定します。
- (e) コンペア・レジスタ (CR00) にPWM信号のパルス幅を決定する値を設定します。
- (f) タイマ・レジスタ0 (TM0) のカウント動作を許可します。

INTC00割り込み処理 [レベル名称 : INTC00]

- (a) レジスタ・バンク 1 に切り替えます。
- (b) パルス幅変更要求フラグ (CNGFLAG) でPWMのパルス幅の変更要求のある / なしをチェックします。

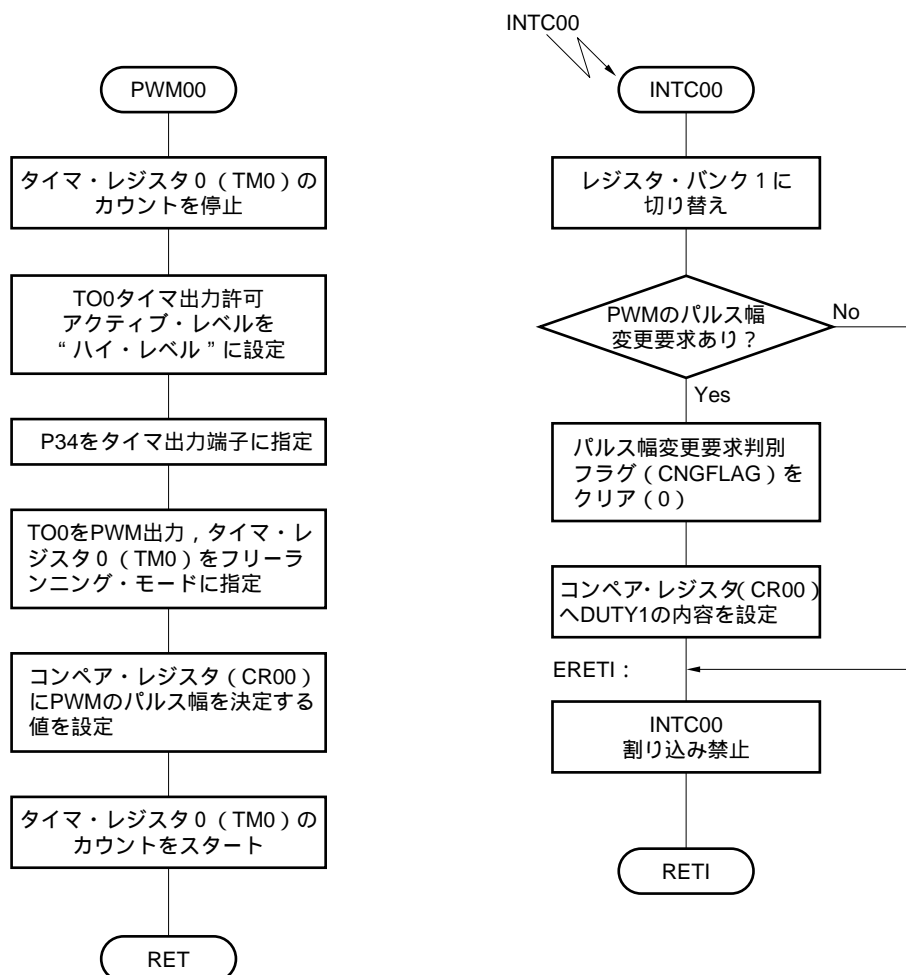
CNGFLAG = 0 : パルス幅変更要求なし (d) の処理へジャンプする。

CNGFLAG = 1 : パルス幅変更要求あり (c) の処理へ進む。

(c) 変更するパルス幅の設定値をコンペア・レジスタ (CR00) に書き込みます。

(d) INTC00割り込み要求をマスクします。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

DUTY1 : パルス幅 (CR00) を決定する設定値
 CNGFLAG : パルス幅変更判別フラグ
 CNGFLAG = 0...デューティ変更要求なし
 CNGFLAG = 1...デューティ変更要求あり

メイン・ルーチンのプログラム・リスト記述例

```

      .
      .
      CSEG
OUT00: LOCATION OH
      .
      MOVW    DUTY1,#7FFFH    ; TMO PWM first duty
      CALL    !PWM00
      EI
      .
;      <<デューティ変更要求>>
INTP4: SEL     RB1             ;
      SET1    CNGFLAG         ;
      MOV     A,ADCR          ; A/D conversion result read
      MOV     X,#01H          ;
      MOVW   DUTY1,AX        ; set PWM duty
      .
      .

```

本アプリケーション・ルーチンのプログラム・リスト

```

NAME      PWM_0
;
;*****
;*          PWM output          *
;*****
;
PUBLIC    PWM00
EXTRN     DUTY1
EXTBIT    CNGFLAG
INTC20VT  CSEG  AT 000EH      ; INTC00 vector
          DW     INTC00
;
          CSEG
PWM00:
MOV       TMC0,#00000000B ; timer stop
MOV       TOC,#00000010B ; timer output,active level high
MOV       PMC3,#00010000B ; P3 control port
MOV       CRC0,#01010000B ; TMO free running mode
MOV       PRM0,#00010001B ; TMO prescaler fxx/8
MOVW      AX,DUTY1        ; set first data of duty
MOVW      CR00,AX
MOV       TMC0,#00001000B ; start timer
RET
;
;*****
;*          INTC00 interrupt routine          *
;*****
;
INTC00:
SEL       RB1
BF        CNGFLAG,$ERETI  ; PWM duty change ?
CLR1     CNGFLAG          ;
MOVW     AX,DUTY1         ; [yes] set next data of duty
MOVW     CR00,AX          ;
ERETI:
SET1     CMK00            ; [no] mask INTC00
RETI
;
          END

```

タイマ/カウンタを8ビット動作モードで使用する場合

タイマ/カウンタ2は、タイマ機能を8ビット動作モードで 사용할 ことができます。8ビット動作モードの使用方法は、基本的に16ビット動作モードと同じです。

次に、8ビット動作モードの使用方法を示します。

- ・タイマ・コントロール・レジスタ (TMC0, TMC1) で8ビット/16ビット動作モードを指定します。

	7	6	5	4	3	2	1	0
TMC1	CE2	OVF2	CMD2	BW2	CE1	OVF1	0	BW1

TM2のビット長指定^注

- 注** 0 : 8ビット動作モード
1 : 16ビット動作モード

- ・PWM信号の1周期は、タイマ・レジスタ2 (TM2) の下位8ビットがフルカウント (FFH) する期間になります。
また、パルス幅を変更する場合は、タイマ・レジスタ2 (TM2) とコンペア・レジスタ (CR20) のそれぞれ下位8ビットの一致による割り込み要求を発生します。その割り込み処理内で、パルス幅を決定するコンペア・レジスタ (CR20) の設定値 (下位8ビット) を書き換えます。
- ・コンペア・レジスタの下位8ビットにパルス幅を決定する設定値を設定します。

<p>パルス周期 = $256 \times x / f_{xx}$ パルス幅 = $n \times x / f_{xx}$ $\{n: 1 \quad n \quad FFH\}$ (f_{xx} = クリスタル/セラミック発振周波数) n = コンペア・レジスタ設定値 $x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048$ (x = クリスタル/セラミック発振周波数の分周比)</p>

2.6 PPG出力の応用

PPG出力は、タイマ割り込み要求によりパルス幅を制御する矩形波出力です。

PPG出力では1つのタイマから発生する2系統の一致信号のうち、一方によって1周期が決定されます。

(1) 処理概要

(a) 仕様

TO0端子から、INTC01割り込み要求によって周期を、INTC00割り込み要求によってパルス幅を決定するPPG信号を出力するプログラム例を紹介します。

なお、パルス幅変更要求処理のプログラム例において、次のPPG出力のパルス幅は、A/Dコンバータ変換結果レジスタ (ADCR) の格納値を設定しています。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

コンペア・レジスタ (CR01) の設定値で決まる時間を1周期とし、コンペア・レジスタ (CR00) の設定値で決まる時間をパルス幅とする矩形波を出力するタイマ出力モードです。

図2 - 9に、TO0端子からのPPG出力の機能を示します。

図2 - 9 TO0端子からのPPG出力

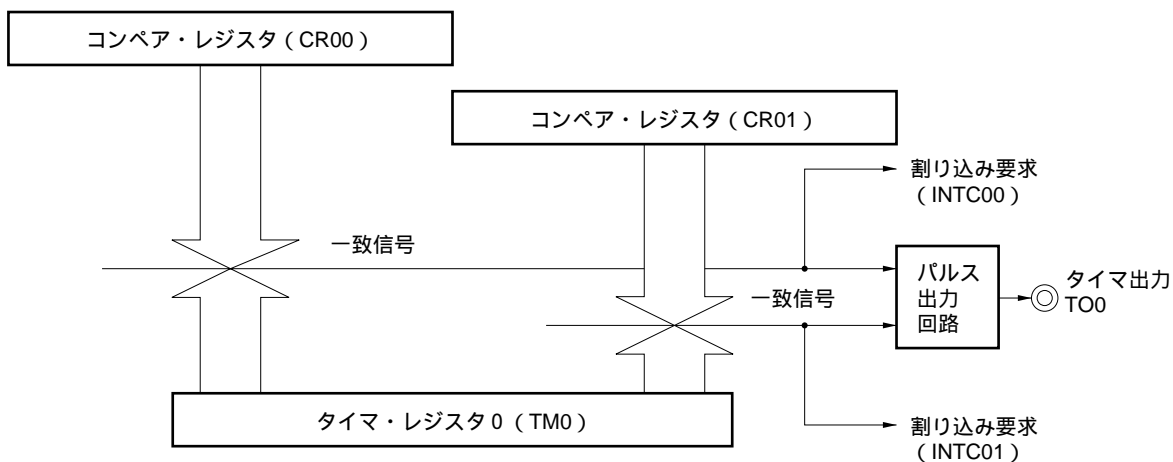
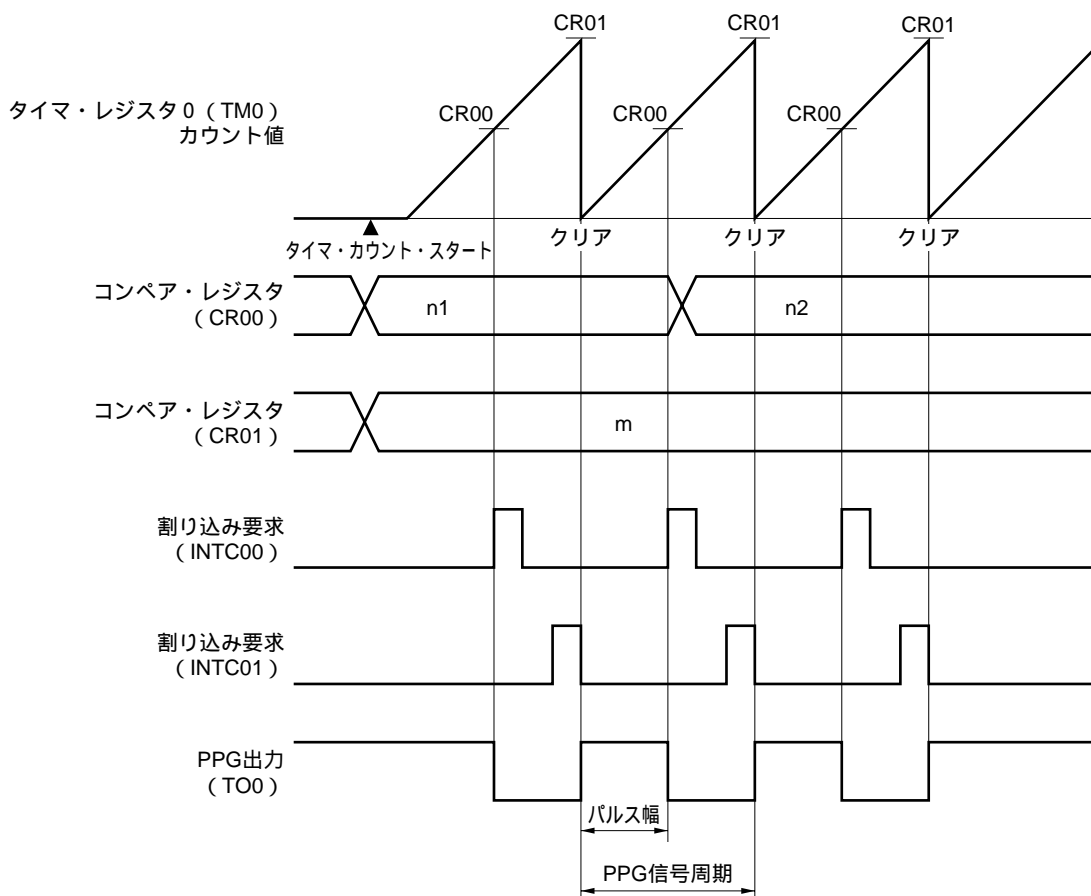


図2 - 10 TO0端子からのPPG出力のタイミング・チャート



備考 PPG信号周期 = $(m + 1) \times x / f_{xx}$ { m : 2 m FFFFH }

パルス幅 = $n \times x / f_{xx}$ { n : 1 n FFFFH }

(f_{xx} = クリスタル / セラミック発振周波数)

x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048

(x = クリスタル / セラミック発振周波数の分周比)

ただし、パルス幅 < PPG信号周期

PPG信号の1周期は、コンペア・レジスタ (CR01) の設定値とタイマ・レジスタ0 (TM0) のカウント値が一致するまでの期間になり、一致と同時にタイマ・レジスタ0 (TM0) のカウント値をクリアします。パルス幅は、コンペア・レジスタ (CR00) の設定値とタイマ・レジスタ0 (TM0) のカウント値が一致するまでの期間になります。

パルス幅変更の要求があった場合は、タイマ・レジスタ0 (TM0) のカウント値とコンペア・レジスタ (CR00) の設定値の一致によるINTC00割り込み要求発生時に、割り込み処理内で、パルス幅を決定するコンペア・レジスタ (CR00) の設定値を書き換えます。

(2) 使用RAM領域

本プログラムでは、表2 - 3に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表2 - 3 TM0によるPPG出力のプログラムで用いるワーク/フラグ

RAM/フラグ名称	用途	バイト数	初期値
DUTY2	パルス幅を決定する設定値を定義するRAM領域	2	7FFFH
CYCL0	TO0端子からのPPGの周期を定義するRAM領域	2	59999
CNGFLAG	PPGのパルス幅変更判別フラグ CNGFLAG = 0...変更要求なし CNGFLAG = 1...変更要求あり	1ビット	0

(3) 使用レジスタ

(a) 汎用レジスタ

イニシャライズ処理 [レーベル名称 : PPG00]

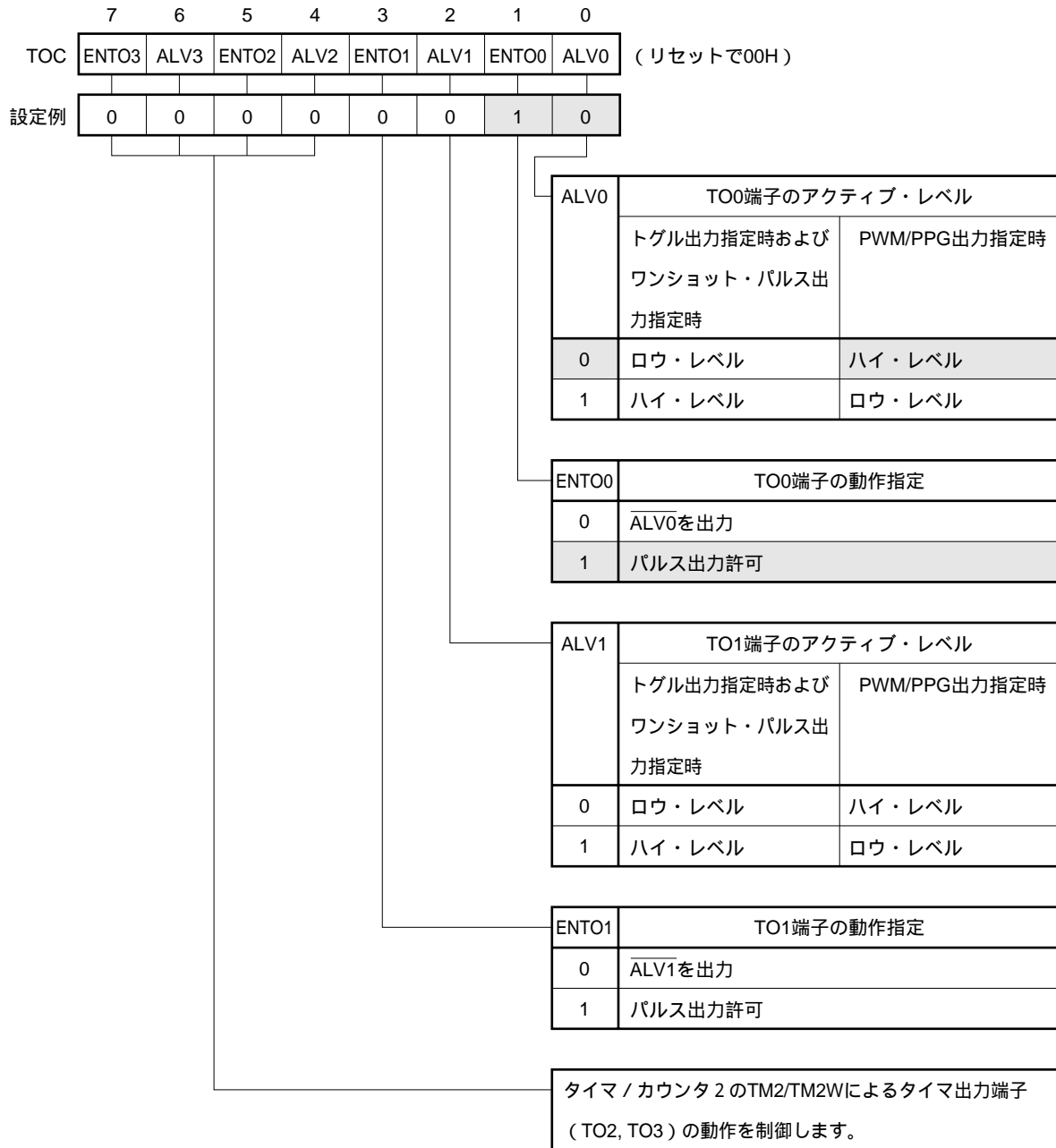
AXレジスタ

INTC00割り込み処理 [レーベル名称 : INTC00]

AXレジスタ

(b) 特殊機能レジスタ

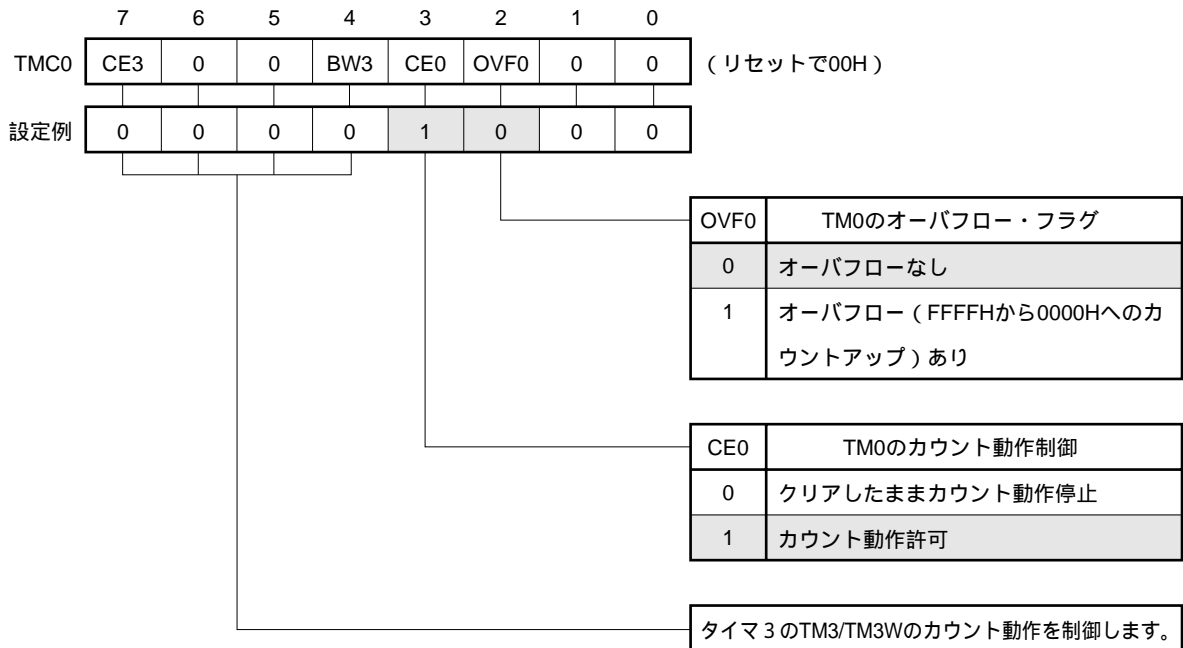
タイマ出力コントロール・レジスタ (TOC)



キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)



タイマ・コントロール・レジスタ0 (TMC0)



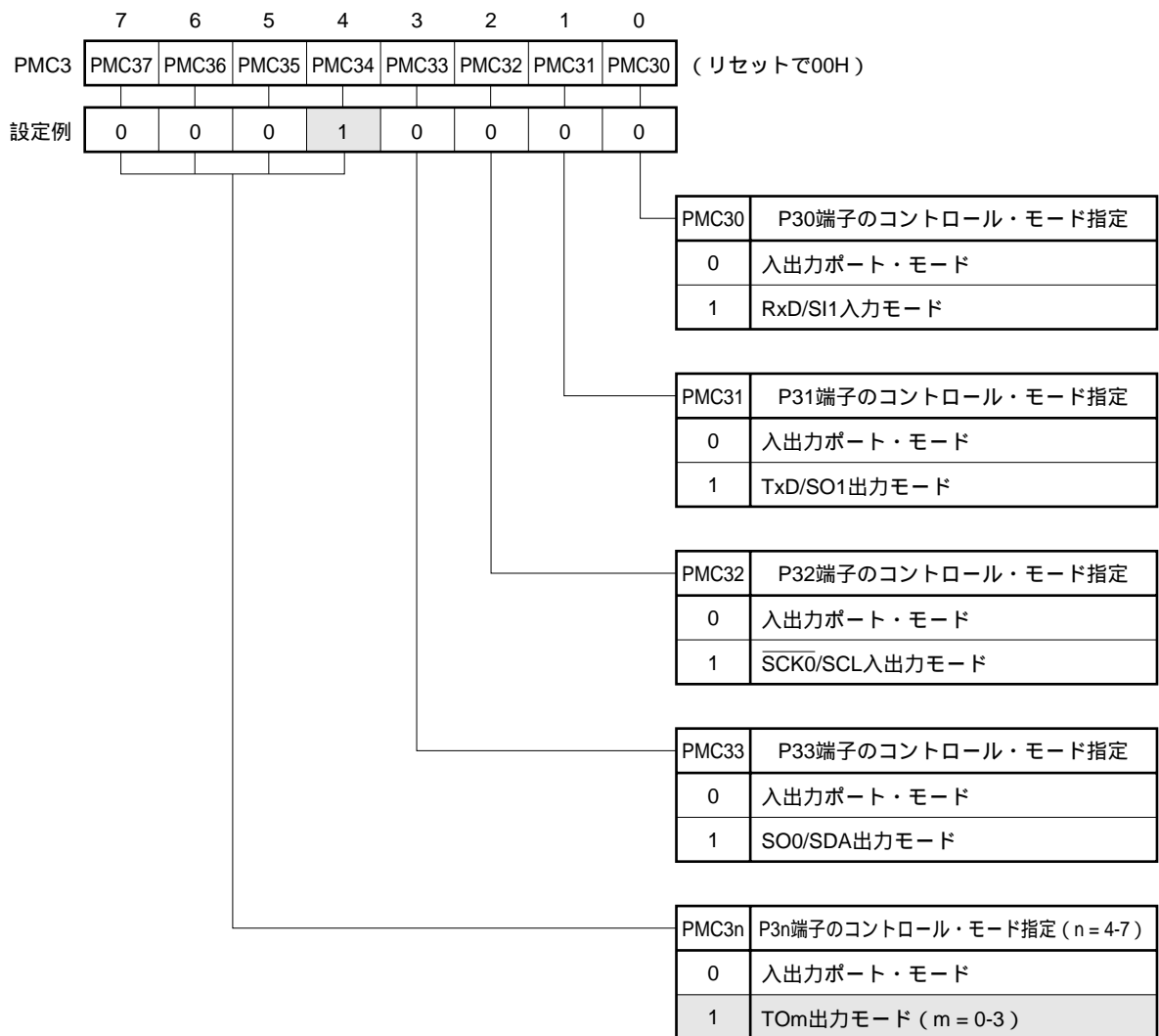
備考 OVF0ビットはソフトウェアでのみリセットされます。

プリスケアラ・モード・レジスタ0 (PRM0)



備考 f_{xx} : クリスタル/セラミック発振周波数

ポート3モード・コントロール・レジスタ (PMC3)



(4) 入力方法

RAM領域に定義した2バイトのワーク領域に次の内容を設定します。

DUTY2 : パルス幅を決定する設定値を定義します。コンペア・レジスタ (CR00) の設定値を示します。

CYCL0 : TO0端子からのPPGの周期を定義します。コンペア・レジスタ (CR00) の設定値を示します。

(5) 出力方法

なし

(6) プログラム説明

イニシャライズ処理 [レーベル名称 : PPG00]

(a) タイマ・レジスタ 0 (TM0) のカウント動作を停止します。

(b) タイマ出力コントロール・レジスタ (TOC) でタイマ出力TO0端子のアクティブ・レベルをハイ・レベルに設定し、パルス出力を許可にします。

注意 PWM/PPG出力では、タイマ・コントロール・レジスタ (TOC) のALV0フラグを0に設定した場合は、アクティブ・レベルはハイ・レベルとなります。

(c) P34をTO0出力端子として用いるためにポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに指定します。

注意 P34端子を、ポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替える場合、出力信号にレベル差があるとTO0出力モード設定時に不定なレベルが出力され、エッジ割り込みが発生するなどの誤動作をする場合が考えられます。したがって必ず先にTO0端子の出力アクティブ・レベルをタイマ出力コントロール・レジスタ (TOC) で設定したのち、ポート3モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替えてください。

(d) タイマ・レジスタ 0 (TM0) とコンペア・レジスタ (CR00, CR01) の一致による、タイマ・レジスタ 0 (TM0) のクリアを許可し、TO0端子をPPG出力モードに設定します。

(e) コンペア・レジスタ (CR01) にPPGの周期を決定する値を設定します。また、コンペア・レジスタ (CR00) にパルス幅を決定する設定値を設定します。

(f) タイマ・レジスタ 0 (TM0) のカウント動作を許可します。

INTC00割り込み処理 [レーベル名称 : INTC00]

- (a) レジスタ・バンク 1 に切り替えます。
- (b) パルス幅変更要求フラグ (CNGFLAG) でPPGのパルス幅の変更要求のある / なしをチェックします。

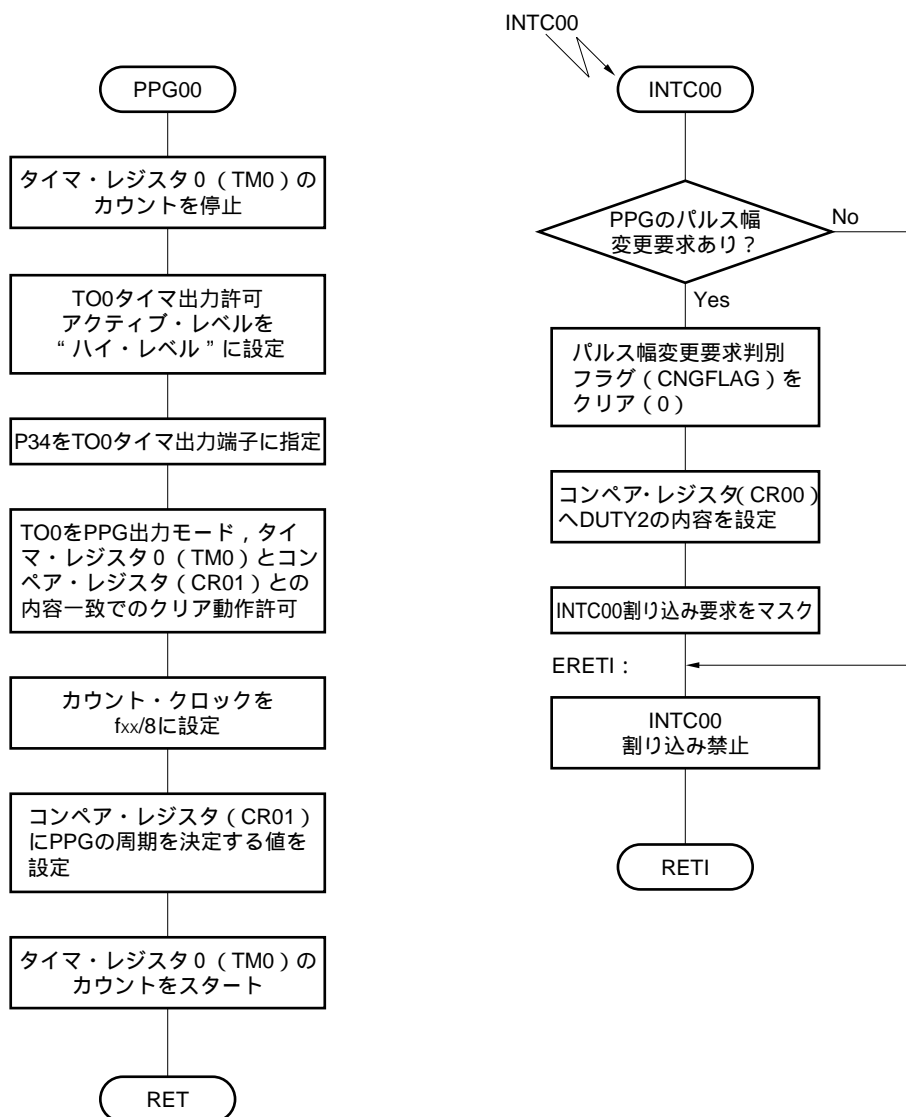
CNGFLAG = 0 : パルス幅変更要求なし (d) の処理へジャンプする。

CNGFLAG = 1 : パルス幅変更要求あり (c) の処理へ進む。

- (c) 変更するパルス幅の設定値をコンペア・レジスタ (CR00) に書き込みます。

- (d) INTC00割り込み要求をマスクします。

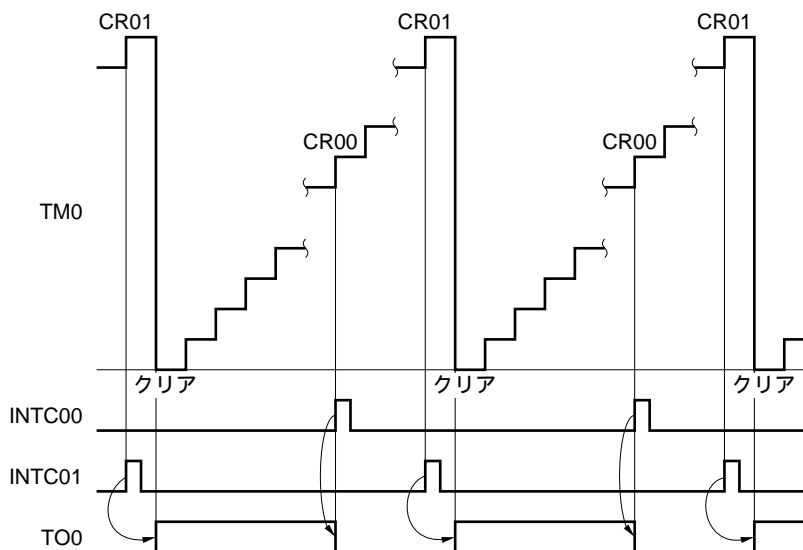
(7) フロー・チャート



(8) プログラム・リスト

PPG出力の場合、INTC00, INTC01割り込み要求の発生タイミングとPPG出力のタイミングは、図2-11のようになります。

図2-11 PPG出力タイミング(TM0)



アプリケーション・ルーチン時に使用するレーベルの説明

- DUTY2 : パルス幅を決定する設定値
- CYCL0 : TO0端子からのPPG出力周期を定義
- CNGFLAG : パルス幅変更判別フラグ
 - CNGFLAG = 0...デューティ変更要求なし
 - CNGFLAG = 1...デューティ変更要求あり

メイン・ルーチンのプログラム・リスト記述例

```
      .
CYCLO EQU    59999          ; TM0 PPG out CR01 data
      .
      .
      CSEG
OUT00:
      LOCATION 0H
      .
      .
      MOVW    DUTY2,#7FFFH   ; TM0 PPG first duty
      CALL    !PPG00
      EI
      .
      .
;      <<デューティ変更要求>>
INTP4:
      SEL     RB1            ;
      SET1    _CHANGE        ;
      MOV     A,ADCR         ; A/D conversion result read
      MOV     X,#01H
      .
      .
      BNC     $INTP41        ; [NO]
      BR      $JPRETI        ; [YES]
INTP41:
      MOVW    DUTY2,AX       ; set PPG duty
      .
      .
```

本アプリケーション・ルーチンのプログラム・リスト

```

NAME      PPG_0
;
; *****
; * 16bit-Timer / Counter *
; *   PPG output *
; *       cycle : 12.5Hz *
; *****
;
PUBLIC    PPG00
EXTRN    CYCLO,DUTY2
EXTBIT    CNGFLAG
;
;
;
INTC00VT CSEG    AT 0000EH      ; INTC00 vector
        DW      INTC00
;
;
        CSEG
PPG00:
MOV      TOC,#00000010B ; timer output,active level high
MOV      PMC3,#00010000B ; P3 control port
MOV      CRC0,#11011000B ; TM0 PPG output mode
MOV      PRM0,#00010001B ; TM0 prescaler fxx/8
MOVW     CR01,#CYCLO      ; set output cycle
MOVW     AX,DUTY2         ; set first output duty
MOVW     CR00,AX
MOV      TMC0,#00001000B ; start timer
RET
;
; *****
; *                   INTC00 interrupt routine *
; *****
;
INTC00:
SEL      RB1
BF       CNGFLAG,$ERETI ; PWM duty change ?
CLR1     CNGFLAG
MOVW     AX,DUTY2
MOVW     CR00,AX
;
ERETI:
SET1     CMK00           ; [no] mask INTC00
RETI
;
END

```

タイマ/カウンタを8ビット動作モードで使用する場合

タイマ/カウンタ2は、タイマ機能を8ビット動作モードで 사용할 ことができます。
 8ビット動作モードの使用方法は、基本的に16ビット動作モードと同じです。
 次に、8ビット動作モードの使用方を示します。

- ・タイマ・コントロール・レジスタ (TMC0, TMC1) で8ビット/16ビット動作モードを指定します。

	7	6	5	4	3	2	1	0
TMC1	CE2	OVF2	CMD2	BW2	CE1	OVF1	0	BW1

TM2のビット長指定^注

- 注** 0 : 8ビット動作モード
 1 : 16ビット動作モード

- ・PPG信号の1周期は、タイマ・レジスタ2 (TM2) のカウンタ値とコンペア・レジスタ (CR21) の設定値のそれぞれ8ビットが一致するまでの期間になります。
 タイマは一致によるクリアを許可にします。
 また、パルス幅を変更する場合は、タイマ・レジスタ2 (TM2) とコンペア・レジスタ (CR20) のそれぞれ下位8ビットの一致によって割り込み要求を発生します。その割り込み処理内で、パルス幅を決定するコンペア・レジスタの設定値を書き換えます。
- ・コンペア・レジスタ (CR20) の下位8ビットにパルス幅を決定する設定値を設定します。

<p>パルス周期 = (m + 1) × x / f_{xx}</p> <p style="text-align: center;">{ m : 2 m FFH }</p> <p>パルス幅 = n × x / f_{xx}</p> <p style="text-align: center;">{ n : 1 n FFH }</p> <p>(f_{xx} = クリスタル / セラミック発振周波数)</p> <p>n = コンペア・レジスタ (CR20) 設定値</p> <p>m = コンペア・レジスタ (CR21) 設定値</p> <p>x = 8, 16, 32, 64, 128, 256, 512, 1024, 2048</p> <p>(x = クリスタル / セラミック発振周波数の分周比)</p>

2.7 ソフト・トリガド・ワンショット・パルス出力の応用

ソフト・トリガド・ワンショット・パルス出力機能は、タイマ/カウンタ0のみが備えている機能で、ソフトウェアでトリガを設定し、ワンショット・パルスをタイマ出力（TO0, TO1）します。

（1）処理概要

（a）仕様

TO0端子から500 μ sのワンショット・パルスを出力するプログラム例を紹介します。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注/32

注 f_{xx} : クリスタル/セラミック発振周波数

- ・ 入力パラメータ（OSPWID）：下記計算式参照

$$\begin{aligned}
 \text{OSPWID} &= \frac{500 \mu\text{s}}{\text{カウント} \cdot \text{クロック}} \\
 &= \frac{500 \times 10^{-6}}{32 / (25 \times 10^6)} \\
 &= 391 \text{ (小数点以下四捨五入)}
 \end{aligned}$$

（b）使用周辺機能説明

ソフト・トリガド・ワンショット・パルス出力は、ワンショット・パルスをソフトウェアでトリガをかけて出力するモードです。ソフトウェア・トリガ（ワンショット・パルス出力制御レジスタ（OSPC）のST0フラグをセットする）によって、TO0端子からアクティブ・レベルが出力されます。タイマ・レジスタ0（TM0）のカウント値とコンペア・レジスタ（CR00）の設定値が一致すると、TO0端子の出力は反転します。INTC00割り込み要求でタイマ・レジスタ0（TM0）はカウントを停止します。TO0端子、TO1端子に対して独立に制御することができます。

図2-12に、TO0端子からの矩形波出力の機能のブロック図を示します。

図2 - 12 TO1端子からの矩形波出力

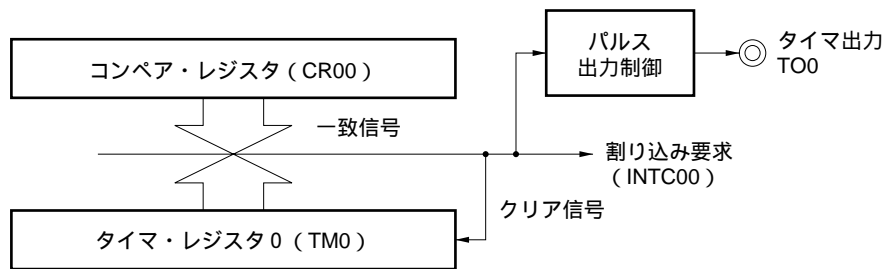
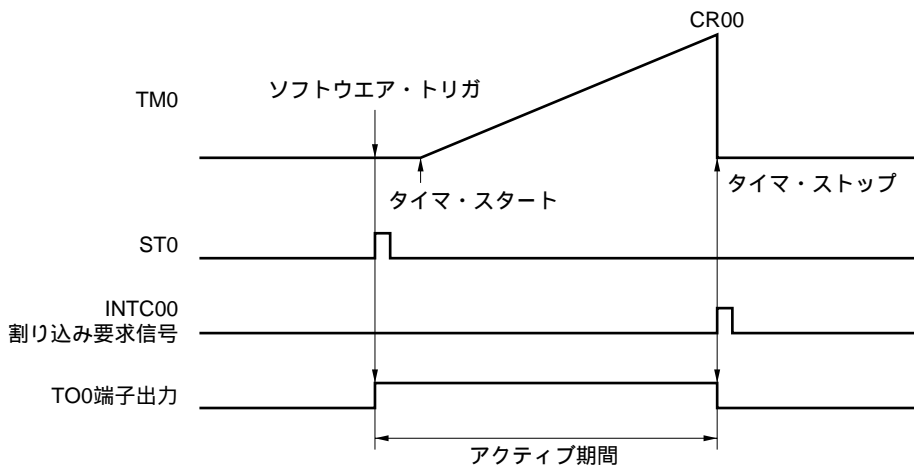


図2 - 13に、TO0端子からワンショット・パルスを出力する例を示します。

図2 - 13 TO0端子からのワンショット・パルス出力例



備考 ALV0 = 1 (アクティブ・ハイ) の場合

(2) 使用RAM領域

なし

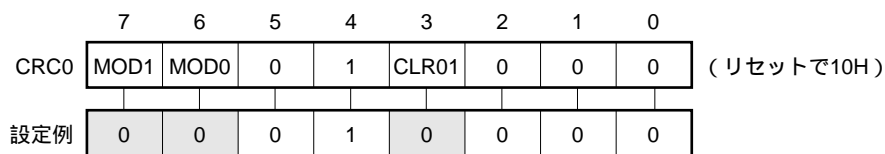
(3) 使用レジスタ

(a) 汎用レジスタ

なし

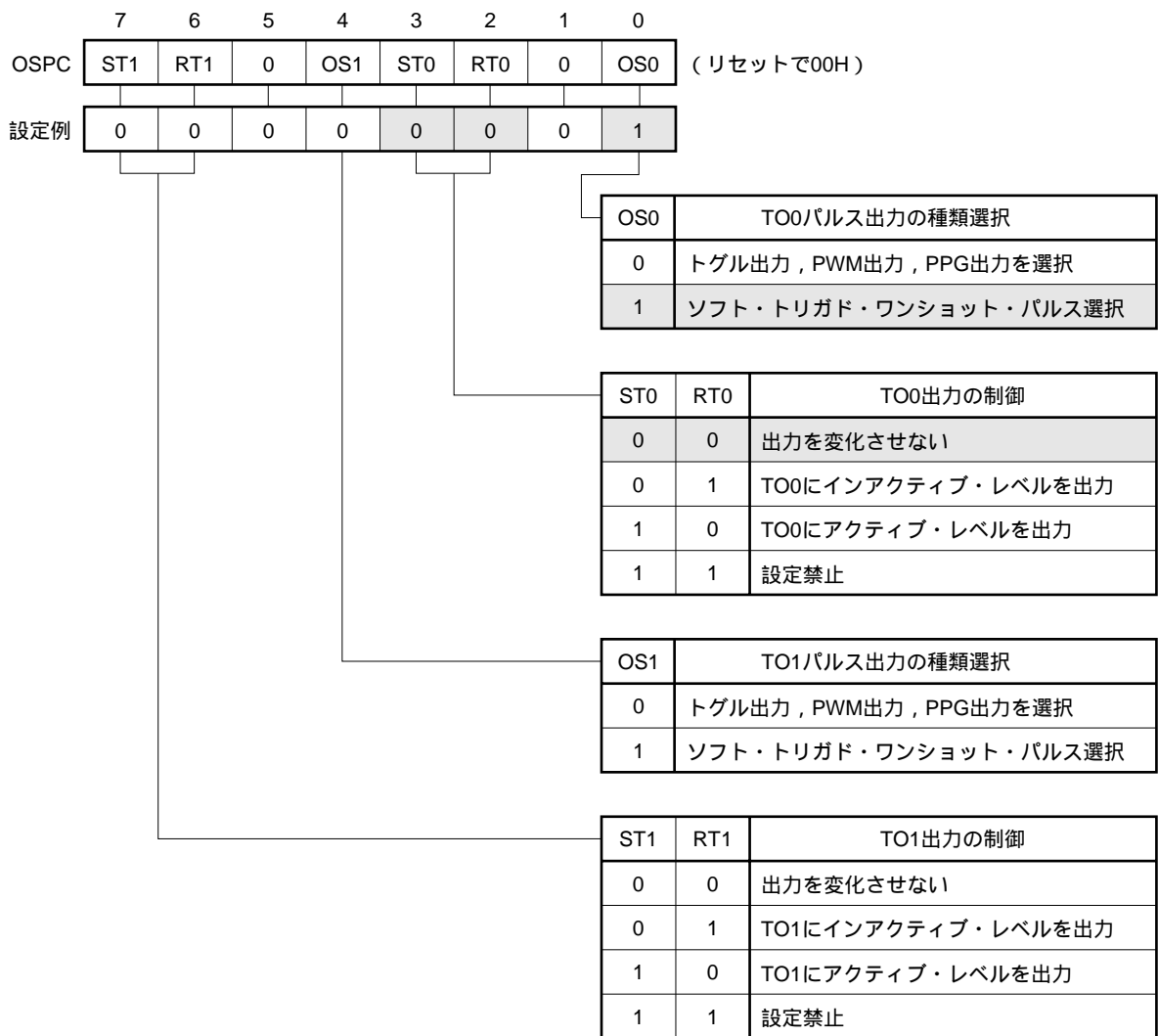
(b) 特殊機能レジスタ

キャプチャ/コンペア・コントロール・レジスタ0 (CRC0)



MOD1	MOD0	CLR01	タイマ出力モード指定		TM0 = CR01時の TM0のクリア動作
			TO0	TO1	
0	0	0	トグル出力	トグル出力	禁止
0	0	1	トグル出力	トグル出力	許可
0	1	0	PWM出力	トグル出力	禁止
0	1	1	設定禁止		
1	0	0	PWM出力	PWM出力	禁止
1	0	1	設定禁止		
1	1	0	設定禁止		
1	1	1	PPG出力	トグル出力	許可

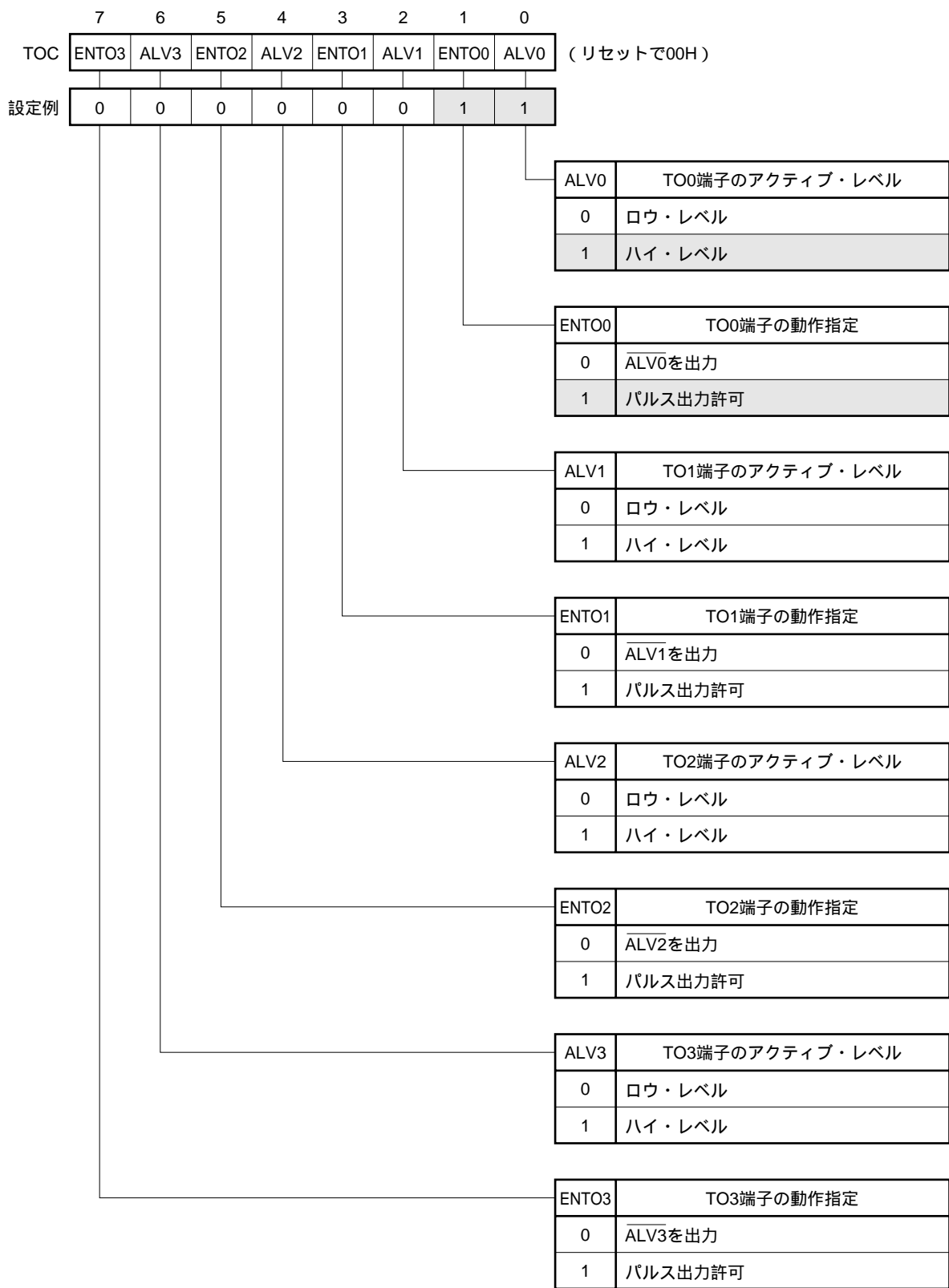
ワンショット・パルス出力制御レジスタ (OSPC)



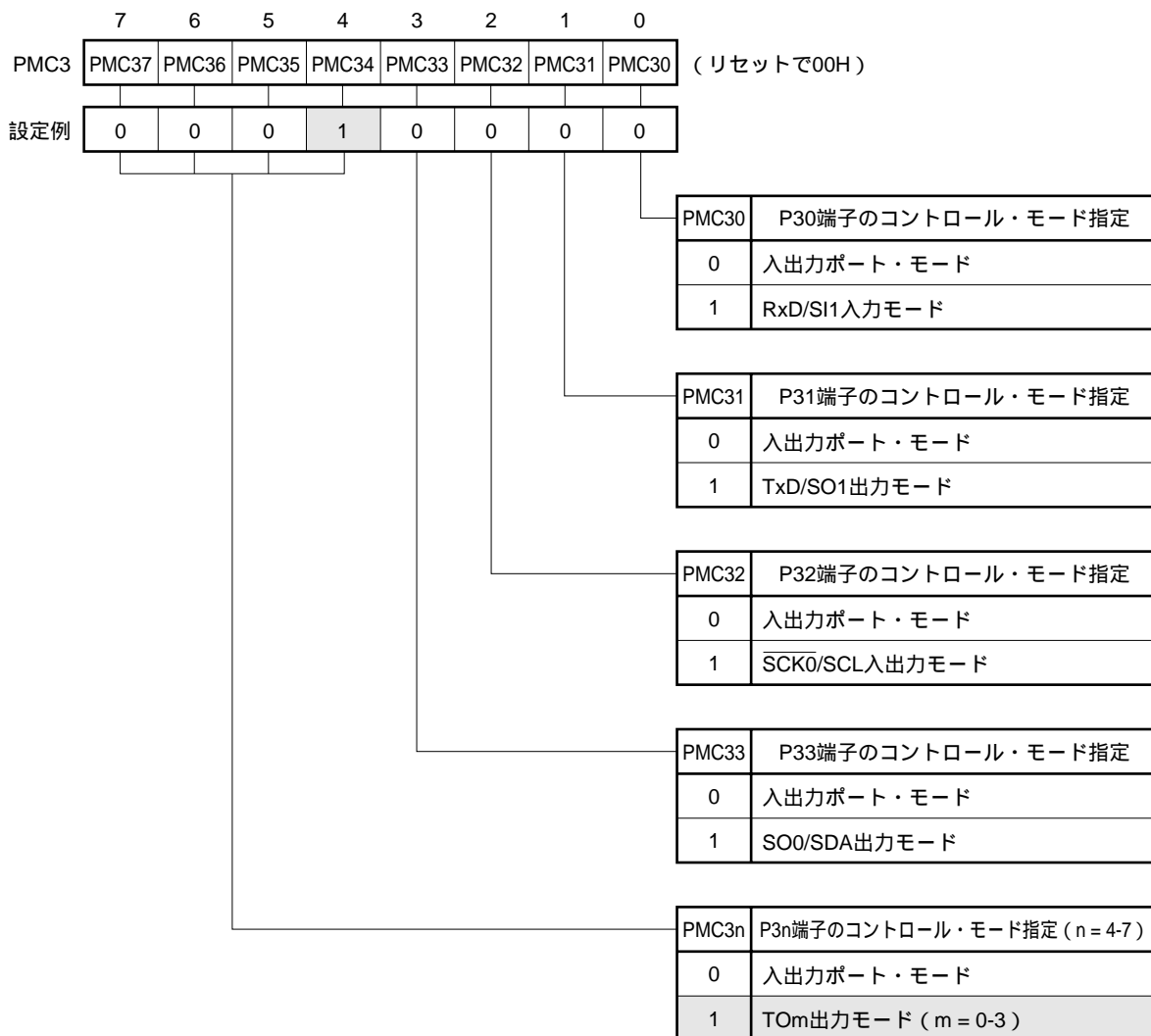
備考1 . RT0, ST0, RT1, ST1の各ビットは書き込みのみが可能で, 読み出し時は“ 0 ”になります。

2 . 端子からのパルス出力の禁止/許可およびアクティブ・レベルの指定は, タイマ出力制御レジスタ (TOC) で行います。

タイマ出力コントロール・レジスタ (TOC)



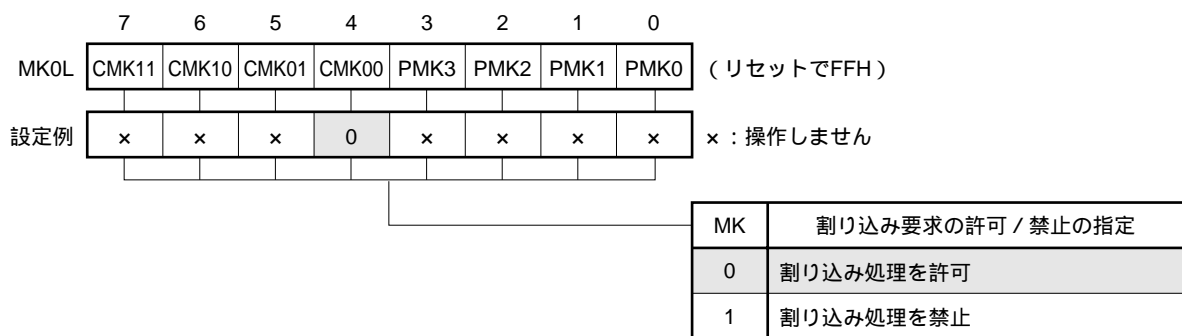
ポート3モード・コントロール・レジスタ (PMC3)



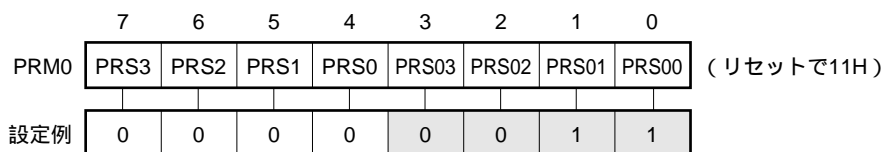
タイマ・コントロール・レジスタ0 (TMC0)



割り込みマスク・レジスタL (MK0L)



プリスケアラ・モード・レジスタ0 (PRM0)



(fxx = 25 MHz)

PRS3	PRS2	PRS1	PRS0	タイマ/カウンタ0のTM0のカウント・クロック指定	分解能
				カウント・クロック [Hz] の指定	
0	0	0	0	設定禁止	-
0	0	0	1	fxx/8	0.32 μs
0	0	1	0	fxx/16	0.64 μs
0	0	1	1	fxx/32	1.28 μs
0	1	0	0	fxx/64	2.56 μs
0	1	0	1	fxx/128	5.12 μs
0	1	1	0	fxx/256	10.2 μs
0	1	1	1	fxx/512	20.5 μs
1	0	0	0	fxx/1024	41.0 μs
1	0	0	1	fxx/2048	81.9 μs
1	1	1	1	外部クロック (INTP3)	-
上記以外				設定禁止	

(fxx = 25 MHz)

PRS3	PRS2	PRS1	PRS0	タイマ3のTM3/TM3Wのカウント・クロック指定	分解能
				カウント・クロック [Hz] の指定	
0	0	0	0	設定禁止	-
0	0	0	1	fxx/8	0.32 μs
0	0	1	0	fxx/16	0.64 μs
0	0	1	1	fxx/32	1.28 μs
0	1	0	0	fxx/64	2.56 μs
0	1	0	1	fxx/128	5.12 μs
0	1	1	0	fxx/256	10.2 μs
0	1	1	1	fxx/512	20.5 μs
1	0	0	0	fxx/1024	41.0 μs
1	0	0	1	fxx/2048	81.9 μs
上記以外				設定禁止	

(4) 入力方法

下記に示すレーベルに次の内容を定義します。

OSPWID : パルス幅を決定する設定値を定義します。コンペア・レジスタ (CR00) の設定値を示します。

(5) 出力方法

なし

(6) プログラム説明

イニシャライズ処理 [レーベル名称 : OSPOUT]

- (a) タイマ・レジスタ 0 (TM0) のカウント動作を停止します。
- (b) タイマ出力コントロール・レジスタ (TOC) でタイマ出力TO0端子のアクティブ・レベルをハイ・レベルに設定し、パルス出力を許可します。
- (c) P34をTO0出力端子として用いるためにポート 3 モード・コントロール・レジスタ (PMC3) でTO0出力モードに指定します。

注意 P34端子を、ポート 3 モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替える場合、出力信号にレベル差があるとTO0出力モード設定時に不定なレベルが出力され、エッジ割り込みが発生するなどの誤動作をする場合が考えられます。したがって必ず先にTO0端子の出力アクティブ・レベルをタイマ出力コントロール・レジスタ (TOC) で設定したのち、ポート 3 モード・コントロール・レジスタ (PMC3) でTO0出力モードに切り替えてください。

- (d) タイマ・レジスタ 0 (TM0) のカウント動作を停止します。
- (e) ワンショット・パルス出力制御レジスタ (OSPC) でTO0端子出力をソフト・トリガド・ワンショット・パルスに設定します。
- (f) コンペア・レジスタ (CR00) にワンショット・パルスのパルス幅を設定します。
- (g) タイマ・レジスタ 0 (TM0) のカウント・クロックを $f_{xx}/32$ に指定します。
- (h) ソフトウェア・トリガとしているINTP5の割り込み要求のマスクを解除します。
- (i) INTC00割り込み要求のマスクを解除します。

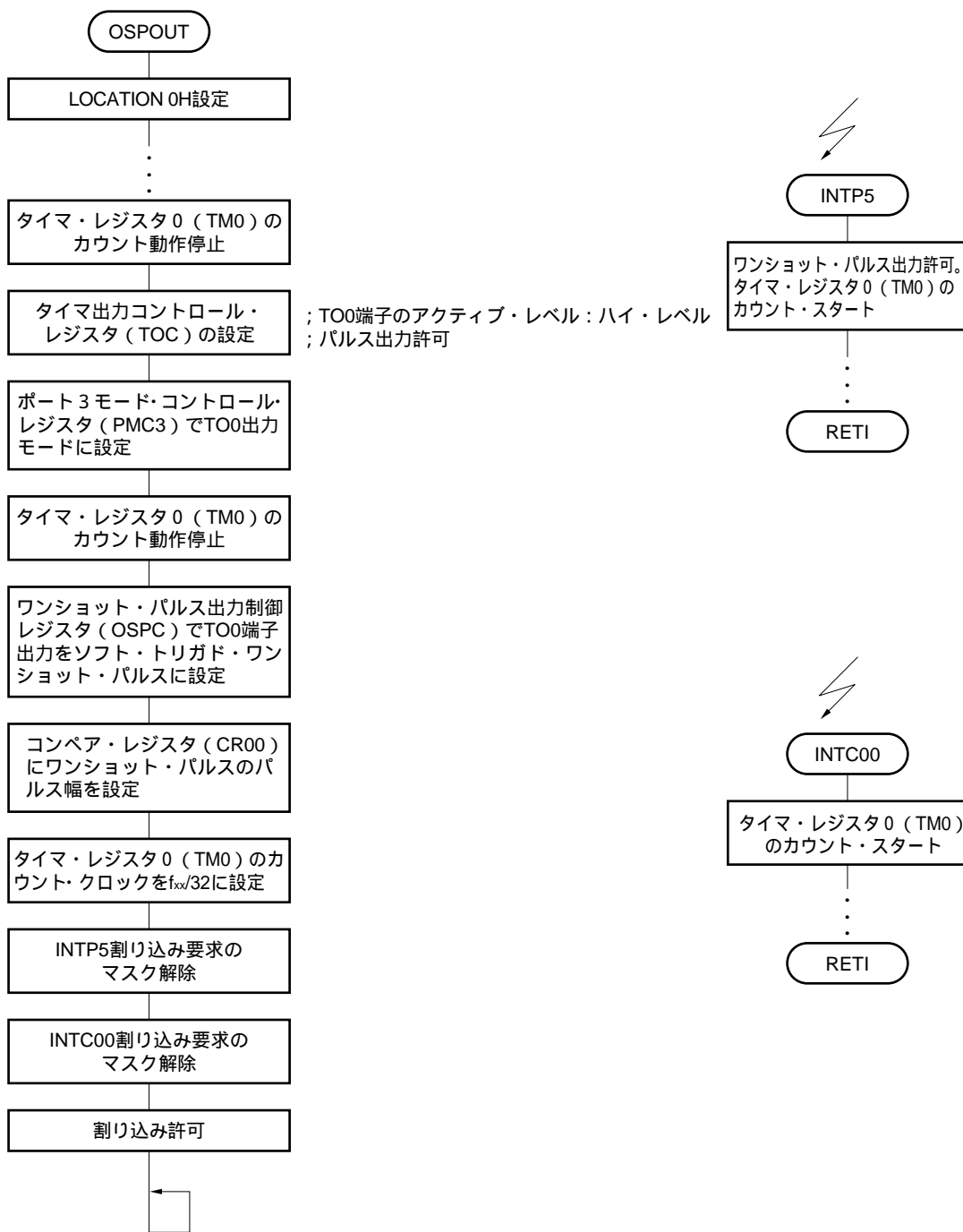
ソフトウェア・トリガ設定処理 [レーベル名称 : INTP5]

ワンショット・パルス出力許可にし、タイマ・レジスタ 0 (TM0) のカウントを開始します。

INTC00割り込み処理 [レーベル名称 : INTC00]

タイマ・レジスタ 0 (TM0) のカウント動作を停止します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

OSPWID : パルス幅 (CR00) を決定する設定値

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
OSPWID EQU 391 ; one-shot pulse width

INTP5VT CSEG AT 001EH ; INTP5 vector table
      DW INTP5
INTC00VT CSEG AT 000EH ; INTC00 vector table
      DW INTC00
      .
      .
; *****
; * main routine *
; *****
;

      CSEG

OSPOUT:
      LOCATION OH
      MOVG SP,#STACK ; stack pointer

      MOV TMC0,#00000000B ; timer stop
      MOV TOC,#00000011B ; set T00 high active, enable output
      MOV PMC3,#00010000B ; set T00 output mode
      MOV CRC0,#00010000B ; set T00 timer out, disable clear TMO
      MOV OSPC,#00000001B ; set T00 one-shot pulse output mode
      MOVW CRO0,#OSPWID-1 ; set one-shot pulse width
      MOV PRMO,#00000011B ; set prescaler fxx/32
      CLR1 PMK5 ; open INTP5 mask
      CLR1 CMK00 ; open INTC00 mask
      EI ; enable interrupt
      .
      .
; *****
; * INTP5 interrupt routine *
; * --- request OSP out --- *
; *****
;
INTP5:
      SET1 ST0 ; output active level from T00
      MOV TMC0,#00001000B ; start TMO
      .
      .
      RETI

; *****
; * INTC00 interrupt routine *
; * --- stop OSP out --- *
; *****
;
INTC00:
      MOV TMC0,#00000000B ; stop TMO
      .
      .
      RETI

      END

```

2.8 ワンショット・タイマ機能応用

ワンショット・モードは、タイマ/カウンタ2が備えている動作モードで、タイマがフルカウント (FFFFH) になると自動的にカウント動作を停止します。

(1) 処理概要

(a) 仕様

タイマ/カウンタ2は、16ビット動作モードで使用します。

タイマ・レジスタ2 (TM2W) のワンショット・モードを利用して、P60端子から5種類のパルスを出力するプログラム例を紹介します。なお、タイマ・レジスタ2 (TM2W) のカウント動作を開始させるタイミングは、外部入力端子 (INTP4) の立ち上がりを検出したときとします。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : $f_{xx}/32$

注 f_{xx} : クリスタル/セラミック発振周波数

- ・ パルス幅 : コンペア・レジスタ (CR21W) の設定値は、それぞれ次の値を設定しています。

パルス幅	CR21W設定値
16 ms	0 3 0 D 3 H
17 ms	0 3 2 E 0 H
18 ms	0 3 6 E E H
19 ms	0 3 9 F B H
20 ms	0 3 D 0 8 H

- ・ パルス幅が16 msの場合

$$\frac{16 \times 10^{-3}}{32 / (25 \times 10^6)} = 12500$$

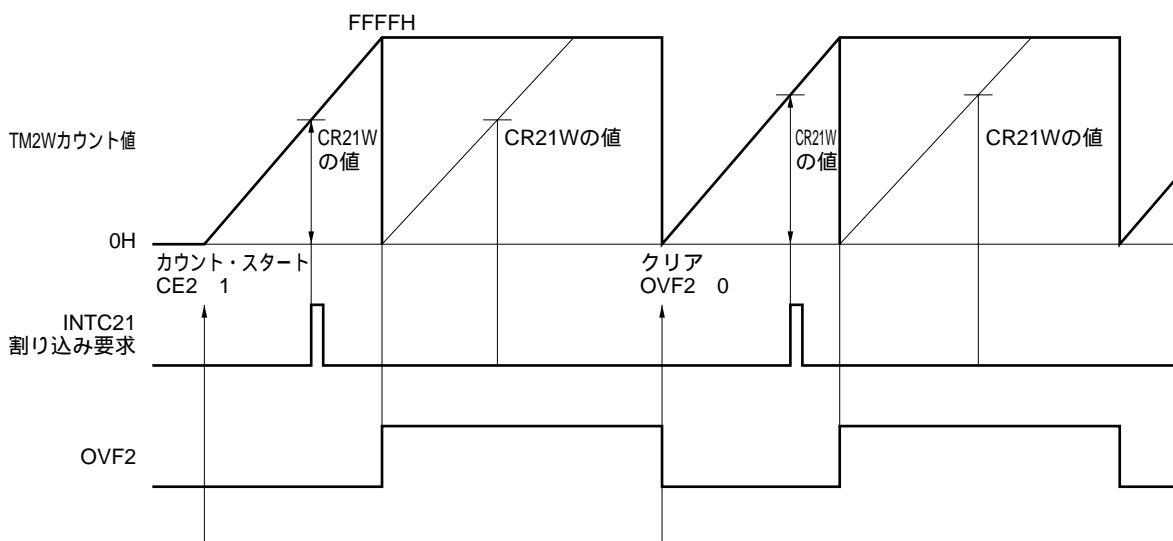
(= 030D3H)
($f_{xx} = 25 \text{ MHz}$)

(b) 使用周辺機能説明

タイマ・レジスタ2 (TM2W) のカウントの結果、フルカウントになると自動的に停止する動作モードです。コンペア・レジスタ (CR21W) の設定値と、タイマ・レジスタ2 (TM2W) のカウント値との一致で、ワンショット割り込みを発生します。

図2 - 14に、ワンショット・タイマ動作を示します。

図2 - 14 ワンショット・タイマ動作



タイマ・コントロール・レジスタ1 (TMC1) のCMD2フラグをセット (1) してワンショット・モードに設定すると、図2 - 14に示すように、コンペア・レジスタ (CR21W) の設定値 (0000H-FFFFH) とタイマ・レジスタ2 (TM2W) のカウント値との一致で、INTC21の割り込み要求を発生します。

タイマ・レジスタ2 (TM2W) のカウント動作は、タイマ・コントロール・レジスタ1 (TMC1) のCE2フラグで制御します。

CE2フラグをセット (1) すると、最初のカウント・クロックでタイマ・レジスタ2 (TM2W) の内容が00Hとなり、カウント動作を開始します。

カウント値がフルカウント (FFFFH) になると、タイマ・コントロール・レジスタ1 (TMC1) のOVF2フラグがセット (1) され、タイマ・レジスタ2 (TM2W) はカウント値がFFFFHのまま停止します。

カウント動作停止状態からカウント動作を開始させる場合は、OVF2フラグをクリア (0) してください。

(2) 使用RAM領域

なし

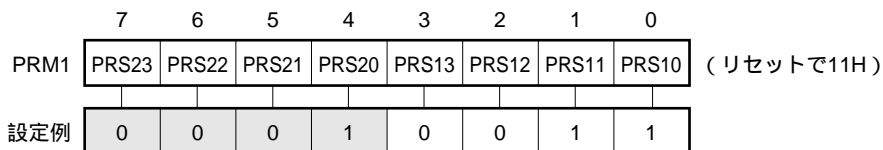
(3) 使用レジスタ

(a) 汎用レジスタ

AX, WHL, Bレジスタ

(b) 特殊機能レジスタ

プリスケアラ・モード・レジスタ1 (PRM1)



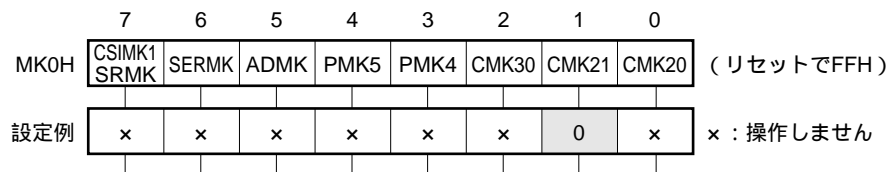
タイマ/カウンタ1のTM1/TM1Wのカウント・クロックを指定します。

($f_{xx} = 25 \text{ MHz}$)

PRS23	PRS22	PRS21	PRS20	タイマ/カウンタ2のTM2/TM2Wのカウント・クロック指定	
				カウント・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	$f_{xx}/8$	0.32 μs
0	0	1	0	$f_{xx}/16$	0.64 μs
0	0	1	1	$f_{xx}/32$	1.28 μs
0	1	0	0	$f_{xx}/64$	2.56 μs
0	1	0	1	$f_{xx}/128$	5.12 μs
0	1	1	0	$f_{xx}/256$	10.2 μs
0	1	1	1	$f_{xx}/512$	20.5 μs
1	0	0	0	$f_{xx}/1024$	41.0 μs
1	0	0	1	$f_{xx}/2048$	81.9 μs
1	1	1	1	外部クロック (CI/INTP2)	-
上記以外				設定禁止	

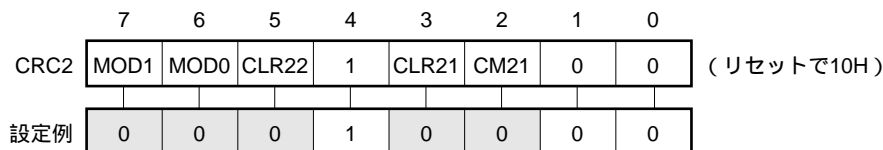
備考 f_{xx} : クリスタル/セラミック発振周波数

割り込みマスク・レジスタH (MK0H)



MK	割り込み要求の許可 / 禁止の指定
0	割り込み処理を許可
1	割り込み処理を禁止

キャプチャ/コンペア・コントロール・レジスタ2 (CRC2)

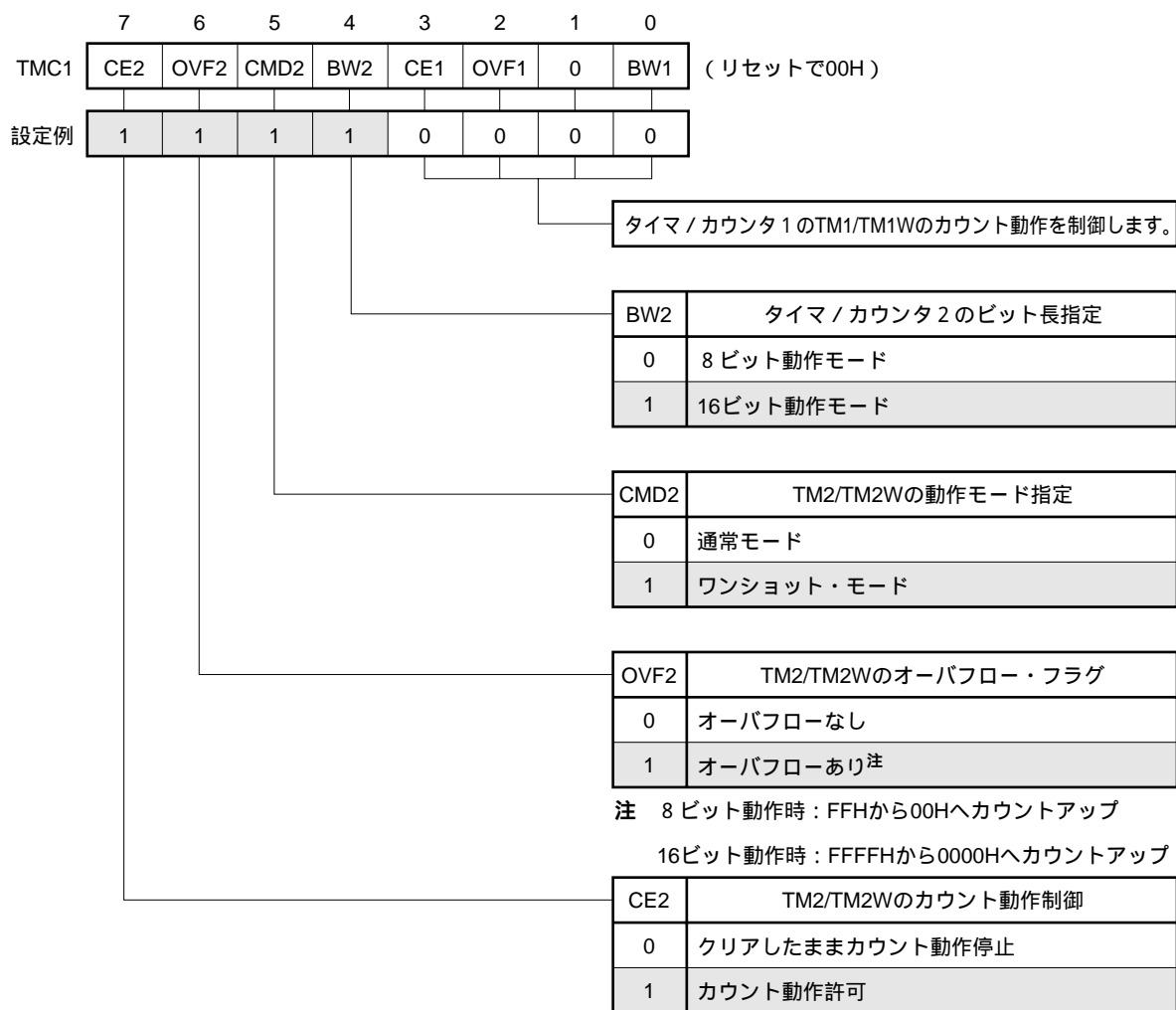


MOD1	MOD0	CLR22	CLR21	CM21	CR21Wの動作 指定	タイマ出力モード指定		TM2Wのクリア動作
						TO2	TO3	
0	0	0	0	0	コンペア動作	トグル出力	トグル出力	クリア動作しない
0	0	0	1	0		トグル出力	トグル出力	TM2WとCR21Wが一致したらクリア
0	0	1	0	0		トグル出力	トグル出力	INTP1により、TM2Wの内容をCR22Wへキャプチャしたあとにクリア
0	0	1	1	0		トグル出力	トグル出力	TM2WとCR21Wの一致またはINTP1により、TM2Wの内容をCR22Wレジスタへキャプチャしたあとにクリア
0	1	0	0	0		PWM出力	トグル出力	クリアしない
1	0	0	0	0		PWM出力	PWM出力	クリアしない
1	1	0	1	0		PPG出力	トグル出力	TM2WとCR21Wが一致したらクリア
0	0	0	0	1	キャプチャ動作	トグル出力	/	クリアしない
0	0	0	1	1		トグル出力		INTP2により、TM2Wの内容をCR21Wへキャプチャしたあとにクリア
0	1	0	0	1		PWM出力		クリアしない
上記以外					設定禁止			

備考 16ビット動作モード時のレジスタ名称で記載しています。

8ビット動作モード時は、TM2W, CR20W, CR21W, CR22Wは、TM2, CR20, CR21, CR22になります。

タイマ・コントロール・レジスタ1 (TMC1)



(4) 入力方法

なし

(5) 出力方法

なし

(6) プログラム説明**イニシャライズ処理 [レーベル名称 : OSPINI]**

- (a) P60端子出力をロウ・レベルに設定します。
- (b) タイマ・コントロール・レジスタ 1 (TMC1) で16ビット動作モードのワンショット・モードに設定します。
- (c) タイマ・レジスタ 2 (TM2W) のカウント・クロックをf_{xx}/32に指定します。
- (d) コンペア・レジスタ (CR21) に最初に出力するパルス幅の値を設定します。
- (e) タイマ・レジスタ 2 (TM2W) のカウント動作をスタートさせます。
- (f) INTC21割り込み要求のマスクを解除します。
- (g) 割り込みを許可します。

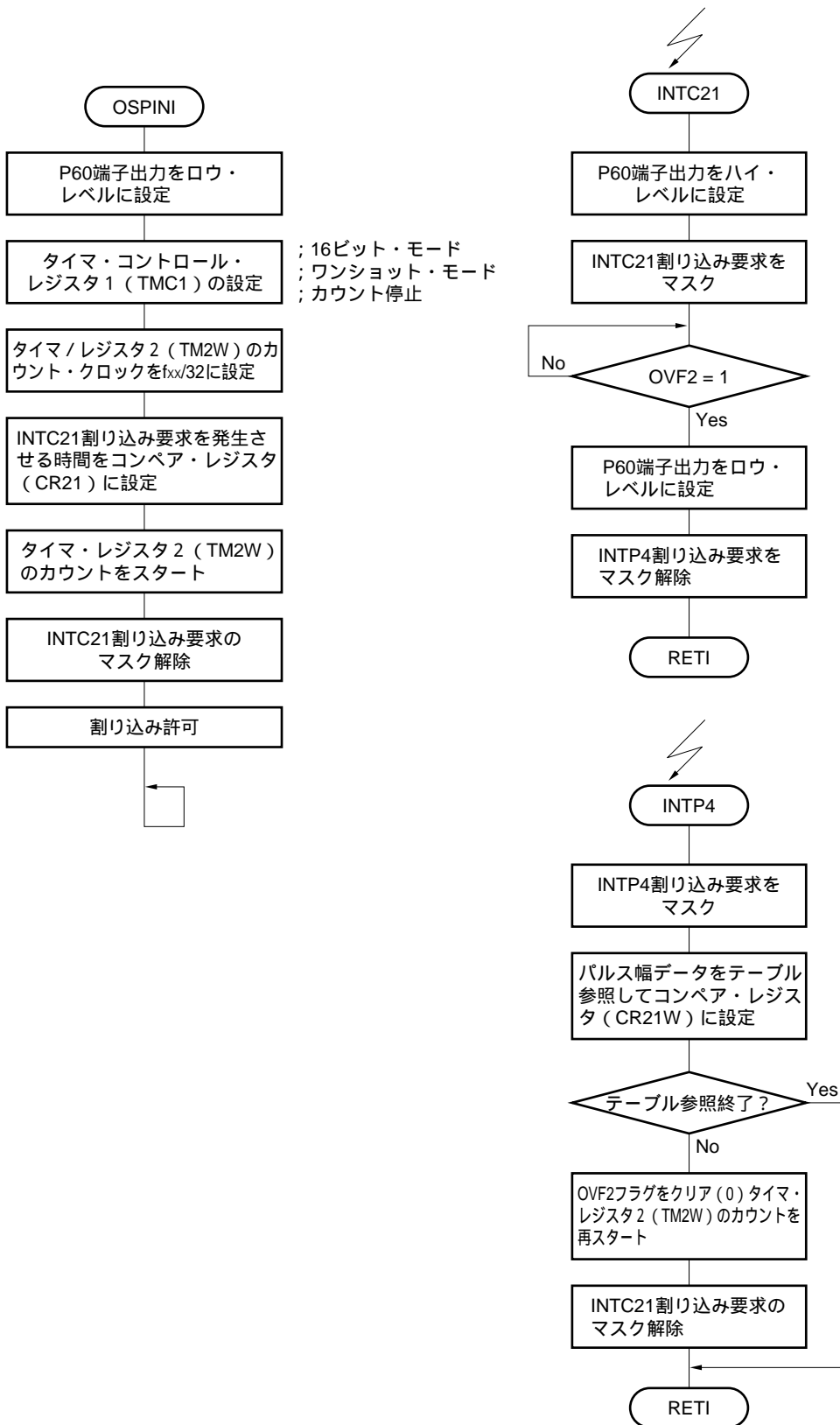
INTC21割り込み処理 [レーベル名称 : INTC21]

- (a) P60端子出力をハイ・レベルに設定します。
- (b) INTC21割り込み要求をマスクします。
- (c) タイマ・レジスタ 2 (TM2W) のカウント値がフルカウントし、タイマ・コントロール・レジスタのOVF2フラグがセットされると、P60端子をロウ・レベルに設定します。
- (d) INTP4割り込み要求のマスクを解除します。

INTP4割り込み処理 [レーベル名称 : INTP4]

- (a) INTP4割り込み要求をマスクします。
- (b) パルス幅データをテーブル参照して、コンペア・レジスタ (CR21W) に設定します。
- (c) すべての設定値テーブルのテーブル参照が終了したか否かを確認します。
すべてのテーブル値の参照終了 : 割り込み処理を終了します。
すべてのテーブル値の参照未終了 : (d) の処理へ進みます。
- (d) タイマ・コントロール・レジスタ 1 (TMC1) のOVF2フラグをクリア (0) して、カウント動作を再開させます。
- (e) INTC21割り込み要求のマスクを解除します。

(7) フロー・チャート



(8) プログラム・リスト

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
INTC21VT CSEG  AT 0018H          ; INTC21 vector table
          DW    INTC21
INTP5VT  CSEG  AT 001EH          ; INTP5 vector table
          DW    INTP5
      .
      .
      CSEG
OSPOUT:
      LOCATION 0H
      .
      .
      MOV     B,#0                ;
      MOVG    WHL,#OSPWIDT        ;
OSPINI:
      CLR1    P6.0                ; P60 Low
      MOV     TMC1,#01110000B     ; set 16bitmode,one_shot mode,
                                   timer stop
      MOV     PRM1,#00110001B     ; set prescaler fxx/32
      MOVW    CR21W,#0C350H       ; set one-shot pulse width 16ms
      MOV     CRC2,#00010000B     ; set T02 timer out,
                                   disable clear TM2W
      SET1    CE2                  ; count start
      CLR1    CMK21                ; open INTC21 mask
      EI                      ; enable interrupt
      .
      .
;*****
;
;*      INTC21 interrupt routine      *
;*      --- request OSP out ---      *
;*****
;
INTC21:
      SET1    P6.0                ; P60 Hi
      SET1    CMK21                ; INTC21 mask
LOOP:
      BF     OV2,$LOOP            ;
      CLR1    P6.0                ; P60 Low
      CLR1    PMK4                ; open INTP4 mask
      RETI

```

```

;
; *****
; *      INTP4 interrupt routine      *
; *      --- Timer clear and start --- *
; *****
;
INTP4:
    SET1    PMK4                ; INTP4 mask
    INC     B                    ;
    INC     B                    ;
    CMP     B,#0AH              ;
    BZ      $ENDRETI            ;
    MOVW    AX,[WHL+B]          ;
    MOVW    CR21W,AX            ; set one-shot pulse width next
                                ; data
    CLR1    CMK21                ; open INTC21 mask
    CLR1    OVF2                ; again start TM2W
ENDRETI:
    RETI

; *****
; *      CR21 set data table          *
; *****
;
OSPWIDT:
    DW      030D3H              ; 16ms
    DW      032E0H              ; 17ms
    DW      036EEH              ; 18ms
    DW      039FBH              ; 19ms
    DW      03D08H              ; 20ms
    END

```


2.9 パルス周期測定

パルス周期測定は、外部割り込み要求 (INTP2) 入力端子に入力される外部入力パルスのパルス幅を測定するものです。INTP2端子入力される外部入力パルスは、有効エッジを検出するためにハイ・レベル、ロウ・レベルとも4システム・クロック ($0.32 \mu\text{s} : f_{\text{CLK}} = 12.5 \text{ MHz}$) 以上必要でこれ以下の場合には有効エッジが検出されず、キャプチャ動作を行わない場合があります。

(1) 処理概要

(a) 仕様

キャプチャ・レジスタ (CR21W) を応用したパルス周期測定のプログラム例を紹介します。

- ・ f_{CLK} ^注 : 12.5 MHz
- ・ カウント・クロック : $8/f_{\text{xx}}$ ^注

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

タイマ/カウンタ2は、16ビット動作モードで使用します。

図2-15に示すようにINTP2端子入力の有効エッジ (ここでは立ち上がりエッジ) に同期して、カウント中のタイマ・レジスタ2 (TM2W) のカウント値をキャプチャ・レジスタ (CR21W) に取り込み、保持します。

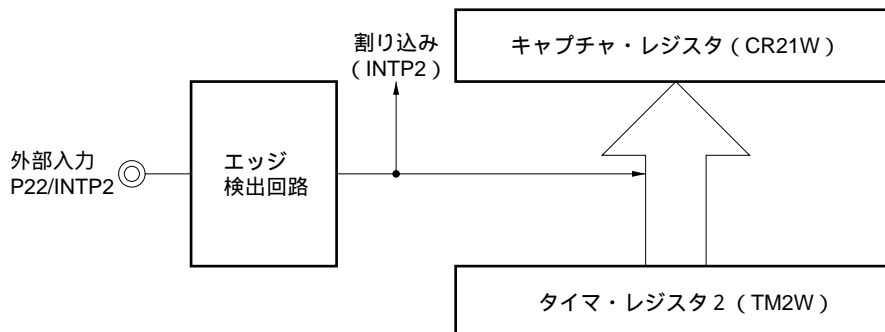
パルス周期は、 n 回目の有効エッジ検出によりキャプチャ・レジスタ (CR21W) に取り込み、保持されたキャプチャ値 (D_n) と、前回 ($n-1$) 回目の有効エッジ検出によりキャプチャ・レジスタ (CR21W) に取り込み、保持されたキャプチャ値 (D_{n-1}) との差と、カウント・クロック ($8/f_{\text{xx}}$) から求められます。

パルス周期を求める式を次に示します。

$$\text{パルス周期} = (D_{n+1} - D_n) \times 8/f_{\text{xx}}$$

図2 - 15に、パルス周期測定のパルス周期測定のブロック図を示します。

図2 - 15 パルス周期測定のパルス周期測定のブロック図

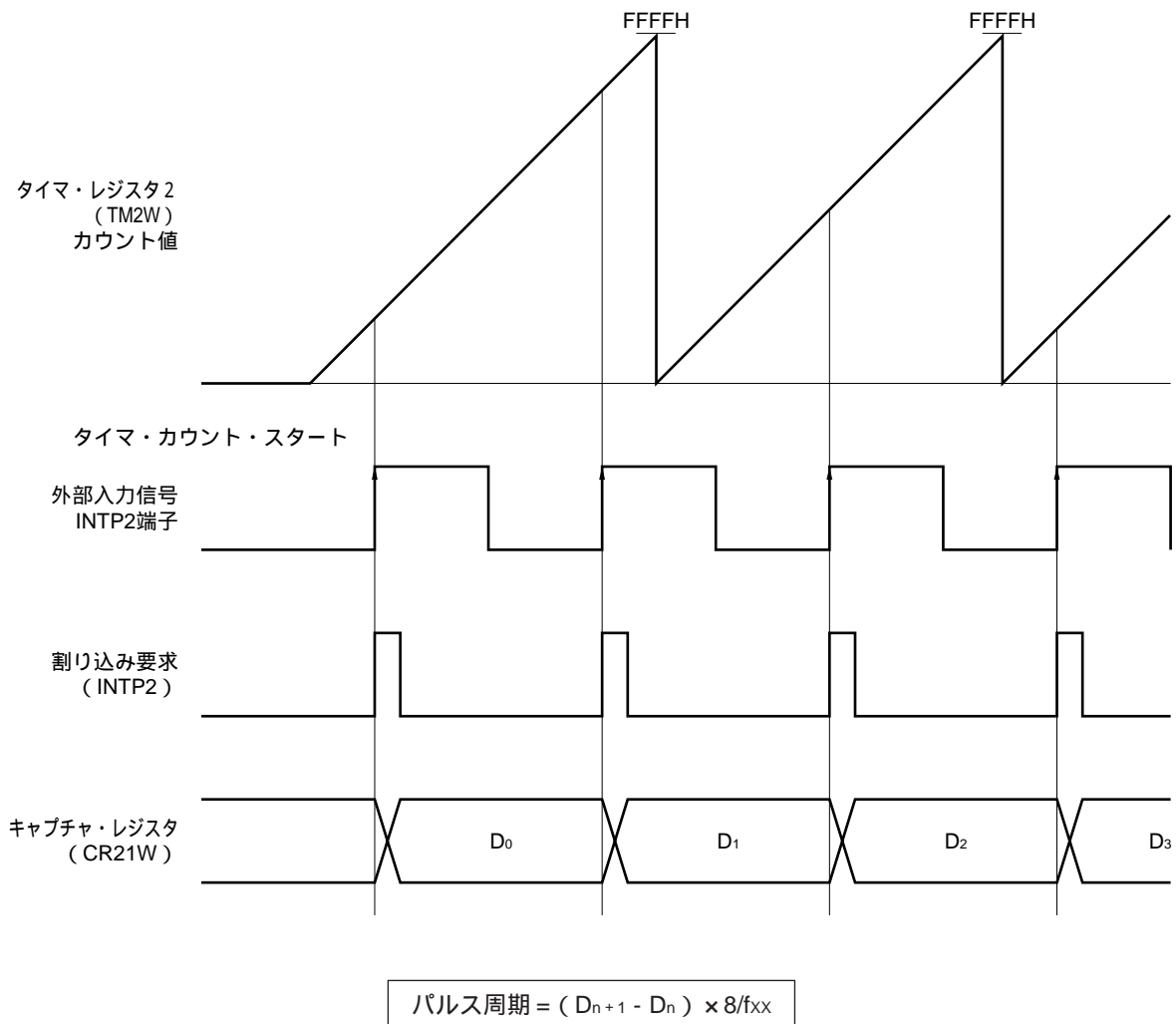


注意 P22/INTP2端子のエッジ検出には、クロック・サンプリングによるデジタル・ノイズ除去回路を内蔵しています。サンプリング・クロックは f_{CLK} 注固定となります。

デジタル・ノイズ除去は、 f_{CLK} のクロックでサンプリングを行い、入力されているレベルが連続して4回以上同一でなければノイズとして除去されます。したがって、有効エッジとして認められるには、 f_{CLK} のクロック4周期以上の期間 ($0.32 \mu\text{s}$: $f_{CLK} = 12.5 \text{ MHz}$, $f_{CLK} = 1/2f_{XX}$) レベルを保持しなければなりません。

注 f_{CLK} : 内部システム・クロック周波数

図2 - 16 INTP2入力のパルス幅測定のタイミング・チャート



ここでは、INTP2端子への立ち上がりエッジを検出して、INTP2入力のパルス周期を測定します。ただし、タイマ・レジスタ2 (TM2W) のカウント値のオーバーフローを考慮していませんので、測定範囲は0.32 μsから21 msとなります (8/f_{xx} : f_{xx} = 25 MHz動作時)。

(2) 使用RAM領域

本プログラムでは、表2 - 4に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表2 - 4 パルス周期測定プログラムで用いるワーク・エリア

RAM名称	用 途	バイト数	初期値
CAPWK	前回のタイマ・レジスタ2 (TM2W) のカウント値をキャプチャした値を保存するワーク領域	2	-
WIDTH1	パルス周期算出結果を格納するワーク領域	2	-

(3) 使用レジスタ

(a) 汎用レジスタ

イニシャライズ処理 [レーベル名称 : PULSE]

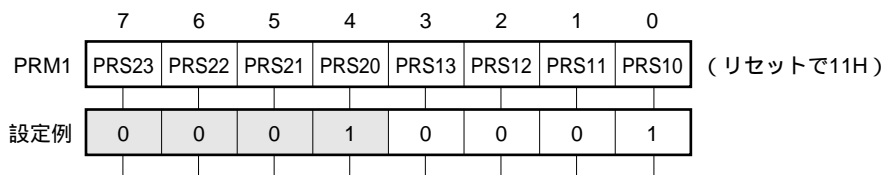
なし

パルス周期算出処理 [レーベル名称 : PLSANA]

AX, BCレジスタ

(b) 特殊機能レジスタ

プリスケアラ・モード・レジスタ1 (PRM1)



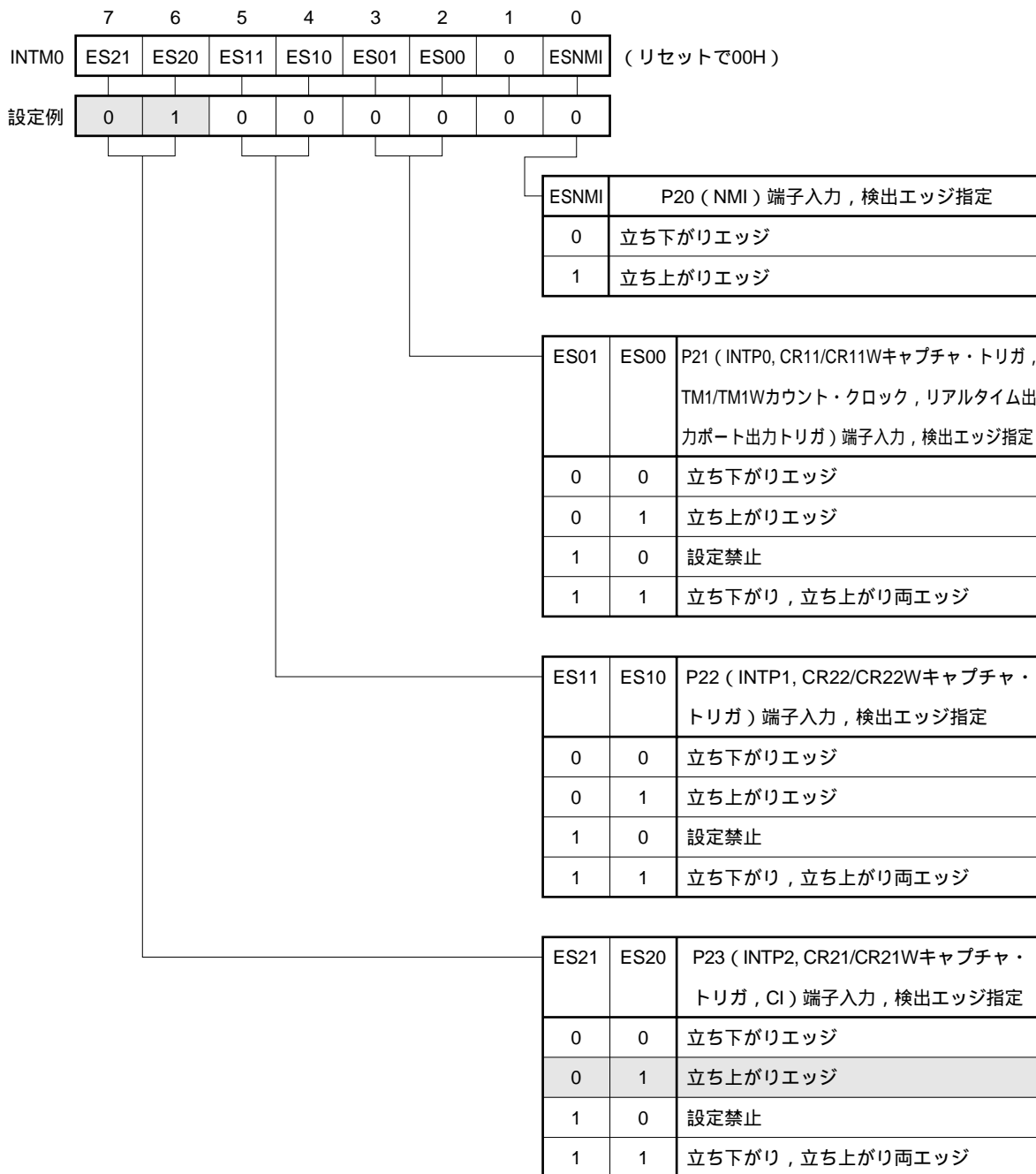
(f_{xx} = 25 MHz)

PRS13	PRS12	PRS11	PRS10	タイマ/カウンタ1のTM1/TM1Wカウント・クロック指定	
				カウント・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (INTP0)	-
上記以外				設定禁止	

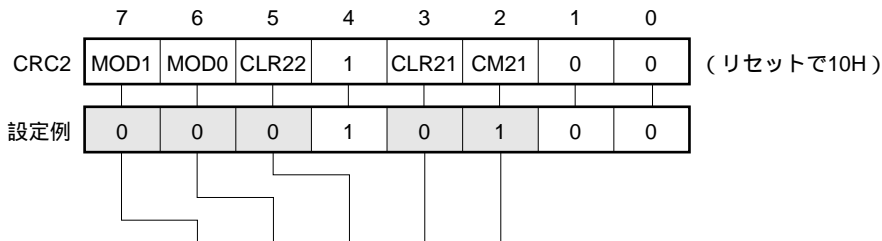
(f_{xx} = 25 MHz)

PRS23	PRS22	PRS21	PRS20	タイマ/カウンタ2のTM2/TM2Wのカウント・クロック指定	
				カウント・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (CI/INTP2)	-
上記以外				設定禁止	

外部割り込みモード・レジスタ0 (INTM0)



キャプチャ/コンペア・コントロール・レジスタ2 (CRC2)

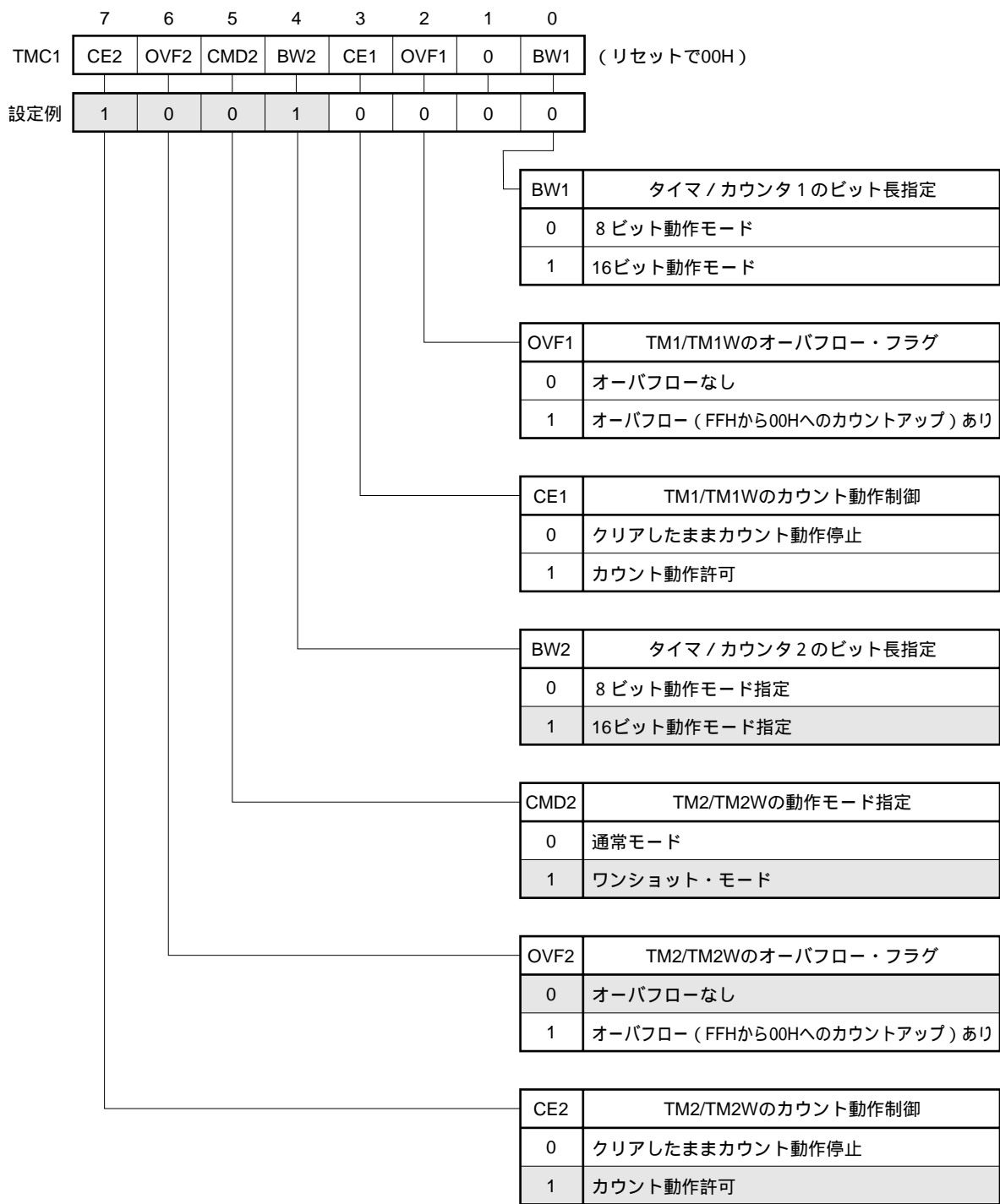


MOD1	MOD0	CLR22	CLR21	CM21	CR21Wの動作 指定	タイマ出力モード指定		TM2Wのクリア動作
						TO2	TO3	
0	0	0	0	0	コンペア動作	トグル出力	トグル出力	クリア動作しない
0	0	0	1	0		トグル出力	トグル出力	TM2WとCR21Wが一致したらクリア
0	0	1	0	0		トグル出力	トグル出力	INTP1により, TM2Wの内容をCR22Wへキャプチャしたあとにクリア
0	0	1	1	0		トグル出力	トグル出力	TM2WとCR21Wの一致またはINTP1により, TM2Wの内容をCR22Wレジスタへキャプチャしたあとにクリア
0	1	0	0	0		PWM出力	トグル出力	クリアしない
1	0	0	0	0		PWM出力	PWM出力	クリアしない
1	1	0	1	0		PPG出力	トグル出力	TM2WとCR21Wが一致したらクリア
0	0	0	0	1	キャプチャ動作	トグル出力	/	クリアしない
0	0	0	1	1		トグル出力		INTP2により, TM2Wの内容をCR21Wへキャプチャしたあとにクリア
0	1	0	0	1		PWM出力		クリアしない
上記以外					設定禁止			

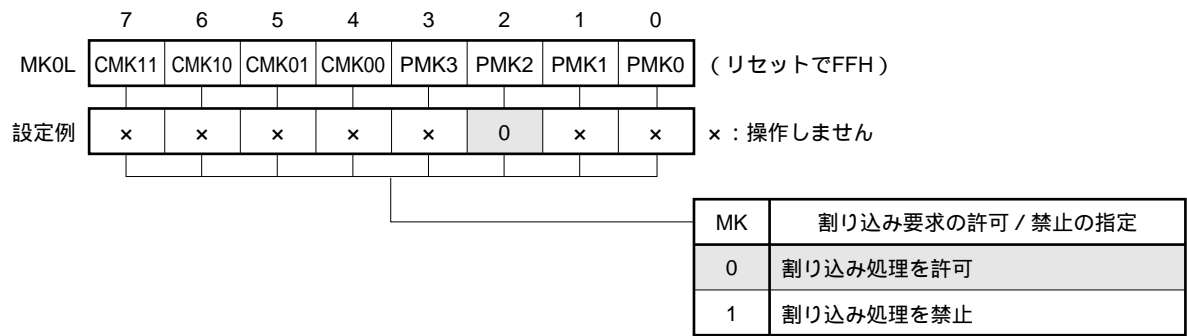
備考 16ビット動作モード時のレジスタ名称で記載しています。

8ビット動作モード時は, TM2W, CR20W, CR21W, CR22Wは, TM2, CR20, CR21, CR22になります。

タイマ・コントロール・レジスタ1 (TMC1)



割り込みマスク・レジスタL (MKOL)



(4) 入力方法

なし

(5) 出力方法

RAM領域に定義したワーク領域に、次の内容が格納されます。

WIDTH1 : パルス周期の算出結果が格納されます。

CAPWK : 前回にカウントしたタイマ・レジスタ2 (TM2W) のキャプチャ値が格納されます。

(6) プログラム説明

イニシャライズ処理 [レーベル名称 : PULSE]

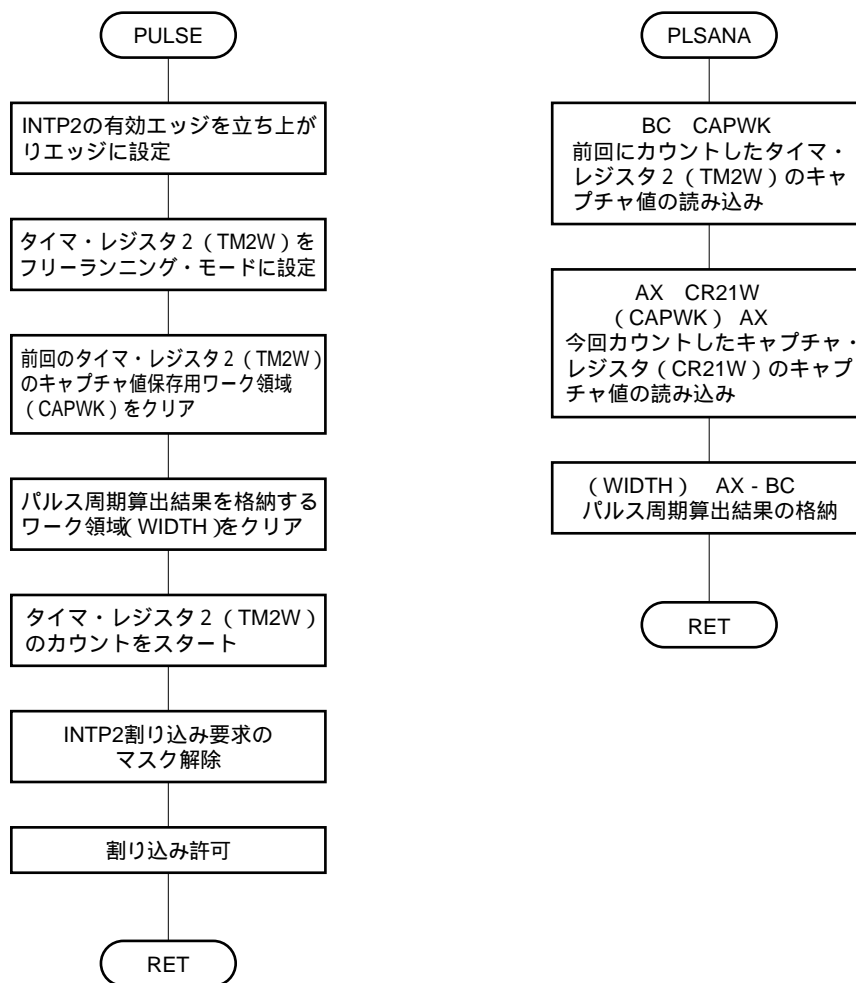
- (a) INTP2端子入力の有効エッジを立ち上がりエッジとします。
- (b) タイマ・レジスタ2 (TM2W) とコンペア・レジスタ (CR21W) の一致による、タイマ・レジスタ2 (TM2W) のクリアを禁止します。
- (c) 前回カウントしたタイマ・レジスタ2 (TM2W) のキャプチャ値保存用のワーク領域 (CAPWK) および、パルス周期算出結果を格納するワーク領域 (WIDTH1) をクリアします。
- (d) タイマ・レジスタ2 (TM2W) のカウント動作を許可します。
- (e) INTP2割り込み要求のマスクを解除します。

パルス周期算出処理 [レーベル名称 : PLSANA]

INTP2割り込み要求によるベクタ割り込み処理です。

- (a) BCレジスタに前回カウントしたタイマ・レジスタ2 (TM2W) のキャプチャ値をワーク領域 (CAPWK) から読み込みます。
- (b) AXレジスタに今回カウントしたタイマ・レジスタ2 (TM2W) のキャプチャ値をキャプチャ・レジスタ (CR21W) から読み込み、AXレジスタの値をワーク領域 (CAPWK) に格納します。
- (c) AXレジスタの値とBCレジスタの値の差を取り、パルス周期算出結果としてパルス周期算出結果を格納するワーク領域 (WIDTH1) に格納します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

WIDTH1 : パルス周期算出結果を格納するワーク領域

CAPWK : 前回のタイマ・レジスタ2 (TM2W) のカウント値をキャプチャした値を保存する
ワーク領域

本アプリケーション・ルーチンのプログラム・リスト

```

NAME      PULSEM
;
; *****
; * 16bit-Timer / Counter Unit *
; *   measure pulse cycle      *
; *****
;
;
PUBLIC    PULSE,PLSANA
EXTRN    CAPWK,WIDTH1      ; work area

;
CSEG
PULSE:
MOV      INTM0,#01000000B; INTP2's enable edge is rise
MOV      CRC2,#00010100B ; clear disable TM2W by CR21W
MOV      PRM1,#00010001B ; fxx/8
MOVW     CAPWK,#0        ; clear result
MOVW     WIDTH1,#0

MOV      TMC1,#10010000B ; timer start
CLR1     PMK2            ; open INTP2 mask
EI       ; interrupt enable

RET

;
PLSANA:
MOVW     BC,CAPWK        ; read last capture data
MOVW     AX,CR21W        ; capture read
MOVW     CAPWK,AX        ; save capture data
SUBW     AX,BC           ;
MOVW     WIDTH1,AX      ;

RET

;
END

```

2.10 外部イベント・カウンタ

外部イベント・カウンタは、外部割り込み端子から入力されるクロック・パルスをカウントするモードです。

(1) 処理概要

(a) 仕様

タイマ/カウンタ 1 は、16ビット動作モードで使用します。

タイマ/カウンタ 1 の外部イベント・カウント機能を使用し、INTP0端子から入力されるクロックの立ち上がりエッジを 1 秒間カウントするプログラム例を紹介します。

- ・ f_{xx} ^注 : 25 MHz
- ・ カウント・クロック : f_{xx} ^注 / 512

注 f_{xx} : クリスタル/セラミック発振周波数

(b) 使用周辺機能説明

外部イベント・カウンタ・モードでは、プリスケアラ・モード・レジスタ 1 (PRM1) の下位 4 ビットの設定によってタイマ・レジスタ 1 (TM1W) のカウント・クロックを外部クロック入力に指定することで、タイマ・レジスタ 1 (TM1W) が外部イベント・カウンタとして動作します。

外部イベント・カウンタとしてカウントできる外部クロック・パルスの最大周波数は、両エッジをカウントする場合も、片エッジのみカウントする場合も、1.56 MHz です ($f_{CLK} = 12.5$ MHz 動作時)。

また、パルス幅もハイ・レベル、ロウ・レベルともに 4 システム・クロック ($0.32 \mu s$: $f_{CLK} = 12.5$ MHz 動作時) 以上必要で、これ以下の場合は、カウントされない場合があります。

またINTP0端子のエッジ検出は、サンプリング・クロック選択レジスタ (SCS0) で指定されるクロック (f_{SMP}) でデジタル・ノイズ除去を行います。デジタル・ノイズ除去は、 f_{SMP} のクロックでサンプリングを行い、入力されているレベルが連続して 4 回以上同一でなければノイズとして除去されます。したがって有効エッジと認められるためには、 f_{SMP} のクロックで最低 4 周期以上の期間、レベルを保持しなければなりません。

図 2 - 17 にサンプリング・クロック選択レジスタ (SCS0) の設定を示します。

図2 - 17 サンプルング・クロック選択レジスタ (SCS0) の設定



図2 - 18に、外部イベント・カウンタの機能を示します。

図2 - 18 INTP0端子から入力されるクロックをカウントする外部イベント・カウンタ

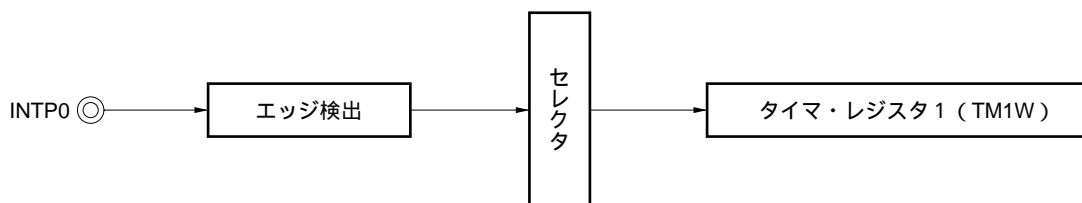
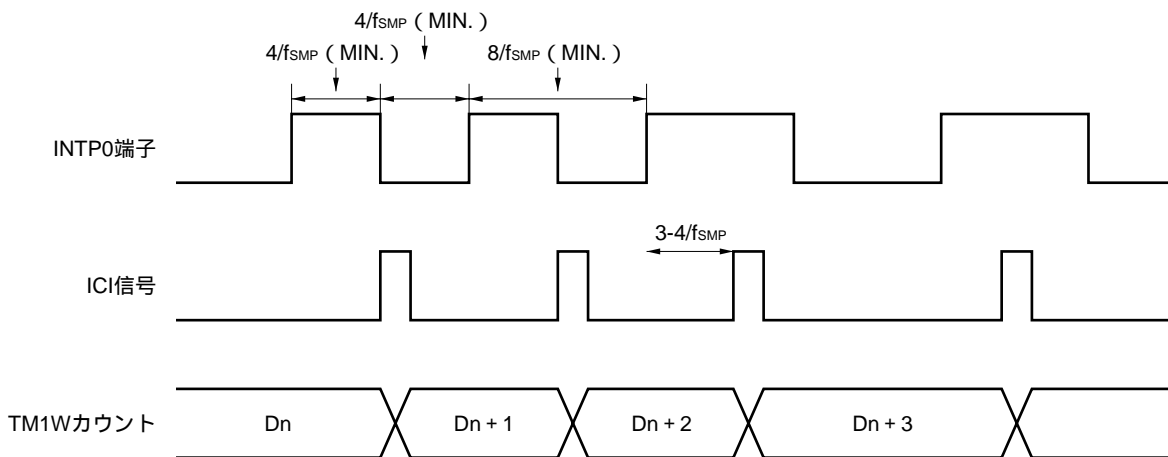


図2 - 19に、タイマ/カウンタ1の外部イベント・カウント・タイミングを示します。

図2 - 19 タイマ/カウンタ1の外部イベント・カウント・タイミング



備考1 . ICI : INTPO入力のエッジ検出回路通過後の信号

2 . f_{SMP} : サンプリング・クロック選択レジスタ (SCS0) で選択

(2) 使用RAM領域

本プログラムでは、表2 - 5に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表2 - 5 外部イベント・カウンタ機能のプログラムで用いるRAM

RAM名称	用途	バイト数	初期値
MEMORY	カウントしたタイマ・レジスタ1 (TM1W) のカウント値を格納するワーク領域	2	-

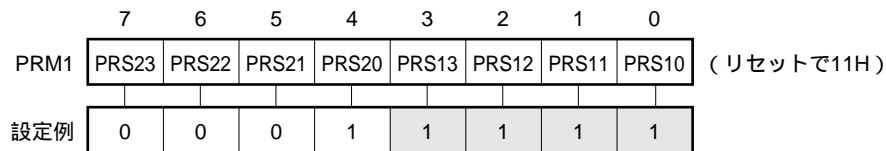
(3) 使用レジスタ

(a) 汎用レジスタ

なし

(b) 特殊機能レジスタ

プリスケアラ・モード・レジスタ1 (PRM1)



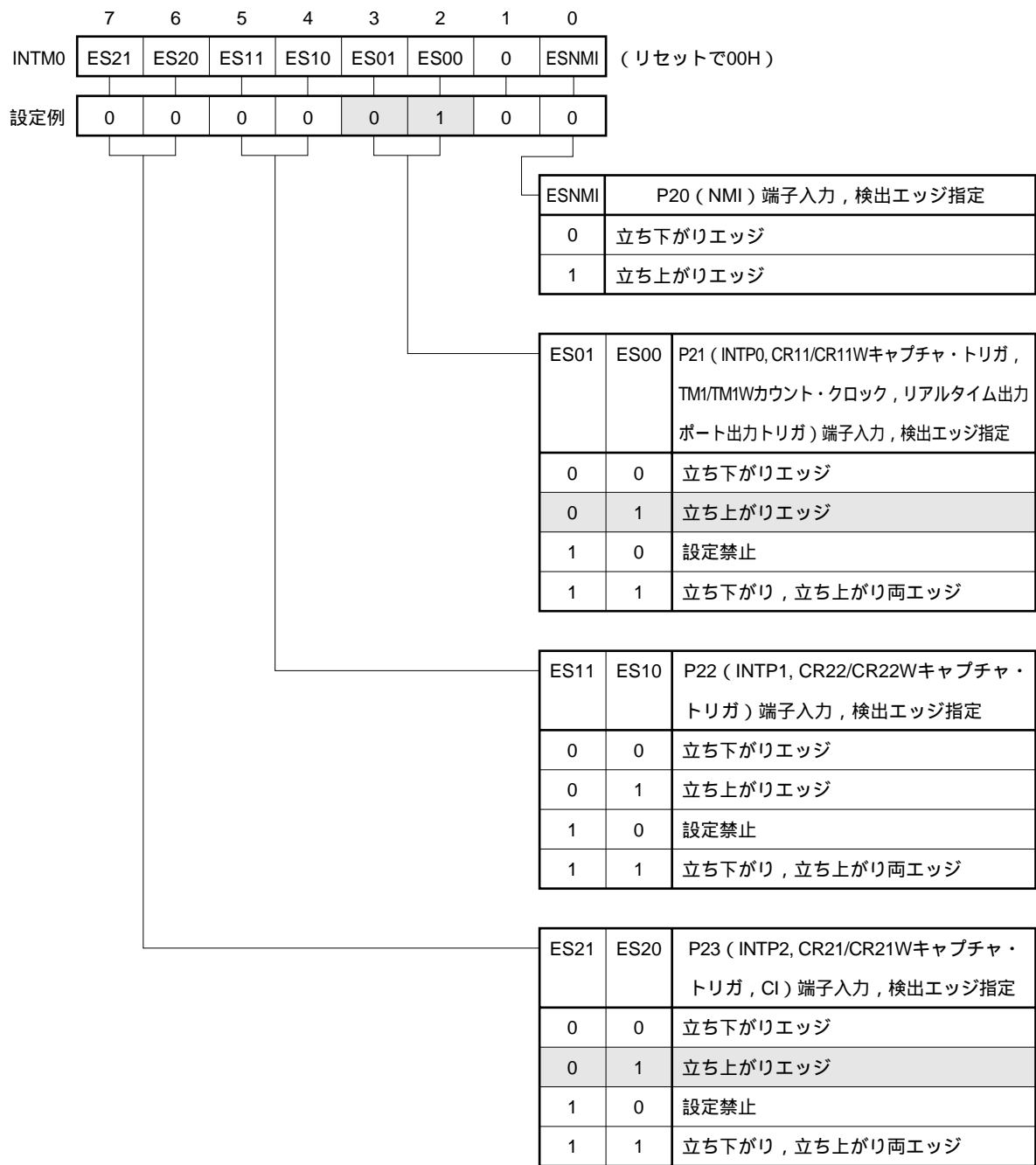
(f_{xx} = 25 MHz)

PRS13	PRS12	PRS11	PRS10	タイマ/カウンタ1のTM1/TM1Wのカウンタ・クロック指定	
				カウンタ・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (INTP0)	-
上記以外				設定禁止	

(f_{xx} = 25 MHz)

PRS23	PRS22	PRS21	PRS20	タイマ/カウンタ2のTM2/TM2Wのカウンタ・クロック指定	
				カウンタ・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (CI/INTP2)	-
上記以外				設定禁止	

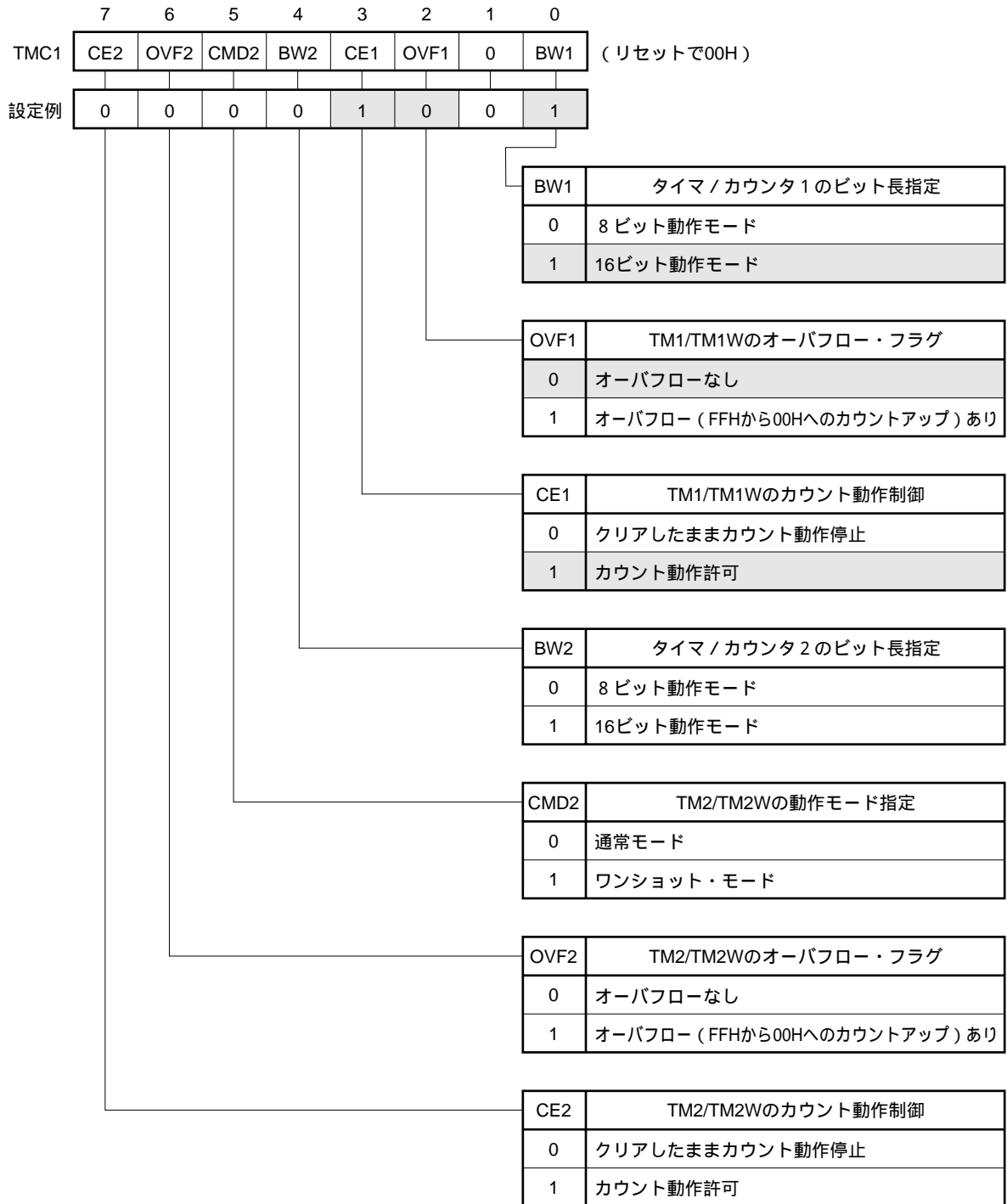
外部割り込みモード・レジスタ0 (INTM0)



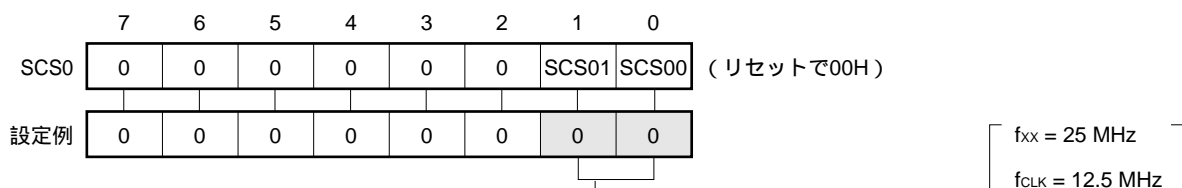
キャプチャ/コンペア・コントロール・レジスタ1 (CRC1)



タイマ・コントロール・レジスタ1 (TMC1)



サンプリング・クロック選択レジスタ (SCS0)



SCS01	SCS00	サンプリング・クロック (f _{SMP})	ノイズとして除去されるパルス幅	信号として認められる最小パルス幅
0	0	f _{CLK}	2/f _{CLK} (160 ns)	4/f _{CLK} (320 ns)
0	1	f _{xx} /64	128/f _{xx} (5.12 μs)	256/f _{xx} (10.24 μs)
1	0	f _{xx} /128	256/f _{xx} (10.24 μs)	512/f _{xx} (20.48 μs)
1	1	f _{xx} /256	512/f _{xx} (20.48 μs)	1024/f _{xx} (40.96 μs)

f_{xx} : クリスタル/セラミック発振周波数

f_{CLK} : 内部システム・クロック

(4) 入力方法

なし

(5) 出力方法

なし

(6) プログラム説明

外部イベント・カウント処理 [レーベル名称 : EVNTCNT]

- (a) タイマ・レジスタ1 (TM1W) とコンペア・レジスタ (CR10W) の一致によるタイマ・レジスタ1 (TM1W) のクリアを禁止し、キャプチャ/コンペア・レジスタ (CR11W) をキャプチャ動作に指定します。
- (b) プリスケアラ・モード・レジスタ1 (PRM1) に外部クロック (INTP0) を設定します。
- (c) サンプリング・クロック選択レジスタ (SCS0) でINTP0端子のエッジ検出サンプリング・クロックをf_{CLK}に設定します。
- (d) 外部割り込みモード・レジスタ0 (INTM0) でINTP0の検出エッジを、立ち上がりエッジに設定します。
- (e) タイマ・コントロール・レジスタ1 (TMC1) で16ビット動作モードでカウント動作を許可します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

MEMORY : 外部イベント機能でカウントしたタイマ・レジスタ1 (TM1W) のカウント値を格納

本アプリケーション・ルーチンのプログラム・リスト

```

PUBLIC  EVNTMAIN
      .
      .
WORKSEG DSEG saddr
MEMORY: DS      2
      .
      .
CSEG
EVNTMAIN:
      LOCATION OH
      SEL      RBO
      .
      .
      MOV      CRC0,#00011000B
      MOVW     CR01,#48781          ; 1s
      MOV      PRM0,#00010111B    ; fxx/512
      ;
      ;*****      Event Counter      *****
      ;
EVNTCNT:
      MOV      CRC1,#00000100B    ; disable clear TM1W
      MOV      PRM1,#00011111B    ; set TM1W prescaler INTPO
      MOV      SCS0,#00000000B    ; INTPO sampling clock fclk set
      MOV      INTM0,#00000100B   ; edge set
      MOV      TMC1,#00001001B    ; TM1W count start

      CLR1     CIF01               ;
      CLR1     CMK01               ; TM1W MASK EI
      ;
      MOV      TMC0,#00001000B    ; Timer0 start
      EI

START:
      NOP
      BR      START
      ;
      ;*****      INTC01      *****
      ;
INTC01:
      SEL      RB1                 ;
      MOV      TMC0,#00000000B    ; Timer0 Stop
      MOVW     AX,TM1W             ;
      MOVW     MEMORY,AX          ;
      DI
      RETI

      END

```

第3章 シリアル・インタフェースのプログラム例

3.1 シリアル・インタフェース機能概要

μPD784038, 784038Yサブシリーズのシリアル・インタフェースについて説明します。

表3 - 1に、μPD784038, 784038Yサブシリーズのシリアル・インタフェース機能一覧を示します。

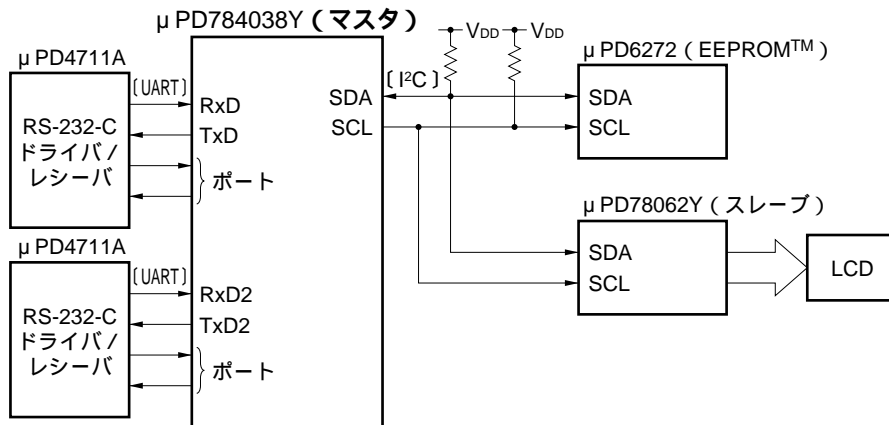
表3 - 1 μPD784038, 784038Yサブシリーズのシリアル・インタフェース機能一覧

機能 \ 種類	アシンクロナス・シリアル・インタフェース	クロック同期式シリアル・インタフェース
モード	アシンクロナス・シリアル・インタフェース・モード	3線式シリアル/I/Oモード
	3線式シリアル/I/Oモード	2線式シリアル/I/Oモード I ² Cバス・モード (μPD784038Yサブシリーズのみ)
チャンネル数	2チャンネル (ポーレート・ジェネレータ内蔵)	1チャンネル

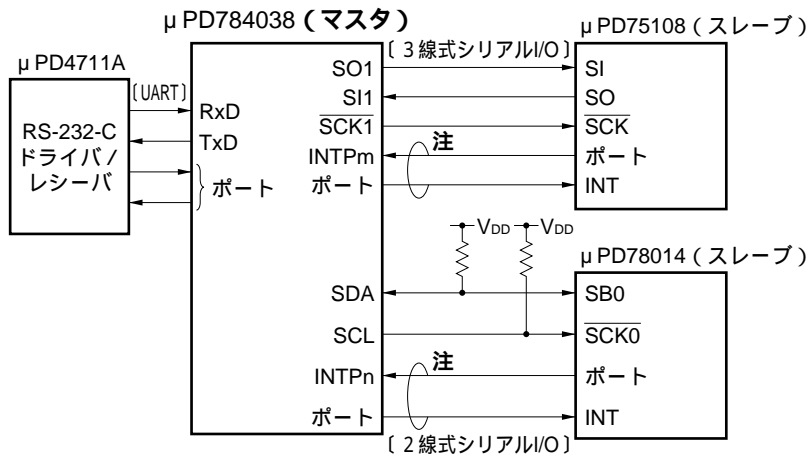
図3 - 1に、シリアル・インタフェース例を示します。

図3 - 1 シリアル・インタフェース例

(1) UART + I²C



(2) UART + 3線式シリアルI/O + 2線式シリアルI/O



注 ハンドシェイク・ライン

本章ではシリアル・インタフェース機能の説明として、次に示す3つのプログラム例を紹介します。

・ 3.3 アシクロナス・シリアル・インタフェースのプログラム例

アシクロナス・シリアル・インタフェース・モードを使用

・ 3.4 クロック同期式シリアル・インタフェースのプログラム例

3線式シリアルI/Oモードを使用

I²Cバス・モードを使用 (μPD784038Yサブシリーズのみ)

μPD784038, 784038Yサブシリーズは、アシクロナス・シリアル・インタフェース (UART) モードと3線式シリアルI/O (IOE) モードを選択できるシリアル・インタフェースを2チャンネル内蔵しています。

2チャンネルのUART/IOEは、まったく同一の機能です。したがって、この章では特に違いがないかぎり、UART/IOE1を代表として説明しています。UART2/IOE2として使用する場合は、UART/IOE1のレジスタ名称、ビット名称、端子名称を表3 - 2を参照に読み替えてください。

表3 - 2 UART/IOE1とUART2/IOE2の名称の違い

項 目	UART/IOE1	UART2/IOE2
端子名称	P25/ASCK/SCK1, P30/RxD/SI1, P31/TxD/SO2	P12/ASCK2/SCK2, P13/RxD2/SI2, P14/TxD2/SO2
アシンクロナス・シリアル・インタフェース・モード・レジスタ	ASIM	ASIM2
アシンクロナス・シリアル・インタフェース・モード・レジスタ 内のビット名称	TXE, RXE, PS1, PS0, CL, SL, ISRM, SCK	TXE2, RXE2, PS21, PS20, CL2, SL2, ISRM2, SCK2
アシンクロナス・シリアル・インタフェース・ステータス・レジスタ	ASIS	ASIS2
アシンクロナス・シリアル・インタフェース・ステータス・レジスタ 内のビット名称	PE, FE, OVE	PE2, FE2, OVE2
クロック同期式シリアル・インタフェース・モード・レジスタ	CSIM1	CSIM2
クロック同期式シリアル・インタフェース・モード・レジスタ 内のビット名称	CTXE1, CRXE1, DIR1, CSCK1	CTXE2, CRXE2, DIR2, CSCK2
ポー・レート・ジェネレータ・コントロール・レジスタ	BRGC	BRGC2
ポー・レート・ジェネレータ・コントロール・レジスタ内の ビット名称	TPS0-TPS3, MDL0-MDL3	TPS20-TPS23, MDL20-MDL23
割り込み要求名称	INTSR/INTCSI1, INTSER, INTST	INTSR2/INTCSI2, INTSER2, INTST2
割り込み制御レジスタおよび本章で使用するビット名称	SRIC, CSIIC1, SERIC, STIC, SRIF, CSIIF1, SERIF, STIF	SRIC2, CSIIC2, SERIC2, STIC2, SRIF2, CSIIF2, SERIF2, STIF2

また、表3 - 1 に示すように、 μ PD784038サブシリーズと μ PD784038Yサブシリーズでは、 μ PD784038YサブシリーズがI²Cパス・モードに対応している以外、2つのサブシリーズの機能は同一です。

3.2 アシクロナス・シリアル・インタフェース・モード

ここでは μ PD784038を代表品種として説明します。

μ PD784038のアシクロナス・シリアル・インタフェース・モードでは、スタート・ビットに続く1バイトのデータを送受信する方式で、全二重動作が可能です。

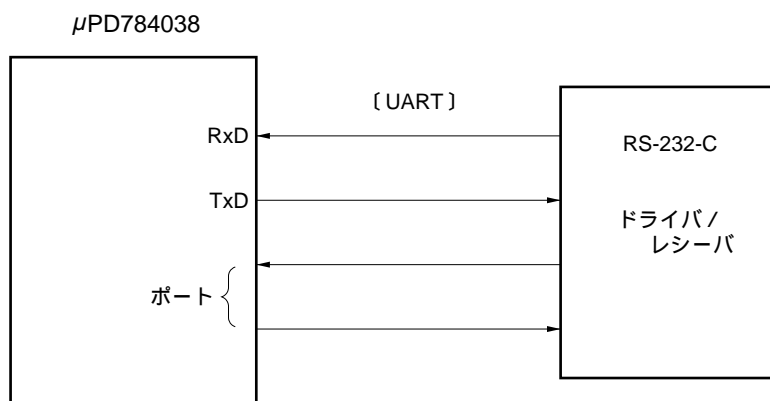
ポー・レート・ジェネレータを内蔵しており、広範囲な任意のポー・レートで通信できます。

(1) システム構成

(a) システム構成例

図3 - 2に、アシクロナス・シリアル・インタフェース・モードのシステム構成例を示します。

図3 - 2 アシクロナス・シリアル・インタフェース・モードのシステム構成例

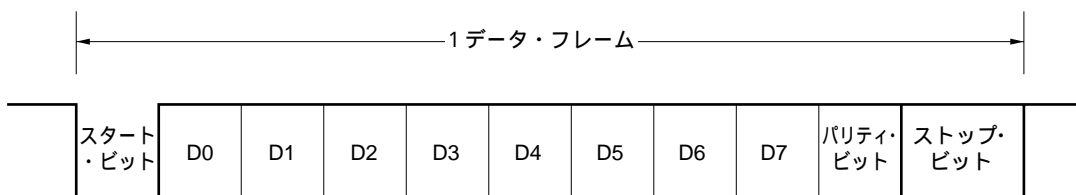


(b) データ・フォーマット

全二重のアシクロナス・モードによるシリアル・データの送受信を行います。

図3 - 3に、送受信データ・フォーマットを示します。

図3 - 3 アシクロナス・シリアル・インタフェースの送受信データのフォーマット



- ・スタート・ビット : 1 ビット
- ・キャラクタ・ビット : 7 ビット / 8 ビット
- ・パリティ・ビット^注 : 偶数パリティ / 奇数パリティ / 0 パリティ / パリティなし

注 偶数パリティ / 奇数パリティ...パリティ・エラーを検出します。
 0 パリティ / パリティなし ...パリティ・エラーを検出しません。

- ・ストップ・ビット : 1 ビット / 2 ビット
- ・ボー・レート : 1200, 2400, 4800, 9600, 19200 (単位 : bps) の設定可能 (ただし, ボー・レート・ジェネレータ内蔵なので, この範囲外でも使用可能)。

(2) 通信動作

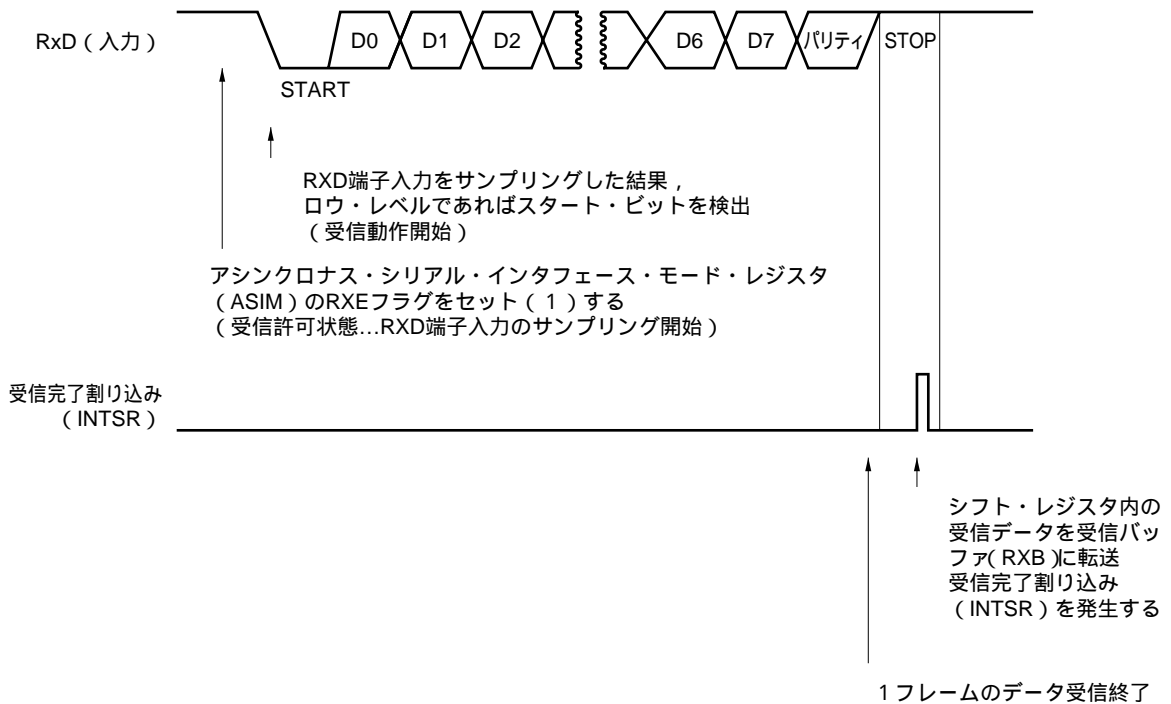
(a) 送受信動作

μ PD784038のアシクロナス・シリアル・インタフェースの送信動作を図3 - 4に、受信動作を図3 - 5に示します。

図3 - 4 送信時の動作タイミング(ストップ・ビット長=1)



図3 - 5 受信時の動作タイミング



・送信動作の中断

送信動作中に送信動作フラグ (TXE) をクリア (0) すると、送信動作を停止します。

・送信動作時の注意事項

1. $\overline{\text{RESET}}$ 入力後、送信シフト・レジスタ (TXS) は空になりますが、送信完了割り込み (INTST) は発生しません。送信シフト・レジスタ (TXS) に送信データを書き込むことによって送信動作を起動できます。
2. 送信動作中にはアシンクロナス・シリアル・インタフェース・モード・レジスタ (ASIM) を書き換えないでください。送信中にASIMを書き換えると、それ以降の送信動作ができなくなる場合があります ($\overline{\text{RESET}}$ 入力で復帰可能です)。
送信中か否かは、送信完了割り込み (INTST) または、INTSTによりセットされる割り込み要求フラグ (STIF) を用いて、ソフトウェアで判断できます。

・受信動作の中断

受信動作中に受信動作フラグ (RXE) をクリア (0) すると、受信動作を停止します。このとき受信バッファ (RXB) およびアシンクロナス・シリアル・インタフェース・ステータス・レジスタ (ASIS) の内容は変化せず、また、受信完了割り込み (INTSR, INTSER) も発生しません。受信エラーを防ぐため、受信バッファは読み出してください。

・受信動作時の注意事項

1. アシンクロナス・シリアル・インタフェース・ステータス・レジスタ (ASIS) の内容は、受信バッファ (RXB) を読み出すか、次のデータを受信することにより、クリア (0) されてしまいます。エラーの内容が知りたい場合には、必ず受信バッファ (RXB) を読み出す前にASISを読み出してください。
2. 受信エラー発生時にも受信バッファ (RXB) は必ず読み出してください。受信バッファ (RXB) を読み出さないと、次のデータ受信時にオーバーラン・エラーが発生し、いつまでも受信エラーの状態が続いてしまいます。

(b) 受信エラー

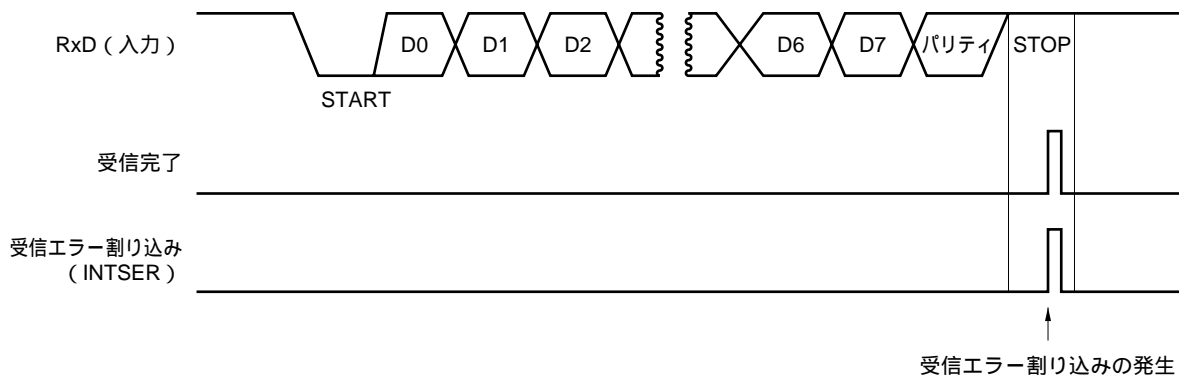
受信動作時のエラーには、表3-3の受信エラーの要因に示すように、3種類のエラー要因があります。アシンクロナス・シリアル・インタフェース・レジスタ (ASIS, ASIS2) の受信エラー・フラグがセット (1) されると同時に、受信エラー割り込み (INTSER) が発生します。

表3-3 受信エラーの要因

受信エラー	要 因
パリティ・エラー	送信時のパリティ指定と受信データのパリティが一致しない エラー発生時：ASIS, ASIS2のビット2 (PE, PE2) がセット (1) される
フレーミング・エラー	ストップ・ビットが検出されない エラー発生時：ASIS, ASIS2のビット1 (FE, FE2) がセット (1) される
オーバラン・エラー	受信バッファからデータを読み出す前に次のデータ受信完了 エラー発生時：ASIS, ASIS2のビット0 (OVE, OVE2) がセット (1) される

図3-6に、受信エラーのタイミングを示します。

図3-6 受信エラーのタイミング



・エラー要因の判断

受信エラー割り込み処理 (INTSER) 内で、アシンクロナス・シリアル・インタフェース・ステータス・レジスタ (ASIS, ASIS2) の内容を読み出すことによって、いずれのエラーが受信時に発生したかを判断します。

・受信エラー信号 (PEフラグ, FEフラグ, OVEフラグ) のクリア

受信エラー要因を示すアシンクロナス・シリアル・インタフェース・モード・レジスタ (ASIS) の内容は、次に示すどちらかの方法でクリアされます。

受信バッファ (RXB) を読み出す。

次のデータを受信する。

3.3 アシクロナス・シリアル・インタフェースのプログラム例

ここでは μ PD784038 を代表製品として説明します。

(1) 処理概要

(a) 仕様

μ PD784038 とターミナル・ソフトを起動させたパソコン（以下パソコンと表記）とを、アシクロナス・シリアル・インタフェース・モードを使用し、通信するプログラム例を紹介します。

ここでは、パソコンのキー・ボードより入力したキャラクタ・データを受信し、このデータをパソコンに送信することにより、パソコンのディスプレイに表示させる例を紹介します。

(b) 使用周辺機能説明

図3-7に、 μ PD784038 とパソコンとの接続を示します。ソフトウェアの動作としては、ループ・バック動作^注となります。データ・ラインとハンドシェイク・ラインはそれぞれTxD, RxD, $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ とし、互いにクロス接続とします。

注 ループ・バック動作：伝送路で信号が変わっていないことを確認するために、受信側から送信側へ送り返す動作のこと。

図3-7 μ PD784038 とパソコンとの接続

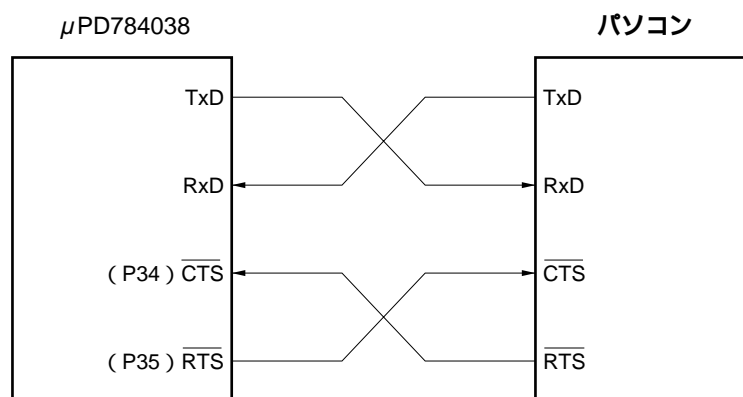


表3 - 4に、通信プロトコルを示します。

表3 - 4 UARTのプログラム例の仕様

項 目	仕 様		
ボー・レート	9600 bps		
ストップ・ビット	2 ビット		
キャラクタ長	8 ビット		
パリティ・ビット	奇数パリティ		
ハンドシェーク・ライン		通信状態 BUSY	通信状態 アクティブ
	CTS端子 (P34)	1	0
	RTS端子 (P35)	1	0

(2) 使用RAM領域

本プログラムでは、表3 - 5に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表3 - 5 アシクロナス・シリアル・インタフェースのプログラム例で使用するRAM / フラグ

RAM / フラグ名称	用 途	バイト数	初期値
RCV_DT	アシクロナス・シリアル・インタフェース・モードを使用し、受信したデータを格納するRAM領域	1	-
TRN_DT	アシクロナス・シリアル・インタフェース・モードを使用し、送信するための送信データを格納するRAM領域	1	-
RCVFLAG	受信完了判別フラグ。 RCVFLAG = 0...受信未完了 RCVFLAG = 1...受信完了	1ビット	0

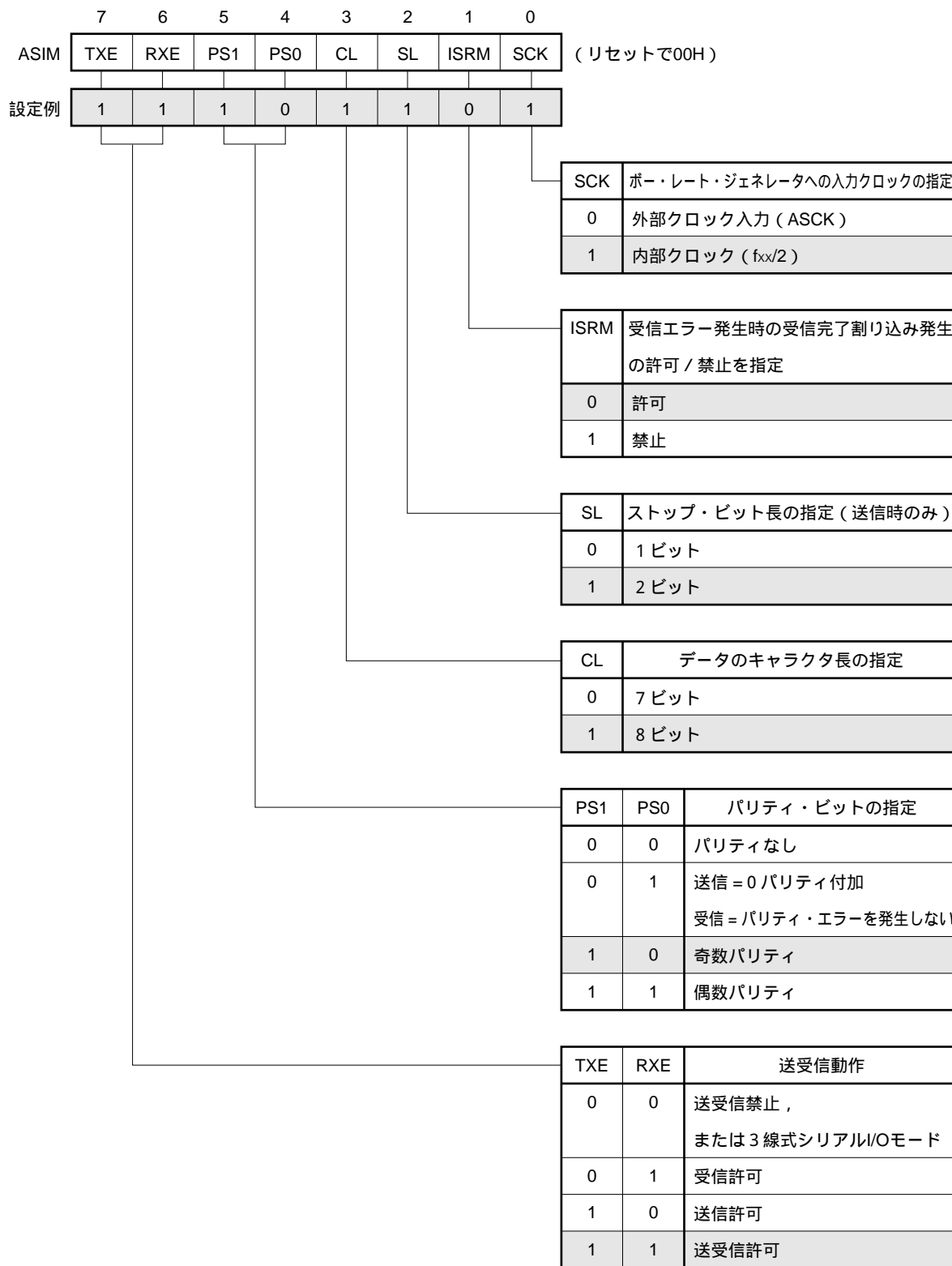
(3) 使用レジスタ

(a) 汎用レジスタ

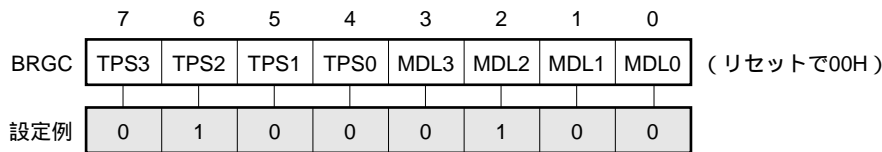
- ・イニシャライズ処理 (レーベル名称: ASY4038)
なし
- ・受信処理 (レーベル名称: RECIV)
Aレジスタ
- ・送信要求処理 (レーベル名称: CLRRTS)
なし
- ・送信処理 (レーベル名称: TRANS)
Aレジスタ

(b) 特殊機能レジスタ

アシンクロナス・シリアル・インタフェース・モード・レジスタ (ASIM)



ポー・レート・ジェネレータ・コントロール・レジスタ (BRGC)



(f_{PRS} : プリスケアラ出力の選択クロック)

MDL3	MDL2	MDL1	MDL0	k	ポー・レート・ジェネレータの入力クロック ^{注1}
0	0	0	0	0	f _{PRS} /16
0	0	0	1	1	f _{PRS} /17
0	0	1	0	2	f _{PRS} /18
0	0	1	1	3	f _{PRS} /19
0	1	0	0	4	f _{PRS} /20
0	1	0	1	5	f _{PRS} /21
0	1	1	0	6	f _{PRS} /22
0	1	1	1	7	f _{PRS} /23
1	0	0	0	8	f _{PRS} /24
1	0	0	1	9	f _{PRS} /25
1	0	1	0	10	f _{PRS} /26
1	0	1	1	11	f _{PRS} /27
1	1	0	0	12	f _{PRS} /28
1	1	0	1	13	f _{PRS} /29
1	1	1	0	14	f _{PRS} /30
1	1	1	1	15	f _{PRS} ^{注2}

注1 . ASCK入力使用時は、f_{PRS}/16しか選択できません。

2 . 3線式シリアルI/Oモード時のみ使用可能。

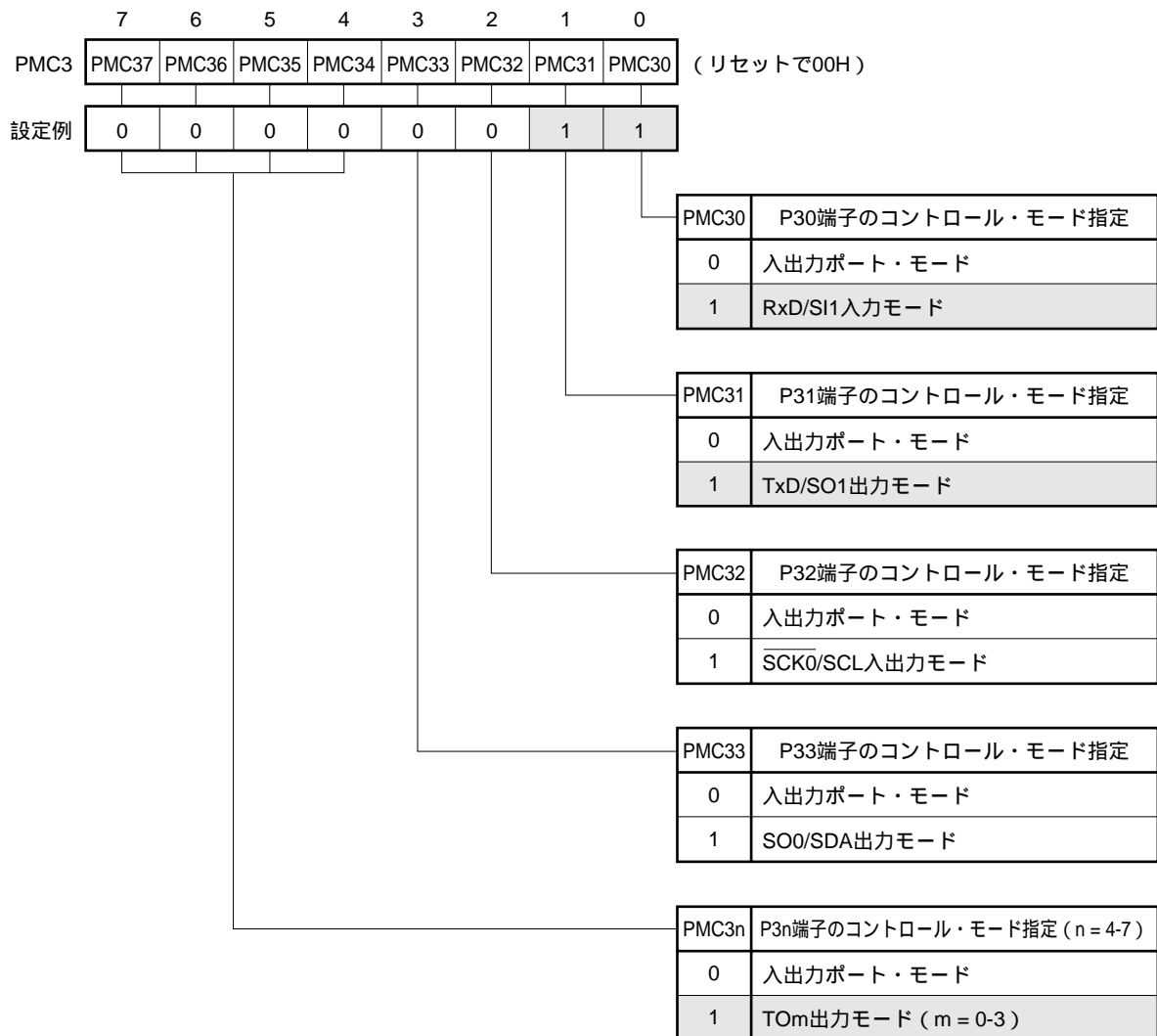
(f_{XX} : 発振周波数または外部クロック入力)

TPS3	TPS2	TPS1	TPS0	n	12ビット・プリスケアラのタップ選択 (f _{PRS})
0	0	0	0	0	f _{XX} /4, f _{ASCK} /2 ^注
0	0	0	1	1	f _{XX} /8, f _{ASCK} /4
0	0	1	0	2	f _{XX} /16, f _{ASCK} /8
0	0	1	1	3	f _{XX} /32, f _{ASCK} /16
0	1	0	0	4	f _{XX} /64, f _{ASCK} /32
0	1	0	1	5	f _{XX} /128, f _{ASCK} /64
0	1	1	0	6	f _{XX} /256, f _{ASCK} /128
0	1	1	1	7	f _{XX} /512, f _{ASCK} /256
1	0	0	0	8	f _{XX} /1024, f _{ASCK} /512
1	0	0	1	9	f _{XX} /2048, f _{ASCK} /1024
1	0	1	0	10	f _{XX} /4096, f _{ASCK} /2048
1	0	1	1	11	f _{XX} /8192, f _{ASCK} /4096
上記以外				設定禁止	

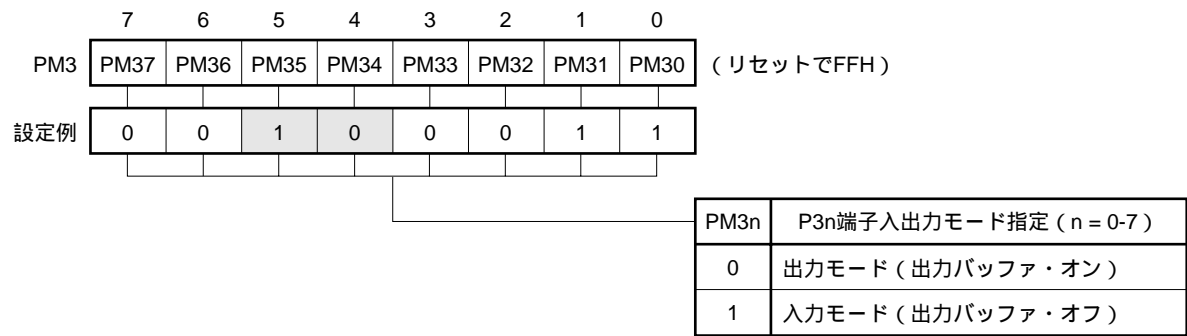
注 MDL3-MDL0でk = 15を選択した場合は、選択できません。

注意 通信動作中にポー・レート・ジェネレータ・コントロール・レジスタ (BRGC) への書き込みを行うと、生成されるポー・レート用クロックが乱れてしまい、正常な動作ができなくなる可能性があります。通信動作中にBRGCへの書き込みは行わないでください。

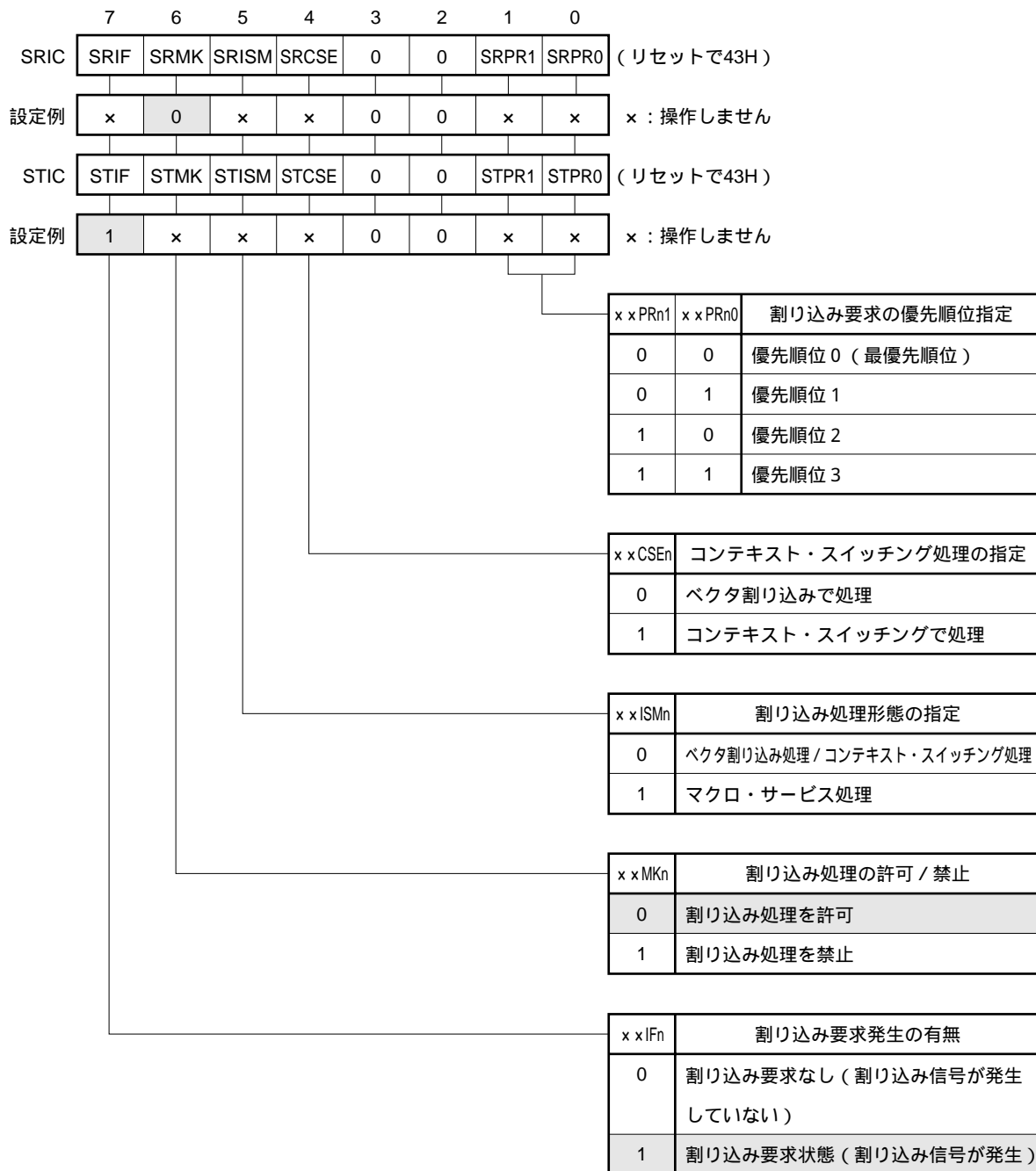
ポート3モード・コントロール・レジスタ (PMC3)



ポート3モード・レジスタ (PM3)



割り込み制御レジスタ



(4) 入力方法

下記に示すRAM領域に次の内容を設定します。

TRN_DT : 送信データを設定します。

BRGCD : ポー・レート・ジェネレータ・コントロール・レジスタ (BRGC) の設定値を設定します。

ポー・レートの計算方法は、次のようになります。

$$\text{ポー・レート} = \frac{f_{xx}}{(k+16) \cdot 2^{n+3}}$$

f_{xx} : 発振周波数または外部クロック入力

k : BRGCのMDL3-MDL0ビットの設定値 ($k = 0-14$)

n : BRGCのTPS3-TPS0ビットの設定値 ($n = 0-11$)

備考 BRGCDの値は、 k , n の値によって決定します。

次に本プログラムで使用するポー・レートを、上記計算式を用いて算出します。

パソコンとの通信を9600 bps ($f_{xx} = 25 \text{ MHz}$)で行います。

$$9600 = \frac{25 \times 10^6}{(k+16) \cdot 2^{n+3}} \quad (\text{内部システム・クロック : } f_{xx}/2)$$

計算式より $k = 4$, $n = 4$ となり、入力パラメータBRGCDへの設定値は '01000100B' となります。

$$\text{ポー・レート} = \frac{25 \times 10^6}{20 \times 2^7} = 9766$$

(5) 出力方法

下記に示すRAM領域/フラグに、次の内容が格納されます。

RCV_DT : 受信データがこのRAM領域に格納されます。

RCVFLAG : 設定回数分の受信が完了すると受信判別フラグ (RCVFLAG) がセット (1) されます。

(6) プログラム説明

イニシャライズ処理 [レーベル名称 : ASY4038]

(a) 各使用ポートをインアクティブに設定します。

・ P30 , P34 ... “ 0 ” 設定

・ P31 , P35 ... “ 1 ” 設定

ハンドシェーク・ライン用の $\overline{\text{RTS}}$ 端子をBUSY状態に設定します。この場合、パソコンからの通信は行われません。

(b) $\overline{\text{RTS}}$ 端子 (P35) を出力ポートに設定します。

(c) アシクロナス・シリアル・インタフェース・モード・レジスタ (ASIM) をイニシャライズします。

(d) ポー・レート・ジェネレータ・コントロール・レジスタ (BRGC) を設定します。

ポー・レートを9600 bpsに設定します。

(e) P30 , P31をコントロール・ポート (RxD , TxD) として使用します。

(f) ハンドシェーク・ライン用の $\overline{\text{CTS}}$ 端子として使用するP34を入力ポートに設定します。

(g) 送信完了割り込み要求フラグ (STIF) をセット (1) します。

送信処理は割り込み処理を使用せず、STIFフラグを判断することで前回の送信が終了したことをチェックしています。1番最初の送信処理判別時に永久ループすることを防ぐために、STIF割り込み要求フラグをセット (1) する必要があります。

(h) 受信完了割り込み要求フラグ (INTSR) のマスクを解除します。

(i) 割り込みを許可します。

(j) $\overline{\text{RTS}}$ 端子のBUSY状態を解除し、送信要求を行います。

注意 送信動作中にはアシクロナス・シリアル・インタフェース・モード・レジスタ (ASIM , ASIM2) を書き換えしないでください。送信中にASIM , ASIM2を書き換えると、それ以降の送信動作ができなくなる場合があります ($\overline{\text{RESET}}$ 入力で復帰可能です) 。

受信処理 [レーベル名称 : INTSR]

(a) $\overline{\text{RTS}}$ 端子をBUSY状態に設定します。

(b) Aレジスタに受信データを、受信バッファ (RXB) から読み込みます。

(c) RAM領域に受信データを保存します。

(d) 受信完了フラグ (RCVFLG) をセット (1) します。

備考 μ PD784038のアシクロナス・シリアル・インタフェースにおいて、受信時に受信エラーが発生した場合には、アシクロナス・シリアル・インタフェース受信終了フラグ（SRIF）とアシクロナス・シリアル・インタフェース受信エラー・フラグ（SERIF）が同時にセット（1）されます。このとき、双方ともベクタ割り込みで処理をする場合には、INTSERの方がデフォルトの優先順位が高いために、先に受け付けられます。

送信処理 [レーベル名称 : TRANS]

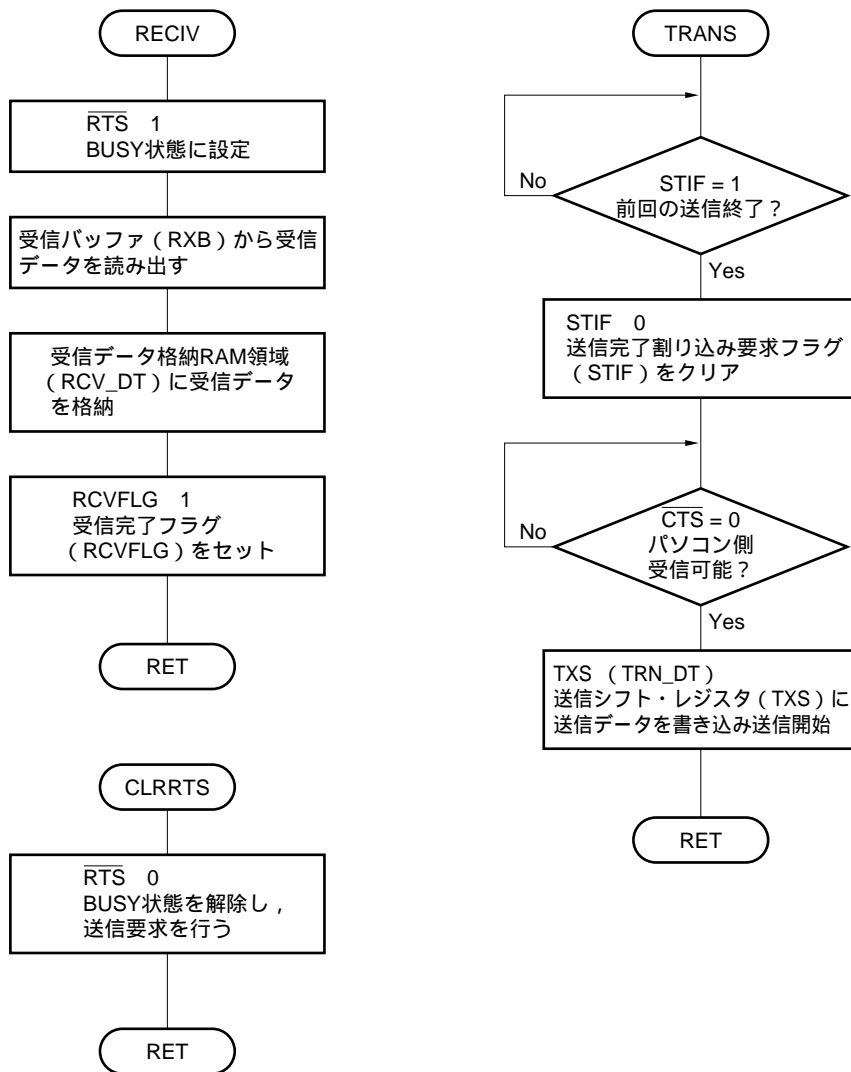
- (a) 前回の送信が終了したことを送信完了割り込み要求フラグ（STIF）でチェックしています。送信完了割り込み要求フラグ（STIF）が1となるまでウエイトします。
- (b) パソコンが受信可能かをチェックします。 $\overline{\text{CTS}}$ 端子がアクティブ状態になるまでウエイトします。
- (c) 送信シフト・レジスタ（TXS）に送信データを書き込みます。

送信要求処理 [レーベル名称 : CLRRTS]

- (a) $\overline{\text{RTS}}$ 端子のBUSY状態を解除し、パソコンに対して送信要求します。

(7) フロー・チャート





(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

RCV_DT : 受信データを格納するRAM領域
 TRN_DT : 送信データを格納するRAM領域
 RCVFLAG : 受信完了判別フラグ
 RCVFLAG = 0...受信未完了
 RCVFLAG = 1...受信完了
 RTS : ハンドシェーク・ライン
 RTS = 0...アクティブ (BUSY解除) 状態
 RTS = 1...BUSY状態

本アプリケーション・ルーチンのプログラム・リスト

```

PUBLIC BRGCD ; PARAMETER
PUBLIC RCVFLAG ; FLAG
EXTRN ASY4038 ; PACKAGE
.
; BAUD RATE : 9600bps
; STOP BIT : 2bit
; CHARACTER LENGTH : 8bit
; PARITY : NO PARITY
;
BRGCD EQU 01000100B ; BRGC DATA (9600bps, fxx=25MHz)
CTS EQU P3.4 ; CTS for RS-232-C BUSY=1 ACTV=0
RTS EQU P3.5 ; RTS for RS-232-C BUSY=1 ACTV=0
.
DSEG
ORG OFE0H
FLAGWK: DS 1 ; FLAGS WORK AREA
TRN_DT: DS 1 ; TRANSMIT DATA
RCV_DT: DS 1 ; RECEIVE DATA
;
;***** define vector table *****
;
INTSRVT CSEG AT 0024H ; INTSR
DW INTSR
.
MAINS CSEG
RST:
LOCATION OH ; specify LOCATION address
.
CALL !ASY4038 ; <<< ASY4038 >>>
    
```

```

SERL1:
    BTCLR    RCVFLAG,$SERJ1      ; CHECK RECEIVE FLAG
    BR       $SERL1
SERJ1:
    MOV      TRN_DT,RCV_DT       ; TRANS DATA
;*****
;
;    trans process
;*****
TRANS:
    BTCLR    STIF,$TRNJ1         ; check complete of transmit
    BR       TRANS
TRNJ1:
    BT       CTS,$TRNJ1         ; check CTS
    MOV      A,TRN_DT
    MOV      TXS,A               ; transmit
CLRRTS:
    CLR1     RTS                 ; clear RTS
    BR       $SERL1
;*****
;
;    INTSR
;    receive process
;*****
INTSR:
    SEL      RB1                 ;
    SET1     RTS                 ; RTS <- 1
    MOV      A,RXB               ; read receive data
    MOV      RCV_DT,A
    SET1     RCVFLAG             ; set receive flag
    RETI

```

```

$ TITLE ('asynchronous application for 784038')
  NAME AS4038

;*****
; asynchronous serial interface for 784038
; at 25MHz
;*****

PUBLIC ASY4038
EXTRN BRGCD
EXTBIT RCVFLG ; receive flag
;
RTS EQU P3.5 ; RTS for RS-232-C

; %%% FOR ASYNC %%%
;
; BAUD RATE : 9600bps
; STOP BIT : 2bit
; CHARACTER LENGTH : 8bit*
; PARITY : NO PARITY
;
CSEG
ASY4038:
MOV P3,#00100010B ; P30,P31,P34,P35 inactive set
CLR1 PM3.5 ; P3.5=OUT
MOV ASIM,#11101101B ; ASIM initialize
MOV BRGC,#LOW(BRGCD) ; Baud rate = 9600bps
OR PMC3,#00000011B ; P30,P31 = Control port
SET1 PM3.4 ; P3.4=IN

SET1 STIF ; dummy set INTST
CLR1 SRMK ; open mask of INTSR
EI ; interrupt enable

CLR1 RTS ; receive enable

RET

```

3.4 クロック同期式シリアル・インタフェースのプログラム例

クロック同期式シリアル・インタフェースは、マスタ・デバイスがシリアル・クロックを供給し、このクロックに同期して1バイトのデータを通信する方法です。

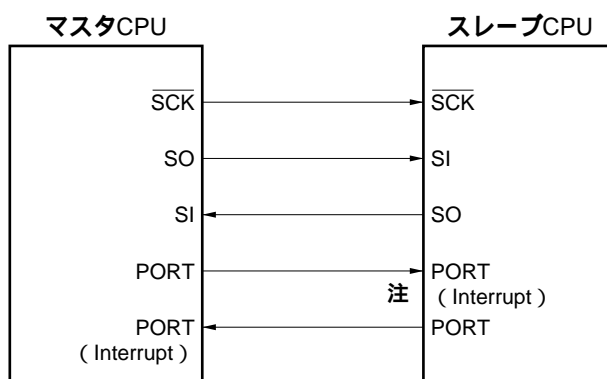
クロック同期式シリアル・インタフェースの動作モードに、3線式シリアルI/Oモードがあります。

3線式シリアルI/Oモードは、シリアル・クロック ($\overline{\text{SCK0}}$) と、シリアル・パス (SO0, SI0) の3本のラインによって8ビット長のデータ転送を行います。従来のクロック同期式シリアル・インタフェースを内蔵する周辺I/Oや表示コントローラなどを接続する場合に有効です。

(1) システム構成例

図3 - 8 に、3線式シリアルI/Oモードのシステム構成例を示します。

図3 - 8 3線式シリアルI/Oモードのシステム構成例



注 ハンドシェイク・ライン

(2) 通信動作

(a) 送信のみ動作許可

送信動作は、クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) のCTXEフラグがセット (1) されているときに、シフト・レジスタ (SIO) への書き込みが行われると起動されます。

シリアル・クロックは、内部クロックと外部クロックのいずれかを選択できます。ここでは内部クロックをシリアル・クロックとして選択した場合を説明します。

・シリアル・クロックとして内部クロックを選択した場合

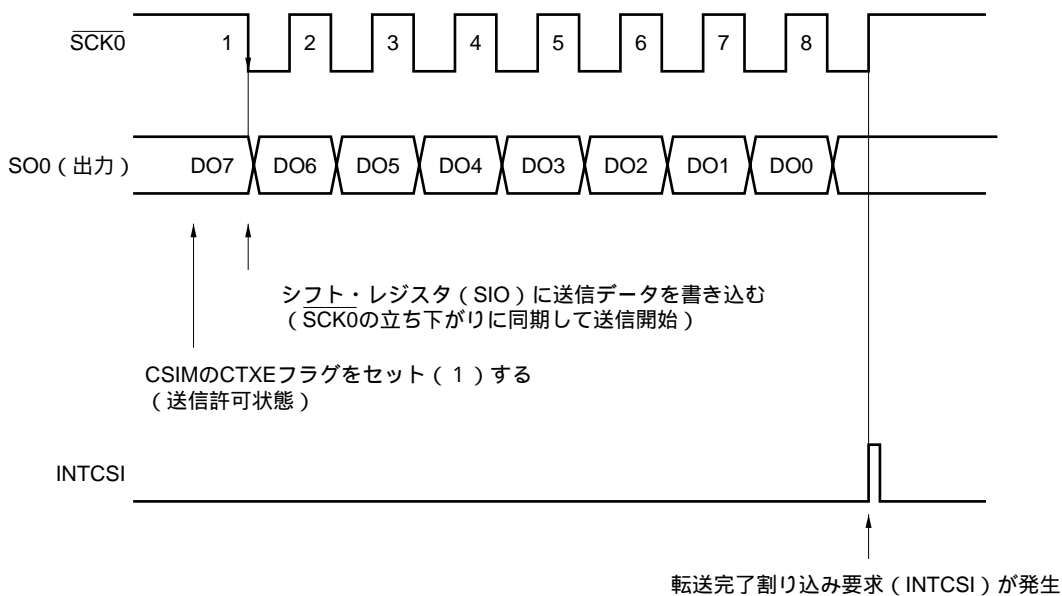
送信が起動されると、シリアル・クロックを $\overline{\text{SCK0}}$ 端子から出力し、同時にシリアル・クロックの立ち下がりに同期してSIOからデータをSO0端子へ順次出力します。

なお、送信の起動から $\overline{\text{SCK0}}$ の最初の立ち下がりにまでには、最大 $\overline{\text{SCK0}}$ 1クロック分の時間がかかります。

図3 - 9に、3線式シリアルI/Oモード (送信時) のタイミングを示します。

図3 - 9 3線式シリアルI/Oモード (送信時) のタイミング

・MSBファースト時



・送信動作の中断

送信動作中に送信を禁止 (CTXEフラグをクリア (0)) すると、次の $\overline{\text{SCK0}}$ の立ち上がりで $\overline{\text{SCK0}}$ クロックの出力を停止し、送信動作を中断します。このとき、転送完了割り込み要求 (INTCSI) は発生しません。また、SO0端子はハイ・インピーダンス状態になります。

(b) 受信のみ動作許可

受信動作は、クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) のCRXEフラグがセット (1) されているときに行われます。CRXEフラグを“0”から“1”へ変化させるか、シフト・レジスタ (SIO) からの読み出しを行うと受信動作は起動されます。

シリアル・クロックは、内部クロックと外部クロックのいずれかを選択できます。ここでは、内部クロックをシリアル・クロックとして選択した場合を説明します。

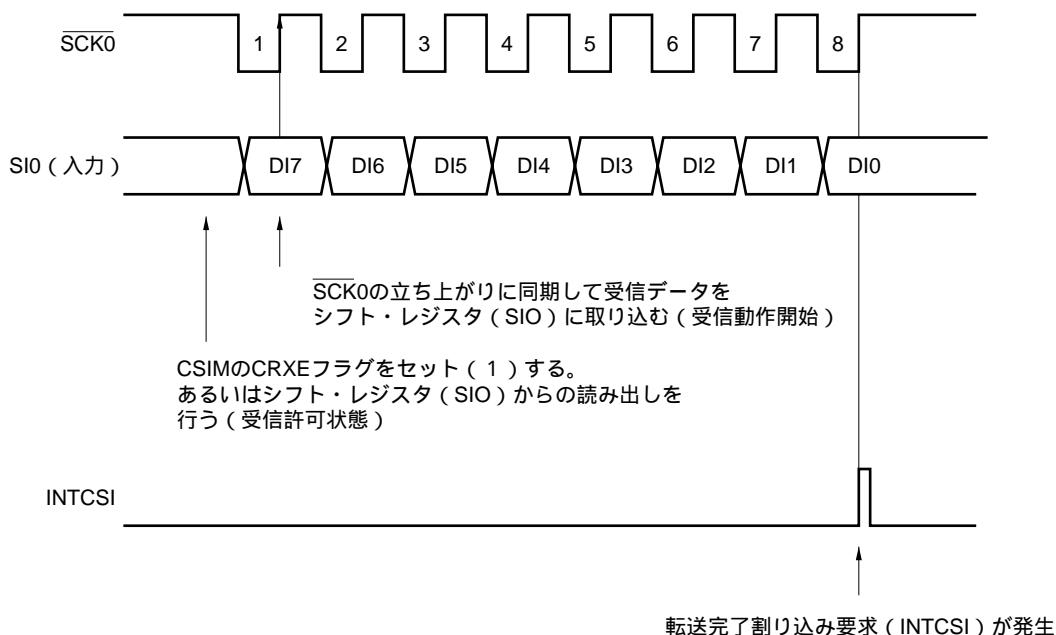
・シリアル・クロックとして内部クロックを選択した場合

受信が起動されると、シリアル・クロックを $\overline{\text{SCK0}}$ 端子から出力し、シリアル・クロックの立ち上がりに同期して、SIO端子のデータをシフト・レジスタ (SIO) へ順次取り込みます。なお、受信の起動から $\overline{\text{SCK0}}$ の最初の立ち下がりまでには、最大 $\overline{\text{SCK0}}$ 1クロック分の時間がかかります。

図3 - 10に、3線式シリアルI/Oモード (受信時) のタイミングを示します。

図3 - 10 3線式シリアルI/Oモード (受信時) のタイミング

・MSBファースト時



・受信動作の中断

受信動作中に受信を禁止 (CTXEフラグをクリア (0)) すると、次の $\overline{\text{SCK0}}$ の立ち上がりで $\overline{\text{SCK0}}$ クロックの出力を停止し、受信動作を中断します。このとき、転送完了割り込み要求 (INTCSI) は発生しません。また、SIOの内容は不定になります。

(c) 送受信動作

クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) のCTXEフラグとCRXEフラグが両方ともセット (1) されているときに、送信動作と受信動作を同時に行うことができます。CRXEフラグを“0”から“1”へ変化させるか、シフト・レジスタ (SIO) からの読み出しを行うと送受信動作は起動されます。

シリアル・クロックは、内部クロックと外部クロックのいずれかを選択できます。ここでは、内部クロックをシリアル・クロックとして選択した場合を説明します。

・シリアル・クロックとして内部クロックを選択した場合

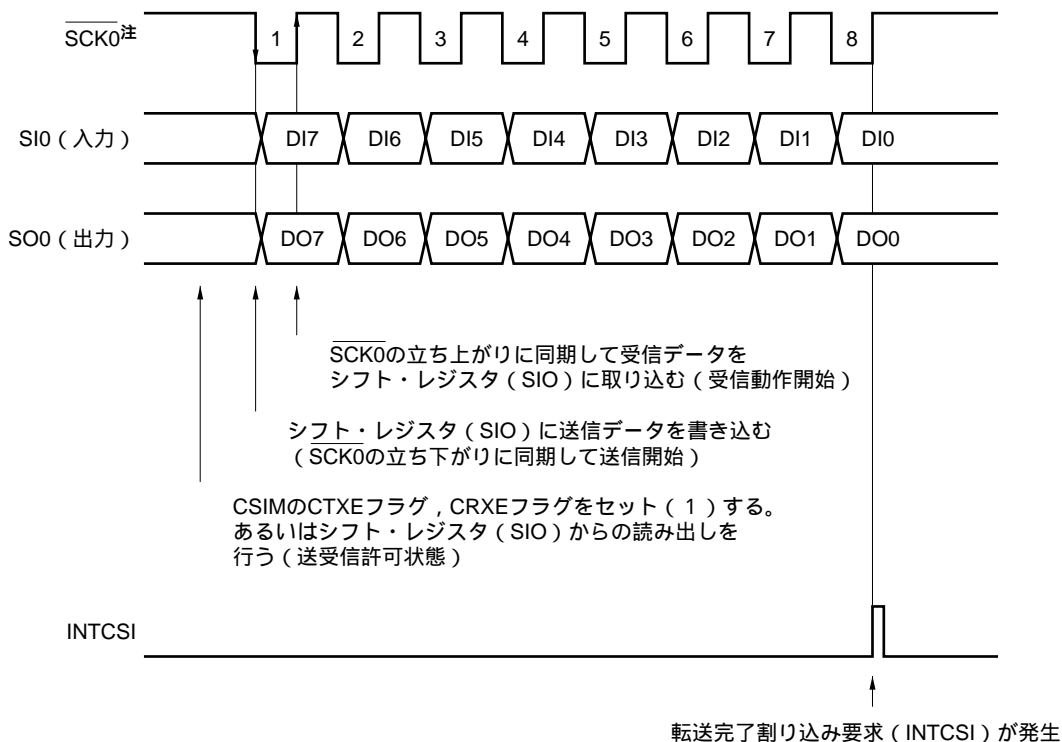
送受信が起動されると、シリアル・クロックを $\overline{\text{SCK0}}$ 端子から出力され、シリアル・クロックの立ち下がりに同期してシフト・レジスタ (SIO) からSO0端子へ順次データが出力され、シリアル・クロックの立ち上がりに同期して、SI0端子のデータを順次シフト・レジスタ (SIO) へ、シフト・インします。

なお、送受信の起動から $\overline{\text{SCK0}}$ の最初の立ち下がりにまでには、最大 $\overline{\text{SCK0}}$ 1クロック分の時間がかかります。

図3 - 11に、3線式シリアルI/Oモード (送受信時) のタイミングを示します。

図3 - 11 3線式シリアルI/Oモード (送受信時) のタイミング

・MBSファースト時



注意 送受信動作の1回目の起動時は、必ずCRXEフラグが“0”から“1”へ変化するので、すぐに送受信動作が起動され、不定なデータが出力される可能性があります。したがって、送信および受信とも禁止されているときに、あらかじめシフト・レジスタ(SIO)へ最初の送信データを書き込んでから、送受信を許可してください。

・送信および受信動作の中断

送受信動作中に送信または受信の一方を禁止にした場合は、禁止された動作のみが中断されます。送受信を同時に禁止した場合は、 $\overline{\text{SCK0}}$ の次の立ち上がりで $\overline{\text{SCK0}}$ クロックの出力を停止し、送信および受信動作を中断します。この場合、シフト・レジスタ(SIO)の内容は不定となり、割り込み要求(INTCSI)は発生しません。また、SO0端子は出力ハイ・インピーダンス状態になります。

3.4.1 3線式シリアルI/Oモードのプログラム例

ここではμPD784038を代表製品として説明します。

(1) 処理概要

(a) 仕様

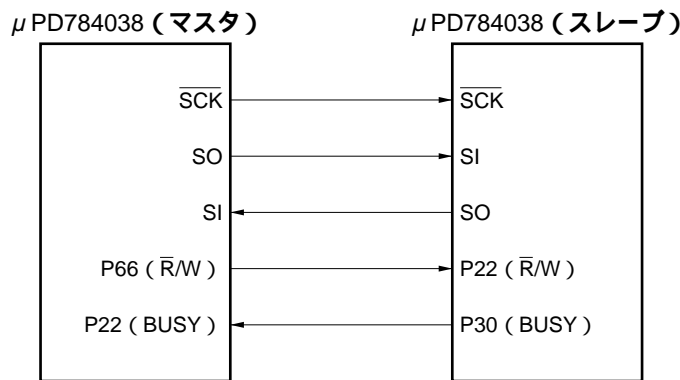
3線式シリアルI/Oモードのプログラム例として、μPD784038間での通信例を紹介します。

(b) 使用周辺機能説明

μPD784038を2つ使用し、それぞれマスタ、スレーブとします。

図3-12に、接続図を示します。

図3-12 マスタ、スレーブとの接続図



通信プロトコルの仕様を次に示します。

ボー・レートは312.5 kbpsとします。ただし、 $f_{xx} = 25 \text{ MHz}$ とします。ハンドシェイク・ラインとして、 \bar{R}/W ^{注1}とBUSY^{注2}を設けます。マスタとスレーブで交互に送信を行います。

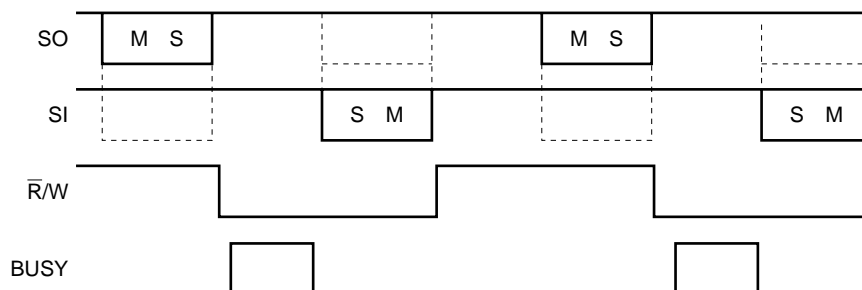
注1. \bar{R}/W 信号 (マスタ スレーブ) : 送信中 (READ) または受信 (WRITE) 状態を判別するための信号。

2. BUSY信号 (スレーブ マスタ) : スレーブ側のBUSY状態をマスタ側に知らせるための信号。

	0	1
\bar{R}/W 信号	READ	WRITE
BUSY信号	ACTIV	BUSY

図3 - 13に、マスタから見たタイミング・チャートを示します。

図3 - 13 マスタ側から見たタイミング・チャート



備考 M S: マスタからスレーブへの送信
S M: スレーブからマスタへの送信

(2) 使用RAM領域

本プログラムで、マスタは表3 - 6に示すように、スレーブは表3 - 7に示すようにRAM領域を使用します。

使用するRAM領域およびフラグ領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

【マスタ側の使用RAM領域】

表3 - 6 3線式シリアル・インタフェースのプログラム例(マスタ側)で使用するRAM/フラグ

RAM/フラグ名称	用途	バイト数	初期値
RCV_DT	3線式シリアル・インタフェースで受信したデータを格納するワーク領域	1	-
TRN_DT	3線式シリアル・インタフェースで送信するためのデータを設定するワーク領域	1	-
TNEDFLAG	送信完了フラグ TNEDFLAG = 0...送信未完了 TNEDFLAG = 1...送信完了	1ビット	0
RVEDFLAG	受信完了フラグ RVEDFLAG = 0...受信未完了 RVEDFLAG = 1...受信完了	1ビット	0
MBSYFLAG	BUSY信号フラグ MBSYFLAG = 0...BUSY解除状態 MBSYFLAG = 1...BUSY状態	1ビット	0
MRWFLAG	通信方向フラグ MRWFLAG = 0...スレーブ側 マスタ側 MRWFLAG = 1...マスタ側 スレーブ側	1ビット	0

【スレーブ側の使用RAM領域】

表3 - 7 3線式シリアル・インタフェースのプログラム例（スレーブ側）で使用するRAM / フラグ

RAM / フラグ名称	用 途	バイト数	初期値
RCV_DT	3線式シリアル・インタフェースで受信したデータを格納するワーク領域	1	-
TRN_DT	3線式シリアル・インタフェースで送信するためのデータを設定するワーク領域	1	-
RVEDFLAG	受信完了フラグ RVEDFLAG = 0...受信未完了 RVEDFLAG = 1...受信完了	1ビット	0
SBSYFLAG	BUSY信号フラグ SBSYFLAG = 0...BUSY解除状態 SBSYFLAG = 1...BUSY状態	1ビット	0
SRWFLAG	通信方向フラグ SRWFLAG = 0...スレーブ側 マスタ側 SRWFLAG = 1...マスタ側 スレーブ側	1ビット	0

(3) 使用レジスタ

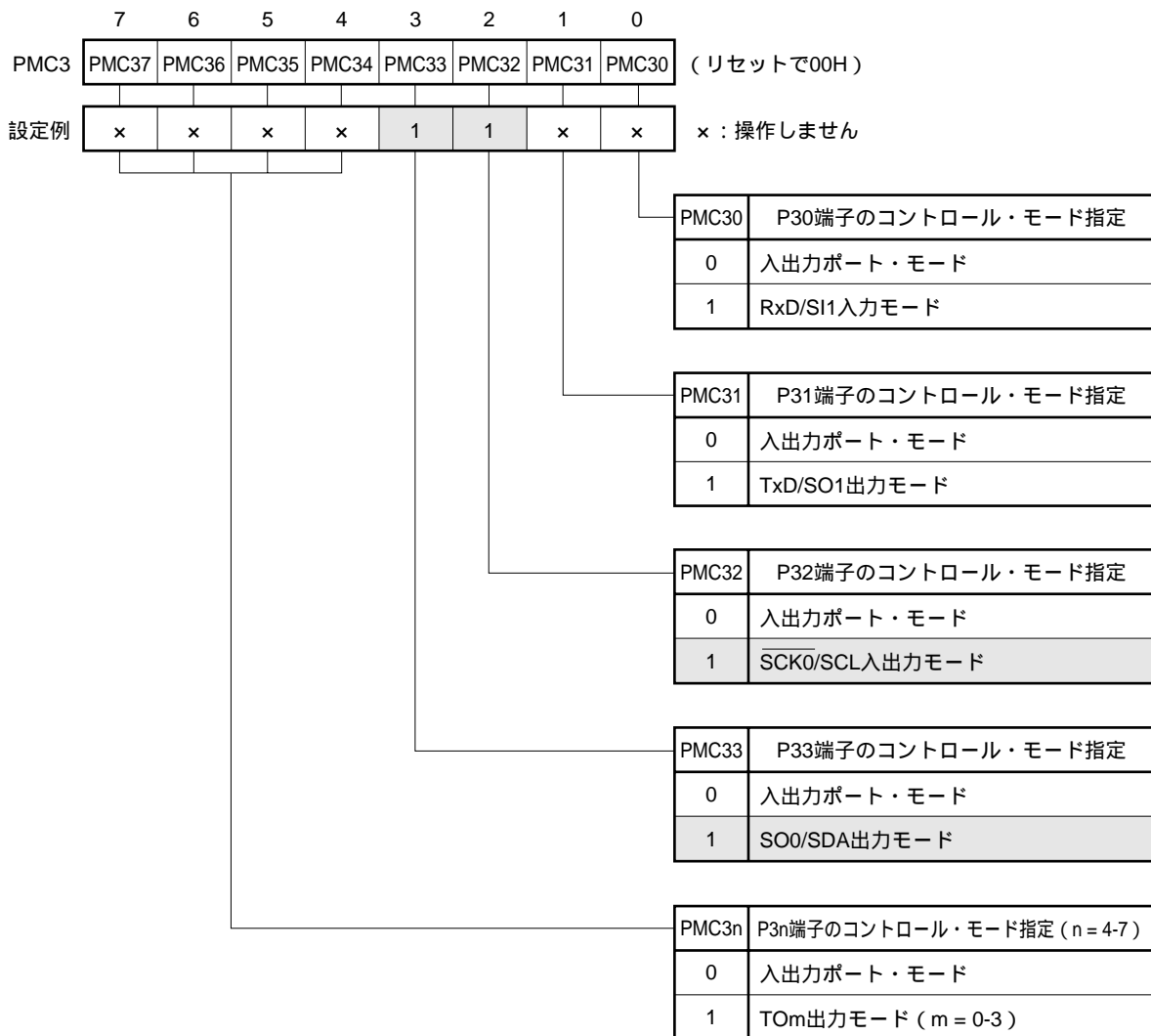
【マスタ側の使用レジスタ】

(a) 汎用レジスタ

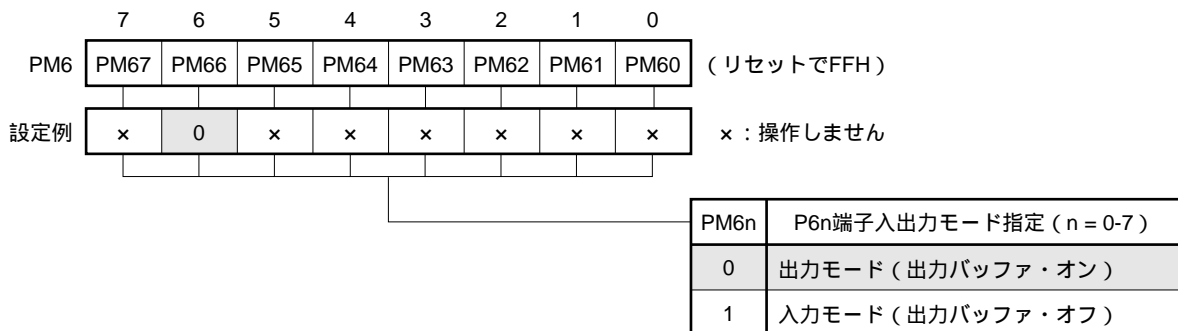
- ・イニシャライズ処理（レーベル名称：CLKMIN）
なし
- ・送信処理（レーベル名称：TRANS）
Aレジスタ
- ・受信処理（レーベル名称：RECIIVE）
なし
- ・転送終了割り込み処理（レーベル名称：ENDTR）
Aレジスタ

(b) 特殊機能レジスタ

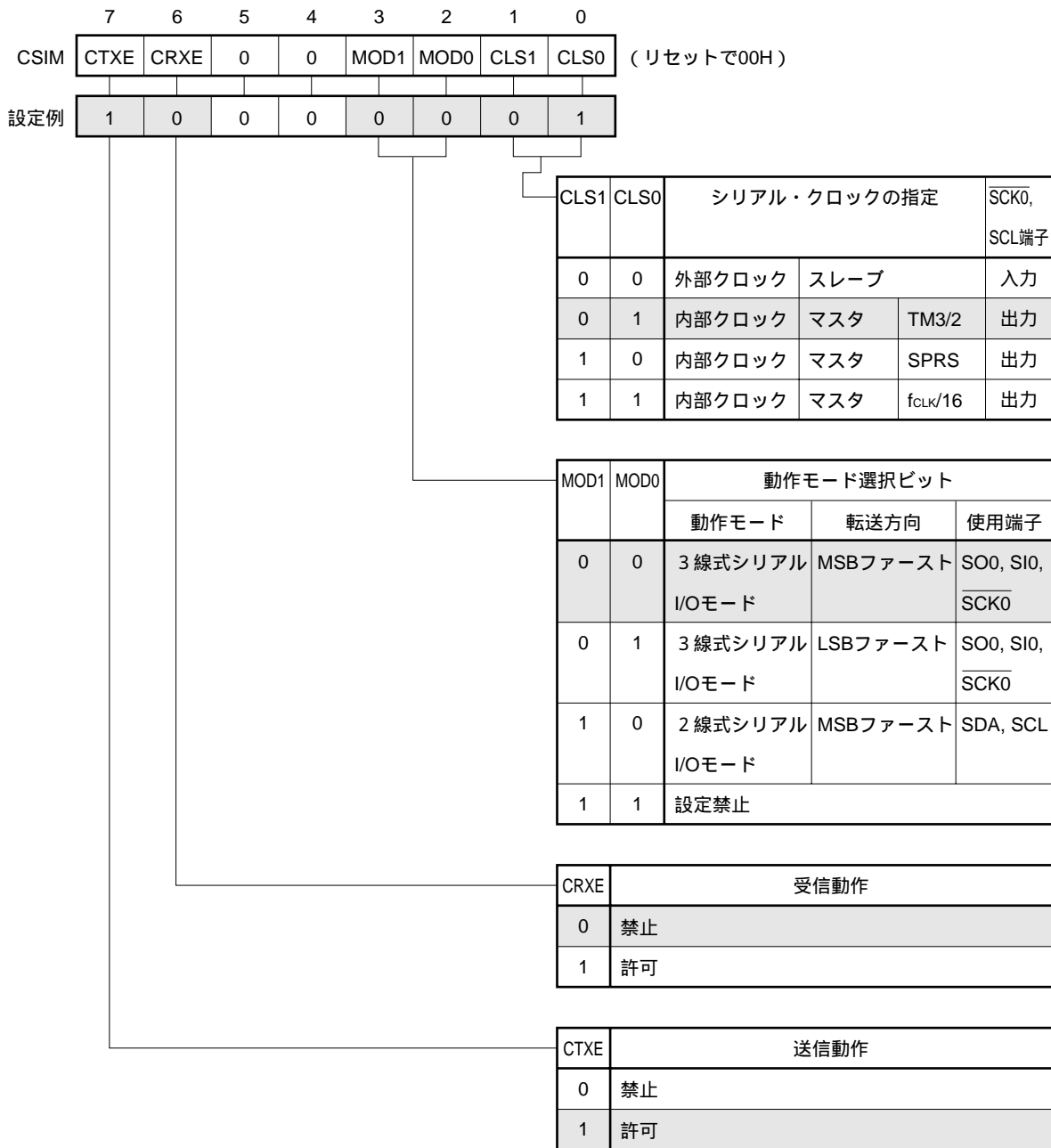
ポート3モード・コントロール・レジスタ (PMC3)



ポート6モード・レジスタ (PM6)



クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM)

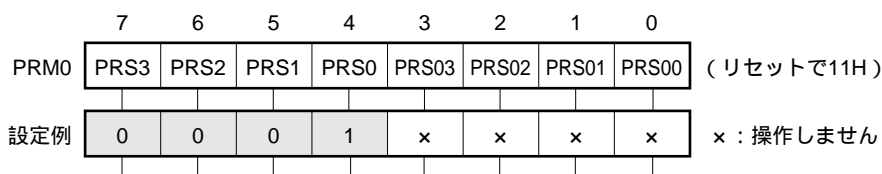


注意 CTXE = 0, CRXE = 1の受信状態からCTXE = 1, CRXE = 0の送信状態に変更する場合、および、CTXE = 1, CRXE = 0の送信状態からCTXE = 0, CRXE = 1の受信状態に変更する場合は、1命令で行わないようにしてください。1命令で変更を行った場合、シリアル・クロック・カウンタが誤動作し、変更後の1回目の通信が8ビット未満で終了してしまいます。したがって、次の例に示すように2命令に分けて変更してください。

例 CTXE = 1, CRXE = 0をCTXE = 0, CRXE = 1に変更する場合

```
CLR1 CTXE
SET1 CRXE
```

プリスケアラ・モード・レジスタ0 (PRM0)

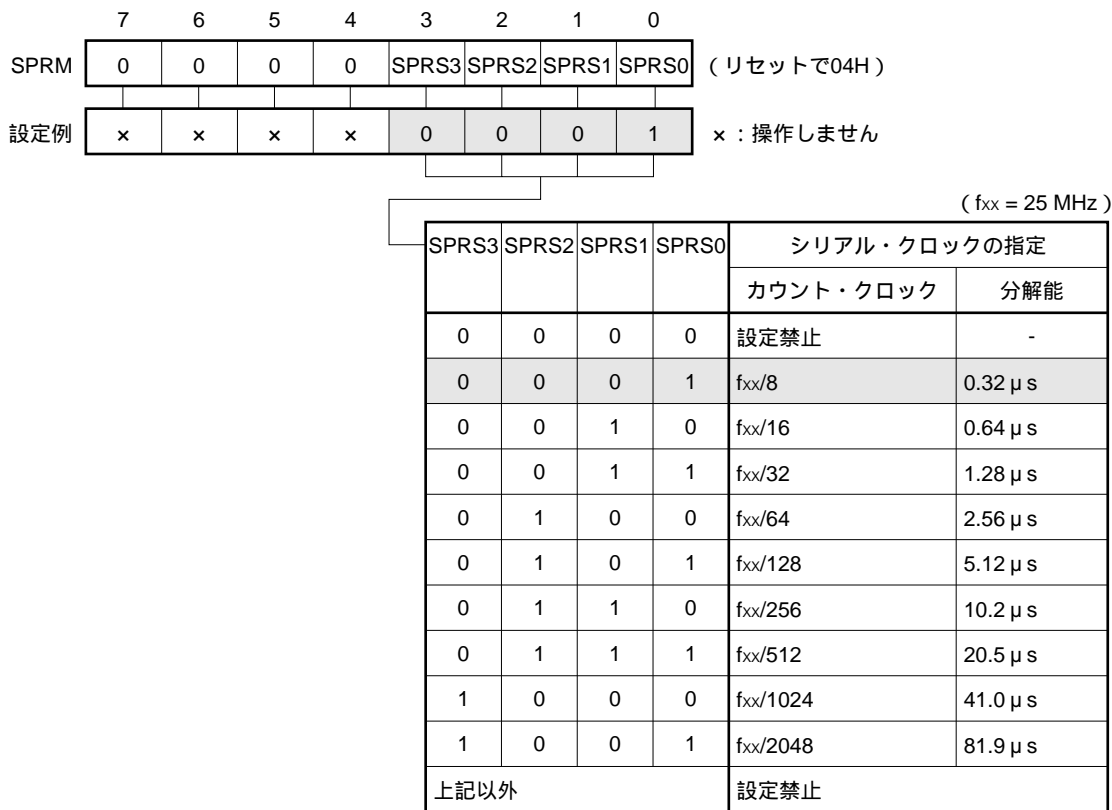


タイマ/カウンタ0のTM0のカウンタ・クロックを指定します。

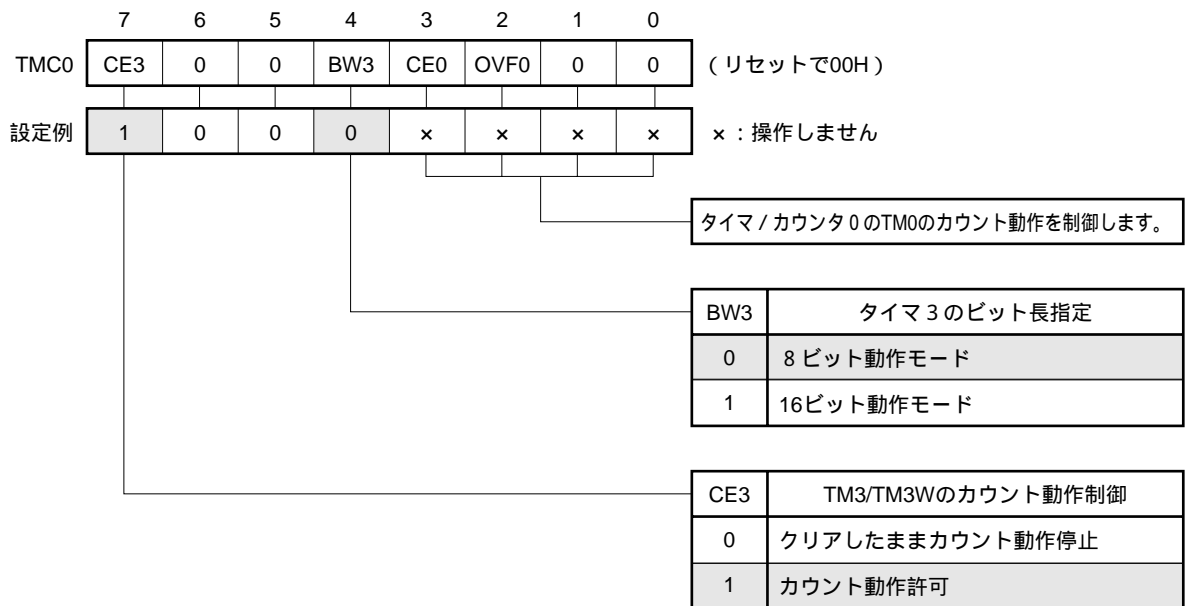
(f_{xx} = 25 MHz)

PRS3	PRS2	PRS1	PRS0	タイマ3のTM3/TM3Wのカウンタ・クロック指定	
				カウンタ・クロック [Hz]の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
上記以外				設定禁止	

シリアル・クロック用プリスケアラ・モード・レジスタ (SPRM)



タイマ・コントロール・レジスタ0 (TMC0)



割り込み制御レジスタ



外部割り込みモード・レジスタ0 (INTM0)



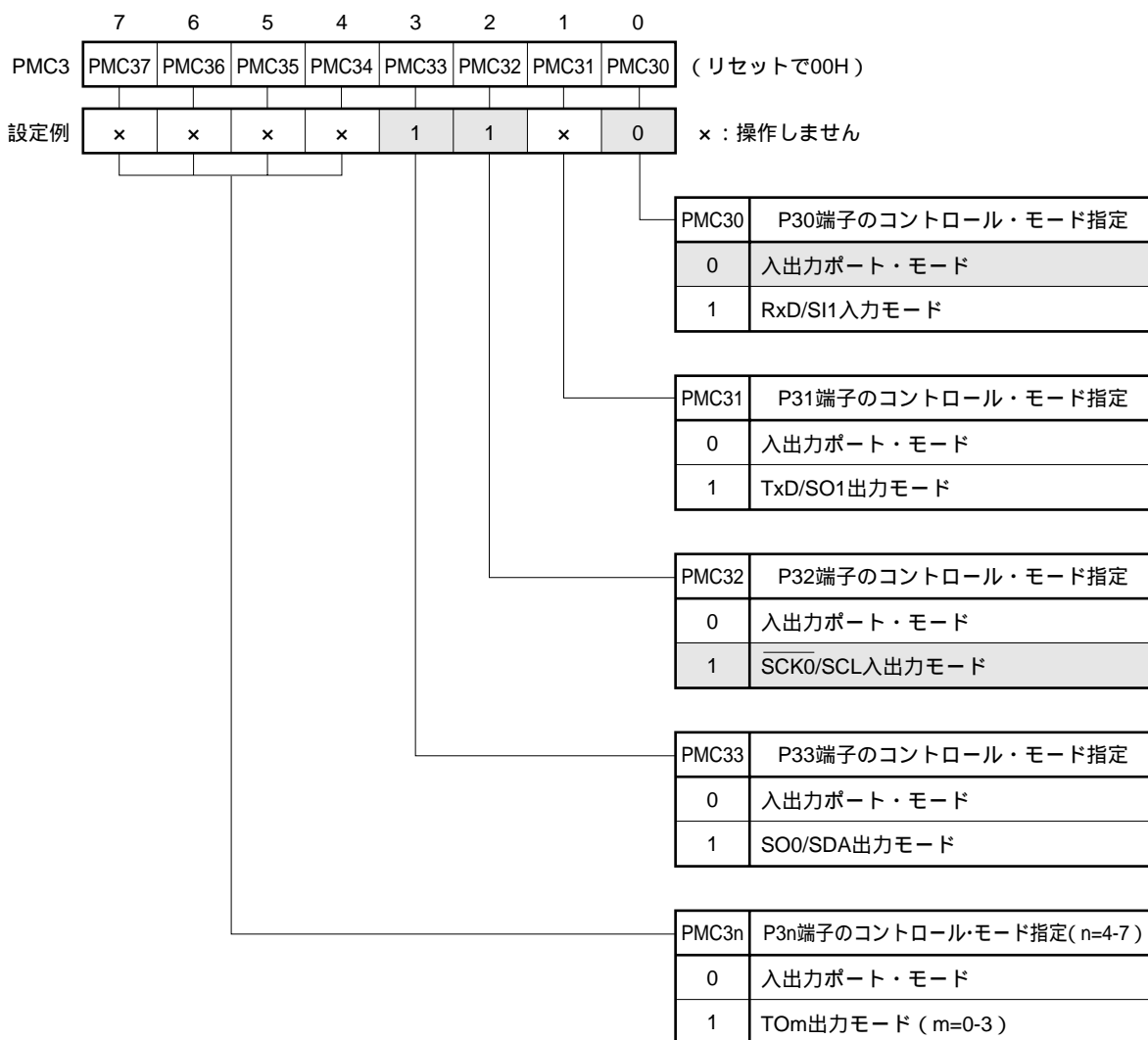
【スレーブ側の使用レジスタ】

(a) 汎用レジスタ

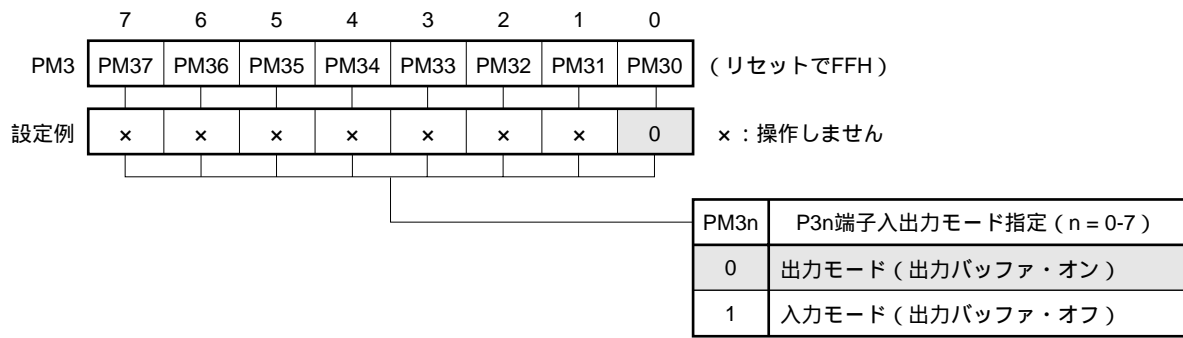
- ・イニシャライズ処理 (レーベル名称: CLKSIN)
なし
- ・送信処理 (レーベル名称: TRANS)
Aレジスタ
- ・転送終了割り込み処理 (レーベル名称: ENDTR)
Aレジスタ

(b) 特殊機能レジスタ

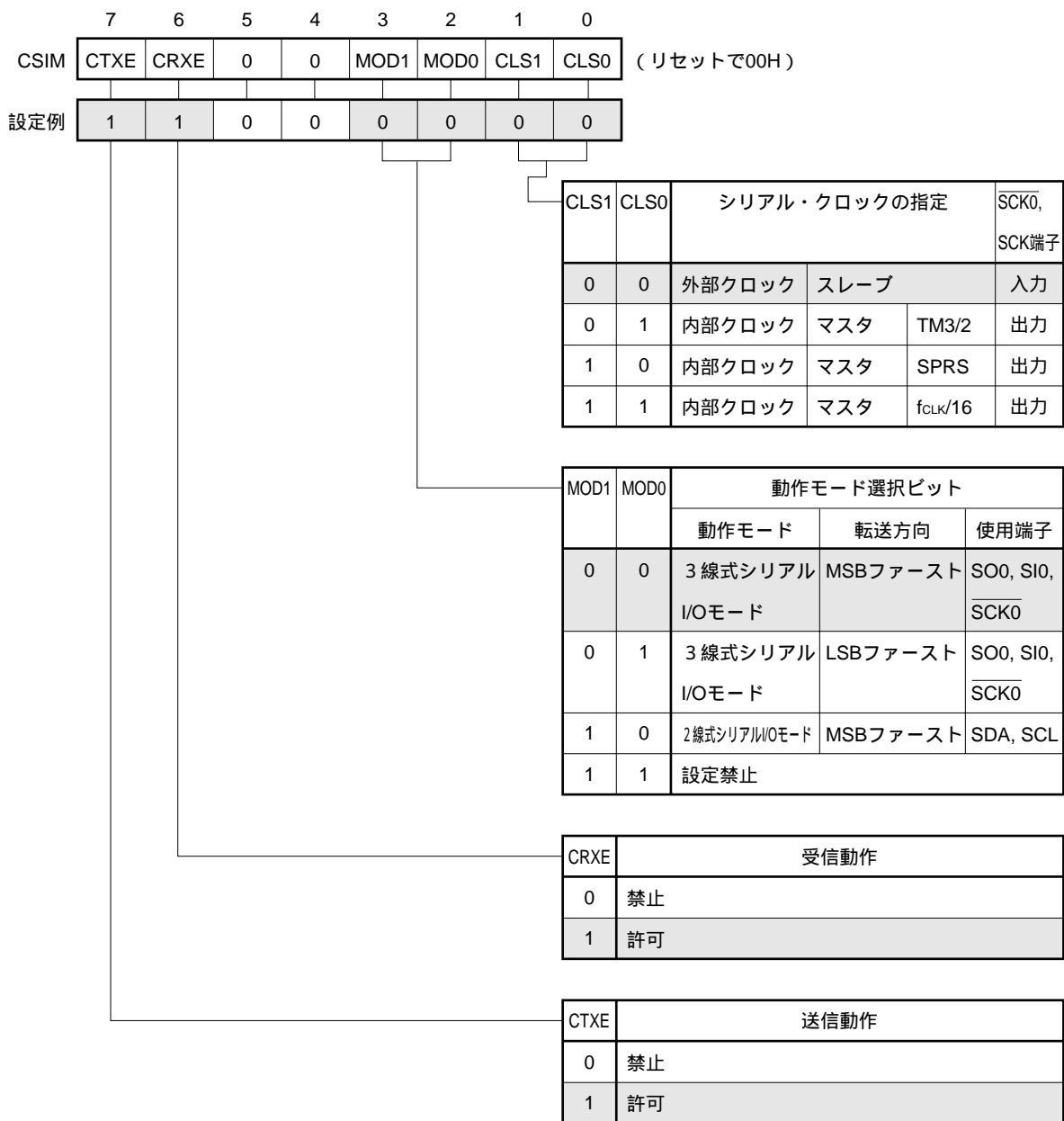
ポート3モード・コントロール・レジスタ (PMC3)



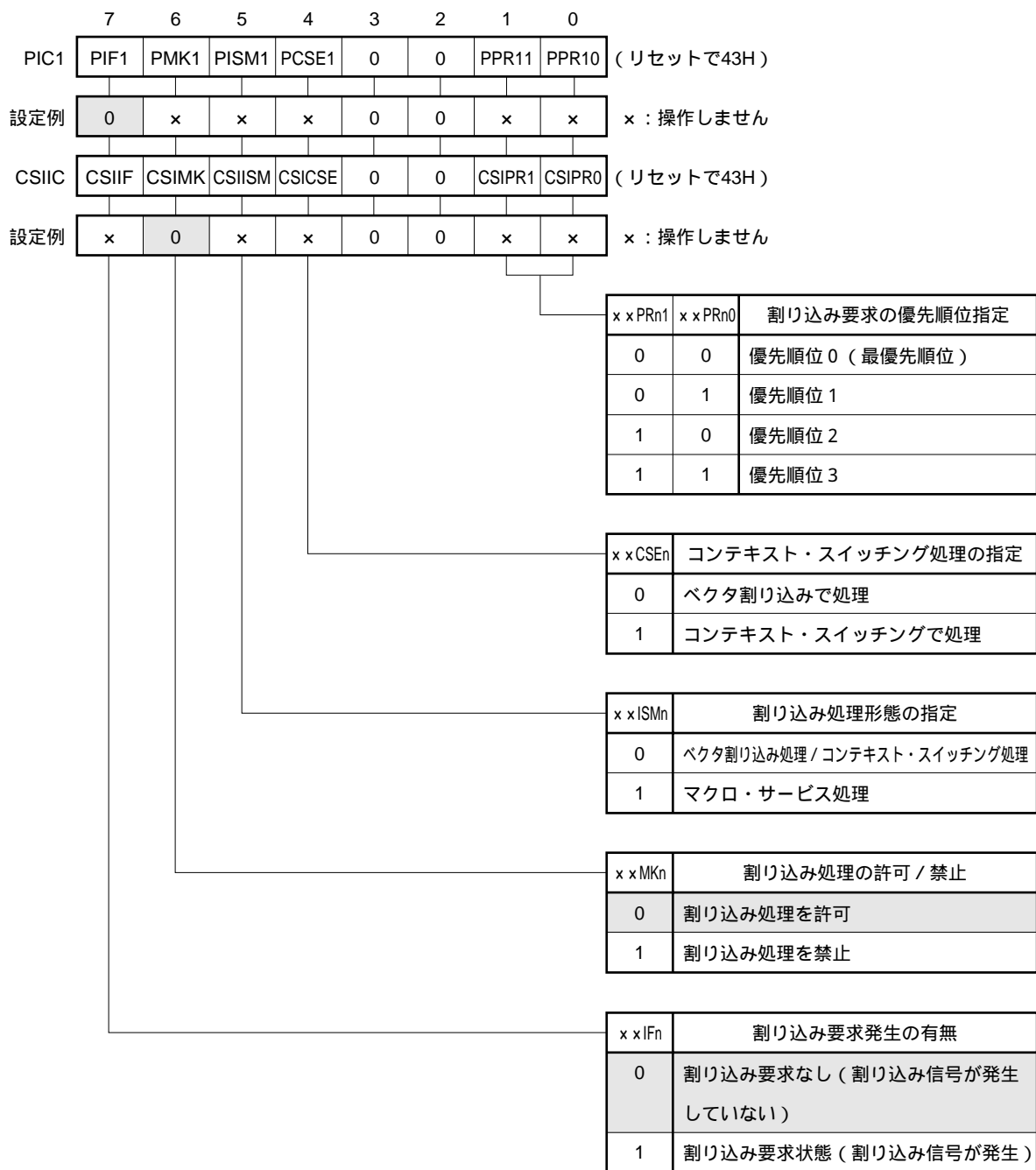
ポート3モード・レジスタ (PM3)



クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM)



割り込み制御レジスタ



外部割り込みモード・レジスタ0 (INTM0)



(4) 入力方法**・マスタ側**

下記に示すRAM領域に、次の内容を設定します。

PRMOD : プリスケアラ・モード・レジスタ0 (PRM0) に設定する値です。

CR30D : タイマ3のコンペア・レジスタ (CR30) に設定する値です。

(PRMOD, CR30Dはシリアル・クロック (ボー・レート) の設定用です)。

ボー・レートの計算方法は、次のとおりです。

$$\begin{aligned} \text{ボー・レート} &= \frac{\text{TM3のカウント・クロック}}{\text{CR30} + 1} \times \frac{1}{2} \\ &= \frac{25 \times 10^6 / 8}{\text{CR30} + 1} \times \frac{1}{2} \end{aligned}$$

したがって、312.5 kbpsの場合、PRM0 = 0001 × × × B (× : 操作しません) , CR30 = 4です。

TRN_DT : 送信データを設定します。

・スレーブ側

下記に示すRAM領域に、次の内容を設定します。

TRN_DT : 送信データを設定します。

(5) 出力方法**・マスタ側**

下記に示すRAM領域に、次の内容が格納されます。

RCV_DT : 受信データが格納されます。

TNEDFLAG : 送信完了フラグです。1バイト送信するごとにセット (1) されます。

RVEDFLAG : 受信完了フラグです。INTCSI割り込み処理により1バイト受信するごとにセット (1) されます。

・スレーブ側

下記に示すRAM領域に、次の内容が格納されます。

RCV_DT : 受信データが格納されます。

RVEDFLAG : 受信完了フラグです。INTCSI割り込み処理により1バイト受信するごとにセット(1)されます。

(6) プログラム説明

【マスタ側のプログラム説明】

イニシャライズ処理 [レーベル名称 : CLKMIN]

- (a) 割り込み禁止状態 (DI) に設定します。
- (b) P32, P33をコントロール・ポート ($\overline{\text{SCK0}}$, SO0) として設定します。
- (c) 通信方向フラグ (RWFLAG) をセット (1) し、通信方向を “ マスタ スレーブ ” とします。
- (d) P66 ($\overline{\text{R/W}}$) を出力ポートに設定します。
- (e) クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) で3線式シリアルI/Oモードのイニシャライズを行い、送信を許可します。
- (f) シリアル・クロック (ボー・レート) の設定を行います。
- (g) BUSY端子 (INTP1) の有効エッジを立ち下がりエッジに設定します。
- (h) INTP1割り込み要求フラグ (PIF1) をクリア (0) します。
- (i) クロック同期式シリアル・インタフェース通信終了割り込みのマスク (CSIMK) を解除します。

送信処理 [レーベル名称 : TRANS]

- (a) スレーブ側のBUSY信号フラグ (SBSYFLAG) が解除され、マスタ側のBUSY信号フラグ (MBSYFLAG) が解除されるまでウエイトします。
- (b) 送信許可とします。
- (c) シフト・レジスタ (SIO) に送信データを書き込みます。

受信処理 [レーベル名称 : RECIVE]

- (a) 受信許可状態にします。

通信終了割り込み処理 [レーベル名称 : ENDTR]

- (a) 通信方向フラグ (RWFLAG) を反転します。
- (b) 通信方向フラグ (RWFLAG) の状態を判別します。
RWFLAG = 0 (スレーブ マスタ) ... (c) の処理へ進む。
RWFLAG = 1 (マスタ スレーブ) ... (f) の処理へ進む。
- (c) スレーブ・マイコンが受信を完了したかどうかを、INTP1割り込み要求フラグ (PIF1) のポーリングで判別します。
- (d) 送信を禁止し、送信完了フラグ (TNEDFLAG) をセット (1) して処理を終了します。
- (e) 受信を禁止し、受信データを格納するRAM領域 (RCV_DT) に書き込みます。
- (f) 受信完了フラグ (RVEDFLAG) をセット (1) して処理を終了します。

【スレーブ側のプログラム説明】

イニシャライズ処理 [レーベル名称 : CLKSIN]

- (a) P32, P33コントロール・ポート ($\overline{\text{SCK0}}$, SO0) として設定します。
- (b) P30をBUSY端子として用いるため、出力ポートとして設定します。
- (c) クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) で3線式シリアルI/Oモードのイニシャライズを行い、送信を許可します。
- (d) $\overline{\text{R/W}}$ 端子 (INTP1) の有効エッジを両エッジに設定します。
- (e) 受信完了フラグ (RVEDFLAG) をクリア (0) します。
- (f) INTP1割り込み要求フラグ (PIF1) をクリア (0) します。
- (g) クロック同期式シリアル・インタフェース通信終了割り込みのマスク (CSIMK) を解除します。

通信処理 [レーベル名称 : TRANS]

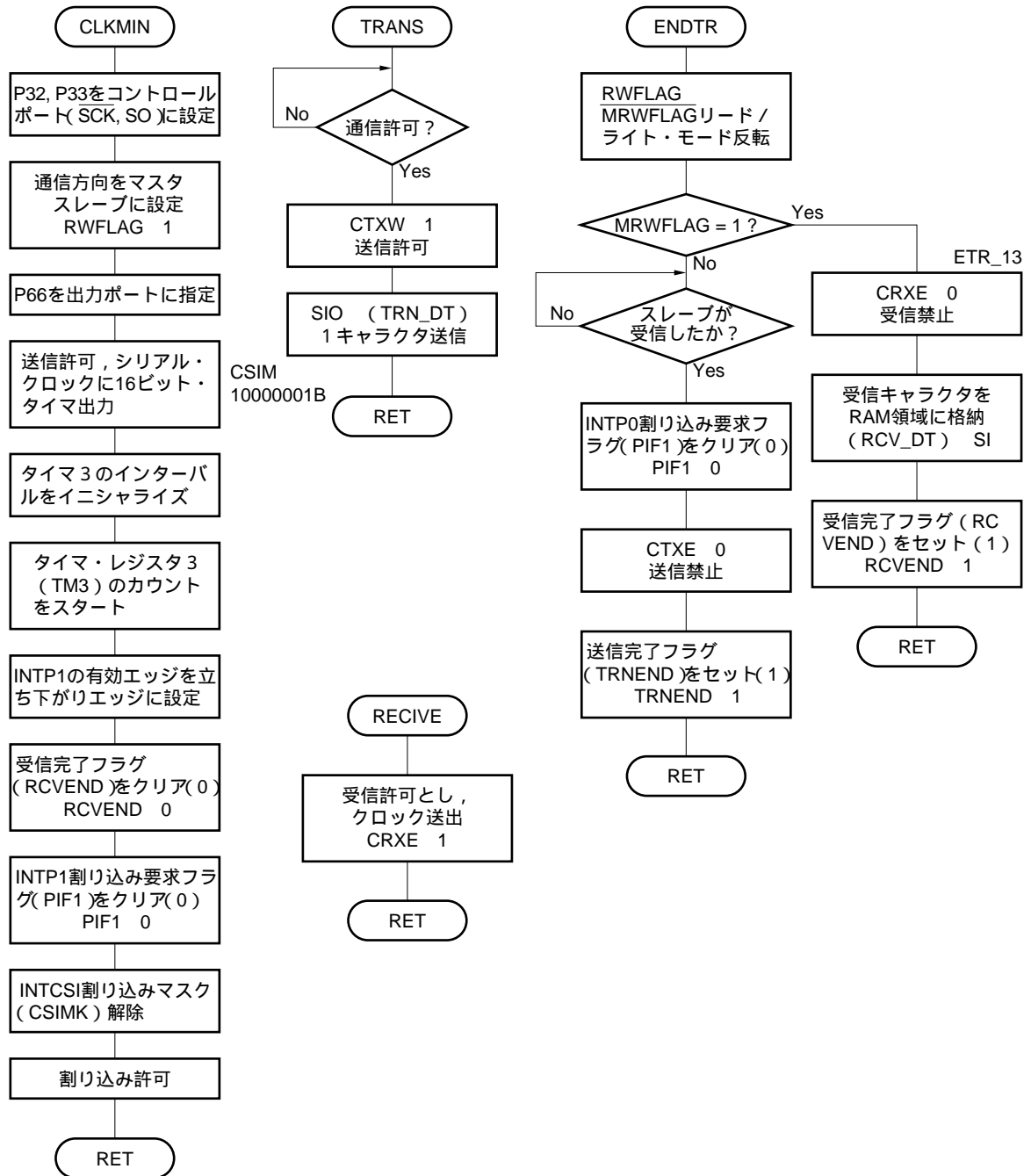
- (a) 送信データをシフト・レジスタ (SIO) に書き込みます。
- (b) BUSY信号フラグ (SBSYFLAG) をクリア (0) し、マスタ・マイコンに通信を許可します。

通信終了割り込み処理 [レーベル名称 : ENDTR]

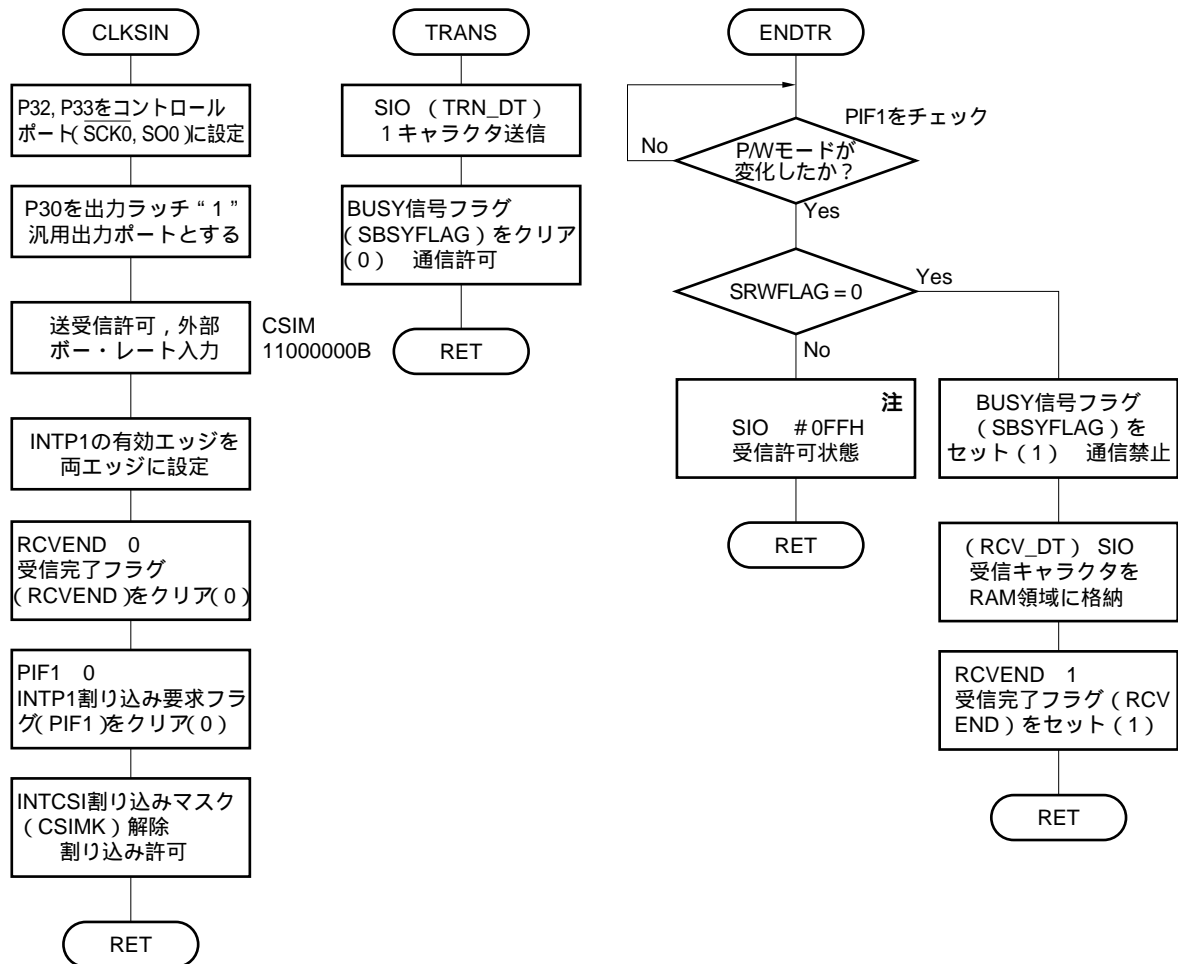
- (a) $\overline{\text{R/W}}$ 端子が変化したかどうかを、INTP1割り込み要求フラグ (PIF1) のポーリングで判別します。
- (b) 通信方向フラグ (SRWFLAG) の状態を判別します。
SRWFLAG = 0 (スレーブ マスタ) ... (d) の処理へ進む。
SRWFLAG = 1 (マスタ スレーブ) ... (c) の処理へ進む。
- (c) シフト・レジスタ (SIO) にダミー・データを書き込み、受信許可状態にして処理を終了します。
- (d) BUSY信号フラグをセット (1) し、通信を禁止します。
- (e) 受信データを格納するRAM領域 (RCV_DT) に書き込みます。
- (f) 受信完了判別フラグ (RVEDFLAG) をセット (1) して処理を終了します。

(7) フロー・チャート

【マスタ側のフロー・チャート】



【スレーブ側のフロー・チャート】



注 送受信動作は, CRXEビットを“0”から“1”へ変化させるか, SIOへの書き込みを行うと起動します。ここでは, SIOヘダミー・データの0FFHを書き込むことによって受信動作許可状態になります(0FFH=ハイ・インピーダンス状態)。

(8) プログラム・リスト

【マスタ側のプログラム・リスト】

アプリケーション・ルーチン時に使用するレーベルの説明

RCV_DT : 受信データRAM領域
TRN_DT : 送信データRAM領域
TNEDFLAG : 送信完了判別フラグ
 TNEDFLAG = 0...送信未完了
 TNEDFLAG = 1...送信完了
RVEDFLAG : 受信完了判別フラグ
 RVEDFLAG = 0...受信未完了
 RVEDFLAG = 1...受信完了
MBSYFLAG : BUSY信号フラグ
 MBSYFLAG = 0...BUSY解除状態
 MBSYFLAG = 1...BUSY状態
MRWFLAG : 通信方向フラグ
 MRWFLAG = 0...スレーブ マスタ
 MRWFLAG = 1...マスタ スレーブ

メイン・ルーチンのプログラム・リスト記述例

```

MAIN$  CSEG
RST:
      LOCATION      0H          ; specify LOCATION address
      .
      .
      MOV    RCV_DT,#0          ;
      CALL  !CLKMIN             ; << CLKMIN >>
      MOV    STBC,#00H         ; system clock = fxx/2

      .
      .
      . *****
      . ***  trans start  ***
      . *****
      .
      .
      MOV    TRN_DT,A          ;
      CALL  !TRANS             ; << TRANS >>
      .
      .
      . *****
      . ***  RECIVE DATA  ***
      . *****
      .
      FLAGCHK:
          BF    RVEDFLAG,$RCVCHK ;
          .
          CLR1  RVEDFLAG         ;
          MOV   A,RCV_DT         ;
          .
          .
      . *****
      . ***  RECIVE START  ***
      . *****
      .
      RCVCHK:
          BF    TNEDFLAG,$CHKEND ; end of transmit ?
          CLR1  TNEDFLAG         ;
          CALL  !RECIVE          ; << RECIVE >>
      .
      .
      . *****
      . ***  INTCSI  ***
      . *****
      .
      INTCSI:
          SEL   RB1              ;
          CALL  !ENDTR           ; << ENDTR >>
          RETI                   ;

```


本アプリケーション・ルーチンのプログラム・リスト

```

*****
;
;   clocked serial I/O interface
;
*****
;
;
PUBLIC   CLKMIN,TRANS,RECIVE,ENDTR
EXTRN   TRN_DT,RCV_DT
EXTRN   PRMOD,CR30D
EXTBIT  TNEDFLAG,RVEDFLAG
EXTBIT  MBSYFLAG,MRWFLAG

;   *** initialize of clocked serial I/O ***
;
;
CSEG
CLKMIN:
CLR1    P32                ; P32 initialize
CLR1    P33                ; P33 initialize
OR      PMC3,#00001100B    ; P32 -> SCK0 , P33 -> S00
SET1    MRWFLAG           ; default=write mode
CLR1    PM6.6             ; P66 -> R/W
MOV     CSIM,#10000001B    ; CSIM initialize ( RCV:bad , TRN:ok )
OR      PRMO,#11110000B    ;
AND     PRMO,#LOW(PRMOD)   ; initialize PRMO for TIMER3
MOVW    CR30W,#CR30D      ; initialize CR30
CLR1    TMC0.4            ; TIMER3=8 bit mode select
SET1    TMC0.7            ; internal serial clock start
AND     INTMO,#11001111B   ; INTP1 rise edge enable
;
CLR1    RVEDFLAG          ; clear receive end flag
;
CLR1    PIF1              ; clear INTP1 flag
CLR1    CSIMK             ; INTCSI enable
EI                      ; interrupt enable

RET

;
;   *** trans process ***
;
TRANS:
BT      MBSYFLAG,$TRANS   ; check busy
SET1    CTXE              ; transmit enable
MOV     A,TRN_DT
MOV     SIO,A             ; transmit data
RET

;
;   *** receive data ***
;
RECIVE:
SET1    CRXE              ; receive enable
RET

;
;   *** transmit/receive complete ***
;

```

```

ENDTR:
    NOT1    MRWFLAG                ; change read/write mode
    BT      MRWFLAG,$ETR_J3        ; check R/W mode
ETR_J1:
    BTCLR   PIF1,$ETR_J2          ; slave receive check
    BR      ETR_J1
ETR_J2:
    CLR1    CTXE                  ; transmit disable
    SET1    TNEDFLAG              ; set transmit complete flag
    RET
ETR_J3:
    CLR1    CRXE                  ; receive disable
    MOV     A,SIO                  ; read data
    MOV     RCV_DT,A              ;
    SET1    RVEDFLAG              ; set receive end flag
    RET
;
    END

```

【スレーブ側のプログラム・リスト】

アプリケーション・ルーチン時に使用するレーベルの説明

RCV_DT : 受信データRAM領域
 TRN_DT : 送信データRAM領域
 RVEDFLAG : 受信完了判別フラグ
 RVEDFLAG = 0...受信未完了
 RVEDFLAG = 1...受信完了
 SBSYFLAG : BUSY信号フラグ
 SBSYFLAG = 0...BUSY解除状態
 SBSYFLAG = 1...BUSY状態
 SRWFLAG : 通信方向フラグ
 SRWFLAG = 0...スレーブ マスタ
 SRWFLAG = 1...マスタ スレーブ

メイン・ルーチンのプログラム・リスト記述例

```

        .
        .
;***** define vector table *****
MAINS   CSEG
RST:
        LOCATION      0H                ; specify LOCATION address
        .
        .
        MOV           RCV_DT,#0          ;
        CALL          !CLKSIN            ; << CLKSIN >>
        CLR1          SBSYFLAG          ; busy clear
        EI
;*****
;***      MAIN      ***
;*****
SIL_MAIN:
        BF           RVEDFLAG,$SIL_MAIN ; data receive ?
        CLR1          RVEDFLAG          ;
        MOV           TRN_DT,RCV_DT     ; trans of receive data
        CALL          !TRANS            ; << TRANS >>
        BR           SIL_MAIN          ;
;*****
;***      INTCSI   ***
;*****
INTCSI:
        SEL          RB1                ;
        CALL          !ENDTR            ; << ENDTR >>
        RETI
    
```

本アプリケーション・ルーチンのプログラム・リスト

```

;
; *****
;
;   clocked serial I/O interface
; *****
;
;
;
;   PUBLIC  CLKSIN,TRANS,ENDTR
;   EXTRN  TRN_DT,RCV_DT
;   EXTBIT  RVEDFLAG
;   EXTBIT  SBSYFLAG,SRWFLAG
;
;
;   *** initialize of clocked serial I/O ***
;
;
;   CSEG
CLKSIN:
;   CLR1   PMC3.0                ; P30 -> in/out port mode
;   OR     PMC3,#00001100B       ; P32 -> SCK0 , P33 -> S00
;   SET1   SBSYFLAG              ; initialize SBSYFLAG
;   CLR1   PM3.0                 ; P3.0 -> output port
;   MOV    CSIM,#11000000B       ; CSIM initialize ( RCV:ok , TRN:ok )
;   OR     INTM0,#00110000B      ; INTP1 all edge enable
;
;   CLR1   RVEDFLAG              ; clear receive end flag
;
;   CLR1   PIF1                  ; clear INTP1 flag
;   CLR1   CSIMK                 ; INTCSI enable
;   EI                                ; interrupt enable
;   RET
;
;
;   *** transmit ***
;
;
;   TRANS:
;   MOV    A,TRN_DT
;   MOV    SIO,A                 ; transmit data
;   CLR1   SBSYFLAG              ; busy clear
;   RET
;
;
;   *** transmit/receive complete ***
;
;
;   ENDTR:
;   BTCLR  PIF1,$EDTJ1           ; check R/W edge
;   BR     ENDTR
;
;   EDTJ1:
;   BF     SRWFLAG,$EDTJ2        ; check R/W mode
;   MOV    SIO,#OFFH             ; data receive start
;   RET
;
;   EDTJ2:
;   SET1   SBSYFLAG              ; set busy
;   MOV    A,SIO                 ; receive data
;   MOV    RCV_DT,A
;   SET1   RVEDFLAG              ; set receive end flag
;   RET
;
;
;   END

```

3.4.2 I²Cバス・モードのプログラム例 (μPD784038Yサブシリーズのみ)

ここではμPD784036Yを代表製品として説明します。

I²Cバス・モードは、シリアル・クロック (SCL) と、シリアル・データ・バス (SDA) の2本のラインにより複数のデバイス間で8ビット・データを転送するシリアル・インタフェースです。

従来のシリアルI/O方式では、データ転送機能しか有していないために、複数のデバイスを接続してシリアル・バスを構成した際に、チップ・セレクト信号やコマンド/データの区別、ビジィ状態の判断などのため多くのポートや配線が必要となりますが、I²Cバス・モードでは、それらの配線を必要とせずに複数のデバイスとの通信が可能となり、マイコンのポートの削減や基板内の配線の引き回しの減少に有効です。

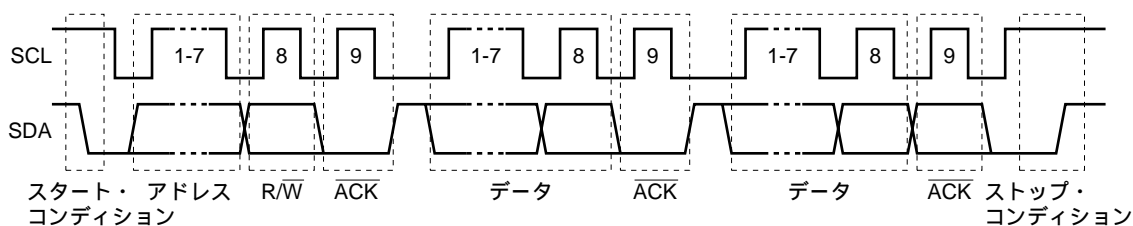
まず(1) I²Cバスの定義および制御方法でI²Cバスの制御について簡単に説明し、(2) 処理概要以降に、プログラム例を示します。

(1) I²Cバスの定義および制御方法

I²Cバスのシリアル・データ通信フォーマットおよび、使用する信号の意味について次に説明します。

I²Cバスのシリアル・データ・バス上に出力されている“スタート・コンディション”、“データ”、および“ストップ・コンディション”の各転送タイミングを図3-14に示します。

図3-14 I²Cバスのシリアル・データ転送タイミング



スタート・コンディション、スレーブ・アドレス、ストップ・コンディションはマスタが出力します。

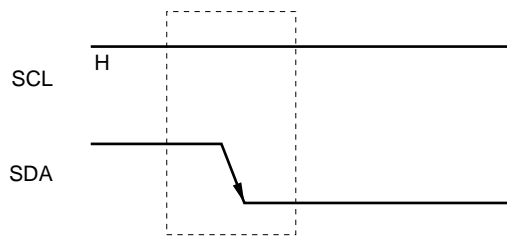
アクノリッジ信号 ($\overline{\text{ACK}}$) は、マスタ、スレーブのどちらでも出力できます (通常、8ビット・データの受信側が出力します)。

シリアル・クロック (SCL) は、マスタが出力し続けます。ただし、スレーブはSCLのロウ・レベル期間を延長し、ウエイトを挿入することができます。

(a) スタート・コンディション

SCL端子がハイ・レベルのときに、SDA端子がハイ・レベルからロウ・レベルに変化するとスタート・コンディションとなります。SCL端子、SDA端子のスタート・コンディションはマスタがスレーブに対してシリアル転送を開始するときに出る信号です。

図3 - 15 スタート・コンディション



スタート・コンディションは、SCLがハイ・レベルのときにSTTをセット（1）することにより出力されます。また、スタート・コンディションを検出すると、STDがセット（1）されます。

また、STTをセット（1）後、必ずNOPなどで1命令以上実行してからSIOにアドレスを書き込んでください。

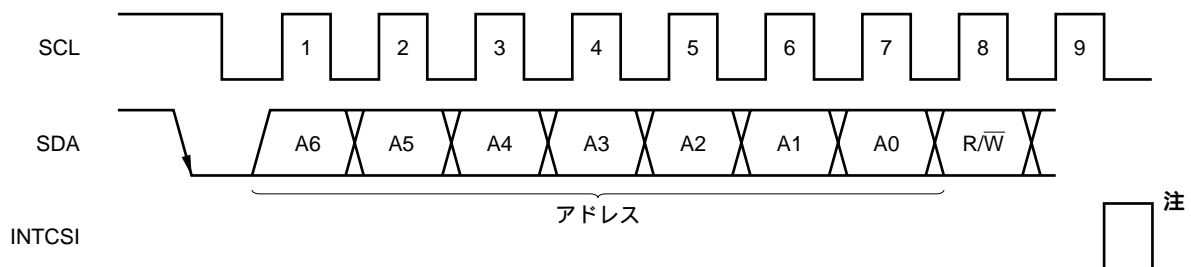
(b) アドレス

スタート・コンディションに続く7ビット・データはアドレスと定義されています。

アドレスは、マスタがバス・ラインに接続されている複数のスレーブの中から、特定のスレーブを選択するために出力する7ビット・データです。したがって、バス・ライン上のスレーブは、すべて異なるアドレスにしておく必要があります。

スレーブは、ハードウェアによってこの条件を検出し、さらに、7ビット・データがスレーブ・アドレス・レジスタ（SVA）と一致しているかを調べます。このとき、7ビット・データとSVAの値が一致すると、そのスレーブが選択されたことになり、以後、マスタがスタート・コンディションまたはストップ・コンディションを送信するまでマスタとの通信を行います。

図3 - 16 アドレス



注 スレーブ動作でWUP = 1 のときに自アドレス以外を受信した場合は、INTCSIは発生しません。

アドレスの出力は、スレーブのアドレスと(c) 転送方向指定に説明する転送方向をSIOへ同時に書き込むことで行います。また、受信したアドレスは、転送方向の指定とともにSIOに書き込まれます。

なお、スレーブのアドレスは、SIOの上位7ビットに割り当てられます。

SIOへのアドレスの書き込みは、STTをセット（1）後、必ずNOPなどで1命令以上実行してから行ってください。

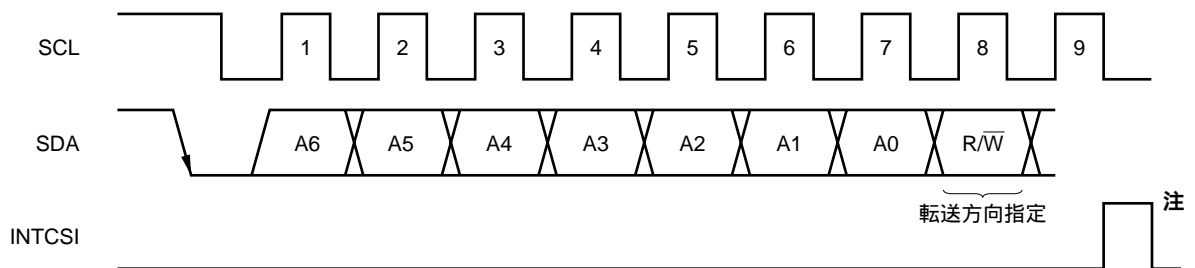
（c）転送方向指定

マスタは、7ビットのアドレスに続いて転送方向を指定するため、1ビット・データを送信します。

この転送方向指定ビットが0のとき、マスタがスレーブにデータを送信することを示します。

また、転送方向指定ビットが1のとき、マスタがスレーブからデータを受信することを示します。

図3 - 17 転送方向指定



注 スレーブ動作でWUP = 1のときに自アドレス以外を受信した場合は、INTCSIは発生しません。

転送方向の指定ビットは、アドレスと同時にSIOに書き込むことで出力されます。また、受信した転送方向は、アドレスとともにSIOに書き込まれると同時に、スレーブ・アドレス・レジスタ（SVA）のTREビット（ビット0）にも書き込まれます。

なお、SIOでは、転送方向は最下位ビットに割り当てられます。

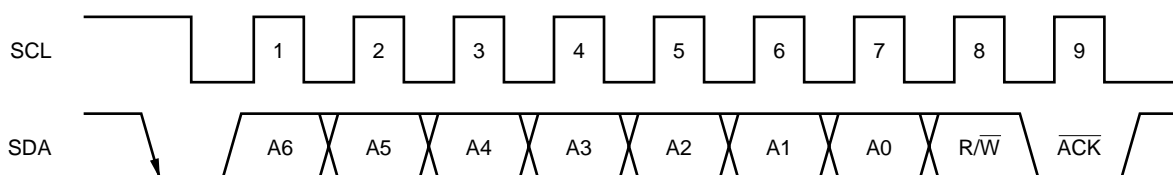
SIOへの転送方向の書き込みは、STTをセット（1）後、必ずNOPなどで1命令以上実行してから行ってください。

(d) アクノリッジ信号 ($\overline{\text{ACK}}$)

アクノリッジ信号は、送信側と受信側における、シリアル・データ受信の確認のための信号です。

受信側は、8ビット・データを受信するごとにアクノリッジ信号を返します。ただし、マスタ受信時にスタート・コンディションまたはストップ・コンディションを発行する場合、最後の受信データ時にACKを返さないでください。送信側は、8ビット送信後、受信側からアクノリッジ信号が返されたか検出を行います。アクノリッジ信号が返されたとき、受信が正しく行われたものとして次の処理を行います。また、スレーブからアクノリッジ信号が返らない場合は受信が正しく行われないので、マスタはストップ・コンディションを出力し、送信を中止します。

図3 - 18 アクノリッジ信号



備考 8クロック・ウェイト選択時：ウェイト解除を行う前に $\text{ACKE} = 1$ とすることにより、SCLの8クロック目の立ち下がりに同期してアクノリッジ信号を出力します。

9クロック・ウェイト選択時：あらかじめ $\text{ACKE} = 1$ とすることにより、アクノリッジ信号をSCLの8クロック目の立ち下がりに同期してアクノリッジ信号を自動出力します。

アクノリッジ信号の出力は、 ACKE をセット(1)することにより、SCLの8クロック目の立ち下がりに同期して出力します。ただし、 $\text{WUP} = 1$ の場合、自アドレス受信時は、 ACKE の値にかかわらずSCLの8クロック目の立ち下がりに同期してアクノリッジの自動出力を行い、自アドレス以外受信時は、アクノリッジ信号の出力を行いません。

アクノリッジ信号の出力方法は、ウェイト・タイミングの設定により次のようになります。

- ・ 8クロック・ウェイト選択時：ウェイト制御を行う前に $\text{ACKE} = 1$ とすることにより、SCLの8クロック目の立ち下がりに同期してアクノリッジ信号を出力します。
- ・ 9クロック・ウェイト選択時：あらかじめ $\text{ACKE} = 1$ とすることにより、アクノリッジ信号をSCLの8クロック目の立ち下がりに同期してアクノリッジ信号を自動出力します。

ただし、WUP = 1 でアドレス受信時は次の動作となります。

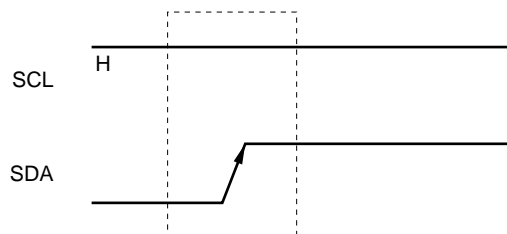
- ・自アドレス受信時 : アクノリッジ信号の自動出力を行う。
- ・自アドレス以外受信時 : アクノリッジ信号の自動出力は行わない。

(e) ストップ・コンディション

SCL端子がハイ・レベルのときに、SDA端子がロウ・レベルからハイ・レベルに変化すると、ストップ・コンディションとなります。

ストップ・コンディションは、マスタがスレーブに対してシリアル転送が終了したときに出力する信号です。

図3 - 19 ストップ・コンディション



ストップ・コンディションは、SPTをセット(1)することで発生します。

また、ストップ・コンディションを検出すると、SPDがセット(1)され、INTSPCが発生します。

(f) ウェイト信号 ($\overline{\text{WAIT}}$)

ウェイト信号は、マスタまたはスレーブがデータの送受信のための準備中（ウェイト状態）であることを相手に知らせるための信号です。

SCL端子をロウ・レベルにすることにより、相手にウェイト状態を知らせます。マスタは、ウェイト状態が解除されると、次の転送を開始することができます。

図3 - 20 ウェイト信号 (1/2)

(i) マスタ、スレーブとも8クロック・ウェイト時

(マスタ：送信，スレーブ：受信， $\text{ACKE} = 0$)

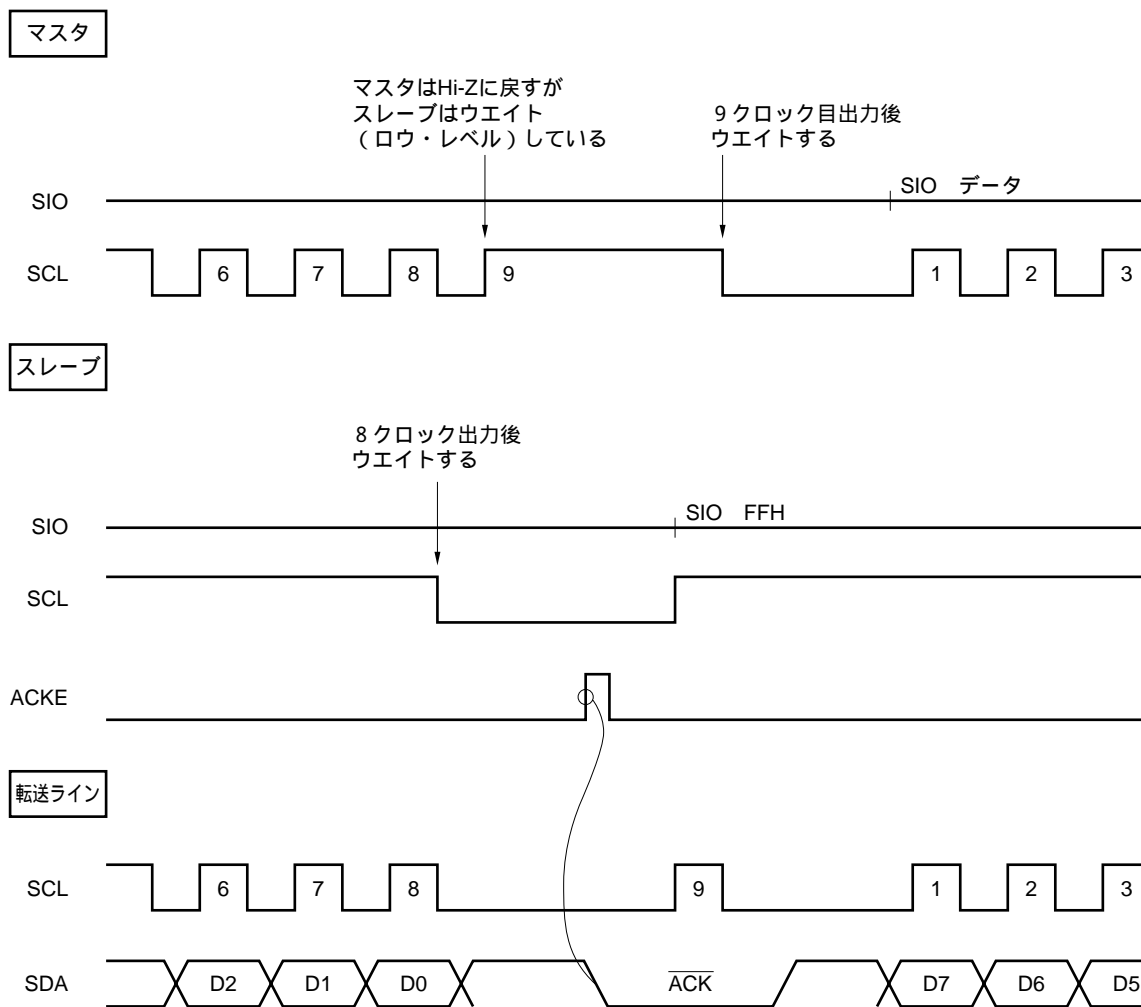
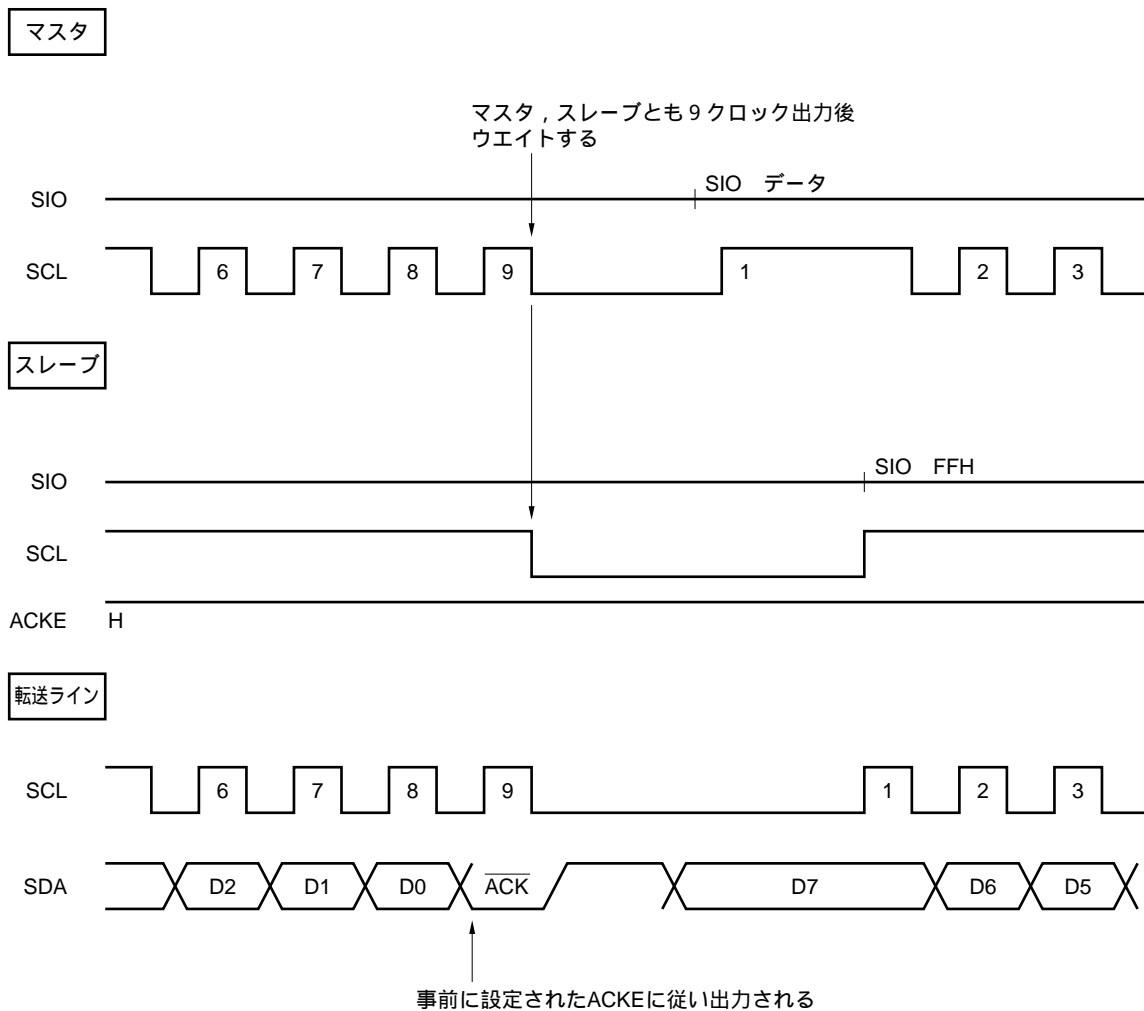


図3 - 20 ウェイト信号 (2/2)

(ii) マスタ, スレーブとも9クロック・ウェイト時
 (マスタ: 送信, スレーブ: 受信, ACKE = 1)



ウェイトの発生は, WUPとWTIMの設定により自動的に発生します。

また, ウェイトの解除は, WREL = 1 にするか, SIOに書き込みを行うか, CTXEをクリア (0)
 すると行われます。

(g) 割り込み要求 (INTCSI) 発生タイミングおよびウェイト制御

クロック同期式シリアル・インタフェース (CSIM) のWUPビットとI²Cバス・コントロール・レジスタ (IICC) のWTIMビットの組み合わせにより、表3 - 8に示すタイミングでINTCSIが発生し、また、ウェイト制御を行います。

表3 - 8 INTCSI発生タイミングおよびウェイト制御

WUP	WTIM	スレーブ動作時			マスタ動作時		
		アドレス	データ受信	データ送信	アドレス	データ受信	データ送信
0	0	8	8	9	9	8	9
0	1	8	9	9	9	9	9
1	0	g注1, 2	g注2	g注2	9	8	9
1	1	g注1, 2	g注2	g注2	9	9	9

注1 . WUP = 1のときのスレーブのINTCSI信号およびウェイトは、スレーブ・アドレス・レジスタ (SVA) に設定しているアドレスと一致したときのみ、9クロック目の立ち下がりで発生します。

また、このとき、ACKEの設定にかかわらず、 \overline{ACK} が出力されます。

2 . WUP = 1のときに、SVAと受信したアドレスが一致していない場合は、INTCSIもウェイトも発生しません。

備考 表中の数字は、シリアル・クロックのクロック数を示しています。また、割り込み要求、ウェイト制御ともにシリアル・クロックの立ち下がりに同期します。

(i) アドレス送受信時

- ・スレーブ動作時 : WTIMビットにかかわらず、WUPビットにより、割り込みおよびウェイト・タイミングが決まります。
- ・マスタ動作時 : WUPビットおよびWTIMビットにかかわらず、割り込みおよびウェイト・タイミングは、9クロック目の立ち下がりで発生します。

(ii) データ受信時

- ・マスタ/スレーブ動作時 : WUPビットにかかわらず、WTIMビットにより、割り込みおよびウェイト・タイミングが決まります。

(iii) データ送信時

- ・ マスタ/スレーブ動作時：WUPビットおよびWTIMビットにかかわらず，割り込みおよびウェイト・タイミングは，9クロック目の立ち下がりで発生します。

(iv) ウェイト解除方法

ウェイトの解除方法には次の3つがあります。

- ・ I²Cバス・コントロール・レジスタ (IICC) のWREL = 1
- ・ シリアル・シフト・レジスタ (SIO) のライト動作
- ・ クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM) のCTXE = 0

8クロック・ウェイト選択 (WTIM = 0) 時は，ウェイト解除前に $\overline{\text{ACK}}$ の出力レベルを決定する必要があります。

(h) 割り込み要求 (INTSPC) 発生タイミング

INTSPCは，ストップ・コンディションを検出すると発生します。

INTSPCの割り込み処理ルーチンで，次にスタート・コンディションが発生するのを待つための処理を行う必要があります。

スレーブとして使用する場合で，WUP = 0として使用するときなどに使用します。

(i) アドレスの一致検出方法

I²Cバス・モードでは，マスタがスレーブ・アドレスを送信することにより，特定のスレーブ・デバイスを選択することができます。

アドレス一致の検出は，ハードウェアで自動的に行えます。ウエイク・アップ機能指定ビット (WUP) = 1の設定でスレーブ・アドレス・レジスタ (SVA) に自アドレスを設定した場合，マスタから送信されたスレーブ・アドレスとSVAに設定したアドレスが一致したときのみ，INTCSI割り込み要求が発生します。

スレーブの受信データがアドレスかどうかの判断は，スタート・コンディション検出ビット (STD) の値を確認することにより行います。

注意 WUP = 0の場合は，スタート・コンディションのあとに受信したデータ (アドレス) とスレーブ・アドレス・レジスタ (SVA) に設定した自アドレスが一致しない場合でも，INTCSI割り込み要求が発生します。

(j) エラーの検出

I²Cバス・モードでは、送信中のシリアル・バス (SDA) の状態が、送信しているデバイスのシリアル・シフト・レジスタ (SIO) にも取り込まれるため、送信開始前と送信終了後のSIOデータを比較することにより、送信エラーを検出することができます。この場合、2つのデータが異なっていれば送信エラーが発生したものと判断します。

(2) 処理概要

(a) 仕様

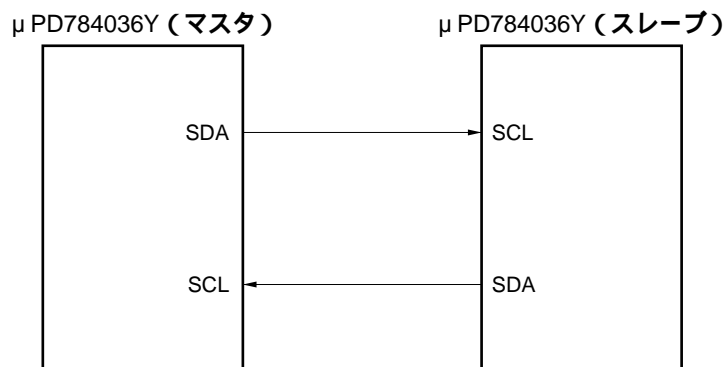
I²Cバス・モードのプログラム例として、μPD784036Yでの通信例を紹介します。

(b) 仕様周辺機説明

μPD784036Yを2つ使用し、それぞれマスタ、スレーブとします。

図3 - 21に、接続図を示します。

図3 - 21 マスタとスレーブの接続図



通信プロトコルの仕様を次に示します。

転送スピードは、390 kHzとします。ただし、 $f_{xx} = 25$ MHzです。また、データの送受信時には8クロック目にウェイトが入るように設定し、スレーブ側は、自アドレス受信時に割り込み要求を発生するウエイク・アップ機能を使用します。プログラム例では、マスタからデータを2回送信したあとに、スレーブからのデータを2回受信する動作を行います。図3 - 22 ~ 図3 - 28に本プログラムのタイミング・チャートを示します。

図3-22 I²Cバス・モードのプログラム例のタイミング・チャート

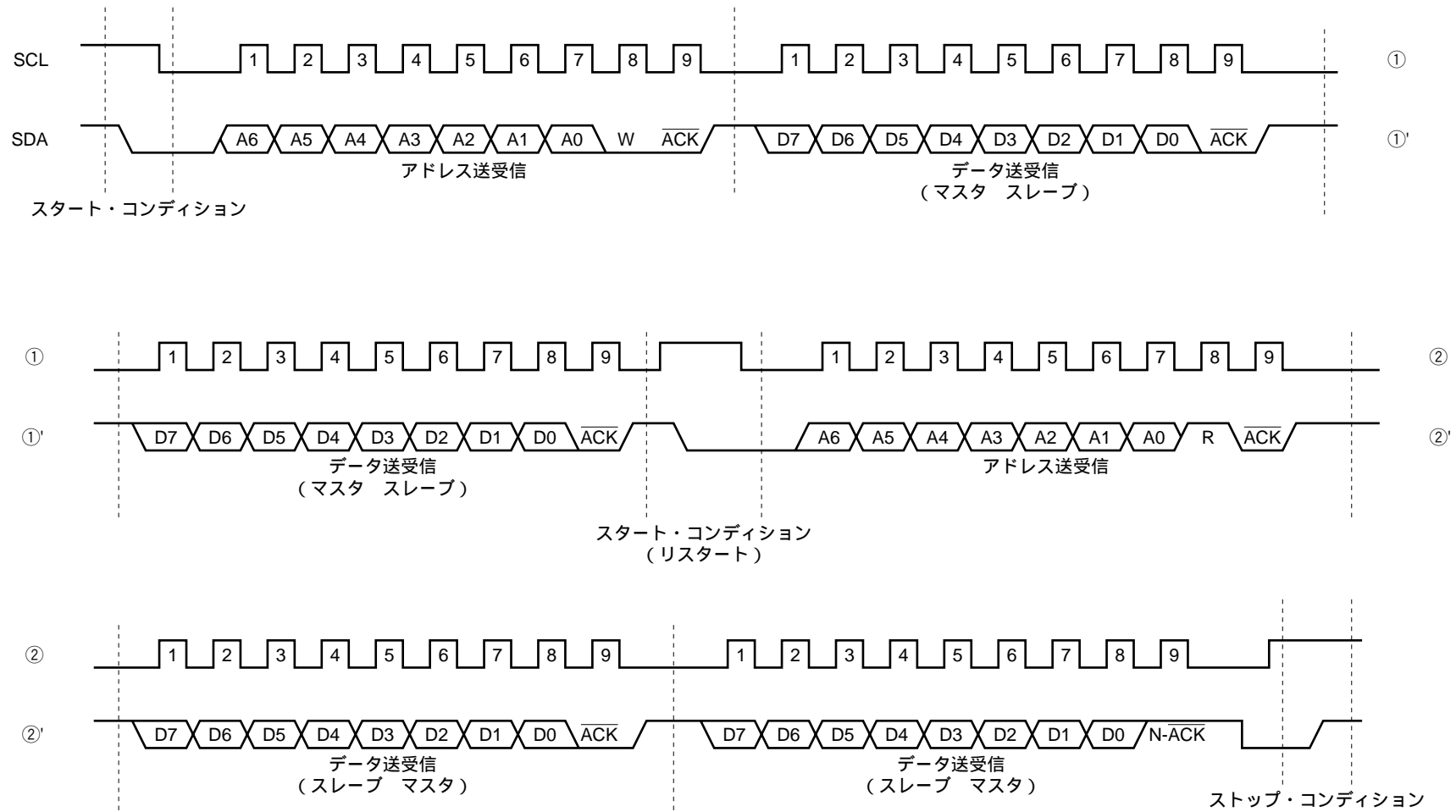


図3 - 23 I²Cバス・モードのプログラム例のタイミング・チャート
 (スタート・コンディション アドレスの送受信)

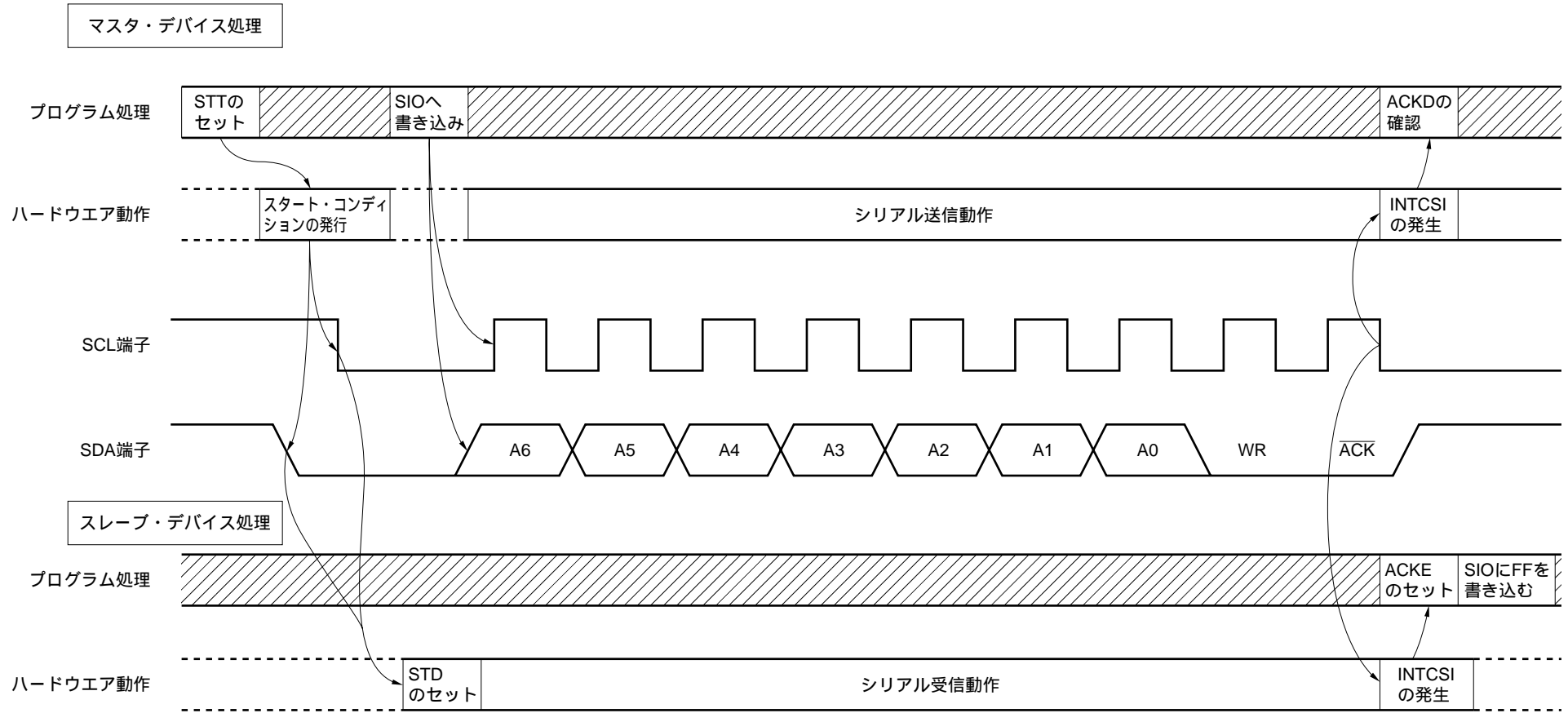


図3 - 24 I²Cバス・モードのプログラム例のタイミング・チャート
 (データの送受信: マスタ スレーブ)

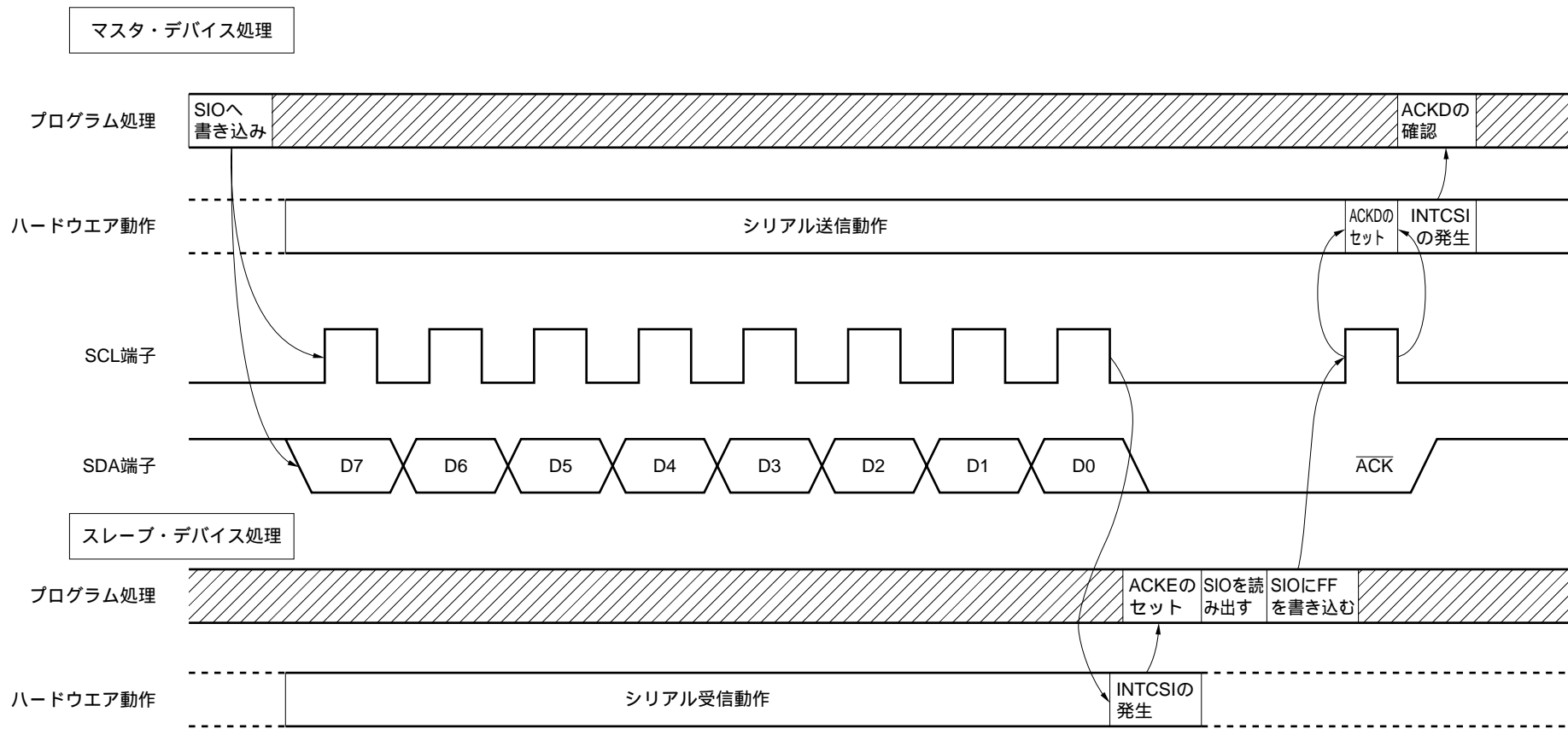


図3 - 25 I²Cバス・モードのプログラム例のタイミング・チャート
 (データの送受信 スタート・コンディション)

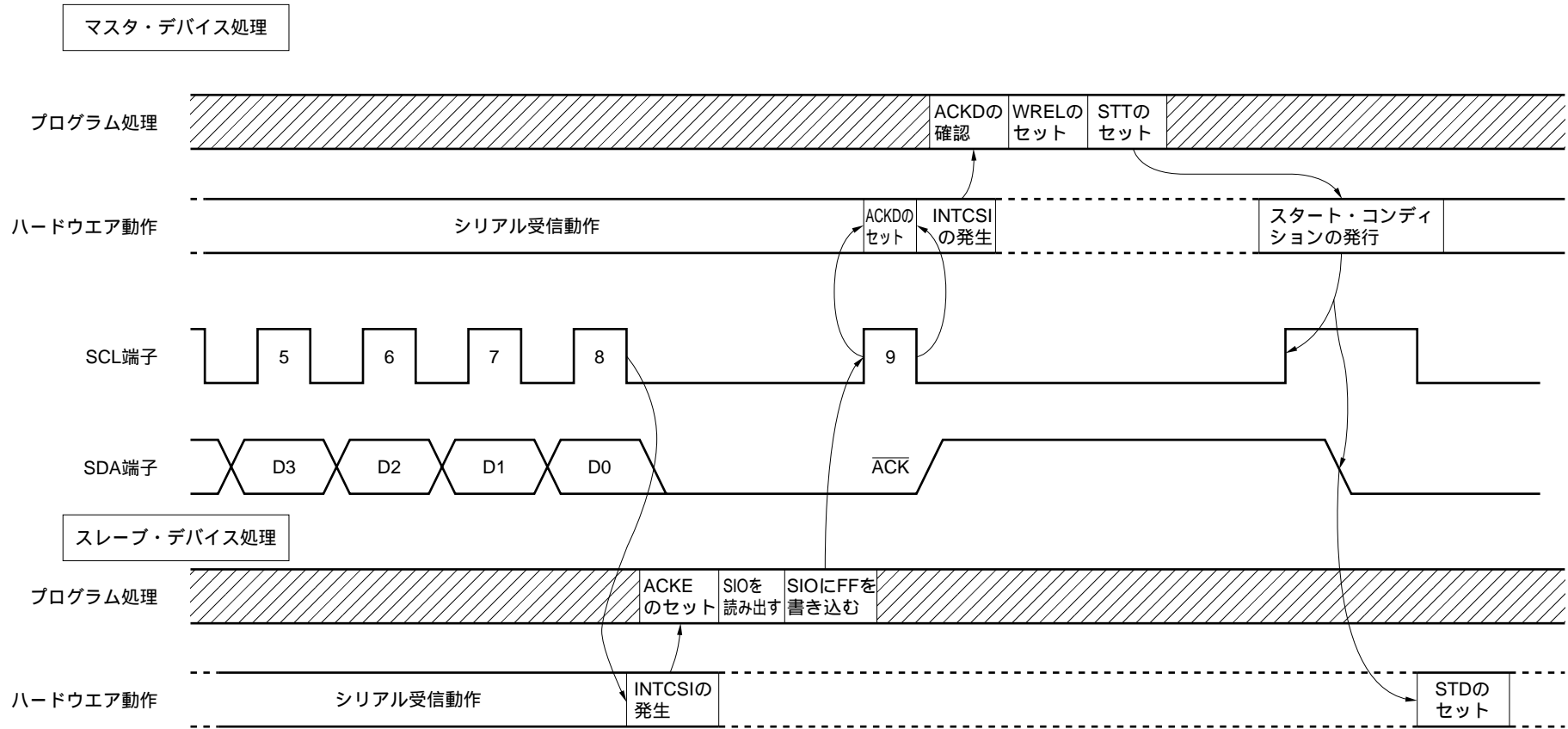


図3 - 26 I²Cバス・モードのプログラム例のタイミング・チャート
 (アドレスの送受信 : マスタ受信時)

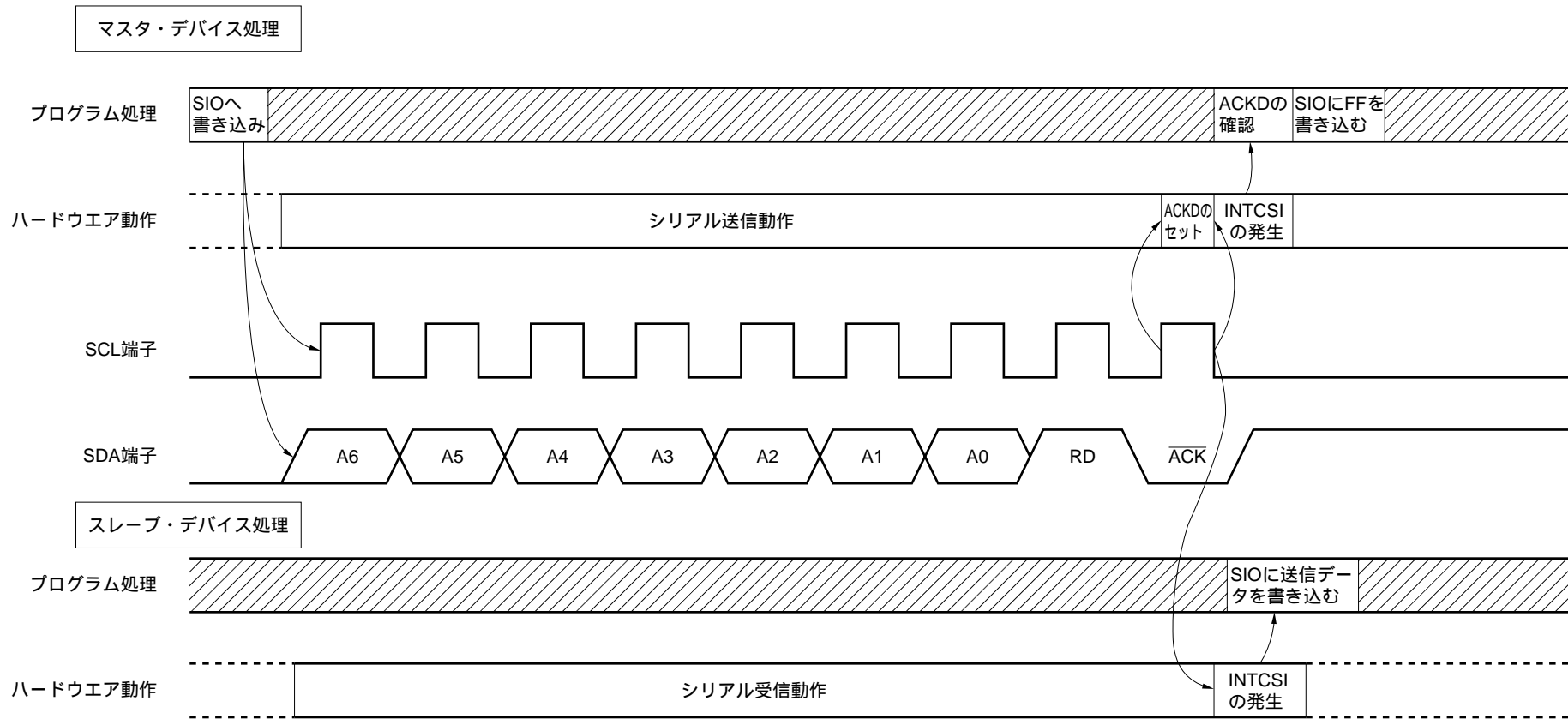


図3 - 27 I²Cバス・モードのプログラム例のタイミング・チャート
 (データの送受信 : スレーブ マスタ)

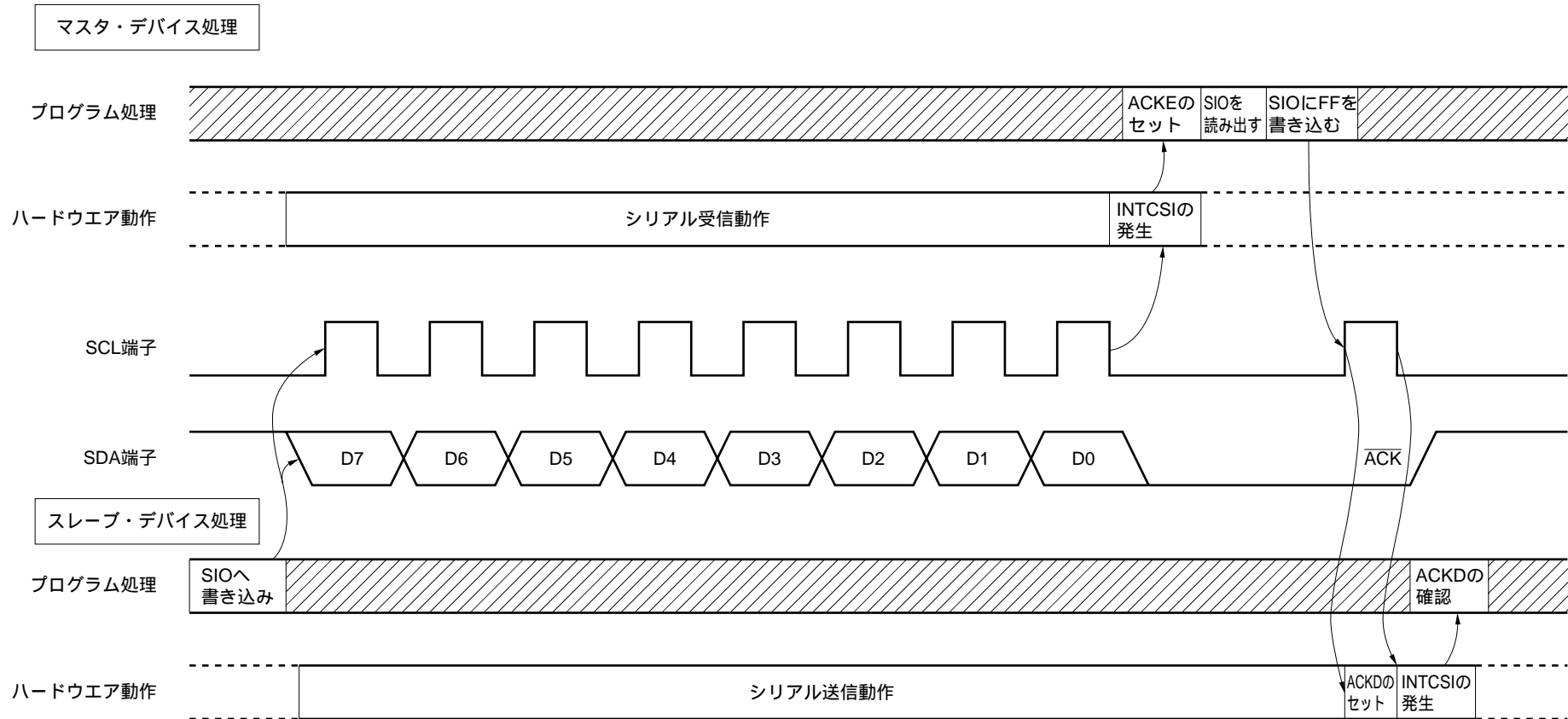
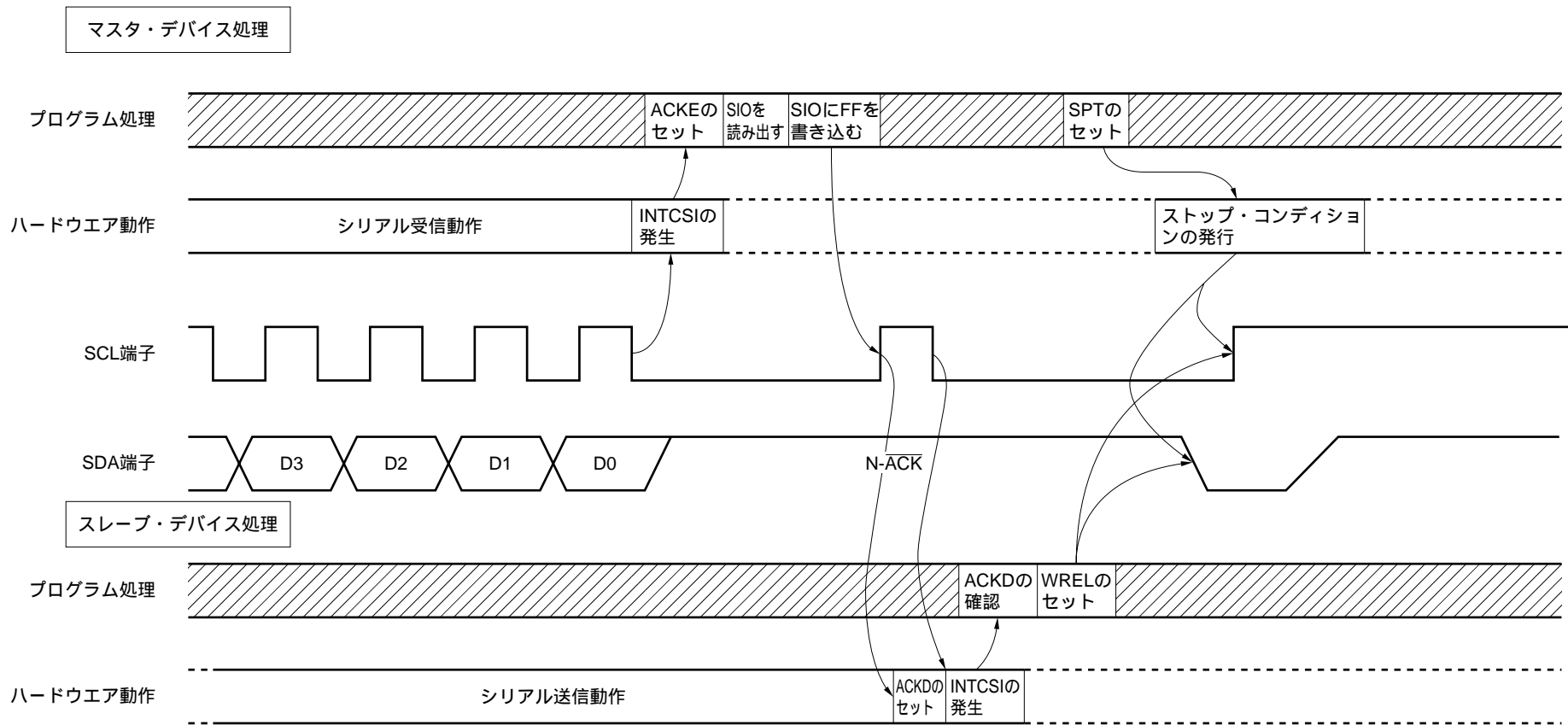


図3-28 I²Cバス・モードのプログラム例のタイミング・チャート
 (データの送受信 ストップ・コンディション)



(3) 使用RAM領域

本プログラムで、マスタは表3 - 9に示すように、スレーブは表3 - 10に示すようにRAM領域を使用します。

【マスタ側の使用RAM領域】

表3 - 9 I²Cバスのプログラム例(マスタ側)で使用するRAM/フラグ

RAM/フラグ名称	用 途	バイト数/ ビット数	初期値
M_SHORI	送受信の処理カウンタのためのワーク領域	1 バイト	0
ADDRESS	マスタが通信しようとするスレーブのアドレスを格納するためのワーク領域	1 バイト	0
DATA_T	I ² Cバスで送信するためのデータを格納するワーク領域	1 バイト	0
DATA_R	I ² Cバスで受信したデータを格納するワーク領域	1 バイト	0
_RECEIVE	受信動作中フラグ _RECEIVE = 0 ...受信未動作 _RECEIVE = 1 ...受信動作中	1 ビット	0
_TRANSFER	送信動作中フラグ _TRANSFER = 0 ...送信未動作 _TRANSFER = 1 ...送信動作中	1 ビット	0
LAST_DATA	最終データ・フラグ LAST_DATA = 0 ...未最終データ LAST_DATA = 1 ...最終データ	1 ビット	0

【スレーブ側の使用RAM領域】

表3 - 10 I²Cバスのプログラム例(スレーブ側)で使用するRAM/フラグ

RAM/フラグ名称	用 途	バイト数/ ビット数	初期値
DATA_T	I ² Cバスで送信するためのデータを格納するワーク領域	1 バイト	0
DATA_R	I ² Cバスで受信したデータを格納するワーク領域	1 バイト	0
TRANCE_COUNTER	送信回数のカウント・データを格納するワーク領域	1 バイト	0
_TRANSFER	送信動作中フラグ _TRANSFER = 0 ...送信未動作 _TRANSFER = 1 ...送信動作中	1 ビット	0
Rset_Treset	送受信動作確認フラグ Rset_Treset = 0 ...送信動作 Rset_Treset = 1 ...受信動作	1 ビット	0

(4) 使用レジスタ

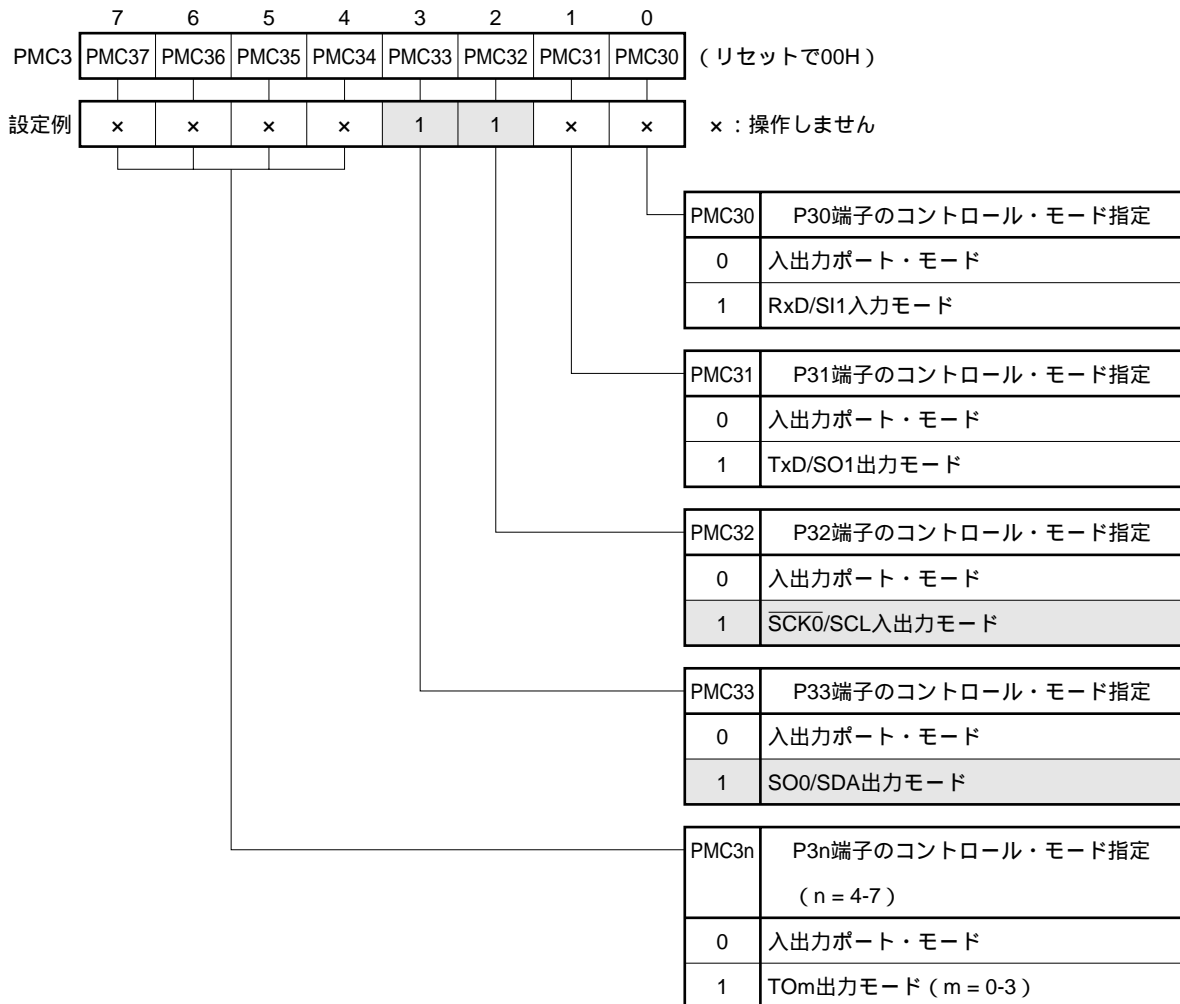
【マスタ側の使用レジスタ】

(a) 汎用レジスタ

- ・イニシャライズ処理 (レーベル名称 : INIT_I2C)
なし
- ・I²C割り込み処理 (レーベル名称 : I2C_INT)
なし
- ・送受信スタート処理 (レーベル名称 : START_TASK)
Aレジスタ
- ・送信処理 (レーベル名称 : TRANS_TASK)
Aレジスタ
- ・受信処理 (レーベル名称 : RECEIVE_TASK)
Aレジスタ
- ・ACK確認処理 (レーベル名称 : ACK_TASK)
なし

(b) 特殊機能レジスタ

ポート3モード・コントロール・レジスタ (PMC3)



クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM)



I²Cバス・コントロール・レジスタ (IICC) (1/2)

	7	6	5	4	3	2	1	0	
IICC	WTIM	WREL	ACKD	ACKE	STD	SPD	STT	SPT	(リセットで00H)
設定例	0	0	0	0	0	0	0	0	

WTIM	0	8クロック・ウェイト。SCLの8クロック目の立ち下がりでの割り込み要求 (INTCSI) を発生する。 <ul style="list-style-type: none"> ・マスタの場合 : 8クロック出力後, SCL出力をロウ・レベルにしてウェイトする。 ・スレーブの場合 : 8クロック入力後, SCL端子をロウ・レベルにしてウェイト要求をする。
	1	9クロック・ウェイト。SCLの9クロック目の立ち下がりでの割り込み要求 (INTCSI) を発生する。 <ul style="list-style-type: none"> ・マスタの場合 : 9クロック出力後, SCL出力をロウ・レベルにしてウェイトする。 ・スレーブの場合 : 9クロック入力後, SCL端子をロウ・レベルにしてウェイト要求をする。

WREL	WREL = 1 により, ウェイト状態を解除 (SCLをハイ・レベル) する。 ウェイト状態を解除後, WRELビットは自動的にクリア (0) される。
------	--

ACKD	クリアされる条件 (ACKD = 0)	セットされる条件 (ACKD = 1)
	アクノリッジ信号 (\overline{ACK}) 検出後のウェイト解除 (WREL = 1 または SIOライトまたは SPT = 1) 時 CTXE = 0 リセット入力時	アクノリッジ信号 (\overline{ACK}) 検出時 (9発目のSCLの立ち上がりで, SDAがロウ・レベルのとき)

ACKE	0	アクノリッジ信号の自動出力を禁止。 送信時, または 8クロック・ウェイト選択時に使用する。ただし, WUP = 1 でアドレス受信時は次の動作となる。 <ul style="list-style-type: none"> ・自アドレス受信時 : ACEK = 0 の場合でも SCL の 9クロック目の立ち下がりに同期してアクノリッジ信号の自動出力を行う。 ・自アドレス以外受信時 : アクノリッジ信号の自動出力は行わない。
	1	8クロック・ウェイト選択時 : ウェイト制御を行う前に ACEK = 1 とすることにより, SCL の 8クロック目の立ち下がりに同期してアクノリッジ信号を出力する。 9クロック・ウェイト選択時 : あらかじめ ACEK = 1 とすることにより, アクノリッジ信号を SCL の 8クロック目の立ち下がりに同期してアクノリッジ信号を自動出力する。 ただし, WUP = 1 でアドレス受信時は次の動作となる。 <ul style="list-style-type: none"> ・自アドレス受信時 : アクノリッジ信号の自動出力を行う。 ・自アドレス以外受信時 : アクノリッジ信号の自動出力は行わない。

I²Cバス・コントロール・レジスタ (IICC) (2/2)

	7	6	5	4	3	2	1	0	
IICC	WTIM	WREL	ACKD	ACKE	STD	SPD	STT	SPT	(リセットで00H)
設定例	0	0	0	0	0	0	0	0	

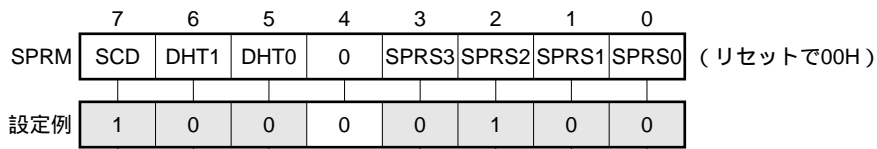
STD	クリアされる条件 (STD = 0)	セットされる条件 (STD = 1)
	スタート・コンディション検出後のウエイト解除 (WREL = 1 または SIO ライト ^注 または SPT = 1) 時 CTXE = 0 リセット入力時 注 マスタ時のアドレス・ライトは除く	・ WUP = 0 のとき：スタート・コンディション検出時 ・ WUP = 1 のとき：自アドレス検出時

SPD	クリアされる条件 (SPD = 0)	セットされる条件 (SPD = 1)
	スタート・コンディション検出時 CTXE = 0 リセット入力時	ストップ・コンディション検出時

STT	SCLおよびSDAがハイ・レベルの状態 ^注 でSTT = 1 にすることにより、SOラッチをクリア (0) する。SOラッチをクリア (0) 後、SCLをロウ・レベルにする。SCLをロウ・レベルにしたあと、STTビットは自動的にクリア (0) される。 注 SCLのレベル状態は、P32/SCL端子を入力モード (PM32 = 1) とし、SCL*を読み込むことにより確認できます。また、SDAのレベル状態も、P33/SDA端子を入力モード (PM33 = 1) とし、SDA*を読み込むことにより確認できます。 * SCLおよびSDAは、NEC製アセンブラでは予約語に、Cコンパイラでは #pragma sfr 指令で、sfr変数として定義されています。 注意 1 . SCLおよびSDAがロウ・レベルの状態でSTT = 1 にしても、スタート・コンディションは出力されません (STT = 1 にしたあと、SCLがハイ・レベルとなってもスタート・コンディションは出力されません)。 2 . STTをセット (1) 後、必ずNOPなどで1命令以上実行してからSIOにアドレスを書き込んでください。
-----	--

SPT	SPT = 1 により、SOラッチがクリア (0) され、SOラッチがクリア (0) されたあと、SCLをハイ・レベルにする。SCLがハイ・レベルになったあと、SOラッチをセット (1) する。SOラッチをセット (1) 後、SPTビットは自動的にクリア (0) される。
-----	--

シリアル・クロック用プリスケアラ・モード・レジスタ (SPRM)

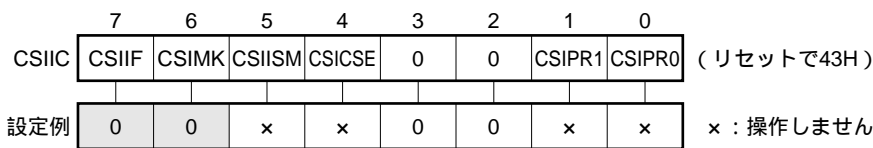


SPRS3	SPRS2	SPRS1	SPRS0	シリアル・クロックの指定
0	0	0	0	$f_{xx}/16$
0	0	0	1	$f_{xx}/24$
0	0	1	0	$f_{xx}/32$
0	0	1	1	$f_{xx}/48$
0	1	0	0	$f_{xx}/64$
0	1	0	1	$f_{xx}/96$
0	1	1	0	$f_{xx}/128$
0	1	1	1	$f_{xx}/192$
1	0	0	0	$f_{xx}/256$
1	0	0	1	$f_{xx}/384$

DHT1	DHT0	データ・ホールド時間指定	
0	0	16 MHz < f_{xx} 32 MHz	$10/f_{xx} \sim 12/f_{xx}$
0	1	8 MHz < f_{xx} 16 MHz	$5/f_{xx} \sim 7/f_{xx}$
1	x	4 MHz < f_{xx} 8 MHz	$3/f_{xx} \sim 5/f_{xx}$

SCD	シリアル・クロック・デューティ指定
0	標準モード用
1	高速モード用

割り込み制御レジスタ (CSIIC)



CSIPR1 (ビット1)	CSIPR0 (ビット0)	割り込み要求の優先順位 指定
0	0	優先順位 0 (最優先順位)
0	1	優先順位 1
1	0	優先順位 2
1	1	優先順位 3

CSICSE (ビット4)	コンテキスト・スイッチング処理の 指定
0	ベクタ割り込みで処理
1	コンテキスト・スイッチングで処理

CSIISM (ビット5)	割り込み処理形態の指定
0	ベクタ割り込み処理 / コンテキスト・スイッチング処理
1	マクロ・サービス処理

CSIMK (ビット6)	割り込み処理の許可 / 禁止
0	割り込み処理を許可
1	割り込み処理を禁止

CSIIF (ビット7)	割り込み要求発生の有無
0	割り込み要求なし (割り込み信号が発生していない)
1	割り込み要求状態 (割り込み信号が発生)

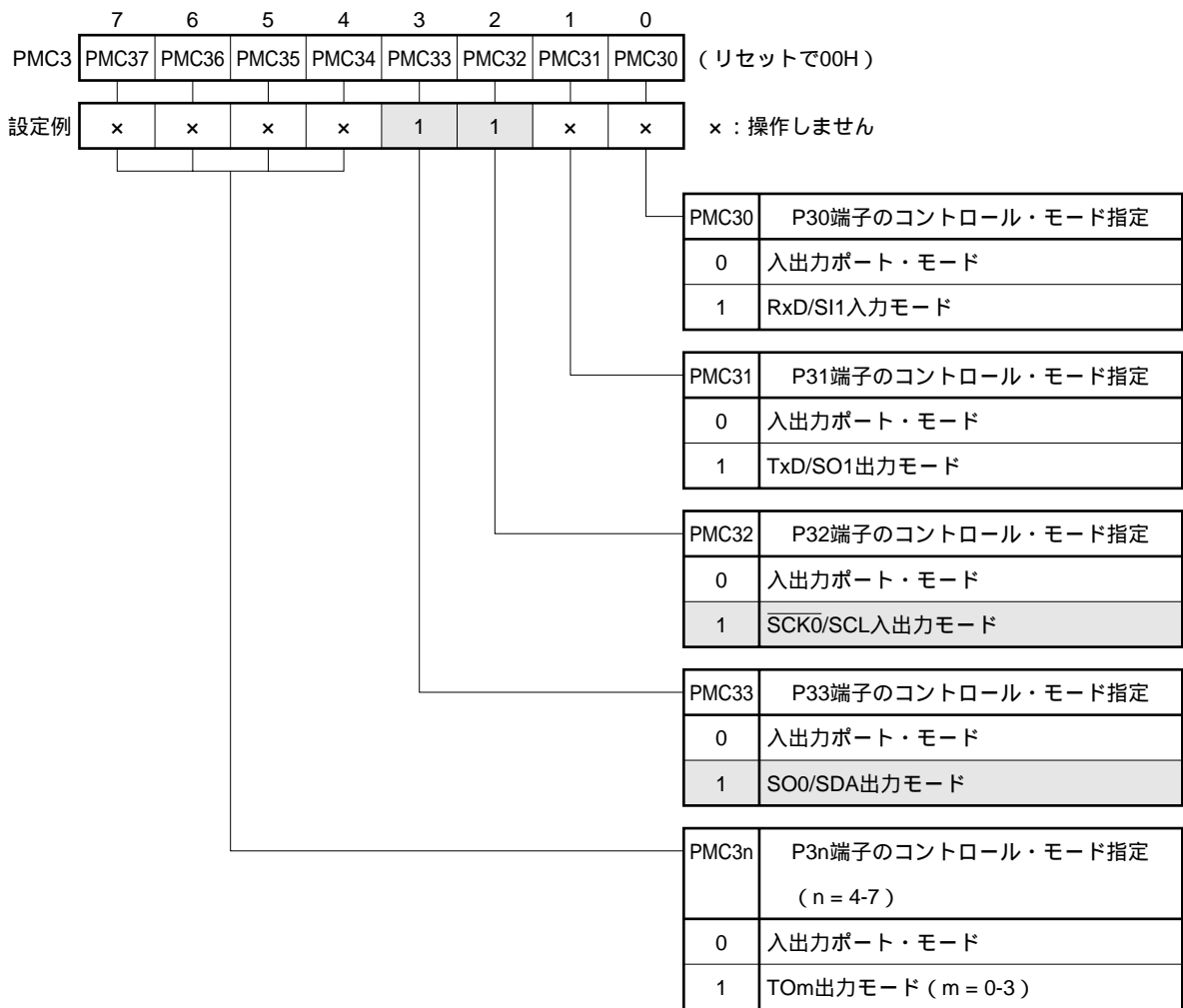
【スレーブ側の使用レジスタ】

(a) 汎用レジスタ

- ・イニシャライズ処理 (レーベル名称 : INIT_I2C)
なし
- ・I²C割り込み処理 (レーベル名称 : I2C_INT)
なし
- ・送受信スタート処理 (レーベル名称 : START_TASK)
なし
- ・送信処理 (レーベル名称 : TRANS_TASK)
Aレジスタ
- ・受信処理 (レーベル名称 : RECEIVE_TASK)
Aレジスタ
- ・ACK確認処理 (レーベル名称 : ACK_TASK)
なし

(b) 特殊機能レジスタ

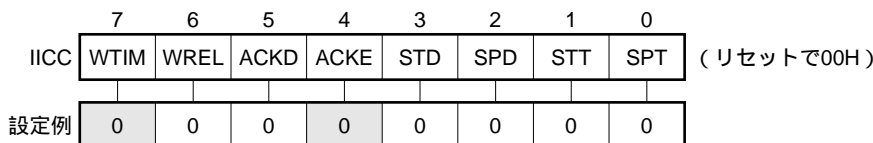
ポート3モード・コントロール・レジスタ (PMC3)



クロック同期式シリアル・インタフェース・モード・レジスタ (CSIM)



I²Cバス・コントロール・レジスタ (IICC) (1/2)



WTIM	0	8クロック・ウェイト。SCLの8クロック目の立ち下がりで割り込み要求 (INTCSI) を発生する。 ・マスタの場合 : 8クロック出力後, SCL出力をロウ・レベルにしてウェイトする。 ・スレーブの場合 : 8クロック入力後, SCL端子をロウ・レベルにしてウェイト要求をする。
	1	9クロック・ウェイト。SCLの9クロック目の立ち下がりで割り込み要求 (INTCSI) を発生する。 ・マスタの場合 : 9クロック出力後, SCL出力をロウ・レベルにしてウェイトする。 ・スレーブの場合 : 9クロック入力後, SCL端子をロウ・レベルにしてウェイト要求をする。

WREL	WREL = 1 により, ウェイト状態を解除 (SCLをハイ・レベル) する。 ウェイト状態を解除後, WRELビットは自動的にクリア (0) される。
------	--

ACKD	クリアされる条件 (ACKD = 0)	セットされる条件 (ACKD = 1)
	アクノリッジ信号 (\overline{ACK}) 検出後のウェイト解除 WREL = 1 または SIOライトまたは SPT = 1 時 CTXE = 0 リセット入力時	アクノリッジ信号 (\overline{ACK}) 検出時 (9発目のSCLの立ち上がりで, SDAがロウ・レベルのとき)

ACKE	0	アクノリッジ信号の自動出力を禁止。 送信時, または 8クロック・ウェイト選択時に使用する。ただし, WUP = 1 でアドレス受信時は次の動作となる。 ・自アドレス受信時 : ACEK = 0 の場合でも SCL の 9クロック目の立ち下がりに同期してアクノリッジ信号の自動出力を行う。 ・自アドレス以外受信時 : アクノリッジ信号の自動出力は行わない。
	1	8クロック・ウェイト選択時 : ウェイト制御を行う前に ACEK = 1 とすることにより, SCL の 8クロック目の立ち下がりに同期してアクノリッジ信号を出力する。 9クロック・ウェイト選択時 : あらかじめ ACEK = 1 とすることにより, アクノリッジ信号を SCL の 8クロック目の立ち下がりに同期してアクノリッジ信号を自動出力する。 ただし, WUP = 1 でアドレス受信時は次の動作となる。 ・自アドレス受信時 : アクノリッジ信号の自動出力を行う。 ・自アドレス以外受信時 : アクノリッジ信号の自動出力は行わない。

I²Cバス・コントロール・レジスタ (IICC) (2/2)

	7	6	5	4	3	2	1	0	
IICC	WTIM	WREL	ACKD	ACKE	STD	SPD	STT	SPT	(リセットで00H)
設定例	0	0	0	0	0	0	0	0	

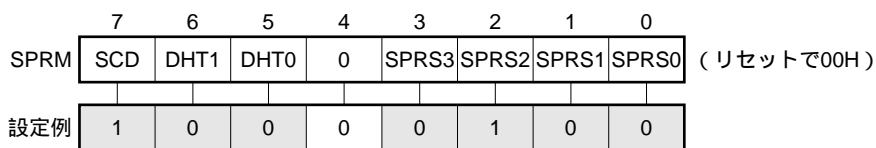
STD	クリアされる条件 (STD=0)	セットされる条件 (STD=1)
	スタート・コンディション検出後のウエイト解除(WREL=1またはSIOライト ^注 またはSPT=1)時 CTXE=0 リセット入力時 注 マスタ時のアドレス・ライトは除く	・WUP=0のとき：スタート・コンディション検出時 ・WUP=1のとき：自アドレス検出時

SPD	クリアされる条件 (SPD=0)	セットされる条件 (SPD=1)
	スタート・コンディション検出時 CTXE=0 リセット入力時	ストップ・コンディション検出時

STT	<p>SCLおよびSDAがハイ・レベルの状態^注でSTT=1にすることにより、SOラッチをクリア(0)する。SOラッチをクリア(0)後、SCLをロウ・レベルにする。SCLをロウ・レベルにしたあと、STTビットは自動的にクリア(0)される。</p> <p>注 SCLのレベル状態は、P32/SCL端子を入力モード(PM32=1)とし、SCL*を読み込むことにより確認できます。また、SDAのレベル状態も、P33/SDA端子を入力モード(PM33=1)とし、SDA*を読み込むことにより確認できます。</p> <p>* SCLおよびSDAは、NEC製アセンブラでは予約語に、Cコンパイラでは#pragma sfr指令で、sfr変数として定義されています。</p> <p>注意1．SCLおよびSDAがロウ・レベルの状態^注でSTT=1にしても、スタート・コンディションは出力されません(STT=1にしたあと、SCLがハイ・レベルとなってもスタート・コンディションは出力されません)。</p> <p>2．STTをセット(1)後、必ずNOPなどで1命令以上実行してからSIOにアドレスを書き込んでください。</p>
-----	--

SPT	SPT=1により、SOラッチがクリア(0)され、SOラッチがクリア(0)されたあと、SCLをハイ・レベルにする。SCLがハイ・レベルになったあと、SOラッチをセット(1)する。SOラッチをセット(1)後、SPTビットは自動的にクリア(0)される。
-----	---

シリアル・クロック用プリスケアラ・モード・レジスタ (SPRM)

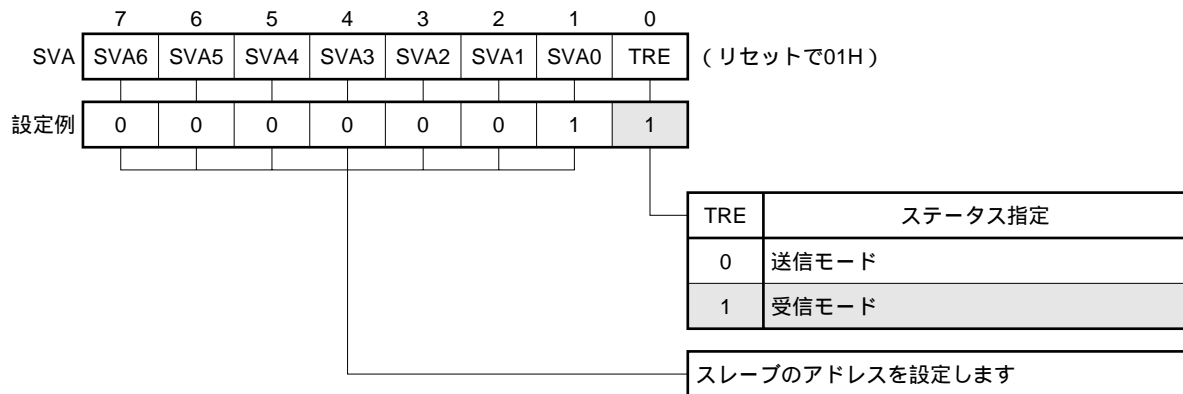


SPRS3	SPRS2	SPRS1	SPRS0	シリアル・クロックの指定
0	0	0	0	$f_{xx}/16$
0	0	0	1	$f_{xx}/24$
0	0	1	0	$f_{xx}/32$
0	0	1	1	$f_{xx}/48$
0	1	0	0	$f_{xx}/64$
0	1	0	1	$f_{xx}/96$
0	1	1	0	$f_{xx}/128$
0	1	1	1	$f_{xx}/192$
1	0	0	0	$f_{xx}/256$
1	0	0	1	$f_{xx}/384$

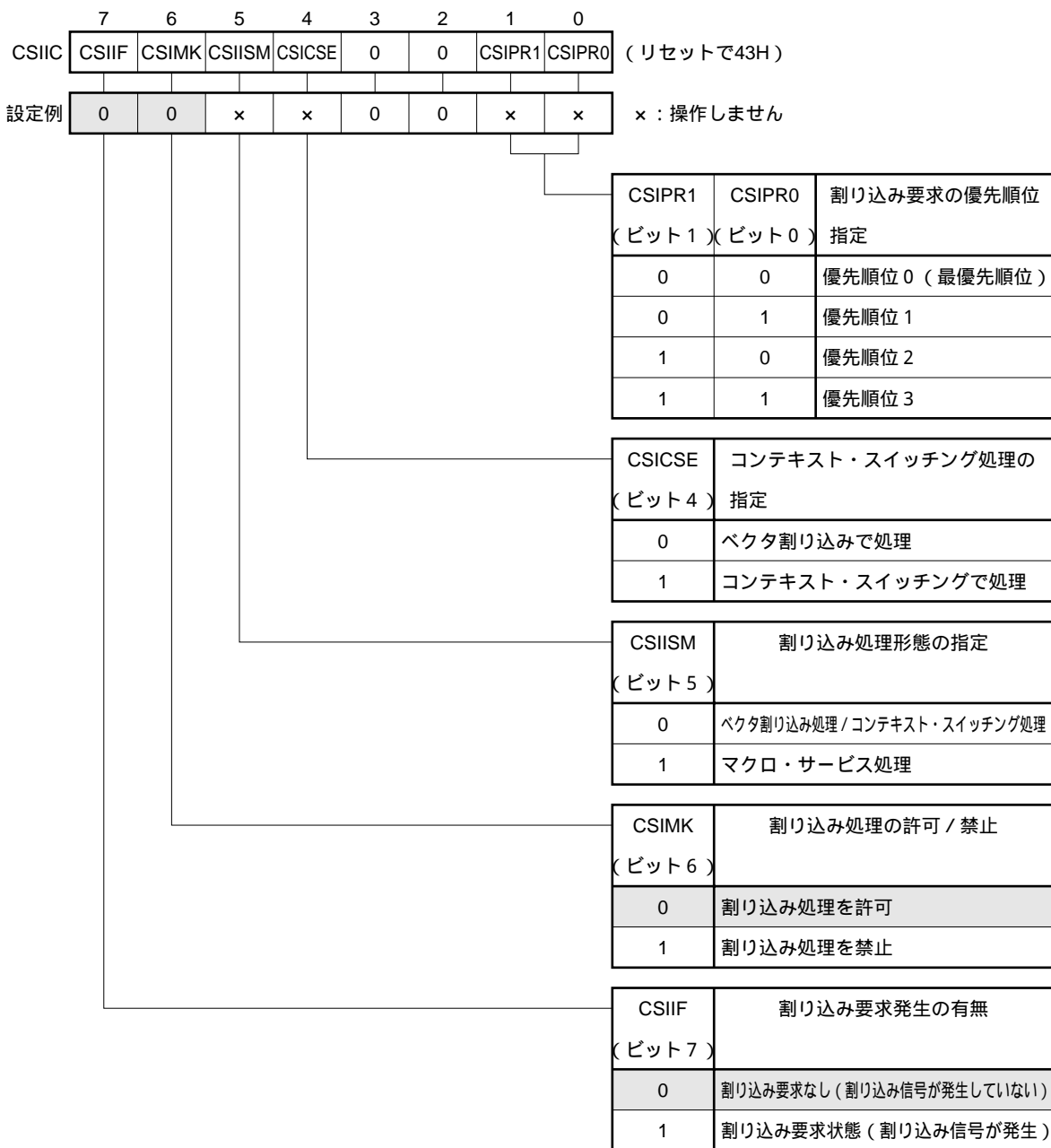
DHT1	DHT0	データ・ホールド時間指定	
0	0	$16 \text{ MHz} < f_{xx}$	32 MHz $10/f_{xx} \sim 12/f_{xx}$
0	1	$8 \text{ MHz} < f_{xx}$	16 MHz $5/f_{xx} \sim 7/f_{xx}$
1	x	$4 \text{ MHz} < f_{xx}$	8 MHz $3/f_{xx} \sim 5/f_{xx}$

SCD	シリアル・クロック・デューティ指定
0	標準モード用
1	高速モード用

スレーブ・アドレス・レジスタ (SVA)



割り込み制御レジスタ (CSIIC)



(5) 入力方法

なし

(6) 出力方法

なし

(7) プログラムの説明

【マスタ側のプログラム説明】

イニシャライズ処理 [レーベル名称 : INIT_I2C]

- (a) 通信終了割り込み要求フラグ (CSIF) のクリアを行います。
- (b) 通信終了割り込みのマスク (CSIMK) のクリアを行います。
- (c) P32, P33をコントロール・ポート (SCL, SDA) として設定します。
- (d) シリアル・クロックの設定を行います。 (SPRMを設定)
- (e) ウエイトを8クロック・ウエイトに設定します。 (IICCを設定)
- (f) 送信許可にします。 (CSIMを設定)

I²C割り込み処理 [レーベル名称 : I2C_INT]

- (a) 1バイト目の処理を行います。 (アドレス送信と $\overline{\text{ACK}}$ の確認)
 - (i) 処理カウンタ (M_SHORI) が “ 0 ” であることを確認し、0の場合には (ii) 以降の処理を続けます。0以外の場合には (b) の処理に飛びます。
 - (ii) 送信動作中フラグの値を確認します。送信動作中の場合 (_TRANSFER = 1) , $\overline{\text{ACK}}$ 確認処理ルーチンをCALLします。送信動作中ではない場合 (_TRANSFER = 0) , 送受信スタート処理ルーチンをCALLします。
- (b) 2バイト目の処理を行います。 (データ送信と $\overline{\text{ACK}}$ の確認)
 - (i) 処理カウンタ (M_SHORI) が “ 1 ” であることを確認し、1の場合には (ii) 以降の処理を続けます。
 - 1以外の場合には (c) の処理に飛びます。
 - (ii) 送信動作中フラグの値を確認します。送信動作中の場合 (_TRANSFER = 1) , $\overline{\text{ACK}}$ 確認処理ルーチンをCALLします。送信動作中ではない場合 (_TRANSFER = 0) , 送信処理ルーチンをCALLします。

(c) 3 バイト目の処理を行います。(データ送信と $\overline{\text{ACK}}$ の確認)

(i) 処理カウンタ (M_SHORI) が “ 2 ” であることを確認し、2 の場合には (ii) 以降の処理を続けます。

2 以外の場合には (d) の処理に飛びます。

(ii) 送信動作中フラグの値を確認します。送信動作中の場合 (_TRANSFER = 1) , $\overline{\text{ACK}}$ 確認処理ルーチンをCALLします。送信動作中ではない場合 (_TRANSFER = 0) , 送信処理ルーチンをCALLします。

(d) 4 バイト目の処理を行います。(リスタートのために、アドレス送信と $\overline{\text{ACK}}$ の確認)

(i) 処理カウンタ (M_SHORI) が “ 3 ” であることを確認し、3 の場合には (ii) 以降の処理を続けます。

3 以外の場合には (e) の処理に飛びます。

(ii) 送信動作中フラグの値を確認します。送信動作中の場合 (_TRANSFER = 1) , $\overline{\text{ACK}}$ 確認処理ルーチンをCALLします。送信動作中ではない場合 (_TRANSFER = 0) , ウェイトを解除し、送受信スタート処理ルーチンをCALLします。

(e) 5 バイト目の処理を行います。(データ受信と $\overline{\text{ACK}}$ の確認)

(i) 処理カウンタ (M_SHORI) が “ 4 ” であることを確認し、4 の場合には (ii) 以降の処理を続けます。

4 以外の場合には (f) の処理に飛びます。

(ii) 受信動作中フラグの値を確認します。受信動作中の場合 (_RECEIVE = 1) , 受信処理ルーチンをCALLします。受信動作中ではない場合 (_RECEIVE = 0) , シフト・レジスタ (SIO) にOFFHを書き込み、受信待ち状態にします。

(f) 6 バイト目の処理を行います。(データ受信と $\overline{\text{ACK}}$ の確認)

(i) 処理カウンタ (M_SHORI) が “ 5 ” であることを確認し、5 の場合には (ii) 以降の処理を続けます。

5 以外の場合には (g) の処理に飛びます。

(ii) 受信動作中フラグの値を確認します。受信動作中の場合 (_RECEIVE = 1) , 受信処理ルーチンをCALLします。受信動作中ではない場合 (_RECEIVE = 0) , シフト・レジスタ (SIO) にOFFHを書き込み、受信待ち状態にします。

(g) 7バイト目の処理を行います。(データ受信と $\overline{\text{ACK}}$ の確認)

(i) 処理カウンタ(M_SHORI)が“6”であることを確認し、6の場合には(ii)以降の処理を続けます。

6以外の場合には本割り込み処理を終了します。

(ii) ストップ・コンディション・トリガ・ビット(STTビット)をセットします。処理カウンタを0にクリアします。

送受信スタート処理 [レーベル名称 : START_TASK]

(a) スタート・コンディション・トリガ・ビット(STTビット)をセットします。STTのセット直後にNOPを実行します。

(b) スレーブ・アドレスと送信方向(マスタ スレーブ)をシフト・レジスタ(SIO)に書き込みます。

送信処理 [レーベル名称 : TRANS_TASK]

(a) 送信データをシフト・レジスタ(SIO)に書き込みます。

受信処理 [レーベル名称 : RECEIVE_TASK]

(a) スタート・レジスタ(SIO)から受信データを読み込みます。

(b) 受信データが最後のデータであるかどうかを確認し、最後のデータではない場合 $\overline{\text{ACK}}$ を返します。最後のデータである場合は、 $\overline{\text{ACK}}$ を返さず^注にウェイトを解除し、本処理を終了します。

注 マスタ受信時に、ストップ・コンディションを発行する直前ではウェイトを解除してはいけません。

$\overline{\text{ACK}}$ 確認処理 [レーベル名称 : ACK_TASK]

(a) $\overline{\text{ACK}}$ が返されたことを確認します。

(b) 送受信の処理カウンタ(M_SHORI)をカウント・アップします。

【スレーブ側のプログラム説明】

イニシャライズ処理 [レーベル名称 : INIT_I2C]

(a) 通信終了割り込み要求フラグ(CSIIF)のクリアを行います。

(b) 通信終了割り込みのマスク(CSIIMK)のクリアを行います。

(c) P32, P33をコントロール・ポート(SCL, SDA)として設定します。

(d) シリアル・クロックの設定を行います。(SPRMを設定)

(e) ウェイトを8クロック・ウェイトに設定します。(IICCを設定)

(f) 送信許可にします。(CSIMを設定)

I²C割り込み処理 [レーベル名称 : I2C_INT]

- (a) スタート・コンディションが発行されているかどうかを確認し、発行されている場合、送受信スタート処理ルーチンをCALLします。
- (b) スタート・コンディションが発行されていない場合、送受信動作確認フラグ (Rset_Treset) の設定により送信動作確認フラグの設定により送信動作中であるのか、受信動作中であるのかを確認します。
- (c) 受信動作中 (Rset_Treset = 1) の場合、受信処理ルーチンをCALLします。
- (d) 送信動作中 (Rset_Treset = 0) の場合、送信動作中か否かを確認し、送信動作中の場合 (_TRANSFER = 1) , $\overline{\text{ACK}}$ を返し、送信動作中でない場合 (_TRANSFER = 0) , 送受信処理ルーチンをCALLします。

送受信スタート処理 [レーベル名称 : START_TASK]

- (a) ステータス・ビット (TREビット) を確認し、送受信のどちらが行われるかを確認するとともに送受信動作確認フラグ (Rset_Treset) の設定を行います。
- (b) 受信動作 (Rset_Treset = 1) の場合、シフト・レジスタ (SIO) に0FFHを設定します。
- (c) 送信動作 (Rset_Treset = 0) の場合、送信処理ルーチンをCALLします。

送信処理 [レーベル名称 : TRANS_TASK]

- (a) 送信データをシフト・レジスタ (SIO) に書き込みます。
- (b) 送信回数カウンタ (TRANCE_TASK) をカウント・アップします。

受信処理 [レーベル名称 : RECEIVE_TASK]

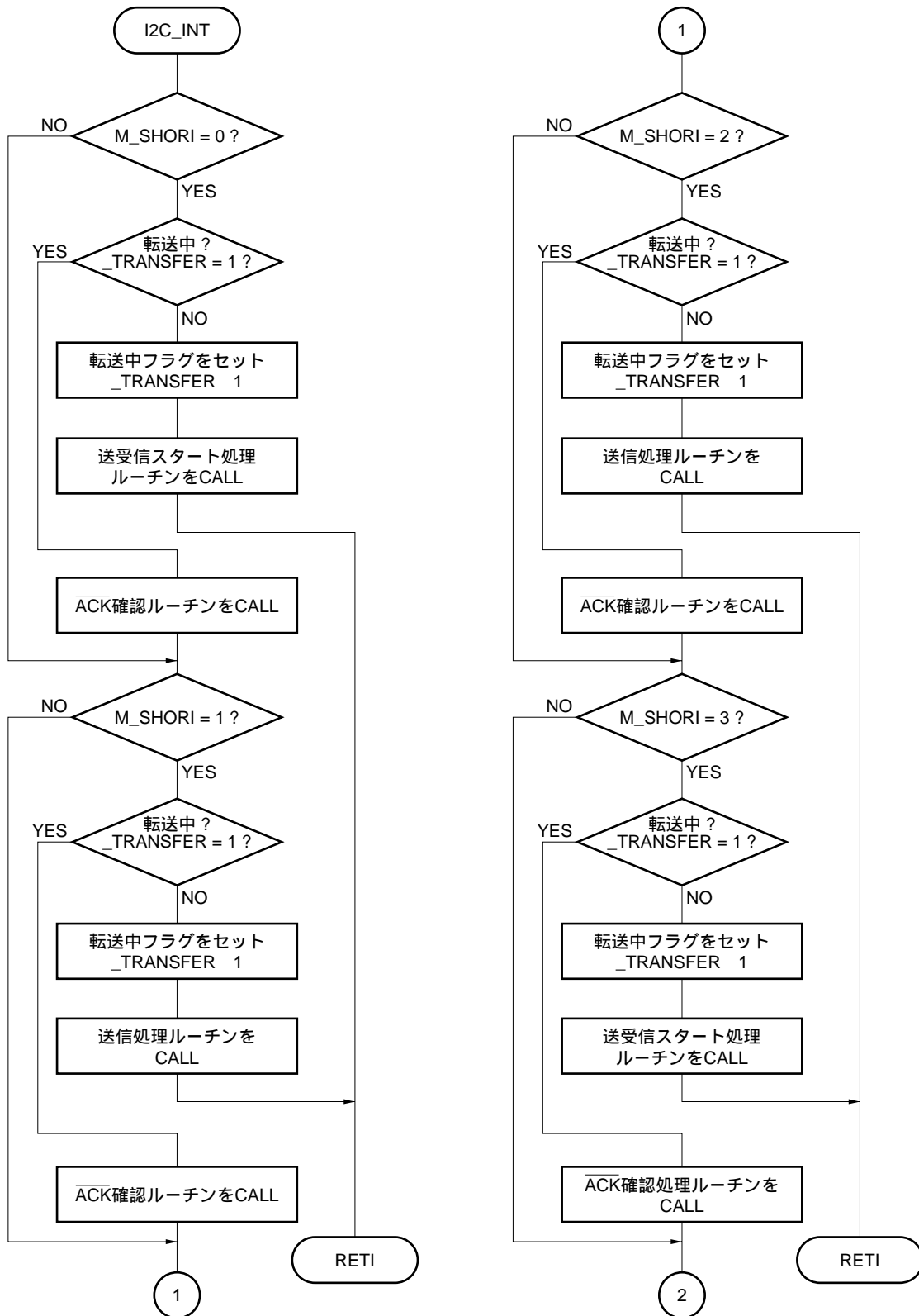
- (a) シフト・レジスタ (SIO) から受信データを読み込みます。
- (b) $\overline{\text{ACK}}$ を返すとともに、シフト・レジスタ (SIO) に0FFHを書き込み、受信待ち状態にします。

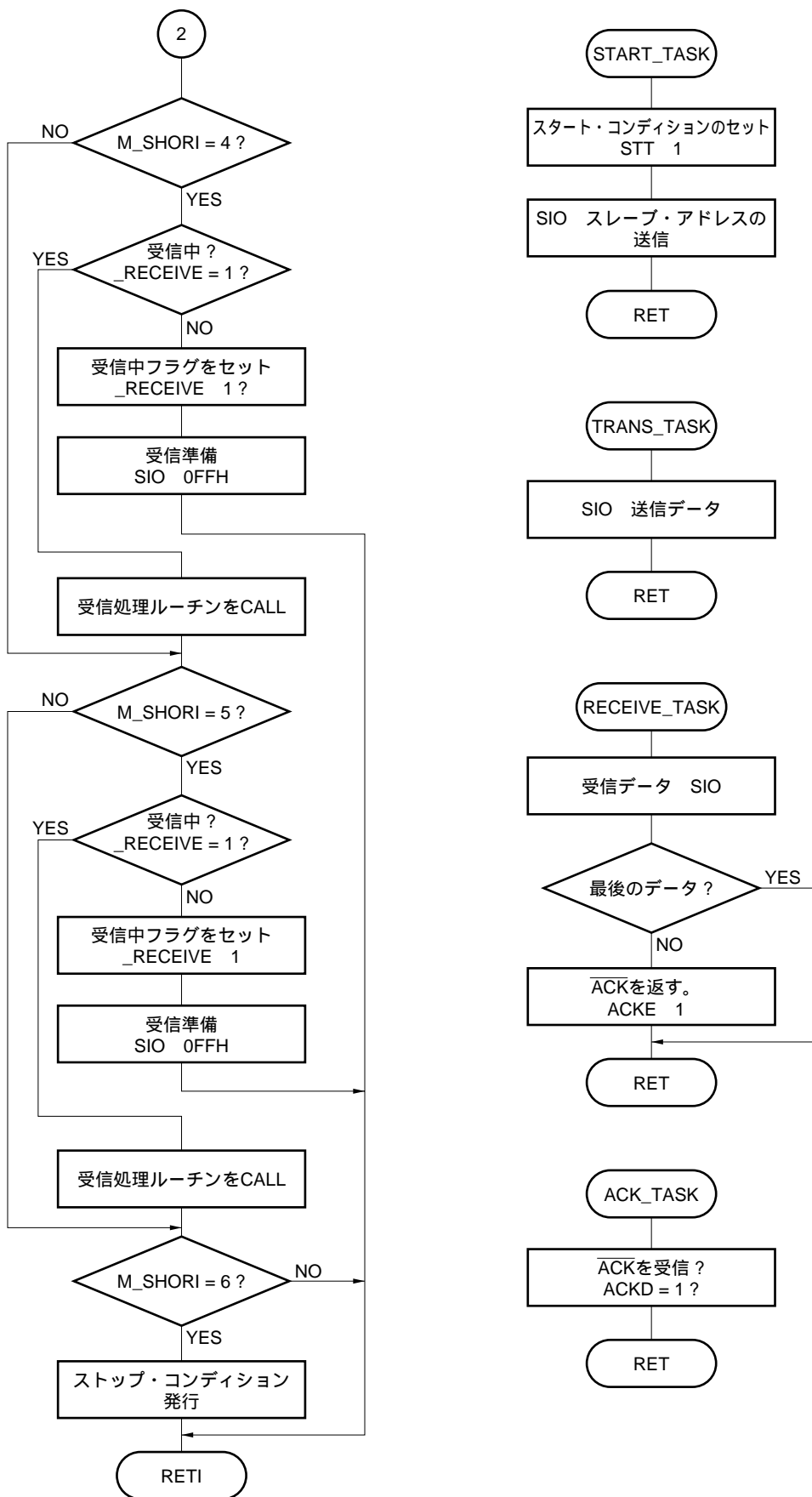
 $\overline{\text{ACK}}$ 確認処理 [レーベル名称 : ACK_TASK]

- (a) $\overline{\text{ACK}}$ が返されたことを確認します。
- (b) $\overline{\text{ACK}}$ が返されていない場合、最終データの受信のACKの確認ならば、ウェイトの解除をするとともに、ストップ・コンディションの発行を確認します。

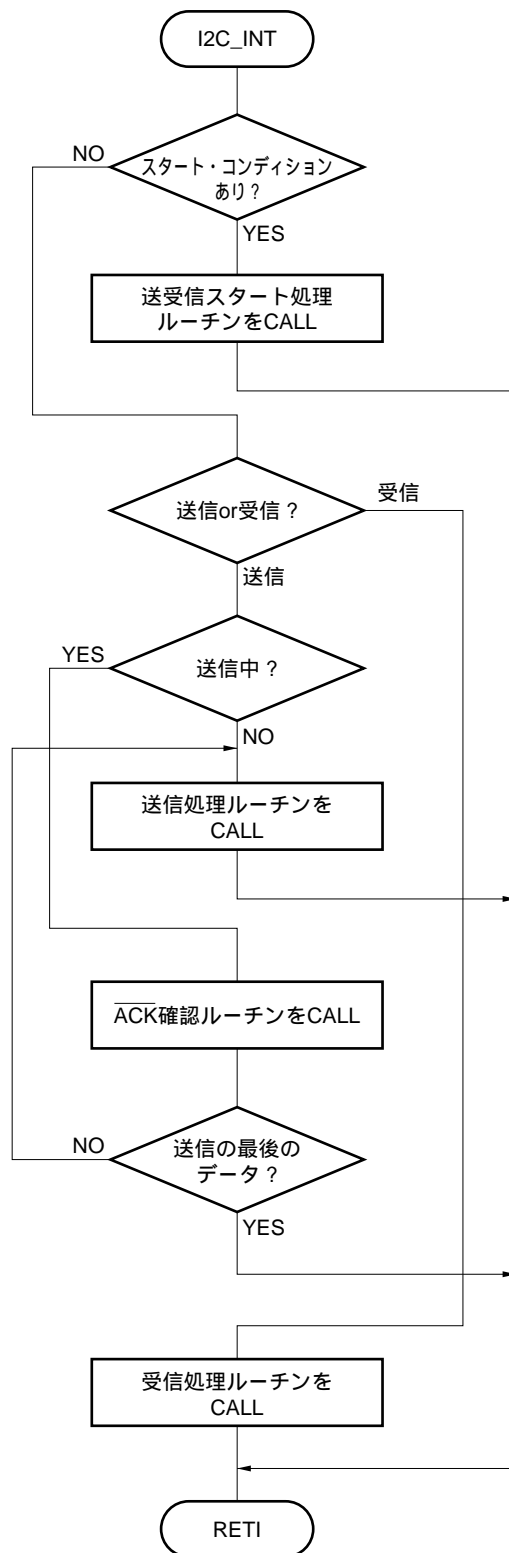
(7) フロー・チャート

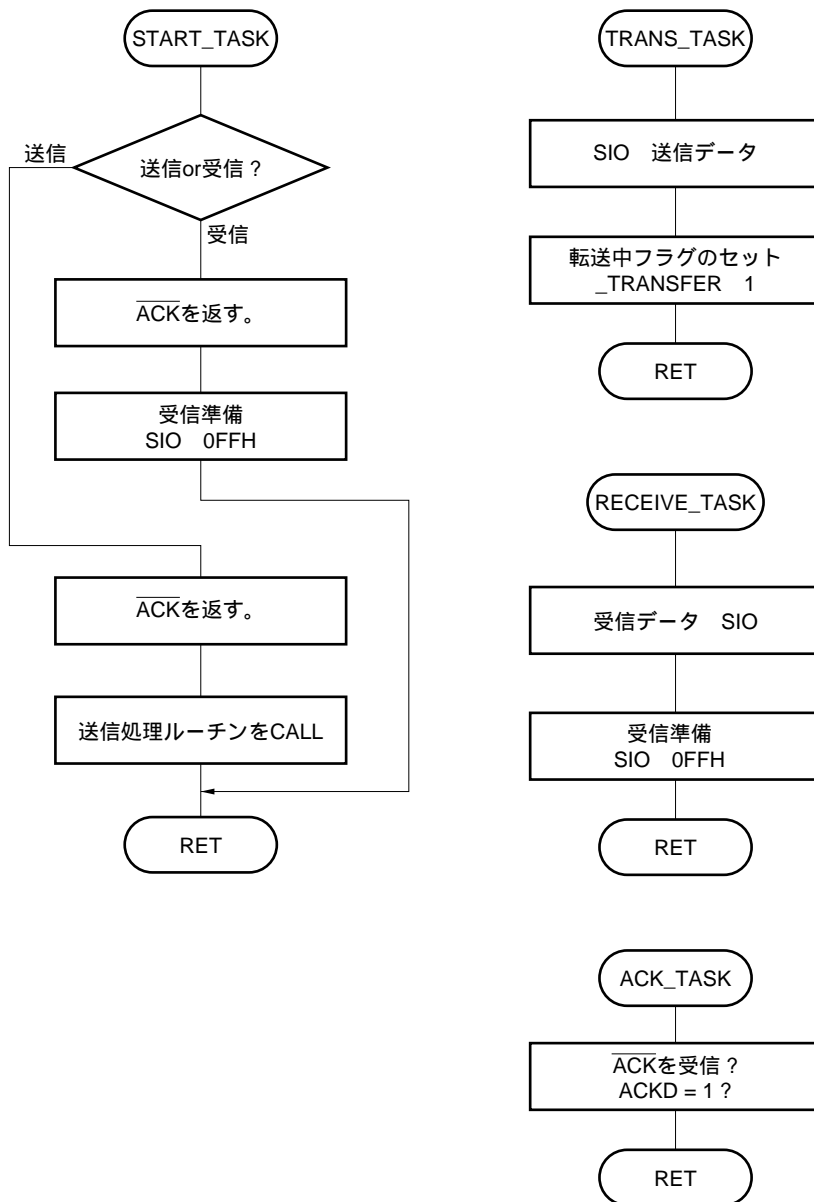
【マスタ側のフロー・チャート】





【スレーブ側のフロー・チャート】





(9) プログラム・リスト

【マスタ側のプログラム・リスト】

```

*****
;
;
;
;*   IIC APPLICATION NOTE PROGRAM
;*
;*   MASTER
;
;
*****
STKBGNEQU      OFE00H
;
;
;***** define work area *****
;
;      DSEG      SADDR
M_SHORI:      DS      1           ;SHORI NUMBER
ADDRESS:      DS      1           ;TRANSFER ADDRESS INPUT
DATA_T:       DS      1           ;TRANSFER DATA
DATA_R:       DS      1           ;RECEIVE DATA
WORK:         DS      1           ;RAM FOR FLAG
;
;      _RECEIVE EQU      WORK.0
;      LAST_DATA EQU      WORK.1
;      _TRANSFER EQU      WORK.2
;
;
;***** define vector table *****
;
;VRST  CSEG      AT      0H
;      DW      RST
;
;VIIC  CSEG      AT      28H
;      DW      IIC_START
;
;
;      CSEG      BASE
;
;
;RST:
;
;      LOCATION      0H
;      MOVG          SP,#STKBGN
;      MOV           STBC,#00H
;      MOV           MM,#80H
;      DI
;
;
;
;      MOV           M_SHORI,#0H
;      MOV           DATA_T,#0H
;      MOV           DATA_R,#0H
;      MOV           WORK,#0H
;
;
;INIT_I2C:
;
;
;      interrupt initialize
;
;
;      CLR1          CSIIF
;      CLR1          CSIMK
;
;
;      IIC serial bus interface initialize
;
;
;      MOV           PMC3,#0CH
;      MOV           CSIM,#10001110B      ;<= CTXE=1,WUP=0,MASTER
;      MOV           IICC,#00000000B     ;<= 8 clock wait

```

```

MOV          SPRM,#10000100B          ;= FAST MODE, fxx/64=390kHz,
;                                         ; fxx=25MHz
;
SET1         CSIIF                     ;<= IIC START
EI
;
; *****
; ***      main      ***
; *****
MAIN:
MOV          PO,M_SHORI
BR          MAIN
;
;=====
;      IIC ROUTINE  "INTERRUPT (INTCSI)"
;=====
;
I2C_INT:
CMP          M_SHORI,#0H
BNZ         $SHORI_2nd

BTCLR       _TRANSFER,$ACK_OK_1?

MOV         ADDRESS,#2H                ;<= ADDRESS INPUT !!
; (ADR=1 & WR)

CALL        START_TASK
SET1        _TRANSFER
BR          IIC_END

ACK_OK_1?:
CALL        ACK_TASK
;
;
;
SHORI_2nd:
CMP          M_SHORI,#1H
BNZ         $SHORI_3rd

BTCLR       _TRANSFER,$ACK_OK_2?

MOV         DATA_T,#55H                ;<= DATA INPUT !!
CALL        TRANS_TASK
SET1        _TRANSFER
BR          IIC_END

ACK_OK_2?:
CALL        ACK_TASK
;
;
;
SHORI_3rd:
CMP          M_SHORI,#2H
BNZ         $SHORI_4th

BTCLR       _TRANSFER,$ACK_OK_3?

MOV         DATA_T,#0AAH                ;<= DATA INPUT !!

```

```

CALL          TRANS_TASK
SET1         _TRANSFER
BR          IIC_END

ACK_OK_3?:
CALL          ACK_TASK
;
;
;
SHORI_4th:
CMP          M_SHORI,#3H          ;<= RESTART
BNZ          $$SHORI_5th

BTCLR       _TRANSFER,$ACK_OK_4?

SET1        WREL          ;<= WAIT RELEASE FOR RESTART
MOV         ADDRESS,#3H      ;<= ADDRESS INPUT !!
; (ADR=1 & RD)

CALL        START_TASK
SET1       _TRANSFER
BR        IIC_END

ACK_OK_4?:
CALL          ACK_TASK
;
;
;
SHORI_5th:
CMP          M_SHORI,#4H
BNZ          $$SHORI_6th
;
BT          _RECEIVE,$SHORI_5th_1 ;<= Under receiving,
;IIC routine is ended.

SET1        _RECEIVE
;
CLR1        ACKE          ;<= Before received,
;ACK must not be output.

MOV         SIO,#OFFH
BR          IIC_END
;
SHORI_5th_1:
CALL        RECEIVE_TASK
;
SHORI_6th:
CMP          M_SHORI,#5H
BNZ          $$SHORI_7th
;
BT          _RECEIVE,$SHORI_6th_1 ;<= Under receiving,
;IIC routine is ended.

SET1        _RECEIVE
;
SET1        LAST_DATA    ;<= LAST DATA FLAG
CLR1        ACKE          ;<= Before received,
;ACK must not be output.

MOV         SIO,#OFFH
BR          IIC_END
;

```



```

;
SHORI_6th_1:
    CALL          RECEIVE_TASK
;
SHORI_7th:
    CMP           M_SHORI,#6H
    BNZ          $IIC_END
    SET1         SPT                ;<= STOP CONDITION
    MOV          M_SHORI,#0H
;
IIC_END:
    RETI
;
;=====
;          START ROUTINE
;=====
;
START_TASK:
    SET1         STT                ;<= START CONDITION
    NOP          ;<= NOP
    MOV          A,ADDRESS
    MOV          SIO,A
    RET
;
;=====
;          ACK ROUTINE
;=====
;
ACK_TASK:
    BF           ACKD,$ACK_TASK
    INC          M_SHORI            ;<= SHORI COUNT
    RET
;
;=====
;          DATA TRANSFER ROUTINE
;=====
;
TRANS_TASK:
    MOV          A,DATA_T
    MOV          SIO,A
    RET
;
;=====
;          DATA RECEIVE ROUTINE
;=====
;
RECEIVE_TASK:
    MOV          A,SIO              ;<= DATA RECEIVE
    MOV          DATA_R,A          ;<= DATA RECEIVE
    BF           LAST_DATA,$NOT_LAST_DATA ;<= In the case of last data,
                                        ;ACK is not returned.
    CLR1         LAST_DATA
    BR          LAST
;
NOT_LAST_DATA:
    SET1         ACKE                ;<= ACK OUTPUT

```

```
LAST:
      SET1          WREL
      CLR1          UNDER_RECEIVE      ;<= RECEIVE END
      INC           M_SHORT            ;<= SHORT COUNT
      RET
      ;
      ;
      ;
      ;
      END
```

【スレーブ側のアドレス】

```

*****
;
; *
; *
; *   IIC APPLICATION NOTE PROGRAM
; *
; *           SLAVE
; *
; *
; *****
STKDBGNEQU          OFE00H
;
;
; ***** define work area *****
;
;           DSEG          SADDR
DATA_T:              DS          1           ;TRANSFER DATA
DATA_R:              DS          1           ;RECEIVE DATA
TRANCE_COUNTER:     DS          1           ;TRANCEFER COUNTER
WORK:                DS          1           ;RAM FOR FLAG
;
;           Rset_Treset    EQU      WORK.0
;           _TRANSFER      EQU      WORK.1
;
; ***** define vector table *****
VRST   CSEG          AT          0H
;
;           DW            RST
VIIC   CSEG          AT          28H
;
;           DW            IIC_START
;
;           CSEG          BASE
;
RST:
;
;           LOCATION      0H
;           MOVG          SP,#STKDBGN
;           MOV           STBC,#00H
;           MOV           MM,#80H
;           DI
;
;
;
;           MOV           DATA_T,#0H
;           MOV           DATA_R,#0H
;           MOV           TRANCE_COUNTER,#0H
;           MOV           WORK,#0H
;
INIT_I2C:
;
;           interrupt initialize
;
;           CLR1          CSIIF
;           CLR1          CSIMK
;
;
;           IIC serial bus interface initialize
;
;           MOV           PMC3,#0CH
;           MOV           CSIM,#10101100B           ;<= CTXE=1,WUP=1,SLAVE
;           MOV           IICC,#00000000B           ;<= 8 clock wait
;           MOV           SPRM,#10000100B           ;<= FAST MODE, fxx/64=390kHz
;
;           MOV           SVA,#3H                   ;in fxx=25MHz
;
;
;           MOV           SVA,#3H                   ;<= SLAVE ADDRESS = 1
;

```

```

EI
;*****
;***      main      ***
;*****
MAIN:
    BR      MAIN
;
;=====
;      IIC ROUTINE  "INTERRUPT (INTCSI)"
;=====
;
IIC_INT:
    BF      STD,$TASK_SELECT
    CALL   START_TASK          ;<= TO START ROUTINE
    BR      IIC_END
;
TASK_SELECT:
    BT      Rset_Treset,$RECEIVE_SELECT
                                ;<= TO RECEIVE or
TRANSFER
;
TRANSFER_SELSC:
    BTCLR   _TRANSFER,$ACK_OK?
TRANS_AGAIN:
    CALL   TRANS_TASK          ;<= TO TRANSFER ROUTINE
    BR      IIC_END
ACK_OK?:
    CALL   ACK_TASK
    CMP    TRANCE_COUNTER,#2H
    BNZ    $TRANS_AGAIN
    MOV    TRANCE_COUNTER,#0H
    BR      IIC_END
;
RECEIVE_SELECT:
    CALL   RECEIVE_TASK        ;<= TO RECEIVE ROUTINE
;
IIC_END:
    RETI
;
;=====
;      START ROUTINE
;=====
;
START_TASK:
    BF      TRE,$TRANS_START
RECEIVE_START:
    SET1   Rset_Treset
    SET1   ACKE
    MOV    SIO,#OFFH
    BR      START_END
TRANS_START:
    CLR1   Rset_Treset
    SET1   ACKE
    CALL   TRANS_TASK
START_END:
    RET
;

```

```

;=====
;      DATA TRANSFER ROUTINE
;=====
;
;
TRANS_TASK:
    MOV     A,DATA_T
    MOV     SIO,A
    SET1    _TRANSFER
    INC     TRANCE_COUNTER
    RET

;
;=====
;      ACK ROUTINE
;=====
;
;
ACK_TASK:
    BT      ACKD,$TRANS_END
    CMP     TRANCE_COUNTER,#2H
    BNZ     $ACK_TASK
    SET1    WREL
    BF      SPD,$ACK_TASK
TRANS_END:
    RET

;
;=====
;      DATA RECEIVE ROUTINE
;=====
;
;
RECEIVE_TASK:
    MOV     A,SIO                ;<= DATA RECEIVE
    MOV     DATA_R,A           ;<= DATA RECEIVE
    SET1    ACKE                 ;<= ACK OUTPUT
    MOV     SIO,#OFFH
    RET

;
;
;
END

```

〔メ モ〕

第4章 割り込み処理のプログラム例

4.1 割り込み機能概要

μPD784038, 784038Yサブシリーズは割り込み要求の処理形態として、表4-1に示すような3つの処理モードを選択できます。

表4-1 割り込み要求の処理形態

処理モード	処理の形態	処 理	割り込み時のPC, PSWの内容
ベクタ割り込み	ソフトウェア	ベクタ・テーブルで指定されたアドレス ^注 のサービス・プログラムに分岐し、実行	スタックへの退避 / 復帰を行う
コンテキスト・スイッチング		ベクタ・テーブルで指定されたレジスタ・バンクへの切り替えを自動的に行い、レジスタ・バンク中の固定エリアで指定されたアドレス ^注 のサービス・プログラムに分岐し、実行	レジスタ・バンク中の固定エリアへの退避 / 復帰を行う
マクロ・サービス	ハードウェア (ファームウェア)	メモリとI/O間とのデータ転送などあらかじめ設定した処理を行う	保持

注 すべての割り込みサービス・プログラムの先頭アドレスは、ベース領域にある必要があります。ベース領域にサービス・プログラムの本体が配置できない場合は、ベース領域内にサービス・プログラムへの分岐命令を記述してください。

ここでは、各割り込み要求の処理モードのプログラム例として、4.2にベクタ割り込みとコンテキスト・スイッチングのプログラム例を、また、4.3にマクロ・サービスのプログラム例を紹介します。

(1) 割り込み要求タイプ別の割り込み機能の特徴

割り込み要求タイプには、ノンマスクابل割り込みとマスクابل割り込みがあります。

表4 - 2に、ノンマスクابل割り込みとマスクابل割り込みの割り込み機能の特徴を説明します。

表4 - 2 割り込み要求タイプ別の割り込み機能の特徴

割り込み要求タイプ 項目	ノンマスクابل割り込み	マスクابل割り込み
割り込み要因	NMI端子割り込み入力または、ウォッチドッグ・タイマにより割り込み発生	割り込みマスク・フラグを持つ割り込みにより割り込み発生
割り込みの受け付け	割り込みの禁止/許可にかかわらず、受け付け可能	割り込み要求許可フラグが許可 (IEフラグ = 1) になっており、対象の割り込みマスク・フラグのマスクが解除 (0) されているとき、その割り込み要求フラグの要求があった (1) のときに受け付け可能
割り込み優先順位制御	NMI端子割り込み入力とウォッチドッグ・タイマの優先制御可能	デフォルト優先順位による制御と優先順位指定フラグの設定によるプログラマブル優先順位制御可能
割り込み優先順位	同一のノンマスクابل割り込みまたは優先順位の高いノンマスクابل割り込みサービス・プログラムの実行中を除いて、必ず受け付ける	<ul style="list-style-type: none"> ・複数の割り込みが同時に発生した場合 (1) マクロ・サービス処理を優先的に受け付け (2) プログラマブル優先順位に従って受け付け (マクロ・サービスを指定した場合を除く) (3) 同一優先順位またはマクロ・サービスが指定された場合は、デフォルト優先順位に従う ・割り込み処理中にさらに別の割り込みを受け付ける多重割り込み可能
割り込み処理モード	ベクタ割り込みによる受け付けのみ可能	ベクタ割り込み、コンテキスト・スイッチングおよびマクロ・サービスによる受け付け可能 (マクロ・サービスは一部の割り込みを除く)

(2) マスカブル割り込みの優先順位

μPD784038, 784038Yサブシリーズは、割り込み処理中にさらに別の割り込みを受け付ける多重割り込みの処理が可能です。さらに多重割り込みは、優先順位の制御も可能です。

優先順位制御には、次の2種類があります。

- ・デフォルト優先順位による優先順位制御
- ・優先順位指定フラグの設定によるプログラマブル優先順位制御

なお、割り込みを受け付けると、割り込み要求許可フラグが自動的にクリア(IEフラグ=0)され、割り込みが禁止されます。したがって、多重割り込みを使用する場合は、割り込み処理プログラム中でEI命令を実行して、割り込み許可状態にしてください。

(a) デフォルト優先順位による優先順位制御

プログラマブル優先順位制御で同一レベルに指定した複数の割り込みの要求、またはマクロ・サービス処理が指定されている割り込みの要求が同時に発生している場合、デフォルト優先順位に従って割り込み処理を行います。

表4-3に、μPD784038, 784038Yサブシリーズのマスクابل割り込みのデフォルト優先順位を示します。

表4-3 μPD784038, 784038Yサブシリーズのマスクابل割り込みのデフォルト優先順位

デフォルト 優先順位	割り込み要求発生ソース	
0	INTP0	端子入力エッジ検出
1	INTP1	端子入力エッジ検出
2	INTP2	端子入力エッジ検出
3	INTP3	端子入力エッジ検出
4	INTC00	タイマ/カウンタ0 (TM0-CR00)一致信号発生
5	INTC01	タイマ/カウンタ0 (TM0-CR01)一致信号発生
6	INTC10	タイマ/カウンタ1 (TM1-CR10またはTM1W-CR10W)一致信号発生
7	INTC11	タイマ/カウンタ1 (TM1-CR11またはTM1W-CR11W)一致信号発生
8	INTC20	タイマ/カウンタ2 (TM2-CR20またはTM2W-CR20W)一致信号発生
9	INTC21	タイマ/カウンタ2 (TM2-CR21またはTM2W-CR21W)一致信号発生
10	INTC30	タイマ3 (TM3-CR30またはTM3W-CR30W)一致信号発生
11	INTP4	端子入力エッジ検出
12	INTP5	端子入力エッジ検出
13	INTAD	A/D変換終了
14	INTSER	UART1受信エラー
15	INTSR	UART1受信終了
	INTCSI1	クロック同期式シリアル1転送終了
16	INTST	UART1送信終了
17	INTCSI	クロック同期式シリアル転送終了
18	INTSER2	UART2受信エラー
19	INTSR2	UART2受信終了
	INTCSI2	クロック同期式シリアル2転送終了
20	INTST2	UART2送信終了
21 ^注	INTSPC	I ² Cバス・ストップ・コンディション割り込み

注 μPD784038Yサブシリーズのみ

(b) 優先順位指定フラグの設定によるプログラマブル優先順位制御

優先順位指定フラグは、22種類（μPD784038サブシリーズは21種類）のマスクابل割り込みに対して、割り込み要因ごとの優先順位を指定するフラグです。

優先順位レベルは、4レベルまで指定でき、複数の割り込み要因を同じレベルに指定することができます。マスクابل割り込み要因のうち、レベル0が最も優先順位が高くなります。

優先順位レベルが同じ割り込み要因どうし内で複数の割り込みが同時に発生した場合は、デフォルト優先順位の順番に受け付けられます。

図4 - 1に、割り込み優先順位指定フラグの指定方法を示します。

図4 - 1 割り込み優先順位指定フラグの指定方法

割り込み制御レジスタ（x x ICn）

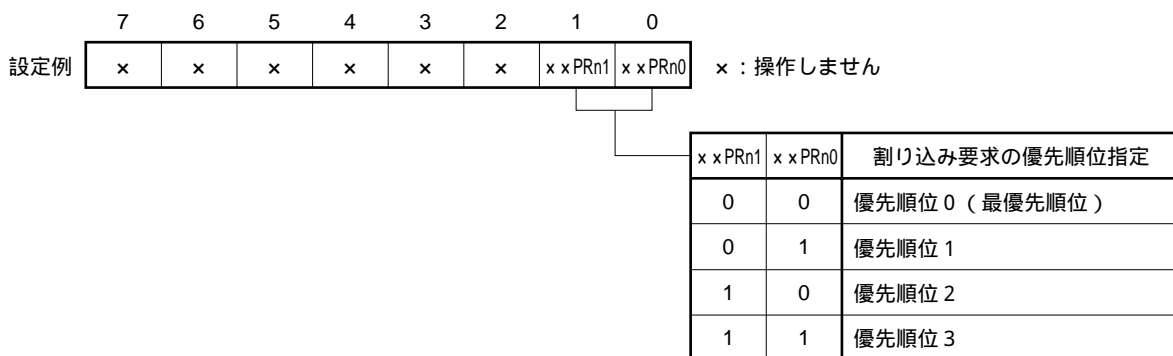
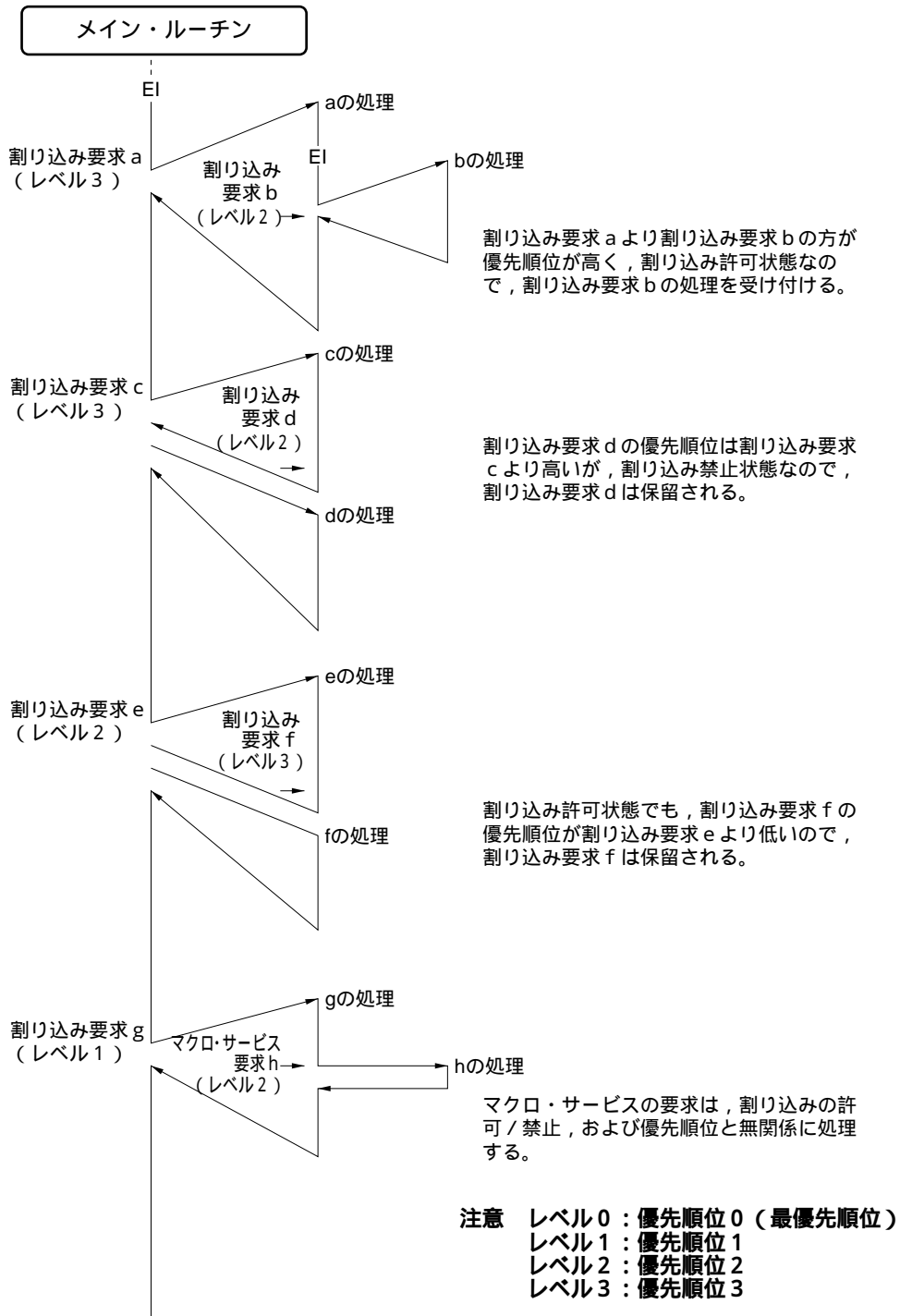


図4-2に、割り込み処理中に他の割り込みが発生した場合の処理例を示します。

図4-2 割り込み処理中に他の割り込みが発生した場合の処理例



4.1.1 ベクタ割り込み

(1) ベクタ割り込みの起動

割り込み要求許可フラグが許可 (IEフラグ = 1) になっており、対象の割り込みマスク・フラグが解除 (0) されているとき、割り込み制御レジスタのコンテキスト・スイッチング許可フラグを禁止 (0) することにより、ベクタ割り込みの起動を許可します。

ベクタ割り込みによるマスカブル割り込み要求が受け付けられると、次の順序で動作します。

プログラム・ステータス・ワード (PSW)、プログラム・カウンタ (PC) の順にスタック領域に退避します。

割り込み要求許可フラグを禁止 (IEフラグ = 0) します。

割り込みに対応するベクタ・テーブル中のデータをプログラム・カウンタ (PC) に設定することにより、割り込みサービス・プログラムに分岐します。

表4 - 4 に、μPD784038, 784038Yサブシリーズの各割り込みに対するベクタ・テーブル・アドレスを示します。

表4 - 4 μPD784038, 784038Yサブシリーズの各割り込みに対するベクタ・テーブル・アドレス

ベクタ・ テーブル・ アドレス	割り込み要求 発生ソース	ベクタ・ テーブル・ アドレス	割り込み要求 発生ソース
00006H	INTP0	0001EH	INTP5
00008H	INTP1	00020H	INTAD
0000AH	INTP2	00022H	INTSER
0000CH	INTP3	00024H	INTSR
0000EH	INTC00		INTCSI1
00010H	INTC01	00026H	INTST
00012H	INTC10	00028H	INTCSI
00014H	INTC11	0002AH	INTSER2
00016H	INTC20	0002CH	INTSR2
00018H	INTC21		INTCSI2
0001AH	INTC30	0002EH	INTST2
0001CH	INTP4	00030H	INTSPC ^注

注 μPD784038Yサブシリーズのみ

(2) ベクタ割り込みからの復帰

ベクタ割り込みを利用した割り込みからの復帰には、RETI命令を使用します。

割り込み処理の最後にRETI命令を実行すると、次の順で復帰します。

プログラム・ステータス・ワード (PSW) をスタック領域から復帰します (PSWには割り込み要求許可フラグ (IEフラグ) を含んでいますので、このときに割り込み要求許可フラグは許可 (IEフラグ=1) します)。

プログラム・カウンタ (PC) をスタック領域から復帰することにより、割り込み処理に移ったときのアドレスに復帰します。

4.1.2 コンテキスト・スイッチング機能

コンテキスト・スイッチングは、割り込みが受け付けられると、ハードウェアにより所定のレジスタ・バンクを選択し、レジスタ・バンク内にあらかじめ設定しておいた現在のPC, PSWをレジスタ・バンク内に退避すると同時に、ベクタ・アドレスに分岐する機能です。

(1) コンテキスト・スイッチングの起動

割り込み要求許可フラグが許可 (IEフラグ=1) になっており、対象の割り込みマスク・フラグが解除 (0) されているとき、割り込み制御レジスタのコンテキスト・スイッチング許可フラグを許可 (1) することにより、コンテキスト・スイッチング機能の起動を許可します。

コンテキスト動作は、次の順序で動作します。

対応するベクタ・テーブル・アドレスの下位アドレス (偶数アドレス) の下位3ビットで指定されるレジスタ・バンクに切り替えます。

割り込み要求許可フラグを禁止 (IEフラグ=0) します。

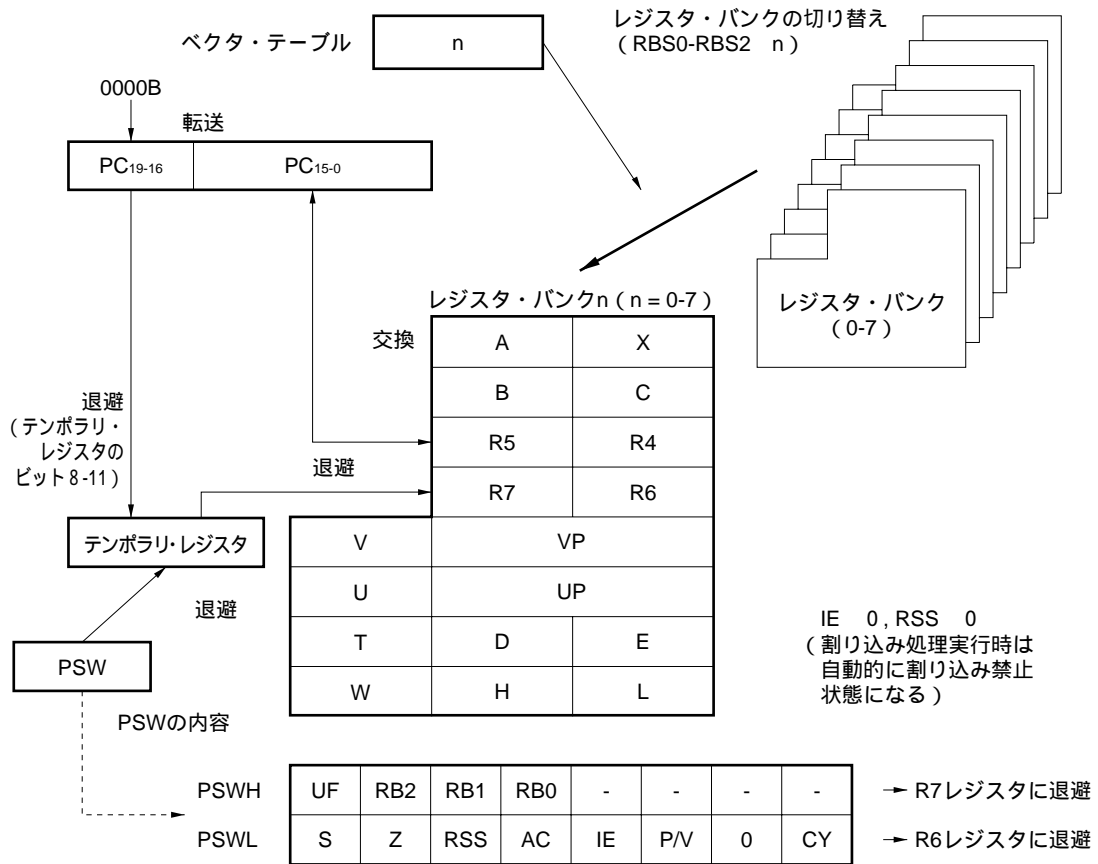
選択されたレジスタ・バンク内のR5レジスタとR4レジスタにあらかじめ設定しておいたベクタ・アドレスをプログラム・カウンタ (PC) の15-0ビットに転送すると同時に、それまでのプログラム・カウンタ (PC) の15-0ビットの内容をR5レジスタとR4レジスタに退避します。

プログラム・カウンタ (PC) の19-16ビットとPSWの内容をレジスタ・バンク内のR7レジスタとR6レジスタに退避します。

割り込みサービス・プログラムに分岐します。

図4-3に、割り込み要求の発生によるコンテキスト・スイッチング動作を示します。

図4-3 割り込み要求の発生によるコンテキスト・スイッチング動作



(2) コンテキスト・スイッチングからの復帰

コンテキスト・スイッチング機能を利用した割り込み復帰には、RETCS命令を使用します。RETCS命令では、次にその割り込みを受け付けたときに実行する割り込みサービス・プログラムの先頭アドレスを指定する必要があります。この割り込みサービス・プログラムの先頭アドレスは、ベース領域内にある必要があります。

割り込み処理の最後にRETCS命令を実行すると、次の順で復帰します。

割り込み時に選択されたレジスタ・バンク内のR5レジスタとR4レジスタに退避していた割り込み処理に移ったときのアドレスを、プログラム・カウンタ(PC)の15-0ビットに復帰します。

割り込み時に選択されたレジスタ・バンク内のR7レジスタとR6レジスタに退避していた割り込み処理に移ったときのアドレスを、プログラム・カウンタ(PC)の19-16ビットに復帰します。

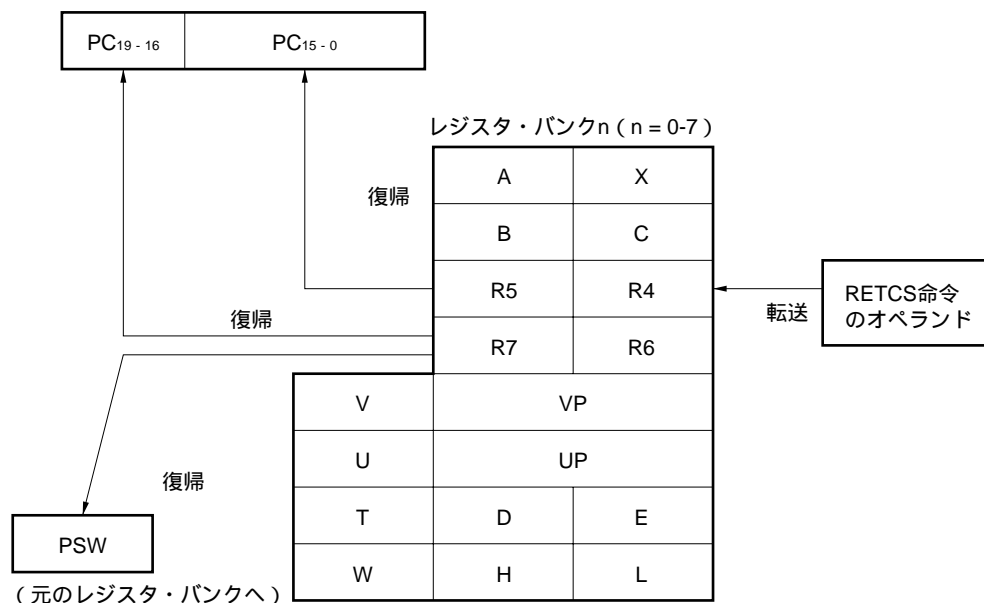
RETCS命令の第2, 第3オペコードで指定された、次にその割り込みを受け付けたときに実行する割り込みサービス・プログラムの先頭アドレスを割り込み時に選択されたレジスタ・バンク内のR5レジスタとR4レジスタへ転送します。

割り込み時に選択されたレジスタ・バンク内のR7レジスタの内容を、プログラム・ステータス・ワード(PSW)に復帰します(PSWには割り込み要求許可フラグ(IEフラグ)を含んでいますので、このときに割り込み要求許可フラグは許可(IEフラグ=1)します。また、レジスタ・バンク選択フラグも含んでいますので、このときに割り込み前のレジスタ・バンクに復帰します)。

割り込み処理に移ったときのアドレスに復帰します。

図4-4に、コンテキスト・スイッチング機能の割り込みからの復帰動作を示します。

図4-4 RETCS命令によるコンテキスト・スイッチング機能を使用した割り込みからの復帰動作



4.1.3 マクロ・サービス機能

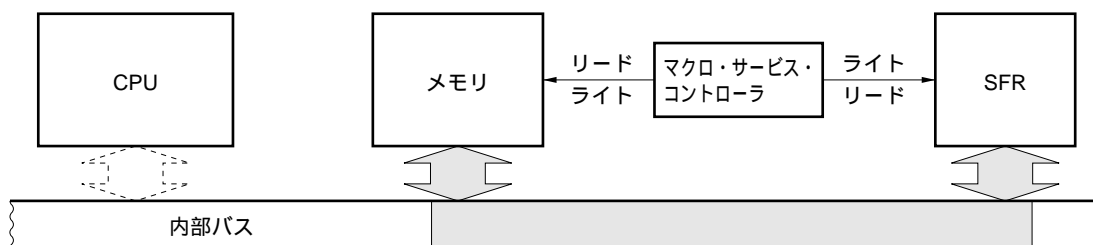
マクロ・サービスは、メモリと特殊機能レジスタ（SFR）間のデータ転送を、CPUを介さずに行う機能です。

マクロ・サービスでは、割り込みが受け付けられるとCPUの実行を一時中断し、ハードウェア的にデータ転送を行います。また、マクロ・サービスはCPUを介さずに行われるため、プログラム・カウンタ（PC）、プログラム・ステータス・ワード（PSW）などのCPUステータスを退避、復帰する必要がありません。

したがって、割り込みの要求に対して高速に応答することが可能です。

図4 - 5に、マクロ・サービスの概要図を示します。

図4 - 5 マクロ・サービスの概要図



(1) マクロ・サービスの種類

マクロ・サービスの種類は、タイプA、タイプB、タイプCおよびカウンタ・モードの4タイプあります。

次に、各タイプのマクロ・サービスの概要を説明します。

(a) マクロ・サービス・タイプA

割り込み要求の発生ごとに1バイトまたは1ワードのデータを特殊機能レジスタ（SFR）とメモリとの間で行い、指定された回数のデータ転送を行うと、ベクタ割り込み要求を発生します。

転送の対象となるメモリは、内部RAM（ショート・ダイレクト・アドレッシング1領域）に限定され、最大256バイト転送できます。

指定方法が簡単で、小容量の高速のデータ転送に向いています。

(b) マクロ・サービス・タイプB

タイプAと同様に、割り込み要求の発生ごとに1バイトまたは1ワードのデータを特殊機能レジスタ（SFR）とメモリとの間で行い、指定された回数のデータ転送を行うとベクタ割り込み要求を発生します。

転送の対象となるメモリは、1 Mバイト全空間使用でき、最大64 Kバイト転送できます。

タイプAの汎用タイプで、転送データ量が多い場合に向いています。

(c) マクロ・サービス・タイプC

割り込み要求の発生ごとに、2つのSFRへメモリからデータ転送します。指定された回数のデータ転送を行うと、ベクタ割り込み要求を発生します。

タイプCのマクロ・サービスは一度の割り込み要求で2箇所へのデータ転送を行うことのほかに、出力データのリング制御（繰り返し出力制御）やコンペア・レジスタとデータを自動的に加算する機能を付加して使用することもできます。

転送の対象となるメモリは、1 Mバイト全空間使用でき、各々最大64 Kバイト転送できます。

(d) カウンタ・モード

割り込み要求の発生ごとに、メモリ中に設定されたカウンタをデクリメントします。

表4 - 5 に、マクロ・サービス・タイプ別の機能の特徴を示します。

表4 - 5 マクロ・サービス・タイプ別の機能の特徴

マクロ・サービス ・タイプ 項目	タイプA	タイプB	タイプC	カウンタ・ モード
データ転送タイミング	割り込み要求発生ごとに転送する。	割り込み要求発生ごとに転送する。	割り込み要求発生ごとに転送する。	データ転送なし
データ転送方向	メモリ SFR	メモリ SFR	メモリ SFR	なし
データ転送バイト数	1バイトまたは2バイト	1バイトまたは2バイト	・MPTで指定されるタイプ用1バイトまたは2バイト ・MPDで指定されるデータ用1バイトのみ	なし
対象SFR	任意のSFR 1つ	任意のSFR 1つ	任意のSFR 2つ	なし
転送の対象となるメモリ	最大256バイトまで LOCATION命令によって指定された内部RAM領域に限定される LOCATION 0H 0FE00H-0FEFFH LOCATION 0FH FFE00H-FFEFFH	最大64 Kバイトまで 1 Mバイトの任意のアドレス空間	最大64 Kバイトまで (SFR 1つに対して) 1 Mバイトの任意のアドレス空間	なし
リング制御	制御不可	制御不可	制御可能	制御不可
自動加算制御	制御不可	制御不可	制御可能	制御不可
用途	小容量の高速データの転送に使用	タイプAの汎用タイプで大容量の高速データの転送に使用	ステッピング・モータの制御などに使用	割り込み要求回数のカウンタ

(2) マクロ・サービスで設定する制御レジスタとRAM領域について

マクロ・サービスは、マクロ・サービス・コントロール・ワードのマクロ・サービス・モード・レジスタの設定値により、マクロ・サービスのタイプや転送方向などを決定します。

さらに、マクロ・サービス・チャンネル・ポインタで指定されるマクロ・サービス・チャンネルの内容をマクロ・サービスのタイプに従って転送処理を行います。

マクロ・サービス・チャンネルは、転送回数を記憶するマクロ・サービス・カウンタ(MSC)や転送先や転送元のポインタなどが配置されている内部RAM領域で、各マクロ・サービス・タイプにより構成が異なります。

次に、マクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルについて説明します。

(3) マクロ・サービス・コントロール・ワード

μPD784038, 784038Yサブシリーズのマクロ・サービス機能は、マクロ・サービス・モード・レジスタとマクロ・サービス・チャンネル・ポインタで制御します。

マクロ・サービス・モード・レジスタによってマクロ・サービス処理タイプを設定し、マクロ・サービス・チャンネル・ポインタによって転送回数や特殊機能レジスタ (SFR) の転送アドレスなどが配置されているマクロ・サービス・チャンネルのアドレスを指し示します。

マクロ・サービス・モード・レジスタとマクロ・サービス・チャンネル・ポインタは、マクロ・サービス・コントロール・ワードとして、各マクロ・サービス割り込み要因ごとに、図4 - 6に示すように、内部RAM領域の一部に割り付けられています。

マクロ・サービスを使用するときは、各マクロ・サービス割り込み要因ごとにマクロ・サービス・モード・レジスタとマクロ・サービス・チャンネル・ポインタの値を設定してください。

図4 - 6 マクロ・サービス・コントロール・ワードのアドレス (LOCATION 0の場合)

予約語	アドレス		要因
SPCHP	0FE31H	チャンネル・ポインタ	INTSPC ^注
SPMMD	0FE30H	モード・レジスタ	
STCHP2	0FE2FH	チャンネル・ポインタ	INTST2
STMMD2	0FE2EH	モード・レジスタ	
SRCHP2/CSICHP2	0FE2DH	チャンネル・ポインタ	INTSR2/INTCSI2
SRMMD2/CSIMMD2	0FE2CH	モード・レジスタ	
CSICHP	0FE29H	チャンネル・ポインタ	INTCSI
CSIMMD	0FE28H	モード・レジスタ	
STCHP	0FE27H	チャンネル・ポインタ	INTST
STMMD	0FE26H	モード・レジスタ	
SRCHP/CSICHP1	0FE25H	チャンネル・ポインタ	INTSR/INTCSI1
SRMMD/CSIMMD1	0FE24H	モード・レジスタ	
ADCHP	0FE21H	チャンネル・ポインタ	INTAD
ADMMD	0FE20H	モード・レジスタ	
PCHP5	0FE1FH	チャンネル・ポインタ	INTP5
PMMD5	0FE1EH	モード・レジスタ	
PCHP4	0FE1DH	チャンネル・ポインタ	INTP4
PMMD4	0FE1CH	モード・レジスタ	
CCHP30	0FE1BH	チャンネル・ポインタ	INTC30
CMMD30	0FE1AH	モード・レジスタ	
CCHP21	0FE19H	チャンネル・ポインタ	INTC21
CMMD21	0FE18H	モード・レジスタ	
CCHP20	0FE17H	チャンネル・ポインタ	INTC20
CMMD20	0FE16H	モード・レジスタ	
CCHP11	0FE15H	チャンネル・ポインタ	INTC11
CMMD11	0FE14H	モード・レジスタ	
CCHP10	0FE13H	チャンネル・ポインタ	INTC10
CMMD10	0FE12H	モード・レジスタ	
CCHP01	0FE11H	チャンネル・ポインタ	INTC01
CMMD01	0FE10H	モード・レジスタ	
CCHP00	0FE0FH	チャンネル・ポインタ	INTC00
CMMD00	0FE0EH	モード・レジスタ	
PCHP3	0FE0DH	チャンネル・ポインタ	INTP3
PMMD3	0FE0CH	モード・レジスタ	
PCHP2	0FE0BH	チャンネル・ポインタ	INTP2
PMMD2	0FE0AH	モード・レジスタ	
PCHP1	0FE09H	チャンネル・ポインタ	INTP1
PMMD1	0FE08H	モード・レジスタ	
PCHP0	0FE07H	チャンネル・ポインタ	INTP0
PMMD0	0FE06H	モード・レジスタ	

注 μPD784038Yサブシリーズのみ

次に、マクロ・サービス・コントロール・ワード内のマクロ・サービス・モード・レジスタと、マクロ・サービス・チャンネルについて説明します。

(a) マクロ・サービス・モード・レジスタ

マクロ・サービス・モード・レジスタは、マクロ・サービスの動作を指定する8ビットのレジスタです。

マクロ・サービス・コントロール・ワードの一部として内部RAM上に書いておきます。

図4 - 7に、マクロ・サービス・モード・レジスタのフォーマットを示します。

図4-7 マクロ・サービス・モード・レジスタのフォーマット(1/2)

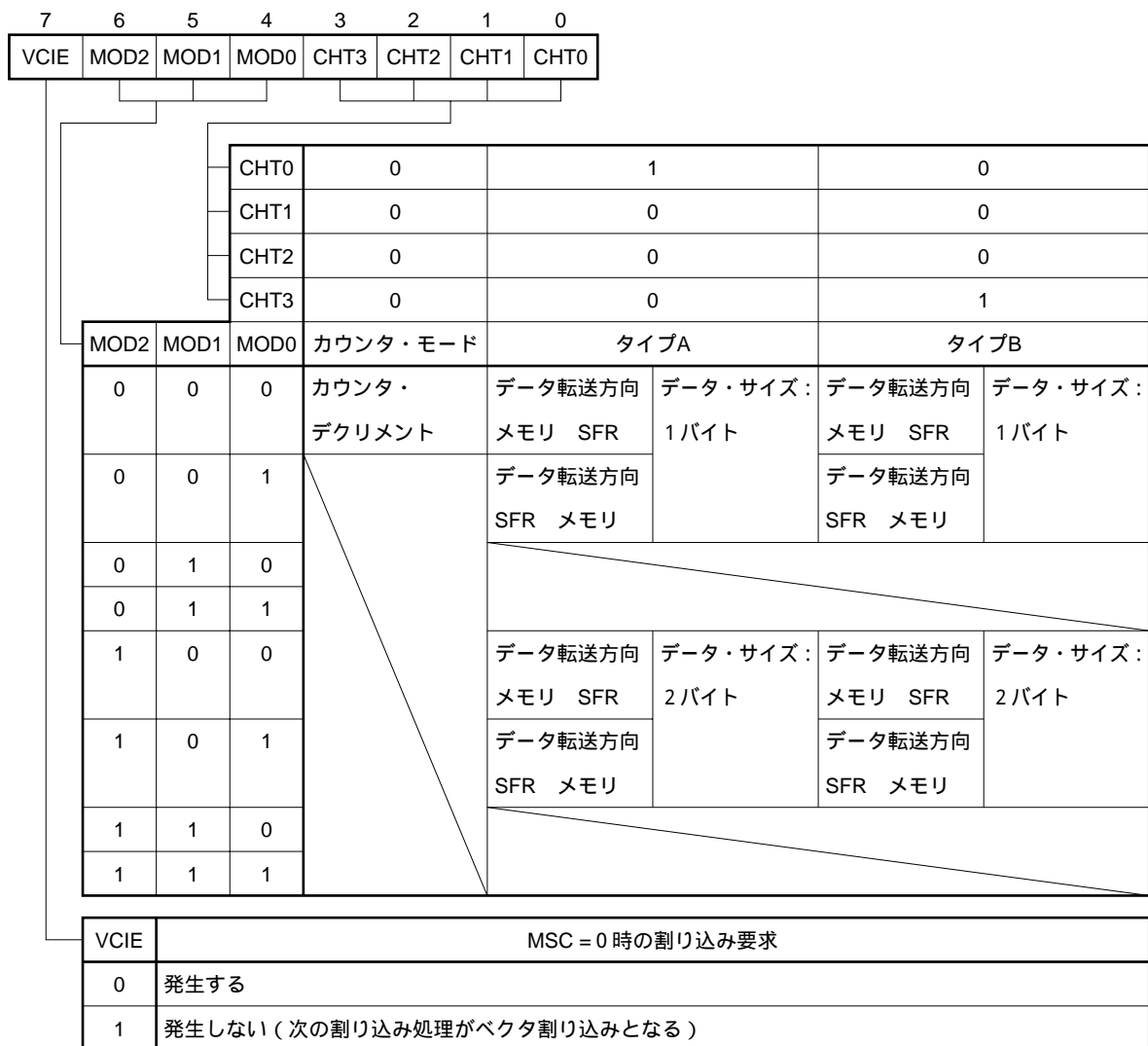


図4-7 マクロ・サービス・モード・レジスタのフォーマット(2/2)



(b) マクロ・サービス・チャンネル・ポインタ

マクロ・サービス・チャンネル・ポインタは、転送回数や特殊機能レジスタ（SFR）の転送アドレスなどが配置されている、マクロ・サービス・チャンネルのアドレスを指定するポインタです。

マクロ・サービス・チャンネル・ポインタには、マクロ・サービス・チャンネルの最上位アドレスの下位8ビットを設定します。

(4) マクロ・サービス・チャンネル

マクロ・サービス・チャンネルは、転送回数を記憶するマクロ・サービス・カウンタ（MSC）や転送先や転送元のポインタなどが配置されている内部RAM領域です。

マクロ・サービス・チャンネルは、「LOCATION」の設定により異なり、次に示す内部RAM領域に配置します。

LOCATION 0 命令実行時：0FE00H - 0FEFFH

LOCATION 0FH 命令実行時：FFE00H - FFEFFH

次に、各マクロ・サービスごとのマクロ・サービス・チャンネルの構成を示します。

(a) マクロ・サービス・タイプAのマクロ・サービス・チャンネルの構成

表4-6に、マクロ・サービス・タイプAのマクロ・サービス・チャンネル構成を示します。

表4-6 マクロ・サービス・タイプAのマクロ・サービス・チャンネル構成

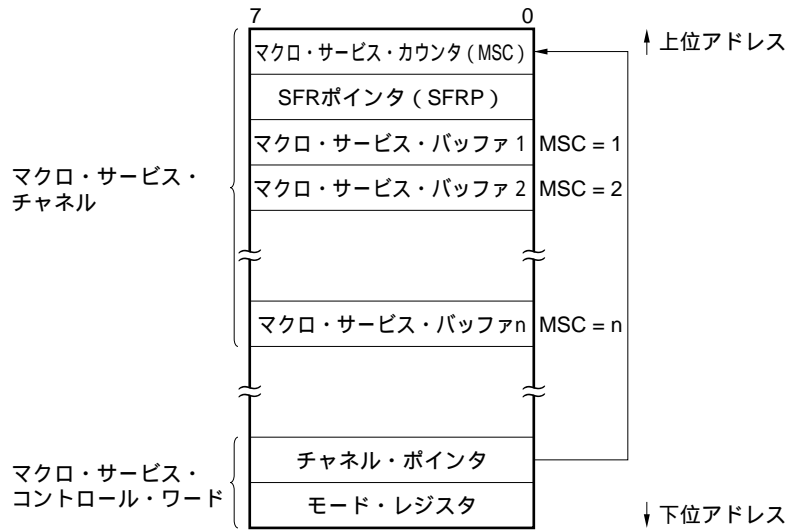
構 成	用 途	ビット数
マクロ・サービス・カウンタ（MSC）	データ転送の回数を指定します。	8ビット
SFRポインタ（SFRP）	転送先または転送元となるSFRのアドレスの下位8ビットを書き込みます。	8ビット

マクロ・サービス・タイプAでは、転送先となるRAM領域は、マクロ・サービス・チャンネル内に含まれます。

図4-8に、マクロ・サービス・タイプAのマクロ・サービス・チャンネルの配置を示します。

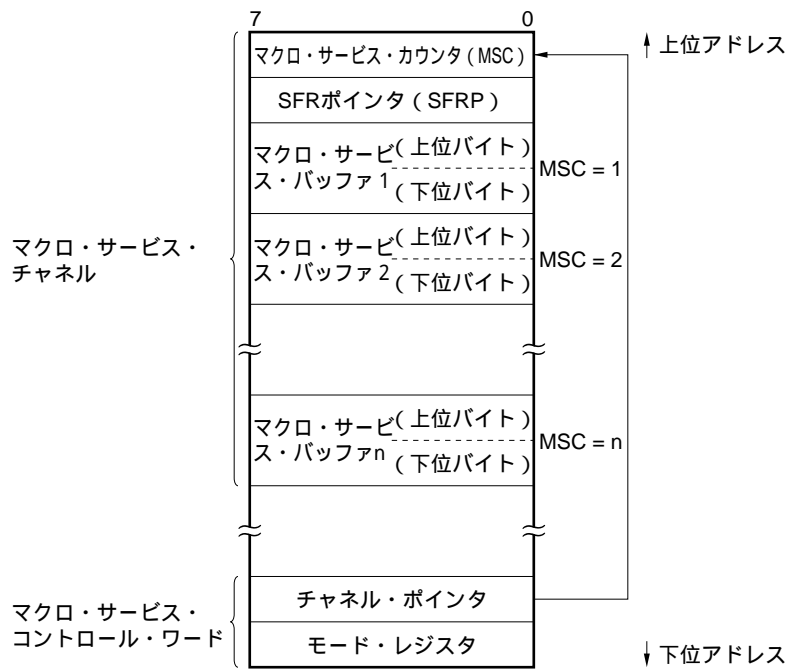
図4 - 8 マクロ・サービス・タイプAのマクロ・サービス・チャンネル配置

(a) 1バイト転送時



$$\text{マクロ・サービス・バッファ・アドレス} = (\text{チャンネル・ポインタ}) - (\text{マクロ・サービス・カウンタ}) - 1$$

(b) 2バイト転送時



$$\text{マクロ・サービス・バッファ・アドレス} = (\text{チャンネル・ポインタ}) - (\text{マクロ・サービス・カウンタ}) \times 2 - 1$$

(b) マクロ・サービス・タイプBのマクロ・サービス・チャンネルの構成

表4-7に、マクロ・サービス・タイプBのマクロ・サービス・チャンネル構成を示します。

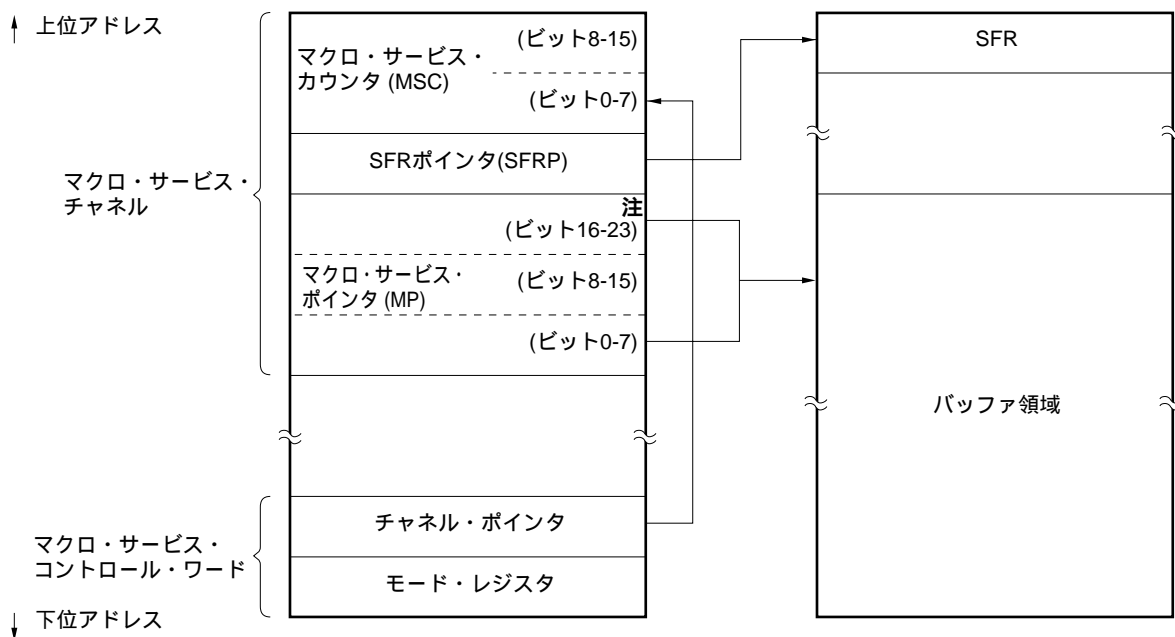
表4-7 マクロ・サービス・タイプBのマクロ・サービス・チャンネル構成

構成	用途	ビット数
マクロ・サービス・カウンタ (MSC)	データ転送の回数を指定します。	16ビット
SFRポインタ (SFRP)	転送先または転送元となるSFRのアドレスの低位8ビットを書き込みます。	8ビット
マクロ・サービス・ポインタ (MP)	転送先または転送元になる1 Mメモリ空間のデータ・バッファ領域を示します。	20ビット

マクロ・サービス・タイプBでは、転送元または転送先となるメモリ領域は、1 Mバイト全空間のメモリが対象となります。

図4-9に、マクロ・サービス・タイプBのマクロ・サービス・チャンネルの配置を示します。

図4-9 マクロ・サービス・タイプBのマクロ・サービス・チャンネルの配置



マクロ・サービス・バッファ・アドレス = マクロ・サービス・ポインタ

注 ビット20-23は必ず0にしてください。

(c) マクロ・サービス・タイプCのマクロ・サービス・チャンネルの構成

表4 - 8に、マクロ・サービス・タイプCのマクロ・サービス・チャンネル構成を示します。

表4 - 8 マクロ・サービス・タイプCのマクロ・サービス・チャンネル構成

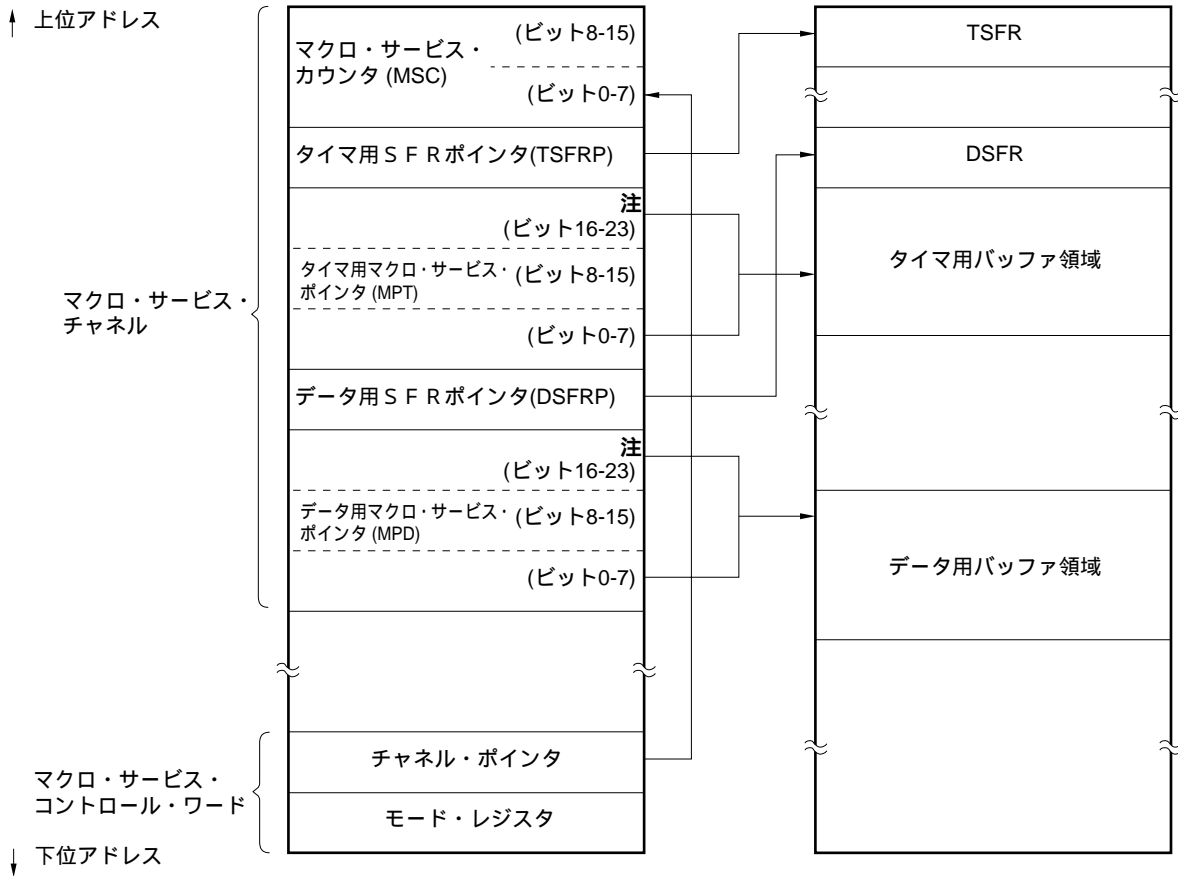
構 成	用 途	ビット数
マクロ・サービス・カウンタ (MSC)	データ転送の回数を指定します。	16ビット
タイマ用SFRポインタ (TSFRP)	転送先となるSFRのアドレスの下位8ビットを書き込みます。	8ビット
タイマ用マクロ・サービス・ポインタ (MPT)	主にタイマ/カウンタのコンペア・レジスタに転送または加算する1 Mメモリ空間内のデータ・バッファ領域を示します。	20ビット
データ用SFRポインタ (DSFRP)	転送先となるSFRのアドレスの下位8ビットを書き込みます。	8ビット
データ用マクロ・サービス・ポインタ (MPD)	リアルタイム出力ポートなどへ転送する1 Mメモリ空間のデータ・バッファ領域を示します。	20ビット
モジュロ・レジスタ (MR)	リング制御を使用するとき繰り返しのパターン数を指定します。	8ビット
リング・カウンタ (RC)	リング制御を使用するときのパターン内のステップを保持します。	8ビット

マクロ・サービス・タイプCでは、転送元となるメモリ領域は、1 Mバイト全空間のメモリが対象となります。

図4 - 10に、マクロ・サービス・タイプCのマクロ・サービス・チャンネルの配置を示します。

図4 - 10 マクロ・サービス・タイプCのマクロ・サービス・チャンネルの配置 (1/2)

(a) リング制御なし

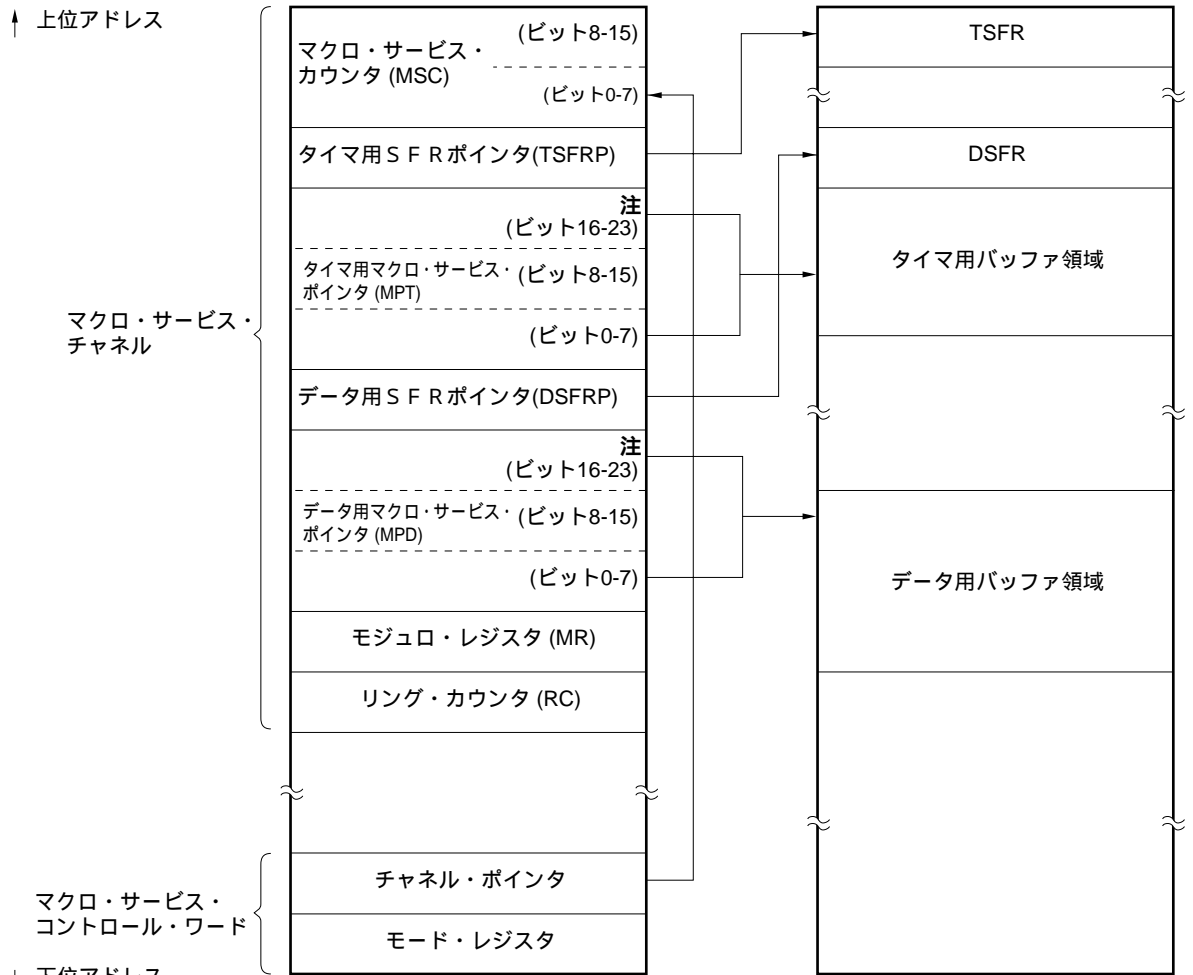


マクロ・サービス・バッファ・アドレス = マクロ・サービス・ポインタ

注 ビット20-23は必ず0にしてください。

図4-10 マクロ・サービス・タイプCのマクロ・サービス・チャンネルの配置 (2/2)

(b) リング制御あり



マクロ・サービス・バッファ・アドレス=マクロ・サービス・ポインタ

注 ビット20-23は必ず0にしてください。

(5) マクロ・サービス・タイプCの付加機能説明

タイプCのマクロ・サービスは、基本的なデータ転送に加えて、バッファ領域の圧縮とソフトウェアの負担を軽減するために、次の機能を付加することができます。

次の機能は、マクロ・サービス・コントロール・ワードのモード・レジスタで指定します。

(a) タイマ用マクロ・サービス・ポインタの更新

タイマ用のマクロ・サービス・ポインタ (MPT) を保持するか、インクリメント/デクリメントするか選択できます。

(b) データ用マクロ・サービス・ポインタの更新

データ用のマクロ・サービス・ポインタ (MPD) をインクリメント/デクリメントするか選択できます。

(c) 自動加算

タイマ用のマクロ・サービス・ポインタ (MPT) でアドレスされるデータと現在のコンペア・レジスタの値を加算して、加算結果をコンペア・レジスタへ転送します。

自動加算を指定しない場合は、タイマ用のマクロ・サービス・ポインタ (MPT) でアドレスされるデータをコンペア・レジスタに転送するだけです。

(d) リング制御

あらかじめ指定された長さの出力データ・パターンを、自動的に繰り返し出力します。

(6) マクロ・サービス終了時の動作

マクロ・サービスは、他のプログラム実行中に指定された回数の処理を行います。指定された回数の処理を行うと (マクロ・サービス・カウンタ (MSC) が0になると)、マクロ・サービスは終了します。

マクロ・サービスが終了すると、マクロ・サービス許可フラグはハードウェアで自動的にクリア (0) され、割り込み処理形態の指定が、コンテキスト・スイッチング許可フラグにより次のように変わります。

- ・コンテキスト・スイッチング許可フラグが禁止 (0) : ベクタ割り込み
- ・コンテキスト・スイッチング許可フラグが許可 (1) : コンテキスト・スイッチング

また、終了時の動作には2通りあり、マクロ・サービスごとのマクロ・サービス・モード・レジスタのビット7にあるVCIEフラグでその動作を指定します。

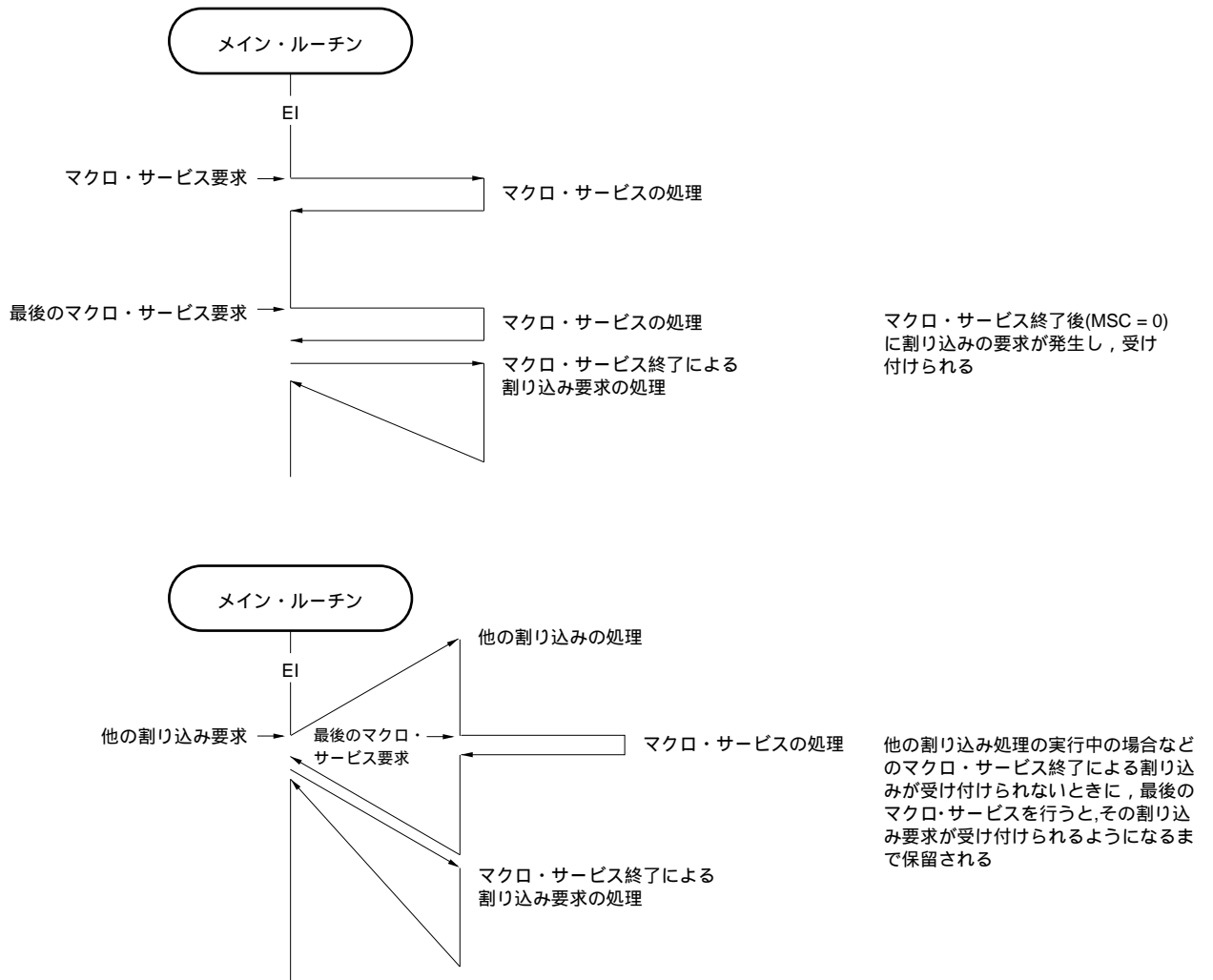
(a) VCIEフラグが0の場合

マクロ・サービスの終了後、ただちに割り込みの要求を発生するモードです。

このモードは、最後に行ったマクロ・サービスの処理によって、一連の動作が終了する場合などに使用します。

図4-11 VCIEフラグが0の場合のマクロ・サービス終了時の動作

(アシンクロナス・シリアル・インタフェースによるデータの受信例)

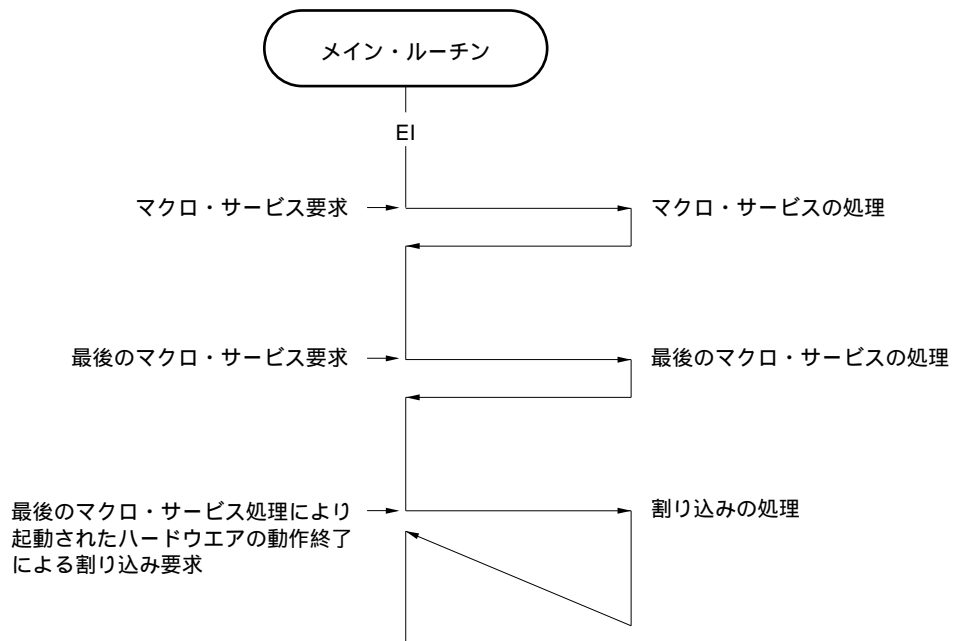


(b) VCIEフラグが1の場合

マクロ・サービスの終了後、割り込みの要求を発生しないモードです。

このモードは、最後に行ったマクロ・サービスの処理によって、最後の動作を開始させる場合などに使用します。

図4 - 12 VCIEフラグが1の場合のマクロ・サービス終了時の動作
(アシンクロナス・シリアル・インタフェースによるデータの送信例)



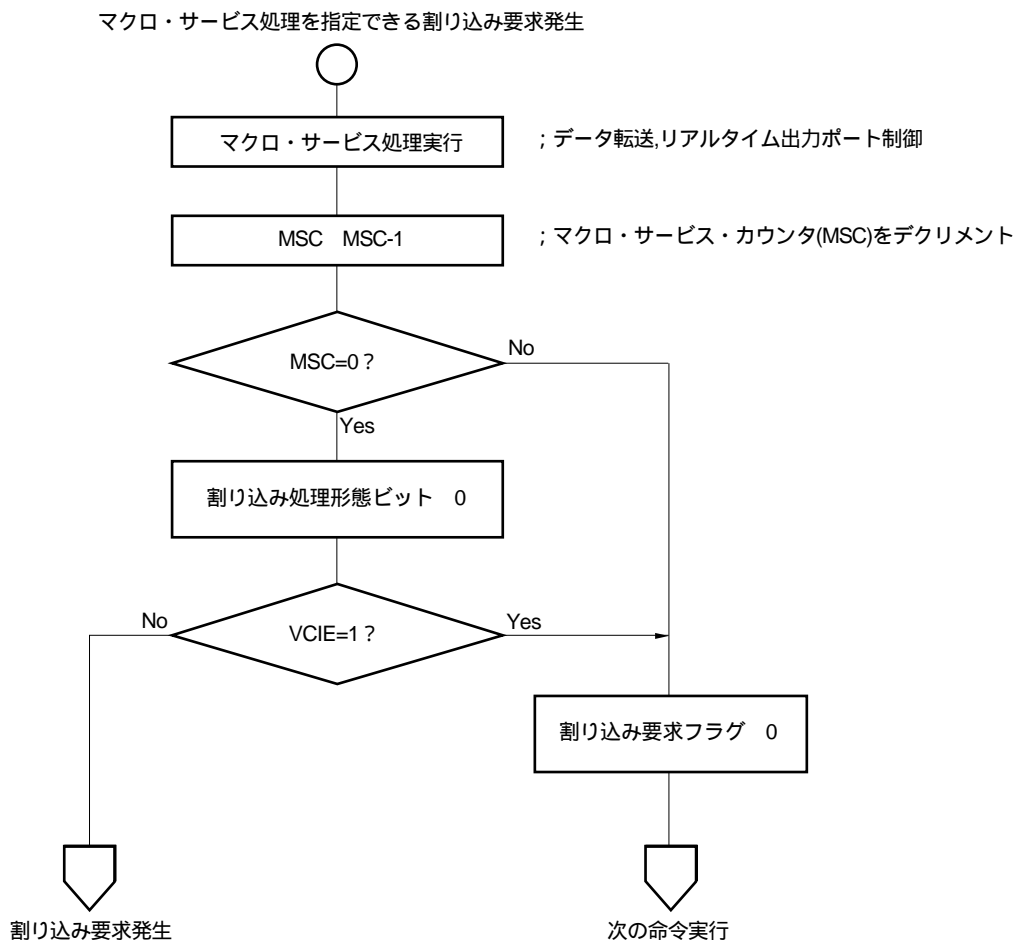
(7) マクロ・サービスの基本動作

マクロ・サービスを指定した割り込み要求は、割り込み要求許可フラグ(IEフラグ)の状態には影響されません。対象の割り込みマスク・フラグが解除(0)されているとき、割り込み制御レジスタのマクロ・サービス許可フラグを許可(1)することにより、マクロ・サービスの起動を許可します。

割り込み禁止状態、また割り込み処理プログラム実行中でもマクロ・サービスは実行されます。

図4-13に、マクロ・サービスの基本処理シーケンスを示します。

図4-13 マクロ・サービスの基本処理シーケンス



次に、マクロ・サービス・タイプごとのデータ転送処理フローを示します。

図4-14 マクロ・サービス・タイプAのデータ転送処理フロー

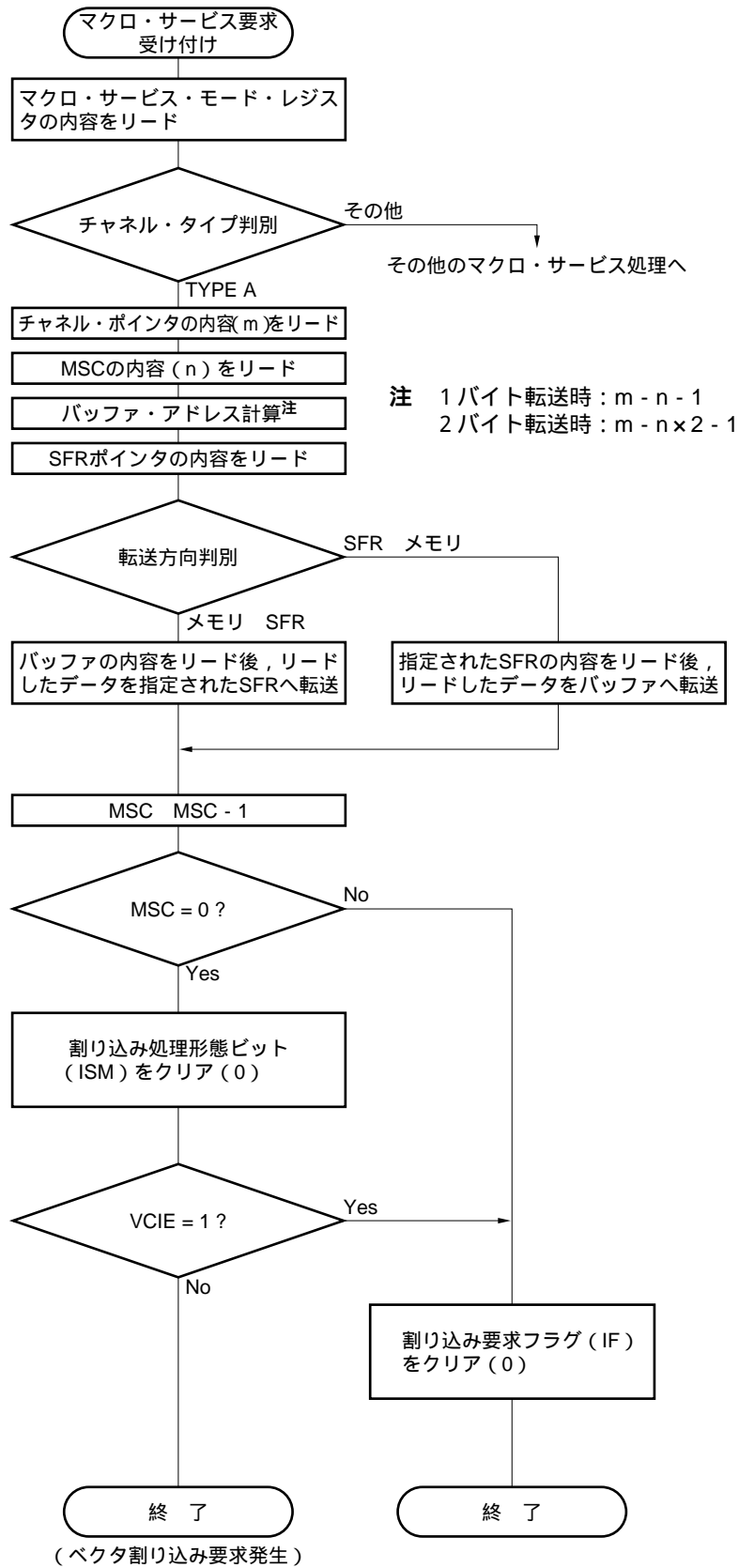


図4-15 マクロ・サービス・タイプBのデータ転送処理フロー

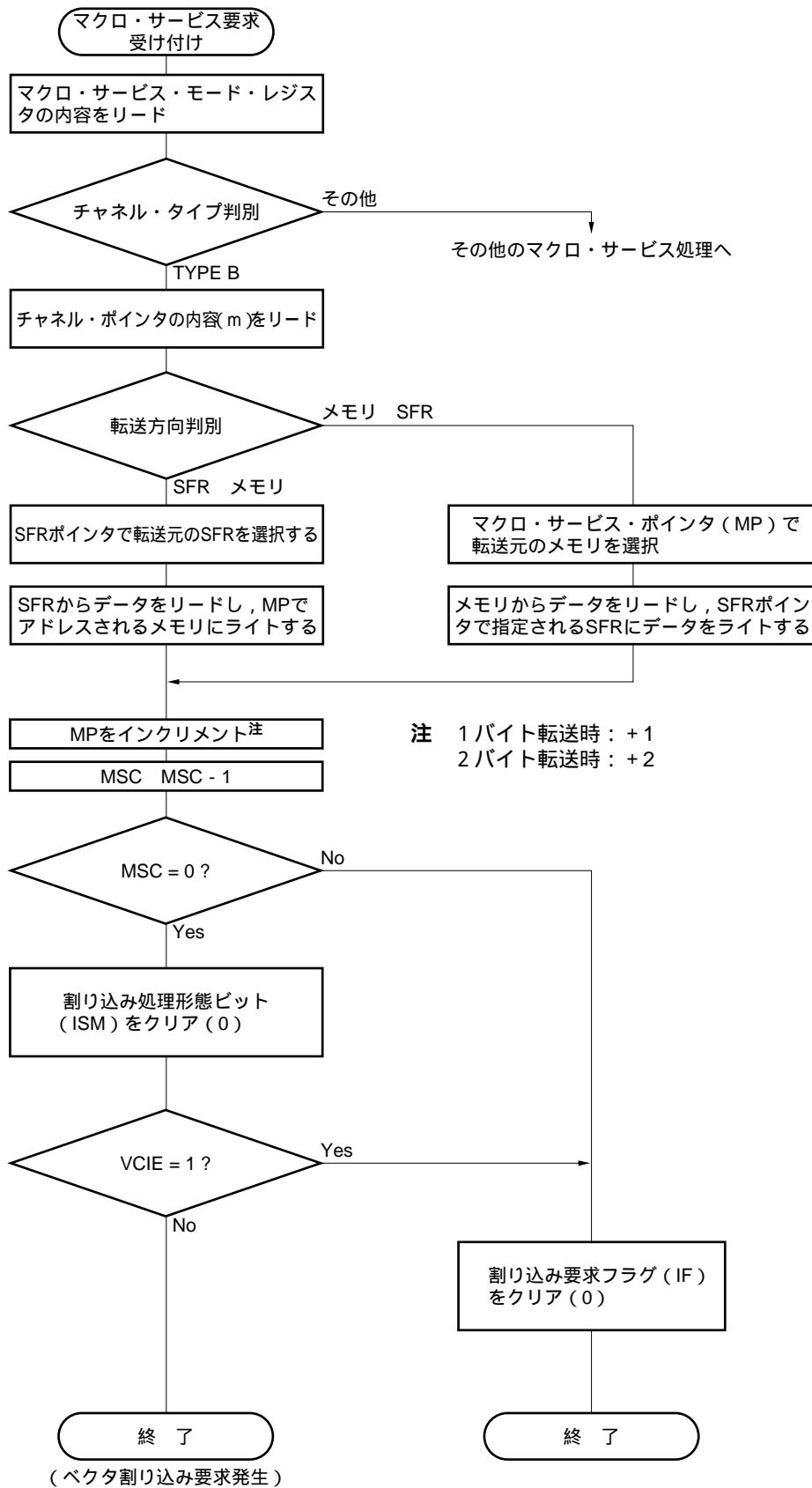
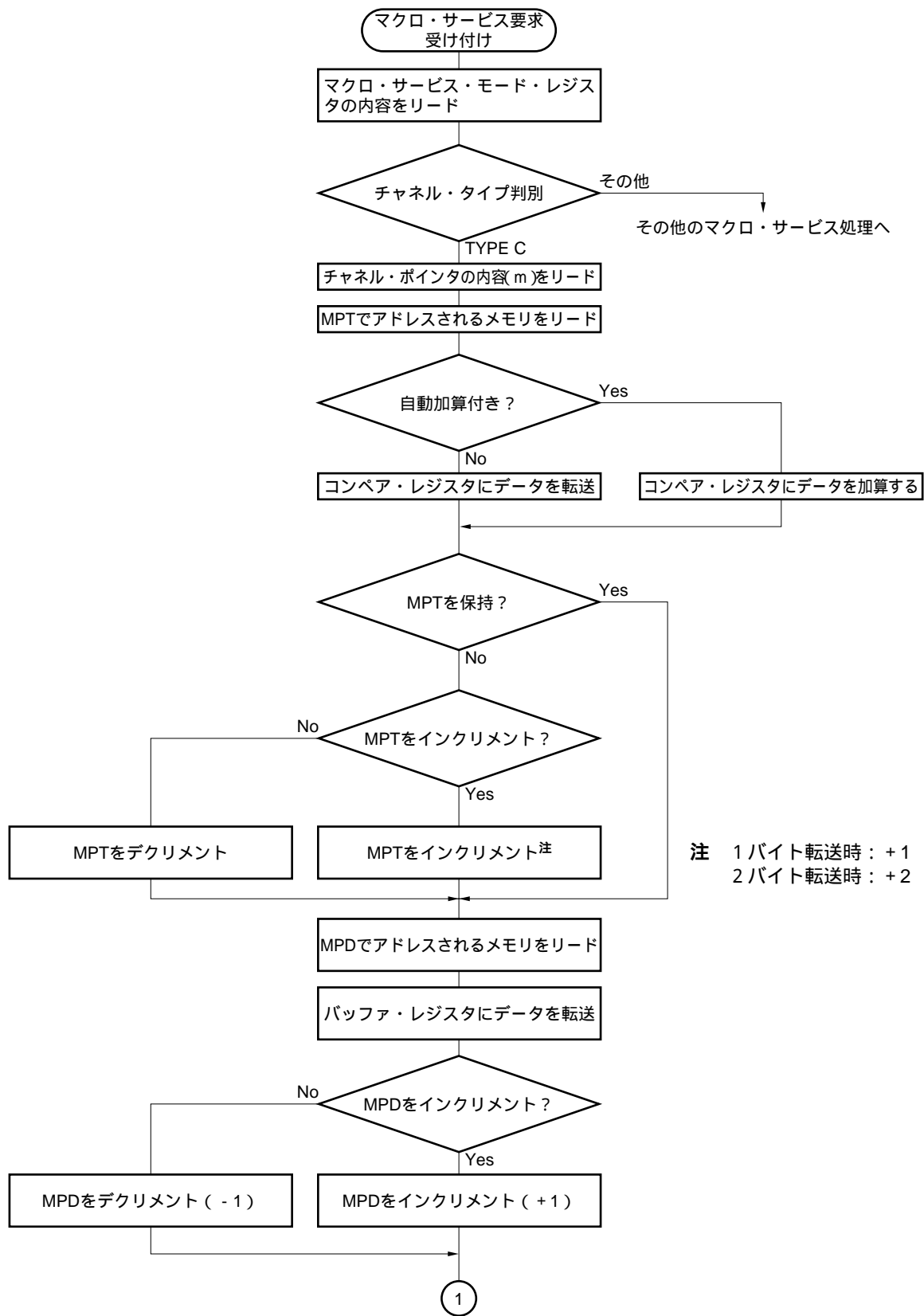
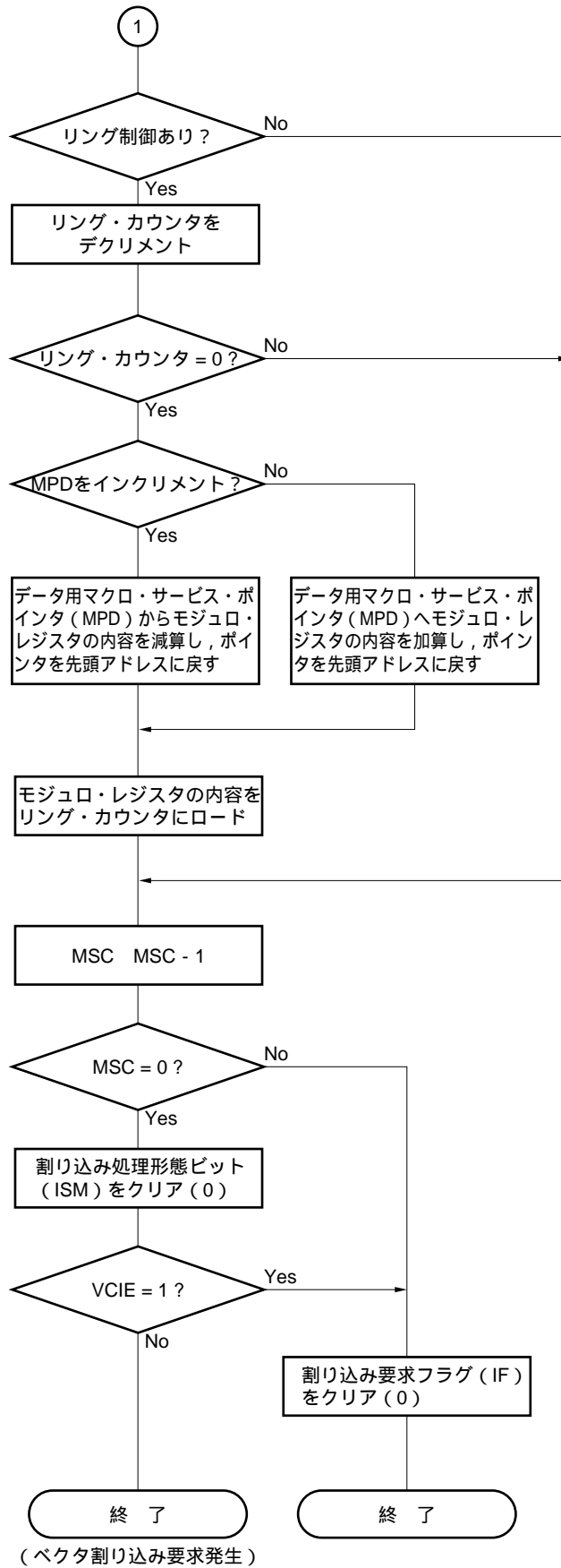


図4 - 16 マクロ・サービス・タイプCのデータ転送処理フロー





4.2 ベクタ割り込みとコンテキスト・スイッチングのプログラム例

ここでは、同じ仕様において、ベクタ割り込みとコンテキスト・スイッチング機能使用時の割り込み処理のプログラムの違いを紹介します。

(1) 処理概要

(a) 仕様

UART受信終了割り込み要求により、ベクタ割り込み、またはコンテキスト・スイッチング機能を使用した割り込みによって、アシンクロナス・シリアル・インタフェース (UART) を用いて受信した15バイトのデータを、RAM領域のワーク領域内に転送するプログラム例を紹介します。

ベクタ割り込み、またはコンテキスト・スイッチング機能を使用した割り込みは、UARTで1バイト受信するごとに割り込み処理を行います。

(b) 使用周辺機能説明

ベクタ割り込みも、コンテキスト・スイッチング機能を使用した場合も、割り込み処理内のUARTの受信バッファからRAM領域のワーク領域に転送するプログラムは同じ処理になります。

異なるのは、割り込み要求の受け付け処理形態と、それに対するプログラム処理の設定のみです。

表4-9に、ベクタ割り込みとコンテキスト・スイッチング機能を使用した場合の割り込み処理プログラムの違いを示します。

表4-9 ベクタ割り込みとコンテキスト・スイッチング機能使用時のプログラム処理の違い

	ベクタ割り込み	コンテキスト・スイッチング機能
レジスタの退避	・レジスタ・バンクをプログラムにより切り替えるかあるいは、PUSH/POP命令でレジスタを退避する。	・あらかじめ設定したレジスタ・バンクに自動的に切り替わる。
割り込み処理用のイニシャライズ	・割り込みサービス・プログラムのアドレスをベクタ・テーブルに設定する。 ・割り込み要求に対する処理形態をベクタ割り込みに設定する。	・ベクタ・テーブルには、割り込み時に切り替えるレジスタ・バンクを設定し、割り込みサービス・プログラムのアドレスは、ベクタ・テーブルで指定したレジスタ・バンクのRP2レジスタに格納する。 ・割り込み要求に対する処理形態をコンテキスト・スイッチング機能に設定する。
割り込みからの復帰命令	RETI命令を使用	RETCS命令を使用

コンテキスト・スイッチング機能を使用すれば、ベクタ割り込みに比べてPUSH/POP命令やSEL (レジスタ・バンク切り替え) 命令により、レジスタ退避や切り替えの必要がなくなり、割り込み処理への分岐が速くなります。

(2) 使用RAM領域

本プログラムでは、表4 - 10に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

なお、使用RAM領域は、ベクタ割り込みもコンテキスト・スイッチング機能使用時も共通となります。

表4 - 10 ベクタ割り込み、コンテキスト・スイッチング機能使用時のプログラムで用いるRAM / フラグ

RAM / フラグ名称	用 途	バイト数	初期値
STRDTA	UARTで受信したデータを格納するRAM領域 受信データ転送回数分のRAM領域を確保する	15	-
SPINTA	受信データの書き込み先ポインタ	1	0H
RCVFLAG	受信完了判別フラグ RCVFLAG = 0...受信未完了 RCVFLAG = 1...受信完了	1ビット	0

(3) 使用レジスタ**(a) 汎用レジスタ**

【ベクタ割り込み、コンテキスト・スイッチング機能使用時に共通で使用する汎用レジスタ】

A, Bレジスタ

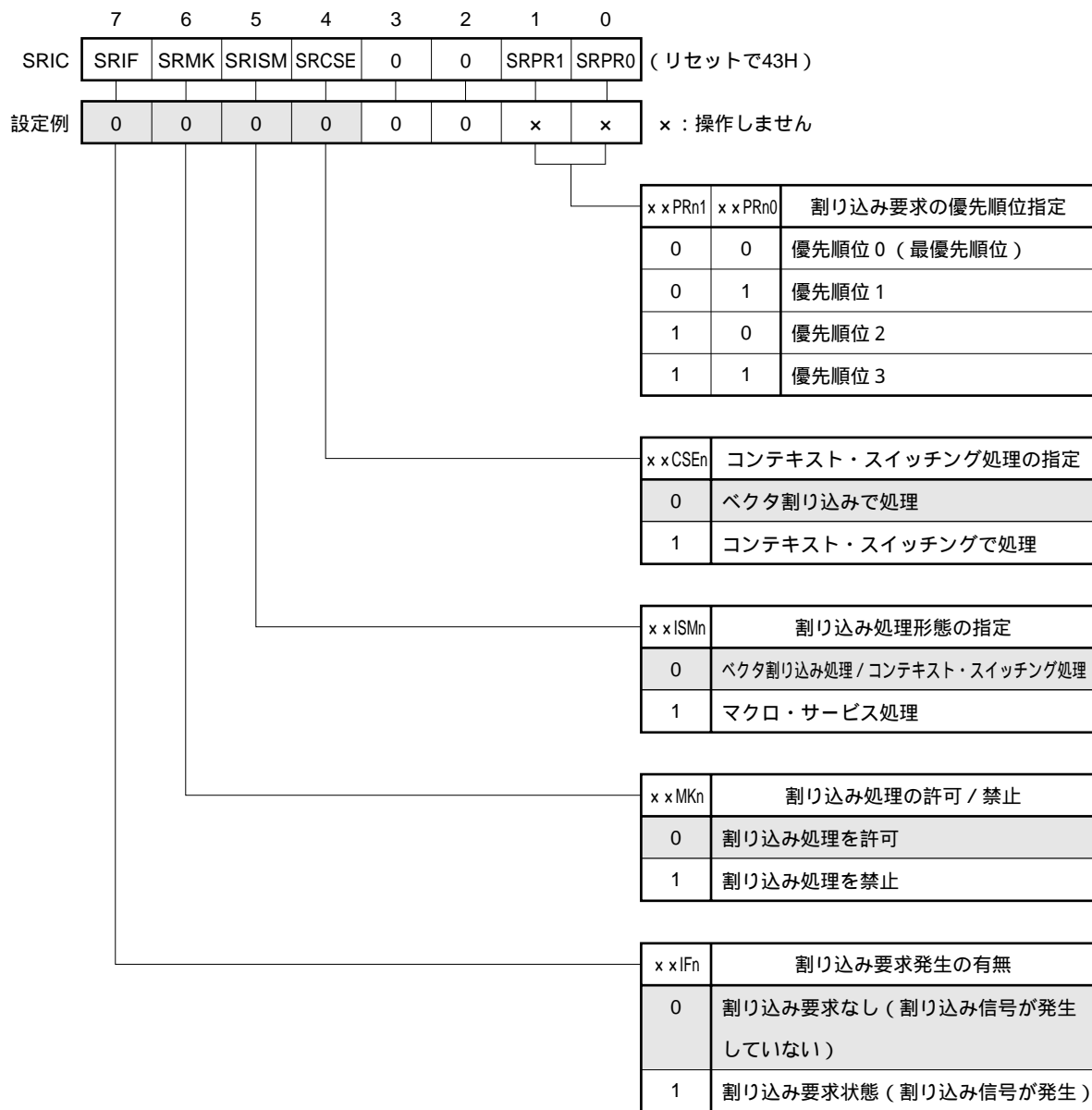
【コンテキスト・スイッチング機能使用時にのみ使用する汎用レジスタ】

RP2, RP3レジスタ

(b) 特殊機能レジスタ

【ベクタ割り込みの場合に使用する特殊機能レジスタの設定】

割り込み制御レジスタ



【コンテキスト・スイッチング機能使用時の場合に使用する特殊機能レジスタの設定】

割り込み制御レジスタ



(4) 入力方法

入力の引数は、ベクタ割り込みもコンテキスト・スイッチング機能使用時も共通となります。
下記に示すRAM領域に、次の内容を設定します。

SPINTA : 受信データの書き込み先ポインタで、イニシャライズ・データとして0を設定します。

下記に示すレーベルに、次の内容を定義します。

CNTNMA : UARTでデータの転送回数を定義します。

(5) 出力方法

出力の引数は、ベクタ割り込みもコンテキスト・スイッチング機能使用時も共通となります。
下記に示すRAM領域に、次のデータが格納されます。

STRDTA ~ STRDTA + 0EH : UART機能を用いて受信した受信データが格納されます。

下記に示すフラグ情報が設定されます。

RCVFLAG : 転送回数分の受信が完了すると、受信判別フラグ (RCVFLAG) がセット (1) されます。

(6) プログラム説明

ベクタ割り込みとコンテキスト・スイッチング機能使用時の割り込み処理プログラムで違う箇所を、アンダーラインで示します。その他の処理は同様になります。

【ベクタ割り込みの場合】

本仕様を実現するためには、次の2つの処理が必要です。

ベクタ割り込みイニシャライズ処理 [レーベル名称 : IR_INI]

(a) INTSR割り込み要求に対する処理形態をベクタ割り込みに設定します。

UART受信終了のベクタ割り込み処理 [レーベル名称 : INTSR]

(a) レジスタ・バンクをレジスタ・バンク1に切り替えます。

(b) 受信データを (STRDTA + (SPINTA)) に格納します。

(c) 受信データ書き込みポインタ (SPINTA) をインクリメントします。

(d) 受信データ書き込みポインタ = 転送回数 (CNTNMA) ならば受信データ書き込みポインタ (SPINTA) を0とし、(e) の処理に進みます。

受信データ書き込みポインタ 転送回数 (CNTNMA) ならば割り込み処理を終了します。

(e) 受信完了フラグ (RCVFLAG) をセット (1) します。

(f) RETI命令で復帰します。

注意 ベクタ・テーブルには、割り込みサービス・プログラムのアドレスを設定します。

【コンテキスト・スイッチング機能使用時の場合】

本仕様を実現するためには、次の2つの処理が必要です。

コンテキスト・スイッチング・イニシャライズ処理 [レーベル名称 : IR_INI]

(a) コンテキスト・スイッチング機能を使用した場合の割り込みプログラム分岐先アドレスをレジスタ・バンク1の指定エリア (RP2レジスタ) に格納します。

(b) INTSR割り込み要求に対する処理形態をコンテキスト・スイッチング機能に設定します。

UART受信終了のコンテキスト・スイッチング処理 [レーベル名称 : INTSR]

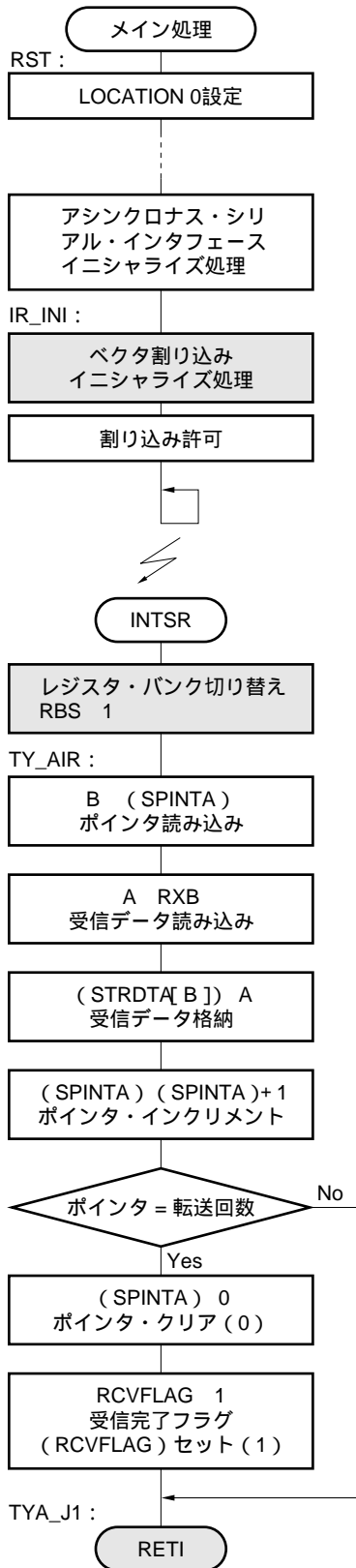
- (a) 受信データを (STRDTA + (SPINTA)) に格納します。
- (b) 受信データ書き込みポインタ (SPINTA) をインクリメントします。
- (c) 受信データ書き込みポインタ = 転送回数 (CNTNMA) ならば受信データ書き込みポインタ (SPINTA) を 0 とし , (d) の処理に進みます。
受信データ書き込みポインタ 転送回数 (CNTNMA) ならば割り込み処理を終了します。
- (d) 受信完了フラグ (RCVFLAG) をセット (1) します。
- (e) RETCS命令で復帰します (このとき , 次に割り込みを受け付けたときに実行する割り込みサービス・プログラムのアドレスを設定します) 。

注意 ベクタ・テーブルには , 割り込み時に切り替えるレジスタ・バンクを設定します。

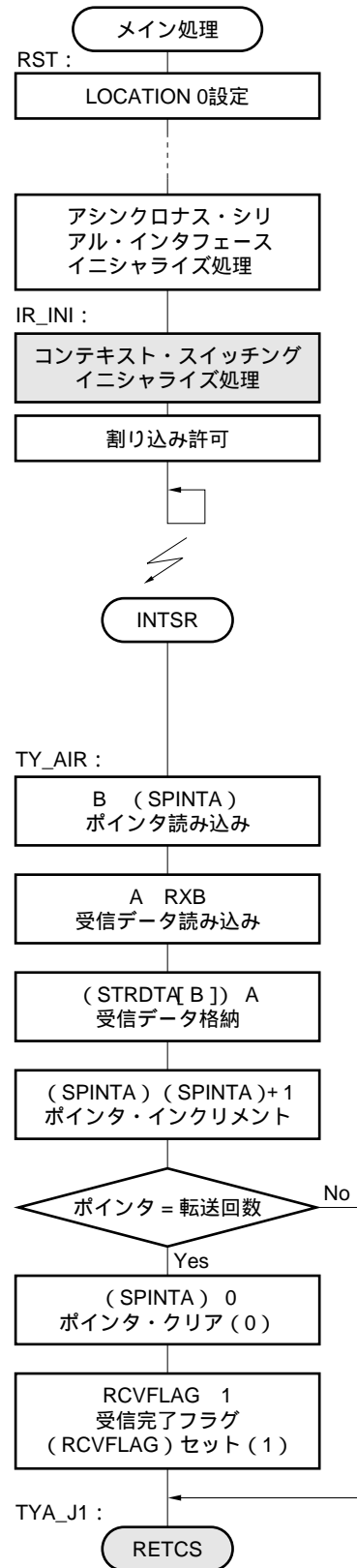
(7) フロー・チャート

ベクタ割り込みとコンテキスト・スイッチング機能使用時の処理の違う箇所を で示します。

【ベクタ割り込みの場合】



【コンテキスト・スイッチングの場合】



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

【ベクタ割り込み，コンテキスト・スイッチング機能使用時で共通に使用するレーベル】

- SPINTA : 受信データの書き込み先ポインタ
書き込み先のアドレスはSTRDTA + (SPINTA) で表されます。
- CNTNMA : 転送する回数
この設定値の回数の受信が完了すると，受信完了判別フラグ (RCVFLAG) がセット (1) されます。
- STRDTA : 受信データを転送するRAM領域の先頭アドレス
- RTS : ハンドシェーク・ライン
RTS = 0...BUSY解除状態
RTS = 1...BUSY状態
- RXB : UART機能でデータを受信するときの受信バッファ

本アプリケーション・ルーチンのプログラム・リスト

ベクタ割り込みとコンテキスト・スイッチング機能使用時のプログラム処理上で違う箇所を#####
というコメントで示します。

【ベクタ割り込みの場合】

```

      .
      .
;***** define vector table *****
RSTVT      CSEG      AT 0000H      ; RESET
           DW        RST

;
;***** ベクタ割り込みのベクタ・テーブルの設定*****
INTSRVT     CSEG     AT 0024H      ; INTSR
           DW        INTSR      ; #####

MAINS CSEG
RST:
      LOCATION      0H      ; specify LOCATION address
      .
      .
;*** asynchronous serial interface initialize
MOV     P3,#00100010B      ; P30,P31,P34,P35 inactive set
CLR1    PM3.5              ; P3.5=OUT
MOV     ASIM,#11101101B    ; ASIM initialize
MOV     BRGC,#01000100B    ; Baud rate = 9600bps(fxx=25MHz)
OR      PMC3,#00000011B    ; P30,P31 = Control port
SET1    PM3.4              ; P3.4=IN
           ;
           ;
MOV     SPINTA,#0          ;
CLR1    RCVFLAG           ;

IR_INI:
;***** ベクタ割り込みのインシャリス処理*****
MOV     SRIC,#00000011B    ; enable vector int. #####
EI
           ;

;*****
;
;***      MAIN      ***
;*****
SIL_MAIN:
BF      RCVFLAG,$SIL_MAIN ;
CLR1    RCVFLAG           ;
           .
           .
BR      SIL_MAIN          ;

```

```

.*****
;
;***      INTSR      ***
;*****
INTSR:
    *****      ハソクレジスタの切り替え      *****
        SEL      RB1                ; bank1 select          #####
        SET1     RTS                ; RTS<-1
;
;      *** receive data ***
;
;
TY_AIR:
    MOV      B,SPINTA                ; read store pointer
    MOV      A,RXB                   ; read receive data
    MOV      STRDTA[B],A             ; store receive data
    INC      SPINTA                  ; increment store pointer
    CMP      SPINTA,#LOW(CNTNMA)
    BNZ      $TYA_J1
    MOV      SPINTA,#0
    SET1     RCVFLAG                 ; set end flag
TYA_J1:
    CLR1     RTS                    ; RTS<-0
;*****      ハク割り込みからの復帰      *****
        RETI                        ;                      #####

```


【コンテキスト・スイッチングの場合】

```

      .
      .
;***** define vector table *****
RSTVT      CSEG      AT 0000H      ; RESET
           DW        RST

;***** コンテキスト・スイッチングのバンクレジスタの設定 *****
INTSRVT      CSEG      AT 00024H
           DW        1              ; INTSR          #####
                                           ; bank1 select      #####

MAINS CSEG
RST:
      LOCATION      0H              ; specify LOCATION address
      .
      .
;*** asynchronous serial interface initialize
MOV      P3,#00100010B              ; P30,P31,P34,P35 inactive set
CLR1     PM3.5                      ; P3.5=OUT
MOV      ASIM,#11101101B           ; ASIM initialize
MOV      BRGC,#01000100B           ; Baud rate = 9600bps(fxx=25MHz)
OR       PMC3,#00000011B           ; P30,P31 = Control port
SET1     PM3.4                      ; P3.4=IN
           ;
           ;
MOV      SPINTA,#0                  ;
CLR1     RCVFLAG                    ;

IR_INI:
;***** コンテキスト・スイッチングのイニシャライズ処理 *****
SEL      RB1                        ; bank1 select          #####
MOVW     RP2,#INTSR                 ;                      #####
SEL      B0                          ;                      #####
MOV      SRIC,#00010011B           ; context disposition
                                           designate          #####
EI
           ;

;*****
;***      MAIN      ***
;*****
SIL_MAIN:
BF       RCVFLAG,$SIL_MAIN         ;
CLR1     RCVFLAG                    ;
           .
           .
BR       SIL_MAIN                   ;

```

```
.*****
;
;***      INTSR      ***
;*****
INTSR:
    SET1      RTS                ; RTS<-1
;
;      *** receive data ***
;
TY_AIR:
    MOV      B,SPINTA           ; read store pointer
    MOV      A,RXB              ; read receive data
    MOV      STRDTA[B],A       ; store receive data
    INC      SPINTA             ; increment store pointer
    CMP      SPINTA,#LOW(CNTNMA)
    BNZ      $TYA_J1
    MOV      SPINTA,#0
    SET1     RCVFLAG            ; set end flag
TYA_J1:
    CLR1     RTS                ; RTS<-0

;***** コンテキスト・スイッチング* 割り込み機能からの復帰 *****
    RETCS    !INTSR                ; #####
```

4.3 マクロ・サービスのプログラム例

マクロ・サービス機能の3タイプを、次の3種類のプログラム例で紹介します。

- ・ 4.3.1 マクロ・サービス・タイプAのプログラム例
- ・ 4.3.2 マクロ・サービス・タイプBのプログラム例
- ・ 4.3.3 マクロ・サービス・タイプCのプログラム例

4.3.1 マクロ・サービス・タイプAのプログラム例

(1) 処理概要

(a) 仕様

マクロ・サービス・タイプAを使用し、アシンクロナス・シリアル・インタフェース(UART)を用いて受信した15バイトのデータを、RAM領域のワーク領域内に転送するプログラム例を紹介します(本プログラムでは、LOCATION 0命令実行時の例を示します)。

(b) 使用周辺機能説明

マクロ・サービス・タイプAを使用し、SFRポインタ(SFRP)で示されるUARTの受信バッファ(RXB)に取り込んだ1バイトの受信データをマクロ・サービス・チャンネル・ポインタとマクロ・サービス・カウンタ(MSC)で示されるRAM領域のワーク領域に転送します。15バイトのデータを転送終了後、ベクタ割り込みを発生します。

表4-11に、本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定を示します。

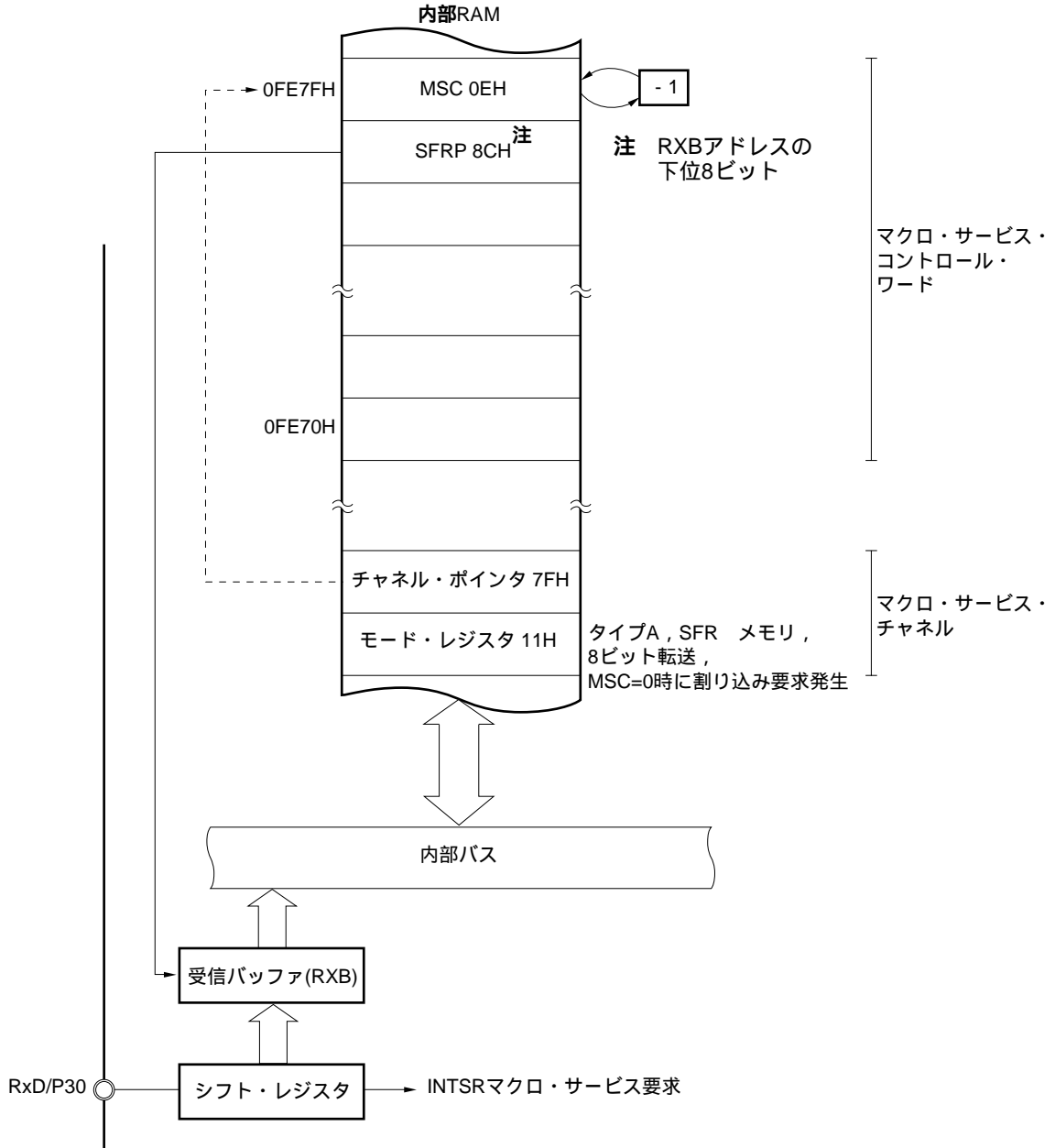
表4 - 11 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定

マクロ・サービス・コントロール・ワード領域	用 途	バイト数	設定値
マクロ・サービス・モード・レジスタ	マクロ・サービスの処理モードを次のように設定する。 マクロ・サービス・タイプ：タイプAに設定 データ転送方向 : SFR メモリ 転送データ・サイズ : 1バイト VCIE = 0 : MSC = 0のとき割り込み要求を発生	1	11H
マクロ・サービス・チャンネル・ポインタ	マクロ・サービス・カウンタ (MSC) のアドレスの下位 8 ビットを設定する。	1	7FH

マクロ・サービス・チャンネル領域	用 途	バイト数	設定値
マクロ・サービス・カウンタ (MSC)	UART受信データの転送回数の15回を設定する。	1	0EH
マクロ・サービスSFRポインタ (SFRP)	転送の対象となるUARTの受信バッファ (RXB) のSFRアドレスを設定する。	1	RXBの下位8ビットのアドレス
受信領域	UARTで受信したデータを格納するRAM領域。 マクロ・サービス・カウンタ分のワーク領域を確保する。	15	-

図4 - 17に、本プログラムのマクロ・サービス・タイプAの処理を行うときのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図を示します。

図4 - 17 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図
(マクロ・サービス・タイプA)



備考 図中のアドレスは、LOCATION 0 命令実行時の値です。

LOCATION 0FH命令実行時には、図中の値に0F0000Hを加えてください。

(2) 使用RAM領域

本プログラムでは、表4 - 12に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表4 - 12 マクロ・サービス・タイプAのプログラムで用いるRAM領域

フラグ名称	用 途	バイト数	初期値
RCVFLAG	受信完了判別フラグ。 RCVFLAG = 0...受信未完了 RCVFLAG = 1...受信完了	1ビット	0

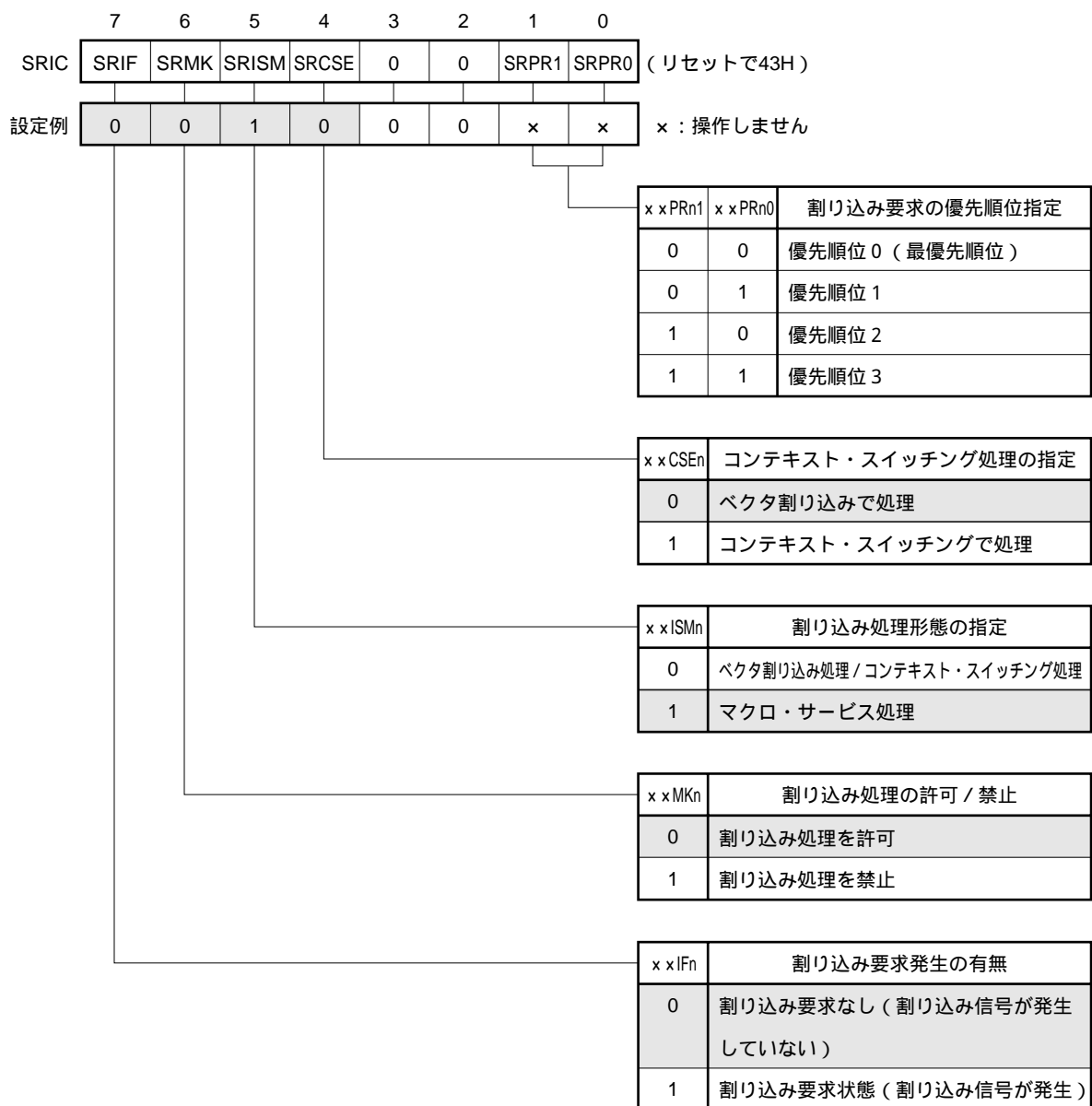
(3) 使用レジスタ

(a) 汎用レジスタ

なし

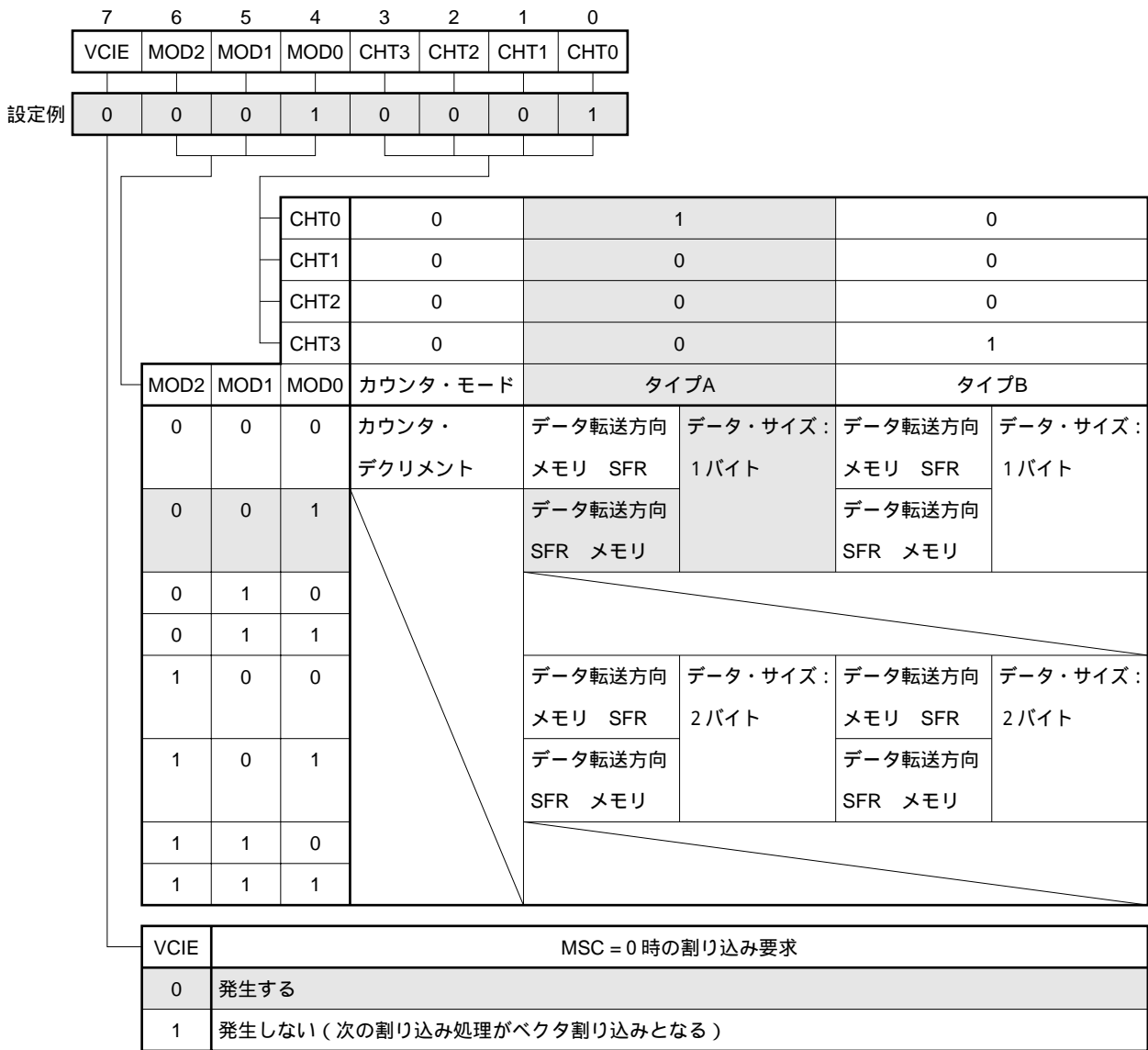
(b) 特殊機能レジスタ

割り込み制御レジスタ



(c) マクロ・サービス制御レジスタ

マクロ・サービス・モード・レジスタ (1/2)



マクロ・サービス・モード・レジスタ (2/2)



(4) 入力方法

下記に示すレーベルに、次の内容を定義します。

CNTNMA : マクロ・サービス・カウンタへの設定値を定義します。

UART受信データの転送回数を定義します。

CHASR : マクロ・サービス・チャンネル・ポインタへの設定値を定義します。

マクロ・サービス・カウンタ(MSC)のアドレスの下位8ビットを定義します。

(5) 出力方法

下記に示すRAM領域に、次の内容が格納されます。

BRSR ~ BRSR + 0EH : UART機能を用いて受信したデータが格納されます。

(6) プログラム説明

本仕様を実現するためには、次の2つの処理が必要です。

マクロ・サービス・イニシャライズ処理 [レーベル名称: TY_AMI]

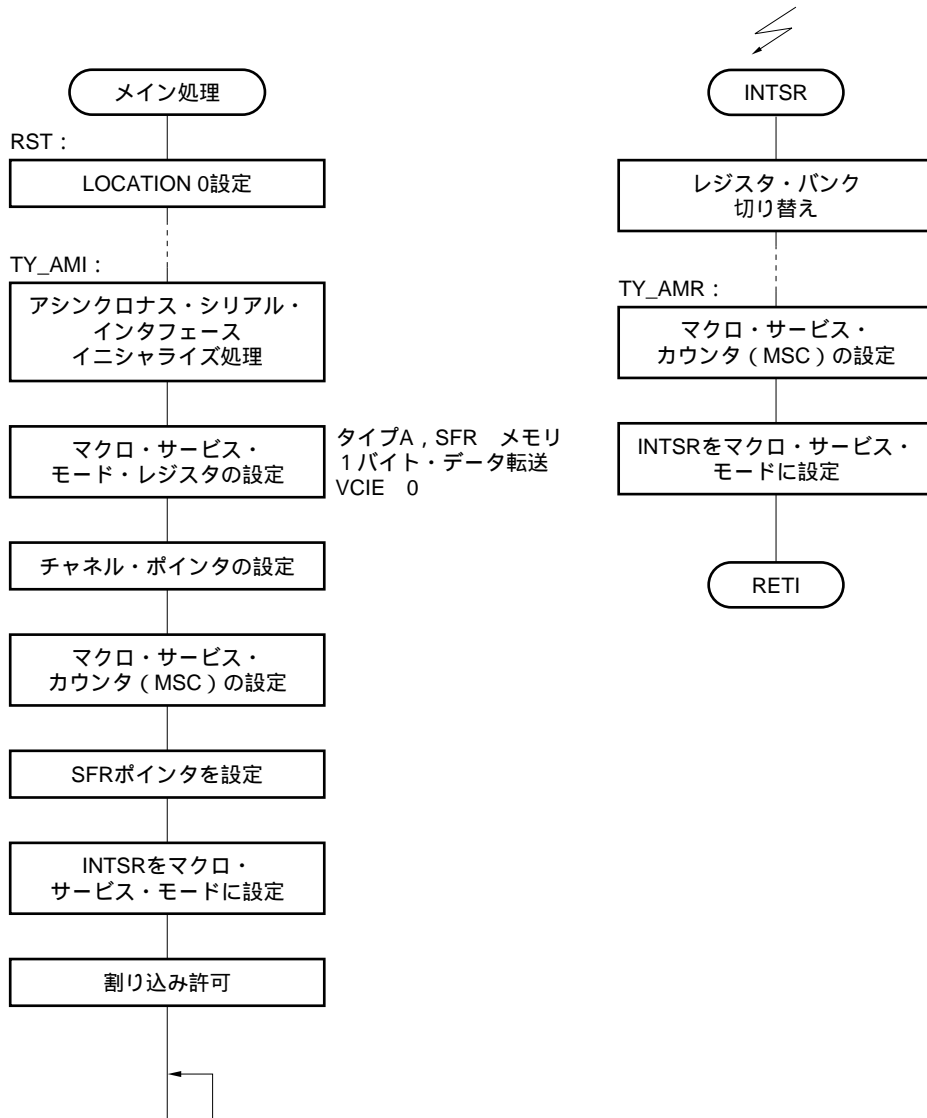
- (a) マクロ・サービス・モード・レジスタを設定します。
- (b) チャンネル・ポインタを設定します。
- (c) マクロ・サービス・カウンタ(MSC)にデータ転送回数(CNTNMA)を設定します。
- (d) SFRポインタを設定します。UART受信バッファ(RXB)のアドレス下位8ビットを設定します。
- (e) INTSR割り込み要求に対する処理形態をマクロ・サービス処理に指定します。
また同時に、割り込みマスクの許可(0)と、割り込み要求のクリア(1)を行います。
- (f) 割り込みを許可します。

マクロ・サービス完了割り込み処理 [レーベル名称: TY_AMR]

マクロ・サービスを再起動する処理を行っています。

- (a) マクロ・サービス・カウンタの値を再設定します。
- (b) マクロ・サービス完了割り込み時はINTSRの割り込み要求に対する処理形態は、ベクタ割り込みになっていますので、再び処理形態をマクロ・サービス処理とします。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

- MSMSR : マクロ・サービス・モード・レジスタ
- CHPSR : マクロ・サービス・チャンネル・ポインタ
- CHASR : チャンネル・ポインタへの設定アドレス
- MSCSR : マクロ・サービス・カウンタ
転送回数値を設定します。
- SFRPSR : マクロ・サービスSFRポインタ
対象となるSFRレジスタ・アドレスをこのエリアに指定します。
ここでは、UART受信バッファ (RXB) アドレスの下位8ビットを格納します。
- RXB : UART機能でデータ受信するときの受信バッファ
- CNTNMA : マクロ・サービス・カウンタへの設定値
ここで設定した回数分の転送が行われたのちにマクロ・サービス完了割り込みが発生します。
- RTS : ハンドシェイク・ライン
RTS = 0...BUSY解除状態
RTS = 1...BUSY状態

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
CNTNMA EQU 0FH ; store pointer data

;***** define work area *****
MCWSR DSEG AT 0FE24H
MSMSR: DS 1 ; INTSR macro service mode register
CHPSR: DS 1 ; INTSR macro service channel pointer

MCHSR DSEGAT 0FEB0H
BASR: DS CNTNMA ;
SFRPSR: DS 1 ; macro service sfr pointer for INTSR
MSCSR: DS 1 ; macro service counter for INTSR
CHASR EQU $-1 ; address of macro service counter for
; INTSR
      .
      .
;***** define vector table *****
RSTVT CSEG AT 0000H ; RESET
      DW RST
INTSRVT CSEG AT 0024H ; INTSR
      DW INTSR
MAINS CSEG
RST:
      LOCATION 0H ; specify LOCATION address
      .
      .
;
; asynchronous serial interface initialize
;
MOV P3,#00100010B ; P30,P31,P34,P35 inactive set
CLR1 PM3.5 ; P3.5=OUT
MOV ASIM,#11101101B ; ASIM initialize
MOV BRGC,#01000100B ; Baud rate = 9600bps(fxx=25MHz)
OR PMC3,#00000011B ; P30,P31 = Control port
SET1 PM3.4 ; P3.4=IN

;
; *** macro service initialize ***
;
TY_AMI:
MOV MSMSR,#00010001B ; type A(1 byte) sfr->memory VCIE 0
; set macro service mode register
MOV CHPSR,#LOW(CHASR) ; set macro service channel pointer
MOV MSCSR,#LOW(CNTNMA) ; set macro service counter
MOV SFRPSR,#LOW(RXB) ; set macro service sfr pointer
MOV SRIC,#00100011B ; initialize interrupt
EI

```

```
.*****
;
;***      MAIN      ***
;*****
;
SIL_MAIN:
    .
    .
    .
    BR    SIL_MAIN                ; メイン処理
    ;

.*****
;
;***      INTSR     ***
;*****
;
INTSR:
    SEL  RB1                      ;
    SET1 RTS                      ; RTS<-1
    .
    .
;   *** interrupt of complete macro-service ***
TY_AMR:
    MOV  MSCSR,#LOW(CNTNMA)       ; restore macro service counter
    SET1 SRISM                    ; set interrupt service mode
    CLR1 RTS                      ; RTS<-0
    RETI                          ;
```

4.3.2 マクロ・サービス・タイプBのプログラム例

(1) 処理概要

(a) 仕様

マクロ・サービス・タイプBを用いて、外部入力信号に同期して、P30-P37端子から入力した32バイトの平行データ、RAM領域のワーク領域に転送するプログラム例を紹介します。

外部入力信号との同期は、外部割り込み要求入力端子 (INTP4) を用います (本プログラムでは、LOCATION 0命令実行時の例を示します)。

(b) 使用周辺機能説明

マクロ・サービス・タイプBを使用し、外部割り込み要求入力端子 (INTP4) の外部入力信号のエッジに同期して、SFRポインタ (SFRP) が示すP3から取り込んだ1バイトのデータをマクロ・サービス・ポインタ (MP) が示すワーク領域の先頭アドレスから、マクロ・サービス・カウンタ (MSC) で設定された32回分の1バイトのデータを順次ワーク領域に転送します。32バイトの転送終了後、ベクタ割り込みを発生します。

表4-13に、本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定を示します。

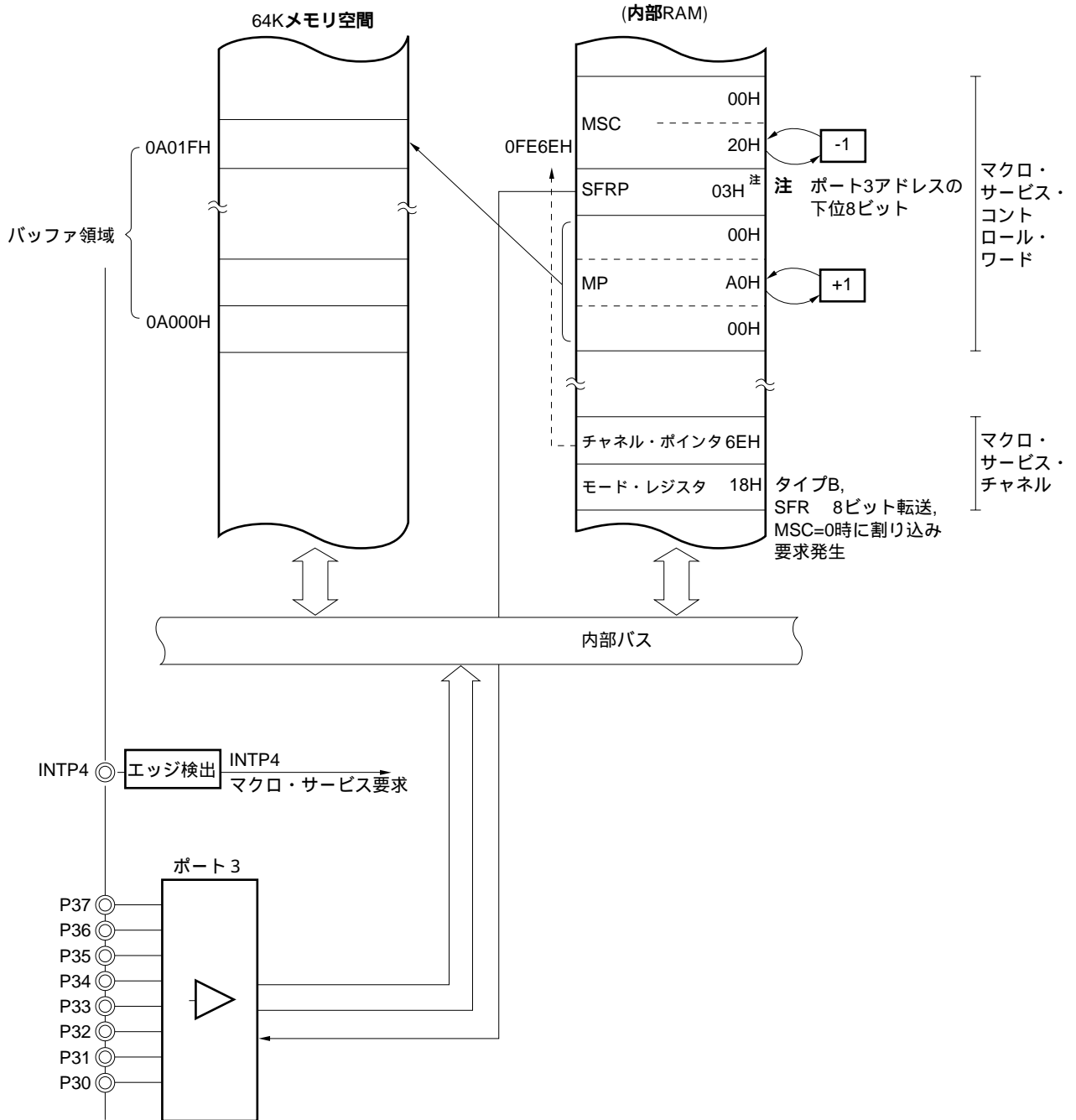
表4-13 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定

マクロ・サービス・コントロール・ワード領域	用途	バイト数	設定値
マクロ・サービス・モード・レジスタ	マクロ・サービスの処理モードを次のように設定する。 マクロ・サービス・タイプ：タイプBに設定 データ転送方向：SFR メモリ 転送データ・サイズ：1バイト VCIE = 0：MSC = 0のとき割り込み要求を発生	1	18H
マクロ・サービス・チャンネル・ポインタ	マクロ・サービス・カウンタ (MSC) のアドレスの下位8ビットを設定する。	1	6EH

マクロ・サービス・チャンネル領域	用途	バイト数	設定値
マクロ・サービス・カウンタ (MSC)	外部入力信号のエッジに同期して、P3から取り込む平行データの転送回数を32回に設定する。	2	0020H
マクロ・サービスSFRポインタ (SFRP)	転送の対象となるP3のSFRアドレスを設定する。	1	P3アドレスの下位8ビットのアドレス
マクロ・サービス・ポインタ (MP)	転送データを格納するワーク領域の先頭アドレスを設定する。	3	0A000H

図4-18に、本プログラムのマクロ・サービス・タイプBの処理を行うときのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図を示します。

図4-18 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図
(マクロ・サービス・タイプB)



備考 図中のマクロ・サービス・チャンネルのアドレスは、LOCATION 0命令実行時の値です。
LOCATION 0FH命令実行時には、図中の値に0F0000Hを加えてください。

(2) 使用RAM領域

本プログラムでは、表4 - 14に示すようにRAM領域を使用します。

表4 - 14 マクロ・サービス・タイプBのプログラムで用いるRAM領域

RAM名称	用 途	バイト数	初期値
STRDTB	外部入力信号のエッジに同期してP3から取り込んだデータを格納するRAM領域	32	-

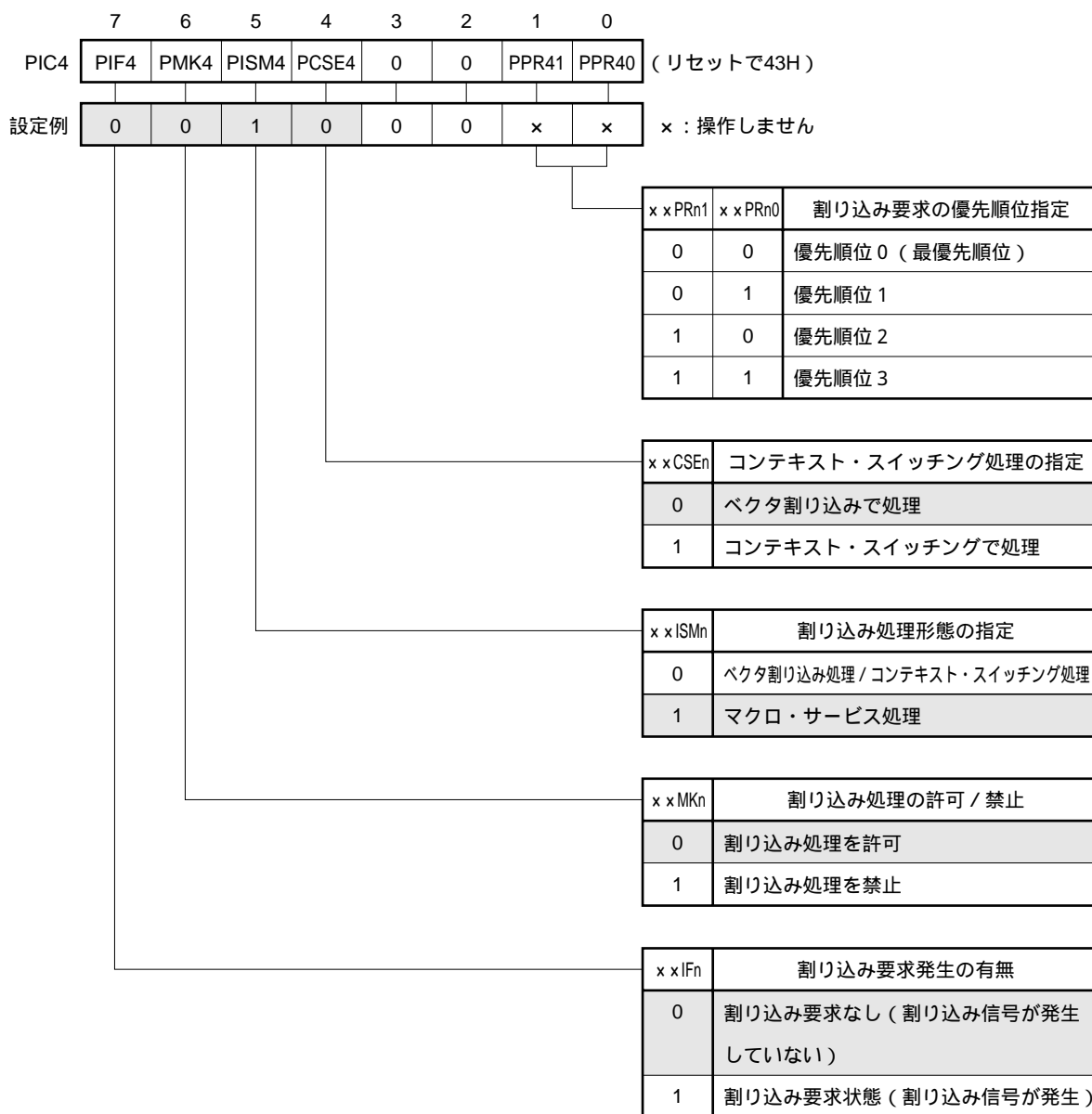
(3) 使用レジスタ

(a) 汎用レジスタ

WHLレジスタ

(b) 特殊機能レジスタ

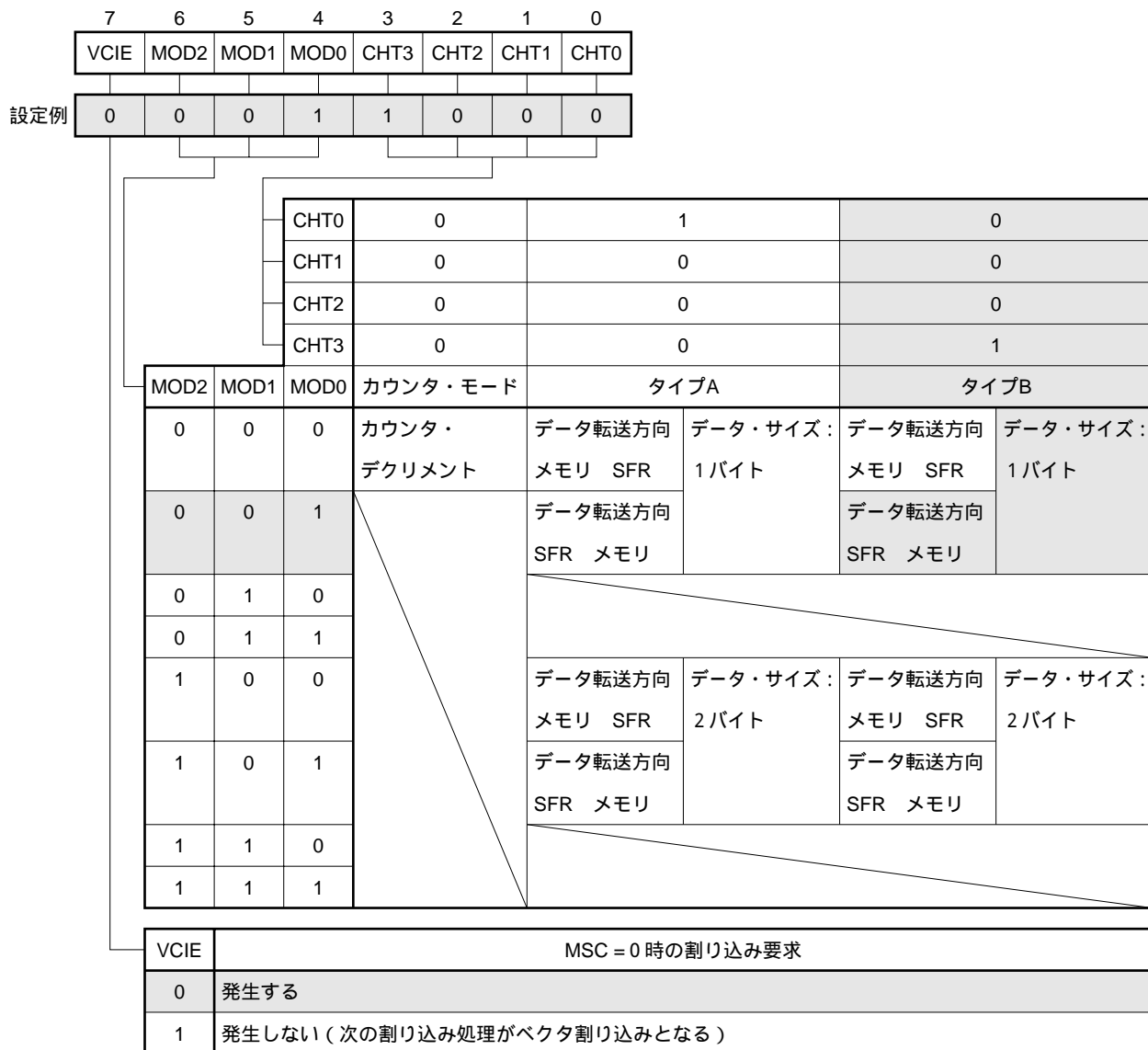
割り込み制御レジスタ



(c) マクロ・サービス制御レジスタ

マクロ・サービス・モード・レジスタ

マクロ・サービス・モード・レジスタ (1/2)



マクロ・サービス・モード・レジスタ (2/2)



(4) 入力方法

下記に示すレーベルに、次の内容を定義します。

CNTNMB : マクロ・サービス・カウンタへの設定値を定義します。

外部入力信号のエッジに同期してマクロ・サービスを起動し、P3から取り込むパラレル・データの転送回数を定義します。

STRDTB : マクロ・サービス・ポインタへの設定値を定義します。

転送されたデータを格納するワーク領域の先頭アドレスを定義します。

(5) 出力方法

下記に示すRAM領域に、転送されたデータが格納されます。

STRDTB ~ STRDTB + 1FH

(6) プログラム説明

本仕様を実現するためには、次の2つの処理が必要です。

マクロ・サービス・イニシャライズ処理 [レーベル名称: TY_BMI]

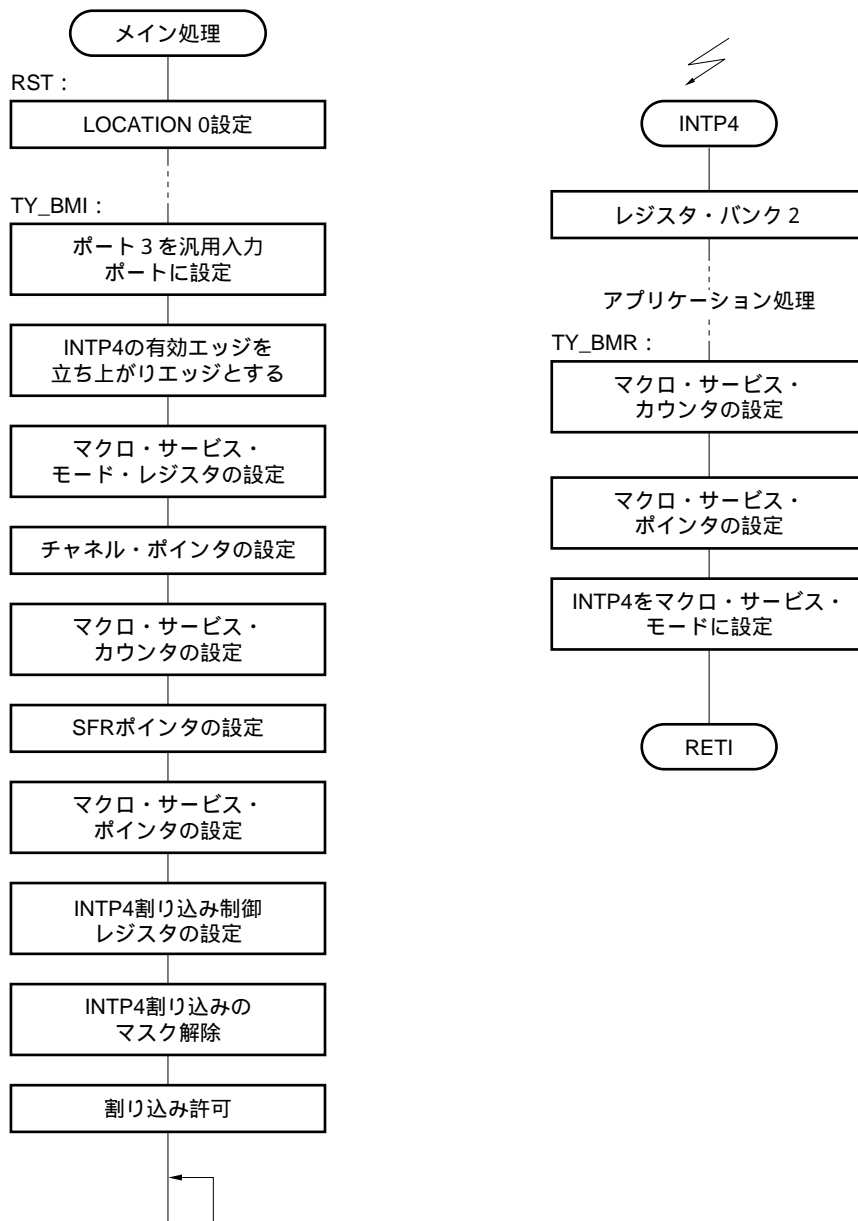
- (a) P30-P37端子を汎用入力ポートに設定します。
- (b) INTP4の有効エッジを立ち上がりエッジとします。
- (c) マクロ・サービス・モード・レジスタを設定します。
- (d) チャネル・ポインタを設定します。
- (e) マクロ・サービス・カウンタ (MSC) に転送回数 (CNTNMB) を設定します。
- (f) SFRポインタにポート3 (P3) のアドレスの下位8ビットを設定します。
- (g) マクロ・サービス・ポインタ (MP) に転送先の先頭アドレスを設定します。
- (h) INTP4割り込み要求に対する処理形態をマクロ・サービス処理に指定します。
また同時に、割り込みマスクの許可(0)と、割り込み要求のクリア(1)を行います。
- (i) 割り込みを許可します。

マクロ・サービス完了割り込み処理 [レーベル名称: TY_BMR]

マクロ・サービスを再起動させるための処理を行っています。

- (a) マクロ・サービス・カウンタ (MSC) を再設定します。
- (b) マクロ・サービス・ポインタ (MP) を再設定します。
- (c) マクロ・サービス完了割り込み時は、INTP4の割り込み要求の処理形態が、ベクタ割り込みモードになっていますので、再び処理形態をマクロ・サービス処理にします。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

- MSMP4 : マクロ・サービス・モード・レジスタ
- CHPP4 : マクロ・サービス・チャンネル・ポインタ
- MSCP4 : マクロ・サービス・カウンタ
転送回数を設定します。
- SFRP4 : マクロ・サービスSFRポインタ
対象となるSFRのアドレスを指定します。
ポート3 (P3) の下位8ビットのアドレスを設定します。
- MPP4P : マクロ・サービス・ポインタ
- STRAREA : マクロ・サービス・タイプBを使って転送するデータを格納するワーク領域。
- CNTNMB : マクロ・サービス・カウンタの設定値
ここで設定した回数分のマクロ・サービスを起動したのちにマクロ・サービス完了
割り込みが発生します。
- STRDTB : マクロ・サービス・タイプBを使って転送するデータを格納するワーク領域の先頭
アドレス

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
;***** define work area *****
;
MCHP4      DSEG      AT 0FE6EH
MPP4P:     DS        3                ; macro service pointer
SFRP4:     DS        1                ; SFR pointer
MSCP4:     DS        2                ; macro service counter
;
MCWP4      DSEG      AT 0FE1CH
MSMP4:     DS        1                ; INTP4 macro service mode register
CHPP4:     DS        1                ; INTP4 macro service channel
pointer
;
STRDTB     EQU 0A000H                ; sampling data store area
CNTNMB     EQU 20H                  ; number of data
;
STRAREA    DSEG      AT STRDTB
           DS        CNTNMB
;
;***** define vector table *****
RSTVT      CSEG      AT 0000H
           DW        RST                ; reset
INTP4VT    CSEG      AT 001CH
           DW        INTP4            ; INTP4 vector
;
;      *** initialize ***
MAINS      CSEG
RST:
           LOCATION    0H                ; specify LOCATION address
           .
           .
TY_BMI:
           MOV        PMC3,#0            ; initialize P3
           MOV        PM3,#OFFH
           MOV        INTM1,#00000100B  ; INTP4 active edge is rise
           MOV        MSMP4,#00011000B  ; set macro service mode register
           MOV        CHPP4,#LOW(MSCP4)  ; set macro service channel pointer
           MOVW       MSCP4,#LOW(CNTNMB) ; set macro service counter
           MOV        SFRP4,#LOW(P3)    ; set SFR pointer
           MOVG       WHL,#STRDTB
           MOVG       MPP4P,WHL        ; set macro service memory pointer

           MOV        SRIC,#00100011B   ; initialize interrupt
           EI
;*****
;      ***      MAIN      ***
;*****
TYB_L1:
           .
           .
           ;      メイン処理
           BR        TYB_L1

```



```
.*****
;
;***      INTP4      ***
;*****
INTP4:                                ; FOR END OF SAMPLING
    SEL      RB2                        ; CHANGE REG BANK
        .
        .
;    *** interrupt of complete macro-service ***
TY_BMR:
    MOV      MSCP4,#LOW(CNTNMB)        ; restore macro service counter
    MOVG     WHL,#STRDTB                ;
    MOVG     MPP4P,WHL                  ; set macro service memory pointer
    SET1     PISM4                       ; set interrupt service mode
    RETI
```

4.3.3 マクロ・サービス・タイプCのプログラム例

(1) 処理概要

(a) 仕様

マクロ・サービス・タイプCを用いて、リアルタイム出力ポート（POL）に接続されている、4相ステップング・モータの開ループ制御例を紹介します。ただし、 $f_{xx} = 25 \text{ MHz}$ とします（本プログラムでは、LOCATION 0命令実行時の例を示します）。

- ・リアルタイム出力ポートより出力を行い、ステップング・モータを回転させる。
（リアルタイム出力ポート：P03-P00端子使用）
- ・ステップング・モータの最小ステップ角：1.8度
- ・200 ppsの定速度回転動作（連続回転）
- ・励磁方式：1相励磁（パターン数4）
- ・リング制御：あり
- ・自動加算：あり

(b) 使用周辺機能説明

マクロ・サービス・タイプCを使用し、データ用SFRポインタ（DSFRP）で示されるリアルタイム出力ポートのバッファ・レジスタ（POL）にP00-P03端子に出力する4ビットの出力データをデータ用マクロ・サービス・ポインタ（MPD）とマクロ・サービス・カウンタ（MSC）で示されるROMの出力データ領域（あらかじめプログラムでROMに定義する）から転送します。

また同時に、タイマ用SFRポインタ（TSFRP）で示されるコンペア・レジスタ（CR10）に設定する1バイトの出力タイミング・データをタイマ用マクロ・サービス・ポインタ（MPT）とマクロ・サービス・カウンタ（MSC）で示されるROMの出力タイミング・データ領域（あらかじめプログラムでROMに定義する）から転送します。

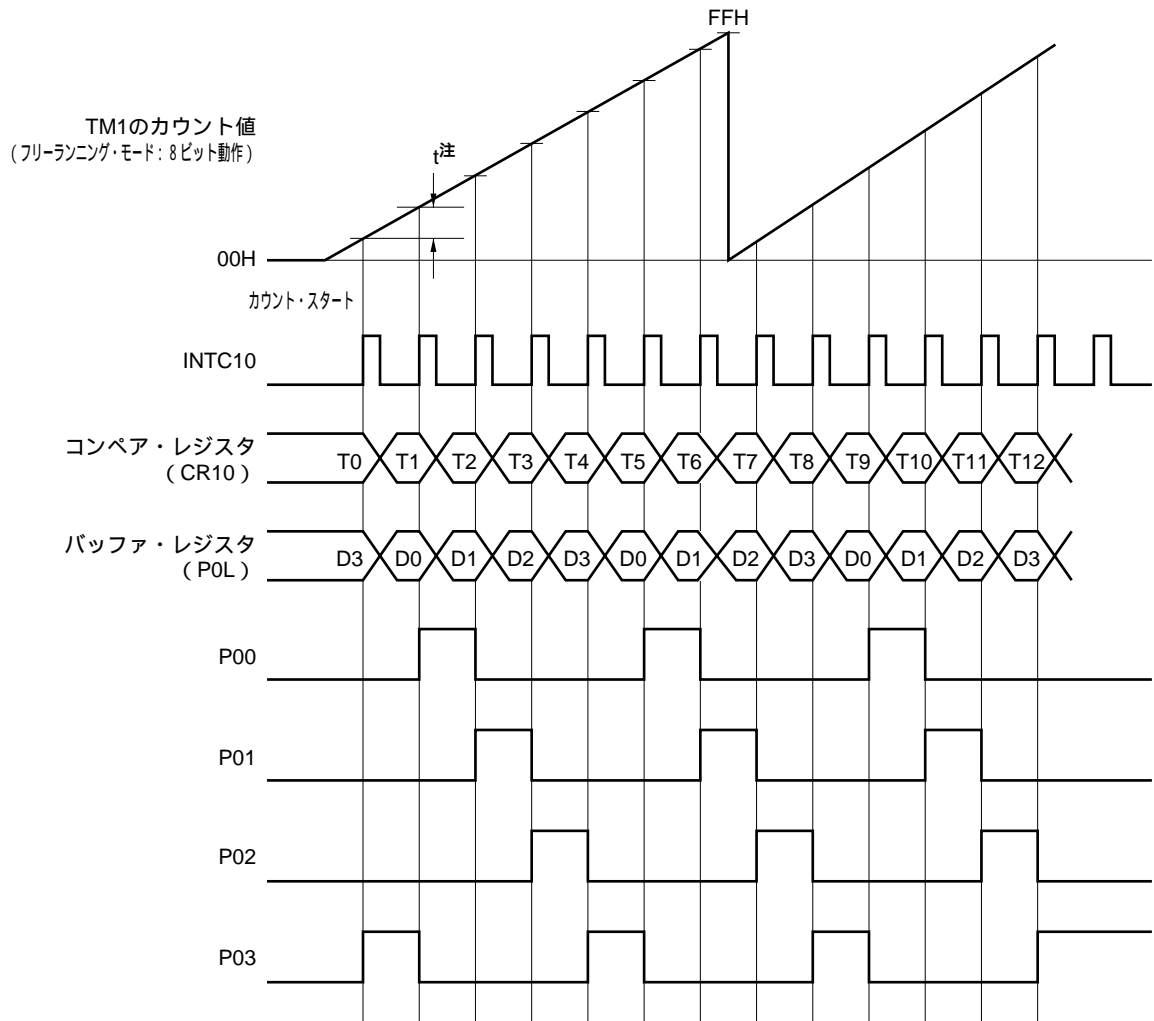
ステップング・モータを一定速度で回転させる場合、マクロ・サービス完了割り込み処理は、マクロ・サービスの再設定のみとします。

また、リング制御を使用し、リング・カウンタ（RC）、モジュロ・レジスタ（MR）に励磁パターン数4を設定し、励磁データのポインタが、自動的に元の位置に戻されるようにします。

さらに、自動加算モードを使用すると、図4-19のタイミング・チャートのような動作が実現できます。

タイミング・データ用ポインタを保持に設定しておく、タイミング・データは、 t の値が、前のコンペア・レジスタ（CR10）の値に加算されて出力されます。この場合、タイマ1（TM1）は、フリーランニング・モードで使用しますので、もう1系統のステップング・モータを動作させる場合でも、INTC11マクロ・サービスを使用して、容易に行うことができます。

図4 - 19 本プログラムのタイミング・チャート（1相励磁）



注 t (出力タイミング間隔)

200 ppsの場合，ステッピング・モータは1秒間に200ステップ回転します。

したがって，1つのパルスステップ・モータに与えてから次のパルスを与えるまでの間隔は5 msとなります（1000 ms ÷ 200ステップ = 5 ms/ステップ）。

INTC10割り込み要求が5 msごとに発生するようにタイマ/カウンタ1（8ビット動作モード使用）のコンペア・レジスタ（CR10）に値を設定します。

CR10への設定値計算方法を次に示します。

$$\begin{aligned}
 \text{CR10} &= (5 \text{ (ms)} \times \text{TM1のカウンタ・クロック}) - 1 \\
 &= (5 \text{ (ms)} \times f_{xx}/512) - 1 \\
 &= (5 \times 10^{-3} \times 25 \times 10^6/512) - 1 \\
 &= 243 - 1 \\
 &= 0F2 \text{ (H)}
 \end{aligned}$$

（ $f_{xx} = 25 \text{ MHz}$ ，タイマ/カウンタ1（8ビット動作モード使用）のカウンタ・クロックは $f_{xx}/512$ を選択します）。

表4 - 15に、本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定を示します。

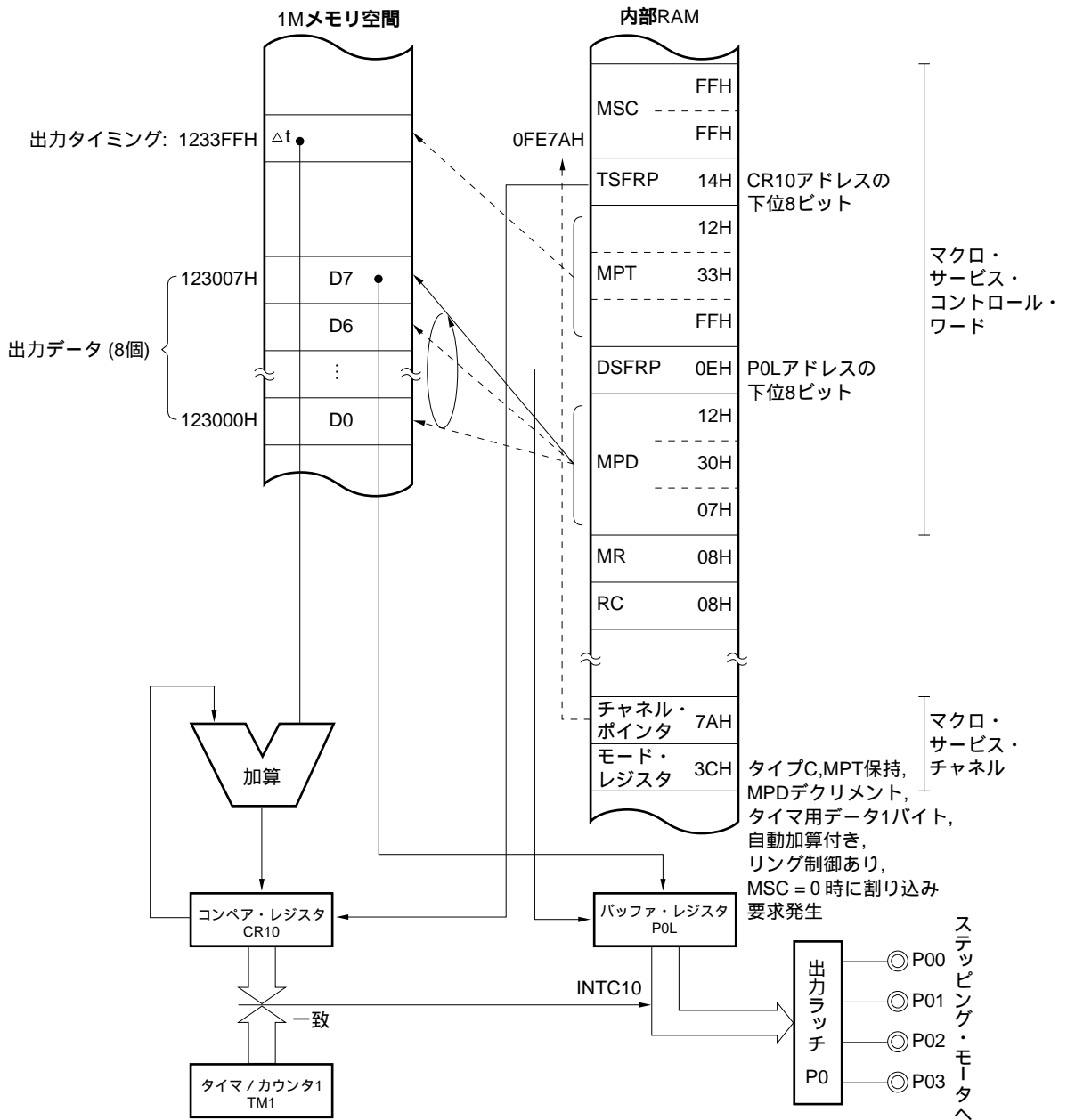
表4 - 15 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの設定

マクロ・サービス・コントロール・ワード領域	用 途	バイト数	設定値
マクロ・サービス・モード・レジスタ	マクロ・サービスの処理モードを次のように設定する。 マクロ・サービス・タイプ : タイプCに設定 リング制御 : あり 自動加算 : あり MPDの更新 : インクリメント MPTの更新 : 保持 タイマ用転送データ・サイズ : 1バイト VCIE = 0 : MSC = 0のとき割り込み要求を発生	1	3EH
マクロ・サービス・チャンネル・ポインタ	マクロ・サービス・カウンタ (MSC) のアドレスの下位 8 ビットを設定する。	1	28H

マクロ・サービス・チャンネル領域	用 途	バイト数	設定値
マクロ・サービス・カウンタ (MSC)	リアルタイム出力ポートのバッファ・レジスタ (P0L) およびコンペア・レジスタ (CR10) のデータの転送回数を設定する。 本プログラムでは、同パターンの連続回転とするため最大回数 (65536回) を設定する。	2	0000H
タイマ用マクロ・サービス・SFRポインタ (TSFRP)	転送の対象となるコンペア・レジスタ (CR10) のSFRアドレスを設定する。	1	CR10アドレスの下位 8ビットのアドレス
タイマ用マクロ・サービス・ポインタ (MPT)	あらかじめプログラムでROMに定義する、出力タイミング・データ領域先頭アドレスを設定する。	3	003100H
データ用マクロ・サービス・SFRポインタ (DSFRP)	転送の対象となるリアルタイム出力ポートのバッファ・レジスタ (P0L) のSFRアドレスを設定する。	1	P0Lアドレスの下位 8ビットのアドレス
データ用マクロ・サービス・ポインタ (MPD)	あらかじめプログラムでROMに定義する、出力データ領域の先頭アドレスを設定する。	3	003000H
モジュール・レジスタ (MR)	リング制御による繰り返しパターン数4を設定する (リング・カウンタが0になるごとに、このMRの値がRCに設定される)。	1	04H
リング・カウンタ (RC)	リング制御のパターン数をカウントする。 初期値は、MRと同一の値にする。	1	04H

図4 - 20に、本プログラムのマクロ・サービス・タイプCの処理を行うときのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図を示します。

図4 - 20 本プログラムのマクロ・サービス・コントロール・ワードとマクロ・サービス・チャンネルの配置図
(マクロ・サービス・タイプC)



備考 図中の内部RAMのアドレスは、LOCATION 0命令実行時の値です。

LOCATION 0FH命令実行時には、図中の値に0F0000Hを加えてください。

(2) 使用RAM領域

なし

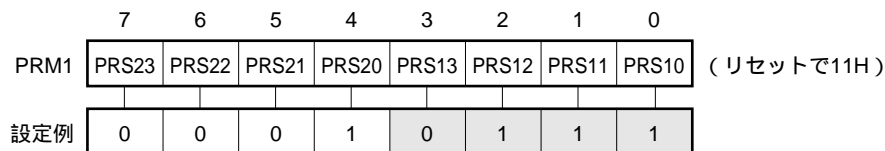
(3) 使用レジスタ

(a) 汎用レジスタ

WHLレジスタ

(b) 特殊機能レジスタ

プリスケアラ・モード・レジスタ1 (PRM1)



(f_{xx} = 25 MHz)

PRS13	PRS12	PRS11	PRS10	タイマ / カウンタ1のTM1/TM1Wのカウンタ・クロック指定	
				カウンタ・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (INTP0)	-
上記以外				設定禁止	

(f_{xx} = 25 MHz)

PRS23	PRS22	PRS21	PRS20	タイマ / カウンタ2のTM2/TM2Wのカウンタ・クロック指定	
				カウンタ・クロック [Hz] の指定	分解能
0	0	0	0	設定禁止	-
0	0	0	1	f _{xx} /8	0.32 μs
0	0	1	0	f _{xx} /16	0.64 μs
0	0	1	1	f _{xx} /32	1.28 μs
0	1	0	0	f _{xx} /64	2.56 μs
0	1	0	1	f _{xx} /128	5.12 μs
0	1	1	0	f _{xx} /256	10.2 μs
0	1	1	1	f _{xx} /512	20.5 μs
1	0	0	0	f _{xx} /1024	41.0 μs
1	0	0	1	f _{xx} /2048	81.9 μs
1	1	1	1	外部クロック (CI/INTP2)	-
上記以外				設定禁止	

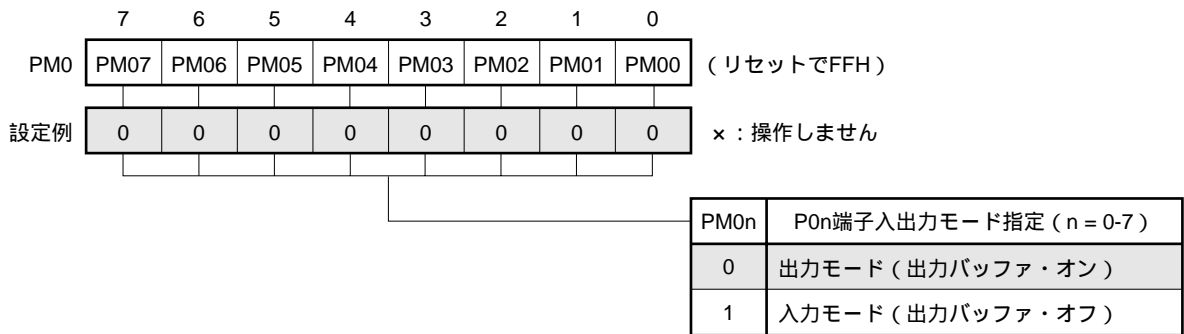
リアルタイム出力ポート・コントロール・レジスタ (RTPC)



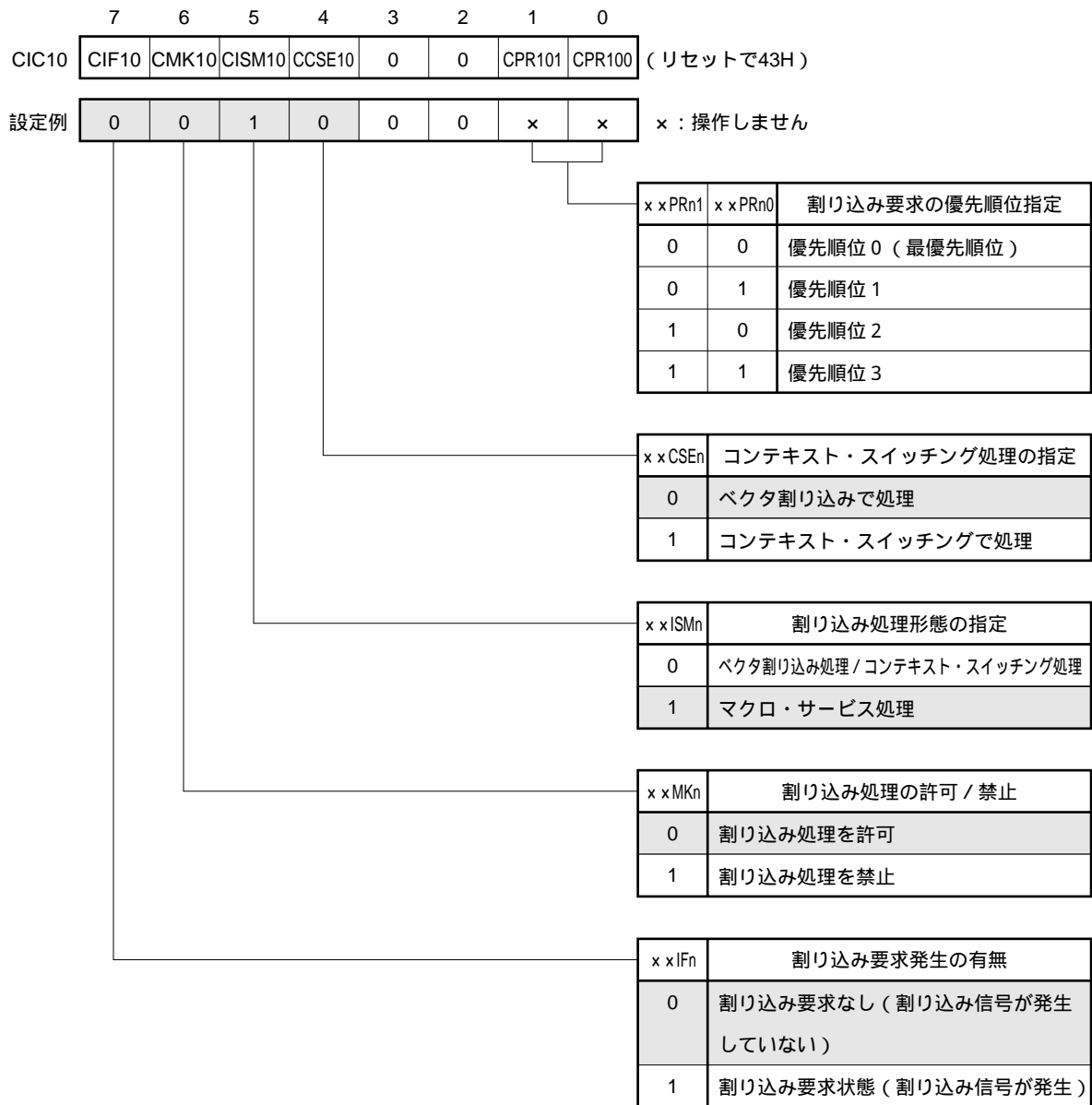
注意1 . ビット2 , 5 , 6 には必ず “ 0 ” を書き込んでください。

2 . P0ML, P0MHビットをセット (1) すると , ポート0モード・レジスタ (PM0) の内容にかかわらず , 該当するポートの出力バッファがオンとなり , ポート0出力ラッチの内容が出力されます。したがって , リアルタイム出力ポートとして指定する前に , 出力ラッチの内容を初期化してください。

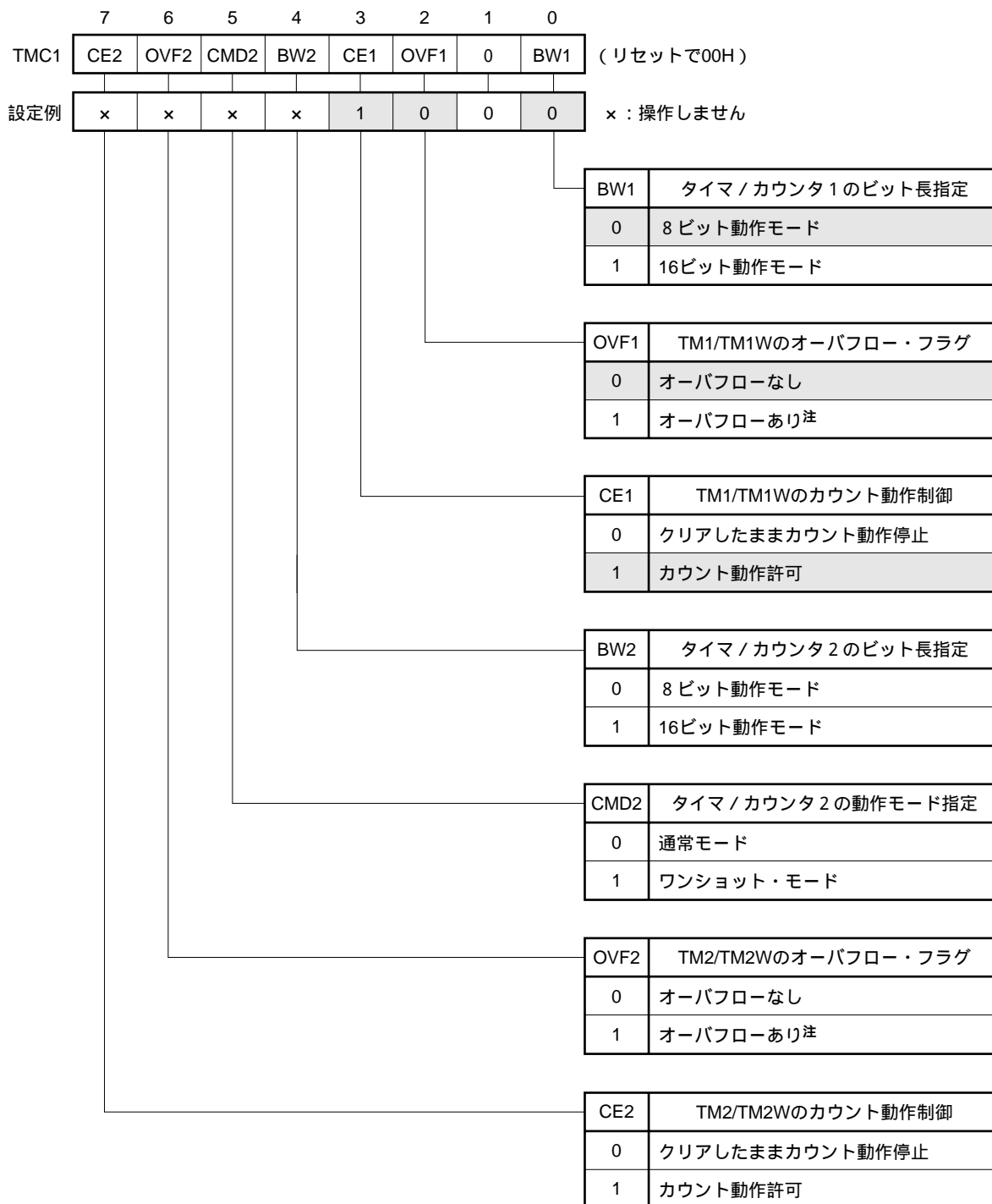
ポート0モード・レジスタ (PM0)



割り込み制御レジスタ



タイマ・コントロール・レジスタ1 (TMC1)



注 8ビット動作モード時：FFHから00Hへカウント・アップ
 16ビット動作モード時：FFFFHから0000Hへカウント・アップ

備考 OVF0ビットはソフトウェアでのみリセットされます。

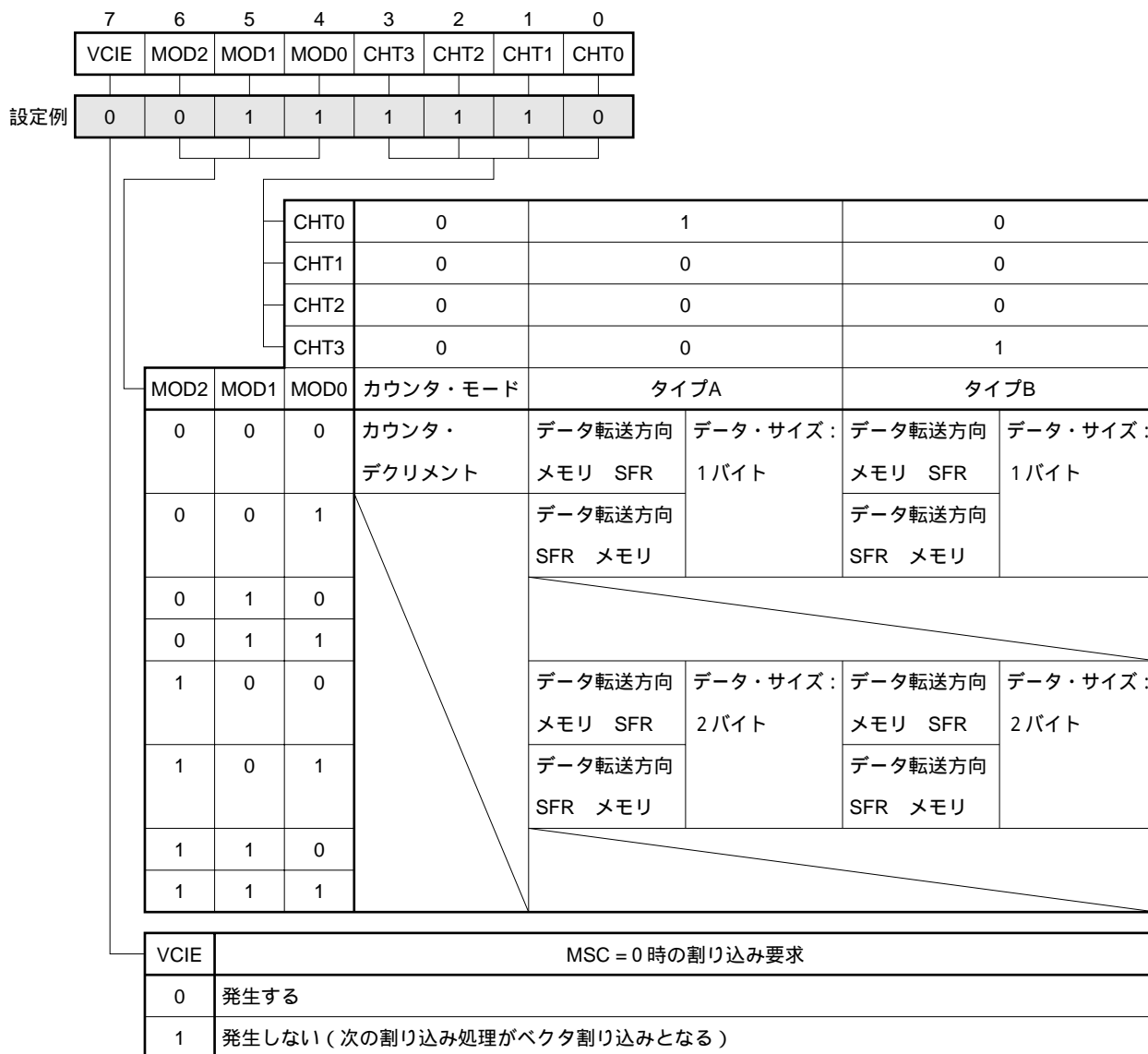
キャプチャ/コンペア・コントロール・レジスタ1 (CRC1)



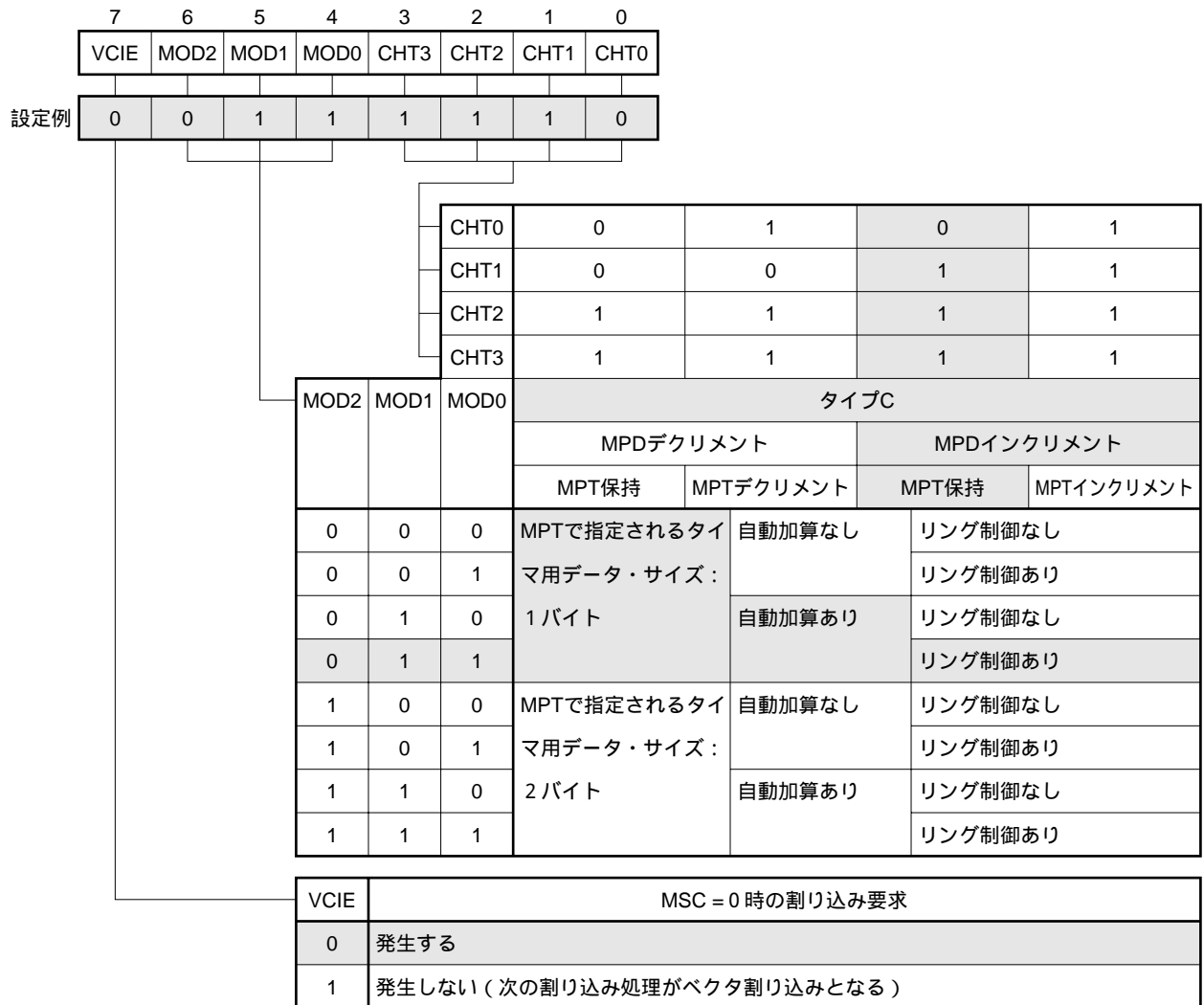
(c) マクロ・サービス制御レジスタ

マクロ・サービス・モード・レジスタ

マクロ・サービス・モード・レジスタ (1/2)



マクロ・サービス・モード・レジスタ (2/2)



(4) 入力方法

なし

(5) 出力方法

なし

(6) プログラム説明

マクロ・サービス・イニシャライズ処理 [レーベル名称 : TYC16_M]

- (a) ポート 0 に初期励磁データを設定します。
- (b) ポート 0 を出力ポートに設定します。
- (c) ポート 0 の下位 4 ビットをリアルタイム出力ポートに設定します。
- (d) タイマ・レジスタ 1 (TM1) (8 ビット動作モードで使用) のカウント動作を停止します。
- (e) タイマ・レジスタ 1 (TM1) とコンペア・レジスタ (CR10) の内容一致でのカウント・クリアを禁止します。
- (f) タイマ・レジスタ 1 (TM1) のカウント・クロックを設定します。
- (g) マクロ・サービス・モード・レジスタを設定します。
- (h) チャンネル・ポインタを設定します。
- (i) データ用SFRポインタを設定します。
- (j) タイマ用SFRポインタを設定します。
- (k) マクロ・サービス・カウンタに励磁パターンの数を設定します。
- (l) リング・カウンタを設定します。
- (m) モジユロ・レジスタを設定します。
- (n) データ用マクロ・サービス・ポインタを設定します。
出力データ : 1 相励磁方式でステッピング・モータを回転させるための波形形成用データ。
- (o) タイマ用マクロ・サービス・ポインタを設定します。
出力タイミング・データ : 単位時間にステッピング・モータに与えるパルス数。
- (p) INTC10割り込み要求に対する処理形態をマクロ・サービス処理に設定します。
また同時に、割り込みマスクの許可 (0) と、割り込み要求のクリア (1) を行います。
- (q) タイマ・レジスタ 1 (TM1) (8 ビット動作モードで使用) のカウントを開始します。

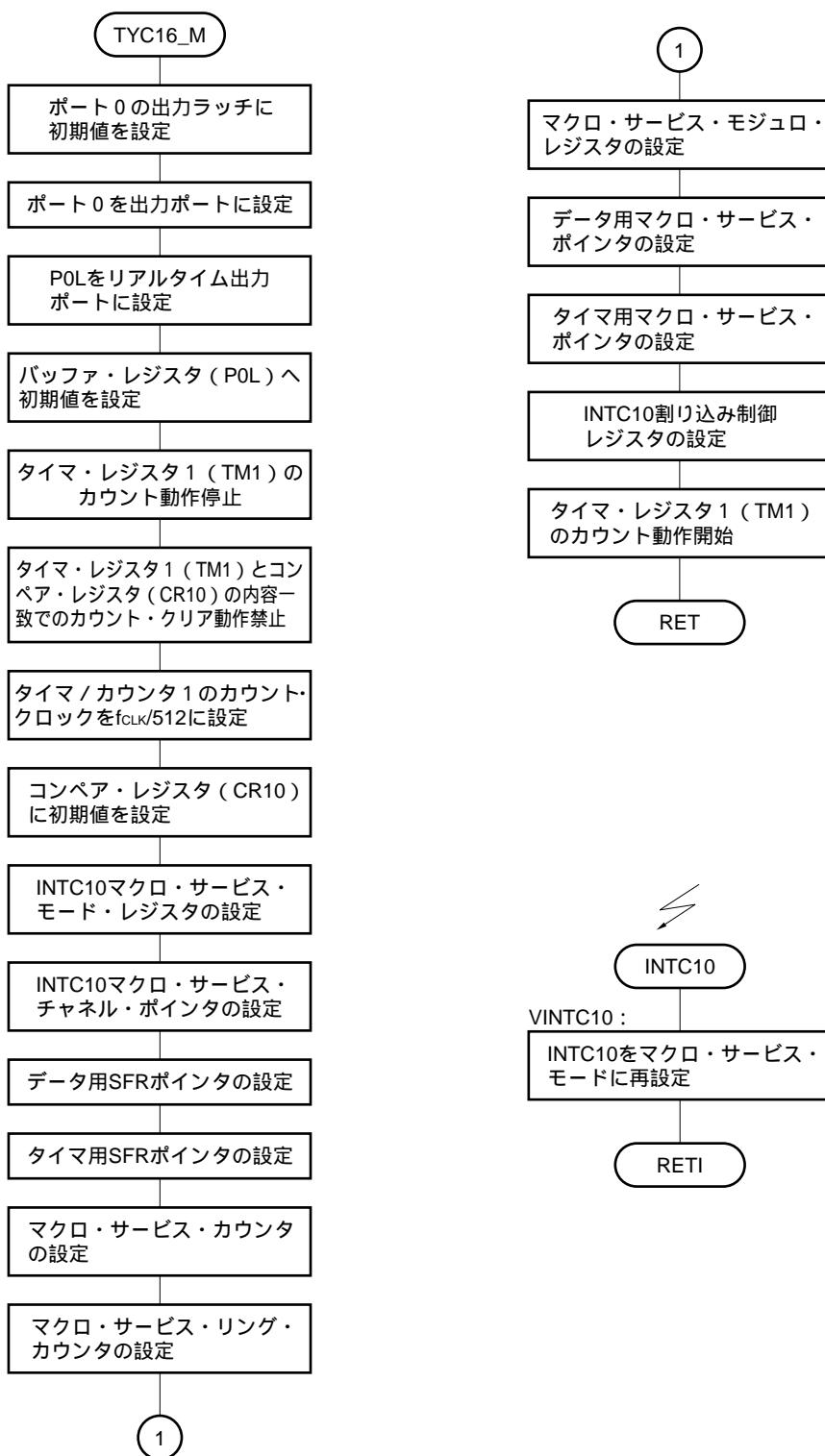
マクロ・サービス完了割り込み処理 [レーベル名称 : VINTC10]

(a) INTC10の割り込み要求の処理形態をマクロ・サービス処理に再設定します。

備考 この応用例では、ステッピング・モータを回転させ続けているため、完了割り込み発生時点で、マクロ・サービス・カウンタ (MSC) の再設定を行っていません (完了割り込み発生時点でマクロ・サービス・カウンタ (MSC) が最大の転送回数である 0 になっているため) 。

また、リング制御を使用しているため、マクロ・サービス・ポインタ (MPD, MPT) は、自動的に設定されるため、これも再設定の必要がありません。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

- MSMC10 : マクロ・サービス・モード・レジスタ
- CHPC10 : マクロ・サービス・チャンネル・ポインタ
- MSCC10P : マクロ・サービス・カウンタ
STPCNTに定義した値を設定します。
- MSCC10P : マクロ・サービス・カウンタ
- TSFRPC10 : タイマ用マクロ・サービスSFRポインタ
- MPTC10P : タイマ用マクロ・サービス・ポインタ
あらかじめプログラムでROMに定義する, 出力タイミング・データ領域の先頭アドレスを設定します。
- DSFRPC10 : データ用マクロ・サービスSFRポインタ
- MPDC10P : データ用マクロ・サービス・ポインタ
あらかじめプログラムでROMに定義する, 出力データ領域の先頭アドレスを設定します。
- MODC10 : モジュール・レジスタ
- RCC10 : リング・カウンタ
- STPCNT : マクロ・サービス・カウンタへの設定値
ここで設定した回数分の転送が行われたのちにマクロ・サービス完了割り込みが発生します。
- PTNCNT : モジュール・レジスタ, リング・カウンタへの設定値

メイン・ルーチンのプログラム・リスト記述例

```

;
; --- FOR TYPE C ---
;
PUBLIC MAINST
EXTRN TYC16_M, VINTC10
.
.
INTC10VT CSEG AT 0012H
        DW INTC10 ;INTC10 vector
;
;
; *** MAIN ROUTINE ***
;
CSEG
MAINST:
        .
        .
        CALL !TYC16_M ; macro service type-c initialize
        EI ;
        .
        .
;
;
; *** VECTOR ROUTINE ***
;
INTC10:
        CALL! !VINTC10 ;
        RETI

```

本アプリケーション・ルーチンのプログラム・リスト

```

NAME      TYC16M
;
; *****
;* macro service type-C application at macro service      *
;* for stepping motor control No.2                        *
; *****
;

PUBLIC    TYC16_M,VINTC10          ; package

STPCNT    EQU      0              ; step count (65536)
PTNCNT    EQU      4              ; pattern count

WORKEQU    DSEGAT 0FE12H
MSMC10:    DS       1              ; macro service mode register
CHPC10:    DS       1              ; macro service channel pointer

;
; DSEGSADDR
RCC10:     DS       1              ; macro service ring counter
MODC10:    DS       1              ; macro service modulo register
MPDC10P:   DS       3              ; macro service pointer for data
DSFRPC10:  DS       1              ; macro service SFR pointer for data
MPTC10P:   DS       3              ; macro service pointer for timing
TSFRPC10:  DS       1              ; macro service SFR pointer for timing data
MSCC10P:   DS       2              ; macro service counter

;
; *****
;* define table                                          *
; *****
D0 EQU     0000001B
D1 EQU     00000010B
D2 EQU     00000100B
D3 EQU     00001000B
TN EQU     0F2H

;
; *** for data ***
DTFLIE    CSEGAT 3000H
CYLDATA:  DB      D0,D1,D2,D3

;
; *** for timing ***
TMFILE    CSEGAT 3100H
CYLTIM:   DB      TN

```

```

;
;
;   *** initialize   ***
;
CSEG
TYC16_M:
MOV     P0,#LOW(D3)           ; P0 output latch <- first data
MOV     PM0,#00000000B       ; set P0 output port
MOV     RTPC,#00000001B     ; initialize real-time output port
MOV     POL,#LOW(D3)        ; set P0 buffer register first data

MOV     TMC1,#00000000B     ; stop TM1
MOV     CRC1,#00000000B     ; disable clear TM1
MOV     PRM1,#00010111B    ; set TM1 prescaler fxx/512
MOV     CR10,#0            ; set first timing data

MOV     MSMC10,#00111110B   ; set mode register (TYPE C)
MOV     CHPC10,#LOW(MSCC10P) ; set channel pointer
MOV     DSFRPC10,#LOW(POL)  ; set macro service SFR pointer for data
MOV     TSFRPC10,#LOW(CR10) ; set macro service SFR pointer for timing
MOVW    MSCC10P,#STPCNT     ; set macro service counter
MOV     RCC10,#PTNCNT       ; set ring counter

MOV     MODC10,#PTNCNT     ; set modulo register
MOVG    WHL,#CYLDATA       ; set pointer for data
MOVG    MPDC10P,WHL        ; set pointer for data
MOVG    WHL,#CYLTIME       ; set pointer for timing
MOVG    MPTC10P,WHL        ; set pointer for timing

MOV     SRIC,#00100011B    ; initialize interrupt

MOV     TMC1,#00001000B    ; start TM1
RET

;
; *****
; *   complete of macro service   *
; *****
;

VINTC10:
SET1    CISM10             ; set INTC10 macro service mode again
RET

END

```

第5章 A/Dコンバータのプログラム例

5.1 A/Dコンバータ機能概要

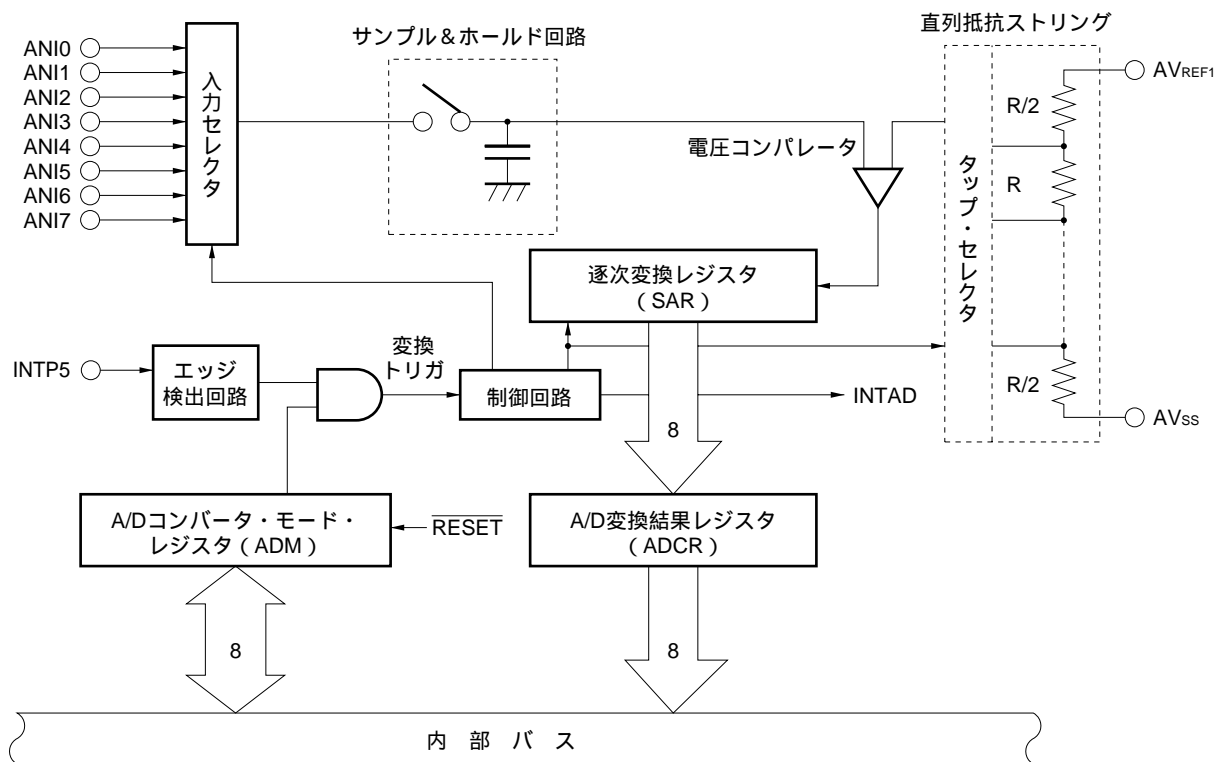
μPD784038, 784038Yサブシリーズには、高速、高分解能の8ビット・アナログ/デジタル(A/D)コンバータを内蔵しています。

アナログ信号を入力する8本のアナログ入力端子(ANI0-ANI7)と、変換結果を保持するA/D変換結果レジスタ(ADCR)を持っています。

変換方式は逐次比較で、変換結果はA/D変換結果レジスタ(ADCR)に保持します(変換時間9.6 μs: $f_{CLK} = 12.5 \text{ MHz}$)。

図5-1に、A/Dコンバータのブロック図を示します。

図5-1 A/Dコンバータのブロック図



A/Dコンバータの起動には、次のモードがあります。

ハードウェア・スタート：外部割り込み端子 (INTP5) 入力の有効エッジにより変換開始。

ソフトウェア・スタート：A/Dコンバータ・モード・レジスタ (ADM) のCSフラグをセット (1) することにより変換開始。

また、起動後の変換動作モードには次の2種類があります。

スキャン・モード：複数のアナログ入力端子を順次選択し、各端子からの変換結果を順次A/D変換結果レジスタ（ADCR）に保持します。

セレクト・モード：アナログ入力端子の任意の1端子の連続的な変換結果を、A/D変換結果レジスタ（ADCR）に保持します。

変換動作モードと変換動作の停止は、A/Dコンバータ・モード・レジスタ（ADM）で指定します。

5.2 A/Dコンバータのプログラム例

(1) 処理概要

(a) 仕様

4本のアナログ入力端子（AN10-AN13）より入力された電圧をマイコンで扱うため、数値に置き換えるプログラム例を紹介します。

- ・変換動作モードは、スキャン・モードを使用します。
- ・4本のアナログ入力端子の各A/D変換結果を16回取り出して、平均値を求めます（マクロ・サービス・タイプBを使用）。

(b) 使用周辺機能説明

A/Dコンバータの変換時間は、 $9.6\mu\text{s}$ ($f_{\text{CLK}} = 12.5\text{MHz}$) です。マクロ・サービス・タイプBを使用し、A/D変換結果レジスタ（ADCR）に保持された変換結果を、RAM領域に確保した変換結果格納領域（C000H-C03FH）に転送します。

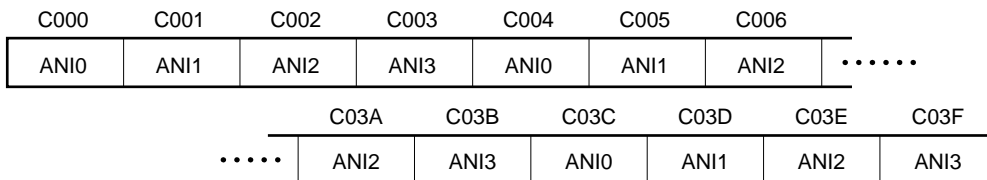
平均値は、マクロ・サービス終了による割り込み処理内で算出します。

本プログラムでは、マクロ・サービス終了による割り込み処理でA/Dコンバータの変換動作を停止させています。

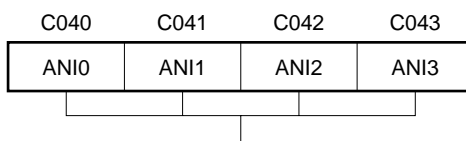
図5-2に、本プログラムで使用するRAM領域を示します。

図5 - 2 A/Dコンバータのプログラム例の使用RAM領域

・変換結果格納領域



・平均値格納領域



{ 変換時間 (9.6μs) × サンプリング回数 } × 変換するアナログ入力端子数 } 時間ごとに、各アナログ入力端子の変換結果の平均値を格納します。

各アナログ入力端子 (ANI0-ANI3) の変換結果を保持するA/D変換結果レジスタ (ADCR) から、16個の変換結果をマクロ・サービス・タイプBを用いて変換結果格納領域に転送します。転送が終了するとマクロ・サービスの終了による割り込みが発生します。

変換結果の平均値は、マクロ・サービス終了による割り込み処理内にて計算、更新します。

本例では、スキャン・モード2 (遅延制御付き) を使用しています。スキャン・モード1 (遅延制御なし) を使用するプログラムを設計する場合は、次に示す内容に注意してください。

注意 割り込み要求の発生後、9.6 μ s以内にマクロ・サービス処理が完了するようにしてください。

割り込み要求、マクロ・サービスが一時的に保留される命令を実行すると、マクロ・サービスの実行が遅れます。連続して割り込み要求、マクロ・サービスが保留される命令が実行されると、A/D変換時間(9.6 μ s)内にマクロ・サービスが実行できずに変換結果を取りこぼす可能性があります。NOP命令などを前記命令の間に挿入し、割り込みや、マクロ・サービスが受け付けられるタイミングを作るか、A/Dコンバータ・モード・レジスタ(ADM)のSCMDビットをセット(1)し、遅延制御付きでA/Dコンバータを使用してください。

次に、割り込み要求およびマクロ・サービスが一時的に保留される命令を示します。

```

EI
DI
BRK
BRKCS
RETCS
RETCSB !addr16
RETI
RETB
LOCATION 0HまたはLOCATION 0FH
POP PSW
POPU post
MOV PSWL, A
MOV PSWL, byte
MOVG SP, imm24

```

割り込み制御レジスタ^注、MK0、MK1L、IMC、ISPRの各レジスタに対する書き込み命令およびビット操作命令(BT、BF命令を除く)

PSWのビット操作命令

(ただし、BT PSWL. bit, \$ addr20, BF PSWL. bit, \$ addr20, BT PSWH. bit, \$addr20, BF PSWH. bit, \$addr20, SET1 CY, NOT1 CY, CLR1 CY命令を除く)。

注 割り込み制御レジスタ：PIC0、PIC1、PIC2、PIC3、PIC4、PIC5、CIC00、CIC01、CIC10、CIC11、CIC20、CIC21、CIC30、ADIC、SERIC、SRIC、CSIIC1、STIC、CSIIC、SERIC2、SRIC2、CSIIC2、STIC2、SPCIC(μPD784038Yサブシリーズのみ)

マクロ・サービスの要求が競合した場合、デフォルト優先順位の高いマクロ・サービスが終了するまで低優先のマクロ・サービスは保留されるため、A/D変換時間（9.6 μ s）内にマクロ・サービスが実行できなくなる可能性があります。このような場合、A/Dコンバータ・モード・レジスタ（ADM）のSCMDビットをセット（1）し、遅延制御付きでA/Dコンバータを使用してください。

（2）使用RAM領域

本プログラムでは、表5 - 1に示すようにRAM領域を使用します。

表5 - 1 A/Dコンバータのプログラム例で使用するRAM領域

RAM / フラグ名称	用 途	バイト数	初期値
STRDATA	マクロ・サービス・タイプBを用いて、各アナログ入力端子（ANI0-ANI3）の変換結果を保持するA/D変換結果レジスタ（ADCR）からデータを格納するRAM領域。	64	-
STRDATA + 40H	各アナログ入力端子の変換結果の平均値を格納するRAM領域。	4	-

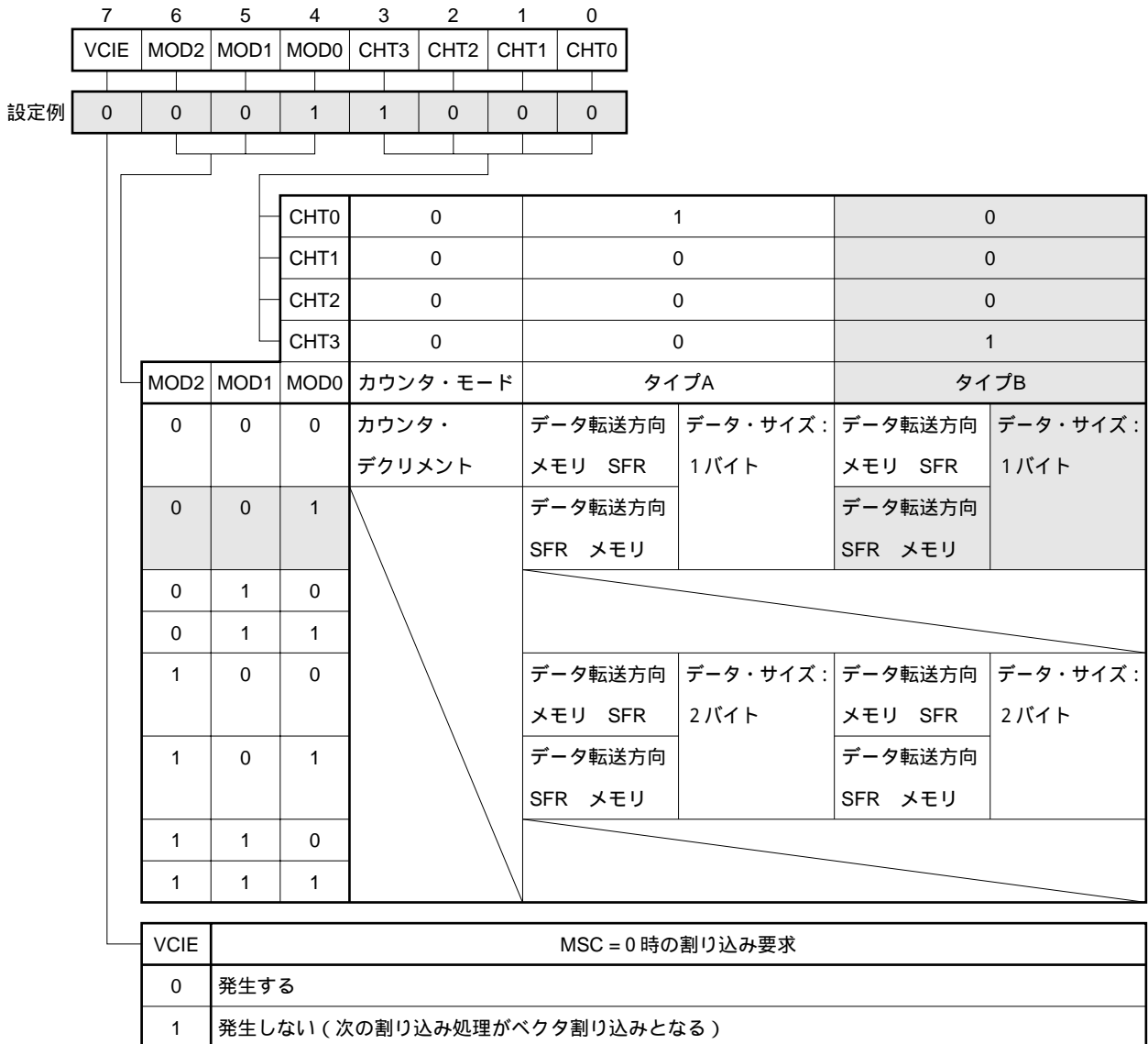
（3）使用レジスタ

（a）汎用レジスタ

A, X, B, C, WHLレジスタ

(b) 特殊機能レジスタ

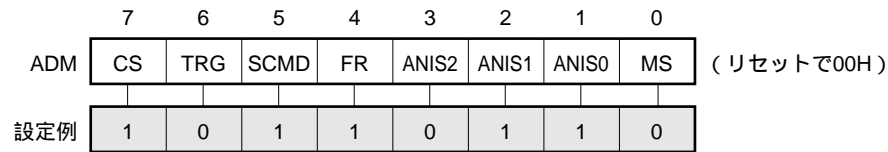
マクロ・サービス・モード・レジスタ (1/2)



マクロ・サービス・モード・レジスタ (2/2)



A/Dコンバータ・モード・レジスタ (ADM)



ANIS2	ANIS1	ANIS0	MS	A/D変換動作モードの設定	
0	0	0	0	スキャン・モード	ANI0入力をスキャン
0	0	1	0		ANI0, ANI1入力をスキャン
0	1	0	0		ANI0-ANI2入力をスキャン
0	1	1	0		ANI0-ANI3入力をスキャン
1	0	0	0		ANI0-ANI4入力をスキャン
1	0	1	0		ANI0-ANI5入力をスキャン
1	1	0	0		ANI0-ANI6入力をスキャン
1	1	1	0		ANI0-ANI7入力をスキャン
0	0	0	1	セレクト・モード	ANI0入力をセレクト
0	0	1	1		ANI1入力をセレクト
0	1	0	1		ANI2入力をセレクト
0	1	1	1		ANI3入力をセレクト
1	0	0	1		ANI4入力をセレクト
1	0	1	1		ANI5入力をセレクト
1	1	0	1		ANI6入力をセレクト
1	1	1	1		ANI7入力をセレクト

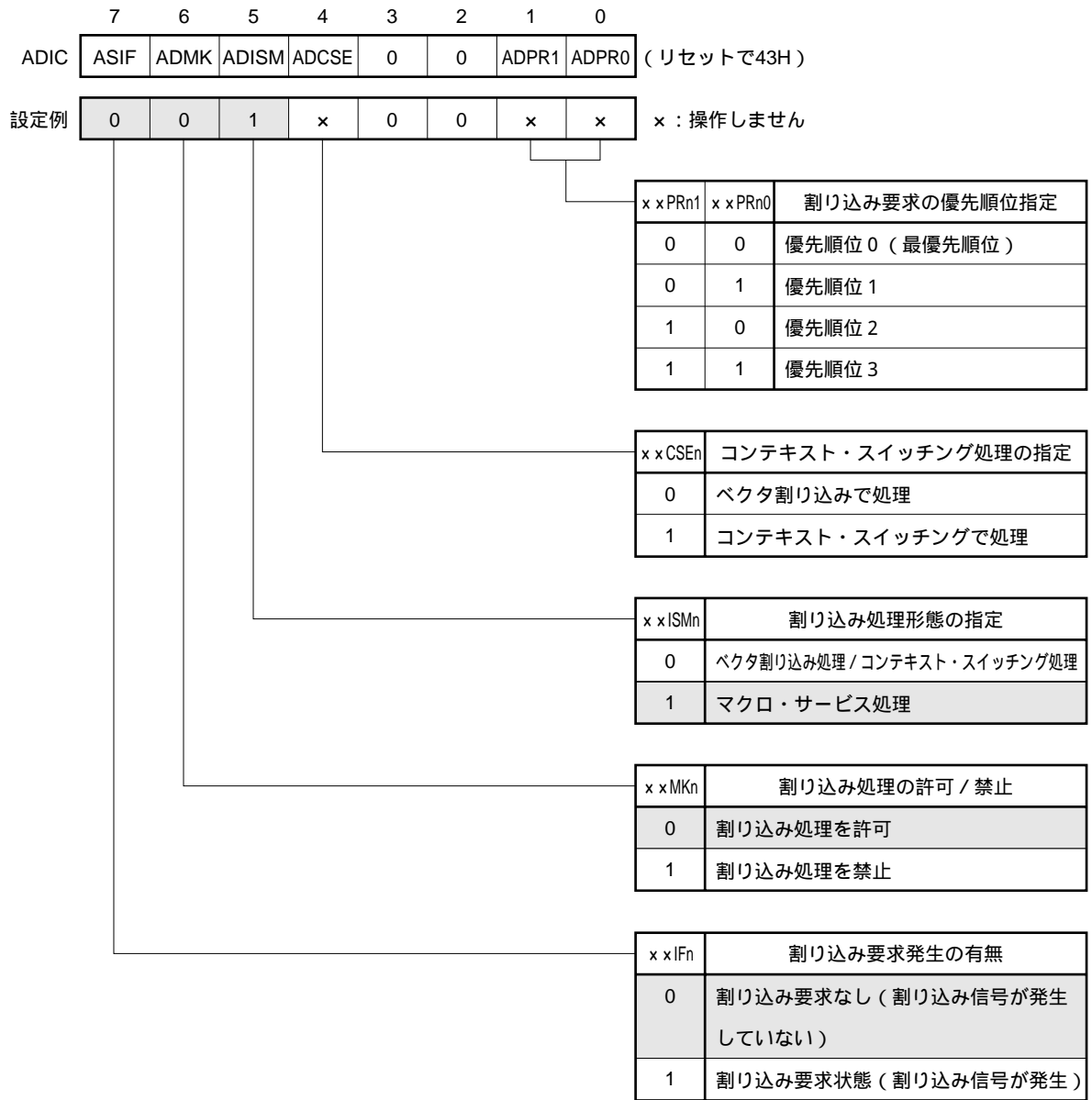
FR	変換速度制御 ($f_{CLK} = 12.5 \text{ MHz}$)	
0	$180/f_{CLK}$ (14.4 μs)	低速変換
1	$120/f_{CLK}$ (9.6 μs)	高速変換

SCMD	MS	スキャン・モードの選択
0	0	スキャン・モード0 (遅延制御なし)
1	0	スキャン・モード1 (遅延制御付き)
0	1	セレクト・モード
1	1	設定禁止

TRG	外部端子トリガの制御
0	外部トリガ禁止
1	外部トリガ許可

CS	A/D変換動作制御
0	A/D変換動作停止
1	A/D変換動作開始

割り込み制御レジスタ



(4) 入力方法

マクロ・サービス・タイプBのイニシャライズ処理 [レーベル名称 : MS_AD]

下記に示すRAM領域に、次の内容を設定します。

CNTNMA : A/D変換結果の転送回数を設定します。

マクロ・サービス完了割り込み処理 [レーベル名称 : END_AD]

なし

(5) 出力方法

マクロ・サービス・タイプBのイニシャライズ処理 [レーベル名称 : MS_AD]

下記に示すRAM領域に、次のデータが格納されます。

STRDAT ~ STRDAT + 3FH : A/Dコンバータ機能を用いて変換した変換結果が格納されます。

マクロ・サービス完了割り込み処理 [レーベル名称 : END_AD]

下記に示すRAM領域に、次のデータが格納されます。

STRDAT + 40H ~ STRDAT43H : 各アナログ入力端子の変換結果の平均値が格納されます。

(6) プログラム説明

本仕様を実現するためには、次の処理が必要です。

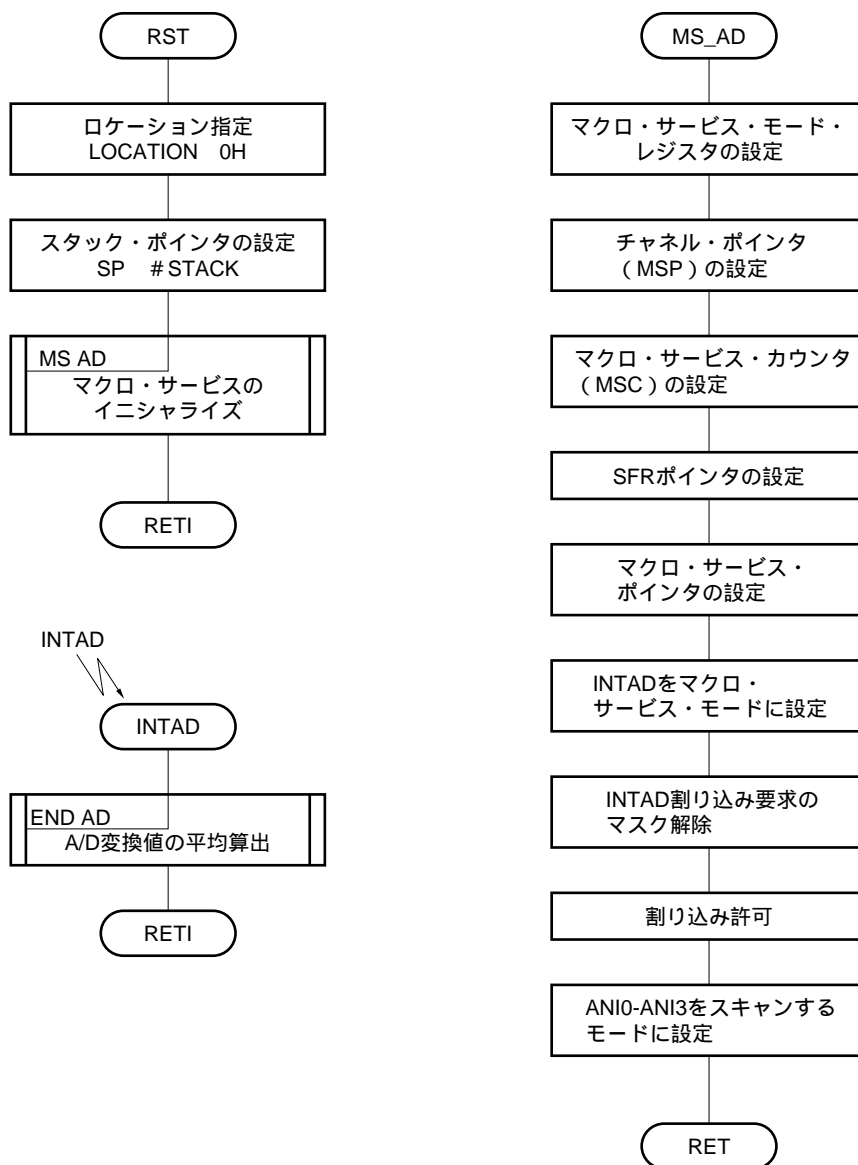
マクロ・サービス・タイプBのイニシャライズ処理 [レーベル名称 : MS_AD]

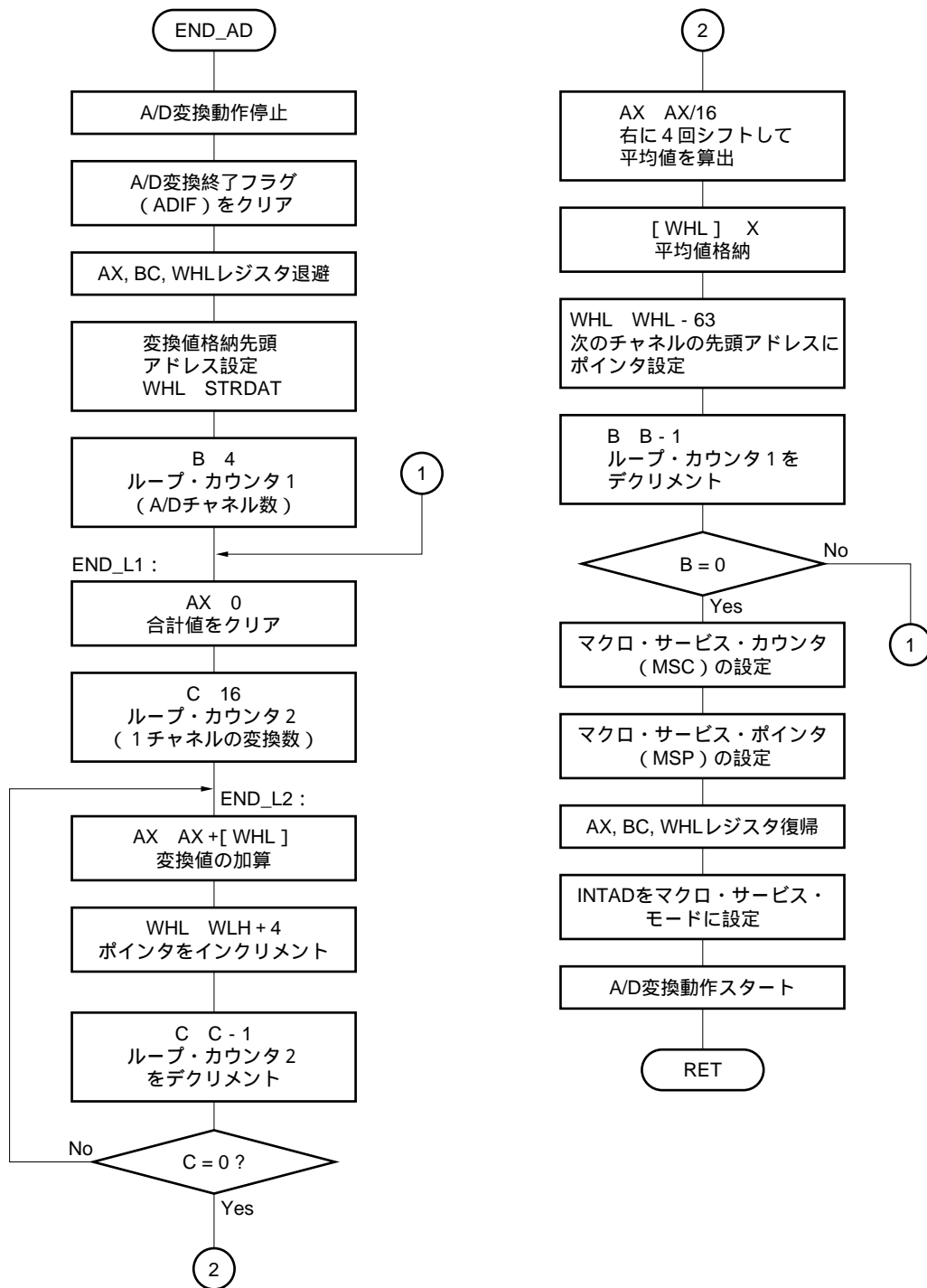
- (a) マクロ・サービス・モード・レジスタを設定します。
 - マクロ・サービス・タイプ : タイプBに設定
 - データ転送方向 : SFR メモリ
 - 転送データ・サイズ : 1バイト
 - VCIE = 0 : MSC = 0のとき割り込み要求を発生します。
- (b) チャンネル・ポインタ (MSP) を設定します。
- (c) マクロ・サービス・カウンタ (MSC) にデータ転送回数 (CNTNMA) を設定します。
- (d) SFRポインタの設定を行います。A/D変換結果レジスタ (ADCR) のアドレス下位8ビットを格納します。
- (e) マクロ・サービス・ポインタに、変換結果領域の先頭アドレス (STRDAT) を設定します。
- (f) A/D変換動作モードをスキャン・モード (ANI0-ANI3) に指定し、A/Dコンバータの変換動作をスタートします。
- (g) INTAD割り込み要求に対する処理形態を、マクロ・サービス処理に指定します。
- (h) INTAD割り込み要求に対する割り込みのマスクを解除します。
- (i) 割り込みを許可します。

マクロ・サービス完了割り込み処理 [レーベル名称 : END_AD]

- (a) A/Dコンバータの変換動作を停止します。
- (b) A/D変換終了割り込み要求フラグ (ADIF) をクリア (0) します。
- (c) それぞれのチャンネルの変換結果の合計を求めます。
- (d) 平均値を算出します (合計値を右に4回シフトすることにより求めます)。
- (e) 平均値をSTRDAT + 40H ~ STRDAT + 43Hに格納します。
- (f) マクロ・サービス・カウンタ (MSC) とマクロ・サービス・ポインタ (MSP) を再設定します。
- (g) INTAD割り込み要求に対する処理形態を、マクロ・サービス処理に指定します。
- (h) A/Dコンバータの変換動作を再スタートします。

(7) フロー・チャート





(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベル/フラグの説明

CNTNMA : 転送回数 (64回) を設定します。

STRDAT : A/D変換結果を転送するRAM領域の先頭アドレスです。

LP1 : 使用する入力端子 (ANI0-ANI3) の本数 (4本) を設定します。

LP2 : 使用する入力端子 (ANI0-ANI3) の1入力端子に対する変換回数 (16回) を設定します。

CSAD : A/Dコンバータ変換開始フラグ

CSAD = 0...A/Dコンバータ変換停止状態

CSAD = 1...A/Dコンバータ変換開始状態

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
WADREG   DSEG   AT 0FE20H
MSMAD:   DS     1           ; INTAD macro service mode register
CHPAD:   DS     1           ; INTAD macro service channel pointer
      DSEG   SADDR
MPADP:   DS     3           ; INTAD memory pointer
SFRAD:   DS     1           ; INTAD SFR pointer
MSCAD:   DS     2           ; INTAD macro service counter
;
CSAD     EQU    ADM.7       ; A/D convert start bit
      .
      .
;
; --- vector table ---
;
RSTVT     CSEG   AT 0000H
          DW     RST        ; reset
INTADV   CSEG   AT 0020H
          DW     INTAD      ; INTAD
;
; --- main ---
;
      CSEG
RST:
      LOCATION 0H
      .
      .
      CALL    !MS_AD        ; macro service initialize for INTAD
      .
      .
;
INTAD:                                       ; complete of INTAD macro service
      SEL    RB1           ;
      CALL    !END_AD
      RETI
;

```

```

; *** initialize ***
;
;
MS_AD:
    PUSH    WHL                ;
;
    MOV     MSMAD,#00011000B   ; set macro service mode register
;
    MOV     CHPAD,#LOW(MSCAD)  ; set macro service channel pointer
    MOV     MSCAD,#CNTNUM      ; set macro service counter
    MOV     SFRAD,#LOW(ADCR)   ; set SFR pointer
    MOVG    WHL,#STRDAT        ;
    MOVG    MPADP,WHL          ; set macro service memory pointer
;
;
    POP     WHL                ;
;
;
    MOV     ADM,#10110110B     ; initialize A/D converter
    CLR1    ADIF                ;
    SET1    ADISM               ; set interrupt mode
    CLR1    ADMK                ; open mask of INTAD
    EI                      ; interrupt enable
    RET
;
;
; *** interrupt of complete macro-service ***
;
;
END_AD:
    CLR1    CSAD                ; A/D converter stop
    CLR1    ADIF                ; INTAD flag clear
;
;
    MOVG    WHL,#STRDAT        ; pointer set
    MOV     B,#LP1              ; loop counter_1 set (4)
EAD_L1:
    MOVW    AX,#0               ; sum clear
    MOV     C,#LP2              ; loop counter_2 set (16)
EAD_L2:
    XCH     A,X                 ; addition result
    ADD     A,[WHL]
    XCH     A,X
    ADDC    A,#0                ; carry addition
;
;
    ADDG    WHL,#4              ; pointer <- pointer + 4
    DBNZ   C,$EAD_L2           ; check loop counter_2
;
;
    SHRW    AX,4                ; average of result
    XCH     A,X
    MOV     [WHL],A
    SUBG    WHL,#63             ; next pointer set 16*4-1
    DBNZ   B,$EAD_L1           ; check loop counter_1
;
;
    MOVW    MSCAD,#CNTNUM      ; set macro service counter
    MOVG    WHL,#STRDAT        ;
    MOVG    MPADP,WHL          ; set macro service memory pointer
;
;
    SET1    ADISM               ;
    SET1    CSAD                ; A/D convert start
;
;
    RET
END

```

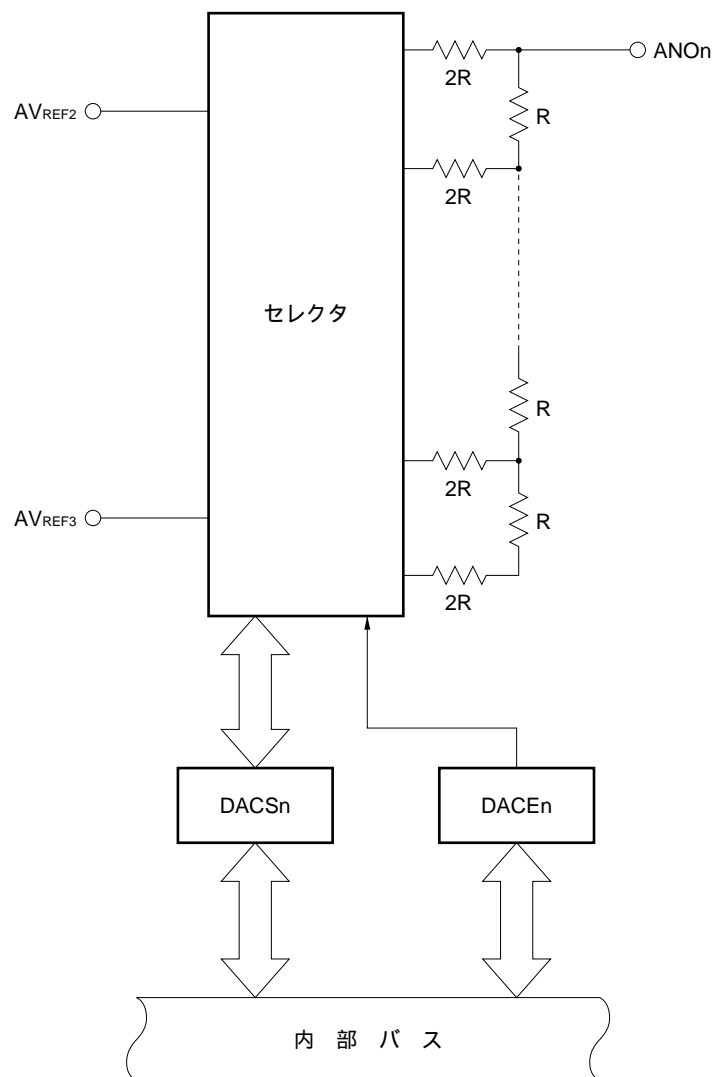
第6章 D/Aコンバータのプログラム例

6.1 D/Aコンバータ機能概要

μ PD784038, 784038Yサブシリーズには、8ビット分解能の電圧出力型デジタル/アナログ(D/A)コンバータを2回路内蔵しています。変換方式は、R-2R抵抗ラダー方式です。

図6-1に、D/Aコンバータのブロック図を示します。

図6-1 D/Aコンバータのブロック図



備考 $n = 0, 1$

6.2 D/Aコンバータのプログラム例

(1) 処理概要

(a) 仕様

D/Aコンバータ用アナログ電圧出力端子 (ANO0) を使用して、周波数50 Hzのサイン波形の出力を行うプログラム例を紹介します。

- ・ タイマ / カウンタ 1 を668 μ sごとに割り込み要求が発生するように設定します。このタイマ割り込みが発生するたびに電圧を変化させ、サイン波形を出力します。
- ・ アナログ出力電圧は、0 ~ 5 Vの範囲となるようにします。

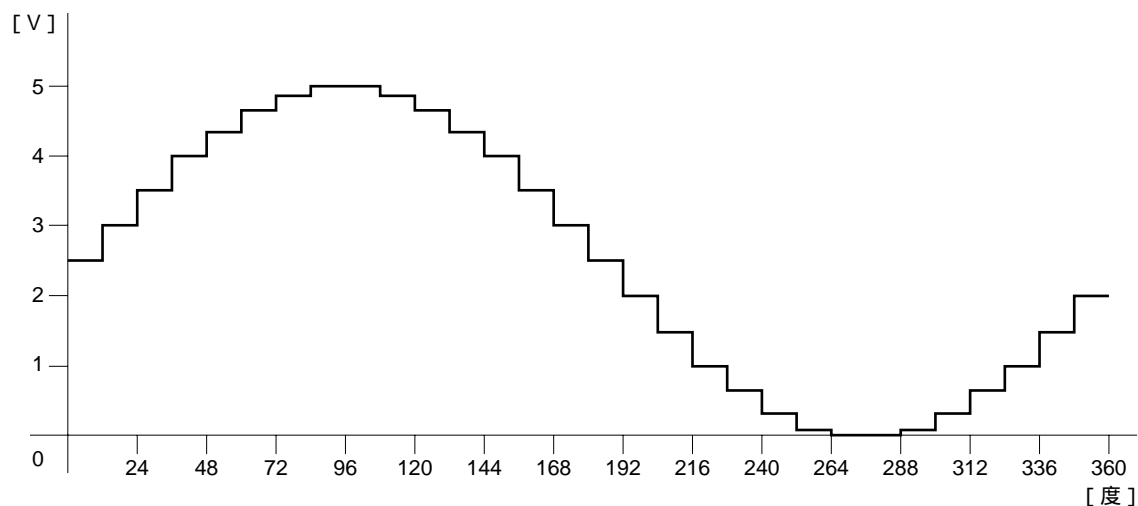
(b) 使用周辺機能説明

D/A変換値設定レジスタ (DACS0) に設定する出力電圧の設定値は、ROM領域に配置します。

タイマ / カウンタ 1 のタイマ割り込み処理内で、順次ROM領域の出力電圧の設定値をテーブル参照し、D/A変換値設定レジスタ (DACS0) に設定します。

図6 - 2 に、出力するサイン波形を示します。

図6 - 2 サイン出力波形



ANO0端子に出力されるアナログ電圧は、次の式で決定されます。

$$\text{ANO0端子出力電圧} = \frac{\text{AV}_{\text{REF2}}^{\text{注1}} - \text{AV}_{\text{REF3}}^{\text{注2}}}{256} \times \text{DACS0}^{\text{注3}} + \text{AV}_{\text{REF3}}$$

注1 . AV_{REF2} = 5 V
注2 . AV_{REF3} = 0 V
注3 . DACS0 = D/A変換値設定レジスタの値

サイン出力波形の電圧と、D/A変換値設定レジスタ (DASC0) に設定する設定値を表6 - 1に示します。

表6 - 1 サイン出力波形の電圧，設定値 (AV_{REF2} = 5 V, AV_{REF3} = 0 V)

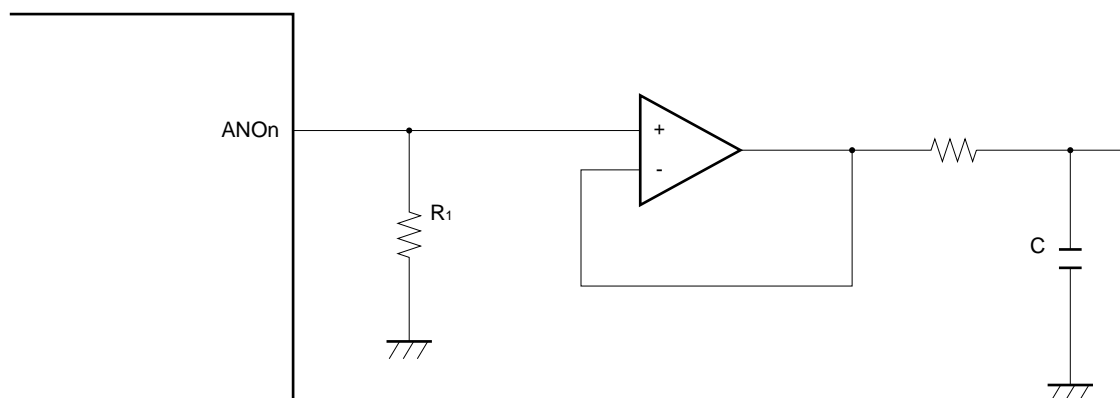
度数	電圧 (V)	設定値 (DACS0)	度数	電圧 (V)	設定値 (DACS0)
0	2.5000	80H	180	2.5000	80H
12	3.0200	9BH	192	1.9802	65H
24	3.5168	B4H	204	1.4832	4CH
36	3.9695	CBH	216	1.0305	35H
48	4.3579	DFH	228	0.6421	21H
60	4.6651	EFH	240	0.3349	11H
72	4.8776	FAH	252	0.1224	06H
84	4.9863	FFH	264	0.0137	01H
96	4.9863	FFH	276	0.0137	01H
108	4.8776	FAH	288	0.1224	06H
120	4.6651	EFH	300	0.3349	11H
132	4.3579	DFH	312	0.6421	21H
144	3.9695	CBH	324	1.0305	35H
156	3.5168	B4H	336	1.4832	4CH
168	3.0200	9BH	348	1.9802	65H

注意 表6 - 1に示した電圧値は、小数点以下第5位で四捨五入したデータです。
 ただし、設定値は、四捨五入する前のデータで計算し、その結果データを小数点以下第1位で四捨五入したデータです。

備考 D/Aコンバータ用アナログ電圧出力端子（AN00）より出力されたアナログ電圧は、階段状の波形となります。階段状の波形を滑らかなサイン波形にして出力するために、図6-3に示すように、バッファ・アンプの後段にロウ・パス・フィルタを通すと、段差のないサイン波形を生成できます。

図6-3 サイン波形変換回路

μPD784038, 784038Yサブシリーズ



注意1 . D/Aコンバータの出力インピーダンスが高いため、ANOn端子 ($n=0, 1$) から電流を取り出すことはできません。負荷の入力インピーダンスが低い場合には、負荷とANOn端子の間にバッファ・アンプを挿入して使用してください。また、バッファ・アンプや負荷までの配線は極力短くしてください（出力インピーダンスが高いため）。

配線が長くなる場合は、グランド・パターンで囲むなどの処理をしてください。

- 2 . D/Aコンバータの出力電圧は階段状に変化しますので、一般的にはD/Aコンバータの出力する信号は、ロウ・パス・フィルタを通してから使用してください。
- 3 . μPD784038, 784038Yサブシリーズの内蔵するD/Aコンバータは、 $\overline{\text{RESET}}$ がロウ・レベルの期間中は出力がハイ・インピーダンスとなります。負荷側の回路は、入力が高インピーダンスになっても良いように考慮してください。
- 4 . D/Aコンバータの出力は、リセット解除後、 AV_{REF3} 端子と同一レベルを出力しますので、リセット解除後、 AV_{REF3} 端子レベルが出力されても良いように考慮してください。

(2) 使用RAM領域

本プログラムでは、表6-2に示すようにRAM領域を使用します。

RAM領域は、ショート・ダイレクト・アドレッシングの適用範囲に配置してください。

表6-2 D/Aコンバータのプログラム例で使用するRAM領域

RAM / フラグ名称	用途	バイト数	初期値
C_DATA	D/A変換値設定レジスタ (DACS0) に設定する値のテーブル参照のカウンタ	1	0FFH

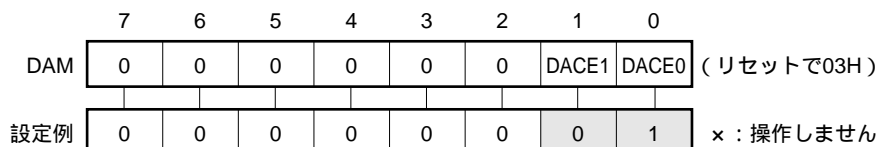
(3) 使用レジスタ

(a) 汎用レジスタ

A, B, WHLレジスタ

(b) 特殊機能レジスタ

D/Aコンバータ・モード・レジスタ (DAM)



DACE1	DACE0	D/Aコンバータの動作
0	0	スタンバイ・モード ANO0, ANO1端子とも出力ハイ・インピーダンス
0	1	ANO0端子は出力許可 ANO1端子は出力ハイ・インピーダンス
1	0	ANO0端子は出力ハイ・インピーダンス ANO1端子は出力許可
1	1	ANO0, ANO1端子とも出力許可

(4) 入力方法

下記に示すRAM領域に、次の内容を設定します。

SDATA : D/A変換値設定レジスタ(DACS0)に設定するD/Aコンバータの出力電圧の設定値
テーブルの先頭番地を設定します。

ENDDAT : D/Aコンバータの出力電圧の設定値数(設定値個数 - 1)を設定します。

C_DATA : D/Aコンバータの出力電圧設定値テーブルのポインタ用カウンタ

(5) 出力方法

なし

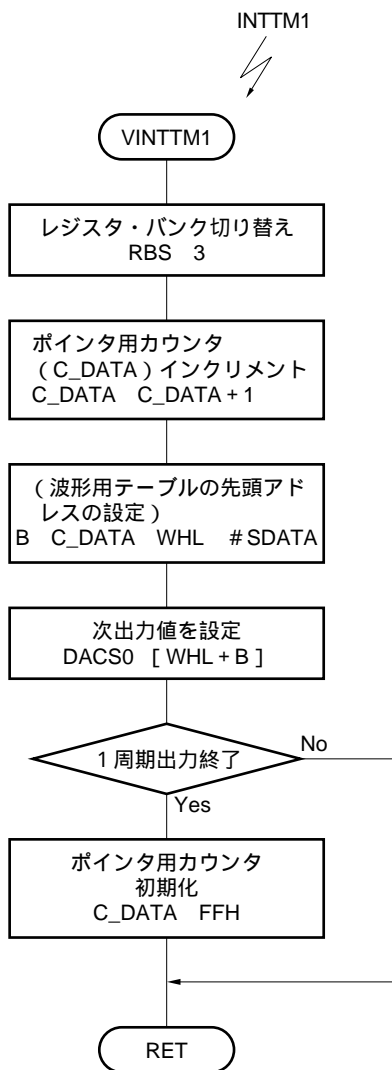
(6) プログラム説明

本仕様を実現するためには、タイマ/カウンタ1のタイマ割り込み処理内で次の処理を行います。

サイン波形の出力値設定を行います [レーベル名称 : INTTM1]

- (a) D/Aコンバータの出力電圧の設定値テーブルのポインタ用カウンタ(C_DATA)をインクリメントします。
- (b) D/Aコンバータの出力電圧の設定値をテーブル参照して、D/A変換値設定レジスタ(DACS0)にその設定値を設定します。
- (c) D/Aコンバータの出力電圧の設定値テーブルのポインタ用カウンタの値で、サイン波形1周期分が終了したかどうかを確認します。
 - 1周期分終了... (d) の処理へ進む。
 - 1周期分未終了... 割り込み処理を終了する。
- (d) D/Aコンバータの出力電圧の設定値テーブルのポインタ用カウンタを初期化します。

(7) フロー・チャート



(8) プログラム・リスト

アプリケーション・ルーチン時に使用するレーベルの説明

- SDATA : D/A変換値設定レジスタ (DACSO) に設定するD/Aコンバータの出力電圧の設定値
テーブルの先頭番地を設定します。
- ENDDAT : D/Aコンバータの出力電圧の設定値数 (設定値個数 - 1) を設定します。
- C_DATA : D/Aコンバータの出力電圧の設定値テーブルのポインタ用カウンタ

本アプリケーション・ルーチンのプログラム・リスト

```

      .
      .
VETIM1 CSEG  AT 12H
      DW    INTTM1
;
SINRAM  DSEG  SADDR
C_DATA: DS    1           ;ROM data counter
ENDDAT  EQU   1DH       ;SIN WAVE data number
      CSEG
RES_STA:
      LOCATION OH
      SEL    RBO
      .
      .
      MOV    TMC1,#00000000B      ;TM1 STOP
      MOV    PRM1,#00010110B      ;SIN_DAT:COUNT CLOCK fxx/256
      MOV    CRC1,#00000010B      ;
      MOV    CR10,#66-1           ;TIMER1 interval 668us
;
      MOV    C_DATA,#ENDDAT      ;
      MOV    G   WHL,#SDATA      ;
      MOV    B,C_DATA            ;
      MOV    A,[WHL+B]           ;
      MOV    DACS0,A             ;
      MOV    DAM,#00000001B      ;
      MOV    PIC4,#01000000B     ;
      EI                          ;
      .
      .
SETFDA:
      MOV    C_DATA,#0FFH        ;
      MOV    TMC1,#00001000B     ;
      CLR1   CIF10               ;
      CLR1   CMK10               ;TIMER1 MASK
      MOV    DAM,#00000001B      ;
      .
      .

```

```

INTTM1:
        SEL      RB3                ;select bank 3
;
        INC      C_DATA            ;
        MOV      B,C_DATA          ;
        MOVG     WHL,#SDATA        ;
        MOV      A,[WHL+B]         ;
        MOV      DACSO,A           ;
        CMP      B,#ENDDAT        ;end data?
        BNZ      $VRETI           ;[NO]
        MOV      C_DATA,#0FFH     ;[YES] counter initialize
VRETI:
        RETI
;
SDATA:
        DB      09BH                ;SIN wave data
        DB      0B4H                ;
        DB      0CBH                ;
        DB      0DFH                ;
        DB      0EFH                ;
        DB      0FAH                ;
        DB      0FFH                ;
        DB      0FFH                ;
        DB      0FAH                ;
        DB      0EFH                ;
        DB      0DFH                ;
        DB      0CBH                ;
        DB      0B4H                ;
        DB      09BH                ;
        DB      080H                ;
        DB      065H                ;
        DB      04CH                ;
        DB      035H                ;
        DB      021H                ;
        DB      011H                ;
        DB      006H                ;
        DB      001H                ;
        DB      001H                ;
        DB      006H                ;
        DB      011H                ;
        DB      021H                ;
        DB      035H                ;
        DB      04CH                ;
        DB      065H                ;
        DB      080H                ;
END

```

〔メ モ〕

第7章 PWM出力ユニットのプログラム例

7.1 PWM出力ユニットの機能概要

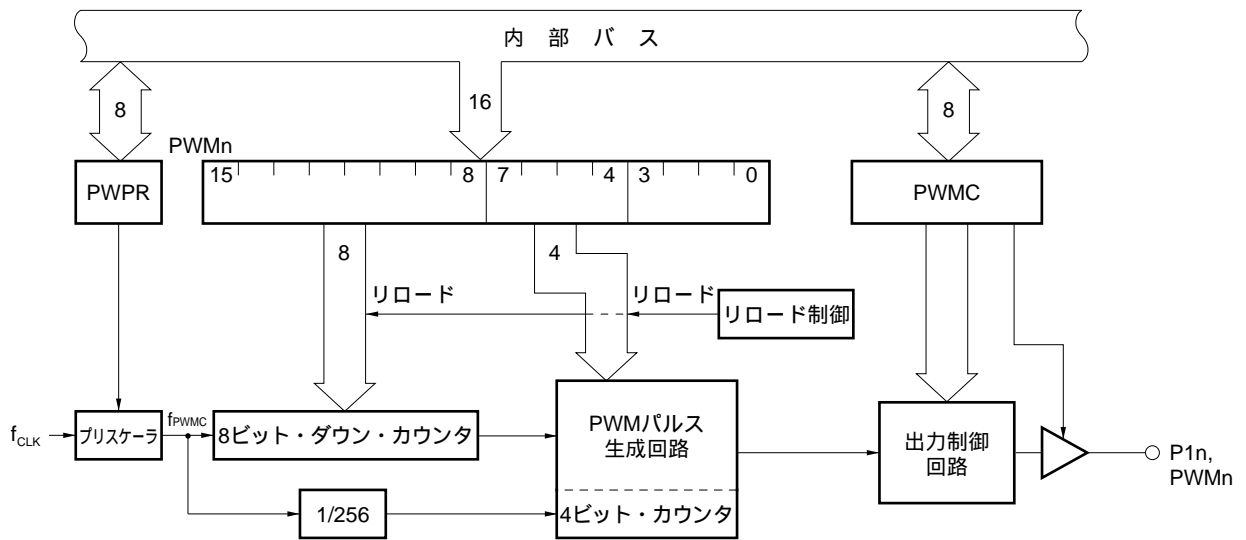
μPD784038, 784038YサブシリーズのPWM出力ユニットは、12ビット分解能のPWM（パルス幅変調）出力回路を2チャンネル内蔵しています。PWM出力パルスはハイ/ロウのアクティブ・レベルを選択することができます。

一般的にPWM出力信号は、ロウ・パス・フィルタにて波形整形し、電圧制御に使用します。

(1) PWM出力ユニットのシステム構成

図7-1に、PWM出力ユニットのブロック図を示します。

図7-1 PWM出力機能のブロック図



備考 n = 0, 1

(2) PWM出力ユニットの基本動作

PWM信号出力のデューティは、PWMモジュロ・レジスタ (PWMn : n = 0, 1) のビット4-15に設定する値で次のように決定されます。

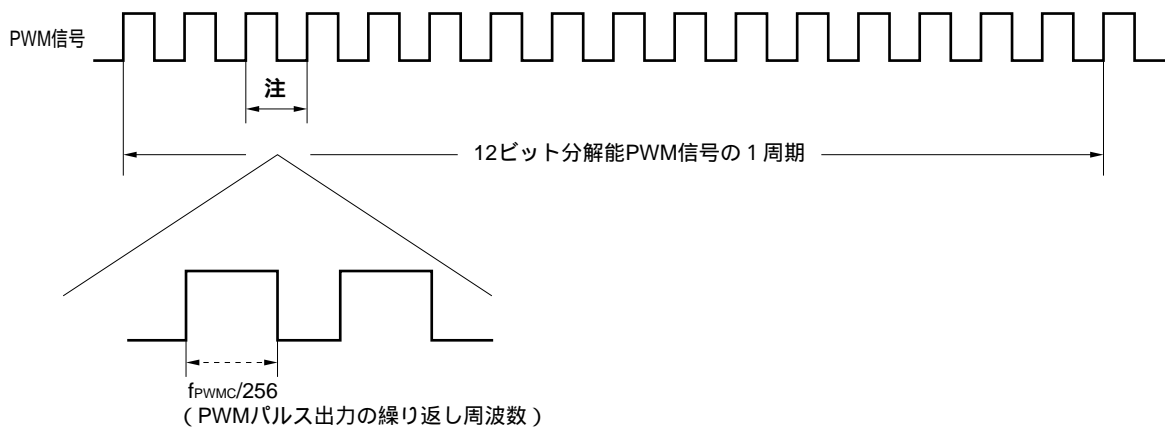
$$\text{PWM信号出力のデューティ} = \frac{(\text{PWMnのビット4-15の値})^{\text{注}+1}}{4096}$$

注 16 (PWMnのビット4-15の値) 4095

PWMパルス出力の繰り返し周波数は、PWMプリスケラ・レジスタ (PWPR) で設定される $f_{\text{CLK}}/1-f_{\text{CLK}}/4$ のPWMクロック (f_{PWMC}) を256分周 ($= f_{\text{PWMC}}/256$) した周波数になり、最小パルス幅は、 $1/f_{\text{PWMC}}$ になります。

PWM信号出力は、繰り返し周波数 $f_{\text{PWMC}}/256$ の8ビット分解能のPWMパルスを16回繰り返し出力することで、12ビット分解能を実現しています。PWMモジュロ・レジスタ (PWMn) のビット8-15で決定される8ビット分解能のPWMパルスに、1周期ごとにPWMモジュロ・レジスタ (PWMn) のビット4-7の値に従って、追加パルス ($1/f_{\text{PWMC}}$) の付加を制御して、16周期で1回の12ビット分解能のPWM信号を実現しています。

図7-2 PWM出力の基本動作



注意 PWMパルス出力の繰り返し周波数は8ビット分解能です。

(3) PWMモジュール・レジスタ (PWM0, PWM1) の設定

PWMモジュール・レジスタ (PWM0, PWM1) は、PWM出力のデューティ値を決定する16ビットのレジスタです。

PWMモジュール・レジスタ (PWM0, PWM1) のビット4-15の内容が12ビットPWMデューティ値を決定します (12ビット分解能)。ビット0-3は意味を持たず、1または0のいずれのデータを書き込んでもPWM出力に影響しません。

注意1 . PWMモジュール・レジスタ (PWM0, PWM1) へは、0000H-00FFHの値を設定しないでください。

PWMレジスタ (PWM0, PWM1) へは0100H-FFFFHの値を設定するようにしてください。出力可能なPWM信号のデューティ値は、17/4096-4096/4096となります。

2 . $\overline{\text{RESET}}$ 入力により、モジュール・レジスタの内容は不定になりますので、イニシャライズ・プログラムでデータを設定してから、PWM出力を許可してください。

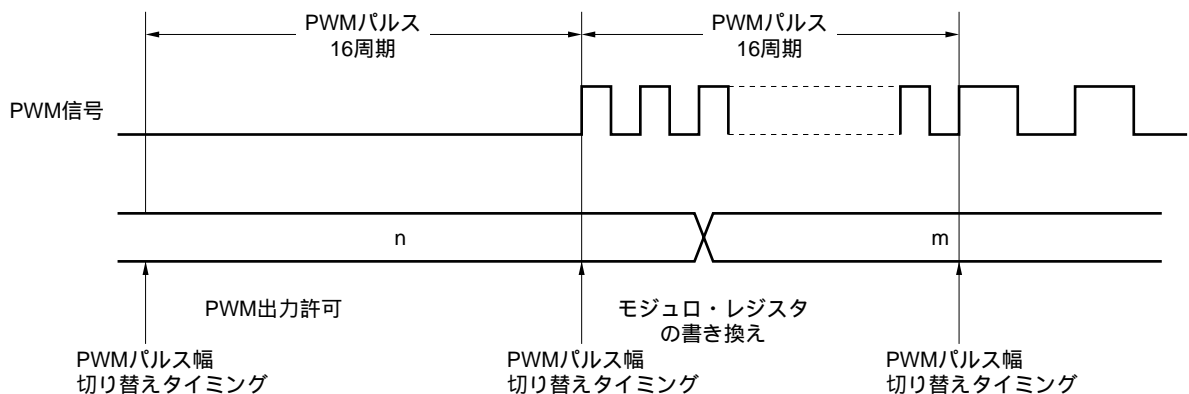
(4) PWMパルス幅書き換え周期の指定

PWM出力の開始、およびデューティ値の変更は、PWMパルス16周期 ($2^{12}/f_{\text{PWM}}$) ごと、あるいは、PWMパルス1周期 ($2^8/f_{\text{PWM}}$) ごとの、いずれかに同期して行われます。このPWMパルス幅の書き換え周期は、PWMコントロール・レジスタ (PWMCR) のSYNn (n = 0, 1) フラグで指定します。したがって、次に示す周期でPWM出力のデューティ値が切り替わることになります。

$2^{12}/f_{\text{CLK}}$	$327.7 \mu\text{s}$ ($f_{\text{CLK}} = 12.5 \text{ MHz}$)
	または
$2^8/f_{\text{CLK}}$	$20.5 \mu\text{s}$ ($f_{\text{CLK}} = 12.5 \text{ MHz}$)

このときのPWM出力タイミング例を、図7-3に示します。

図7-3 PWM出力タイミング例 (PWMパルス幅書き換え周期 $2^{12}/f_{\text{PWM}}$)



注意1 . パルス幅は、PWMパルス16周期ごとに書き換えられます。

2 . PWM信号は、12ビット分解能。

7.2 PWM出力ユニットのプログラム例

(1) 処理概要

(a) 仕様

PWM0端子から、デューティ80%のPWM信号を出力するプログラム例を紹介します。

(b) 使用周辺機能説明

本仕様を実現するためのPWMモジュロ・レジスタ (PWM0) への設定値は、次のようになります。

$$\begin{aligned} \text{PWM信号出力のデューティ} &= \frac{(\text{PWMnのビット4-15の値}) + 1}{4096} \\ 0.8 &= \frac{X + 1}{4096} \\ (4096 \times 0.8) - 1 &= 3275.8 \\ &3276 \text{ (CCCH)} \\ &\text{PWMnのビット4-15の値} \end{aligned}$$

(2) 使用RAM領域

なし

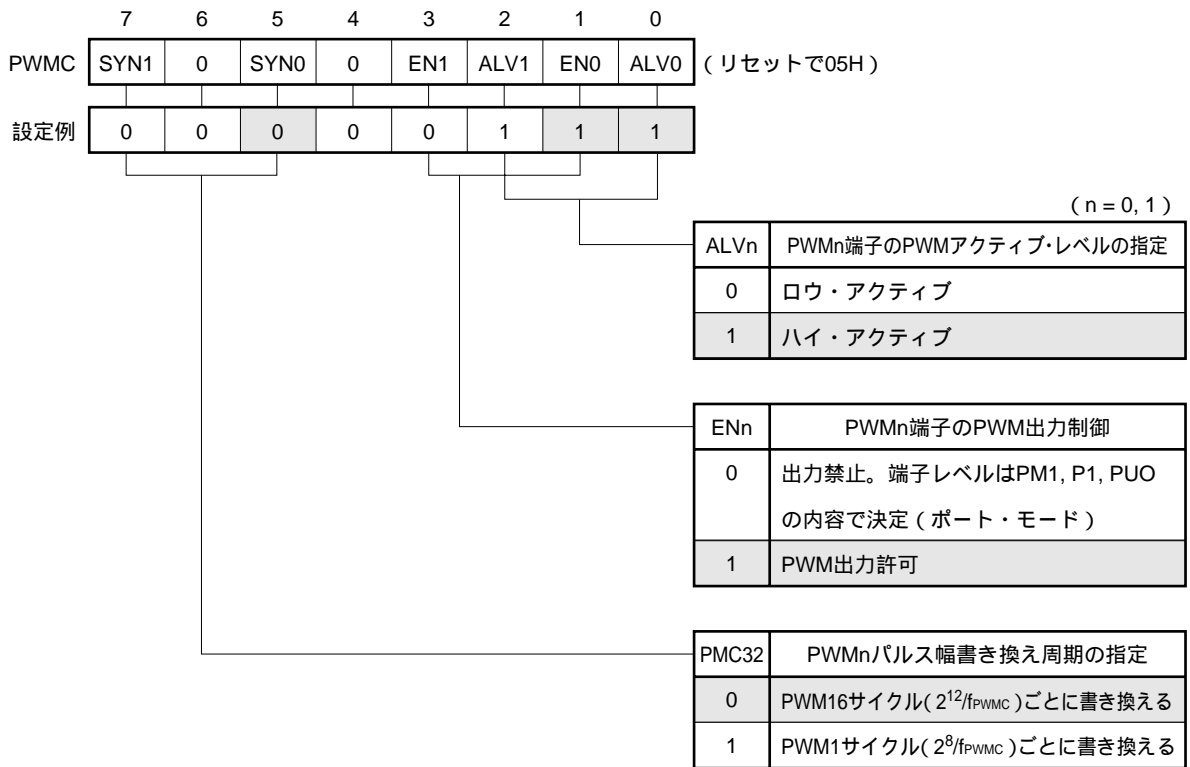
(3) 使用レジスタ

(a) 汎用レジスタ

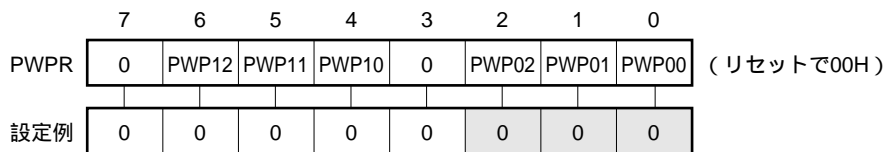
なし

(b) 特殊機能レジスタ

PWMコントロール・レジスタ (PWMC)



PWMプリスケアラ・レジスタ (PWPR)



(n = 0, 1)

PWPn2	PWPn1	PWPn0	PWMnの動作クロック (f_{PWC})	PWMnの繰り返し周波数 ($f_{CLK} = 12.5 \text{ MHz}$)
0	0	0	f_{CLK}	$f_{CLK}/256$ (48.8 kHz)
0	0	1	f_{CLK}	$f_{CLK}/256$ (48.8 kHz)
0	1	0	$f_{CLK}/2$	$f_{CLK}/512$ (24.4 kHz)
0	1	1	$f_{CLK}/3$	$f_{CLK}/768$ (16.30 kHz)
1	0	0	$f_{CLK}/4$	$f_{CLK}/1024$ (12.2 kHz)
上記以外			設定禁止	

(4) 入力方法

なし

(5) 出力方法

なし

(6) プログラム説明

- (a) PWMコントロール・レジスタ (PWMC) でPWMの動作状態を設定します。
PWMアクティブ・レベル : ハイ・アクティブに設定します。
PWMパルス幅書き換え周期 : PWM16周期 ($2^{12}/f_{PWC}$) ごとに書き換えるように設定します。
PWM0端子のPWM出力制御 : PWM出力禁止
- (b) ポートの出力ラッチにインアクティブ・レベルを設定します。
- (c) P10を出力ポート・モードに設定します。
- (d) PWMプリスケアラ・レジスタ (PWPR) で, PWM出力回路の動作クロックを選択します。
- (e) 出力させるPWMパルス幅を, PWMモジュロ・レジスタ (PWM0) に設定します。
- (f) PWMコントロール・レジスタ (PWMC) のEN0フラグでPWM出力を許可状態に設定します。

(7) フロー・チャート

プログラムのステップ数が少ないため, 省略します。

(8) プログラム・リスト

本アプリケーション・ルーチンのプログラム・リスト

```
      .  
      .  
MOV    PPMC,#00000101B      ; high active, 16 cycle, disable output  
CLR1   P1.0                 ; set P10 inactive level  
CLR1   PM1.0               ; set P10 output mode  
MOV    PWPR,#00H           ; set PWM CLOCK fCLK  
MOVW   PWM0,#0CCC0H        ; set PWM0 duty 80%  
SET1   EN0                 ; PWM enable output  
      .  
      .  
END
```

〔メ モ〕

第8章 イニシャライズ処理

プログラムを作成する際、リセット・スタート後の最初の処理として、各機能（モード・レジスタの設定、RAMのイニシャライズなど）のイニシャライズ処理を行います。

注意 アプリケーションに掲題していますプログラム例は、使用する機能、システム状態を仮定した状態でイニシャライズ処理をしております。実際にプログラムを設計される際は、システム、仕様にあわせて設定値をユーザーズ・マニュアルでご確認のうえ、イニシャライズ処理を作成してください。

イニシャライズ処理の内容を次に示します。

割り込みベクタ処理

LOCATIONの設定

スタック・ポインタ（SP）の設定

割り込み禁止（DI）状態に設定

レジスタ・バンクの設定

CPUクロックを低速に設定

不定状態（ポート状態など）の確定

システム、仕様に応じた各機能のイニシャライズ

RAMのイニシャライズ

ここでは、各機能別に設定例を説明します。

8.1 割り込みベクタ・アドレスの設定

すべての割り込みのベクタ・アドレスを定義します。

使用しない割り込みベクタ・テーブルにも特定のアドレスを定義することにより、プログラム暴走時にシステムの停止や再スタートさせることができます。

なお、コンテキスト・スイッチングを使用する割り込みは、レジスタ・バンクの番号を指定します。

・プログラム例

```

;
;
; *****
; ***          vector table          ***
; *****
;
;
RSTVT  CSEG      AT 00000H
        DW      INTM          ; reset
        DW      INTR          ; NMI          *
        DW      INTR          ; WDT          *
        DW      INTR          ; INTP0        *
        DW      INTR          ; INTP1        *
        DW      INTR          ; INTP2        *
        DW      INTR          ; INTP3
        DW      USERB4       ; INTC00     *
        DW      INTC01       ; INTC01     *   使用する割り込み
        DW      INTR          ; INTC10     *
        .
        .

```

備考 * : 使用しない割り込み

INTM : リセット処理のアドレス

8.2 SP（スタック・ポインタ）/RB（レジスタ・バンク）/STBC（CPUクロック）のイニシャライズ

プログラムを作成する際、マイコンのイニシャライズ処理を行います。一般的には、次に示す手順で、スタック・ポインタ（SP）、レジスタ・バンク（RB_n）、CPUクロック（STBC）の設定を行います。

（a）μPD784038, 784038Yサブシリーズでは、LOCATION命令でSFRのアドレスを定義します。

- ・LOCATION 0 : 内部RAM領域, SFR領域は、0F700H-0FFFFH範囲になるようにマッピングされます。
- ・LOCATION 0FH : 内部RAM領域, SFR領域は、0FF700H-0FFFFFFH範囲になるようにマッピングされます。

注意 LOCATION命令は、リセット解除直後に必ず実行してください。2回以上使用することはありません。

（b）スタック・ポインタを設定します。

スタック・ポインタの内容は、リセット入力により不定になりますので、リセット解除後に必ずイニシャライズ・プログラムによりスタック・ポインタをイニシャライズしてください。

また、必ずLOCATON命令の次に実行してください。

（c）割り込みを禁止状態（DI）に設定します。

（d）アプリケーション・プログラムで使用するレジスタ・バンクを設定します。

（e）動作環境にあわせて、スタンバイ・コントロール・レジスタ（STBC）でCPUクロックを設定します。

（f）ポートのイニシャライズ処理を行います。

（i）アプリケーション仕様にあわせて、ポートの出力ラッチの値をイン・アクティブに設定します。

（ii）アプリケーション仕様にあわせて、ポートの入出力状態を設定します。

注意 リセット後、ポートの出力ラッチの内容は不定になりますので、不定なデータが出力されないように、必ず先に出力ラッチの内容をイニシャライズしてから、入出力状態の設定を行ってください。

（iii）アプリケーション仕様にあわせて、ポートに内部プルアップ抵抗を接続します。

（iv）リアルタイム出力ポートのモード・レジスタを設定します。

・プログラム例

```

;
; *****
; ***          initialize routine          ***
; *****
;
;
INTM_C  CSEG
INTM:
    LOCATION 0H                ; set LOCATION
    MOVG     SP,#STACK         ; set stack pointer
    DI                          ; interrupt DI
    SEL      RBO                ; set register bank
    MOV      STBC,#00000000B   ; set cpu clock fxx/12.5MHz
;
; ***** Port initialize *****
;
;
    MOV      MM,#10000000B     ; set memory mapping register
;
    MOV      P0,#00000001B     ; PORT INACTIVE SET
    MOV      P1,#00000000B     ;
    MOV      P3,#00000000B     ;
    MOV      P6,#00000000B     ;
    MOV      P7,#00000000B     ;
;
    MOV      PM0,#00000000B    ; Port mode set
    MOV      PM1,#00000000B    ;
    MOV      PM3,#00100001B    ;
    MOV      PM6,#00000000B    ;
    MOV      PM7,#11111000B    ;
;
    MOV      PU0,#00000100B    ; pull up set
;
    MOV      RTPC,#00000000B   ; real-time mode

```

8.3 使用しない周辺機能のイニシャライズ処理

ハードウェア・リセットがかかると、周辺機能の特殊機能レジスタ関係はすべてリセット値が設定されますが、ここではソフトウェアで、使用しない周辺機能のイニシャライズ処理を行います。

次にA/Dコンバータと、D/Aコンバータの例を示します。

・プログラム例

```

;
; ***** A/D initialize *****
;
;
    MOV      ADM,#00H          ;
;
    CLR1     ADIF              ; INTAD request flag CLR
;
;
; ***** D/A initialize *****
;
;
    MOV      DAM,#00H         ;
;
;

```

8.4 RAMクリア処理

- (a) RAM領域を0でイニシャライズします。
 (b) 使用するRAM領域で初期値データが0以外のRAM領域に設定値を設定します。

設定値：MODE0の設定値を8にする
 HTENFLAGの設定値を1にする

・プログラム例

```

;
; *****          RAM CLR          *****
; ***      00FE00H-00FE7FH RAM CLR      ***
; ***      (short direct addressing1)***
;
;
;      MOVG    TDE,#0FE7FH                ;
;      MOV     A,#00H                      ;
;      MOV     C,#127                      ;
RAMCLR1:
;      MOV     [TDE-],A                    ;
;
; *****          RAM CLR          *****
; ***      00FD20H-00FDFFH RAM CLR      ***
; ***      (short direct addressing2)***
;
;      MOV     C,#223                      ;
RAMCLR2:
;      MOV     [TDE-],A                    ;
;
; *****          RAM initialize data set *****
;
;      MOV     MODE0,#8                    ;
;      SET1    HTENFLAG                    ;

```

8.5 タイマのイニシャライズ処理

使用するタイマの時間を設定します。

- (a) タイマの動作を停止します (タイマ時間とカウント・クロックを設定するときは、いったんタイマ動作を停止させます)。
- (b) プリスケアラ・モード・レジスタ (PRM1) でカウント・クロックを設定します。
- (c) キャプチャ/コンペア・コントロール・レジスタ2 (CRC2) でタイマの動作の許可条件や、タイマ出力モードを指定します。
- (d) タイマ出力コントロール・レジスタ (TOC) でタイマ出力のアクティブ・レベル出力/禁止を設定します。
- (e) コンペア・レジスタに初期値を設定します。
- (f) タイマ・コントロール・レジスタ (TMCX) でタイマの動作を開始させます (TMCXでタイマ動作許可した時点からタイマのカウントが開始されますので、タイマに関するすべてのモード・レジスタの設定が終わってからタイマ動作を許可するようにしてください)。

・プログラム例

```

;
; ***** Timer initialize *****
;
;
;
;
; ///// main timer active /////
;
MOV    TMC1,#00H                ;
MOV    PRM1,#00010100B          ; fxx/64 (2.56)
MOV    CRC1,#00011000B          ;
MOV    TOC,#00001000B           ;
MOVW   CR11W,#0F41H             ; set MAIN timer 10ms
MOV    TMC1,#00001001B          ; TM1 START
;
; ///// timer active /////
;
MOV    TMC0,#00H                ;
MOV    PRM0,#00010011B          ; fxx/32 (1.28)
MOV    CRC0,#00011000B          ;
OR     TOC,#00000010B           ;
MOVW   CR01,#0F423H             ; set timer 80ms
MOV    TMC0,#00001000B          ; TMO START

```

8.6 ウォッチドッグ・タイマの設定

(a) 使用するウォッチドッグ・タイマの時間を設定します。

注意1．ウォッチドッグ・タイマ・モード・レジスタ (WDM) は、専用命令 (MOV WDM, # byte) でのみ書き込みが可能です。他の書き込み命令は無視され、何も動作しません。

2．プログラムのメイン・ルーチンの中で、ウォッチドッグ・タイマのオーバフロー設定時間以内に、ウォッチドッグ・タイマをクリアします。

・プログラム例

```

;
; //////////////// watchdog timer ////////////////
;
MOV     WDM,#10000000B           ; set watchdog timer 10.49ms
;

```

8.7 コンテキスト・スイッチングのイニシャライズ処理

コンテキスト・スイッチングは、割り込みが受け付けられると、ハードウェアにより所定のレジスタ・バンクを選択し、レジスタ・バンク内にあらかじめ設定しておいたベクタ・アドレスに分岐すると同時に、現在のPC、PSWをレジスタ・バンク内に退避する機能です。

使用する割り込みの処理形態をコンテキスト・スイッチングに設定した場合、次に示す設定を行います。

- (a) コンテキスト・スイッチング機能で使用するバンクを設定します。
- (b) コンテキスト・スイッチングの要求があったときの分岐先アドレスを、RP2レジスタに設定します。
- (c) メイン・ルーチンで使用するレジスタ・バンクに戻します。
- (d) 使用する割り込みの処理形態をコンテキスト・スイッチング処理に指定します (割り込みを許可する前に必ず設定してください)。

・プログラム例

```

;
; ***** context initialize *****
;
USERB4 EQU     4           ;use bank4
;
CONTSET:
    SEL     RB4           ;context bank4 select
    MOVW   RP2,#CONTEXT   ;
    SEL     RBO           ;
    MOV    PIC3,#01010011B ;context disposition designate
;

```

8.8 割り込みのイニシャライズ処理

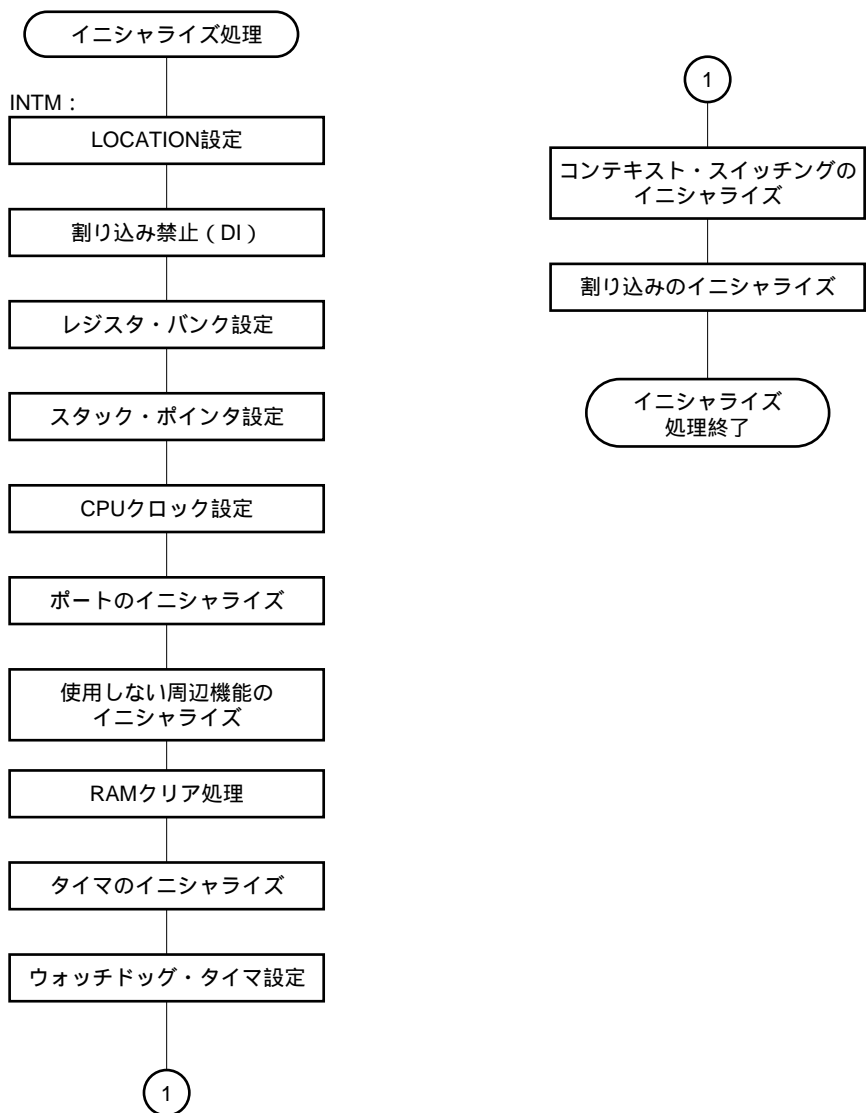
使用するレジスタの割り込みのイニシャライズを行います。

- (a) すべての割り込みのマスクを禁止状態に設定します。
- (b) 使用する割り込みのモードを設定します。
- (c) 使用する割り込みのマスクを許可状態に設定します。

・プログラム例

```
;  
;  
; ***** interrupt initialize *****  
;  
;  
;          MOVW    MK0,#0FFFFH          ;  
;          MOV     MK1L,#0FFH          ;  
;  
;          ;  
; ///// interrupt mode set /////  
;          ;  
;          MOV     CIC01,#01000000B    ; INTC01 vector interrupt priority3  
;          ;  
; ///// interrupt enable /////  
;          ;  
;          CLR1   PMK3                  ; INTP4 MASK EI  
;          CLR1   CMK01                 ; INTC01 MASK EI  
;  
;
```

フロー・チャート



< 使用しない周辺機能の特殊機能レジスタのイニシャライズ >

```

MOV     ADM,#00H           ;
CLR1    ADIF              ; INTAD request flag CLR
MOV     DAM,#00H         ;
      .
      .

;
; *****          RAM CLR          *****
; ***      00FE00H-00FE7FH RAM CLR      ***
; ***      (short direct addressing1)***
;
;
MOV     TDE,#0FE7FH       ;
MOV     A,#00H           ;
MOV     C,#127           ;
RAMCLR1:
MOV     [TDE-],A         ;
;
; ***      00FD20H-00FDFFH RAM CLR      ***
; ***      (short direct addressing2)***
;
;
MOV     C,#223           ;
RAMCLR2:
MOV     [TDE-],A         ;
;
; ***** RAM initialize data set *****
;
MOV     MODE0,#8         ;
SET1    HTENFLAG        ;
;
; ***** Timer initialize *****
;
;
; ///////////////////////////////////////////////////////////////////
; ///////////////////////////////////////////////////////////////////
MOV     TMC1,#00H        ;
MOV     PRM1,#00010100B  ; fxx/64 (2.56)
MOV     CRC1,#00011000B  ;
MOV     TOC,#00001000B   ;
MOVW    CR11W,#0F41H     ; set MAIN timer 10ms
MOV     TMC1,#00001001B  ; TM1 START
;
; ///////////////////////////////////////////////////////////////////
; ///////////////////////////////////////////////////////////////////
MOV     TMC0,#00H        ;
MOV     PRM0,#00010011B  ; fxx/32 (1.28)
MOV     CRC0,#00011000B  ;
OR      TOC,#00000010B   ;
MOVW    CR01,#0F423H     ; set timer 80ms
MOV     TMC0,#00001000B  ; TMO START
;
; ***** WATCHDOG TIMER *****
;
MOV     WDM,#10000000B   ; set watchdog timer 10.49ms

```


【メ モ】

— お問い合わせ先 —

【技術的なお問い合わせ先】

NEC半導体テクニカルホットライン
(電話：午前 9:00～12:00，午後 1:00～5:00)

電話 : 044-435-9494
FAX : 044-435-9608
E-mail : s-info@saed.tmg.nec.co.jp

【営業関係お問い合わせ先】

第一販売事業部

東京 (03)3798-6106, 6107,
6108

名古屋 (052)222-2375

大阪 (06)6945-3178, 3200,
3208, 3212

仙台 (022)267-8740

郡山 (024)923-5591

千葉 (043)238-8116

第二販売事業部

東京 (03)3798-6110, 6111,
6112

立川 (042)526-5981, 6167

松本 (0263)35-1662

静岡 (054)254-4794

金沢 (076)232-7303

松山 (089)945-4149

第三販売事業部

東京 (03)3798-6151, 6155, 6586,
1622, 1623, 6156

水戸 (029)226-1702

広島 (082)242-5504

高崎 (027)326-1303

鳥取 (0857)27-5313

太田 (0276)46-4014

名古屋 (052)222-2170, 2190

福岡 (092)261-2806

【資料の請求先】

上記営業関係お問い合わせ先またはNEC特約店へお申しつけください。

【インターネット電子デバイス・ニュース】

NECエレクトロニクスデバイスの情報がインターネットでご覧になれます。

URL(アドレス)

<http://www.ic.nec.co.jp/>

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] μPD784038, 784038Yサブシリーズ ハードウェア基礎編
(U13285JJ1V0ANJ1 (第1版))

[お名前など] (さしつかえのない範囲で)

御社名(学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価(各欄に をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ()					
()					

2. わかりやすい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所(第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは
NEC販売員, 特約店販売員, その他 ()

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡ししてください。

日本電気(株)NECエレクトロニクス
半導体テクニカルホットライン

FAX: (044) 435-9608

2000.6