

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。

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

ご注意

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

H8/300Lシリーズ

アプリケーションノート

H8/3644シリーズ
H8/3657シリーズ

ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等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。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及びその他諸条件につきましては、弊社保証範囲内でご使用いただきますようお願い致します。
保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。
また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

はじめに

H8/3644シリーズおよびH8/3657シリーズは、高速H8/300L CPUを核にして、システム構成に必要な周辺機能を集積したシングルチップマイクロコンピュータです。

H8/300L CPUは、H8/300 CPUと互換性のある命令体系を備えています。

H8/3644シリーズおよびH8/3657シリーズは、システム構成に必要な周辺機能として、5種類のタイマ、14ビットPWM、2チャンネルのシリアルコミュニケーションインタフェース、A/D変換器を内蔵しており、高度な制御システムの組込み用マイコンとして活用できます。

H8/300Lシリーズ -H8/3644、H8/3657- アプリケーションノートは、H8/3644シリーズおよびH8/3657シリーズの内蔵周辺機能を単独で使用した場合の動作例を示した“基礎編”により構成されており、ユーザにてソフトウェア設計およびハードウェア設計の際、ご参考として役立てていただけるようにまとめたものです。

なお、本アプリケーションノートに掲載されているプログラム、回路等の動作は確認しておりますが、実際にご使用になる場合は、必ず動作確認の上ご使用くださいますようお願い致します。

目次

1. H8/300Lシリーズ -H8/3644、H8/3657-アプリケーションノート使用手引	1
1.1 基礎編構成	2
2. 基礎編	4
2.1 インターバル機能による8ビットカウンタのカウントアップ (タイマA)	5
2.2 時計用タイムベース機能によるLEDの点滅動作 (タイマA)	11
2.3 クロック出力機能によるクロック出力 (タイマA)	18
2.4 オートリロードタイマ機能による割込み周期設定 (タイマB1)	24
2.5 イベントカウンタ機能によるパルスの周波数測定 (タイマB1)	33
2.6 コンペアマッチ機能によるPWM出力 (タイマV)	42
2.7 トリガ入力カウント開始機能によるパルス出力 (タイマV)	52
2.8 16ビットフリーランニング機能による割込み回数のカウント (タイマX)	60
2.9 16ビットイベントカウンタ機能による入力パルス数の測定 (タイマX)	67
2.10 アウトプットコンペア機能による任意の位相差のパルス出力 (タイマX)	74
2.11 インプットキャプチャ機能によるパルスの周期測定 (タイマX)	84
2.12 ウォッチドッグタイマ (ウォッチドッグタイマ)	94
2.13 クロック同期式シリアルデータ送信 (SCI1)	102
2.14 クロック同期式シリアルデータ受信 (SCI1)	110
2.15 クロック同期式シリアルデータ同時送受信 (SCI1)	118
2.16 SSB通信 (SCI1)	127
2.17 調歩同期式シリアルデータ送信 (SCI3)	137
2.18 調歩同期式シリアルデータ受信 (SCI3)	148
2.19 調歩同期式シリアルデータ同時送受信 (SCI3)	161
2.20 マルチプロセッサ通信 (SCI3)	175
2.21 4チャンネルA/D変換による電圧測定 (A/Dコンバータ)	187
2.22 14ビットPWM機能によるデューティパルス出力 (14ビットPWM)	193
2.23 I/Oポートに接続したLEDの点滅動作 (I/Oポート)	199
2.24 外部割込みによるカウントスタート (割込み)	206
2.25 内部割込みによる多重割込み動作 (割込み)	214
2.26 スリープ (中速) モードへの遷移 (低消費電力モード)	223
2.27 スリープ (高速) モードへの遷移 (低消費電力モード)	233
2.28 スタンバイモードへの遷移 (低消費電力モード)	242
2.29 ウォッチモードへの遷移 (低消費電力モード)	253
2.30 サブスリープモードへの遷移 (低消費電力モード)	263
2.31 サブアクティブモードへの遷移 (低消費電力モード)	274
2.32 アクティブ (中速) モードへの遷移 (低消費電力モード)	285

1. H8/300Lシリーズ -H8/3644、H8/3657- アプリケーションノート使用手引

目次

1.1 基礎編構成

2

本アプリケーションノートは、図1に示すように2部構成になっています。



図1 アプリケーションノート構成

(1)H8/300Lシリーズ-H8/3644、H8/3657-アプリケーションノート使用手引
H8/300Lシリーズ-H8/3644、H8/3657-アプリケーションノートの使用法について説明しています。

(2)基礎編
H8/3644シリーズ、H8/3657シリーズの内蔵周辺機能（タイマ、シリアルコミュニケーションインタフェース、A/Dコンバータ、PWM、I/Oポート、割込み、低消費電力モード等）の使用法を簡単なタスク例をもとに説明しています。

1.1 基礎編構成

基礎編は図2に示す構成で内蔵周辺機能の使用法について説明しています。

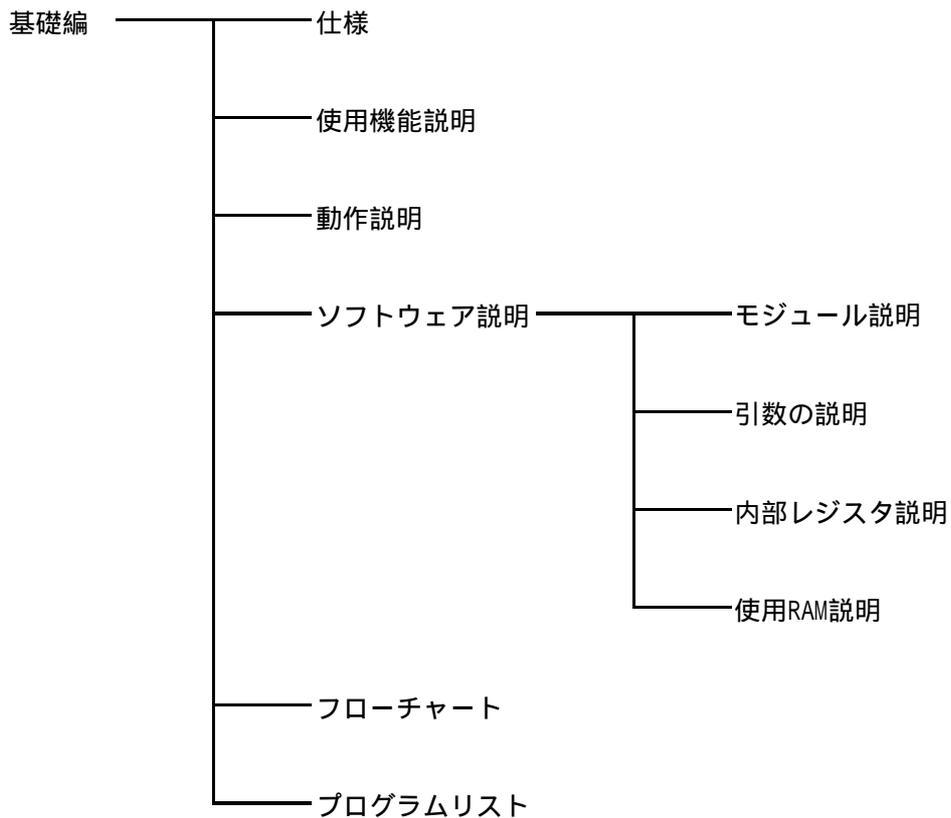


図2 基礎編構成

- (1) 仕様
タスク例のシステム仕様について説明しています。
- (2) 使用機能説明
タスク例で使用する周辺機能の特徴および周辺機能の割付けについて説明しています。
- (3) 動作説明
タスク例の動作をタイミングチャートを使用し説明しています。
- (4) ソフトウェア説明
 - (a) モジュール説明
タスク例を動作させるソフトウェアのモジュールについて説明しています。
 - (b) 引数の説明
モジュールを実行する際に必要な入力引数と、実行後の出力引数について説明しています。
 - (c) 内部レジスタ説明
モジュールで設定する周辺機能の内部レジスタ（タイマコントロールレジスタ、シリアルモードレジスタ等）について説明しています。
 - (d) 使用RAM説明
モジュールで使用するRAMのラベル名及び機能について説明しています。
- (5) フローチャート
タスク例を実行するソフトウェアについてフローチャートを使用し説明しています。
- (6) プログラムリスト
タスク例を実行するソフトウェアのプログラムリストを示しています。

2. 基礎編

目次

2.1	インターバル機能による8ビットカウンタのカウントアップ	5
2.2	時計用タイムベース機能によるLEDの点滅動作	11
2.3	クロック出力機能によるクロック出力	18
2.4	オートリロードタイマ機能による割込み周期設定	24
2.5	イベントカウンタ機能によるパルスの周波数測定	33
2.6	コンペアマッチ機能によるPWM出力	42
2.7	トリガ入力カウンタ開始機能によるパルス出力	52
2.8	16ビットフリーランニング機能による割込み回数のカウント	60
2.9	16ビットイベントカウンタ機能による入力パルス数の測定	67
2.10	アウトプットコンペア機能による任意の位相差のパルス出力	74
2.11	インプットキャプチャ機能によるパルスの周期測定	84
2.12	ウォッチドッグタイマ	94
2.13	クロック同期式シリアルデータ送信	102
2.14	クロック同期式シリアルデータ受信	110
2.15	クロック同期式シリアルデータ同時送受信	118
2.16	SSB通信	127
2.17	調歩同期式シリアルデータ送信	137
2.18	調歩同期式シリアルデータ受信	148
2.19	調歩同期式シリアルデータ同時送受信	161
2.20	マルチプロセッサ通信	175
2.21	4チャンネルA/D変換による電圧測定	187
2.22	14ビットPWM機能によるデューティパルス出力	193
2.23	I/Oポートに接続したLEDの点滅動作	199
2.24	外部割込みによるカウントスタート	206
2.25	内部割込みによる多重割込み動作	214
2.26	スリープ(中速)モードへの遷移	223
2.27	スリープ(高速)モードへの遷移	233
2.28	スタンバイモードへの遷移	242
2.29	ウォッチモードへの遷移	253
2.30	サブスリープモードへの遷移	263
2.31	サブアクティブモードへの遷移	274
2.32	アクティブ(中速)モードへの遷移	285

2.1 インターバル機能による8ビットカウンタのカウントアップ

インターバル機能による 8ビットカウンタのカウントアップ	使用機能	タイマA：インターバル機能
<p>仕様</p> <p>(1) タイマAのインターバル機能を内蔵した8ビットのタイマを使用して、R1Lに設定した8ビットのカウンタのカウントアップを行ないます。</p> <p>(2) タイマカウンタA (TCA) のオーバーフローによりタイマA割込みを発生させ、タイマA割込み処理の中でR1Lに設定したカウンタのカウントアップ、またはイニシャライズを行ないます。</p> <p>(3) R1Lに設定するカウンタは、8ビットのカウンタとし、初期値のH'00からカウントアップを始めて、H'FFまでカウントを行なうと、H'00にイニシャライズして再びカウントアップを継続します。</p> <p>(4) タイマA割込みは、209.7152msごとに発生するように設定します。</p>		

使用機能説明
<p>(1) 本タスク例では、タイマAインターバル機能を使用して、8ビットカウンタのカウントアップを行ないます。図1にタイマAインターバル機能のブロック図を示します。以下にタイマAインターバル機能のブロック図について説明します。</p> <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・PSSは、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・タイマモードレジスタA (TMA) は、8ビットのリード/ライト可能なレジスタで、プリスケアラ、入力クロックの選択を行ないます。本タスク例では、プリスケアラにPSSを、プリスケアラ分周比に4096分周を選択しています。 ・タイマカウンタA (TCA) は、8ビットのリード可能なアップカウンタで、入力する内部クロックによりカウントアップされます。TCAがオーバーフローすると、割込み要求レジスタ1 (IRR1) のタイマAオーバーフロー割込み要求 (IRR1TA) が"1"にセットされます。 ・タイマAオーバーフロー割込み要求フラグ (IRR1TA) は、TCAがオーバーフローすることにより"1"にセットされます。IRR1TAが"1"にセットされていて、割込み許可レジスタ1 (IENR1) のタイマA割込みイネーブル (IENTA) が"1"で、かつコンディションコードレジスタ (CCR) の1ビットが"0"にクリアされている場合にタイマA割込みが受け付けられ、タイマA割込み処理を開始します。 ・以下に、本タスク例におけるTCAのオーバーフロー周期の計算法を示します。

$$\begin{aligned}
 \text{TCAオーバーフロー周期} &= \frac{1}{\text{システムクロック} / 4096} \times 256 \\
 &= 209.7152\text{ms}
 \end{aligned}$$

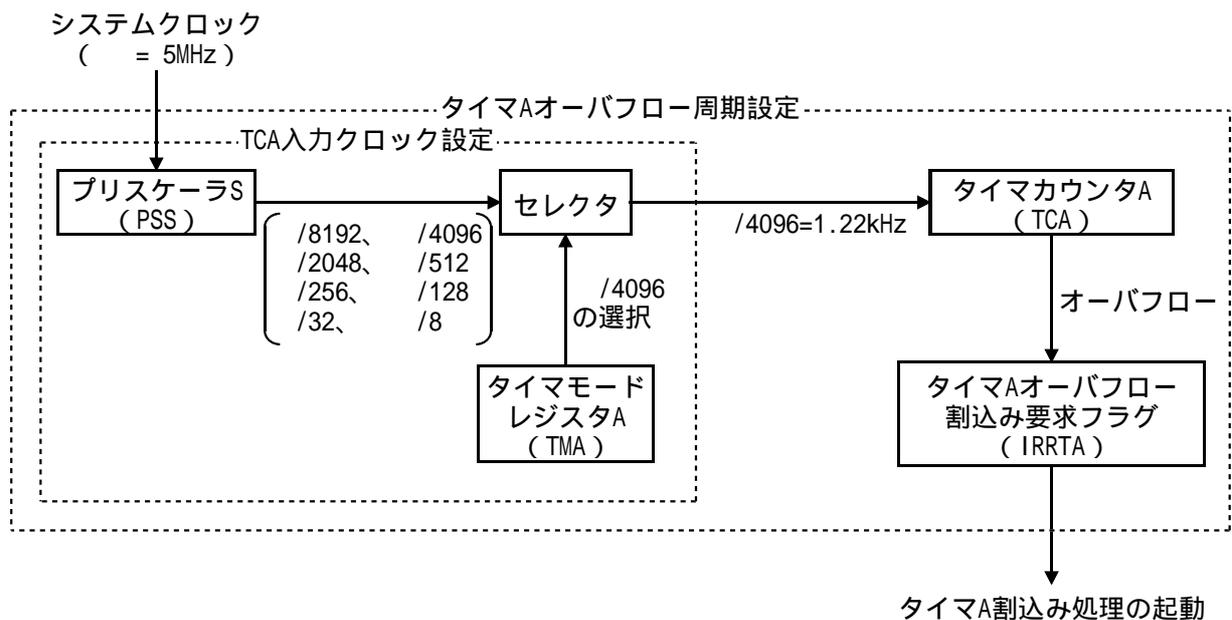


図1 タイマAインターバル機能ブロック図

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマAインターバル機能による8ビットカウンタのカウントアップを行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCA	システムクロックを4096分周したクロックを入力とする8ビットのカウンタ
TMA	PSSの選択、およびプリスケアラ分周比の設定
IRRTA	タイマA割込み要求の有無を反映

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマAインターバル機能による8ビットカウンタのカウントアップを行ないます。

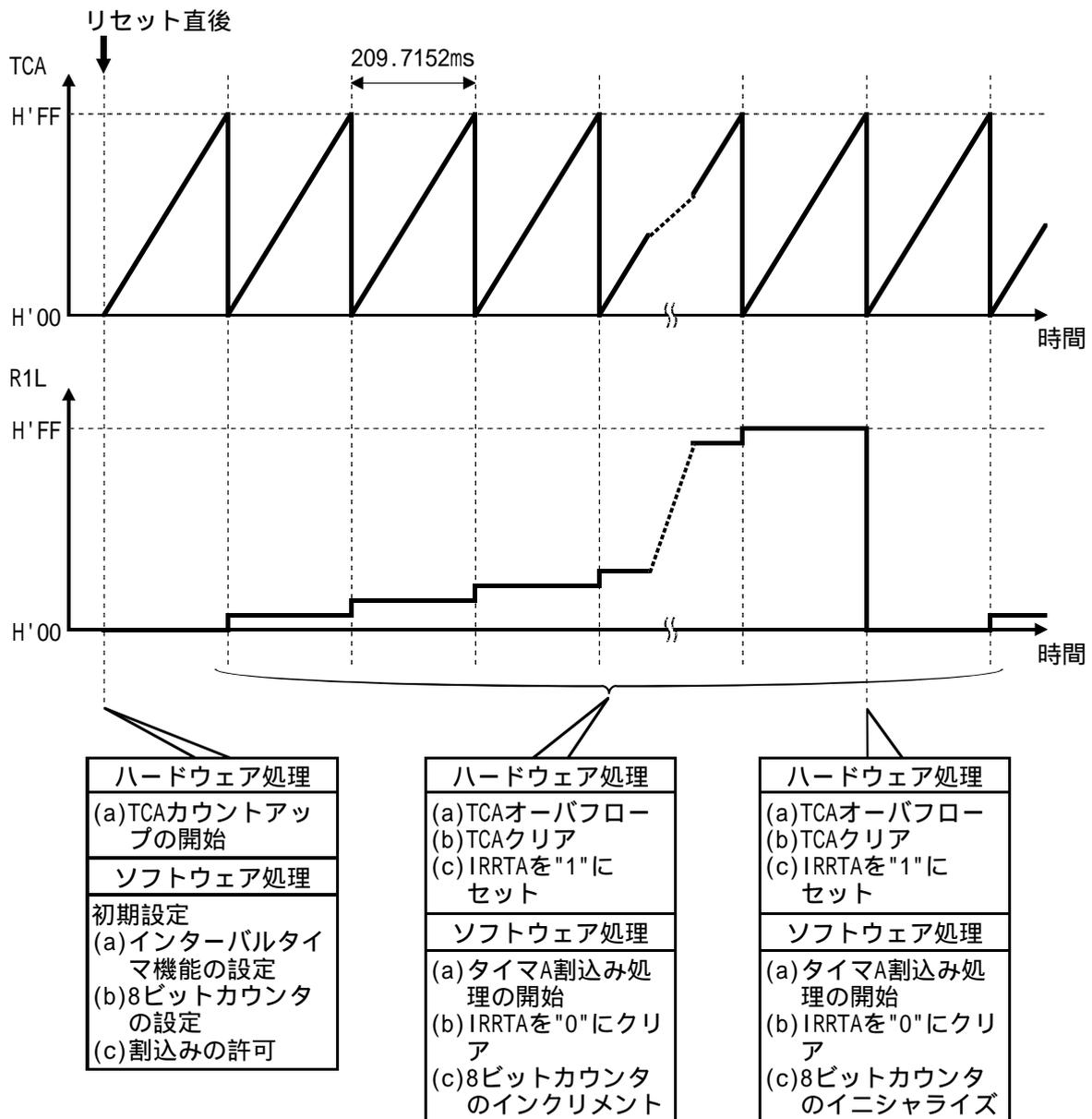


図2 タイマAインターバル機能による8ビットカウンタのカウントアップ動作の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、インターバルタイマの設定、8ビットカウンタの設定、割込みの許可を行なう
カウントアップ	TAINT	タイマA割込み処理ルーチンで、R1Lに設定した8ビットカウンタのインクリメント、またはイニシャライズを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
IRR1 IRR1A	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRR1A=0のとき、タイマA割込みが要求されていない : IRR1A=1のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0
IENR1 IENR1A	割込み許可レジスタ1 (タイマA割込みイネーブル) : IENR1A=1のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
TMA	タイマモードレジスタA : TMA=H'11のとき、タイマA機能をインターバル機能に、TCA入力クロックソースをPSSに、プリスケラ分周比を4096分周に設定	H'FFB0	H'11
TCA	タイマカウンタA : システムクロックを4096分周したクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00

(4) 使用RAM説明

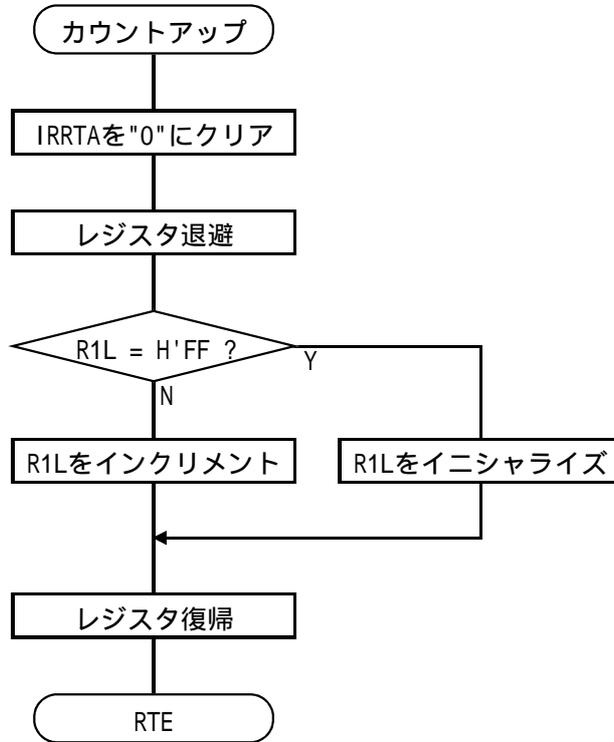
本タスク例では、RAMは使用していません。

フローチャート

(a) メインルーチン



(b) タイマA割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       '8-bit Counter Count-Up by Interval Function'
;
;       Function
;       : Timer A Interval Timer
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
TMA      .equ          H'FFB0      ;Timer Mode Register A
TCA      .equ          H'FFB1      ;Timer Counter A
IENR1    .equ          H'FFF4      ;Interrupt Enable Register 1
IENTA    .bequ         6,IENR1     ;Timer A Interrupt Enable
IRR1     .equ          H'FFF7      ;Interrupt Request Register 1
IRRTA    .bequ         6,IRR1      ;Timer A Interrupt Request Flag
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;
STACK    .equ          H'FF80      ;Stack Pointer
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.W       MAIN          ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.W       MAIN          ;IRQ0 Interrupt
;       .data.W       MAIN          ;IRQ1 Interrupt
;       .data.W       MAIN          ;IRQ2 Interrupt
;       .data.W       MAIN          ;IRQ3 Interrupt
;       .data.W       MAIN          ;INT0 - INT7 Interrupt
;
;
;       .org          H'0014
;       .data.W       TAINT         ;Timer A Interrupt
;       .data.W       MAIN          ;Timer B1 Interrupt
;
;
;       .org          H'0020
;       .data.W       MAIN          ;Timer X Interrupt
;       .data.W       MAIN          ;Timer V Interrupt
;
;
;       .org          H'0026

```


2.2 時計用タイムベース機能によるLEDの点滅動作

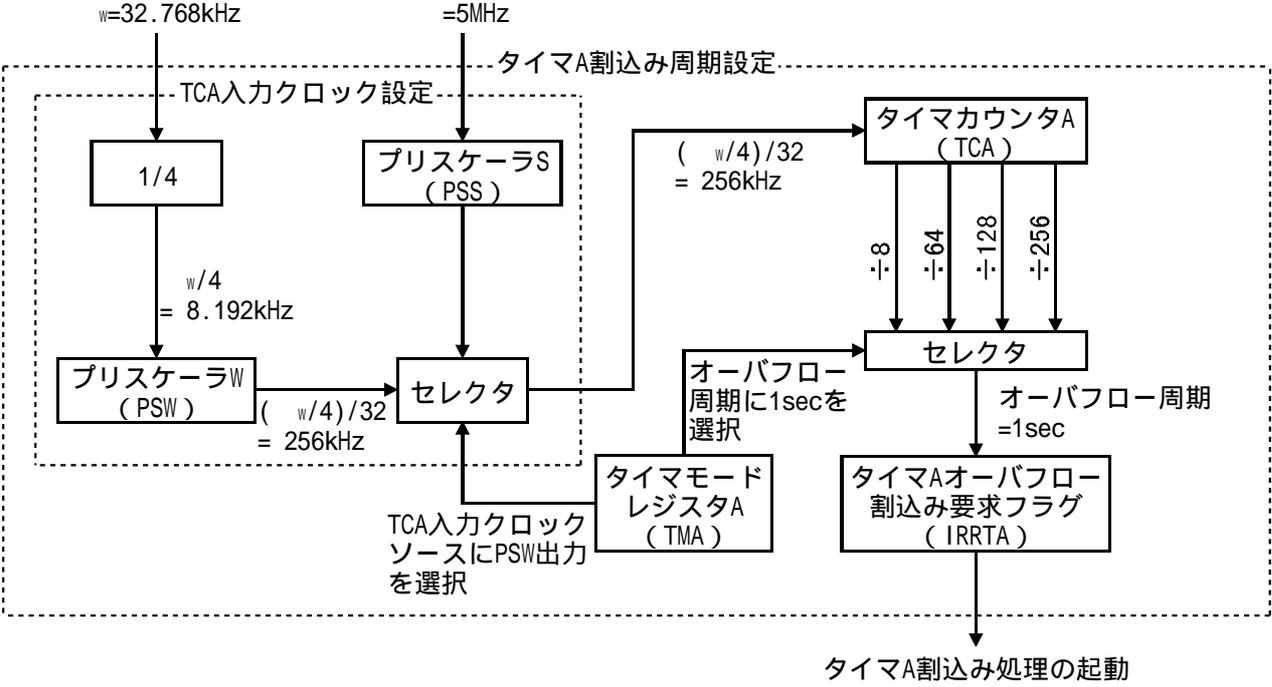
時計用タイムベース機能によるLEDの点滅動作	使用機能	タイマA：時計用タイムベース機能
<p>仕様</p> <p>(1) タイマAの時計用タイムベース機能を使用して、1secごとにLEDの点灯/消灯を交互に行ないます。 (2) タイマカウンタA (TCA) のオーバーフローによりタイマA割込みを発生させ、タイマA割込み処理の中でLEDの点灯/消灯の制御を行ないます。 (3) LEDはポート7のP7₃出力端子に接続しているものとします。 (4) タイマA割込みは時計用タイムベース動作により、1secごとに発生するように設定します。</p>		
<p>使用機能説明</p> <p>(1) 本タスク例では、タイマA時計用タイムベース機能を使用して、1secごとにLEDの点灯/消灯を行ないます。図1にタイマA時計用タイムベース機能のブロック図を示します。以下にタイマA時計用タイムベース機能のブロック図について説明します。</p> <ul style="list-style-type: none"> • ωは、サブクロック発振器の出力クロック (32.768kHz) です。 • プリスケアラW (PSW) は、32.768kHzを4分周したクロック ($\omega/4$) を入力とする5ビットのカウントで、分周した出力をタイマAの時計用タイムベース動作に使用します。 • タイマモードレジスタA (TMA) は、8ビットのリード/ライト可能なレジスタで、TCA入力クロックソース、およびTCAオーバーフロー周期の選択を行ないます。本タスク例では、TCAの入力クロックソースにPSWを、TCAオーバーフロー周期に1secを選択しています。 • タイマカウンタA (TCA) は、8ビットのリード可能なアップカウンタで、入力する内部クロックによりカウントアップされます。TCAがオーバーフローすると、割込み要求レジスタ1 (IRR1) のタイマAオーバーフロー割込み要求フラグ (IRRTA) が"1"にセットされます。 • IRRTAは、TCAがオーバーフローすることにより"1"にセットされます。IRRTAが"1"にセットされていて、割込み許可レジスタ1 (IENR1) のタイマA割込みイネーブル (IENTA) が"1"で、かつコンディションコードレジスタ (CCR) の1ビットが"0"にクリアされている場合にタイマA割込みが受け付けられ、タイマA割込み処理を開始します。  <p style="text-align: center;">タイマA割込み処理の起動</p>		

図1 タイマA時計用タイムベース機能ブロック図

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマA時計用タイムベース機能によるLED点滅動作を行ないます。

表1 機能割付け

機能	機能割付け
PSW	32.768kHzを4分周したクロックを入力とする5ビットのカウンタ
TCA	32.768kHzを128分周したクロックを入力とする8ビットのカウンタ
TMA	PSWの選択、およびTCAオーバフロー周期の設定
IRRTA	タイマA割込み要求の有無を反映
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマA時計用タイムベース機能によるLED点滅動作を行ないます。

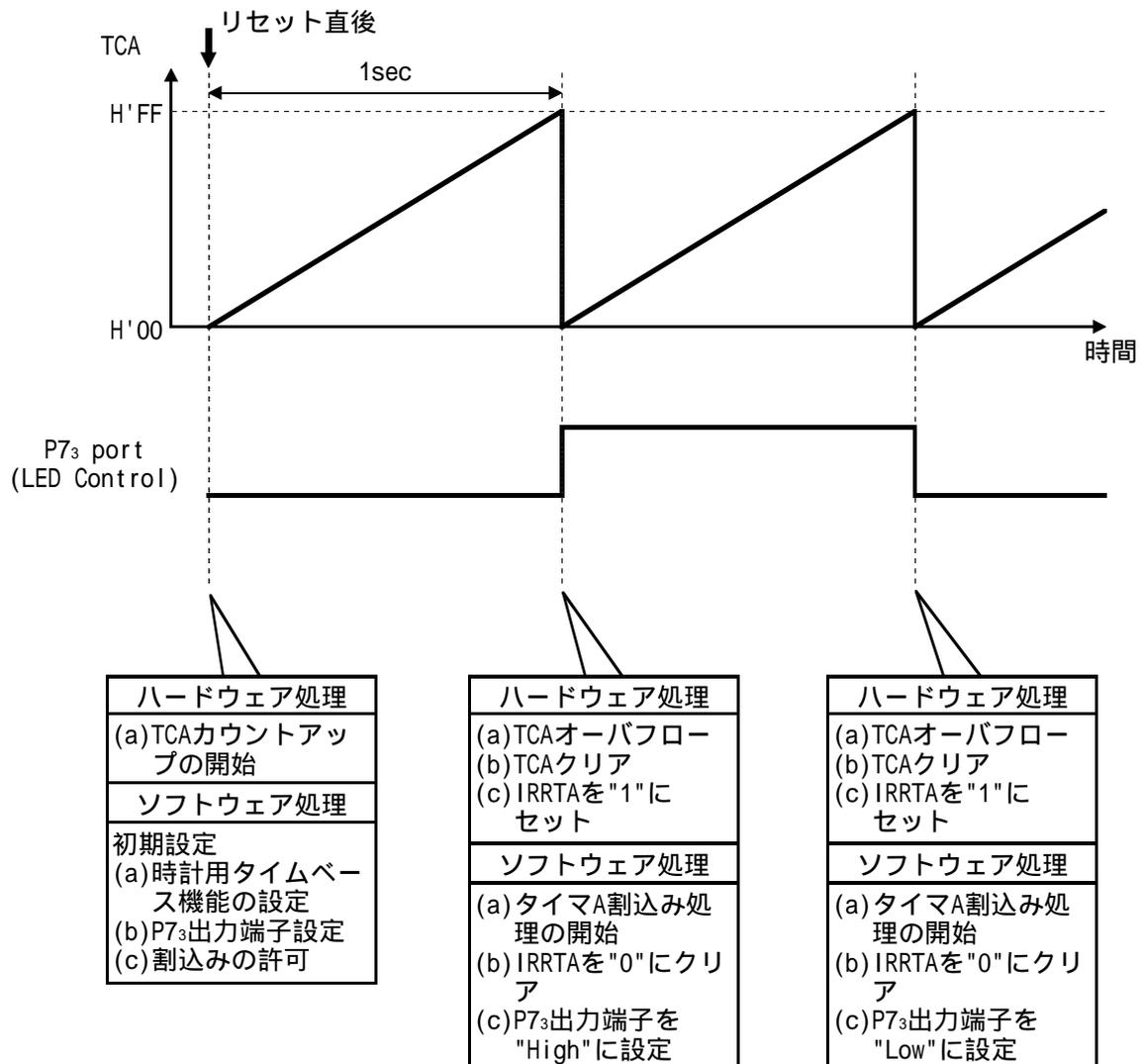


図2 タイマA時計用タイムベース機能によるLED点滅動作の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、時計用タイムベース機能の設定、ポート7の設定、割込みの許可を行なう
LED制御	TAINT	タイマA割込み処理ルーチンで、LEDの点灯/消灯を行う。

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
IRR1 IRR7A	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRR7A=0のとき、タイマA割込みが要求されていない : IRR7A=1のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0
IENR1 IENTA	割込み許可レジスタ1 (タイマA割込みイネーブル) : IENTA=1のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
TMA	タイマモードレジスタA : TMA=H'18のとき、タイマA機能を時計用タイムベース機能に、TCA入力クロックソースをPSWに、TCAオーバフロー周期を1sに設定	H'FFB0	H'18
TCA	タイマカウンタA : 32.768kHzを128分周したクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00
PDR7 P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7 PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1

(4) 使用RAM説明

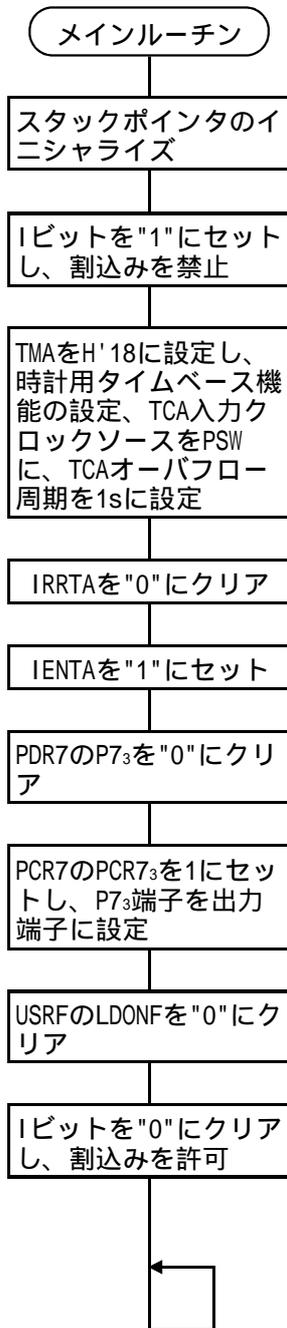
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

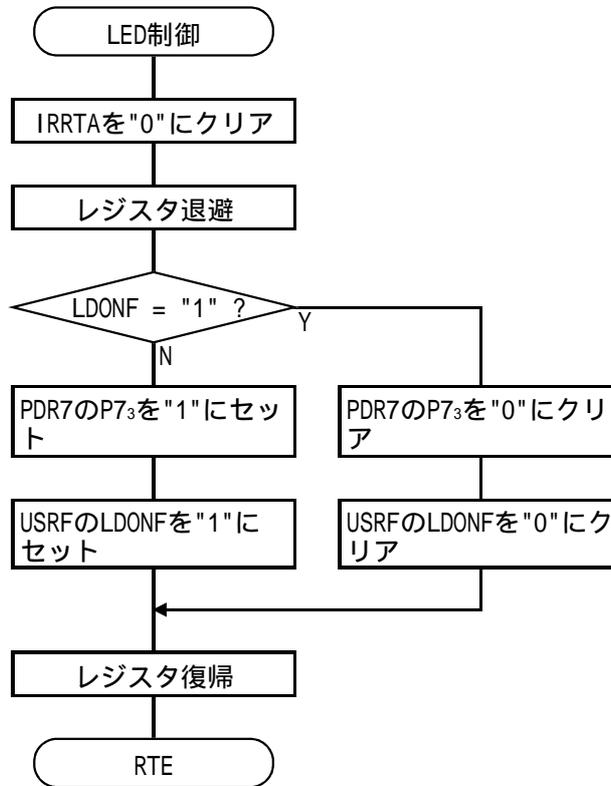
ラベル名	機能	アドレス	使用モジュール名
USRF LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット0	メインルーチン LED制御

フローチャート

(a) メインルーチン



(b) タイマA割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'LED Flickering by Clock Time-Based Function'
;
;           Function
;           : Timer A Clock Time Base
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
;
;           TMA           .equ           H'FFB0           ;Timer Mode Register A
;           TCA           .equ           H'FFB1           ;Timer Counter A
;           PDR7          .equ           H'FFDA           ;Port Data Register 7
;           P73           .bequ         3,PDR7           ;Port Data Register 73
;           PCR7          .equ           H'FFEA           ;Port Control Register 7
;           PCR73         .bequ         3,PCR7           ;Port Control Register 73
;           IENR1         .equ           H'FFF4           ;Interrupt Enable Register 1
;           IENTA         .bequ         6,IENR1          ;Timer A Interrupt Enable
;           IRR1          .equ           H'FFF7           ;Interrupt Request Register 1
;           IRRTA         .bequ         6,IRR1           ;Timer A Interrupt Request Flag
;
;
;           *****
;           Ram Allocation
;           *****
;
;           STACK        .equ           H'FF80           ;Stack Pointer
;           USRF          .equ           H'FB80           ;User Flag Area
;           LDONF         .bequ         0,USRF           ;Led On Flag
;
;
;           *****
;           Vector Address
;           *****
;
;           .org          H'0000
;           .data.w       MAIN           ;Reset Interrupt
;
;
;           .org          H'0008
;           .data.w       MAIN           ;IRQ0 Interrupt
;           .data.w       MAIN           ;IRQ1 Interrupt
;           .data.w       MAIN           ;IRQ2 Interrupt
;           .data.w       MAIN           ;IRQ3 Interrupt
;           .data.w       MAIN           ;INT0 - INT7 Interrupt
;
;
;           .org          H'0014
;           .data.w       TAINT          ;Timer A Interrupt
;           .data.w       MAIN           ;Timer B1 Interrupt

```


プログラムリスト

```
;  
TAINT9      .equ      $  
MOV.B      ROL,@USRF  
;  
POP        R0          ;Restore R0  
;  
RTE  
;  
.end
```

2.3 クロック出力機能によるクロック出力

クロック出力機能によるクロック出力	使用機能	タイマA : クロック出力機能
仕様		
<p>(1) タイマAのクロック出力機能を使用して、TMOW出力端子よりクロック出力を行いません。</p> <p>(2) 出力するクロックは、システムクロックを分周した4種のクロックと、32.768kHzを分周した4種のクロックの計8種のクロックより選択できます。</p> <p>(3) 本タスク例では、TMOW出力端子よりシステムクロックを16分周した2.048kHzのクロックを出力します。</p>		

使用機能説明

- (1) 本タスク例では、タイマAクロック出力機能を使用して、TMOW出力端子よりクロック出力を行いません。図1にタイマAクロック出力機能のブロック図を示します。以下にタイマAクロック出力機能のブロック図について説明します。
- ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - ・ w は、サブクロック発振器の出力クロック (32.768kHz) です。
 - ・プリスケアラS (PSS) は、 w を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - ・プリスケアラW (PSW) は、32.768kHzを4分周したクロック ($w/4$) を入力とする5ビットのカウンタです。
 - ・タイマモードレジスタA (TMA) は、8ビットのリード/ライト可能なレジスタで、TMOW端子から出力するクロックを選択します。本タスク例では、TMAをH' B0に設定することにより、TMOW出力端子より出力するクロックにサブシステムクロックを16分周した2.048Hzのクロックを選択します。
 - ・クロック出力端子 (TMOW) は、タイマ出力クロックの出力端子で、TMOW端子から出力するクロックとして、32.768kHzの32分周、16分周、8分周、4分周したクロック、およびシステムクロックを32分周、16分周、8分周、4分周したクロックの計8種類のクロックより選択可能です。
 - ・表1にTMOW端子より出力するクロックの種類と設定値を示します。

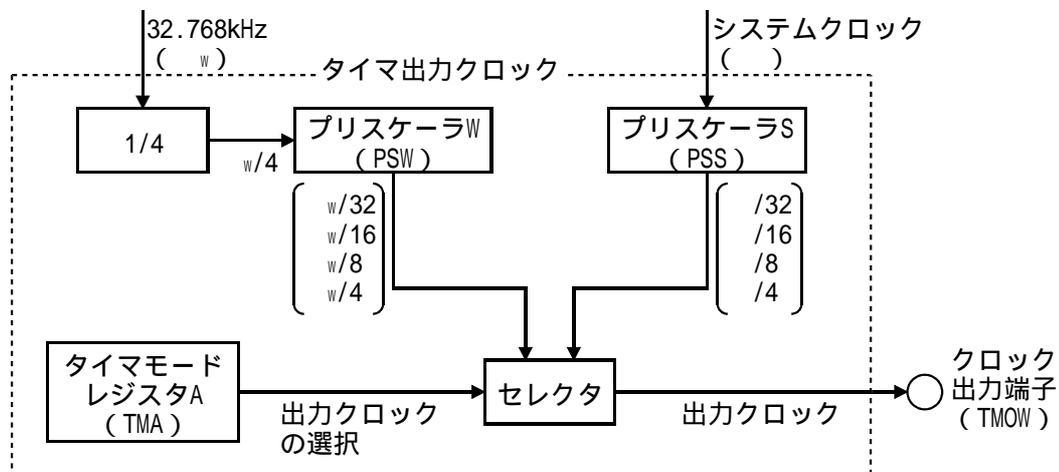


図1 タイマAクロック出力機能ブロック図

表1 TMOW端子より出力するクロックおよび設定値

TMA			出力 クロック	出力クロック周波数	TMA設定値
TMA7	TMA6	TMA5			
0	0	0	/32	5MHz / 32 = 156.25kHz	H' 10
0	0	1	/16	5MHz / 16 = 312.5kHz	H' 30
0	1	0	/8	5MHz / 8 = 625kHz	H' 50
0	1	1	/4	5MHz / 4 = 1.25MHz	H' 70
1	0	0	$w/32$	32.768kHz / 32 = 1.024kHz	H' 90
1	0	1	$w/16$	32.768kHz / 16 = 2.048kHz	H' B0
1	1	0	$w/8$	32.768kHz / 8 = 4.096kHz	H' D0
1	1	1	$w/4$	32.768kHz / 4 = 8.192kHz	H' F0

使用機能説明

(2) 表2に本タスク例の機能割付けを示します。表2に示すように機能を割付け、タイマAクロック出力機能によるクロック出力を行ないます。

表2 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
PSW	32.768kHzを4分周したクロックを入力とする5ビットのカウンタ
TMA	出力クロックの選択
TMOW	クロック出力端子
PMR1	TMOW出力端子機能の設定

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマAクロック出力機能によるクロック出力を行ないます。

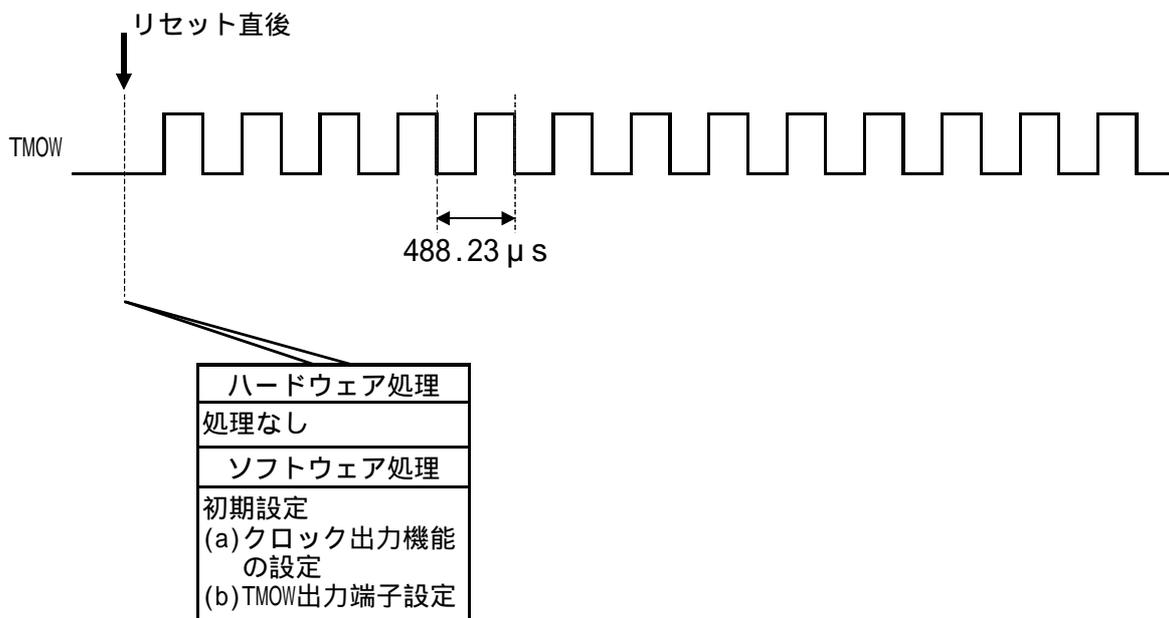


図2 タイマAクロック出力機能によるクロック出力の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、クロック出力機能の設定、TMOW出力端子の設定、割込みの許可を行なう

(2) 引数の説明

本タスク例では、引数は使用しません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

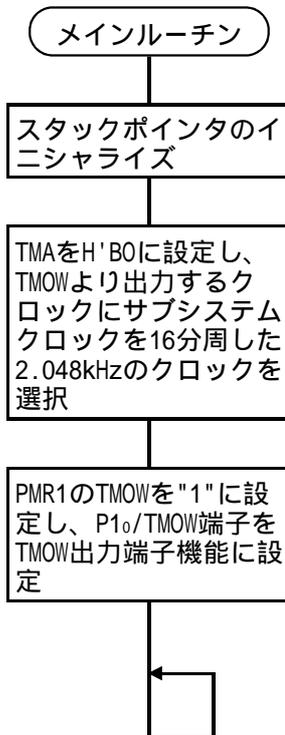
レジスタ名	機能	アドレス	設定値
TMA	タイマモードレジスタA : TMA=H' B0のとき、TMOW出力端子より出力するクロックにサブシステムクロックの16分周した2.048kHzのクロックを選択	H' FF B0	H' B0
PMR1 TMOW	ポートモードレジスタ1 (P1 ₀ /TMOW端子機能切替え) : TMOW=1のとき、P1 ₀ /TMOW端子をTMOW出力端子機能に設定	H' FF FC ビット0	1

(4) 使用RAM説明

本タスク例では、RAMは使用しません。

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Clock Output by Clock Output Function'
;
;           Function
;           : Timer A Clock Output
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
;
;           TMA           .equ           H'FFB0           ;Timer Mode Register A
;           PDR7          .equ           H'FFDA           ;Port Data Register 7
;           PMR1          .equ           H'FFFC           ;Port Mode Register 1
;           TMOW          .bequ          0,PMR1           ;P10/TMOW Terminal Function Change
;
;
;           Ram Allocation
;
*****
;
;           STACK        .equ           H'FF80           ;Stack Pointer
;
;
;           Vector Address
;
*****
;
;           .org          H'0000
;           .data.w       MAIN           ;Reset Interrupt
;
;
;           .org          H'0008
;           .data.w       MAIN           ;IRQ0 Interrupt
;           .data.w       MAIN           ;IRQ1 Interrupt
;           .data.w       MAIN           ;IRQ2 Interrupt
;           .data.w       MAIN           ;IRQ3 Interrupt
;           .data.w       MAIN           ;INT0 - INT7 Interrupt
;
;
;           .org          H'0014
;           .data.w       MAIN           ;Timer A Interrupt
;           .data.w       MAIN           ;Timer B1 Interrupt
;
;
;           .org          H'0020
;           .data.w       MAIN           ;Timer X Interrupt
;           .data.w       MAIN           ;Timer V Interrupt
;
;
;           .org          H'0026
;           .data.w       MAIN           ;Sci1 Interrupt
;
;

```

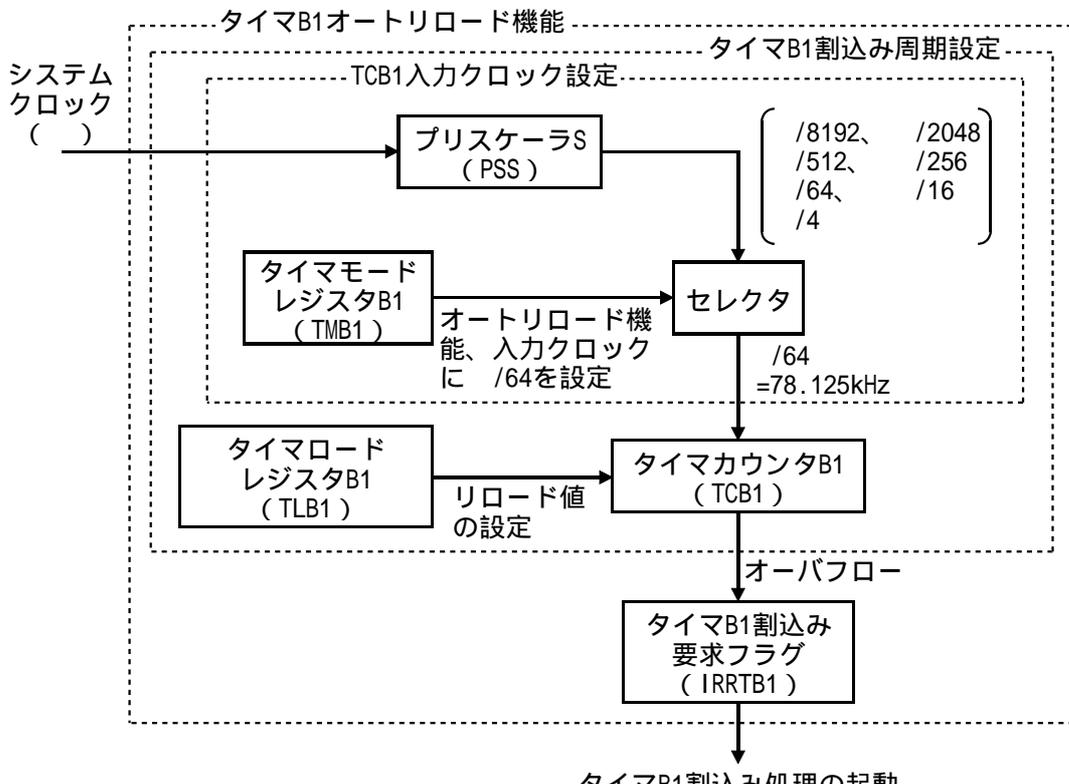
プログラムリスト

```

.org          H'002A
.data.w      MAIN          ;Sci3 Interrupt
.data.w      MAIN          ;A/D Converter Interrupt
.data.w      MAIN          ;Sleep Interrupt
;
;
;*****
;
;          Main Program
;*****
;
;
;          .org          H'1000
;
MAIN          .equ          $
MOV.W        #STACK,SP    ;Initialize Stack Pointer
;
;          MOV.B        #H'BO,ROL
;          MOV.B        ROL,@TMA    ;Initialize Output Clock Function
;
;          BSET         TMOW        ;Initialize TMOW Output Terminal Function
;
MAIN9         .equ          $
;          BRA          MAIN9
;
;          .end

```

2.4 オートリロードタイマ機能による割り込み周期設定

オートリロードタイマ機能による割り込み周期設定	使用機能	タイマB1 : オートリロードタイマ機能
<p>仕様</p> <p>(1) タイマB1のオートリロードタイマ機能を使用して、2.048msごとにタイマB1割り込みを発生させます。</p> <p>(2) タイマB1割り込み処理の中で、タイマB1割り込みの回数をカウントし、250回カウントするごとにLEDの点灯または消灯を行ないます。</p>		
<p>使用機能説明</p> <p>(1) 本タスク例では、タイマB1オートリロード機能を使用して、2.048msごとにタイマB1割り込みを発生させます。</p> <p>(a) 図1にタイマB1オートリロード機能のブロック図を示します。以下にタイマB1オートリロード機能のブロック図について説明します。</p> <ul style="list-style-type: none"> システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 プリスケアラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 タイマモードレジスタB1 (TMB1) は、8ビットのリード/ライト可能なレジスタで、オートリロード機能の選択、および入力クロックの選択を行ないます。 タイマカウンタB1 (TCB1) は、8ビットのリード可能なアップカウンタで、入力する内部クロック/外部イベントによりカウントアップされます。入力するクロックはシステムクロックの8192分周、2048分周、512分周、256分周、64分周、16分周、4分周および外部クロックの計8種類のクロックより選択可能です。本タスク例では、TCB1の入力クロックにシステムクロックの64分周のクロックを選択しています。 タイマロードレジスタB1 (TLB1) は、8ビットのライト専用のレジスタで、TCB1のリロード値を設定します。本タスク例では、2.048msでTCB1がオーバーフローするようにTLB1をH' 60に設定します。 タイマB1割り込み要求フラグ (IRRTB1) は、TCB1がオーバーフローすることにより"1"にセットされます。IRRTB1が"1"にセットされていて、割り込み許可レジスタ1 (IENR1) のタイマB1割り込みイネーブル (IENTB1) が"1"で、かつコンディションコードレジスタ (CCR) の1ビットが"0"にクリアされている場合にタイマB1割り込みが受け付けられ、タイマB1割り込み処理を開始します。 <div style="text-align: center;">  <p>タイマB1オートリロード機能</p> <p>システムクロック ()</p> <p>TCB1入力クロック設定</p> <p>プリスケアラS (PSS)</p> <p>タイマモードレジスタB1 (TMB1)</p> <p>タイマロードレジスタB1 (TLB1)</p> <p>タイマカウンタB1 (TCB1)</p> <p>タイマB1割り込み要求フラグ (IRRTB1)</p> <p>タイマB1割り込み処理の起動</p> <p>図1 タイマB1オートリロード機能ブロック図</p> </div>		

使用機能説明

- (b) タイマB1オートリロード機能による割込み周期の設定方法について以下に説明します。
- ・タイマB1ーオートリロード機能による割込み周期は以下の式によって設定します。

$$\text{タイマB1割込み周期(s)} = (\text{TCB1入力クロック周期(s)}) \times (256 - (\text{リロード設定値}))$$

- ・上記の式によって設定したTCB1入力クロック周期をTMB1に、リロード設定値をTLB1にそれぞれ設定することにより、タイマB1割込み周期を設定します。

- (2) 表1に本タスク例の機能割付けを示します。表2に示すように機能を割付け、タイマB1オートリロード機能による割込み周期設定を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCB1	システムクロックの64分周を入力とする8ビットのカウンタ
TMB1	オートリロード機能の設定、およびTCB1入力クロックを /64に設定
TLB1	TCB1のリロード値を設定
IRRTB1	タイマB1割込み要求の有無を反映
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマB1オートリロード機能による割り込み周期設定を行ないます。

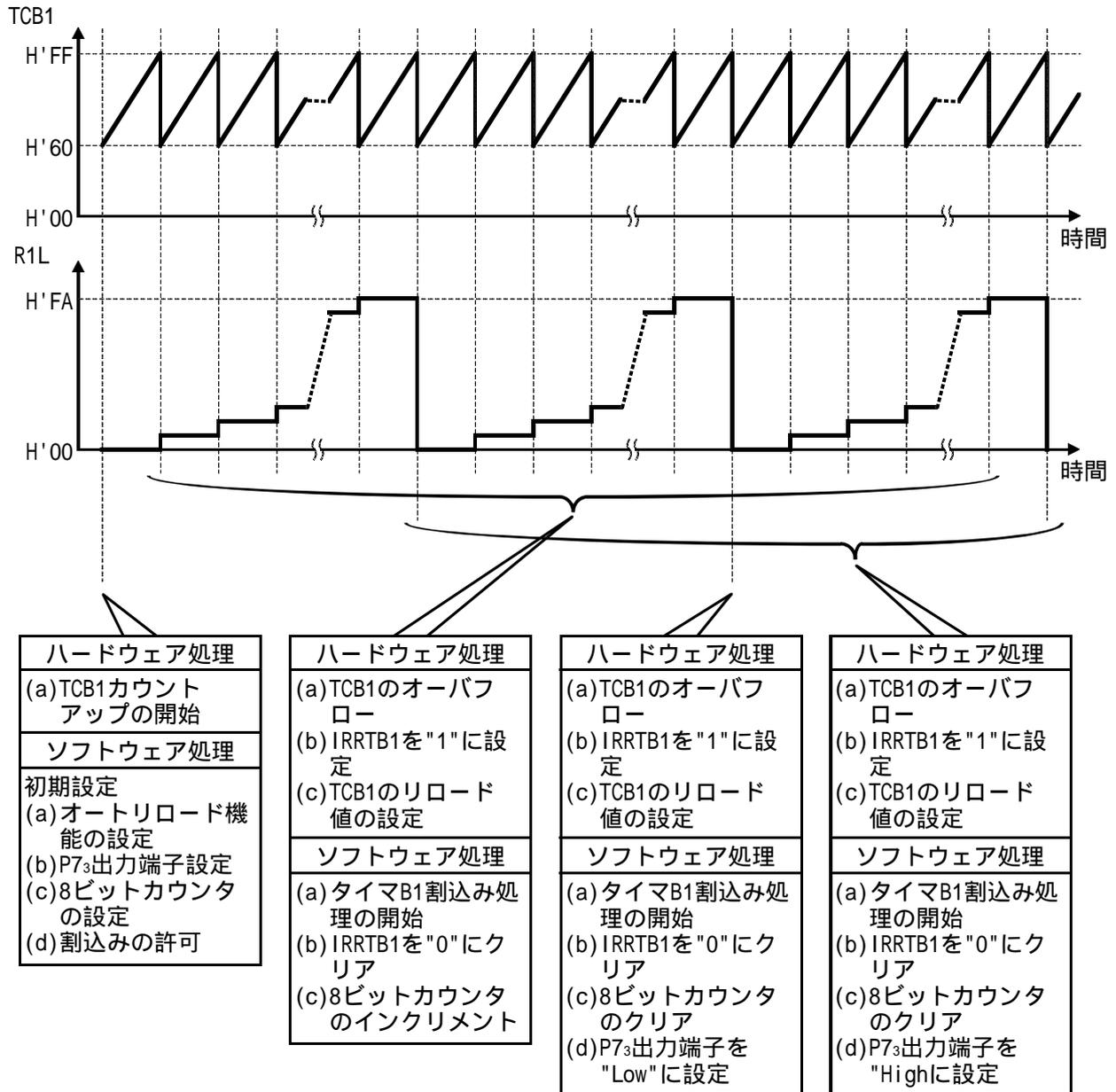


図2 タイマB1オートリロード機能による割り込み周期設定の動作原理

オートリロードタイマ機能による割り込み周期設定	使用機能	タイマB1：オートリロードタイマ機能
-------------------------	------	--------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、オートリロード機能の設定、ポート7 ₃ の設定、8ビットカウンタの設定、割り込みの許可、LEDの点灯/消灯を行なう
カウントアップ	TBINT	8ビットカウンタのインクリメント、および8ビットカウンタのカウント値がH'FAになった時点でCTEDFを"1"にセットする

(2) 引数の説明

本タスク例では、引数は使用しません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
IRR1 IRRTB1	割り込み要求レジスタ1 (タイマB1割り込み要求フラグ) : IRRTB1=0のとき、タイマB1割り込みが要求されていない : IRRTB1=1のとき、タイマB1割り込みが要求されている	H'FFF7 ビット7	0
IENR1 IENTB1	割り込み許可レジスタ1 (タイマB1割り込みイネーブル) : IENTB1=1のとき、タイマB1割り込み要求を許可	H'FFF4 ビット7	1
TMB1	タイマモードレジスタB1 : TMB1=H'FCのとき、タイマB1機能をオートリロード機能に、TCB1入力クロックにシステムクロックの64分周のクロックに設定	H'FFB2	H'FC
TCB1	タイマカウンタB1 : システムクロックを64分周したクロックを入力とする8ビットのアップカウンタ	H'FFB3	H'00
TLB1	タイマロードレジスタB1 : TLB1=H'60に設定すると、TCB1はH'60からカウントアップを開始し、TCB1がオーバーフローするとTCB1にH'60がロードされます	H'FFB3	H'60
PDR7 P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7 PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1

(4) 使用RAM説明

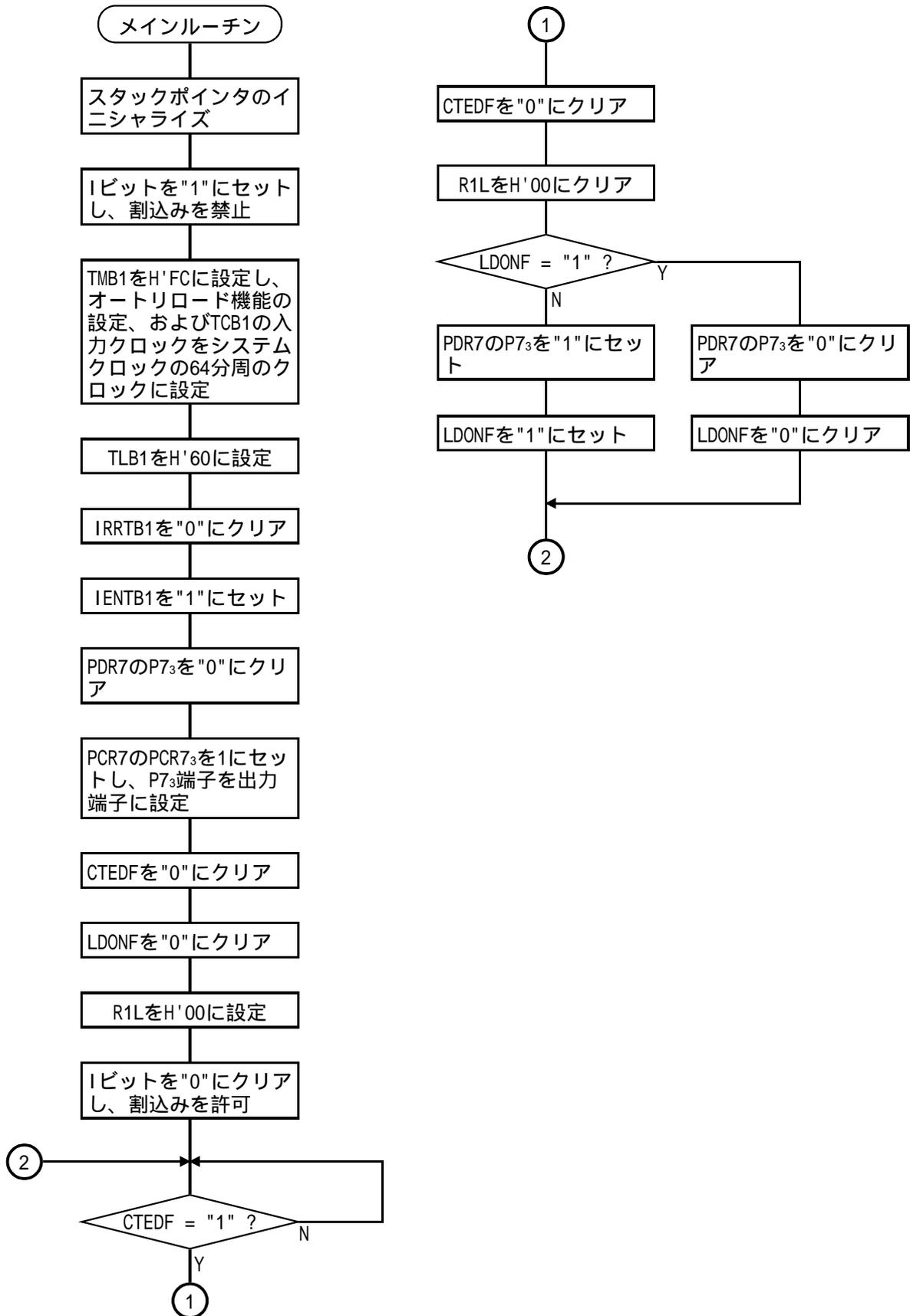
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
USRF	CTEDF	H'FB80 ビット0	メインルーチン カウントアップ
	LDONF	H'FB80 ビット1	メインルーチン

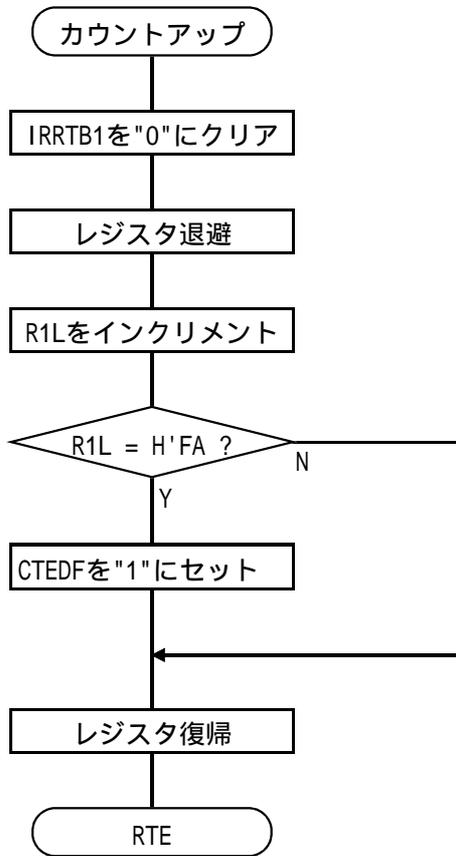
フローチャート

(a) メインルーチン



フローチャート

(b) タイマB1割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Interrupt Period Setting by Auto Reload
;       Timer Function'
;
;       Function
;       : Timer B1 Auto Reload Timer
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
TMB1      .equ        H'FFB2      ;Timer Mode Register B1
TCB1      .equ        H'FFB3      ;Timer Counter B1
TLB1      .equ        H'FFB3      ;Timer Load Register B1
PDR7      .equ        H'FFDA      ;Port Data Register 7
P73       .bequ       3,PDR7      ;Port Data Register 73
PCR7      .equ        H'FFEA      ;Port Control Register 7
PCR73     .bequ       3,PCR7      ;Port Control Register 73
IENR1     .equ        H'FFF4      ;Interrupt Enable Register 1
IENTB1    .bequ       7,IENR1     ;Timer B1 Interrupt Enable
IRR1      .equ        H'FFF7      ;Interrupt Request Register 1
IRRTB1    .bequ       7,IRR1      ;Timer B1 Interrupt Request Flag
;
;
*****
;
;       Ram Allocation
;
*****
;
STACK     .equ        H'FF80      ;Stack Pointer
USRF      .equ        H'FB80      ;User Flag Area
CTEDF     .bequ       0,USRF      ;Count End Flag
LDONF     .bequ       1,USRF      ;Led On Flag
;
;
*****
;
;       Vector Address
;
*****
;
;       .org          H'0000
;       .data.w       MAIN        ;Reset Interrupt
;
;       .org          H'0008
;       .data.w       MAIN        ;IRQ0 Interrupt
;       .data.w       MAIN        ;IRQ1 Interrupt
;       .data.w       MAIN        ;IRQ2 Interrupt
;       .data.w       MAIN        ;IRQ3 Interrupt
;       .data.w       MAIN        ;INT0 - INT7 Interrupt
;
;

```

プログラムリスト

```

.org          H'0014
.data.w      MAIN          ;Timer A Interrupt
.data.w      TBINT        ;Timer B1 Interrupt
;

.org          H'0020
.data.w      MAIN          ;Timer X Interrupt
.data.w      MAIN          ;Timer V Interrupt
;

.org          H'0026
.data.w      MAIN          ;Sci1 Interrupt
;

.org          H'002A
.data.w      MAIN          ;Sci3 Interrupt
.data.w      MAIN          ;A/D Converter Interrupt
.data.w      MAIN          ;Sleep Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org          H'1000
;
MAIN          .equ          $
MOV.W        #STACK,SP    ;Initialize Stack Pointer
ORC          #H'80,CCR     ;Interrupt Disable
;
MOV.W        #H'FC60,R0
MOV.B        ROH,@TMB1    ;Initialize Timer B1 Function & Input Clock
MOV.B        ROL,@TLB1    ;Initialize TCB1 Reload Value
;
BCLR         IRRTB1       ;Clear IRRTB1
BSET         IENTB1       ;Timer B1 Interrupt Enable
;
BSET         PCR73        ;Initialize P73 Output Terminal Function
;
MOV.B        #H'00,R0L
MOV.B        ROL,@USRF    ;Initialize User Flag Area
;
MOV.B        #H'00,R1L    ;Initialize 8bit Counter
;
ANDC         #H'7F,CCR     ;Interrupt Enable
;
MAIN1        .equ          $
MOV.B        @USRF,R0L
BTST         #0,R0L        ;CTEDF = "1" ?
BEQ          MAIN1        ;No.
;
BCLR         #0,R0L
MOV.B        ROL,@USRF    ;Clear CTEDF
;
MOV.B        #H'00,R1L    ;Initialize 8bit Counter
;
MOV.B        @USRF,R0L
BTST         #1,R0L        ;LDONF = "1" ?
BNE          MAIN2        ;Yes.
;
BSET         P73          ;Turn on LED
BSET         #1,R0L
MOV.B        ROL,@USRF    ;Set LDONF
BRA          MAIN1

```

プログラムリスト

```
;  
MAIN2      .equ          $  
           BCLR         P73           ;Turn off LED  
           BCLR         #1,R0L  
           MOV.B        R0L,@USRF     ;Clear LDONF  
           BRA          MAIN1  
  
;  
;*****  
;               Timer B1 Interrupt  
;*****  
;  
TBINT      .equ          $  
           BCLR         IRRTB1        ;Clear IRRTB1  
           PUSH        R0             ;Store R0  
  
;  
           INC          R1L           ;Increment 8bit Counter  
  
;  
           MOV.B        #H'FA,R0L     ;8bit Counter = H'FA ?  
           SUB.B        R1L,R0L  
           BNE         TBINT9        ;No.  
  
;  
           MOV.B        @USRF,R0L  
           BSET        #0,R0L  
           MOV.B        R0L,@USRF     ;Set CTEDF  
  
;  
TBINT9     .equ          $  
           POP          R0            ;Restore R0  
  
;  
           RTE  
  
;  
           .end
```

2.5 イベントカウンタ機能によるパルスの周波数測定

イベントカウンタ機能によるパルスの周波数測定	使用機能	タイマB1 : 8ビットイベントカウンタ機能
仕様		
<p>(1) タイマB1の8ビットイベントカウンタ機能を使用して、タイマB1イベント入力端子 (TMIB) より入力したパルスの周波数を測定します。</p> <p>(2) TMIB入力端子より入力したパルスの立ち上がりエッジ検出の回数を1sec間カウントし、1sec間のカウント数をRAMに格納します。</p> <p>(3) 1secの測定時間は、タイマA時計用タイムベース機能を使用して計測します。</p>		

使用機能説明
<p>(1) 本タスク例では、タイマB1イベントカウンタ機能を使用して、TMIB入力端子に入力されたパルスの周波数測定を行います。</p> <p>(a) 図1にタイマB1イベントカウンタ機能のブロック図を示します。以下にタイマB1イベントカウンタ機能のブロック図について説明します。</p> <ul style="list-style-type: none"> ・タイマモードレジスタB1 (TMB1) は、8ビットのリード/ライト可能なレジスタで、インターバル機能の選択、および入力クロックの選択を行ないます。 ・タイマカウンタB1 (TCB1) は、8ビットのリード可能なアップカウンタで、入力する内部クロック/外部イベントによりカウントアップされます。入力するクロックはシステムクロックの8192分周、2048分周、512分周、256分周、64分周、16分周、4分周および外部クロックの計8種類のクロックより選択可能です。本タスク例では、TCB1の入力クロックにTMIB入力端子のエッジ検出を選択しています。 ・タイマB1割込み要求フラグ (IRRTB1) は、TCB1がオーバーフローすることにより"1"にセットされます。IRRTB1が"1"にセットされていて、割込み許可レジスタ1 (IENR1) のタイマB1割込みイネーブル (IENTB1) が"1"で、かつコンディションコードレジスタ (CCR) のIビットが"0"にクリアされている場合にタイマB1割込みが受け付けられ、タイマB1割込み処理を開始します。 ・タイマB1イベント入力端子 (TMIB) は、周波数測定を行なうパルスを入力端子として機能します。

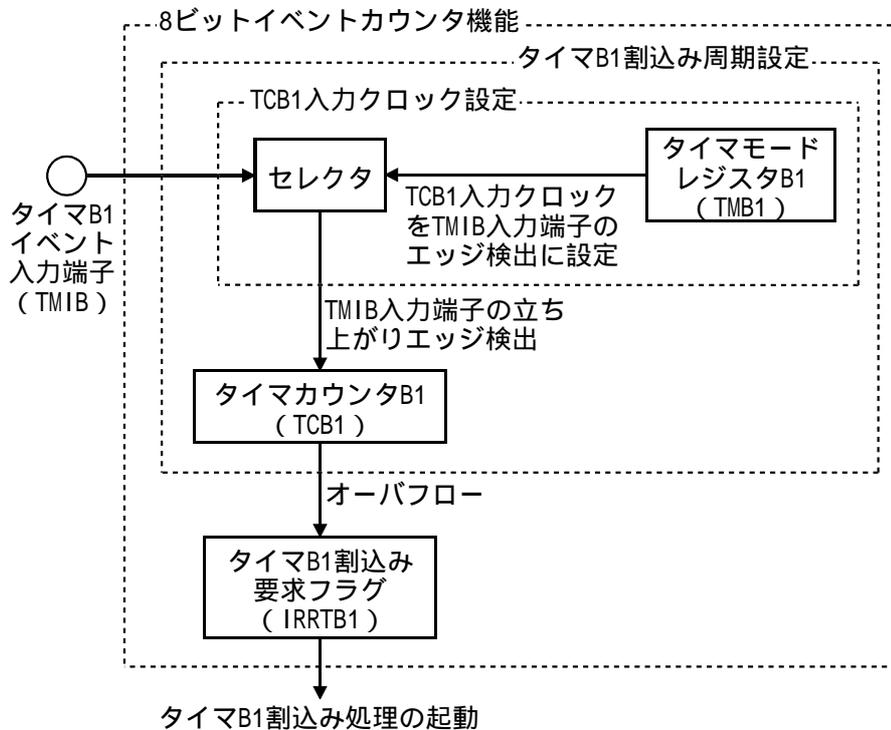


図1 タイマB1イベントカウンタ機能のブロック図

使用機能説明

(b) 周波数の測定方法について以下に説明します。

- ・ TMIB入力端子に256回の立ち上がりエッジが入力されると、TCB1がオーバーフローし、タイマB1割込みが発生します。
- ・ タイマB1割込み処理の中で、R1Lに設定した8ビットカウンタをインクリメントします。
- ・ 1sec経過した時点で、TCB1のカウント値を読み出しR1Hに格納し、TMIB入力端子によるTCB1のカウントアップを終了します。
- ・ TMIB端子に入力したパルスの周波数は、以下の式によって求められます。

$$\begin{aligned} \text{入力パルスの周波数(Hz)} &= (\text{タイマB1割込み発生回数}) \times 256 + (1\text{sec経過時点のTCB1のカウント値}) \\ &= (\text{R1Lの値}) \times 256 + (\text{R1Hの値}) \end{aligned}$$

- ・ タイマB1割込みの発生回数をカウントするカウンタ (R1L) が8ビットのカウンタであるために測定可能な入力パルスの周波数は最大で65.535kHzとなります。
- ・ タイマB1割込み回数をカウントする8ビットカウンタ (R1L) がオーバーフローすると、その時点で周波数測定を中止し、R1Lと1sec経過時点のTCB1のカウント値を格納するレジスタ (R1H) にH'00を書き込んで終了します。

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマB1イベントカウンタ機能による周波数測定を行ないます。

表1 機能割付け

機能	機能割付け
TCB1	TMIB入力端子のエッジ検出を入力とする8ビットのカウンタ
TMB1	インターバル機能の設定、およびTCB1入力クロックをTMIB入力端子のエッジ検出に設定
IRRTB1	タイマB1割込み要求の有無を反映
TMIB	周波数測定を行なうパルスの入力端子
INTEG6	TMIB端子の入力センスを立ち上がりエッジ検出に設定
PSW	32.768kHzを4分周したクロックを入力とする5ビットのカウンタ
TCA	32.768kHzを128分周したクロックを入力とする8ビットのカウンタ
TMA	PSWの選択、およびTCAオーバーフロー周期の設定
IRRTA	タイマA割込み要求の有無を反映

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマB1イベントカウンタ機能によるパルスの周波数測定を行ないます。

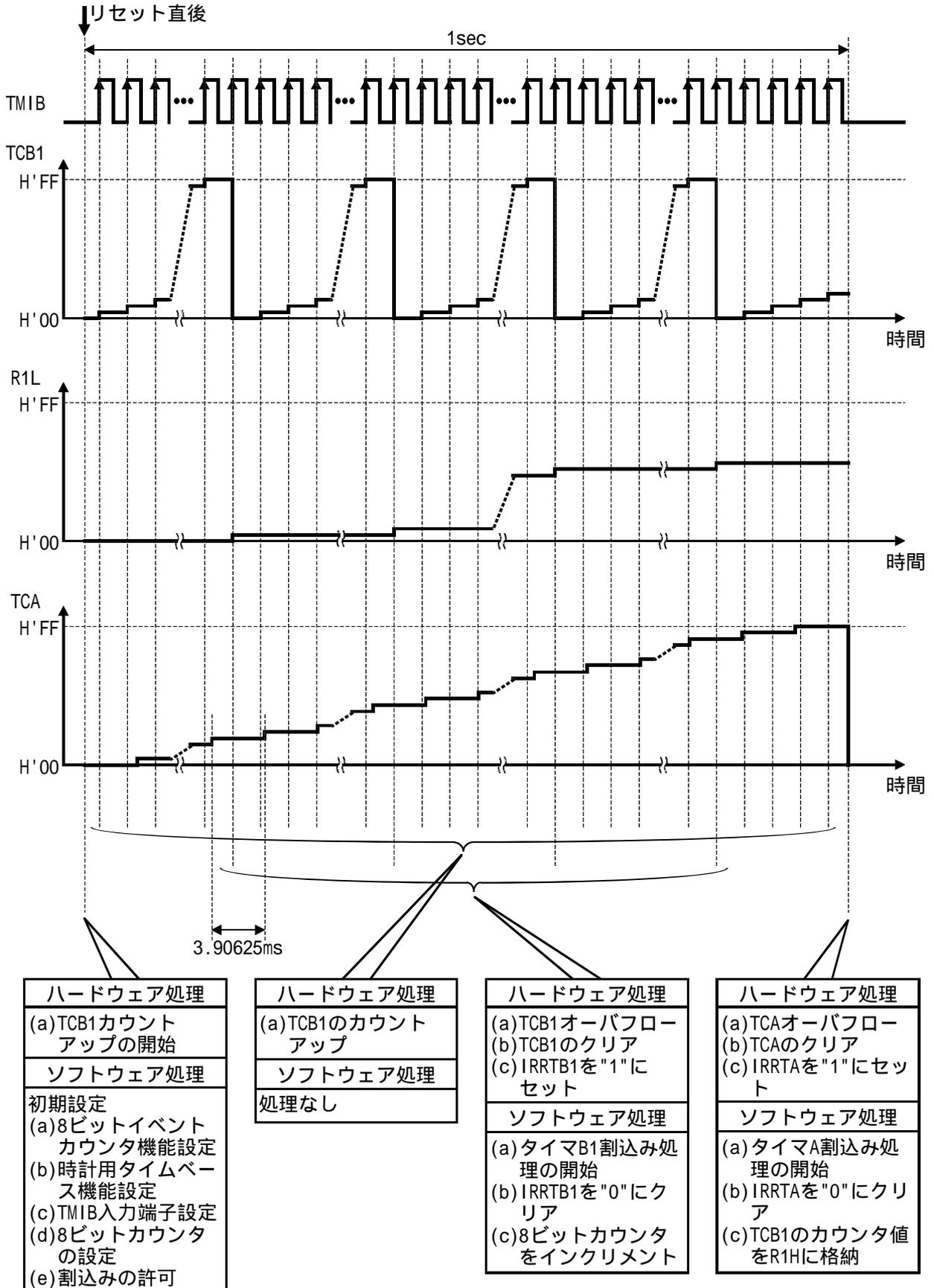


図2 タイマB1イベントカウンタ機能による周波数測定の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	EVCMN	スタックポインタのイニシャライズ、イベントカウンタ機能の設定、時計用タイムベース機能の設定、8ビットカウンタの設定、割込みの許可、測定終了後にタイマB1のイニシャライズを行なう
8ビットカウンタ	TBINT	タイマB1割込み処理で、8ビットカウンタのインクリメント、R1Lがオーバフローしたときの処理を行なう
1sec経過	TAINT	タイマA割込み処理で、1secの経過により割り込みの禁止、TCB1のカウンタ値をR1Hに格納を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
R1L	1sec経過時点の8ビットカウンタのカウンタ値を格納	8ビットカウンタ	1バイト	出力
R1H	1sec経過時点のTCB1のカウンタ値を格納	1sec経過	1バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
IRR1	IRR1B1 割込み要求レジスタ1 (タイマB1割込み要求フラグ) : IRR1B1=0のとき、タイマB1割込みが要求されていない : IRR1B1=1のとき、タイマB1割込みが要求されている	H'FFF7 ビット7	0
	IRR1A 割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRR1A=0のとき、タイマA割込みが要求されていない : IRR1A=1のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0
IENR1	IENR1B1 割込み許可レジスタ1 (タイマB1割込みイネーブル) : IENR1B1=0のとき、タイマB1割込み要求を禁止 : IENR1B1=1のとき、タイマB1割込み要求を許可	H'FFF4 ビット7	1
	IENR1A 割込み許可レジスタ1 (タイマA割込みイネーブル) : IENR1A=0のとき、タイマA割込み要求を禁止 : IENR1A=1のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IEGR2	INTEG6 割込み許可エッジセレクトレジスタ2 (INT6エッジセレクト) : INTEG6=1のとき、TM1B入力端子のエッジ検出を立上りエッジ検出に設定	H'FFF3 ビット6	1
TMB1	タイマモードレジスタB1 : TMB1=H'7Fのとき、タイマB1機能をインターバル機能に、TCB1入力クロックをTM1B端子の入力エッジ検出に設定	H'FFB2	H'7F
TMA	タイマモードレジスタA : TMA=H'18のとき、タイマA機能を時計用タイムベース機能に、TCA入力クロックソースをPSWに、TCAオーバフロー周期を1sに設定	H'FFB0	H'18
TCB1	タイマカウンタB1 : TM1B端子の入力エッジ検出を入力とする8ビットのアップカウンタ	H'FFB3	H'00
TCA	タイマカウンタA : 32.768kHzの128分周のクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00

ソフトウェア説明

(4) 使用RAM説明

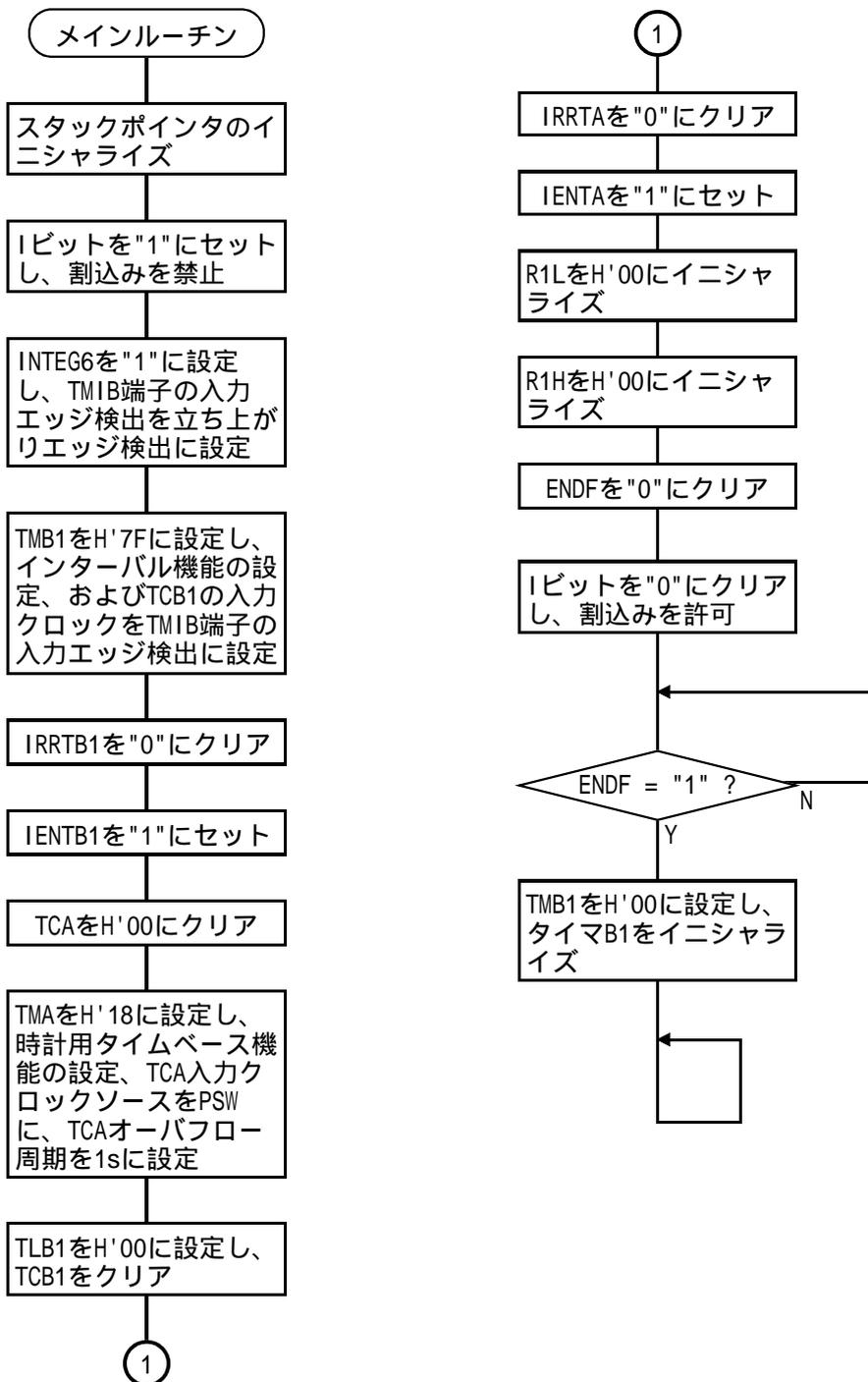
表5に本タスク例で使用するRAMの説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
USRF ENDF	入力パルスの周波数測定が終了したか否かを判定するフラグ	H'FB80 ビット0	メインルーチン 8ビットカウンタ 1sec経過

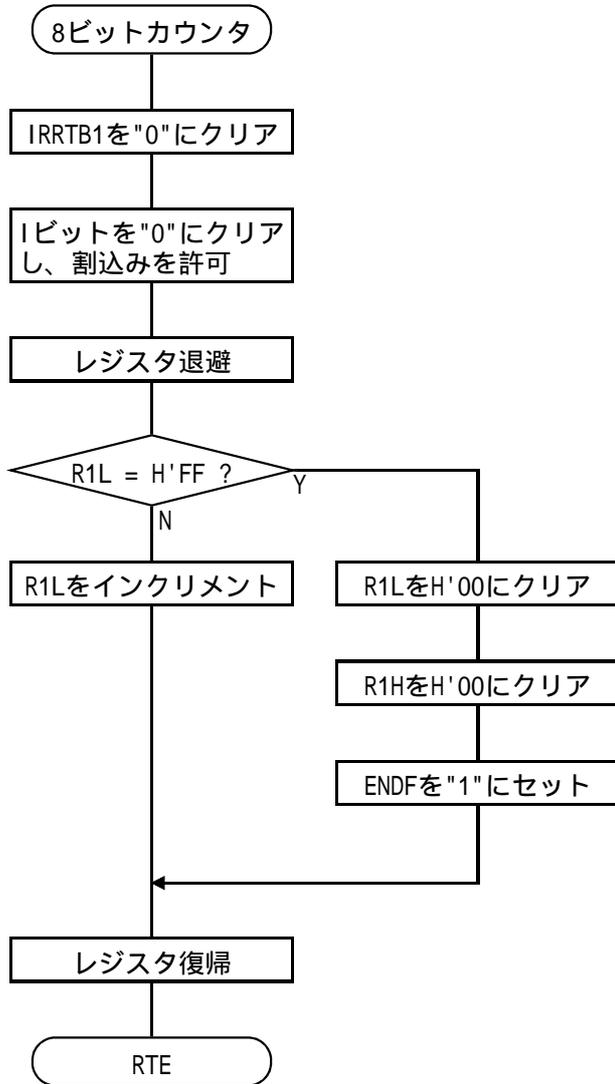
フローチャート

(a) メインルーチン



フローチャート

(b) タイマB1割込み処理ルーチン



(c) タイマA割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Pulse Frequency Measurement by Event
;           Counter Function'
;
;           Function
;           : Timer B1 8bit Event Counter
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
;
;           TMA           .equ           H'FFB0           ;Timer Mode Register A
;           TCA           .equ           H'FFB1           ;Timer Counter A
;           TMB1          .equ           H'FFB2           ;Timer Mode Register B1
;           TCB1          .equ           H'FFB3           ;Timer Counter B1
;           TLB1          .equ           H'FFB3           ;Timer Load Register B1
;           IEGR2         .equ           H'FFF3           ;Interrupt Edge Select Register 2
;           INTEG6        .bequ         6,IEGR2           ;INT6 Edge Select
;           IENR1         .equ           H'FFF4           ;Interrupt Enable Register 1
;           IENTB1        .bequ         7,IENR1           ;Timer B1 Interrupt Enable
;           IENTA         .bequ         6,IENR1           ;Timer A Interrupt Enable
;           IRR1          .equ           H'FFF7           ;Interrupt Request Register 1
;           IRRTB1        .bequ         7,IRR1           ;Timer B1 Interrupt Request Flag
;           IRRTA         .bequ         6,IRR1           ;Timer A Interrupt Request Flag
;
;
;           *****
;           ;
;           ;           Ram Allocation
;           ;
;           *****
;           ;
;           STACK         .EQU           H'FF80           ;Stack Pointer
;           USRF          .EQU           H'FB80           ;User Flag Erea
;           ENDF          .bequ         0,USRF           ;End Flag
;
;           *****
;           ;
;           ;           Vector Address
;           ;
;           *****
;           ;
;           .org           H'0000
;           .data.w        MAIN           ;Reset Interrupt
;
;           ;
;           .org           H'0008
;           .data.w        MAIN           ;IRQ0 Interrupt
;           .data.w        MAIN           ;IRQ1 Interrupt
;           .data.w        MAIN           ;IRQ2 Interrupt
;           .data.w        MAIN           ;IRQ3 Interrupt
;           .data.w        MAIN           ;INT0 - INT7 Interrupt

```


プログラムリスト

```

                BRA          MAIN9
;
;
;*****
;
;          Timer B1 Interrupt
;*****
;
;
TBINT          .equ          $
               BCLR          IRRTB1      ;Clear IRRTB1
               ANDC          #'7F,CCR    ;Interrupt Enable
;
;               PUSH          R0          ;Store R0
;
;               CMP.B        #'FF,R1L    ;R1L = H'FF ?
               BEQ          TBINT1      ;Yes
;
;               INC          R1L         ;8bit Counter Increment
               BRA          TBINT9
;
TBINT1         .equ          $
               MOV.B        #'00,R1L    ;Clear 8bit Counter
               MOV.B        #'00,R1H    ;Clear R1H
;
;               MOV.B        #'01,R0L    ;Set ENDF
               MOV.B        ROL,@USRF
;
TBINT9         .equ          $
               POP          R0          ;Restore R0
;
;               RTE
;
;*****
;
;          Timer A Interrupt
;*****
;
;
TAINT          .equ          $
               BCLR          IRRTA      ;Clear IRRTA
;
;               MOV.B        @TCB1,R1H  ;Store TCB1
;
;               MOV.B        #'01,R0L    ;Set ENDF
               MOV.B        ROL,@USRF
;
;               BCLR          IENTB1     ;Timer B1 Interrupt Disable
               BCLR          IENTA      ;Timer A Interrupt Disable
;
;               RTE
;
               .end

```

2.6 コンペアマッチ機能によるPWM出力

コンペアマッチ機能によるPWM出力	使用機能	タイマV：コンペアマッチ機能
<p data-bbox="188 181 245 210">仕様</p> <p data-bbox="180 264 1406 439"> (1) 図1に示すように、タイマVのコンペアマッチ機能を使用して、TMOV出力端子よりPWM出力を行いません。 (2) 出力するPWM波形の周期は、タイムコンスタントレジスタA (TCORA) により設定します。 (3) 出力するPWM波形のHigh幅は、タイムコンスタントレジスタB (TCORB) により設定します。 (4) 本タスク例では、周期が6.5536ms、デューティが1周期ごとに6.25%～93.75%の間を6.25%ずつ変動するPWM波形を出力します。 </p> <div data-bbox="256 488 1385 763" style="text-align: center;"> <p data-bbox="256 506 456 566">タイマV出力端子 (TMOV)</p> <p data-bbox="628 613 794 642">パルスHigh幅</p> <p data-bbox="711 669 847 698">パルス周期</p> <p data-bbox="900 696 1385 763">デューティ = $\frac{\text{パルスHigh幅}}{\text{パルス周期}} \times 100 (\%)$</p> </div> <p data-bbox="722 797 884 826">図1 PWM出力</p>		
<p data-bbox="137 880 296 909">使用機能説明</p> <p data-bbox="180 969 1406 1899"> (1) 本タスク例では、タイマVコンペアマッチ機能を使用して、TMOV出力端子よりPWM波形を出力します。 (a) 図2にタイマVコンペアマッチ機能のブロック図を示します。以下にタイマVコンペアマッチ機能のブロック図について説明します。 <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・プリスケラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・タイマカウンタV (TCNTV) は、8ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは を分周した6種類のクロックと、3種類の外部クロックより選択できます。 ・タイマコントロールレジスタV0 (TCRV0) は、8ビットのリード/ライト可能なレジスタで、TCNTVの入力クロックの選択、TCNTVのクリア指定、および各割込み要求の許可を行います。本タスク例では、TCNTVの入力クロックに の128分周のクロックを、TCNTVのクリア指定にコンペアマッチAによるクリアを、コンペアマッチAによる割込み要求の許可を選択しています。 ・タイマコントロール/ステータスレジスタV (TCSR) は、8ビットのレジスタで、コンペアマッチフラグのセット、タイマオーバフローフラグのセット、およびコンペアマッチ出力の制御を行いません。本タスク例では、コンペアマッチBでTMOV端子より"0"を出力、コンペアマッチAでTMOV端子より"1"を出力するように設定しています。 ・タイマコントロールレジスタV1 (TCRV1) は、8ビットのリード/ライト可能なレジスタで、TCRV0とともにTCNTVの入力クロックの選択を行いません。 ・タイムコンスタントレジスタA (TCORA) は、8ビットのリード/ライト可能なレジスタで、TCORAの内容はTCNTVと常に比較されており、両者の値が一致するとコンペアマッチAにより、TMOV端子より"1"を出力し、TCNTVをH'00にクリアします。 ・タイムコンスタントレジスタB (TCORB) は、8ビットのリード/ライト可能なレジスタで、TCORBの内容はTCNTVと常に比較されており、両者の値が一致するとコンペアマッチBにより、TMOV端子より"0"を出力します。 ・TCNTVの内容は、TCORA、TCORBと常に比較されており、一致するとTCSRのコンペアマッチフラグA (CMFA)、およびコンペアマッチフラグB (CMFB) が"1"にセットされます。このとき、TCRV0のコンペアマッチインタラプトイネーブルA (CMIEA)、およびコンペアマッチインタラプトイネーブルB (CMIEB) が"1"ならば、CPUに割込みを要求します。 ・タイマV出力端子 (TMOV) よりPWM波形を出力します。 </p>		

使用機能説明

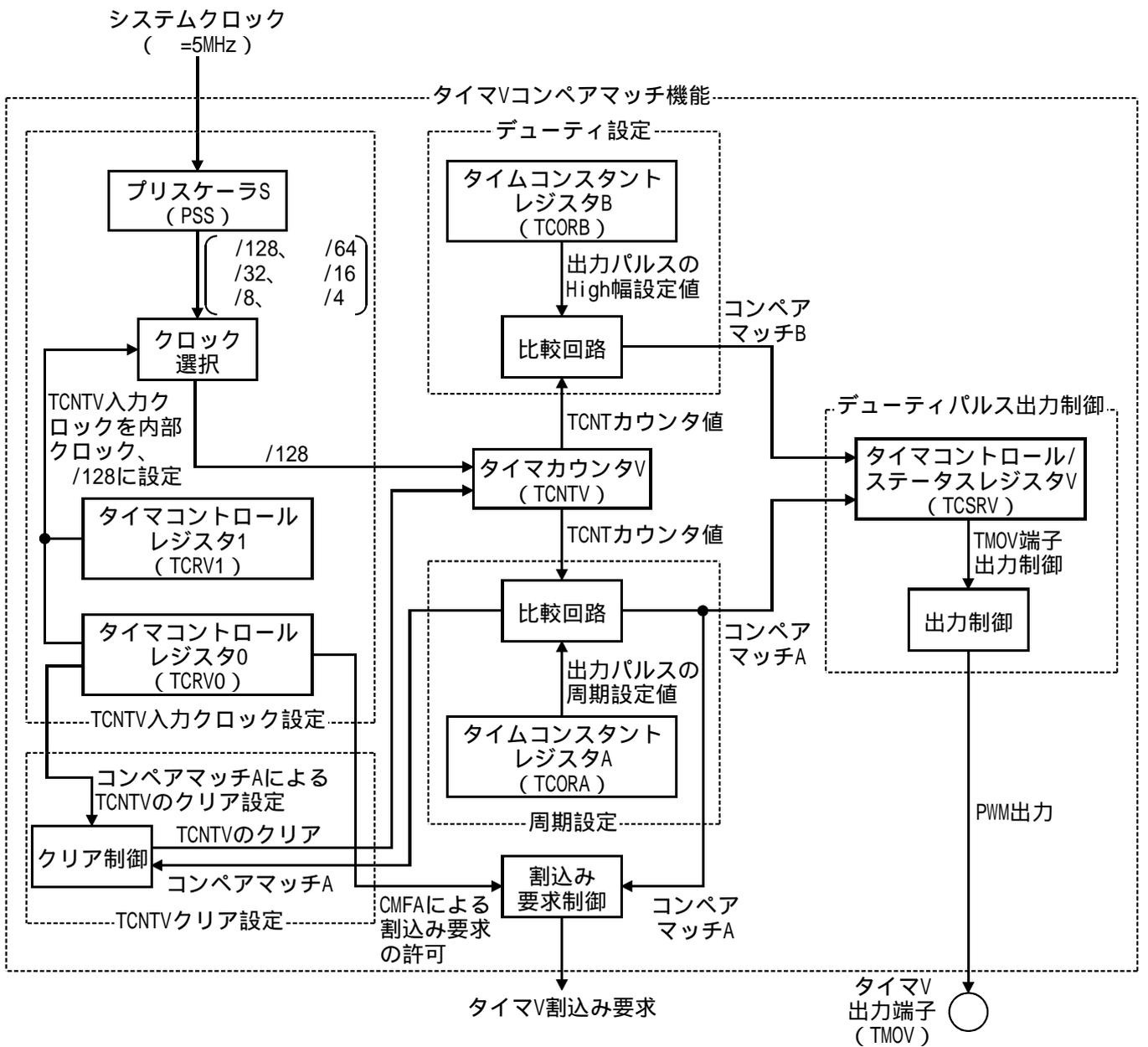


図2 タイマVコンペアマッチ機能ブロック図

使用機能説明

(b) 図3に本タスク例におけるPWM出力波形の周期、およびデューティの設定方法について示します。

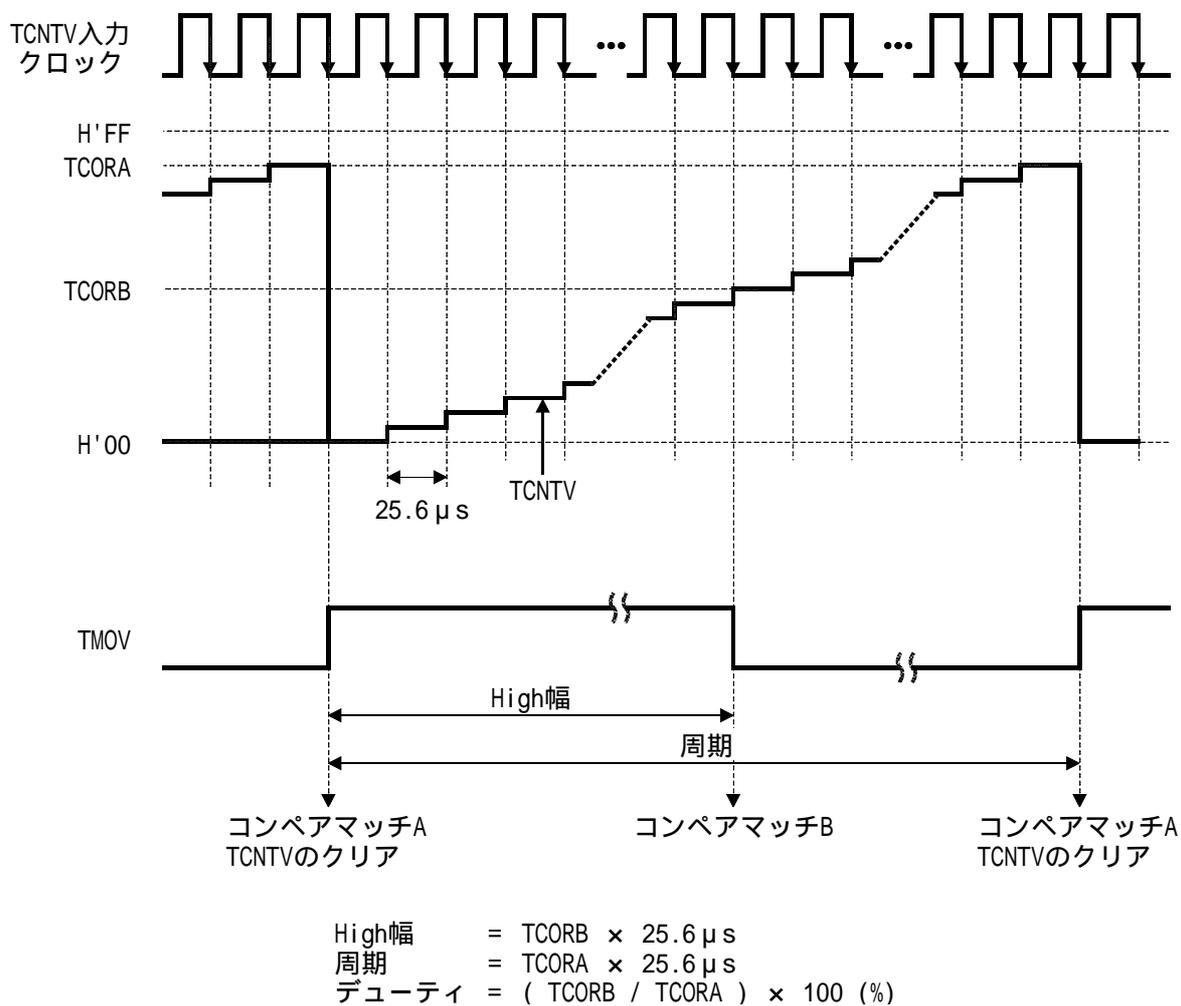


図3 PWM出力波形の周期、およびデューティの設定方法

使用機能説明

(c) 図4にデューティを大きくする場合のTCORBの書き換えのタイミングについて示します。

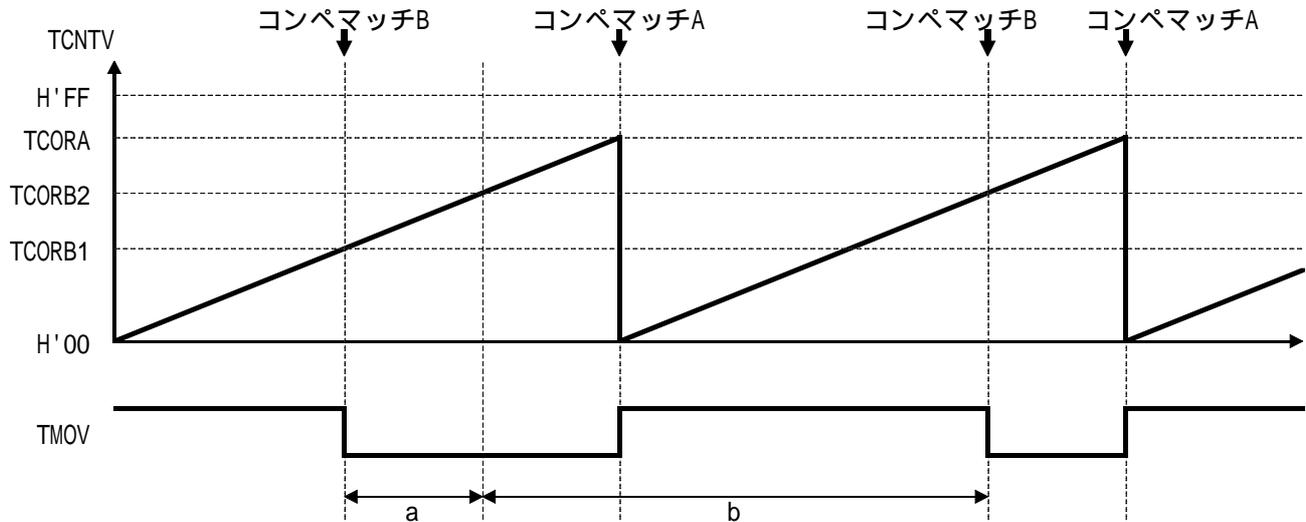


図4 デューティを大きくする場合のTCORB書き換えのタイミング

- ・TCORBの書き換えは、書き込むと同時にTCORBが更新されるため、デューティを大きくする場合、図4中の a の区間でTCORBを書き換えると、連続してコンペアマッチBが発生し、正常なPWM波形が出力されません。したがって、TCORBを書き換えるタイミングは、図4中の b の区間で書き換えなければなりません。
- ・本タスク例におけるTCORBの書き換えのタイミングは、コンペアマッチAによる割込みを使用して、コンペアマッチA割込み処理の中で、TCORBの値を書き換えています。

(d) 図5にデューティを小さくする場合のTCORBの書き換えのタイミングについて示します。

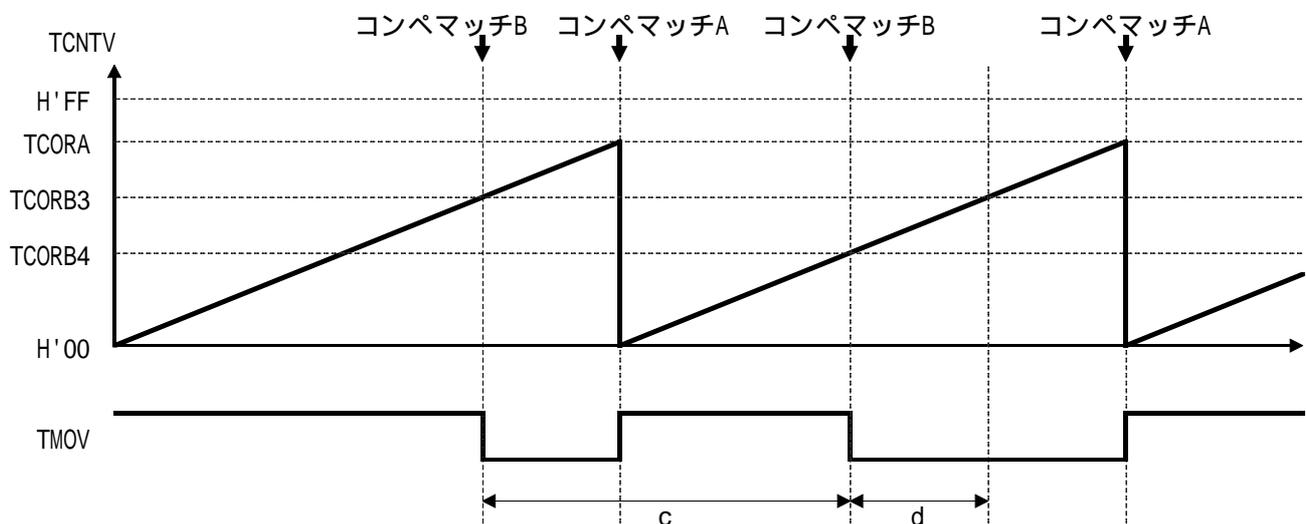


図5 デューティを小さくする場合のTCORB書き換えのタイミング

- ・TCORBの書き換えは、書き込むと同時にTCORBが更新されるため、デューティを小さくする場合、図5中の d の区間でTCORBを書き換えると、次のコンペアマッチBが発生せず、正常なPWM波形が出力されません。したがって、TCORBを書き換えるタイミングは、図5中の c の区間で書き換えなければなりません。
- ・本タスク例におけるTCORBの書き換えのタイミングは、コンペアマッチAによる割込みを使用して、コンペアマッチA割込み処理の中で、TCORBの値を書き換えています。

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマVコンペアマッチ機能によるPWM出力を行ないます。

表1 機能割付け

H8/3644機能	機能
PSS	システムクロックを入力とする13ビットのカウンタ
TCNTV	システムクロックの128分周のクロックを入力とする8ビットのカウンタ
TCORA	PWM出力の周期の設定
TCORB	PWM出力のHigh幅の設定
TCRV0	TCNTVの入力クロックの選択、およびTCNTVのクリア指定の選択
TCRV1	TCNTVの入力クロックの選択
TCSR	コンペアマッチ出力の制御
TMOV	PWM出力端子

動作原理

(1) 図6に動作原理を示します。図6に示すようなハードウェア処理、およびソフトウェア処理によりタイマVコンペアマッチ機能によるPWM出力を行ないます。

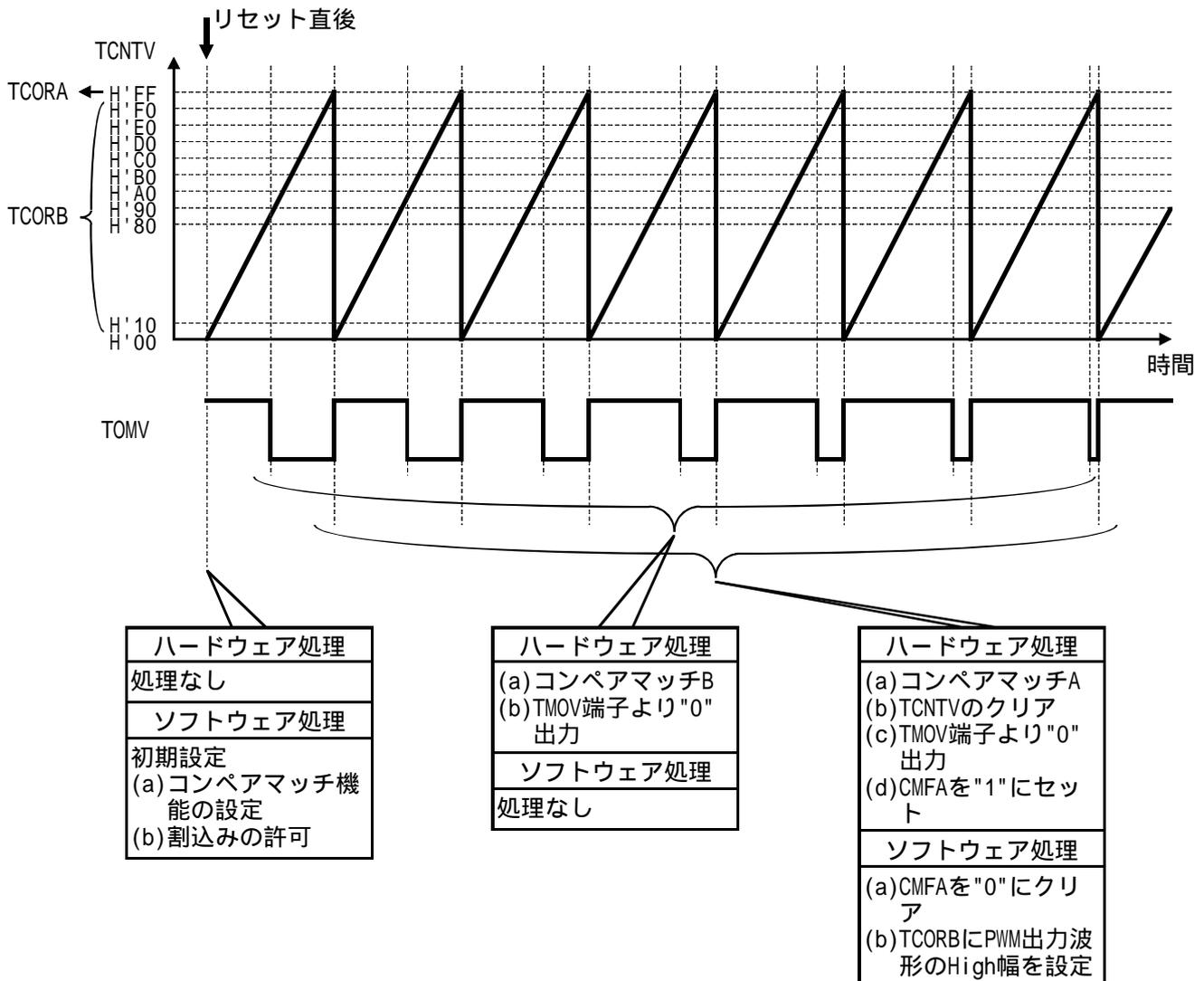


図6 タイマVコンペアマッチ機能によるPWM出力の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、コンペアマッチ機能の設定、割込みの許可を行なう
コンペアマッチA	TVINT	タイマVコンペアマッチA割込み処理ルーチンで、TCORBの書き換えを行なう

(2) 引数の説明

本タスク例では、引数は使用しません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TCRVO	CMIEA タイマコントロールレジスタV0 (コンペアマッチインタラプトイネーブルA) : CMIEA="1"のとき、CMFAによる割込み要求を許可	H'FFB8 ビット6	1
	CCLR1 CCLR0 タイマコントロールレジスタV0 (カウンタクリア1、0) : CCLR1="0"、CCLR0="1"のとき、コンペアマッチAによりTCNTVをクリア	H'FFB8 ビット4 ビット3	CCLR1="0" CCLR0="1"
	CKS2 CKS1 CKS0 タイマコントロールレジスタV0 (クロックセレクト2~0) : CKS2="0"、CKS1="1"、CKS0="1"、ICKS0="1"のとき、TCNTVはシステムクロックの128分周のクロックの立ち上がりエッジでカウント	H'FFB8 ビット2 ビット1 ビット0	CKS2="0" CKS1="1" CKS0="1"
TCSR V	CMFA タイマコントロール/ステータスレジスタV (コンペアマッチフラグA) : CMFA="0"のとき、コンペアマッチAが発生していない : CMFA="1"のとき、コンペアマッチAが発生	H'FFB9 ビット6	0
	OS3 OS2 OS1 OS0 タイマコントロール/ステータスレジスタV (アウトプットセレクト3~0) : OS3="0"、OS2="1"、OS1="1"、OS0="0"のとき、TMOV端子の出力レベルを、コンペアマッチBで"0"出力、コンペアマッチAで"1"出力に設定	H'FFB9 ビット3 ビット2 ビット1 ビット0	OS3="0" OS2="1" OS1="1" OS0="0"
TCORA	タイムコンスタントレジスタA : TCORA=H'FFのとき、TCNTVのカウント値がH'FFまでカウントアップするとコンペアマッチAが発生	H'FFBA	H'FF
TCORB	タイムコンスタントレジスタB : TCORB=H'80のとき、TCNTVのカウント値がH'80までカウントアップするとコンペアマッチBが発生	H'FFBB	H'80
TCNTV	タイマカウンタV : システムクロックの128分周のクロックを入力とする8ビットのアップカウンタ	H'FFBC	H'00
TCRV1	ICKS0 タイマコントロールレジスタV1 (インターナルクロックセレクト0) : ICKS0="1"のとき、TCNTVの入力クロックをシステムクロックの128分周のクロックに設定	H'FFBD ビット0	1

ソフトウェア説明

(4) 使用RAM説明

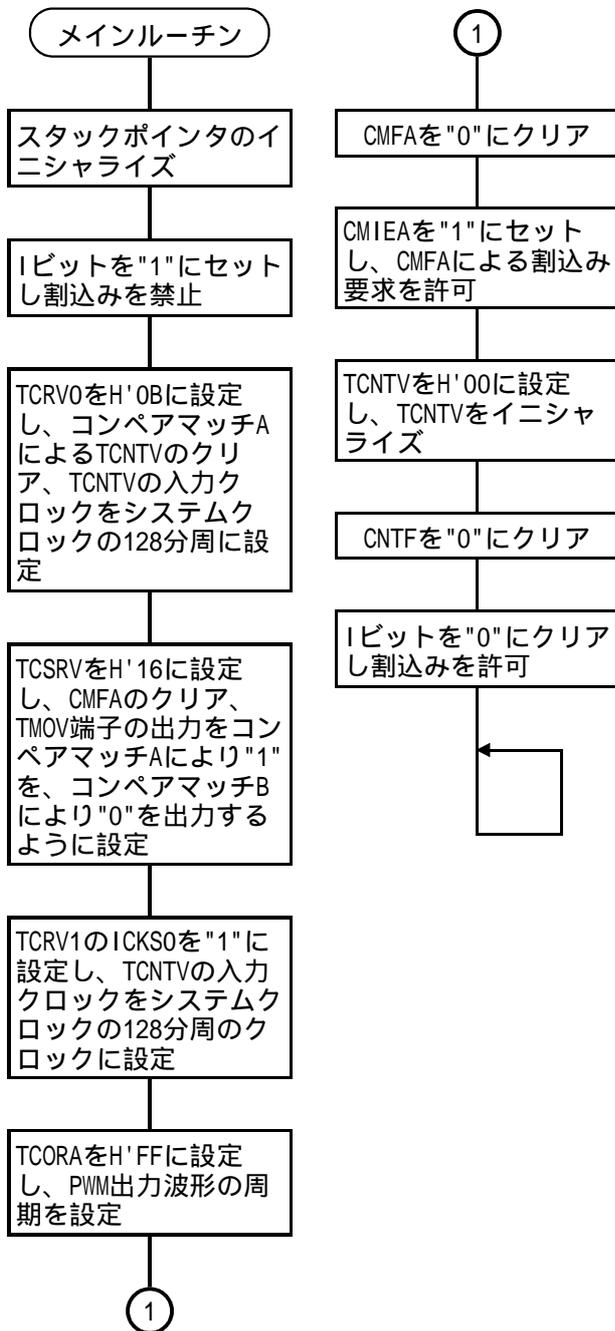
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

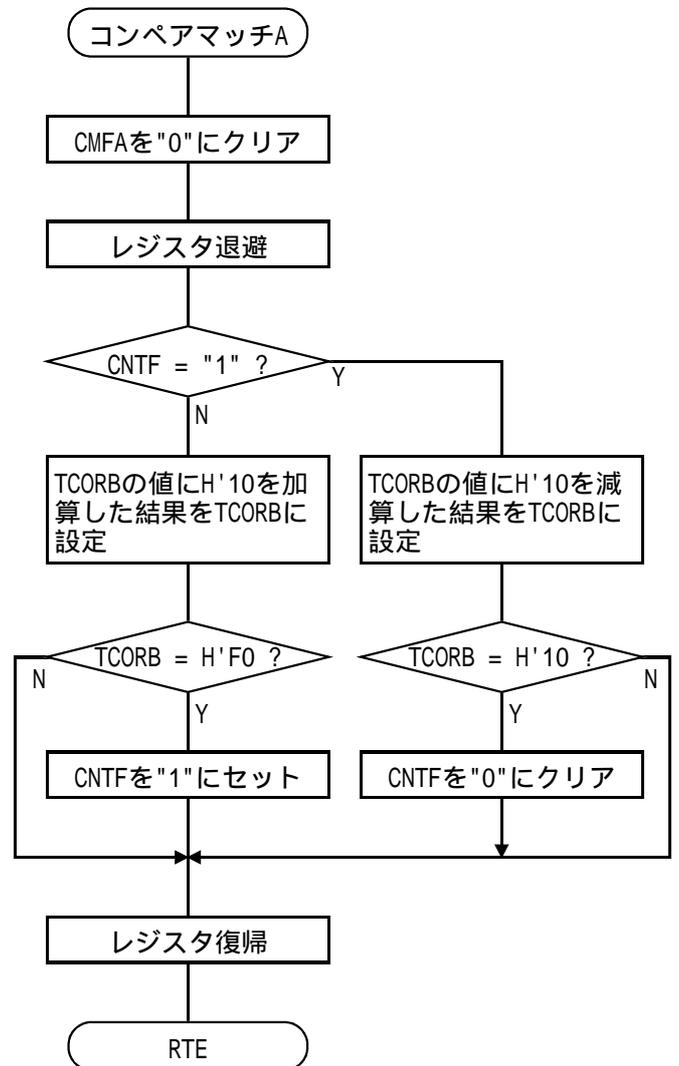
ラベル名	機能	アドレス	使用モジュール名	
USRFB	CNTF	TCORBの設定値をインクリメントするかデクリメントするかを判定するフラグ	H'FB80 ビット0	メインルーチン コンペアマッチA

フローチャート

(a) メインルーチン



(b) タイマV割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'PWM Output by Compare Match Function'
;
;       Function
;       :Timer V Compare Match
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definitio
;
*****
;
;
TCRVO      .equ       H'FFB8      ;Timer Control Register V0
CMIEB      .bequ     7,TCRVO     ;Compare Match Interrupt Enable B
CMIEA      .bequ     6,TCRVO     ;Compare Match Interrupt Enable A
OVIE       .bequ     5,TCRVO     ;Timer Overflow Interrupt Enable
CCLR1      .bequ     4,TCRVO     ;Counter Clear 1
CCLR0      .bequ     3,TCRVO     ;Counter Clear 0
CKS2       .bequ     2,TCRVO     ;Clock Select 2
CKS1       .bequ     1,TCRVO     ;Clock Select 1
CKS0       .bequ     0,TCRVO     ;Clock Select 0
TCSRV      .equ       H'FFB9      ;Timer Control/Status Register V
CMFB       .bequ     7,TCSRV     ;Compare Match Flag B
CMFA       .bequ     6,TCSRV     ;Compare Match Flag A
OVF        .bequ     5,TCSRV     ;Timer Overflow Flag
OS3        .bequ     3,TCSRV     ;Output Select 3
OS2        .bequ     2,TCSRV     ;Output Select 2
OS1        .bequ     1,TCSRV     ;Output Select 1
OS0        .bequ     0,TCSRV     ;Output Select 0
TCORA      .equ       H'FFBA      ;Time Constant Register A
TCORB      .equ       H'FFBB      ;Time Constant Register B
TCNTV      .equ       H'FFBC      ;Timer Counter V
TCRV1      .equ       H'FFBD      ;Timer Control Register V1
TVEG1      .bequ     4,TCRV1     ;TRGV Input Edge Select 1
TVEG0      .bequ     3,TCRV1     ;TRGV Input Edge Select 0
TRGE       .bequ     2,TCRV1     ;TRGV Input Enable
ICKSO      .bequ     0,TCRV1     ;Internal Clock Select 0
;
;
*****
;
;
;       RAM Allocation
;
*****
;
;
STACK      .equ       H'FF80      ;Stack Pointer
USRF       .equ       H'FB80      ;User Flag Area
CNTF       .bequ     0,USRF      ;Counter Flag
;
;
*****
;
;
;       Vector Address
;

```


プログラムリスト

```

;
MAIN9      .equ      $
           BRA      MAIN9
;
;
;*****
;
;           Timer V Interrupt
;*****
;
;
TVINT      .equ      $
           BCLR     CMFA      ;Clear CMFA to 0
;
           PUSH     R0        ;Store R0
;
           MOV.B    @USRF,R0L
           BTST     #0,R0L    ;CNTF = 1 ?
           BNE     TVINT2     ;Yes
;
           ADD.B    R2L,R1L   ;Increment "High" Width
           CMP.B    #H'F0,R1L ;"High" Width = H'F0 ?
           BEQ     TVINT1     ;Yes
           BRA     TVINT9     ;No
;
TVINT1     .equ      $
           BSET     #0,R0L    ;Set CNTF at 1
           BRA     TVINT9
;
TVINT2     .equ      $
           SUB.B    R2L,R1L   ;Decrement "High" Width
           CMP.B    #H'10,R1L ;"High" Width = H'10 ?
           BEQ     TVINT3     ;Yes
           BRA     TVINT9     ;No
;
TVINT3     .equ      $
           BCLR     #0,R0L    ;Clear CNTF to 0
;
TVINT9     .equ      $
           MOV.B    R1L,@TCORB ;Set TCORB Value
           MOV.B    R0L,@USRF
;
           POP     R0        ;Restore R0
;
           RTE      ;Return
           .end

```

2.7 トリガ入力カウント開始機能によるパルス出力

トリガ入力カウント開始機能によるパルス出力	使用機能	タイマV：トリガ入力カウント開始機能
仕様		
<p>(1) 図1に示すように、TRGV入力に対する任意の遅延時間のパルス幅のパルス出力を行ないます。</p> <p>(2) 出力パルスは、タイマV出力 (TMOV) 端子より出力します。</p> <p>(3) カウント開始トリガ入力端子 (TRGV) 入力からの遅延時間は、タイムコンスタントレジスタA (TCORA) により設定します。</p> <p>(4) 出力するパルスのパルス幅は、タイムコンスタントレジスタB (TCORB) により設定します</p> <p>(5) 本タスク例では、TRGV入力からの遅延時間が2.56ms、パルス幅が0.64msのパルスを出力します。</p> <div data-bbox="271 448 1276 828" style="text-align: center;"> <p>The diagram shows two waveforms: TRGV (top) and TMOV (bottom). TRGV is a square wave with two pulses. The first pulse falls, and after a delay (indicated by a double-headed arrow and dashed lines), a pulse appears on the TMOV line. The width of this TMOV pulse is also indicated by a double-headed arrow. The second TRGV pulse also shows a similar delay before a TMOV pulse is generated.</p> </div> <p style="text-align: center;">図1 トリガ入力カウント開始機能によるパルス出力</p>		
使用機能説明		
<p>(1) 本タスク例では、タイマVトリガ入力によるカウント開始機能を使用して、TMOV出力端子より、TRGV入力に対する任意の遅延時間のパルス幅のパルス出力を行ないます。</p> <p>(a) 図2にタイマVトリガ入力によるカウント開始機能のブロック図を示します。以下にタイマVトリガ入力によるカウント開始機能ブロック図について説明します。</p> <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・プリスケールS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・タイマカウンタV (TCNTV) は、8ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは を分周した6種類のクロックと、3種類の外部クロックより選択できます。 ・タイマコントロールレジスタV0 (TCRV0) は、8ビットのリード/ライト可能なレジスタで、TCNTVの入力クロックの選択、TCNTVのクリア指定を行います。本タスク例では、TCNTVの入力クロックに の64分周のクロックを、TCNTVのクリア指定にコンペアマッチBによるクリアを選択しています。 ・タイマコントロールレジスタV1 (TCRV1) は、8ビットのリード/ライト可能なレジスタで、TRGV端子の入力エッジの選択、TRGV端子入力によるTCNTVカウントアップの開始と、コンペアマッチによるTCNTVクリア時のTCNTVカウントアップの停止を許可/禁止、TCRV0とともにTCNTV入力クロックの選択を行ないます。 ・タイマコントロール/ステータスレジスタV (TCRV) は、8ビットのレジスタで、コンペアマッチ出力の制御を行ないます。本タスク例では、コンペアマッチBでTMOV端子より"0"を出力、コンペアマッチAでTMOV端子より"1"を出力するように設定しています。 ・タイムコンスタントレジスタA (TCORA) は、8ビットのリード/ライト可能なレジスタで、TCORAの内容はTCNTVと常に比較されており、両者の値が一致するとコンペアマッチAにより、TMOV端子より"1"を出力します。 ・タイムコンスタントレジスタB (TCORB) は、8ビットのリード/ライト可能なレジスタで、TCORBの内容はTCNTVと常に比較されており、両者の値が一致するとコンペアマッチBにより、TMOV端子より"0"を出力し、TCNTVをH'00にクリアします。 ・タイマV出力端子 (TMOV) よりパルスを出力します。 ・トリガ入力端子 (TRGV) より、トリガを入力します。 		

使用機能説明

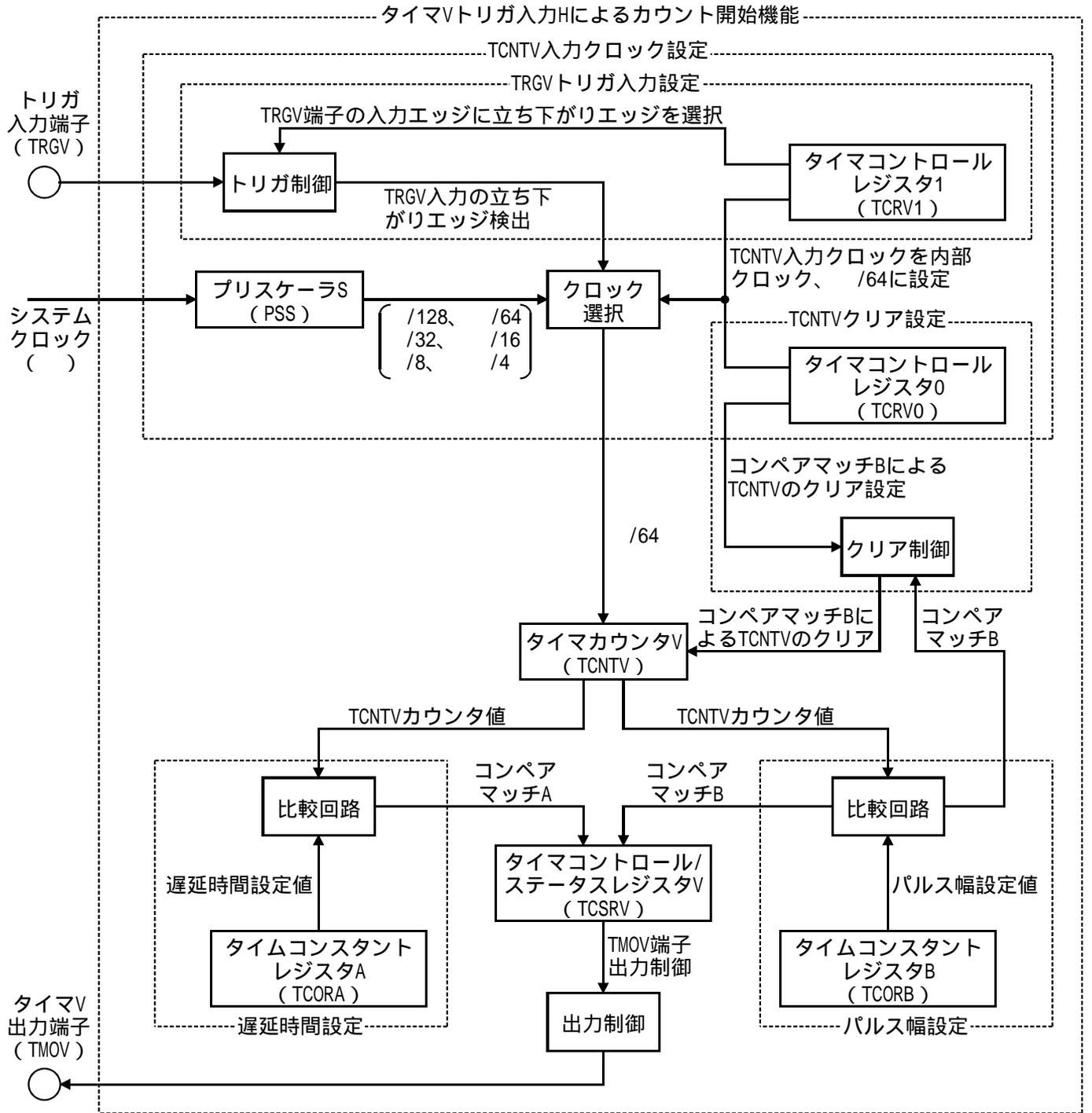
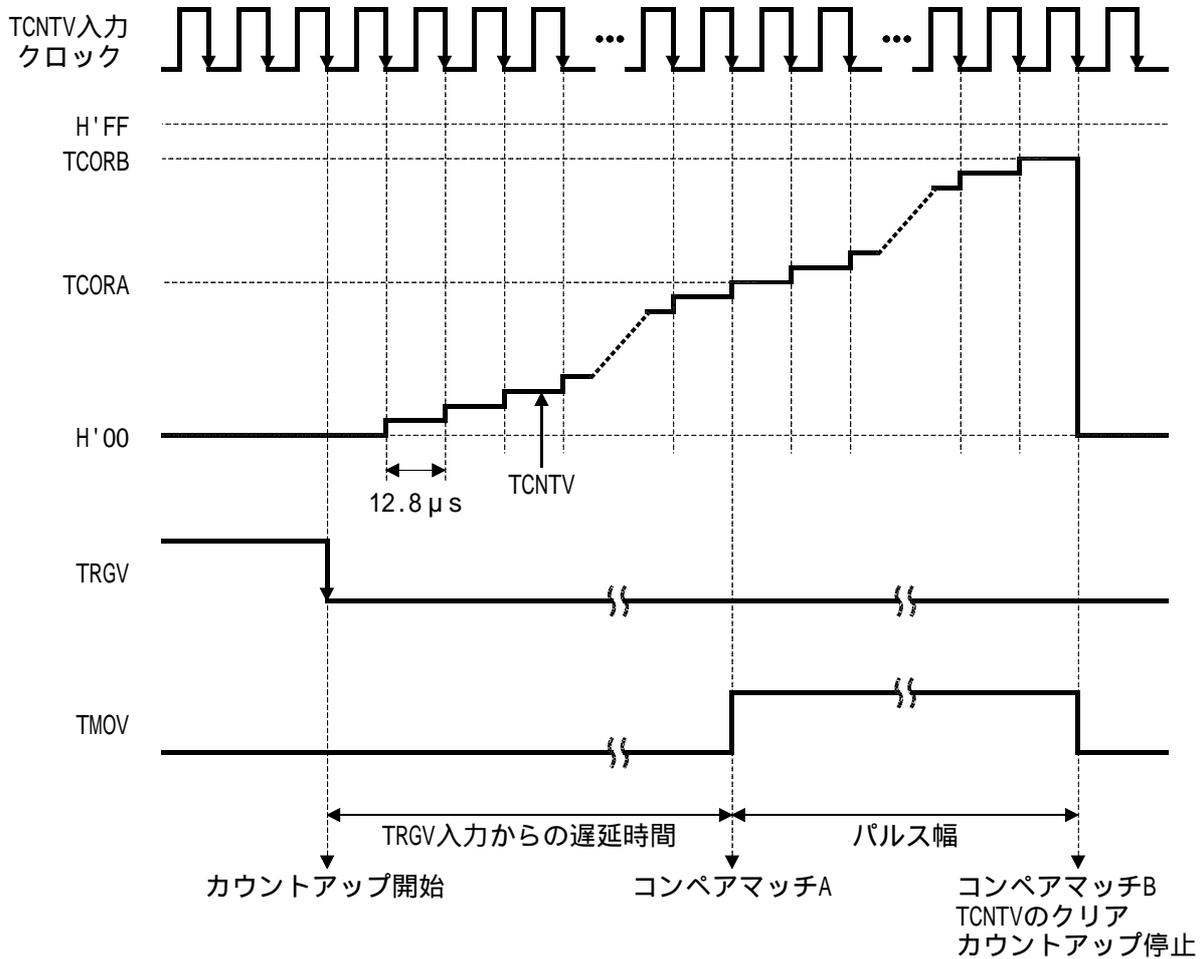


図2 トリガ入力によるカウンタ開始機能ブロック図

使用機能説明

(b) 図3にTRGV入力から任意の遅延時間のパルス幅のパルス出力における遅延時間と、パルス幅の設定方法について示します。



$$\begin{aligned} \text{TRGV入力からの遅延時間} &= \text{TCORA} \times 12.8 \mu\text{s} \\ \text{パルス幅} &= (\text{TCORB} - \text{TCORA}) \times 12.8 \mu\text{s} \end{aligned}$$

図3 TRGV入力からの遅延時間、およびパルス幅の設定方法

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマVトリガ入力によるカウント開始機能によるパルス出力を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCNTV	システムクロックの64分周のクロックを入力とする8ビットのカウンタ
TCORA	遅延時間設定によるコンペアマッチAの設定
TCORB	パルス幅設定によるコンペアマッチBの設定
TCRV0	TCNTVの入力クロックの選択、およびTCNTVのクリア指定の選択
TCRV1	TRGV端子のエッジセレクト、TRGV入力イネーブル、TCNTV入力クロックの設定
TCSR	コンペアマッチ出力の制御
TMOV	パルス出力端子
TRGV	トリガ入力端子

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理によりタイマVトリガ入力カウント開始機能によるパルス出力を行ないます。

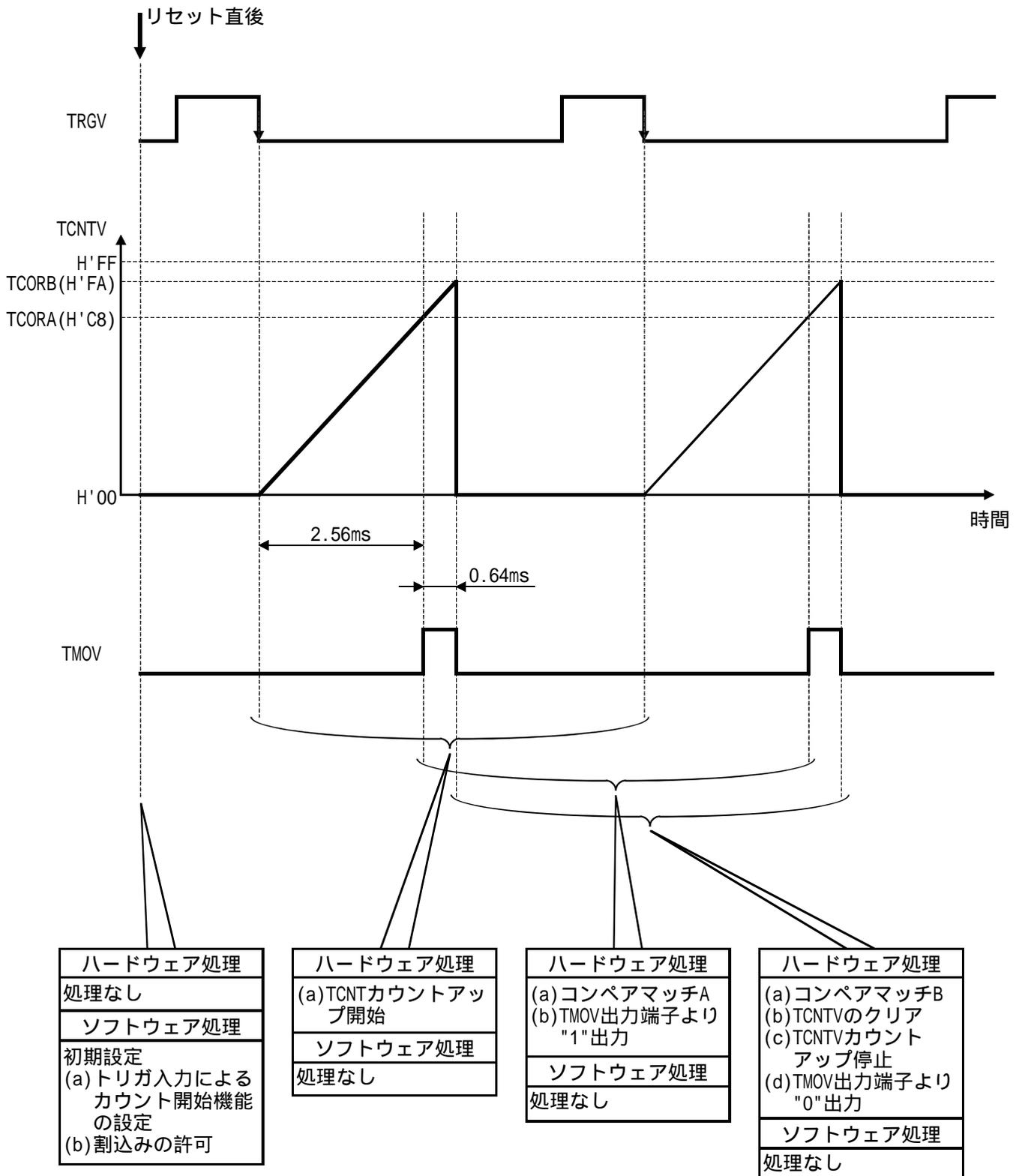


図4 タイマVトリガ入力カウントアップ開始機能によるパルス出力動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	TRGMN	スタックポインタのイニシャライズ、トリガ入力によるカウンタ開始機能の設定、割込みの許可を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
R1L	出力するパルスの遅延時間で、TCORAの設定値	メインルーチン	1バイト	入力
R1H	出力するパルスのパルス幅で、TCORBの設定値	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

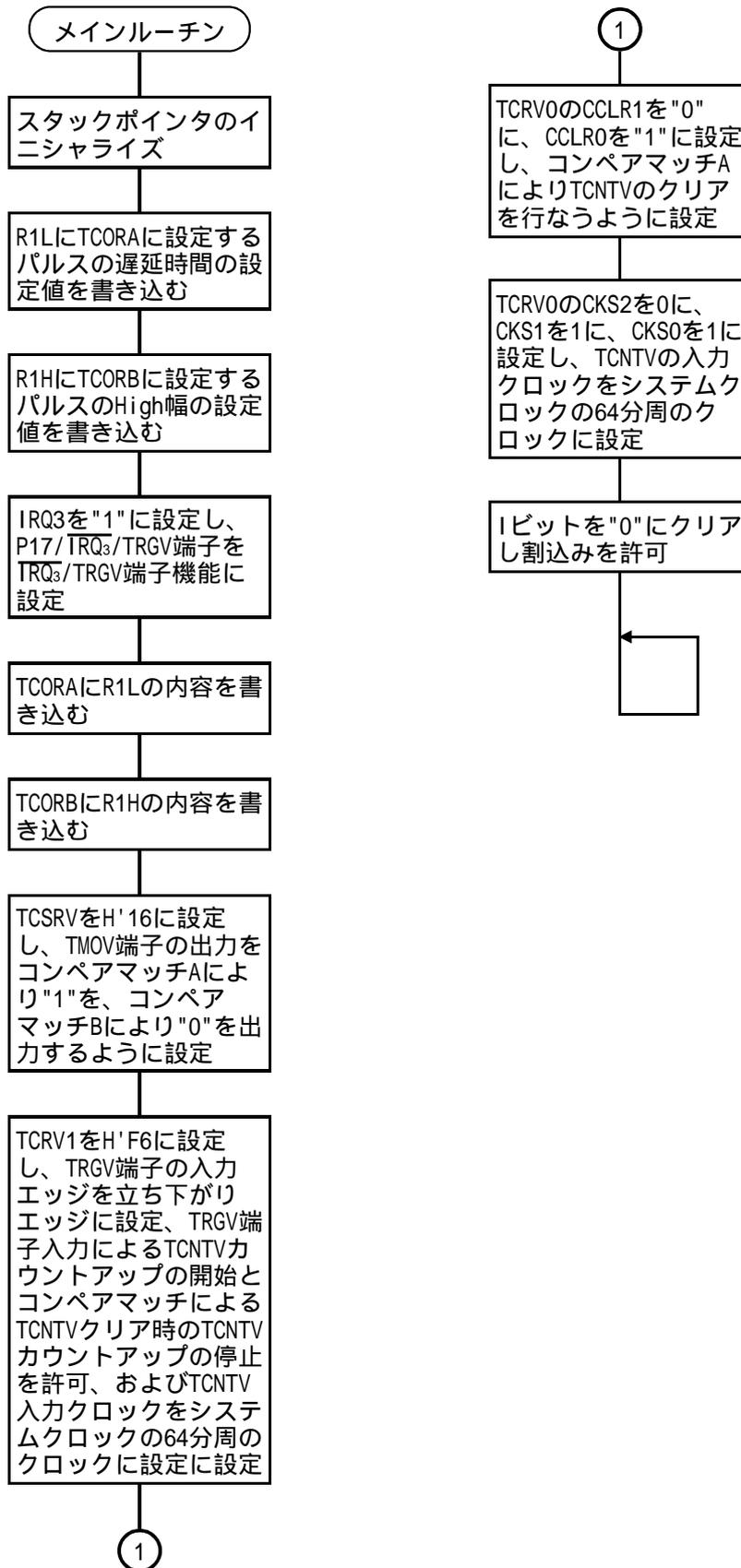
レジスタ名	機能	アドレス	設定値
TCRV0	タイマコントロールレジスタV0 : TCRV0=H'13のとき、CMFB、CMFA、OVFによる割込みの禁止、TCNTVのクリア指定をコンペアマッチBによるクリアに設定、およびTCNTVの入力をシステムクロックの64分周のクロックに設定	H'FFB8	H'13
TCSRv	タイマコントロール/ステータスレジスタV : TCSRv=H'16のとき、コンペアマッチAによりTMOV端子より"1"を出力、コンペアマッチBによりTMOV端子より"0"を出力するように設定	H'FFB9	H'16
TCORA	タイムコンスタントレジスタA : TCORA=H'C8のとき、TCNTVのカウント値がH'C8までカウントアップするとコンペアマッチAが発生	H'FFBA	H'C8
TCORB	タイムコンスタントレジスタB : TCORB=H'FAのとき、TCNTVのカウント値がH'FAまでカウントアップするとコンペアマッチBが発生	H'FFBB	H'FA
TCNTV	タイマカウンタV : システムクロックの64分周のクロックを入力とする8ビットのアップカウンタ	H'FFBC	H'00
TCRV1	タイマコントロールレジスタV1 : TCRV1=H'F6のとき、TRGV端子の入力エッジを立ち下がりエッジに設定、TRGV端子入力によるTCNTVカウントアップの開始とコンペアマッチによるTCNTVクリア時のTCNTVカウントアップの停止を許可、およびTCNTV入力クロックをシステムクロックの64分周のクロックに設定	H'FFBD	H'F6
PMR1 IRQ3	ポートモードレジスタ1 (P17/TRQ3/TRGV端子機能切り替え) : IRQ3="1"のとき、P17/TRQ3/TRGV端子をTRQ3/TRGV入力端子機能に設定	H'FFFC ビット7	1

(4) 使用RAM説明

本タスク例では、RAMは使用しません。

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Pulse Output by Trigger Input Count
;           Start Function'
;
;           Function
;           : Timer V Trigger Input Count Start Function'
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
*****
;
;           .cpu          300L
;
*****
;
;           Symbol Definition
;
*****
TCRVO      .equ          H'FFB8      ;Timer Control Register V0
TCSR      .equ          H'FFB9      ;Timer Control/Status Register V
TCORA     .equ          H'FFBA      ;Time Constant Register A
TCORB     .equ          H'FFBB      ;Time Constant Register B
TCNTV     .equ          H'FFBC      ;Timer Counter V
TCRV1     .equ          H'FFBD      ;Timer Control Register V1
PMR1      .equ          H'FFFC      ;Port Mode Register 1
IRQ3      .beq          7,PMR1      ;P17/IRQ3/TRGV Terminal Function Change
;
;
*****
;
;           Ram Allocation
;
*****
;
STACK     .equ          H'FF80      ;Stack Pointer
;
;
;           Vector Address
;
*****
;
;           .org          H'0000
;           .data.w      MAIN      ;Reset Interrupt
;
;
;           .org          H'0008
;           .data.w      MAIN      ;IRQ0 Interrupt
;           .data.w      MAIN      ;IRQ1 Interrupt
;           .data.w      MAIN      ;IRQ2 Interrupt
;           .data.w      MAIN      ;IRQ3 Interrupt
;           .data.w      MAIN      ;INT0 - INT7 Interrupt
;
;
;           .org          H'0014
;           .data.w      MAIN      ;Timer A Interrupt
;           .data.w      MAIN      ;Timer B1 Interrupt
;
;
;           .org          H'0020
;           .data.w      MAIN      ;Timer X Interrupt

```

プログラムリスト

```

;
;      .data.w      MAIN          ;Timer V Interrupt
;
;      .org         H'0026
;      .data.w      MAIN          ;Sci1 Interrupt
;
;      .org         H'002A
;      .data.w      MAIN          ;Sci3 Interrupt
;      .data.w      MAIN          ;A/D Converter Interrupt
;      .data.w      MAIN          ;Sleep Interrupt
;
;*****
;
;      Main Program
;*****
;
;      .org         H'1000
;
;
MAIN9  .equ          $
      MOV.W        #STACK,SP      ;Initialize Stack Pointer
      ORC          #H'80,CCR      ;Interrupt Disable
;
;      MOV.B        #H'C8,R1L      ;Set Output Pulse Delay Time
;      MOV.B        #H'FA,R1H      ;Set Output Pulse Width
;
;      BSET         IRQ3          ;Initialize TRGV Input Terminal Function
;
;      MOV.B        R1L,@TCORA     ;Initialize Compare Match A
;      MOV.B        R1H,@TCORB     ;Initialize Compare Match B
;
;      MOV.W        #H'16F6,R0     ;Initialize TMOV Terminal Output
;      MOV.B        R0H,@TCSR      ;Initialize Trigger Input Function
;      MOV.B        R0L,@TCRV1
;
;      MOV.B        #H'13,R0L      ;Initialize TCNTV Clear Condition & TCNTV Input Clock
;      MOV.B        R0L,@TCRV0
;
;      ANDC         #H'7F,CCR      ;Interrupt Enable
;
MAIN9  .equ          $
      BRA          MAIN9
;
;      .end

```

2.8 16ビットフリーランニング機能による割込み回数のカウント

16ビットフリーランニング機能による 割込み回数のカウント	使用機能	タイマX：16ビットフリーランニング機能
仕様		
<p>(1) タイマX16ビットフリーランニングタイマ機能を使用して、タイマX割込み回数をカウントし、50回の割込みをカウントしたところで終了します。</p> <p>(2) フリーランニングカウンタ (FRC) がオーバーフローすることにより、タイマXのオーバーフローによる割込み要求が発生し、タイマX割込み処理の中でR1Lに設定した8ビットカウンタをデクリメントします。</p> <p>(3) R1Lに設定する8ビットカウンタはタイマX割込み処理の中でデクリメントします。そして、50回のタイマX割込みをカウントしたところで、タイマX割込み要求を禁止して終了します。</p> <p>(4) タイマX割込みは、FRCがオーバーフローする時間419.4304msごとに発生するように設定します。</p>		

使用機能説明
<p>(1) 本タスク例では、タイマXフリーランニング機能を使用して、タイマX割込み回数のカウントを行ないます。</p> <p>(a) 図1にタイマXフリーランニング機能のブロック図を示します。以下にタイマXフリーランニング機能のブロック図について説明します。</p> <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・プリスケールS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・フリーランニングカウンタ (FRC) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックはシステムクロックの2分周、8分周、32分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、FRCの入力クロックにシステムクロックの32分周のクロックを選択しています。 ・タイマコントロールレジスタX (TCRX) は、8ビットのリード/ライト可能なレジスタで、FRCの入力クロックの選択を行ないます。 ・タイマコントロール/ステータスレジスタX (TCSRX) は、8ビットのレジスタで、割込み要求信号の制御を行ないます。 ・タイマインタラプトイネーブルレジスタ (TIER) は、8ビットのリード/ライト可能なレジスタで、各割込み要求の許可/禁止を制御します。本タスク例では、タイマオーバーフローによる割込み要求を許可し、それ以外の割り込みは禁止しています。 ・以下に、本タスク例におけるFRCのオーバーフロー周期の計算法を示します。 $\begin{aligned} \text{FRCオーバーフロー周期} &= \frac{1}{\text{システムクロック} / 32} \times 65536 \\ &= 419.4304\text{ms} \end{aligned}$ <p style="text-align: center;">図1 フリーランニング機能ブロック図</p>

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマXフリーランニング機能による割り込み回数のカウントを行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
FRC	システムクロックの32分周のクロックを入力とする16ビットのカウンタ
TCRX	FRC入力クロックの設定
TCSRX	タイマオーバーフローによる割り込み要求信号の制御
TIER	タイマオーバーフローによる割り込み要求の許可

動作説明

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマXフリーランニング機能による割り込み回数のカウントを行ないます。

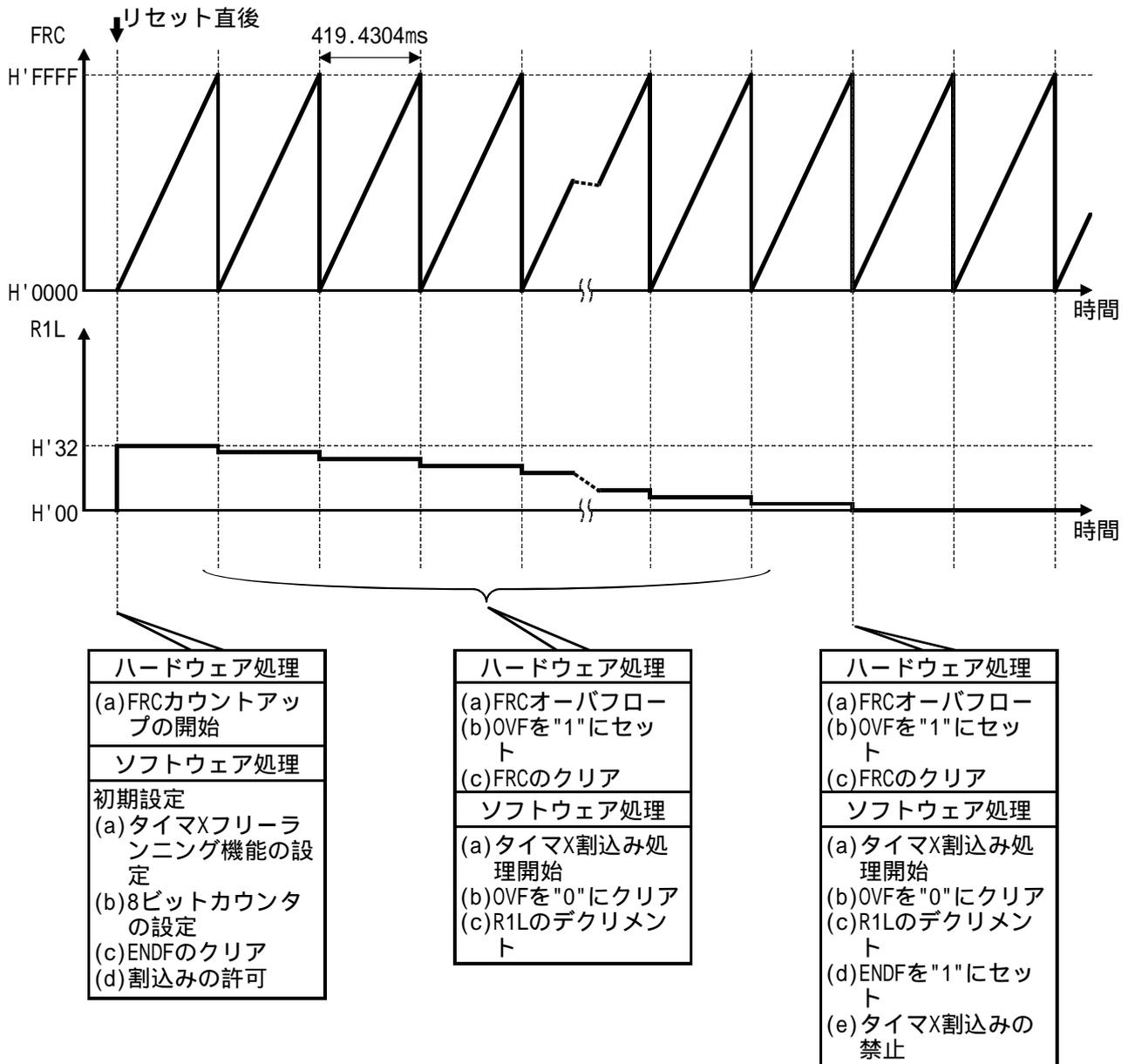


図2 タイマX16ビットフリーランニング機能による割り込み回数のカウント動作原理

16ビットフリーランニング機能による 割込み回数のカウント	使用機能	タイマX : 16ビットフリーランニング機能
----------------------------------	------	------------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、16ビットフリーランニング機能の設定、8ビットカウンタの設定、割込みの許可、およびENDFが"1"にセットされた時点で終了する
割込みカウント	TXINT	タイマX割込み処理で、8ビットカウンタのデクリメント、カウンタ値がH'00の時点でENDFを"1"にセットし、タイマX割込みを禁止する

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TIER OVIE	タイマインタラプトイネーブルレジスタ (タイマオーバーフロー割込みイネーブル) : OVIE="0"のとき、OVFによる割込み要求を禁止 : OVIE="1"のとき、OVFによる割込み要求を許可	H'F770 ビット1	1
TCSRX OVF	タイマコントロール/ステータスレジスタX (タイマオーバーフロー) : OVF="0"のとき、FRCがオーバーフローしていないことを示す : OVF="1"のとき、FRCがオーバーフローしたことを示す	H'F771 ビット1	0
FRCH	フリーランニングカウンタH : システムクロックの32分周のクロックを入力とする16ビットのアップカウンタの上位8ビット	H'F772	H'00
FRCL	フリーランニングカウンタL : システムクロックの32分周のクロックを入力とする16ビットのアップカウンタの下位8ビット	H'F773	H'00
TCRX CKS1 CKS0	タイマコントロールレジスタX (クロックセレクト1、0) : CKS1="1"、CKS0="0"のとき、FRC入力クロックをシステムクロックの32分周のクロックに設定	H'F776 ビット1 ビット0	CKS1="0" CKS0="0"

(4) 使用RAM説明

表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

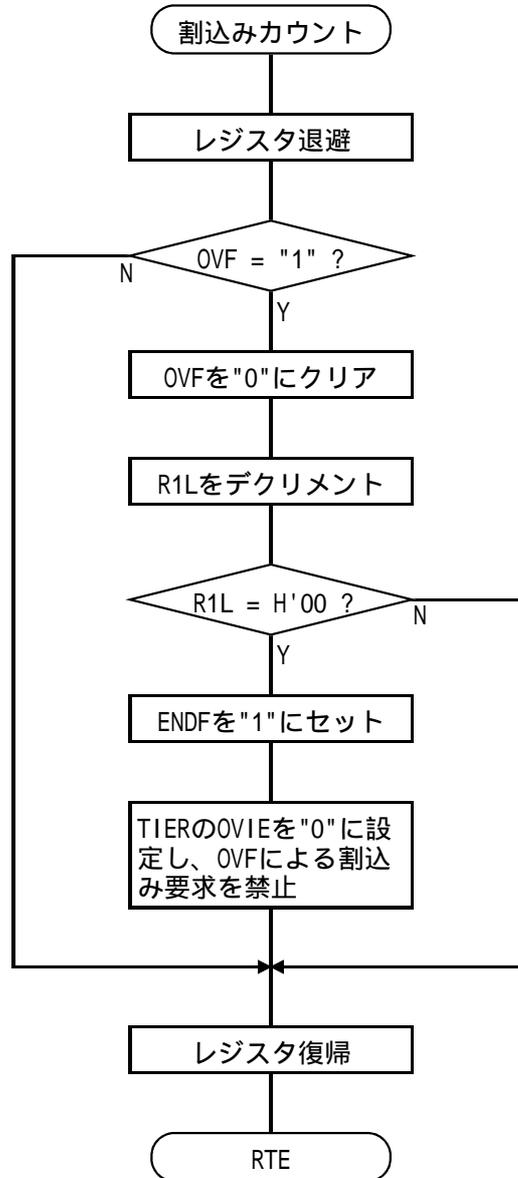
ラベル名	機能	アドレス	使用モジュール名
USRF ENDF	8ビットカウンタのカウンタ値がH'00になったかを判定するフラグ	H'FB80 ビット0	メインルーチン 割込みカウント

フローチャート

(a) メインルーチン



(b) タイマX割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Interrupt Counting by 16-bit Free Running
;       Function'
;
;       Function
;       : Timer X 16bit Free Running Timer
;
;       External Clock : 10MHz
;       Internal Clock  : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
TIER      .equ        H'F770      ;Timer Interrupt Enable Register
OVIE      .bequ       1,TIER      ;Timer Overflow Interrupt Enable
TCSRX     .equ        H'F771      ;Timer Control/Status Register X
OVF       .bequ       1,TCSRX     ;Timer Overflow
FRCH      .equ        H'F772      ;Free Running Counter H
FRCL      .equ        H'F773      ;Free Running Counter L
TCRX      .equ        H'F776      ;Timer Control Register X
CKS1      .bequ       1,TCRX      ;Clock Select 1
CKS0      .bequ       0,TCRX      ;Clock Select 0
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;
STACK     .equ        H'FF80      ;Stack Pointer
USRF      .equ        H'FB80      ;User Flag Area
ENDF      .bequ       0,USRF      ;End Flag
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.w       MAIN          ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w       MAIN          ;IRQ0 Interrupt
;       .data.w       MAIN          ;IRQ1 Interrupt
;       .data.w       MAIN          ;IRQ2 Interrupt
;       .data.w       MAIN          ;IRQ3 Interrupt
;       .data.w       MAIN          ;INT0 - INT7 Interrupt
;
;
;       .org          H'0014
;       .data.w       MAIN          ;Timer A Interrupt
;       .data.w       MAIN          ;Timer B1 Interrupt
;

```


プログラムリスト

```

MOV.B    #H'01,ROL
MOV.B    ROL,@USRF    ;Set ENDF
MOV.B    ROL,@TIER    ;OVF Interrupt Disable
;
TXINT9   .equ        $
POP      R0            ;Restore R0
;
RTE
;
.end

```

2.9 16ビットイベントカウンタ機能による入力パルス数の測定

16ビットイベントカウンタ機能による入力パルス数の測定	使用機能	タイマX : 16ビットイベントカウンタ機能
仕様		
<p>(1) タイマXの16ビットイベントカウンタ機能を使用して、カウンタクロック入力端子(FTCI)に入力されたパルスの立ち上がりエッジをカウントします。</p> <p>(2) フリーランニングカウンタ (FRC) を外部クロックの立ち上がりエッジでカウントするように設定し、1024回の立ち上がりエッジをカウントするまでカウントアップを行ないます。</p> <p>(3) 1024回の立ち上がりエッジをカウントすると、FRCの外部クロック入力を止め、終了します。</p>		

使用機能説明

- (1) 本タスク例では、タイマXイベントカウンタ機能を使用して、FTCI入力端子に入力したパルスのパルス数の測定を行ないます。
- (a) 図1にタイマXフリーランニング機能のブロック図を示します。以下にタイマXイベントカウンタ機能のブロック図について説明します。
- ・フリーランニングカウンタ (FRC) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックはシステムクロックの2分周、8分周、32分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、FRCの入力クロックに外部クロックを選択しています。
 - ・タイマコントロールレジスタX (TCRX) は、8ビットのリード/ライト可能なレジスタで、FRCの入力クロックの選択を行ないます。
 - ・タイマコントロール/ステータスレジスタX (TCSRX) は、8ビットのレジスタで、FRCの割込み要求信号の制御を行ないます。
 - ・タイマインタラプトイネーブルレジスタ (TIER) は、8ビットのリード/ライト可能なレジスタで、各割込み要求の許可/禁止を制御します。本タスク例では、タイマオーバーフローによる割込み要求を許可し、それ以外の割込みは禁止します。
 - ・カウンタクロック入力端子 (FTCI) より、外部クロックを入力します。

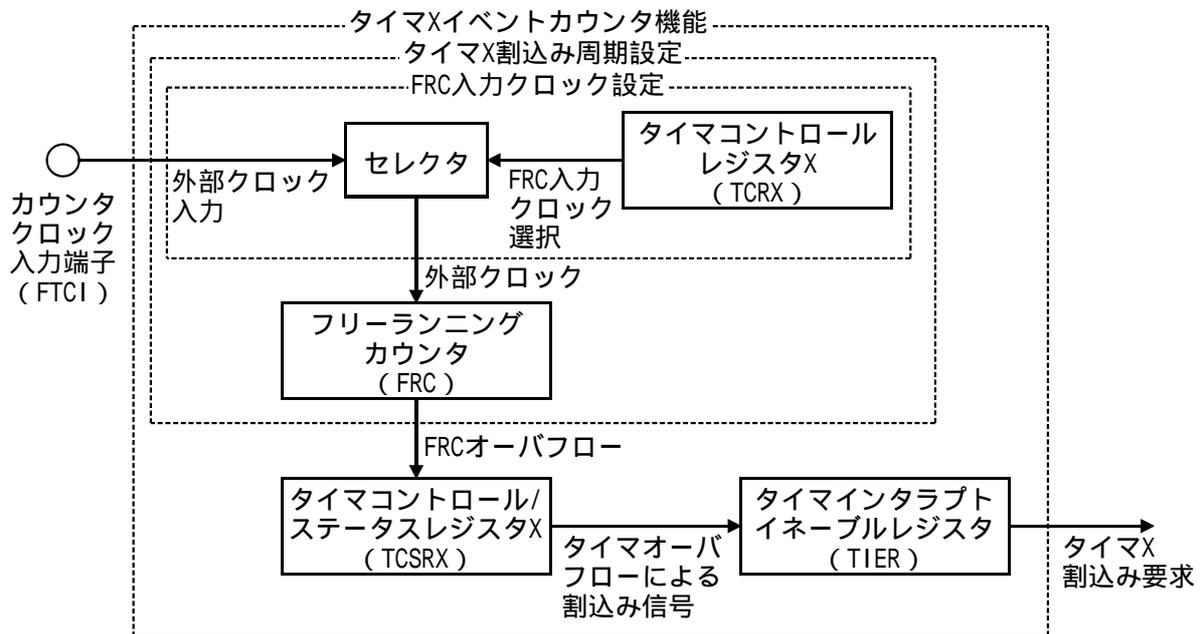


図1 イベントカウンタ機能ブロック図

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマXイベントカウンタ機能による入力パルス数の測定を行ないます。

表1 機能割付け

機能	機能割付け
FRC	外部クロックを入力とする16ビットのカウンタ
TCRX	FRC入力クロックの設定
TCSRX	タイマオーバーフローによる割込み要求信号の制御
TIER	タイマオーバーフローによる割込み要求の許可
FTCI	パルスの入力端子

動作説明

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりタイマXイベントカウンタ機能による入力パルス数の測定を行ないます。

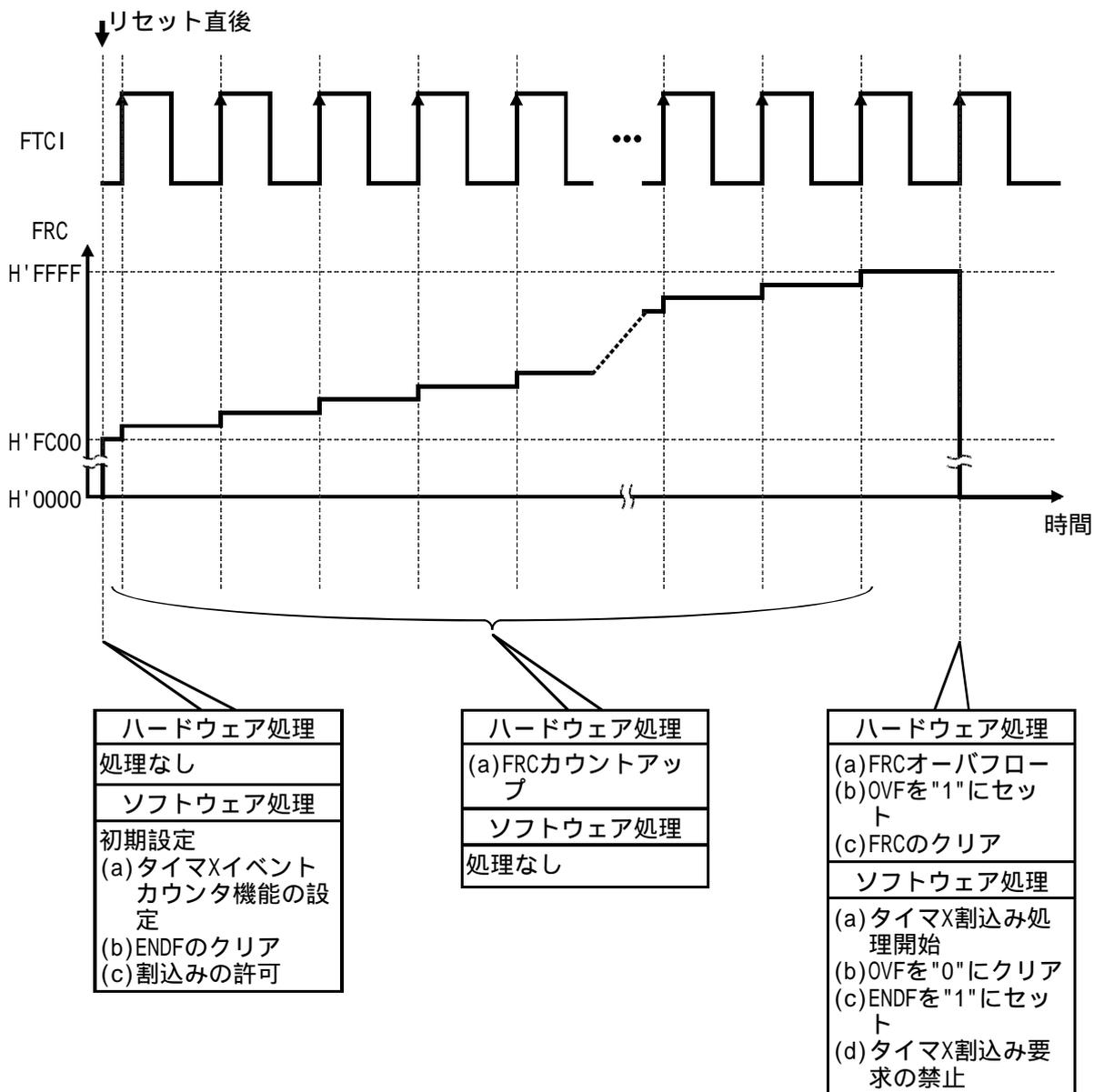


図2 タイマX16ビットイベントカウンタ機能による入力パルス数の測定動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、16ビットイベントカウンタ機能の設定、割込みの許可、1024回のFRC入力クロックの立ち上がりエッジをカウントしたところで終了する
測定終了	TXINT	タイマX割込み処理で、ENDFを"1"にセットし、タイマX割込みを禁止する

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TIER	OVIE	H'F770 ビット1	1
TCSRX	OVF	H'F771 ビット1	0
FRCH	フリーランニングカウンタH : システムクロックの2分周のクロックを入力とする16ビットのアップカウンタの上位8ビット	H'F772	H'FC
FRCL	フリーランニングカウンタL : システムクロックの2分周のクロックを入力とする16ビットのアップカウンタの下位8ビット	H'F773	H'00
TCRX	CKS1 CKS0	H'F776 ビット1 ビット2	CKS1="1" CKS0="1"
	タイマコントロールレジスタX (クロックセレクト1、0) : CLS1="1"、CLS0="1"のとき、FRC入力クロックを外部クロックに設定		

(4) 使用RAM説明

表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

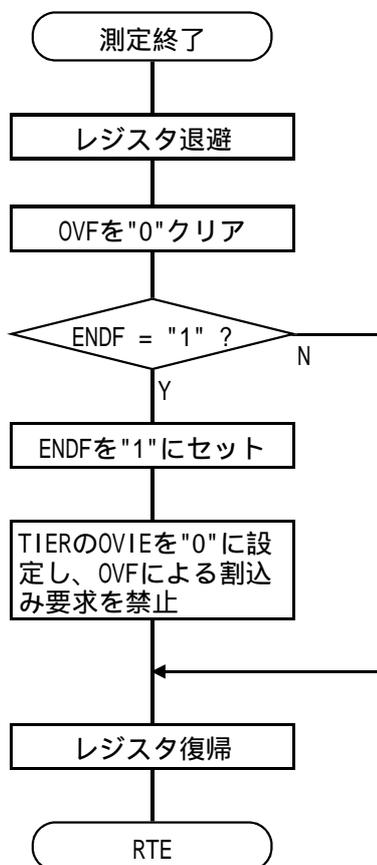
ラベル名	機能	アドレス	使用モジュール名
USRF	ENDF	H'FB80 ビット0	メインルーチン 割込みカウント

フローチャート

(a) メインルーチン



(b) タイマX割込み処理ルーチン



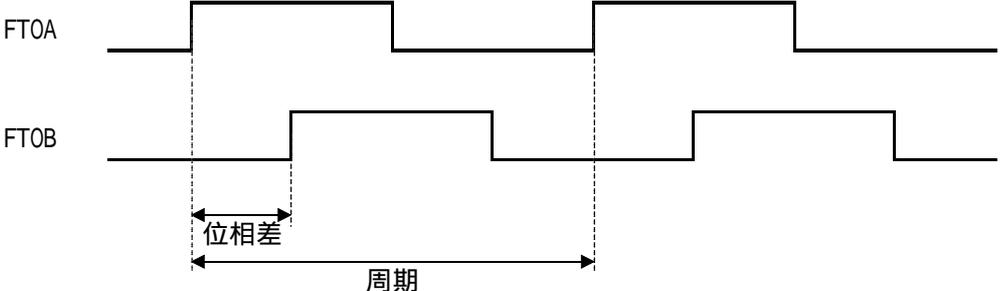
プログラムリスト

```
*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Measurement of Input Pulses by 16-bit
;           Event Counter Function'
;
;           Function
;           : Timer X 16bit Event Counter
;
;           External Clock : 10MHz
;           Internal Clock :  5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
TIER          .equ        H'F770          ;Timer Interrupt Enable Register
OVIE          .bequ      1,TIER          ;Timer Overflow Interrupt Enable
TCSRX        .equ        H'F771          ;Timer Control/Status Register X
OVF          .bequ      1,TCSRX         ;Timer Overflow
FRCH         .equ        H'F772          ;Free Running Counter H
FRCL         .equ        H'F773          ;Free Running Counter L
TCRX         .equ        H'F776          ;Timer Control Register X
CKS1         .bequ      1,TCRX          ;Clock Select 1
CKS0         .bequ      0,TCRX          ;Clock Select 0
;
;
*****
;
;           Ram Allocation
;
*****
STACK        .equ        H'FF80          ;Stack Pointer
USRF         .equ        H'FB80          ;User Flag Area
ENDF         .bequ      0,USRF          ;End Flag
;
;
;           Vector Address
;
*****
;
;           .org          H'0000
;           .data.w      MAIN            ;Reset Interrupt
;
;
;           .org          H'0008
;           .data.w      MAIN            ;IRQ0 Interrupt
;           .data.w      MAIN            ;IRQ1 Interrupt
;           .data.w      MAIN            ;IRQ2 Interrupt
;           .data.w      MAIN            ;IRQ3 Interrupt
;           .data.w      MAIN            ;INT0 - INT7 Interrupt
;
;
;           .org          H'0014
;           .data.w      MAIN            ;Timer A Interrupt
;           .data.w      MAIN            ;Timer B1 Interrupt
```


プログラムリスト

```
BTST      #1,R0L      ;OVF = "1" ?
BEQ       TXINT9      ;No.
;
BCLR      #1,R0L      ;Clear OVF
MOV.B     R0L,@TCSRX
;
MOV.W     #H'0100,R0
MOV.B     R0H,@USRF   ;Set ENDF
MOV.B     R0L,@TIER   ;OVF Interrupt Disable
;
TXINT9    .equ        $
POP       R0          ;Restore R0
;
RTE
;
.end
```

2.10 アウトプットコンペア機能による任意の位相差のパルス出力

アウトプットコンペア機能による任意の位相差のパルス出力	使用機能	タイマX : アウトプットコンペア機能
仕様		
<p>(1) 図1に示すように、タイマXアウトプットコンペア機能を使用して、デューティ50%のパルスを任意の位相差で出力させます。</p> <p>(2) FTOA出力端子およびFTOB出力端子より、任意の位相差のパルスを出します。</p> <p>(3) パルスの周期は、アウトプットコンペアレジスタA (OCRA) により設定します。</p> <p>(4) FTOA、FTOB端子より出力するパルスの位相差は、アウトプットコンペアレジスタB (OCRB) により設定します。</p> <p>(5) 本タスク例では、周期が52ms、位相差が10msのパルスを出します。</p>  <p style="text-align: center;">図1 任意の位相差のパルス出力</p>		
使用機能説明		

- (1) 本タスク例では、タイマXアウトプットコンペア機能を使用して、FTOA、FTOB出力端子より任意の位相差のパルスを出します。
- (a) 図2にタイマXアウトプットコンペア機能のブロック図を示します。以下にタイマXアウトプットコンペア機能のブロック図について説明します。
- ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - ・プリスケラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - ・フリーランニングカウンタ (FRC) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは、システムクロックの2分周、8分周、32分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、FRCの入力クロックにシステムクロックの2分周のクロックを選択しています。
 - ・タイマコントロールレジスタX (TCRX) は、8ビットのリード/ライト可能なレジスタで、FRCの入力クロックの選択を行ないます。
 - ・タイマコントロール/ステータスレジスタX (TCSRX) は、8ビットのレジスタで、カウンタクリアの選択、各割込み要求信号の制御を行ないます。
 - ・タイマインタラプトイネーブルレジスタ (TIER) は、8ビットのリード/ライト可能なレジスタで、各割込み要求の許可/禁止を制御します。
 - ・タイマアウトプットコンペアコントロールレジスタ (TCOR) は、8ビットのリード/ライト可能なレジスタで、アウトプットコンペア出力レベルの選択、アウトプットコンペア出力の許可、およびOCRA、OCRBのアクセスの切替え制御を行ないます。
 - ・アウトプットコンペアレジスタA (OCRA) は、16ビットのリード/ライト可能なレジスタで、OCRAの内容はFRCと常に比較されており、両者の値が一致すると、TCSRXのOCFAが"1"にセットされます。この時、TIERのOCIAEが"1"ならばCPUに割込みを要求します。また、コンペアマッチAが発生したとき、TOCRのOEAが"1"にセットされていると、TOCRのOLVLAで設定したレベル値がFTOA端子に出力されます。
 - ・アウトプットコンペアレジスタB (OCRB) は、16ビットのリード/ライト可能なレジスタで、OCRBの内容はFRCと常に比較されており、両者の値が一致すると、TCSRXのOCFBが"1"にセットされます。この時、TIERのOCIBEが"1"ならばCPUに割込みを要求します。また、コンペアマッチBが発生したとき、TOCRのOEBが"1"にセットされていると、TOCRのOLVLBで設定したレベル値がFTOB端子に出力されます。
 - ・OCRA、およびOCRBは、同一のアドレスに割り付けられているため、このアドレスをリード/ライトするときには、TCORのOCRSによりどちらのレジスタを選択するかを指定します。
 - ・アウトプットコンペアA出力端子 (FTOA) より、コンペアマッチAによるパルスを出します。
 - ・アウトプットコンペアB出力端子 (FTOB) より、コンペアマッチBによるパルスを出します。

使用機能説明

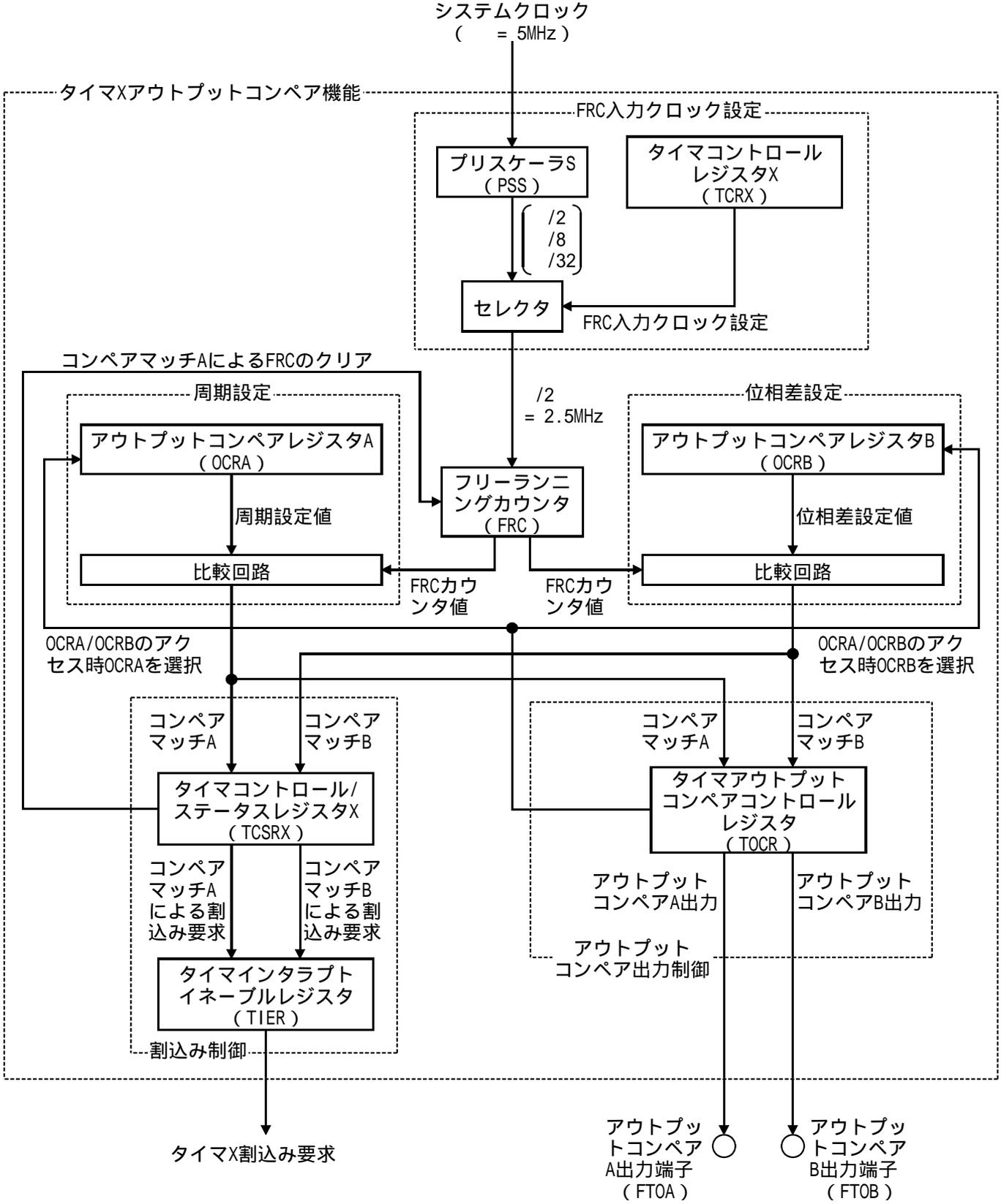
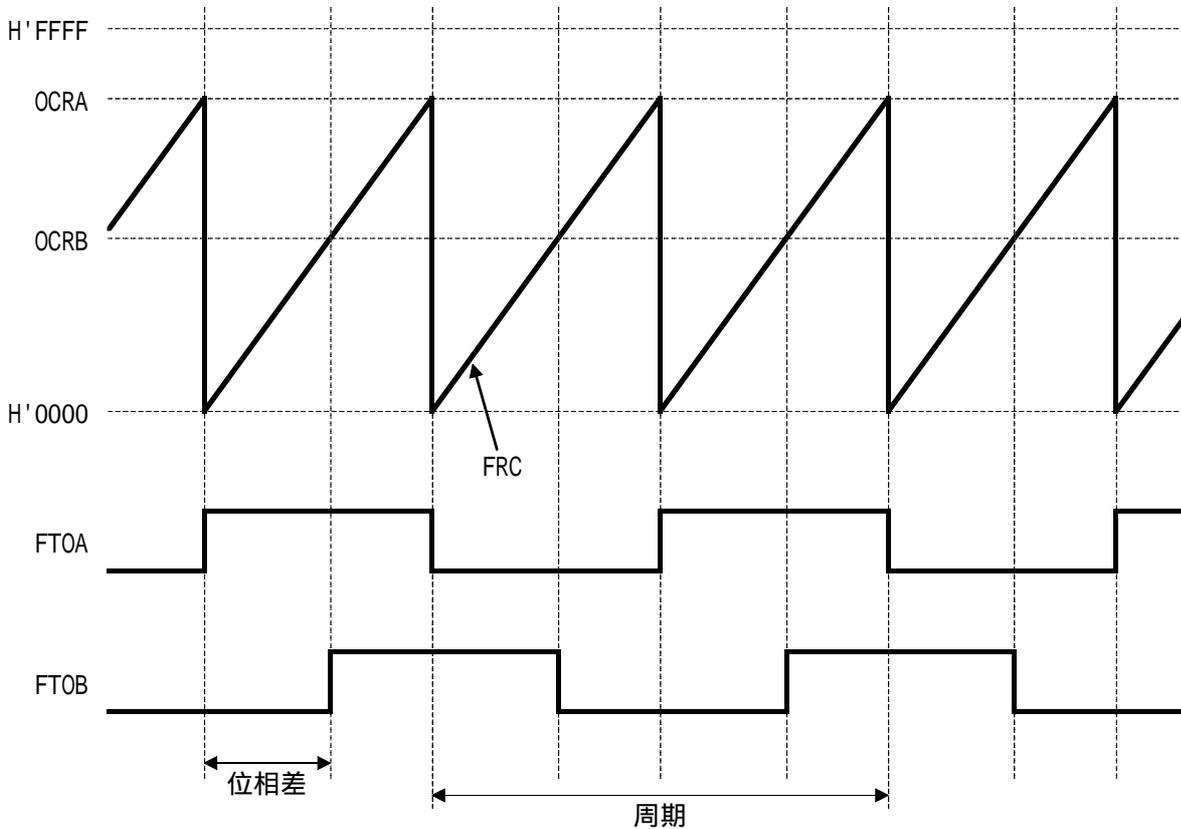


図2 タイマXアウトプットコンペア機能ブロック図

使用機能説明

(b) 図3に出力パルスの周期、および位相差の設定方法について示します。



$$\begin{aligned} \text{周期} &= 2 \times \text{OCRA} \times \text{FRC入力クロック周期} \\ &= 2 \times \text{OCRA} \times 0.4 \mu\text{s} \\ \text{位相差} &= \text{OCRB} \times \text{FRC入力クロック周期} \\ &= \text{OCRB} \times 0.4 \mu\text{s} \end{aligned}$$

図3 出力パルスの周期、および位相差の設定方法

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマXアウトプットコンペア機能による任意の位相差のパルス出力を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TIER	コンペアマッチA、コンペアマッチBによる割込みの許可を設定
TCSRX	コンペアマッチA、コンペアマッチBによる割込み要求信号の制御、コンペアマッチAによるFRCのクリアの許可を設定
FRC	システムクロックを分周したクロックを入力とする16ビットのアップカウンタ
OCRA	出力パルスの周期を設定、FRCのカウンタ値と一致するとコンペアマッチAが発生
OCRB	出力パルスの位相差を設定、FRCのカウンタ値と一致するとコンペアマッチBが発生
TCRX	FRC入力クロックの設定
TOCR	アウトプットコンペア出力レベルの選択、アウトプットコンペア出力の許可、およびOCRA、OCRBのアクセスの切替えを設定
FTOA	コンペアマッチAによるパルスの出力端子
FTOB	コンペアマッチBによるパルスの出力端子

動作原理

(1) 図4に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりタイマXアウトプットコンペア機能による任意の位相差のパルス出力を行ないます。

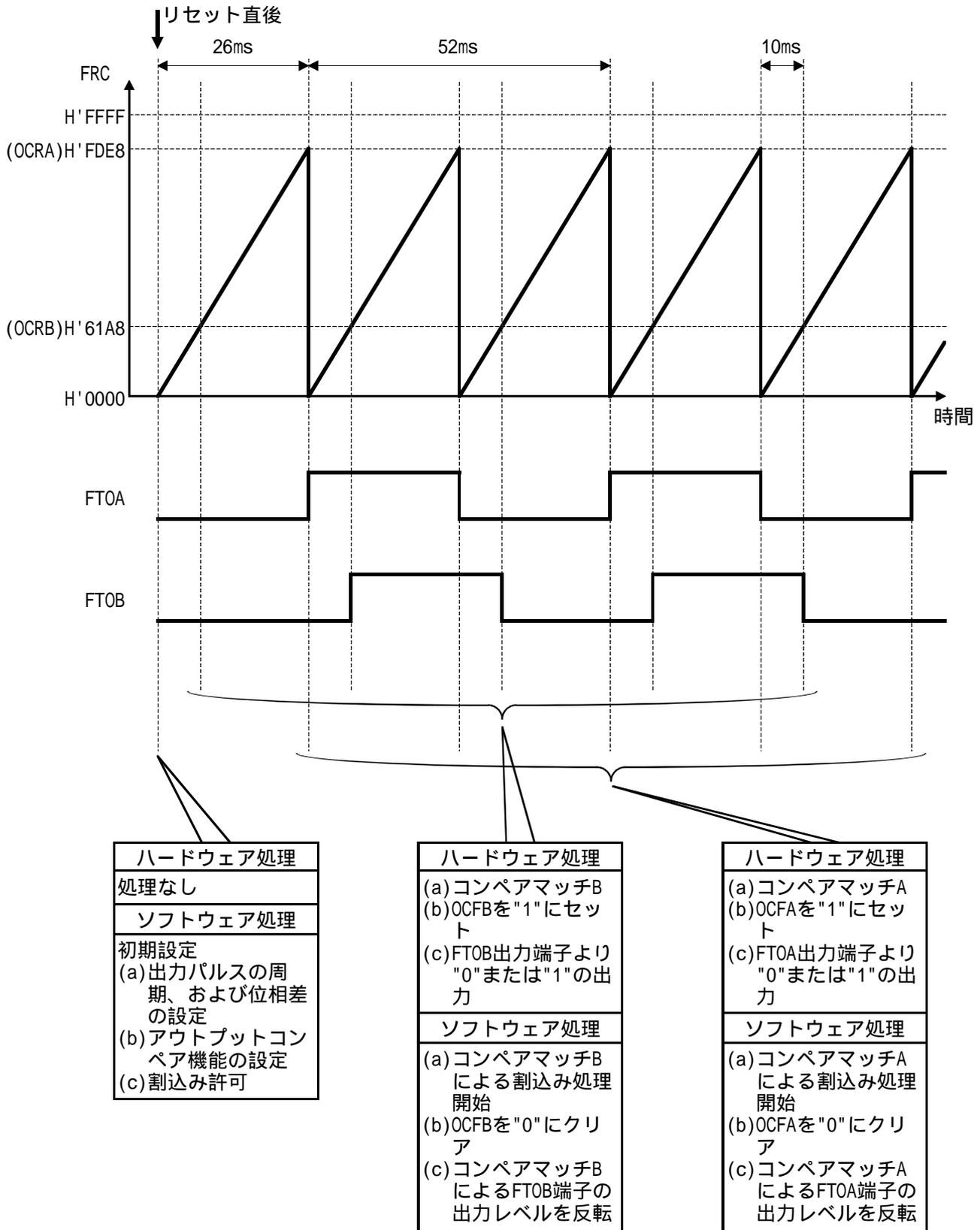


図4 タイマXアウトプットコンペア機能による任意の位相差のパルス出力の動作原理

アウトプットコンペア機能による任意の位相差のパルス出力	使用機能	タイマX：アウトプットコンペア機能
-----------------------------	------	-------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、コンペアマッチ機能の設定、割込みの許可を行なう
アウトプットコンペア出力制御	TXINT	タイマX割込み処理ルーチンで、OCFAによる割込み要求かOCFBによる割込み要求かを判定し、FTOA、またはFTOB端子の出力レベルを反転を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
R1	出力するパルスの周期設定値で、OCRAの設定値	メインルーチン	1ワード	入力
R2	出力するパルスの位相差設定値で、OCRBの設定値	メインルーチン	1ワード	入力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TIER	OCIAE タイマインタラプトイネーブルレジスタ (アウトプットコンペア割込みAイネーブル) : OCIAE="1"のとき、OCFAによる割込みを許可	H'F770 ビット3	1
	OCIBE タイマインタラプトイネーブルレジスタ (アウトプットコンペア割込みBイネーブル) : OCIBE="1"のとき、OCFBによる割込みを許可	H'F770 ビット2	1
TCSRX	OCFA タイマコントロール/ステータスレジスタX (アウトプットコンペアフラグA) : OCFA="0"のとき、FRCとOCRAがコンペアマッチしていないことを示す : OCFA="1"のとき、FRCとOCRAがコンペアマッチしたことを示す	H'F771 ビット3	0
	OCFB タイマコントロール/ステータスレジスタX (アウトプットコンペアフラグB) : OCFB="0"のとき、FRCとOCRBがコンペアマッチしていないことを示す : OCFB="1"のとき、FRCとOCRBがコンペアマッチしたことを示す	H'F771 ビット2	0
	CCLRA タイマコントロール/ステータスレジスタX (カウンタクリアA) : CCLRA="1"のとき、コンペアマッチAによるFRCのクリアを許可	H'F771 ビット0	1
FRCH	フリーランニングカウンタH : システムクロックの2分周のクロックを入力とする16ビットのアップカウンタの上位8ビット	H'F772	H'00
FRCL	フリーランニングカウンタL : システムクロックの2分周のクロックを入力とする16ビットのアップカウンタの下位8ビット	H'F773	H'00

アウトプットコンペア機能による任意の位相差のパルス出力	使用機能	タイマX : アウトプットコンペア機能
-----------------------------	------	---------------------

ソフトウェア説明

表4 使用内部レジスタ説明

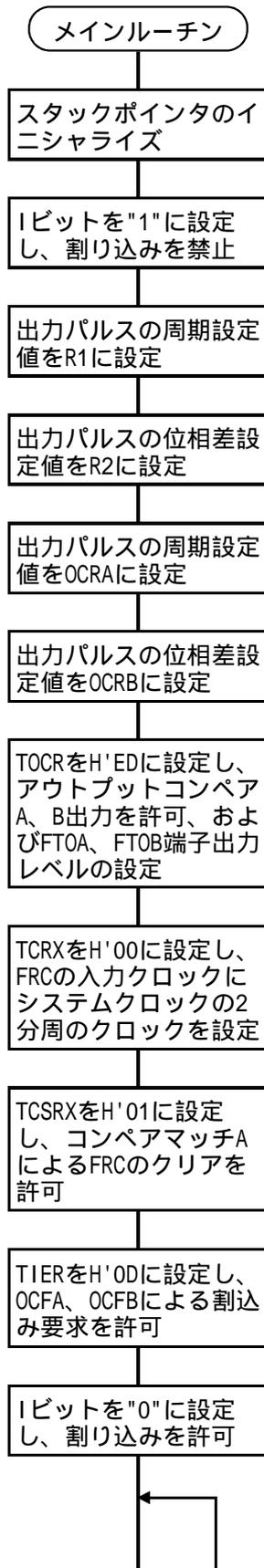
レジスタ名	機能	アドレス	設定値	
OCRAH	アウトプットコンペアレジスタAH : OCRAの上位8ビットで、OCRAの設定値とFRCのカウンタ値が一致すると、コンペアマッチAが発生	H'F774	H'FD	
OCRAL	アウトプットコンペアレジスタAL : OCRAの下位8ビットで、OCRAの設定値とFRCのカウンタ値が一致すると、コンペアマッチAが発生	H'F775	H'E8	
OCRBH	アウトプットコンペアレジスタBH : OCRBの上位8ビットで、OCRBの設定値とFRCのカウンタ値が一致すると、コンペアマッチBが発生	H'F774	H'61	
OCRBL	アウトプットコンペアレジスタBL : OCRBの下位8ビットで、OCRBの設定値とFRCのカウンタ値が一致すると、コンペアマッチBが発生	H'F775	H'A8	
TCRX	CKS1 CKS0 タイマコントロールレジスタX (クロックセレクト1、0) : CKS1="0"、CKS0="0"のとき、FRCの入力クロックをシステムクロックの2分周のクロックに設定	H'F776	CKS1="0" CKS0="0"	
TOCR	OCRS	タイマアウトプットコンペアコントロールレジスタ (アウトプットコンペアレジスタセレクト) : OCRS="0"のとき、OCRA、OCRBのアクセスの切替えをOCRAに設定 : OCRS="1"のとき、OCRA、OCRBのアクセスの切替えをOCRBに設定	H'F777 ビット4	0
	OEA	タイマアウトプットコンペアコントロールレジスタ (アウトプットイネーブルA) : OEA="1"のとき、アウトプットコンペアA出力を許可	H'F777 ビット3	1
	OEB	タイマアウトプットコンペアコントロールレジスタ (アウトプットイネーブルB) : OEB="1"のとき、アウトプットコンペアB出力を許可	H'F777 ビット2	1
	OLVLA	タイマアウトプットコンペアコントロールレジスタ (アウトプットレベルA) : OLVLA="0"のとき、コンペアマッチAにより、FTOA端子に出力するレベルを"Low"レベルに設定 : OLVLA="1"のとき、コンペアマッチAにより、FTOA端子に出力するレベルを"High"に設定	H'F777 ビット1	0
	OLVLB	タイマアウトプットコンペアコントロールレジスタ (アウトプットレベルB) : OLVLB="0"のとき、コンペアマッチBにより、FTOB端子に出力するレベルを"Low"レベルに設定 : OLVLB="1"のとき、コンペアマッチBにより、FTOB端子に出力するレベルを"High"に設定	H'F777 ビット0	1

(4) 使用RAM説明

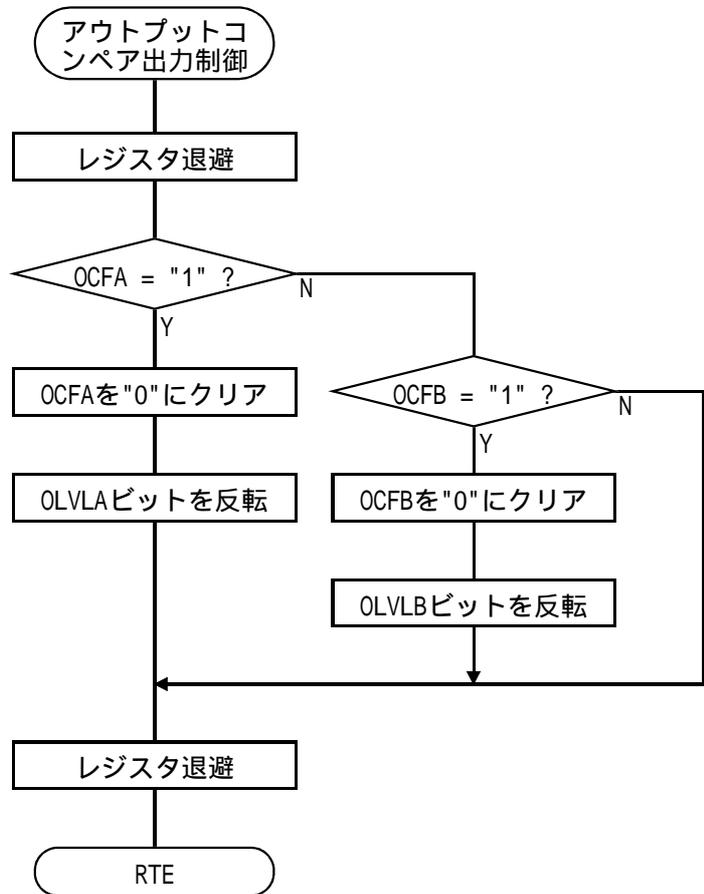
本タスク例では、RAMは使用しません。

フローチャート

(a) メインルーチン



(b) タイマX割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Pulse Output of Random Phase Difference by
;       Output Compare Function'
;
;       Function
;       : Timer X Output Compare
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
TIER          .equ          H'F770          ;Timer Interrupt Enable Register
OCIAE         .bequ        3,TIER          ;Output Compare Interrupt A Enable
OCIBE         .bequ        2,TIER          ;Output Compare Interrupt B Enable
TCSRX        .equ          H'F771          ;Timer Control/Status Register X
OCFA         .bequ        3,TCSRX         ;Output Compare Flag A
OCFB         .bequ        2,TCSRX         ;Output Compare Flag B
CCLRA        .bequ        0,TCSRX         ;Counter Clear A
FRCH         .equ          H'F772          ;Free Running Counter H
FRCL         .equ          H'F773          ;Free Running Counter L
OCRAH        .equ          H'F774          ;Output Compare Register AH
OCRAL        .equ          H'F775          ;Output Compare Register AL
OCRBH        .equ          H'F774          ;Output Compare Register BH
OCRBL        .equ          H'F775          ;Output Compare Register BL
TCRX         .equ          H'F776          ;Timer Control Register X
CKS1         .bequ        1,TCRX          ;Clock Select 1
CKS0         .bequ        0,TCRX          ;Clock Select 0
TOCR         .equ          H'F777          ;Timer Output Compare Control Register
OCRS         .bequ        4,TOCR          ;Output Compare Register Select
OEA          .bequ        3,TOCR          ;Output Enable A
OEB          .bequ        2,TOCR          ;Output Enable B
QLVLA        .bequ        1,TOCR          ;Output Level A
OLVLB        .bequ        0,TOCR          ;Output Level B
;
;
*****
;
;       Ram Allocation
;
*****
;
;       STACK          .equ          H'FF80          ;Stack Pointer
;
;
;       Vector Address
;
*****
;
;       .org          H'0000
;       .data.w      MAIN          ;Reset Interrupt

```

アウトプットコンペア機能による任意の位相差のパルス出力	使用機能	タイマX : アウトプットコンペア機能
プログラムリスト		
;	.org	H'0008
	.data.w	MAIN ;IRQ0 Interrupt
	.data.w	MAIN ;IRQ1 Interrupt
	.data.w	MAIN ;IRQ2 Interrupt
	.data.w	MAIN ;IRQ3 Interrupt
	.data.w	MAIN ;INT0 - INT7 Interrupt
;	.org	H'0014
	.data.w	MAIN ;Timer A Interrupt
	.data.w	MAIN ;Timer B1 Interrupt
;	.org	H'0020
	.data.w	TXINT ;Timer X Interrupt
	.data.w	MAIN ;Timer V Interrupt
;	.org	H'0026
	.data.w	MAIN ;Sci1 Interrupt
;	.org	H'002A
	.data.w	MAIN ;Sci3 Interrupt
	.data.w	MAIN ;A/D Converter Interrupt
	.data.w	MAIN ;Sleep Interrupt
;	*****	
	Main Program	

;	.org	H'1000
;	MAIN	.equ \$
	MOV.W	#STACK,SP ;Initialize Stack Pointer
	ORC	#H'80,CCR ;Interrupt Disable
;	MOV.W	#H'FDE8,R1 ;Set Output Pulse Period
	MOV.W	#H'61A8,R2 ;Set Output Pulse Phase Difference
;	MOV.B	#H'ED,R0L ;Select OCRA
	MOV.B	R0L,@TOCR ;Initialize OCRAH
	MOV.B	R1H,@OCRAH ;Initialize OCRAH
	MOV.B	R1L,@OCRAL ;Initialize OCRAL
;	MOV.B	#H'F0,R0L ;Select OCRB
	MOV.B	R0L,@TOCR ;Initialize OCRBH
	MOV.B	R2H,@OCRBH ;Initialize OCRBH
	MOV.B	R2L,@OCRBL ;Initialize OCRBL
;	MOV.W	#H'ED00,R0 ;Initialize Output Compare Function
	MOV.B	R0H,@TOCR ;Initialize FRC Input Clock Period
	MOV.B	R0L,@TCRX ;Initialize FRC Input Clock Period
;	MOV.W	#H'010D,R0 ;Initialize FRC Clear Enable
	MOV.B	R0H,@TCSRX ;Initialize FRC Clear Enable
	MOV.B	R0L,@TIER ;Initialize OCFA/OCFB Interrupt Enable
;	ANDC	#H'7F,CCR ;Interrupt Enable
;	MAIN9	.equ \$
	BRA	MAIN9
;		

プログラムリスト

```

;*****
;
;           Timer X Interrupt
;*****
;
;
TXINT      .equ          $
           PUSH          R0           ;Store R0
;
           MOV.B         @TCSRX,R0H
           BTST          #3,R0H      ;OCFA = "1" ?
           BEQ           TXINT1      ;No.
;
           BCLR          #3,R0H
           MOV.B         R0H,@TCSRX  ;Clear OCFA
;
           MOV.B         @TOCR,R0L
           BNOT          #1,R0L
           MOV.B         R0L,@TOCR   ;FTOA Terminal Output Level Control
           BRA           TXINT9
;
TXINT1     .equ          $
           BTST          #2,R0H      ;OCFB = "1" ?
           BEQ           TXINT9      ;No.
;
           BCLR          #2,R0H
           MOV.B         R0H,@TCSRX  ;Clear OCFB
;
           MOV.B         @TOCR,R0L
           BNOT          #0,R0L
           MOV.B         R0L,@TOCR   ;FTOB Terminal Output Level Control
;
TXINT9     .equ          $
           POP           R0           ;Restore R0
;
           RTE
;
           .end

```

2.11 インพุットキャプチャ機能によるパルスの周期測定

インพุットキャプチャ機能によるパルスの周期測定	使用機能	タイマX：インพุットキャプチャ機能
<div data-bbox="183 181 245 215" data-label="Section-Header"> <h3>仕様</h3> </div> <div data-bbox="177 262 1426 414" data-label="List-Group"> <ol style="list-style-type: none"> (1) タイマXインพุットキャプチャタイマ機能を使用して、インพุットキャプチャA端子 (FTIA) に入力されたパルスの周期を測定します。 (2) 入力パルスの立ち上がりエッジ間のフリーランニングカウンタ (FRC) のカウンタ値をRAMに格納し、その値より入力パルスの周期を測定します。 (3) 測定可能なパルスの最大の周期は104.856msで、精度は1.6μsです。 </div> <div data-bbox="215 544 1356 768" data-label="Diagram"> </div> <div data-bbox="624 792 967 828" data-label="Caption"> <p>図1 入力パルスの周期測定</p> </div>		
<div data-bbox="132 875 298 907" data-label="Section-Header"> <h3>使用機能説明</h3> </div> <div data-bbox="180 965 1434 1814" data-label="List-Group"> <ol style="list-style-type: none"> (1) 本タスク例では、タイマXインพุットキャプチャ機能を使用して、FTIA端子に入力したパルスの周期の測定を行ないます。 (a) 図2にタイマXインพุットキャプチャ機能のブロック図を示します。以下にタイマXインพุットキャプチャ機能のブロック図について説明します。 <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・プリスケラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・フリーランニングカウンタ (FRC) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックは、システムクロックの2分周、8分周、32分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、FRCの入力クロックにシステムクロックの8分周のクロックを選択しています。 ・タイマコントロールレジスタX (TCRX) は、8ビットのリード/ライト可能なレジスタで、FRCの入力クロックの選択、およびインพุットキャプチャ入力エッジの選択を行ないます。 ・タイマコントロール/ステータスレジスタX (TCSRX) は、8ビットのレジスタで、各割込み要求信号の制御を行ないま。 ・タイマインタラプトイネーブルレジスタ (TIER) は、8ビットのリード/ライト可能なレジスタで、各割込み要求の許可/禁止を制御します。 ・インพุットキャプチャレジスタA (ICRA) は、16ビットのリード専用のレジスタで、インพุットキャプチャ端子Aの入力パルスの立ち上がりエッジが検出されると、そのときのFRCの値がICRAに転送され、TCSRXのICFAが"1"にセットされます。このとき、TIERのICFAが"1"ならばCPUに割込みを要求します。 ・インพุットキャプチャA端子より、周波数測定対象パルスを入力します。 ・以下に本タスク例における入力パルスの周期の計算方法を示します。ただし、FRCがオーバーフローすると、周期測定が正確に行なえないので、入力パルスの周期はFRCのオーバーフロー周期 (104.8576ms) より小さくしなければなりません。 ・1番目の入力パルスの立ち上がりエッジをキャプチャした後、FRCがオーバーフローすると、PDRH、PDRLにH'FFを格納して終了します。 </div> <div data-bbox="220 1861 1372 1951" data-label="Equation-Block"> $\begin{aligned} \text{入力パルスの周期} &= (\text{PDRH、PDRLに格納されたFRCのカウンタ値}) \times (\text{FRC入力クロック周期}) \\ &= (\text{PDRH、PDRLに格納されたFRCのカウンタ値}) \times 1.6\mu\text{s} \end{aligned}$ </div>		

使用機能説明

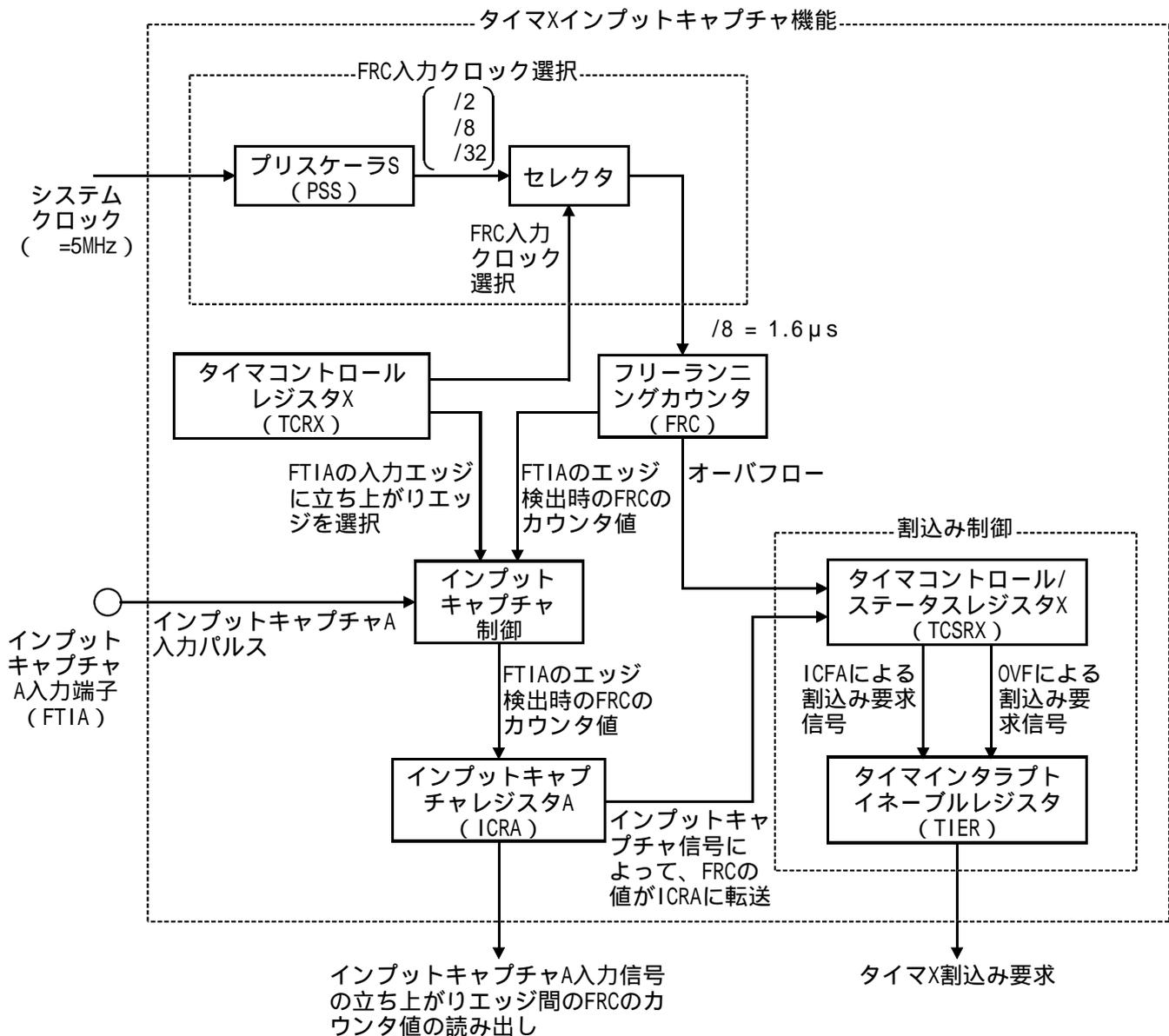


図2 タイマXインプットキャプチャ機能ブロック図

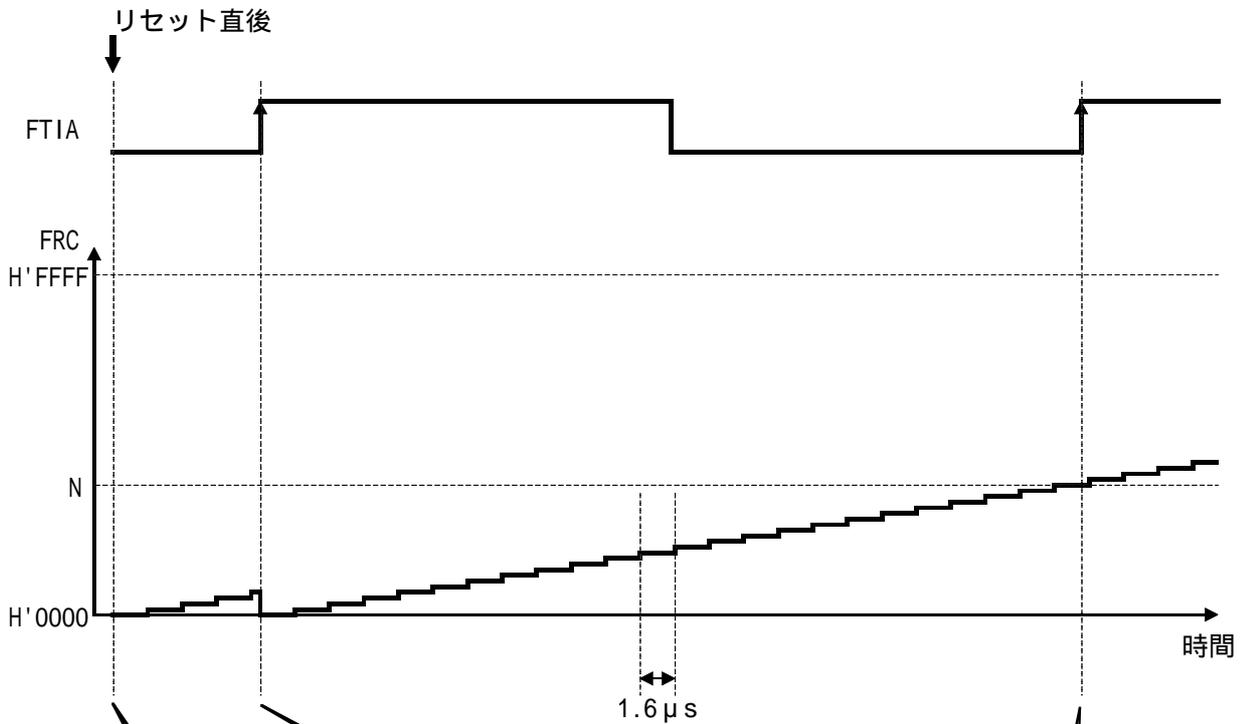
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマXインプットキャプチャ機能による周波数測定を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TIER	インプットキャプチャ信号によるFRCのカウンタ値の転送、およびFRCのオーバーフローによる割り込みの許可
TCSRX	インプットキャプチャ信号によるFRCのカウンタ値の転送、およびFRCのオーバーフローによる割り込みの制御
FRC	システムクロックを8分周したクロックを入力とする16ビットのアップカウンタ
TCRX	FRC入力クロックをシステムクロックの8分周のクロックに設定、およびFTIA入力エッジを立ち上がりエッジ設定
ICRA	FTIA入力パルスの立ち上がりエッジの検出により、その時のFRCのカウンタ値を格納

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりタイマXインプットキャプチャ機能によるパルスの周期測定を行ないます。



【注】N：FTIA入力パルスの立ち上がりエッジ間のFRCのカウンタ値

ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a)インプットキャプチャ機能の設定
(b)割込み許可

ハードウェア処理
(a)FRCのカウンタ値をICRAに転送
(b)ICFAを"1"にセット
ソフトウェア処理
(a)インプットキャプチャ入力信号の立ち上がりエッジの検出による割込み処理の開始
(b)ICFAを"0"にクリア
(c)FRCをクリア
(d)SRTFを"1"にセット

ハードウェア処理
(a)FRCのカウンタ値をICRAに転送
(b)ICFAを"1"にセット
ソフトウェア処理
(a)インプットキャプチャ入力信号の立ち上がりエッジの検出による割込み処理の開始
(b)ICFAを"0"にクリア
(c)ICRAの値をRAMに転送
(d)ENDFを"1"にセット
(d)タイマX割込み要求の禁止

図4 タイマXインプットキャプチャ機能によるパルスの周期測定の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、インプットキャプチャ機能の設定、割込みの許可を行なう
周期測定終了	TXINT	タイマX割込み処理ルーチンで、1回目のICFAによる割込みであればFRCをH'0000にイニシャライズし、2回目のICFAによる割込みであればICRAの内容をRAMに格納し、タイマX割込み要求の禁止を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
PRDH	FTIA入力パルスの立ち上がりエッジ間のFRCのカウント値の上位8ビット	メインルーチン	1バイト	出力
PRDL	FTIA入力パルスの立ち上がりエッジ間のFRCのカウント値の低位8ビット	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TIER	ICIAE タイマインタラプトイネーブルレジスタ (インプットキャプチャ割込みAイネーブル) : ICIAE="1"のとき、IFCAによる割込み要求を許可	H'F770 ビット7	1
	OVIE タイマインタラプトイネーブルレジスタ (タイマオーバーフロー割込みイネーブル) : OVIE="0"のとき、OVFによる割込み要求を禁止 : OVIE="1"のとき、OCFによる割込み要求を許可	H'F770 ビット1	0
TCSRX	ICFA タイマコントロール/ステータスレジスタX (インプットキャプチャフラグA) : ICFA="1"のとき、インプットキャプチャ信号によって、FRCの値がICRAに転送されたことを示す : ICFA="0"のとき、インプットキャプチャ信号が入力されていないことを示す	H'F771 ビット7	0
	OVF タイマコントロール/ステータスレジスタX (タイマオーバーフロー) : OVF="0"のとき、FRCがオーバーフローしていないことを示す : OVF="1"のとき、FRCがオーバーフローしたことを示す	H'F771 ビット1	0
FRCH	フリーランニングカウンタH : システムクロックの8分周のクロックを入力とする16ビットのアップカウンタの上位8ビット	H'F772	H'00
FRCL	フリーランニングカウンタL : システムクロックの8分周のクロックを入力とする16ビットのアップカウンタの低位8ビット	H'F773	H'00
TCRX	IEDGA タイマコントロールレジスタX (インプットエッジセレクトA) : IEDGA="1"のとき、インプットキャプチャ入力Aの立ち上がりエッジでキャプチャ	H'F776 ビット7	1

ソフトウェア説明

表4 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
TCRX	CKS1	タイマコントロールレジスタX (クロックセレクト) : CKS1="0"、CKS="1"のとき、FRCの入力クロックをシステムクロックの8分周のクロックに設定	H'F776 ビット1 ビット0	CKS1="0" CKS0="1"
	CKS0			
ICRAH		インプットキャプチャレジスタAH : インプットキャプチャ入力Aの入力信号の立ち上がりエッジ検出により、FRCのカウンタ値の上位8ビットを格納	H'F778	H'00
ICRAL		インプットキャプチャレジスタAL : インプットキャプチャ入力Aの入力信号の立ち上がりエッジ検出により、FRCのカウンタ値の下位8ビットを格納	H'F779	H'00

(4) 使用RAM説明

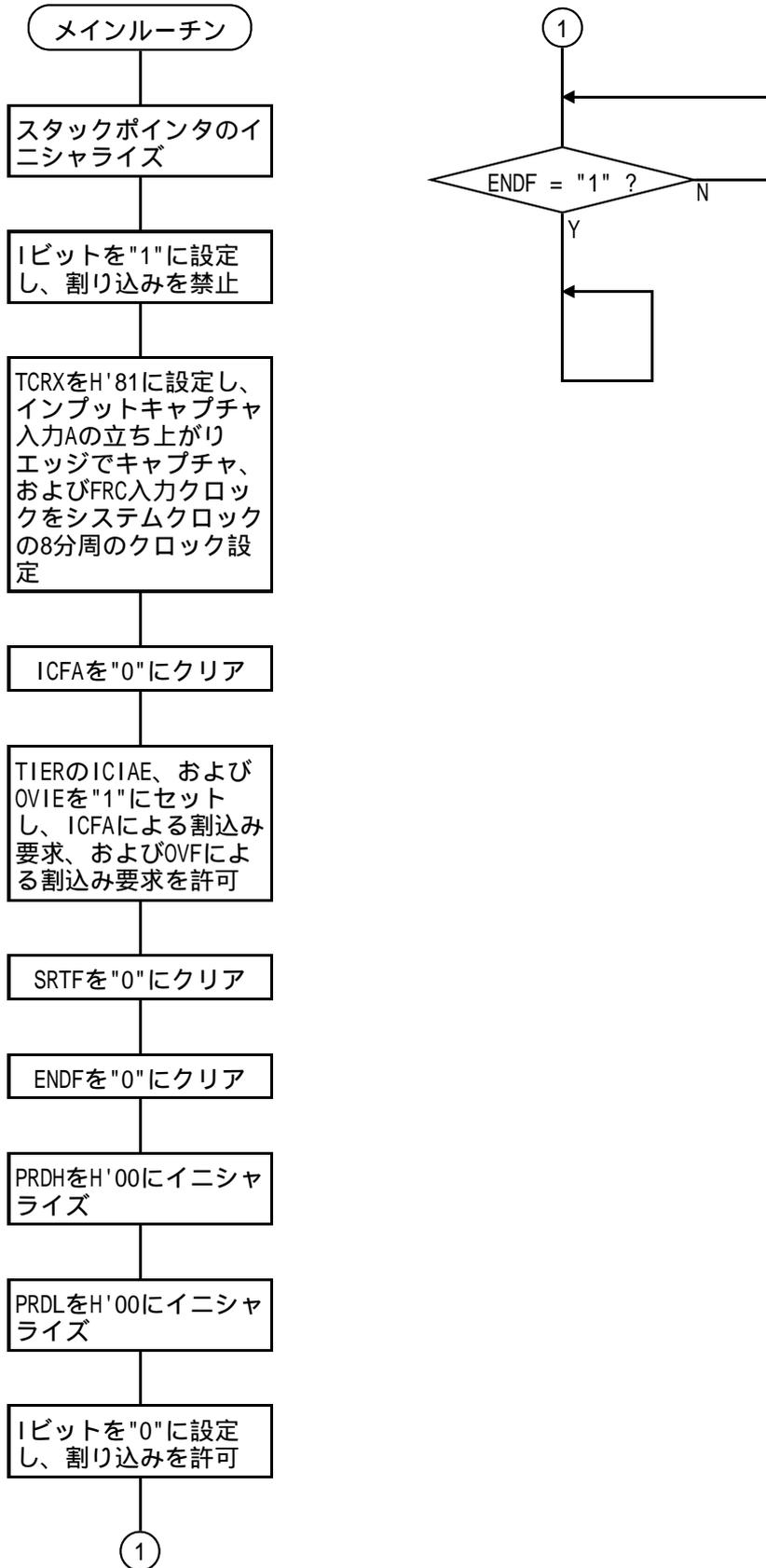
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名		機能	アドレス	使用モジュール名
USRF	ENDF	周期測定が終了したか否かを判定するフラグ	H'FB80 ビット0	メインルーチン 周期測定終了
	STRF	2回目のタイマX割込みか否かを判定するフラグ	H'FB80 ビット1	周期測定終了
PRDH		FTIA入力端子の立ち上がりエッジ間のFRCのカウンタ値の上位8ビットを格納	H'FB81	周期測定終了
PRDL		FTIA入力端子の立ち上がりエッジ間のFRCのカウンタ値の下位8ビットを格納	H'FB82	周期測定終了

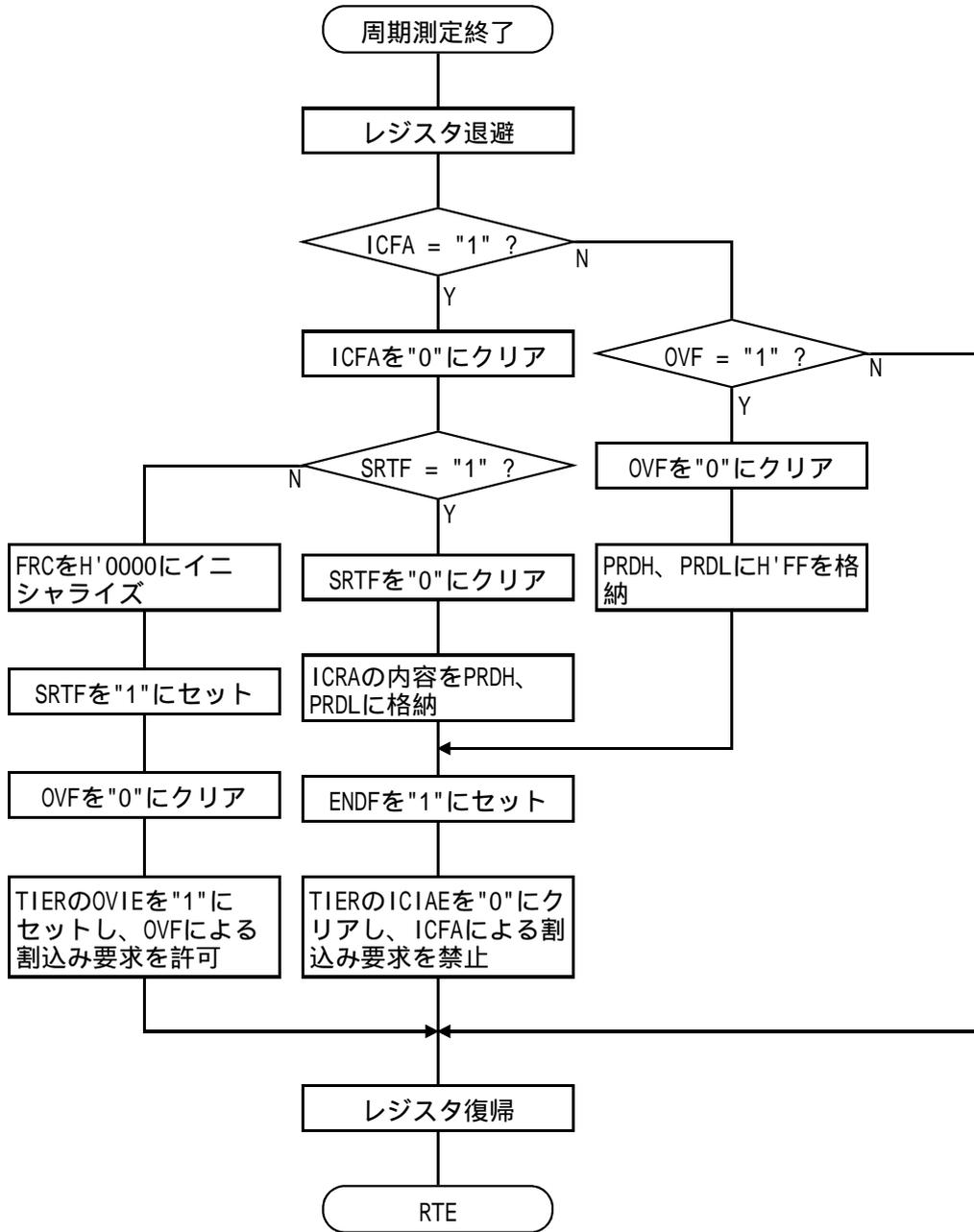
フローチャート

(a) メインルーチン



フローチャート

(b) タイマX割込み処理ルーチン



プログラムリスト

```

*****
:
:           H8/300L Series -H8/3644,H8/3657-
:           Application Note
:
:           'Pulse Period Measurement by Input
:           Caputure Function'
:
:           Function
:           : Timer X Input Capture
:
:           External Clock : 10MHz
:           Internal Clock : 5MHz
:           Sub Clock      : 32.768kHz
:
*****
:
*****
:
:           .cpu          300L
:
*****
:
:           Symbol Dfnition
:
*****
TIER          .equ          H'F770          ;Timer Interrupt Enable Register
ICIAE         .bequ         7,TIER          ;Input Capture Interrupt A Enable
OVIE          .bequ         1,TIER          ;Timer Overflow Interrupt Enable
TCSRX         .equ          H'F771          ;Timer Control / Status Register X
ICFA          .bequ         7,TCSRX         ;Input Capture Flag A
OVF           .bequ         1,TCSRX         ;Timer Overflow
FRCH          .equ          H'F772          ;Free Running Counter H
FRCL          .equ          H'F773          ;Free Running Counter L
TCRX          .equ          H'F776          ;Timer Control Register X
IEDGA         .bequ         7,TCRX         ;Input Edge Select A
CKS1          .bequ         1,TCRX         ;Clock Select 1
CKS0          .bequ         0,TCRX         ;Clock Select 0
ICRAH         .equ          H'F778          ;Input Capture Register AH
ICRAL         .equ          H'F779          ;Input Capture Register AL
:
*****
:
:           Ram Allocation
:
*****
STACK         .equ          H'FF80          ;Stack Pointer
USRF          .equ          H'FB80          ;User Flag Erea
ENDF          .bequ         0,USRF         ;End Flag
SRTF          .bequ         1,USRF         ;Start Flag
PRDH          .equ          H'FB81          ;Period Higher
PRDL          .equ          H'FB82          ;Period Lower
:
*****
:
:           Vector Address
:
*****
:
:           .org          H'0000
:           .data.w      MAIN              ;Reset Interrupt
:
:           .org          H'0008
:           .data.w      MAIN              ;IRQ0 Interrupt

```

プログラムリスト

```

        .data.w      MAIN      ;IRQ1 Interrupt
        .data.w      MAIN      ;IRQ2 Interrupt
        .data.w      MAIN      ;IRQ3 Interrupt
        .data.w      MAIN      ;INT0 - INT7 Interrupt
;
        .org         H'0014
        .data.w      MAIN      ;Timer A Interrupt
        .data.w      MAIN      ;Timer B1 Interrup
;
        .org         H'0020
        .data.w      TXINT     ;Timer X Interrupt
        .data.w      MAIN      ;Timer V Interrupt
;
        .org         H'0026
        .data.w      MAIN      ;Sci1 Interrupt
;
        .org         H'002A
        .data.w      MAIN      ;Sci3 Interrupt
        .data.w      MAIN      ;A/D Converter Interrupt
        .data.w      MAIN      ;Sleep Interrupt
;
;*****
;
; Main Program
;*****
;
        .org         H'1000
;
MAIN     .equ         $
        MOV.W        #STACK,SP      ;Initialize Stack Pointer
        ORC          #H'80,CCR      ;Interrupt Disable
;
        MOV.B        #H'81,R0L
        MOV.B        R0L,@TCRX      ;Initialize FTIA Input Edge & Input Clock Period
;
        MOV.B        @TCSRX,R0L
        MOV.B        #H'00,R0L
        MOV.B        R0L,@TCSRX     ;Clear ICFA & OVF
;
        MOV.B        #H'81,R0L
        MOV.B        R0L,@TIER      ;ICFA Interrupt Enable
;
        MOV.B        #H'00,R0L
        MOV.B        R0L,@USRF      ;Clear SRTF & ENDF
        MOV.B        R0L,@PRDH      ;Initialize PRDH
        MOV.B        R0L,@PRDL      ;Initialize PRDL
;
        ANDC         #H'7F,CCR      ;Interrupt Enable
;
MAIN1    .equ         $
        MOV.B        @USRF,R0L
        BTST         #0,R0L         ;ENDF = "1" ?
        BEQ          MAIN1         ;No.
;
MAIN9    .equ         $
        BRA          MAIN9
;
;*****
;
; Timer X Interrupt
;*****
;

```

プログラムリスト

```

TXINT          .equ          $
                PUSH          R0          ;Store R0
;
                MOV.B         @TCSRX,ROH
                BTST          #7,ROH     ;ICFA = "1" ?
                BEQ           TXINT3     ;No.
;
                BCLR          #7,ROH
                MOV.B         ROH,@TCSRX ;Clear ICFA
;
                MOV.B         @USRF,ROL
                BTST          #1,ROL     ;SRTF = "1" ?
                BEQ           TXINT2     ;No.
;
                BCLR          #1,ROL
                MOV.B         ROL,@USRF  ;Clear STRF
;
                MOV.B         @ICRAH,ROH ;Load ICRAH
                MOV.B         @ICRAL,ROL ;Load ICRAL
;
                MOV.B         ROH,@PRDH  ;Store ICRAH
                MOV.B         ROL,@PRDL  ;Store ICRAL
;
TXINT1         MOV.B         @USRF,ROL
                BSET          #0,ROL
                MOV.B         ROL,@USRF  ;Set ENDF
                MOV.B         #H'01,ROL
                MOV.B         ROL,@TIER  ;ICFA Interrupt Disable
                BRA           TXINT9
;
TXINT2         .equ          $
                MOV.B         #H'00,ROL
                MOV.B         ROL,@FRCH  ;Initialize FRCH
                MOV.B         ROL,@FRCL  ;Initialize FRCL
;
                MOV.B         @TCSRX,ROL
                MOV.B         #H'00,ROL
                MOV.B         ROL,@TCSRX ;Clear OVF
                MOV.B         #H'83,ROL
                MOV.B         ROL,@TIER  ;OVF Interrupt Enable
;
                MOV.B         @USRF,ROL
                BSET          #1,ROL
                MOV.B         ROL,@USRF  ;Set SRTF
                BRA           TXINT9
;
TXINT3         BTST          #1,ROH     ;OVF = 1 ?
                BEQ           TXINT9     ;No.
;
                BCLR          #1,ROH     ;Clear OVF
                MOV.B         ROH,@TCSRX
                MOV.B         #H'FF,ROL
                MOV.B         ROL,@PRDH
                MOV.B         ROL,@PRDL  ;FRC Overflow Process
                BRA           TXINT1
;
TXINT9         .equ          $
                POP           R0          ;Restore R0
                RTE
;
                .end
    
```

2.12 ウォッチドッグタイマ

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
仕様	<p>(1) ウォッチドッグタイマ機能を使用して、ウォッチドッグ動作を行ないます。</p> <p>(2) タイマカウンタWのオーバフロー周期を120msに設定し、120ms以内にタイマカウンタWをイニシャライズしなければ、内部リセット信号が発生するように動作させます。</p> <p>(3) 通常動作では、一定時間ごとにLEDが点灯/消灯を繰り返し、タイマカウンタWがオーバフローする前にタイマカウンタWをイニシャライズするように設定します。</p> <p>(4) $\overline{\text{INT}}_0$入力端子に接続したスイッチのオンにより、タイマカウンタWがイニシャライズされずにオーバフローし、内部リセット信号が発生するように動作させます。</p> <p>(5) 図1に、$\overline{\text{INT}}_0$入力端子のスイッチ接続例を示します。</p> <div data-bbox="518 548 1077 784" data-label="Diagram"> </div> <p style="text-align: center;">図1 $\overline{\text{INT}}_0$入力端子のスイッチ接続例</p>	

使用機能説明
<p>(1) 本タスク例では、ウォッチドッグタイマ機能を使用して、ウォッチドッグ動作を行ないます。</p> <p>(a) 図2にウォッチドッグタイマ機能のブロック図を示します。以下にウォッチドッグタイマ機能のブロック図について説明します。</p> <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・プリスケラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。 ・タイマカウンタW (TCW) は、8ビットのリード/ライト可能なアップカウンタで、入力する内部クロックによりカウントアップされます。入力するクロックは、 /8192です。 ・タイマコントロール/ステータスレジスタW (TCSRW) は、8ビットのリード/ライト可能なレジスタで、TCSRW、TCWの書き込み制御、ウォッチドッグタイマの動作制御、動作状態を示します。 ・以下に、本タスク例におけるTCWのオーバフロー周期の計算法を示します。 $\begin{aligned} \text{TCWオーバフロー周期} &= \frac{1}{\text{システムクロック} / 8192} \times (256 - (\text{TCWのリロード設定値})) \\ &= 1.6384\text{ms} \times (256 - 182) \\ &= 121.2416\text{ms} \end{aligned}$ <div data-bbox="231 1601 1428 1960" data-label="Diagram"> </div> <p style="text-align: center;">図2 ウォッチドッグタイマ機能のブロック図</p>

ウォッチドッグタイマ	使用機能	ウォッチドッグタイマ機能
------------	------	--------------

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、ウォッチドッグタイマ機能によるウォッチドッグ動作を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCW	システムクロックの8192分周のクロックを入力とする8ビットのカウンタ
TCSRW	TCSRW、TCWの書き込み制御、ウォッチドッグタイマの動作制御、動作状態を示す
\overline{INT}_0	SW入力の入力端子

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりウォッチドッグタイマ機能によるウォッチドッグ動作を行ないます。

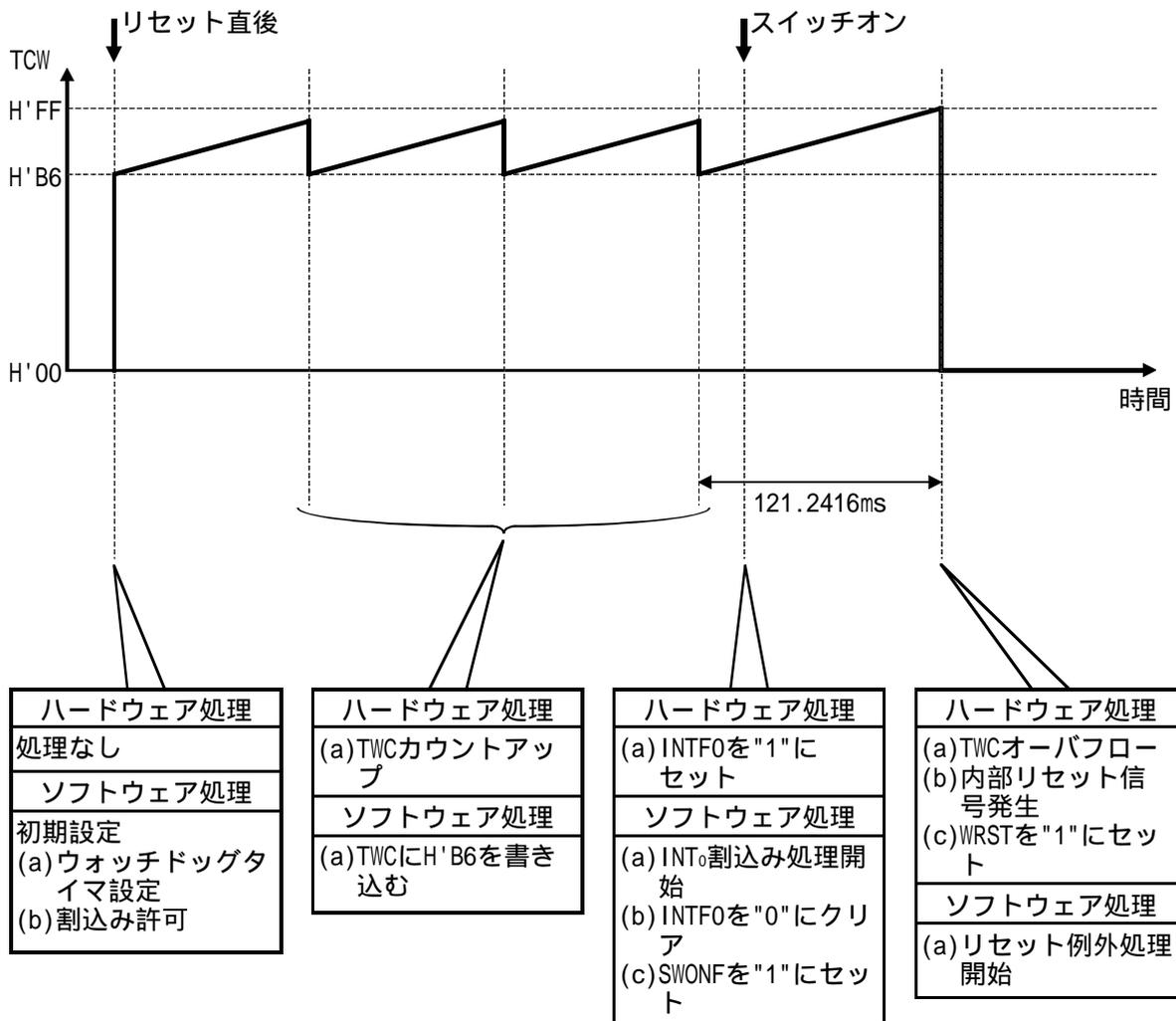


図3 ウォッチドッグタイマ機能によるウォッチドッグ動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	WDTMN	スタックポインタのイニシャライズ、ウォッチドッグタイマ機能の設定、割込みの許可、LEDの制御、およびINT ₀ に接続したSWのONの判定を行なう
スイッチオン	IOINT	INT ₀ 割込み処理ルーチンで、SWONFを"1"にセット

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例におけるH8/3644の使用内部レジスタ説明を示します。

表3 H8/3644の使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TCSRW	B6WI タイマコントロール/ステータスレジスタW (ビット6書き込み禁止) : B6WI="0"のとき、TCSRWビット6への書き込みを許可 : B6WI="1"のとき、TCSRWビット6への書き込みを禁止	H' FFBE ビット7	1
	TCWE タイマコントロール/ステータスレジスタW (タイマカウンタW書き込み許可) : TWCE="1"のとき、TWCへの8ビットデータの書き込みを許可	H' FFBE ビット6	1
	B4WI タイマコントロール/ステータスレジスタW (ビット4書き込み禁止) : B4WI="0"のとき、TCSRWビット4への書き込みを許可 : B4WI="1"のとき、TCSRWビット4への書き込みを禁止	H' FFBE ビット5	1
	TCSRWE タイマコントロール/ステータスレジスタW (タイマコントロール/ステータスレジスタW書き込み許可) : TCSRWE="1"のとき、TCSRWビット2およびビット0への書き込みを許可	H' FFBE ビット4	1
	B2WI タイマコントロール/ステータスレジスタW (ビット2書き込み禁止) : B2WI="0"のとき、TCSRWビット2への書き込みを許可 : B2WI="1"のとき、TCSRWビット2への書き込みを禁止	H' FFBE ビット3	1
	WDON タイマコントロール/ステータスレジスタW (ウォッチドッグタイマオン) : WDON="1"のとき、ウォッチドッグタイマの動作を許可	H' FFBE ビット2	1
	BOWI タイマコントロール/ステータスレジスタW (ビット0書き込み禁止) : BOWI="0"のとき、TCSRWビット0への書き込みを許可 : BOWI="1"のとき、TCSRWビット0への書き込みを禁止	H' FFBE ビット1	1
	WRST タイマコントロール/ステータスレジスタW (ウォッチドッグタイマリセット) : WRST="0"のとき、TCWがオーバフローし内部リセット信号が発生していないことを示す : WRST="1"のとき、TCWがオーバフローし内部リセット信号が発生したことを示す	H' FFBE ビット0	0
TCW	タイマカウンタW : システムクロックの8192分周のクロックを入力とする8ビットのカウンタ	H' FFBF	H' B6

ソフトウェア説明

表3 H8/3644の使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
PDR7	P73	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7	PCR73	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ ="0"のとき、P7 ₃ 端子をP7 ₃ 入力端子機能に設定 : PCR7 ₃ ="1"のとき、P7 ₃ 端子をP7 ₃ 出力端子機能に設定	H'FFEA ビット3	1
IEGR2	INTEG0	割込み許可エッジセレクトレジスタ2 (INT ₀ エッジセレクト) : INTEG0="0"のとき、INT ₀ 端子入力の検出エッジに立ち下がり エッジを選択	H'FFF3 ビット0	0
IENR3	INTEN0	割込み許可レジスタ3 (INT ₀ 割込みイネーブル) : INTEN0="1"のとき、INT ₀ 端子の割込み要求を許可	H'FFF6 ビット0	1
IRR3	INTF0	割込み要求レジスタ3 (INT ₀ 割込み要求フラグ) : INTF0="0"のとき、INT ₀ 端子の割込みが要求されていない : INTF0="1"のとき、INT ₀ 端子の割込みが要求されている	H'FFF9 ビット0	0

(4) 使用RAM説明

表4に本タスク例で使用するRAMの説明を示します。

表4 使用RAM説明

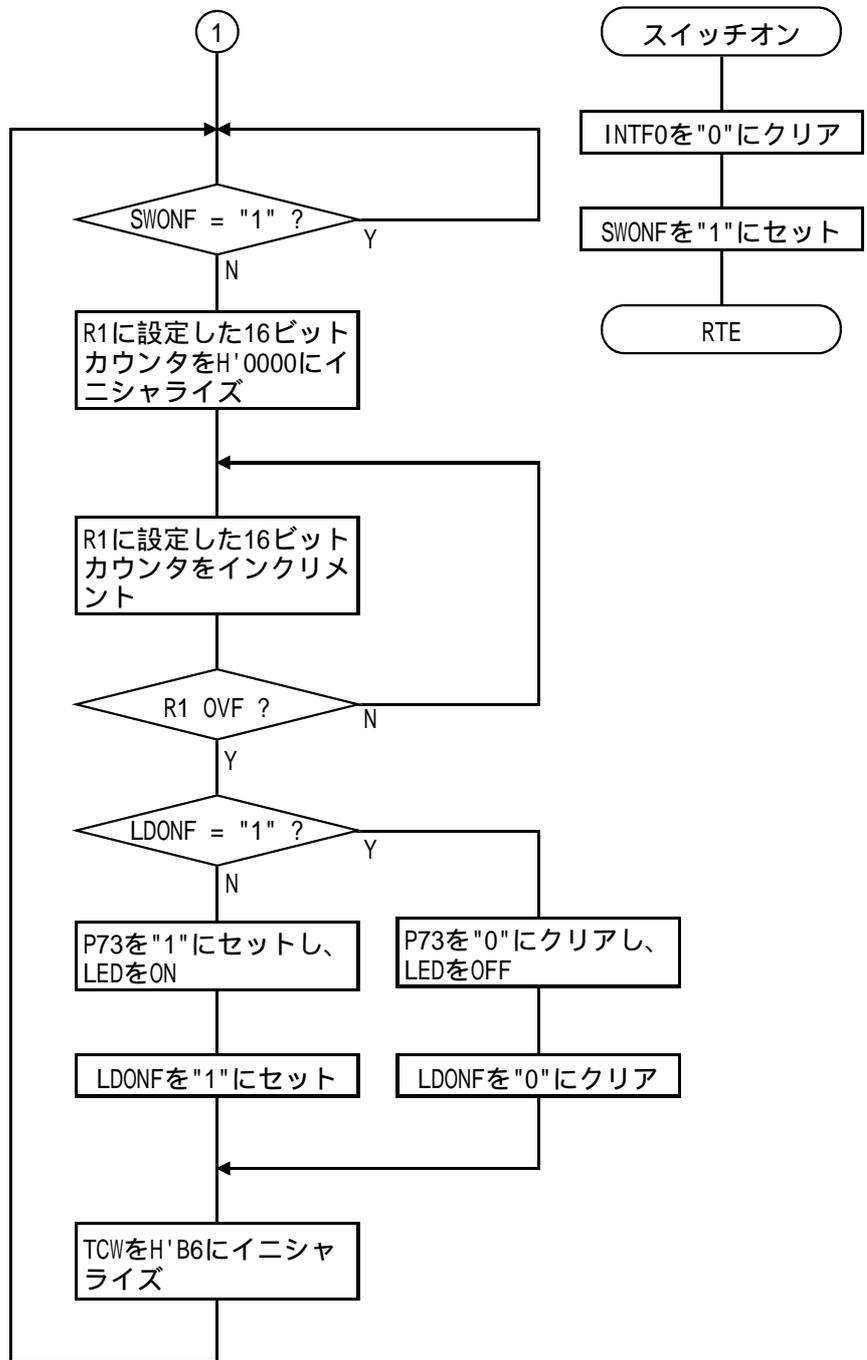
ラベル名	機能	アドレス	使用モジュール名
USRF	SWONF	スイッチ入力のON/OFFを判定するフラグ	H'FB80 ビット0 メインルーチン スイッチオン
	LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット1 メインルーチン

フローチャート

(a) メインルーチン



(b) INT₀割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Watchdog Timer'
;
;       Function
;       : Watchdog Timer
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
;
TCSRW      .equ          H'FFBE          ;Timer Control/Status Register W
B6WI       .bequ        7,TCSRW         ;Bit-6 Write Disable
TCWE       .bequ        6,TCSRW         ;Timer Counter W Write Enable
B4WI       .bequ        5,TCSRW         ;Bit-4 Write Disable
TCSRWE     .bequ        4,TCSRW         ;Timer Control/Status Register W Write Enable
B2WI       .bequ        3,TCSRW         ;Bit-2 Write Disable
WDON       .bequ        2,TCSRW         ;Watchdog Timer ON
BOWI       .bequ        1,TCSRW         ;Bit-0 Write Disable
WRST       .bequ        0,TCSRW         ;Watchdog Timer Reset
TCW        .equ          H'FFBF          ;Timer Counter W
PDR7       .equ          H'FFDA          ;Port Data Register 7
P73        .bequ        3,PDR7          ;Port Data Register 73
PCR7       .equ          H'FFEA          ;Port Control Register 7
PCR73      .bequ        3,PCR7          ;Port Control Register 73
IEGR2      .equ          H'FFF3          ;Interrupt Edge Select Register 2
INTEGO     .bequ        0,IEGR2         ;INT0 Edge Select
IENR3      .equ          H'FFF6          ;Interrupt Enable Register 3
INTENO     .bequ        0,IENR3         ;INT0 Interrupt Enable
IRR3       .equ          H'FFF9          ;Interrupt Request Register 3
INTFO      .bequ        0,IRR3          ;INT0 Interrupt Request Register
;
;
*****
;
;       Ram Allocation
;
*****
;
;
STACK      .equ          H'FF80          ;Stack Pointer
USRF       .equ          H'FFB0          ;User Flag Area
SWONF      .bequ        0,USRF          ;Switch On Flag
LDONF      .bequ        1,USRF          ;LED On Flag
;
;
*****
;
;       Vector Address
;
*****
;
;
.org        H'0000
.data.w    MAIN          ;Reset Interrupt

```

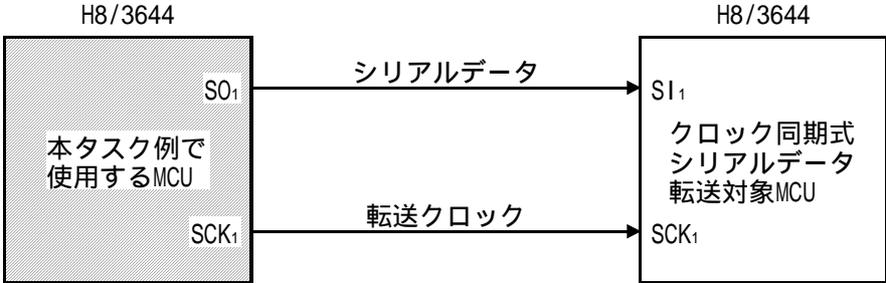

プログラムリスト

```

;
MAIN1      .equ          $
           MOV.B        @USRF,ROL
           BTST        #0,ROL      ;SWONF = 1 ?
           BNE         MAIN1      ;Yes.
;
           SUB.W        R1,R1      ;Initialize 16bit Counter
;
MAIN2      .equ          $
           ADDS        #1,R1      ;Increment 16bit Counter
           MOV.W        R1,R1      ;16bit Counter = H'0000 ?
           BNE         MAIN2      ;No.
;
           MOV.B        @USRF,ROL
           BTST        #1,ROL      ;LDONF = 1 ?
           BNE         MAIN3      ;Yes.
;
           BSET        P73        ;Turn On LED
           BSET        #1,ROL
           MOV.B        ROL,@USRF  ;Set LDONF
           BRA         MAIN4
;
MAIN3      .equ          $
           BCLR        P73        ;Turn Off LED
           BCLR        #1,ROL
           MOV.B        ROL,@USRF  ;Clear LDONF
;
MAIN4      .equ          $
           MOV.B        #H'B6,ROL
           MOV.B        ROL,@TCW  ;Initialize TCW
;
           BRA         MAIN1
;
;*****
;
;          INTO Interrupt
;*****
;
;
IOINT      .equ          $
           BCLR        INTF0      ;Clear INTF0
;
           MOV.B        @USRF,ROL
           BSET        #0,ROL
           MOV.B        ROL,@USRF  ;Set SWONF
;
           RTE
;
           .end

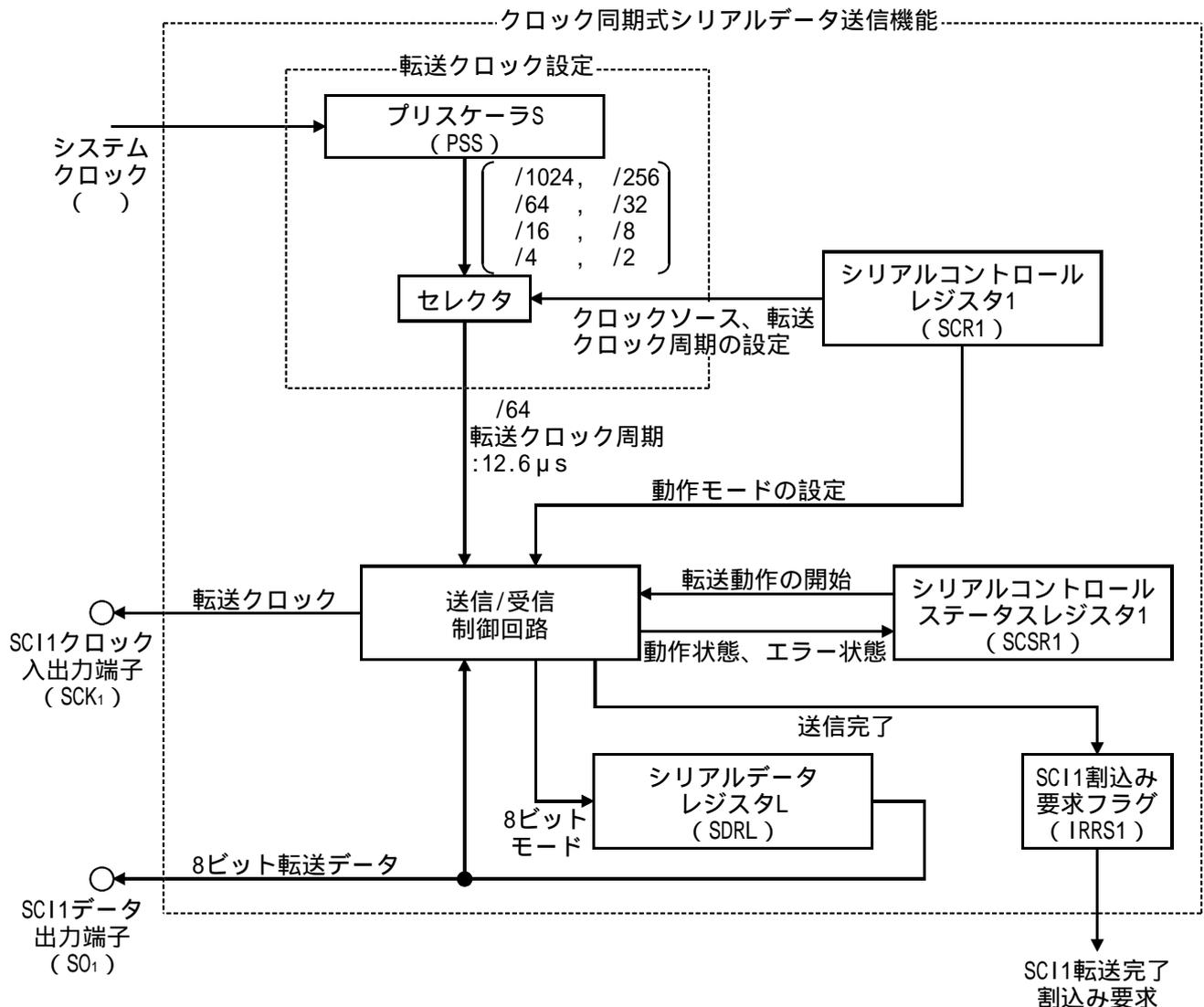
```

2.13 クロック同期式シリアルデータ送信

クロック同期式シリアルデータ送信	使用機能	SCI1 : クロック同期式シリアル転送機能
仕様		
<p>(1) 図1に示すようにクロック同期式シリアル転送機能を使用して、4バイトの8ビットデータを送信します。</p> <p>(2) 転送クロックは、内部クロックを使用し12.8 μsの転送クロック周期でデータを転送します。</p> <p>(3) 送信するデータのデータ長は8ビットで、データの最下位ビットから送信するLSBファースト方式による送信を行ないます。</p> <div style="text-align: center;">  </div> <p>図1 クロック同期式シリアルデータ送信</p>		
使用機能説明		

- (1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、クロック同期式のシリアルデータの送信を行ないます。図2にクロック同期式シリアルデータ送信のブロック図を示します。以下にクロック同期式シリアルデータ送信のブロック図について説明します。
- ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - ・プリスケアラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - ・シリアルコントロールレジスタ1 (SCR1) は、8ビットのリード/ライト可能なレジスタで、動作モード、転送クロックソースとプリスケアラ分周比を選択します。
 - ・シリアルコントロールステータスレジスタ1 (SCSR1) は、動作状態、エラー状態等を示す8ビットのレジスタです。
 - ・シリアルデータレジスタU (SDRU) は、8ビットのリード/ライト可能なレジスタで、16ビット転送時に上位8ビットのデータレジスタとして使用します。SDRUに書き込まれたデータは、SDRLにLSBファーストで出力されます。入れ替わりにSI₁端子よりLSBファーストでデータが入力されて、MSB → LSB方向にデータがシフトします。
 - ・シリアルデータレジスタL (SDRL) は、8ビットのリード/ライト可能なレジスタで、8ビット転送時のデータレジスタ、および16ビット転送時の下位8ビットのデータレジスタとして使用します。8ビット転送時、SDRLに書き込まれたデータは、SO₁端子よりLSBファーストで出力されます。入れ替わりにSI₁端子よりLSBファーストで入力されて、MSB → LSB方向にデータがシフトします。16ビット転送時には、入力データがSDRUより取り込まれることを除けば、8ビット転送時と同様の動作となります。
 - ・SDRU、SDRLのリード/ライトは、データの送信/受信が完了してから行なう必要があります。データの送信/受信中にリード/ライトを行なうとデータの内容は保証されません。
 - ・転送クロックは、8種類の内部クロックと外部クロックから選択できます。内部クロックを選択した場合は、SCK₁端子は出力端子となります。クロック連続出力モードに設定すると選択したクロックをSCK₁端子から連続して出力します。外部クロックを選択した場合は、SCK₁端子はクロック入力端子となります。
 - ・本タスク例では、転送クロックソースをPSSに、プリスケアラ分周比を64分周、転送クロック周期を12.8 μ sに設定しています。
 - ・SCI1の転送フォーマットは8ビットおよび16ビットの転送データを選択可能です。データの最下位ビットから送受信されるLSBファースト方式による転送を行ないます。送信データは、転送クロックの立ち上がりから次の立ち上がりまで出力されます。また、受信データは転送クロックの立ち上がりで取り込まれます。
 - ・本タスク例では、動作モードを8ビットモードに設定し、8ビットのデータ送信を行ないます。
 - ・SCI1クロック (SCK₁) は、SCI₁のクロック入出力端子です。
 - ・SCI1データ入力 (SI₁) は、SCI₁の受信データの入力端子です。
 - ・SCI1が転送完了すると、割込み要求レジスタ2 (IRR2) のCSI1割込み要求フラグ (IRRS1) が "1" にセットされます。SCI1の割込み要求は、割込み許可レジスタ2 (IENR2) のSCI1割込みイネーブル (IENS1) により許可/禁止を選択できます。

使用機能説明



(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、クロック同期式シリアルデータ送信を行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
SCR1	動作モード、転送クロックソース、プリスケラ分周比の設定
SCSR1	動作状態、エラー状態を示す
SDRL	8ビットの送信データのデータレジスタ
SCK ₁	SCI1の転送クロック出力端子
SO ₁	SCI1の送信データの出力端子
IRRS1	SCI1の転送完了の有無を示す
IENS1	SCI1割込み要求の許可/禁止を制御
PMR3	P3 ₂ /SO ₁ 、P3 ₀ /SCK ₁ 端子機能の設定
PMR7	P3 ₂ /SO ₁ 端子出力バッファのPMOSのON/OFFを制御

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりクロック同期式シリアルデータ送信を行います。

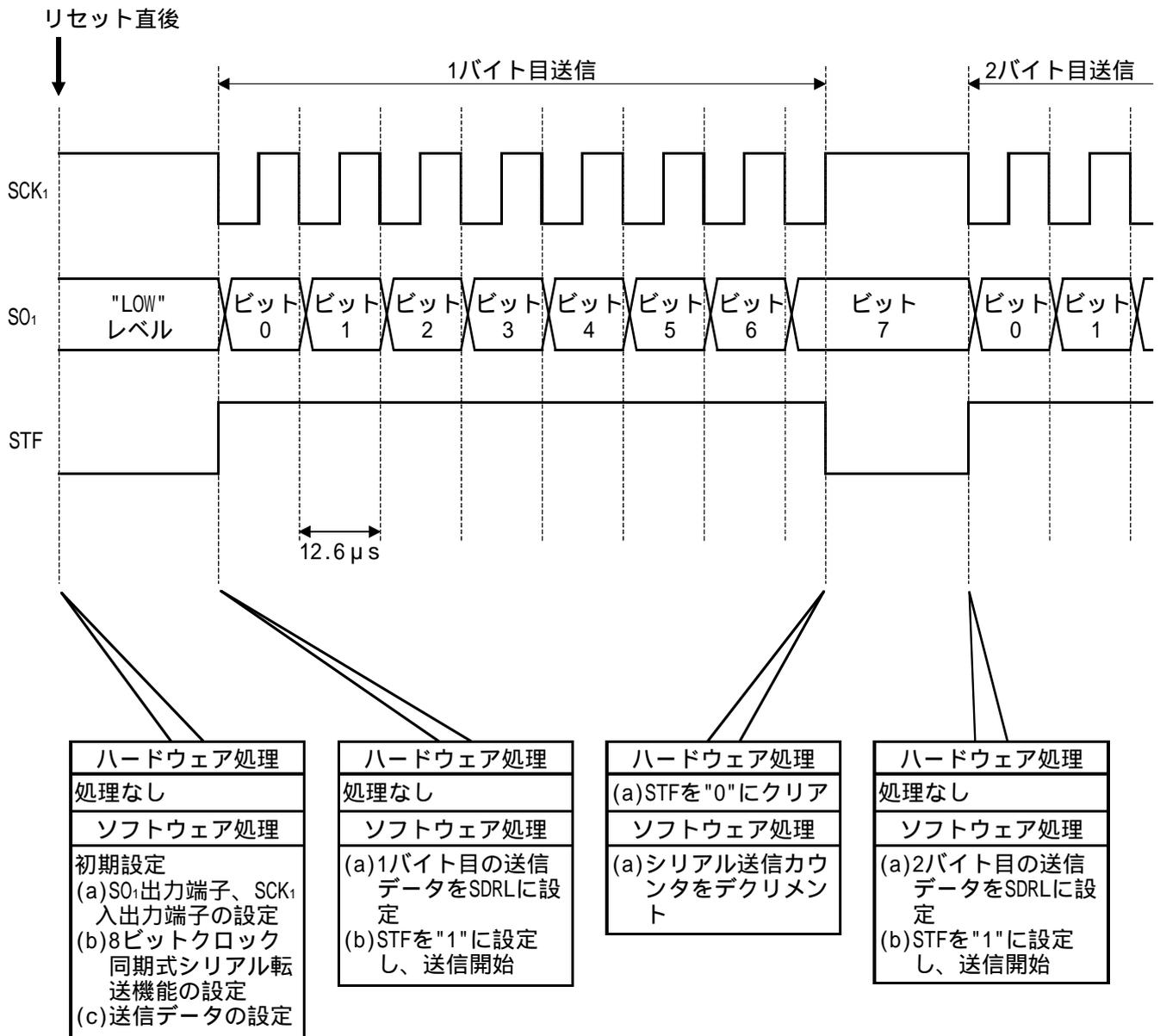


図3 クロック同期式シリアル送信の動作原理

クロック同期式シリアルデータ送信	使用機能	SC11 : クロック同期式シリアル転送機能
------------------	------	------------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、転送データの設定、クロック同期式シリアルデータ送信の設定、割込みの許可、4バイトのデータを送信したところで終了

(2) 引数の説明

表3に本タスク例で使用する引数を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0 ~ STD3	クロック同期式シリアル送信データ	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR1	SNC1 SNCO	シリアルコントロールレジスタ1 (動作モード選択1、0) : SNC1="0"、SNCO="0"のとき、動作モードを8ビットのモードに設定	H' FFA0 ビット7 ビット6	SNC1="0" SNCO="0"
	CKS3	シリアルコントロールレジスタ1 (クロックソース選択3) : CKS3="0"のとき、クロックソースをプリスケラSに、SCK ₁ 端子を出力に設定	H' FFA0 ビット3	0
	CKS2 CKS1 CKS0	シリアルコントロールレジスタ1 (クロック選択2、1、0) : CKS2="0"、CKS1="1"、CKS0="0"のとき、プリスケラ分周比を64分周に、転送クロック周期を12.8μsに設定	H' FFA0 ビット2 ビット1 ビット0	CKS2="0" CKS1="1" CKS0="0"
SCSR1	SOL	シリアルコントロールステータスレジスタ1 (拡張データビット) : SOL="0"のとき、S0 ₁ 端子出力を"Low"レベルに変更 : SOL="1"のとき、S0 ₁ 端子出力を"High"レベルに変更	H' FFA1 ビット6	0
	STF	シリアルコントロールステータスレジスタ1 (スタートフラグ) : STF="0"のとき、転送動作の終了 : STF="1"のとき、転送動作の開始	H' FFA1 ビット0	0
SDRL	シリアルデータレジスタL : 8ビット転送時、8ビット送信データを格納	H' FFA3	-	
IENR2	IENS1	割込み許可レジスタ2 (SC11割込みイネーブル) : IENS1="0"のとき、SC11割込み要求を禁止 : IENS1="1"のとき、SC11割込み要求を許可	H' FFF5 ビット4	0
IRR2	IRRS1	割込み要求レジスタ2 (SC11割込み要求フラグ) : IRRS1="0"のとき、SC11割込みが要求されていない : IRRS1="1"のとき、SC11割込みが要求されている	H' FFF8 ビット4	0
PMR3	S01	ポートモードレジスタ3 (P3 ₂ /S0 ₁ 端子機能切り替え) : S01="1"のとき、S0 ₁ 出力端子に設定	H' FFFD ビット2	1
	SCK1	ポートモードレジスタ3 (P3 ₀ /SCK ₁ 端子機能切り替え) : SCK1="1"のとき、SCK ₁ 入出力端子として機能	H' FFFD ビット0	1
PMR7	POF1	ポートモードレジスタ3 (P3 ₂ /S0 ₁ 端子PMOSコントロール) : POF1="0"のとき、CMOS出力	H' FFFF ビット0	0

ソフトウェア説明

(4) 使用RAM説明

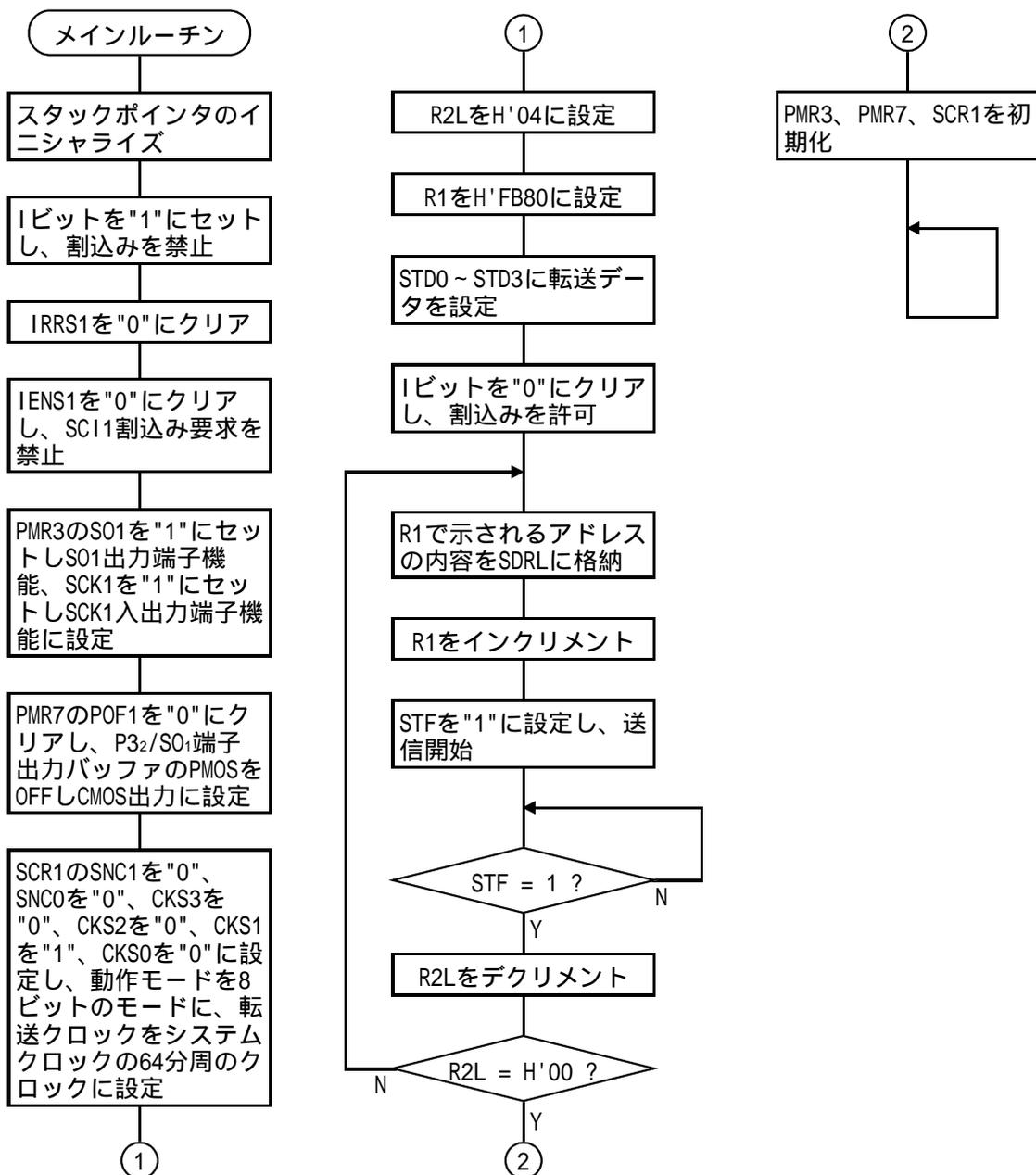
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0	クロック同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	クロック同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	クロック同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	クロック同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Synchronous Serial Data Transmission'
;
;       Function
;       : Serial Communication Interface
;       Synchronous Serial Interface
;       -Transmitting
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
;
SCR1      .equ          H'FFA0          ;Serial Control Register 1
SNC1      .bequ         7,SCR1          ;Select the Operation Mode 1
SNCO      .bequ         6,SCR1          ;Select the Operation Mode 0
MRKON     .bequ         5,SCR1          ;TAIL MARK Control
LTCH      .bequ         4,SCR1          ;LATCH TAIL Select
CKS3      .bequ         3,SCR1          ;Clock Source Slect 3
CKS2      .bequ         2,SCR1          ;Clock Slect 2
CKS1      .bequ         1,SCR1          ;Clock Slect 1
CKS0      .bequ         0,SCR1          ;Clock Slect 0
SCSR1     .equ          H'FFA1          ;Serial Control Status Register 1
SOL       .bequ         6,SCSR1         ;Extended Data Bit
ORER      .bequ         5,SCSR1         ;Overrun Errorr Flag
MTRF      .bequ         1,SCSR1         ;TAIL MARK Transmit Flag
STF       .bequ         0,SCSR1         ;Start Flag
SDRU      .equ          H'FFA2          ;Serial Data Register U
SDRL      .equ          H'FFA3          ;Serial Data Register L
IENR2     .equ          H'FFF5          ;Interrupt Enable Register 2
IENS1     .bequ         4,IENR2         ;SCI1 Interrupt Enable
IRR2      .equ          H'FFF8          ;Interrupt Request Register 2
IRRS1     .bequ         4,IRR2         ;SCI1 Interrupt Request Flag
PMR3      .equ          H'FFFD          ;Port Mode Register 3
S01       .bequ         2,PMR3         ;P32/S01 Pin Function Switch
SI1       .bequ         1,PMR3         ;P31/SI1 Pin Function Switch
SCK1      .bequ         0,PMR3         ;P30/SCK1 Pin Function Switch
PMR7      .equ          H'FFFF          ;Port Mode Register 7
POF1      .bequ         0,PMR7         ;P32/S01 Pin Function Switch
;
;
*****
;
;       RAM Allocation
;
*****
;
;
STACK     .equ          H'FF80          ;Stack Pointer
STD0      .equ          H'FB80          ;Serial Transmitting Data 0
STD1      .equ          H'FB81          ;Serial Transmitting Data 1

```

プログラムリスト

```

STD2      .equ      H'FB82      ;Serial Transmitting Data 2
STD3      .equ      H'FB83      ;Serial Transmitting Data 3
;
;
;*****
;
;          Vector Address
;*****
;
;          .org      H'0000
;          .data.w   MAIN      ;Reset Interrupt
;
;          .org      H'0008
;          .data.w   MAIN      ;IRQ0 Interrupt
;          .data.w   MAIN      ;IRQ1 Interrupt
;          .data.w   MAIN      ;IRQ2 Interrupt
;          .data.w   MAIN      ;IRQ3 Interrupt
;          .data.w   MAIN      ;INT0 - INT7 Interrupt
;
;          .org      H'0014
;          .data.w   MAIN      ;Timer A Interrupt
;          .data.w   MAIN      ;Timer B1 Interrupt
;
;          .org      H'0020
;          .data.w   MAIN      ;Timer X Interrupt
;          .data.w   MAIN      ;Timer V Interrupt
;
;          .org      H'0026
;          .data.w   MAIN      ;SCI1 Interrupt
;
;          .org      H'002A
;          .data.w   MAIN      ;SCI3 Interrupt
;          .data.w   MAIN      ;A/D Converter Interrupt
;          .data.w   MAIN      ;SLEEP Instruction Executed Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org      H'1000
;
MAIN      .equ      $
MOV.W    #STACK,SP ;Initialize Stack Pointer
ORC      #H'80,CCR  ;Interrupt Disable
;
;          BCLR     IRRS1    ;Clear IRRS1
;          BCLR     IENS1    ;SCI1 Interrupt Disable
;
;          MOV.W    #H'05F8,R0
;          MOV.B    ROH,@PMR3 ;Initialize S01 & CKS1 Pin Function
;          MOV.B    ROL,@PMR7 ;Initialize S01 Pin Function
;
;          MOV.B    #H'02,R0L
;          MOV.B    R0L,@SCR1 ;Initialize Synchronous Serial Transfer Function
;
;          MOV.W    #H'FB80,R1 ;Initialize Serial Transmitting Data Address
;          MOV.B    #H'04,R2L  ;Initialize Serial Transmitting Data Counter
;
;          MOV.W    #H'0055,R0
;          MOV.B    ROH,@STD0  ;Set Serial Transfer Data 0
;          MOV.B    ROL,@STD1  ;Set Serial Transfer Data 1
;          MOV.W    #H'AAFF,R0
    
```

プログラムリスト

```

MOV.B    R0H,@STD2    ;Set Serial Transfer Data 2
MOV.B    R0L,@STD3    ;Set Serial Transfer Data 3
;
;
;
MAIN1    .equ          $
MOV.B    @R1,R0L      ;Load Serial Transmitting Data
MOV.B    R0L,@SDRL    ;Save Serial Transmitting Data
ADDS     #1,R1        ;Increment Serial Transmitting Data Address
;
;
;
BSET     STF          ;Start Serial Transmitting
;
MAIN2    .equ          $
BTST     STF          ;End Serial Transmittig ?
BNE      MAIN2        ;No.
;
;
;
DEC      R2L          ;Decrement Serial Transmitting Data Counter
BNE      MAIN1        ;Serial Transmitting Data Counter = H'00 ? No.
;
;
;
MOV.B    #H'00,R0L    ;Initialize S01 & SCK1 Pin Function
MOV.B    R0L,@PMR3    ;Initialize S01 Pin Function
MOV.B    R0L,@PMR7    ;Initialize S01 Pin Function
MOV.B    R0L,@SCR1    ;Initialize Synchronous Serial Transfer Function
;
MAIN9    .equ          $
BRA      MAIN9
;
;
;
.end

```

2.14 クロック同期式シリアルデータ受信

クロック同期式シリアルデータ受信	使用機能	SCI1 : クロック同期式シリアル転送機能
------------------	------	------------------------

仕様

- (1) 図1に示すようにクロック同期式シリアル転送機能を使用して、4バイトの8ビットデータを受信します。
- (2) 転送クロックは、外部クロックを使用します。
- (3) 受信するデータのデータ長は8ビットで、データの最下位ビットから受信するLSBファースト方式による受信を行ないます。

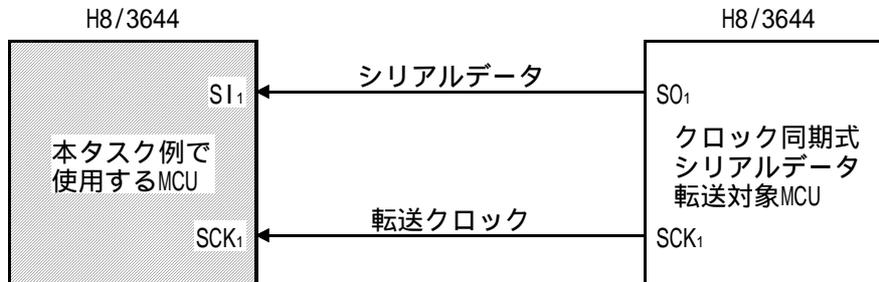


図1 クロック同期式シリアルデータ受信

使用機能説明

- (1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、クロック同期式のシリアルデータの受信を行ないます。図2にクロック同期式シリアルデータ受信のブロック図を示します。以下にクロック同期式シリアルデータ受信のブロック図について説明します。
 - ・シリアルコントロールレジスタ1 (SCR1) は、8ビットのリード/ライト可能なレジスタで、動作モード、転送クロックソースを選択します。
 - ・シリアルコントロールステータスレジスタ1 (SCSR1) は、動作状態、エラー状態等を示す8ビットのレジスタです。受信終了後、引き続き同期クロックが入力されると、受信は行なわれず、オーバラン状態であるとして、SCSR1のORERが"1"にセットされます。
 - ・シリアルデータレジスタU (SDRU) は、8ビットのリード/ライト可能なレジスタで、16ビット転送時に上位8ビットのデータレジスタとして使用します。SDRUに書き込まれたデータは、SDRLにLSBファーストで出力されます。入れ替わりにSI_i端子よりLSBファーストでデータが入力されて、MSB LSB方向にデータがシフトします。
 - ・シリアルデータレジスタL (SDRL) は、8ビットのリード/ライト可能なレジスタで、8ビット転送時のデータレジスタ、および16ビット転送時の下位8ビットのデータレジスタとして使用します。8ビット転送時、SDRLに書き込まれたデータは、SO_i端子よりLSBファーストで出力されます。入れ替わりにSI_i端子よりLSBファーストで入力されて、MSB LSB方向にデータがシフトします。16ビット転送時には、入力データがSDRUより取り込まれることを除けば、8ビット転送時と同様の動作となります。
 - ・SDRU、SDRLのリード/ライトは、データの送信/受信が完了してから行なう必要があります。データの送信/受信中にリード/ライトを行なうとデータの内容は保証されません。
 - ・転送クロックは、8種類の内部クロックと外部クロックから選択できます。内部クロックを選択した場合は、SCK_i端子は出力端子となります。クロック連続出力モードに設定すると選択したクロックをSCK_i端子から連続して出力します。外部クロックを選択した場合は、SCK_i端子はクロック入力端子となります。
 - ・本タスク例では、転送クロックソースを外部クロックに設定しています。
 - ・SCI1の転送フォーマットは8ビットおよび16ビットの転送データを選択可能です。データの最下位ビットから送受信されるLSBファースト方式による転送を行ないます。送信データは、転送クロックの立ち下がりから次の立ち上がりまで出力されます。また、受信データは転送クロックの立ち上がりで取り込まれます。
 - ・本タスク例では、動作モードを8ビットモードに設定し、8ビットのデータ受信を行ないます。
 - ・SCI1クロック (SCK_i) は、SCI1のクロック入出力端子です。
 - ・SCI1データ入力 (SI_i) は、SCI1の受信データの入力端子です。
 - ・SCI1が転送完了すると、割込み要求レジスタ2 (IRR2) のCSI1割込み要求フラグ (IRRS1) が"1"にセットされます。SCI1の割込み要求は、割込み許可レジスタ2 (IENR2) のSCI1割込みイネーブル (IENS1) により許可/禁止を選択できます。

使用機能説明

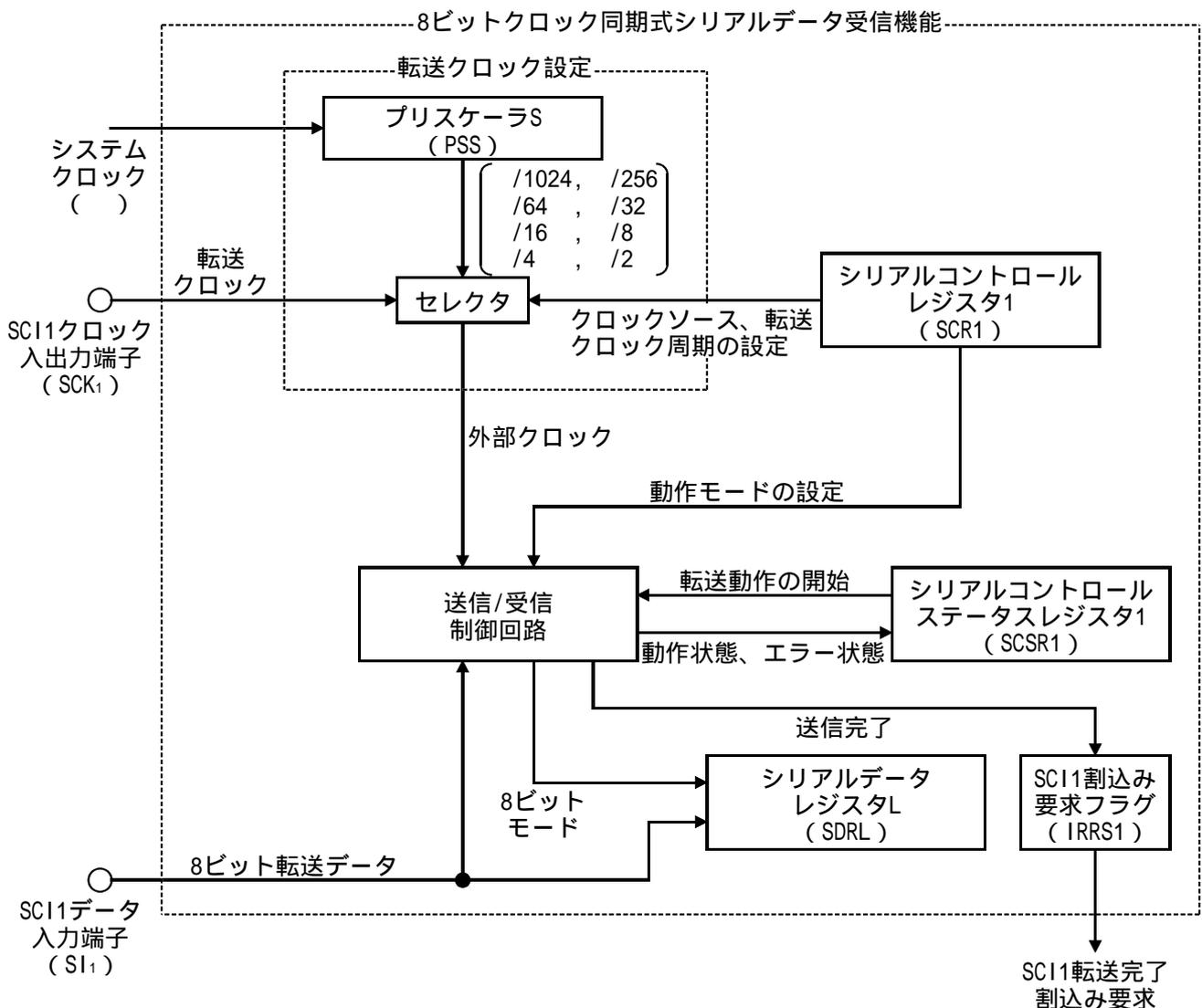


図2 クロック同期式シリアルデータ受信機能のブロック図

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、クロック同期式シリアルデータ受信を行ないます。

表1 機能割付け

機能	機能割付け
SCR1	動作モード、転送クロックソース、プリスケラ分周比の設定
SCSR1	動作状態、エラー状態を示す
SDRL	8ビットの受信データのデータレジスタ
SCK ₁	SCI1の転送クロック入力端子
SI ₁	SCI1の受信データの入力端子
IRRS1	SCI1の転送完了の有無を示す
IENS1	SCI1割り込み要求の許可/禁止を制御
PMR3	P3 ₁ /SI ₁ 、P3 ₀ /SCK ₁ 端子機能の設定

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりクロック同期式シリアルデータ受信を行います。

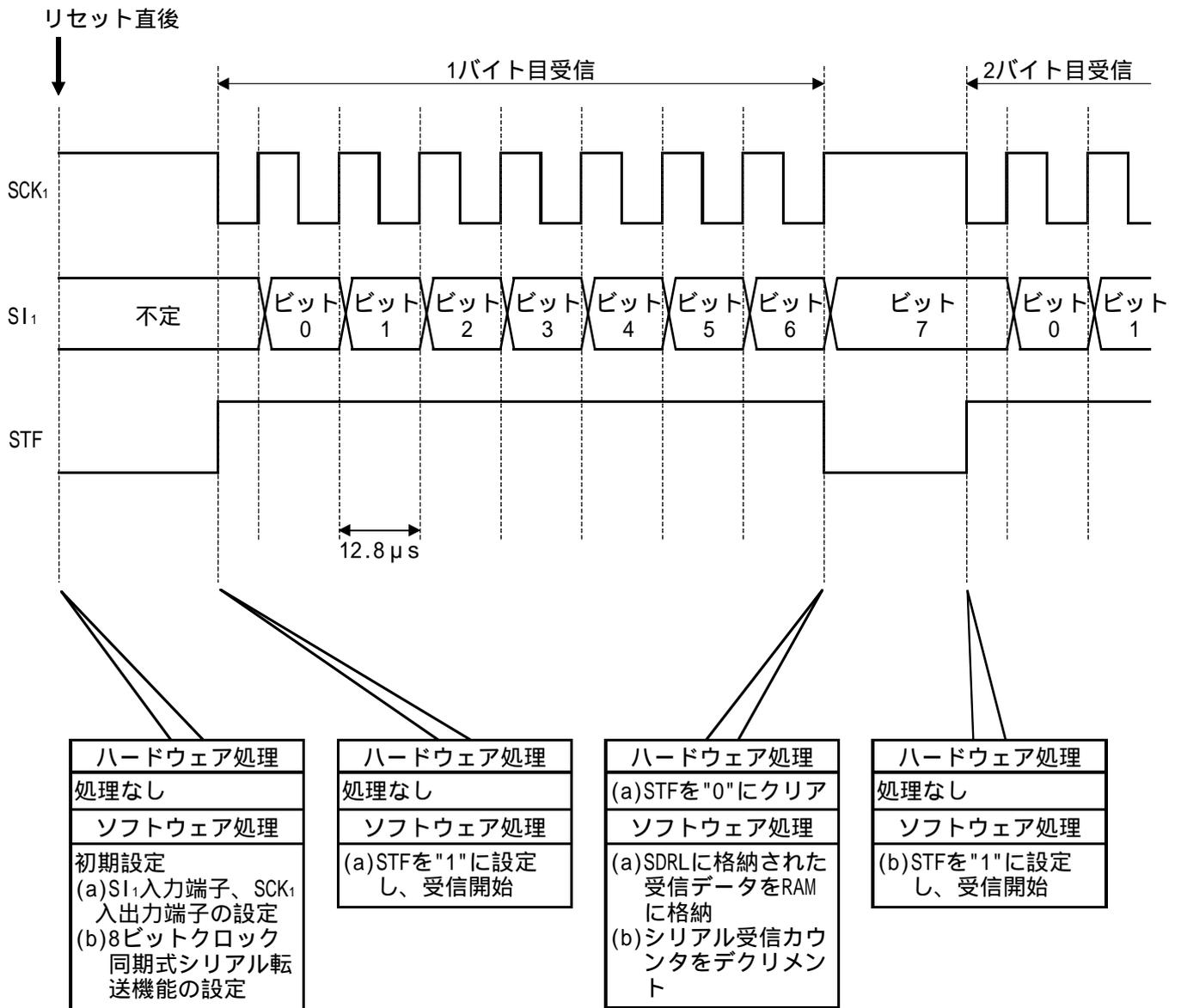


図3 クロック同期式シリアル受信の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、クロック同期式シリアルデータ受信の設定、割込みの許可、受信データをRAMに格納、4バイトのデータを受信したところで終了

(2) 引数の説明

表3に本タスク例で使用する引数を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
SRD0 ~ SRD3	クロック同期式シリアル受信データ	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR1	SNC1 SNCO	シリアルコントロールレジスタ1 (動作モード選択1、0) : SNC1="0"、SNCO="0"のとき、動作モードを8ビットのモードに設定	H' FFA0 ビット7 ビット6	SNC1="0" SNCO="0"
	MRKON	シリアルコントロールレジスタ1 (TAIL MARK制御) : MRKON="0"のとき、TAIL MARKを非出力	H' FFA0 ビット5	0
	CKS3	シリアルコントロールレジスタ1 (クロックソース選択3) : CKS3="1"のとき、クロックソースを外部クロックに設定	H' FFA0 ビット3	1
SCSR1	ORER	シリアルコントロールステータスレジスタ1 (オーバランエラーフラグ) : STF="0"のとき、オーバランエラーが発生していないことを示す : STF="1"のとき、オーバランエラーが発生したことを示す	H' FFA1 ビット5	0
	STF	シリアルコントロールステータスレジスタ1 (スタートフラグ) : STF="0"のとき、転送動作の終了 : STF="1"のとき、転送動作の開始	H' FFA1 ビット0	0
SDRL	シリアルデータレジスタL : 8ビット転送時、8ビット受信データを格納	H' FFA3	-	
IENR2	IENS1	割込み許可レジスタ2 (SC11割込みイネーブル) : IENS1="0"のとき、SC11割込み要求を禁止 : IENS1="1"のとき、SC11割込み要求を許可	H' FFF5 ビット4	0
IRR2	IRRS1	割込み要求レジスタ2 (SC11割込み要求フラグ) : IRRS1="0"のとき、SC11割込みが要求されていない : IRRS1="1"のとき、SC11割込みが要求されている	H' FFF8 ビット4	0
PMR3	SI1	ポートモードレジスタ3 (P3 ₁ /SI ₁ 端子機能切り替え) : SI1="1"のとき、SI ₁ 入力端子に設定	H' FFFD ビット1	1
	SCK1	ポートモードレジスタ3 (P3 ₀ /SCK ₁ 端子機能切り替え) : SCK1="1"のとき、SCK ₁ 入出力端子として機能	H' FFFD ビット0	1

ソフトウェア説明

(4) 使用RAM説明

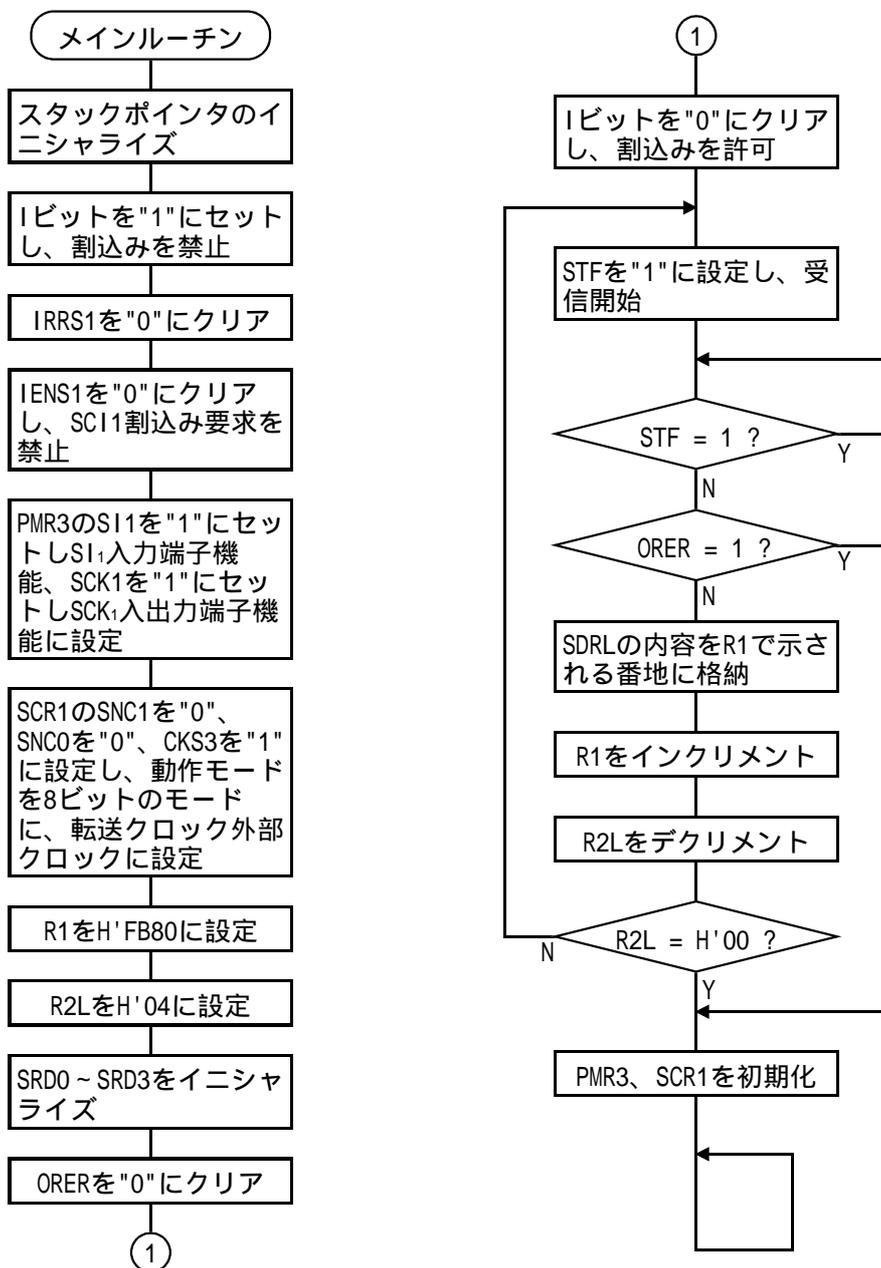
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
SRD0	クロック同期式シリアル受信データの1バイト目を格納	H'FB80	メインルーチン
SRD1	クロック同期式シリアル受信データの2バイト目を格納	H'FB81	メインルーチン
SRD2	クロック同期式シリアル受信データの3バイト目を格納	H'FB82	メインルーチン
SRD3	クロック同期式シリアル受信データの4バイト目を格納	H'FB83	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Synchronous Serial Data Reception'
;
;       Function
;       : Serial Communication Interface
;       Synchronous Serial Interface
;       -Receiving
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
;
SCR1      .equ        H'FFA0      ;Serial Control Register 1
SNC1      .bequ       7,SCR1      ;Select the Operation Mode 1
SNCO      .bequ       6,SCR1      ;Select the Operation Mode 0
MRKON     .bequ       5,SCR1      ;TAIL MARK Control
LTCH      .bequ       4,SCR1      ;LATCH TAIL Select
CKS3      .bequ       3,SCR1      ;Clock Source Slect 3
CKS2      .bequ       2,SCR1      ;Clock Slect 2
CKS1      .bequ       1,SCR1      ;Clock Slect 1
CKS0      .bequ       0,SCR1      ;Clock Slect 0
SCSR1     .equ        H'FFA1      ;Serial Control Status Register 1
SOL       .bequ       6,SCSR1     ;Extended Data Bit
ORER      .bequ       5,SCSR1     ;Overrun Errorr Flag
MTRF      .bequ       1,SCSR1     ;TAIL MARK Transmit Flag
STF       .bequ       0,SCSR1     ;Start Flag
SDRU      .equ        H'FFA2      ;Serial Data Register U
SDRL      .equ        H'FFA3      ;Serial Data Register L
IENR2     .equ        H'FFF5      ;Interrupt Enable Register 2
IENS1     .bequ       4,IENR2     ;SCI1 Interrupt Enable
IRR2      .equ        H'FFF8      ;Interrupt Request Register 2
IRRS1     .bequ       4,IRR2      ;SCI1 Interrupt Request Flag
PMR3      .equ        H'FFFD      ;Port Mode Register 3
S01       .bequ       2,PMR3      ;P32/S01 Pin Function Switch
SI1       .bequ       1,PMR3      ;P31/SI1 Pin Function Switch
SCK1      .bequ       0,PMR3      ;P30/SCK1 Pin Function Switch
;
;
*****
;
;       RAM Allocation
;
*****
;
;
STACK     .equ        H'FF80      ;Stack Pointer
SRD0      .equ        H'FB80      ;Serial Receiving Data 0
SRD1      .equ        H'FB81      ;Serial Receiving Data 1
SRD2      .equ        H'FB82      ;Serial Receiving Data 2
SRD3      .equ        H'FB83      ;Serial Receiving Data 3

```

プログラムリスト

```

;
;*****
;
;      Vector Address
;*****
;
;
;      .org          H'0000
;      .data.w      MAIN          ;Reset Interrupt
;
;
;      .org          H'0008
;      .data.w      MAIN          ;IRQ0 Interrupt
;      .data.w      MAIN          ;IRQ1 Interrupt
;      .data.w      MAIN          ;IRQ2 Interrupt
;      .data.w      MAIN          ;IRQ3 Interrupt
;      .data.w      MAIN          ;INT0 - INT7 Interrupt
;
;
;      .org          H'0014
;      .data.w      MAIN          ;Timer A Interrupt
;      .data.w      MAIN          ;Timer B1 Interrupt
;
;
;      .org          H'0020
;      .data.w      MAIN          ;Timer X Interrupt
;      .data.w      MAIN          ;Timer V Interrupt
;
;
;      .org          H'0026
;      .data.w      MAIN          ;SCI1 Interrupt
;
;
;      .org          H'002A
;      .data.w      MAIN          ;SCI3 Interrupt
;      .data.w      MAIN          ;A/D Converter Interrupt
;      .data.w      MAIN          ;SLEEP Instruction Executed Interrupt
;
;*****
;
;      Main Program
;*****
;
;
;      .org          H'1000
;
;      MAIN
;      .equ          $
;      MOV.W        #STACK,SP      ;Initialize Stack Pointer
;      ORC          #H'80,CCR      ;Interrupt Disable
;
;
;      BCLR         IRRS1          ;Clear IRRS1
;      BCLR         IENS1          ;SCI1 Interrupt Disable
;
;
;      MOV.W        #H'0308,R0     ;Initialize S11 & CKS1 Pin Function
;      MOV.B        R0H,@PMR3      ;Initialize Synchronous Serial Transfer Function
;      MOV.B        R0L,@SCR1
;
;
;      MOV.W        #H'FB80,R1     ;Initialize Serial Receiving Data Address
;      MOV.B        #H'04,R2L      ;Initialize Serial Receiving Data Counter
;
;
;      MOV.B        #H'00,R0L
;      MOV.B        R0L,@SRD0      ;Initialize Serial Receiving Data 0
;      MOV.B        R0L,@SRD1      ;Initialize Serial Receiving Data 1
;      MOV.B        R0L,@SRD2      ;Initialize Serial Receiving Data 2
;      MOV.B        R0L,@SRD3      ;Initialize Serial Receiving Data 3
;
;
;      BTST        ORER
;      BCLR        ORER            ;Clear ORER
;
;

```

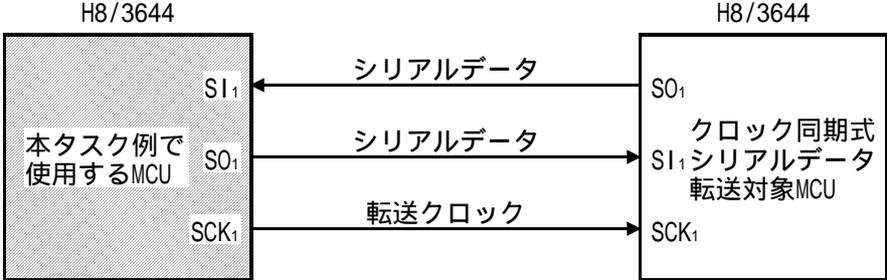
プログラムリスト

```

;
;
MAIN1      .equ          $
           BSET          STF          ;Start Serial Receiving
;
MAIN2      .equ          $
           BTST          STF          ;End Serial Receiving ?
           BNE          MAIN2       ;No.
;
           BTST          ORER        ;Overrun Errorr Flag = 1 ?
           BNE          MAIN3       ;Yes.
;
           MOV.B        @SDRL,R0L   ;Load
           MOV.B        R0L,@R1    ;Save
;
           ADDS         #1,R1       ;Increment Serial Receiving Data Address
           DEC          R2L         ;Decrement Serial Receiving Data Counter
           BNE          MAIN1       ;Serial Receiving Data Counter = H'00 ? No.
;
           MOV.B        #H'00,R0L  ;Initialize S11 & SCK1 Pin Function
           MOV.B        R0L,@PMR3  ;Initialize Synchronous Serial Transfer Function
           MOV.B        R0L,@SCR1
           BRA          MAIN9
;
MAIN3      .equ          $
           MOV.B        #H'FF,R0L  ;Overrun Errorr
           MOV.B        R0L,@SRD0  ;Overrun Errorr
           MOV.B        R0L,@SRD1  ;Overrun Errorr
           MOV.B        R0L,@SRD2  ;Overrun Errorr
           MOV.B        R0L,@SRD3  ;Overrun Errorr
;
           MOV.B        R0L,@PMR3  ;Initialize S11 & SCK1 Pin Function
           MOV.B        R0L,@SCR1  ;Initialize Synchronous Serial Transfer Function
;
MAIN9      .equ          $
           BRA          MAIN9
;
           .end

```

2.15 クロック同期式シリアルデータ同時送受信

クロック同期式シリアルデータ同時送受信	使用機能	SCI1 : クロック同期式シリアル転送機能
仕様		
<p>(1) 図1に示すようにクロック同期式シリアル転送機能を使用して、4バイトの8ビットデータの同時送受信動作を行ないます。</p> <p>(2) 転送クロックは、内部クロックを使用し12.8 μsの転送クロック周期で同時送受信動作を行います。</p> <p>(3) 送受信するデータのデータ長は8ビットで、データの最下位ビットから受信するLSBファースト方式による送受信を行ないます。</p>		
		
図1 クロック同期式シリアルデータ同時送受信		

使用機能説明

- (1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、クロック同期式のシリアルデータの送受信同時動作を行ないます。図2にクロック同期式シリアルデータ同時送受信動作のブロック図を示します。以下にクロック同期式シリアルデータ同時送受信動作のブロック図について説明します。
- ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - ・プリスケラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - ・シリアルコントロールレジスタ1 (SCR1) は、8ビットのリード/ライト可能なレジスタで、動作モード、転送クロックソースとプリスケラ分周比を選択します。
 - ・シリアルコントロールステータスレジスタ1 (SCSR1) は、動作状態、エラー状態等を示す8ビットのレジスタです。受信終了後、引き続き同期クロックが入力されると、受信は行なわれず、オーバラン状態であるとして、SCSR1のORERが"1"にセットされます。
 - ・シリアルデータレジスタU (SDRU) は、8ビットのリード/ライト可能なレジスタで、16ビット転送時に上位8ビットのデータレジスタとして使用します。SDRUに書き込まれたデータは、SDRLにLSBファーストで出力されます。入れ替わりにSI_i端子よりLSBファーストでデータが入力されて、MSB LSB方向にデータがシフトします。
 - ・シリアルデータレジスタL (SDRL) は、8ビットのリード/ライト可能なレジスタで、8ビット転送時のデータレジスタ、および16ビット転送時の下位8ビットのデータレジスタとして使用します。8ビット転送時、SDRLに書き込まれたデータは、SO_i端子よりLSBファーストで出力されます。入れ替わりにSI_i端子よりLSBファーストで入力されて、MSB LSB方向にデータがシフトします。16ビット転送時には、入力データがSDRUより取り込まれることを除けば、8ビット転送時と同様の動作となります。
 - ・SDRU、SDRLのリード/ライトは、データの送信/受信が完了してから行なう必要があります。データの送信/受信中にリード/ライトを行なうとデータの内容は保証されません。
 - ・転送クロックは、8種類の内部クロックと外部クロックから選択できます。内部クロックを選択した場合は、SCK_i端子は出力端子となります。クロック連続出力モードに設定すると選択したクロックをSCK_i端子から連続して出力します。外部クロックを選択した場合は、SCK_i端子はクロック入力端子となります。
 - ・本タスク例では、転送クロックソースをPSSに、プリスケラ分周比を64分周、転送クロック周期を12.8 μ sに設定しています。
 - ・SCI1の転送フォーマットは8ビットおよび16ビットの転送データを選択可能です。データの最下位ビットから送受信されるLSBファースト方式による転送を行ないます。送信データは、転送クロックの立ち上がりから次の立ち上がりまで出力されます。また、受信データは転送クロックの立ち上がりで取り込まれます。
 - ・本タスク例では、動作モードを8ビットモードに設定し、8ビットのデータ送受信を行ないます。
 - ・SCI1クロック (SCK_i) は、SCI1のクロック入出力端子です。
 - ・SCI1データ出力 (SO_i) は、SCI1の送信データの出力端子です。
 - ・SCI1データ入力 (SI_i) は、SCI1の受信データの入力端子です。
 - ・SCI1が転送完了すると、割込み要求レジスタ2 (IRR2) のCSI1割込み要求フラグ (IRRS1) が"1"にセットされます。SCI1の割込み要求は、割込み許可レジスタ2 (IENR2) のSCI1割込みイネーブル (IENS1) により許可/禁止を選択できます。

使用機能説明

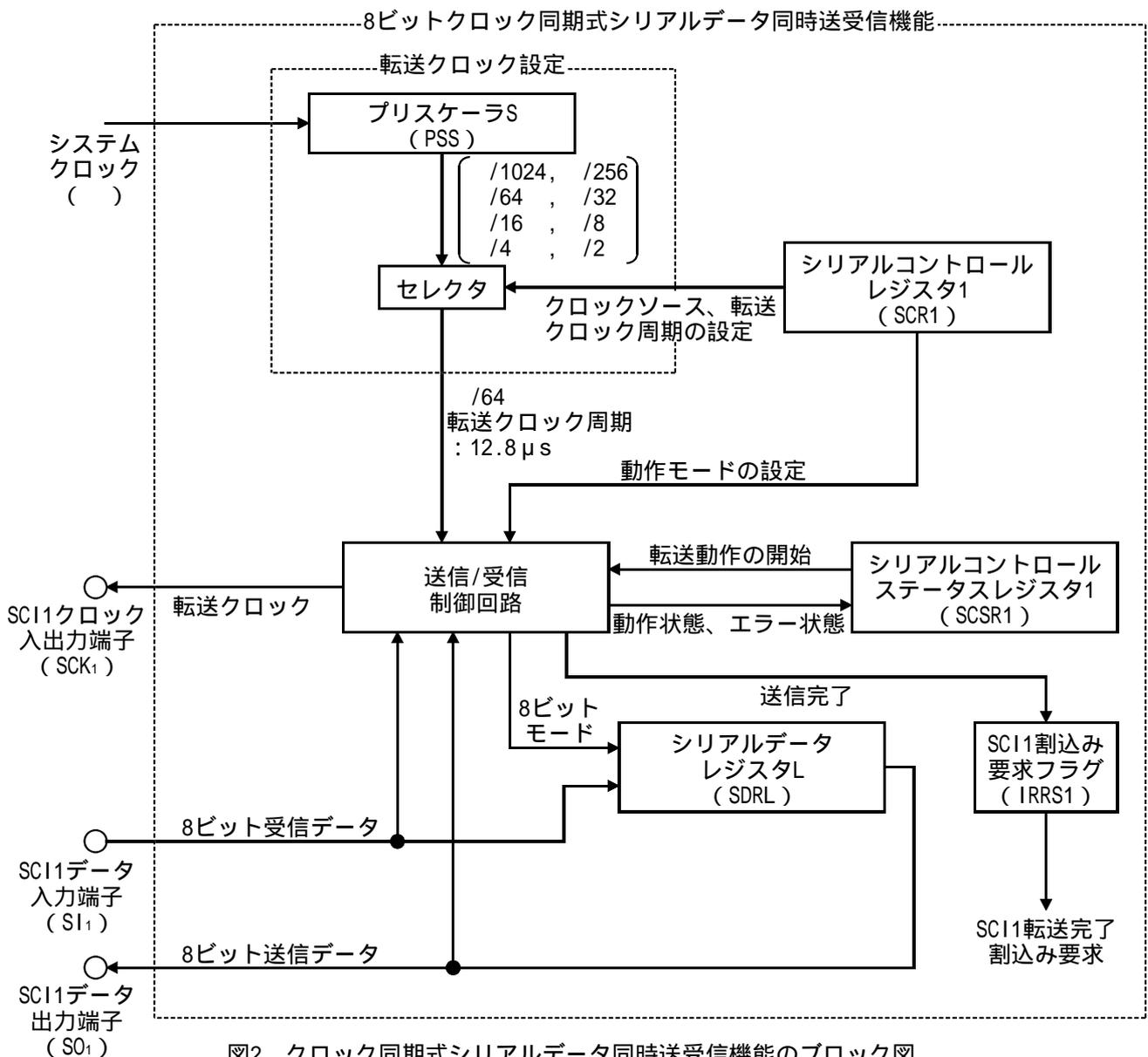


図2 クロック同期式シリアルデータ同時送受信機能のブロック図

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、クロック同期式シリアルデータ同時送受信を行ないます。

表1 機能割付け

機能	機能割付け
SCR1	動作モード、転送クロックソース、プリスケアラ分周比の設定
SCSR1	動作状態、エラー状態を示す
SDRL	8ビットの送受信データのデータレジスタ
SCK ₁	SCI1の転送クロック入力端子
SO ₁	SCI1の送信データの出力端子
SI ₁	SCI1の受信データの入力端子
IRRS1	SCI1の転送完了の有無を示す
IENS1	SCI1割込み要求の許可/禁止を制御
PMR3	P3 ₂ /SO ₁ 、P3 ₁ /SI ₁ 、P3 ₀ /SCK ₁ 端子機能の設定
PMR7	P3 ₂ /SO ₁ 端子出力バッファのPMOSのON/OFFを制御

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりクロック同期式シリアルデータ同時送受信動作を行ないます。

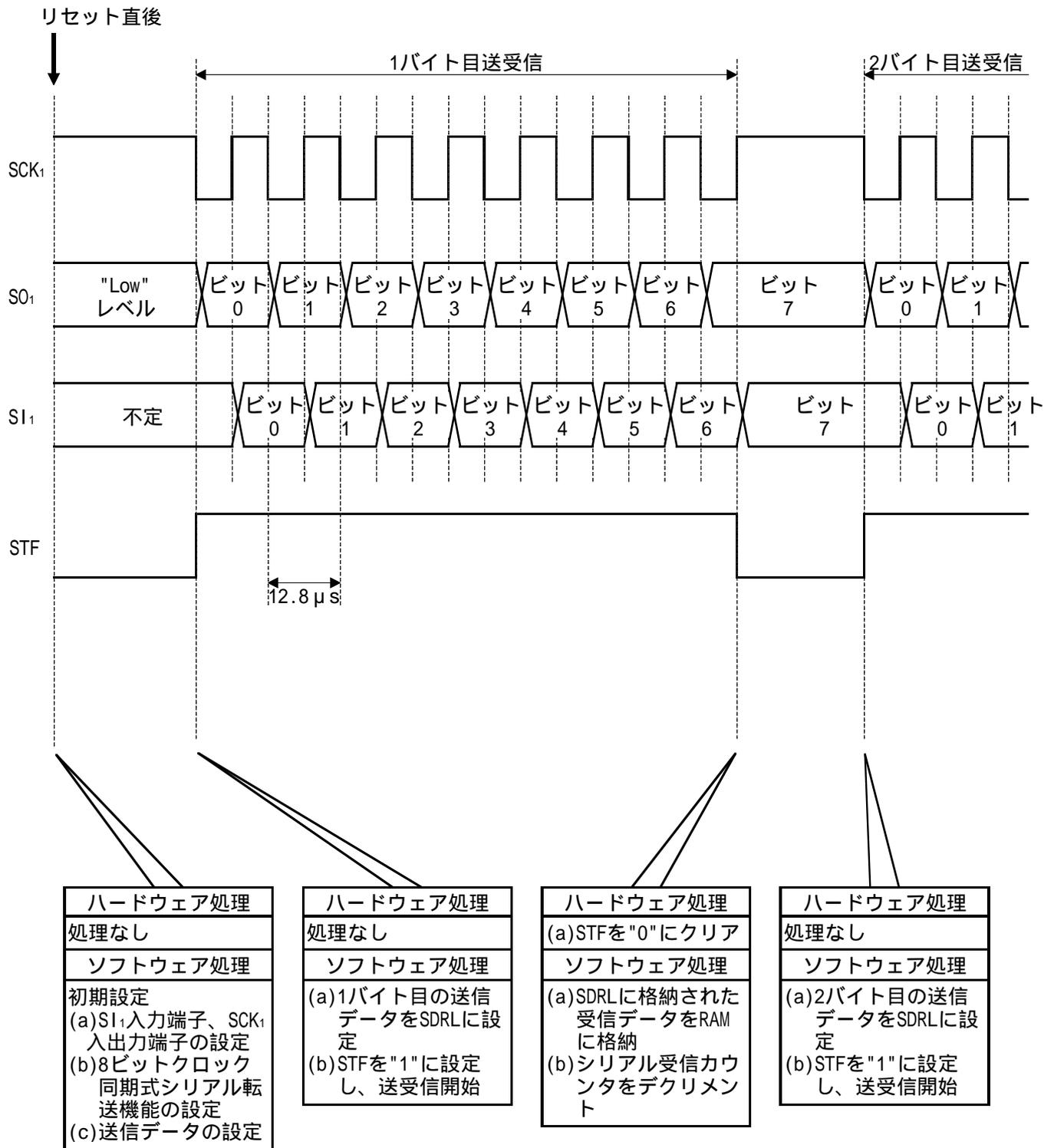


図3 クロック同期式シリアル同時送受信の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、転送データの設定、クロック同期式シリアルデータ送受信の設定、割込みの許可、受信データをRAMに格納、4バイトのデータを送受信したところで終了

(2) 引数の説明

表3に本タスク例で使用する引数を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0 ~ STD3	クロック同期式シリアル送信データ	メインルーチン	1バイト	入力
SRD0 ~ SRD3	クロック同期式シリアル受信データ	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR1	SNC1 SNCO	シリアルコントロールレジスタ1 (動作モード選択1、0) : SNC1="0"、SNCO="0"のとき、動作モードを8ビットのモードに設定	H' FFA0 ビット7 ビット6	SNC1="0" SNCO="0"
	MRKON	シリアルコントロールレジスタ1 (TAIL MARK制御) : MRKON="0"のとき、TAIL MARKを非出力	H' FFA0 ビット5	0
	CKS3	シリアルコントロールレジスタ1 (クロックソース選択3) : CKS3="1"のとき、クロックソースをプリスケラSに設定	H' FFA0 ビット3	0
	CKS2 CKS1 CKS0	シリアルコントロールレジスタ1 (クロック選択2、1、0) : CKS2="0"、CKS1="1"、CKS0="0"のとき、プリスケラ分周比を64分周に、転送クロック周期を12.8μsに設定	H' FFA0 ビット2 ビット1 ビット0	CKS2="0" CKS1="1" CKS0="0"
SCSR1	SOL	シリアルコントロールステータスレジスタ1 (拡張データビット) : SOL="0"のとき、S01端子出力を"Low"レベルに変更 : SOL="1"のとき、S01端子出力を"High"レベルに変更	H' FFA1 ビット6	0
	ORER	シリアルコントロールステータスレジスタ1 (オーバランエラーフラグ) : STF="0"のとき、オーバランエラーが発生していない : STF="1"のとき、オーバランエラーが発生した	H' FFA1 ビット5	0
	STF	シリアルコントロールステータスレジスタ1 (スタートフラグ) : STF="0"のとき、転送動作の終了 : STF="1"のとき、転送動作の開始	H' FFA1 ビット0	0
SDRL	シリアルデータレジスタL : 8ビット転送時、8ビット受信データを格納	H' FFA3	-	

ソフトウェア説明

表4 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
IENR2	IENS1	割込み許可レジスタ2 (SCI1割込みイネーブル) : IENS1="0"のとき、SCI1割込み要求を禁止 : IENS1="1"のとき、SCI1割込み要求を許可	H'FFF5 ビット4	0
IRR2	IRRS1	割込み要求レジスタ2 (SCI1割込み要求フラグ) : IRRS1="0"のとき、SCI1割込みが要求されていない : IRRS1="1"のとき、SCI1割込みが要求されている	H'FFF8 ビット4	0
PMR3	S01	ポートモードレジスタ3 (P3 ₁ /SI ₁ 端子機能切り替え) : S01="1"のとき、S0 ₁ 出力端子に設定	H'FFFD ビット2	1
	SI1	ポートモードレジスタ3 (P3 ₁ /SI ₁ 端子機能切り替え) : SI1="1"のとき、SI ₁ 入力端子に設定	H'FFFD ビット1	1
	SCK1	ポートモードレジスタ3 (P3 ₀ /SCK ₁ 端子機能切り替え) : SCK1="1"のとき、SCK ₁ 入出力端子として機能	H'FFFD ビット0	1

(4) 使用RAM説明

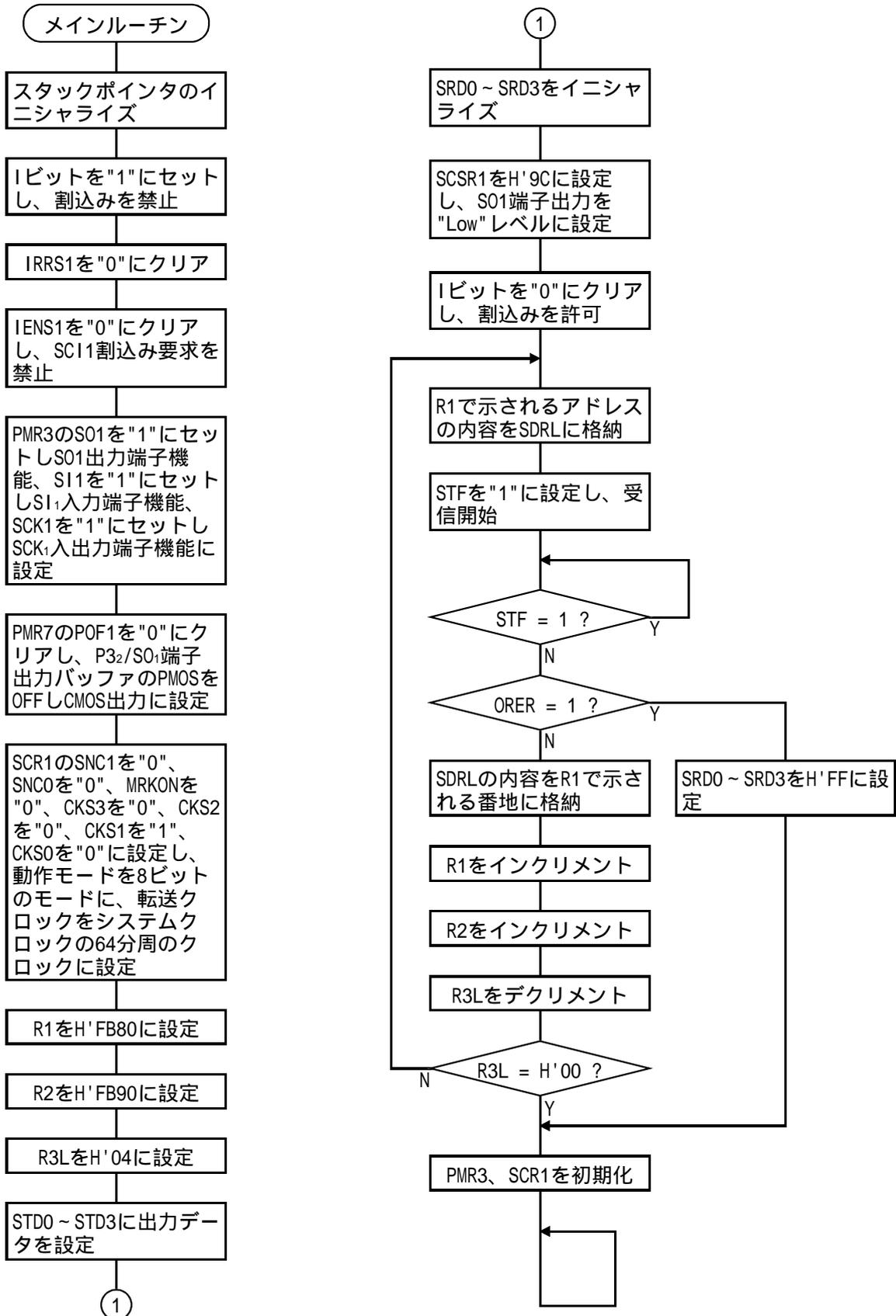
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0	クロック同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	クロック同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	クロック同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	クロック同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン
SRD0	クロック同期式シリアル受信データの1バイト目を格納	H'FB90	メインルーチン
SRD1	クロック同期式シリアル受信データの2バイト目を格納	H'FB91	メインルーチン
SRD2	クロック同期式シリアル受信データの3バイト目を格納	H'FB92	メインルーチン
SRD3	クロック同期式シリアル受信データの4バイト目を格納	H'FB93	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Synchronous Serial Data Simultaneous
;       Transmission/Reception'
;
;       Function
;       : Serial Communication Interface
;       Synchronous Serial Interface
;       -Transmitting/Receiving
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
SCR1      .equ        H'FFA0      ;Serial Control Register 1
SNC1      .bequ       7,SCR1      ;Select the Operation Mode 1
SNCO      .bequ       6,SCR1      ;Select the Operation Mode 0
MRKON     .bequ       5,SCR1      ;TAIL MARK Control
LTCH      .bequ       4,SCR1      ;LATCH TAIL Select
CKS3      .bequ       3,SCR1      ;Clock Source Slect 3
CKS2      .bequ       2,SCR1      ;Clock Slect 2
CKS1      .bequ       1,SCR1      ;Clock Slect 1
CKS0      .bequ       0,SCR1      ;Clock Slect 0
SCSR1     .equ        H'FFA1      ;Serial Control Status Register 1
SOL       .bequ       6,SCSR1     ;Extended Data Bit
ORER      .bequ       5,SCSR1     ;Overrun Errorr Flag
MTRF      .bequ       1,SCSR1     ;TAIL MARK Transmit Flag
STF       .bequ       0,SCSR1     ;Start Flag
SDRU      .equ        H'FFA2      ;Serial Data Register U
SDRL      .equ        H'FFA3      ;Serial Data Register L
IENR2     .equ        H'FFF5      ;Interrupt Enable Register 2
IENS1     .bequ       4,IENR2     ;SCI1 Interrupt Enable
IRR2      .equ        H'FFF8      ;Interrupt Request Register 2
IRRS1     .bequ       4,IRR2      ;SCI1 Interrupt Request Flag
PMR3      .equ        H'FFFD      ;Port Mode Register 3
S01       .bequ       2,PMR3      ;P32/S01 Pin Function Switch
SI1       .bequ       1,PMR3      ;P31/S11 Pin Function Switch
SCK1      .bequ       0,PMR3      ;P30/SCK1 Pin Function Switch
PMR7      .equ        H'FFFF      ;Port Mode Register 7
POF1      .bequ       0,PMR7      ;P32/S01 Pin Function Switch
;
;
;       RAM Allocation
;
*****
;
STACK     .equ        H'FF80      ;Stack Pointer
STD0      .equ        H'FB80      ;Serial Transmitting Data 0

```

プログラムリスト

```

STD1      .equ      H'FB81      ;Serial Transmitting Data 1
STD2      .equ      H'FB82      ;Serial Transmitting Data 2
STD3      .equ      H'FB83      ;Serial Transmitting Data 3
SRD0      .equ      H'FB90      ;Serial Receiving Data 0
SRD1      .equ      H'FB91      ;Serial Receiving Data 1
SRD2      .equ      H'FB92      ;Serial Receiving Data 2
SRD3      .equ      H'FB93      ;Serial Receiving Data 3
;
;*****
;
;          Vector Address
;*****
;
;          .org      H'0000
;          .data.w   MAIN      ;Reset Interrupt
;
;          .org      H'0008
;          .data.w   MAIN      ;IRQ0 Interrupt
;          .data.w   MAIN      ;IRQ1 Interrupt
;          .data.w   MAIN      ;IRQ2 Interrupt
;          .data.w   MAIN      ;IRQ3 Interrupt
;          .data.w   MAIN      ;INT0 - INT7 Interrupt
;
;          .org      H'0014
;          .data.w   MAIN      ;Timer A Interrupt
;          .data.w   MAIN      ;Timer B1 Interrupt
;
;          .org      H'0020
;          .data.w   MAIN      ;Timer X Interrupt
;          .data.w   MAIN      ;Timer V Interrupt
;
;          .org      H'0026
;          .data.w   MAIN      ;SCI1 Interrupt
;
;          .org      H'002A
;          .data.w   MAIN      ;SCI3 Interrupt
;          .data.w   MAIN      ;A/D Converter Interrupt
;          .data.w   MAIN      ;SLEEP Instruction Executed Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org      H'1000
;
;MAIN      .equ      $
;          MOV.W     #STACK,SP   ;Initialize Stack Pointer
;          ORC       #H'80,CCR    ;Interrupt Disable
;
;          BCLR      IRRS1       ;Clear IRRS1
;          BCLR      IENS1       ;SCI1 Interrupt Disable
;
;          MOV.W     #H'07F8,R0
;          MOV.B     ROH,@PMR3    ;Initialize S01 & S11 & CKS1 Pin Function
;          MOV.B     ROL,@PMR7    ;Initialize S01 Pin Function
;
;          MOV.B     #H'02,R0L
;          MOV.B     ROL,@SCR1    ;Initialize Synchronous Serial Transfer Function
;
;          MOV.W     #H'FB80,R1   ;Initialize Serial Transmitting Data Address
;          MOV.W     #H'FB90,R2   ;Initialize Serial Receiving Data Address

```

プログラムリスト

```

;
MOV.B    #H'04,R3L    ;Initialize Serial Data Counter
;
MOV.W    #H'0055,R0
MOV.B    R0H,@STD0    ;Set Serial Transmitting Data 0
MOV.B    R0L,@STD1    ;Set Serial Transmitting Data 1
MOV.W    #H'AAFF,R0
MOV.B    R0H,@STD2    ;Set Serial Transmitting Data 2
MOV.B    R0L,@STD3    ;Set Serial Transmitting Data 3
;
MOV.B    #H'00,R0L
MOV.B    R0L,@SRD0    ;Initialize Serial Receiving Data 0
MOV.B    R0L,@SRD1    ;Initialize Serial Receiving Data 1
MOV.B    R0L,@SRD2    ;Initialize Serial Receiving Data 2
MOV.B    R0L,@SRD3    ;Initialize Serial Receiving Data 3
;
MOV.B    #H'9C,R0L
MOV.B    R0L,@SCSR1   ;Initialize S01 Pin Output Level
;
ANDC     #H'7F,CCR    ;Interrupt Enable
;
MAIN1    .equ         $;
MOV.B    @R1,R0L      ;Load Serial Transmitting Data
MOV.B    R0L,@SDRL    ;Save Serial Transmitting Data
;
BSET     STF          ;Start Serial Transmitting/Receiving
;
MAIN2    .equ         $
BTST     STF          ;End Serial Transmitting/Receiving ?
BNE     MAIN2        ;No.
;
BTST     ORER        ;Overrun Errorr Flag = 1 ?
BNE     MAIN3        ;Yes.
;
MOV.B    @SDRL,R0L   ;Load
MOV.B    R0L,@R2     ;Save
;
ADDS     #1,R1        ;Increment Serial Transmitting Data Address
ADDS     #1,R2        ;Increment Serial Receiving Data Address
DEC     R3L          ;Decrement Serial Data Counter
BNE     MAIN1        ;Serial Data Counter = H'00 ? No.
;
BRA     MAIN4
;
MAIN3    .equ         $
MOV.B    #H'FF,R0L
MOV.B    R0L,@SRD0    ;Overrun Errorr
MOV.B    R0L,@SRD1    ;Overrun Errorr
MOV.B    R0L,@SRD2    ;Overrun Errorr
MOV.B    R0L,@SRD3    ;Overrun Errorr
;
MAIN4    .equ         $
MOV.B    #H'00,R0L
MOV.B    R0L,@PMR3    ;Initialize S11 & SCK1 Pin Function
MOV.B    R0L,@SCR1    ;Initialize Synchronous Serial Transfer Function
;
MAIN9    .equ         $
BRA     MAIN9
;
.end

```

2.16 SSB通信

SSB通信	使用機能	SCI1 : SSB通信機能
-------	------	----------------

仕様

- (1) 図1に示すようにSSB通信機能を使用して、SCL (Serial Clock) とSDA (Serial Data) の2線で接続されたIC1、IC2を制御します。
- (2) 6.4 μ sの転送クロックで、16ビットのデータを送信します。
- (3) 16ビットデータの転送後に、TAIL MARKを付加して送信します。TAIL MARKにはHOLD TAILを選択します。

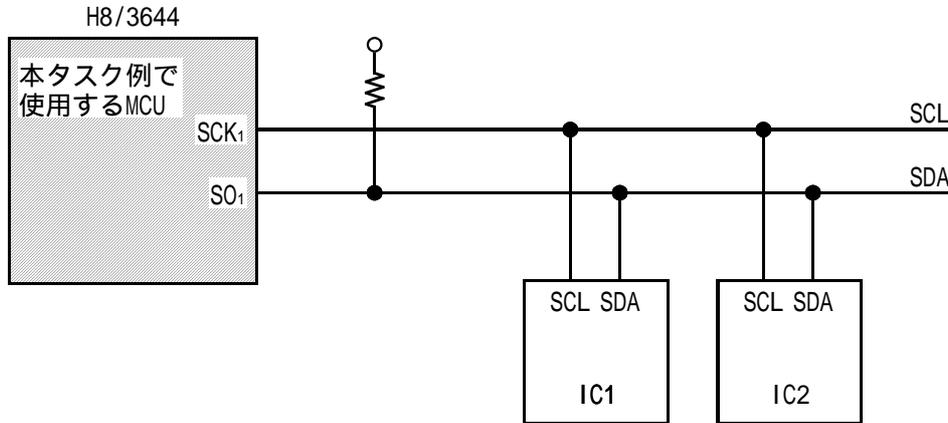


図1 SSB接続例

使用機能説明

- (1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、SSB通信を行ないます。図2にSSB通信のブロック図を示します。以下にSSB通信のブロック図について説明します。
 - SSB通信方式は、SCL (Serial Clock) とSDA (Serial Data) との2線で構成され、接続された複数のICを制御することができます。
 - SSBモードは8ビットまたは16ビットデータ転送後、TAIL MARKを付加して送信します。TAIL MARKはHOLD TAILとLATCH TAILから選択可能です。
 - システムクロック () は、10MHzのOSCクロックを2分周した5MHzのクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
 - プリスケアラS (PSS) は、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
 - シリアルコントロールレジスタ1 (SCR1) は、8ビットのリード/ライト可能なレジスタで、動作モード、転送クロックソースとプリスケアラ分周比を制御します。
 - シリアルコントロールステータスレジスタ1 (SCSR1) は、動作状態、エラー状態などを示す8ビットのカウンタです。
 - シリアルデータレジスタU (SDRU) は、8ビットのリード/ライト可能なレジスタで、16ビット転送時に上位8ビットのデータレジスタとして使用します。SDRUに書き込まれたデータは、SDRLにLSBファーストで出力されます。入れ替わりにSI₁端子よりLSBファーストでデータが入力されて、MSB LSB方向にデータがシフトします。SDRUのリード/ライトは、データの送信/受信が完了してから行なう必要があります。データの送信/受信中にリード/ライトを行なうとデータの内容は保証されません。
 - シリアルデータレジスタL (SDRL) は、8ビットのリード/ライト可能なレジスタで、8ビット転送時のデータレジスタ、および16ビット転送時の下位8ビットのデータレジスタとして使用します。8ビット転送時、SDRLに書き込まれたデータはSO₁端子よりLSBファーストで出力されます。入れ替わりにSI₁端子よりLSBファーストで入力されて、MSB LSB方向にデータがシフトします。16ビット転送時には、入力データがSDRUより取り込まれることを除けば8ビット転送時と同様の動作となります。SDRLのリード/ライトは、データの送信/受信が完了してから行なう必要があります。データの送信/受信中にリード/ライトを行なうとデータの内容は保証されません。
 - 転送クロックは、8種類の内部クロックと外部クロックから選択できますが、本LSIがクロック出力となるため外部クロックは選択できません。また転送レートは、SCR1のCKS2~CKS0で選択できますが、TAIL MARKの転送レートと兼ねているため、転送クロック周期が2 μ s以上となるように設定しなければなりません。
 - SCI1の転送フォーマットを図3に示します。データ転送は、データの最下位ビットから送信されるLSBファースト方式による転送を行ないます。8ビットまたは16ビットデータ転送後、TAIL MARKを付加します。

使用機能説明

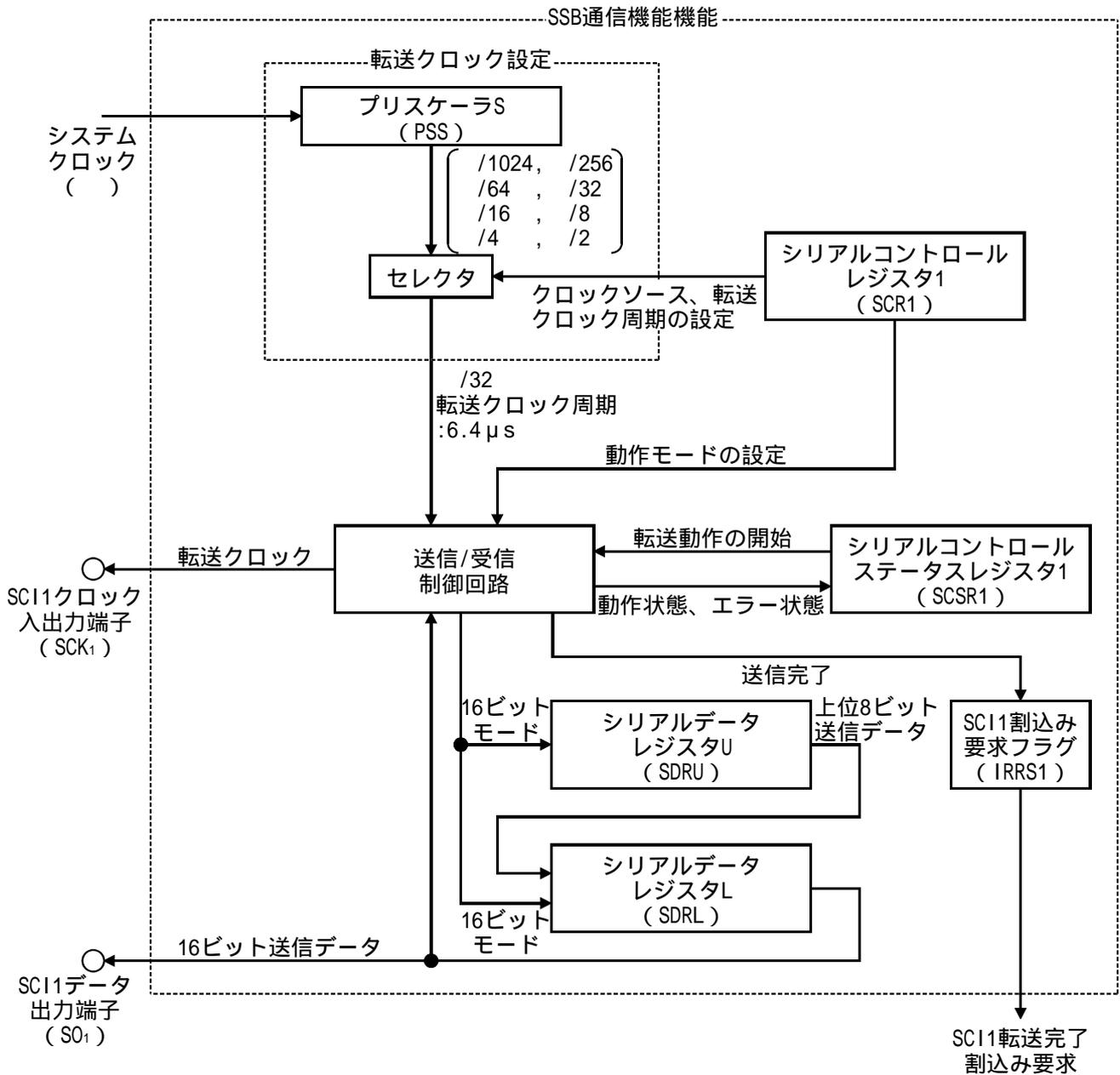


図2 SSB通信機能のブロック図

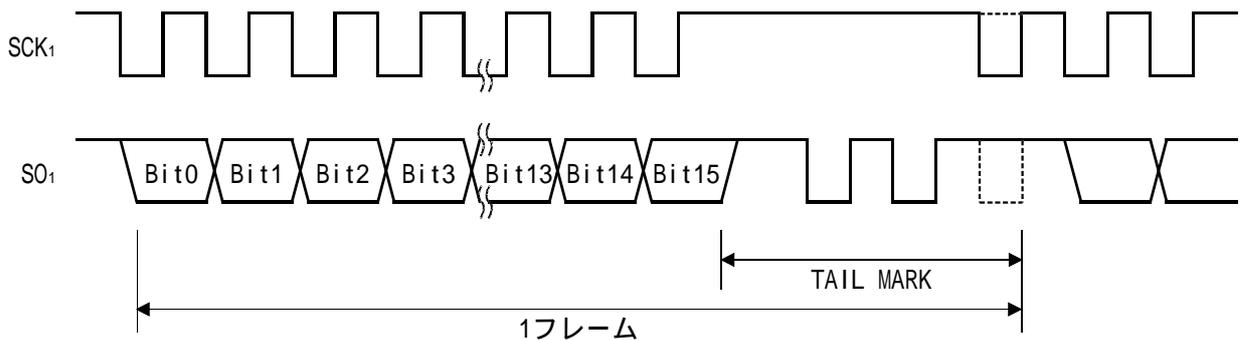


図3 転送フォーマット (SNC1="0"、SNCO="1"、MRKON="1"のとき)

使用機能説明

- ・TAIL MARKにはHOLD TAILとLATCH TAILがあります。HOLD TAILとLATCH TAILの出力波形を図4に示します。図4中の時間tはSCR1のCKS2～CKS0により設定された転送クロックにより決まる時間です。
- ・SC11の割込み要因には転送完了があります。SC11が転送を完了すると、IRR2のIRRS1が"1"にセットされます。SC11の割込み要求は、IENR2のIENS1により許可/禁止を選択できます。

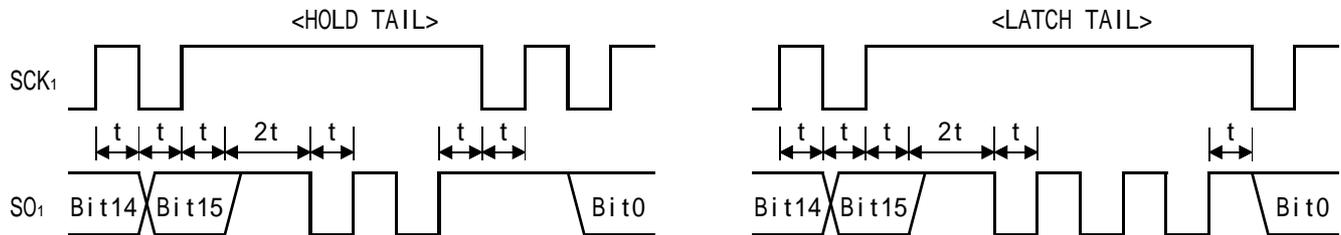


図4 HOLD TAILとLATCH TAILの出力波形

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、SSB通信を行いません。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
SCR1	動作モード、転送クロックソース、プリスケアラ分周比の設定
SCSR1	動作状態、エラー状態を示す
SDRU	16ビットの送信データの上位8ビットのデータレジスタ
SDRL	16ビットの送信データの下位8ビットのデータレジスタ
SCK ₁	SC11の転送クロック出力端子
SO ₁	SC11の送信データの出力端子
IRRS1	SC11の転送完了の有無を示す
IENS1	SC11割込み要求の許可/禁止を制御
PMR3	P3 ₂ /SO ₁ 、P3 ₀ /SCK ₁ 端子機能の設定
PMR7	P3 ₂ /SO ₁ 端子出力バッファのPMOSのON/OFFを制御

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理によりSSB通信を行ないます。

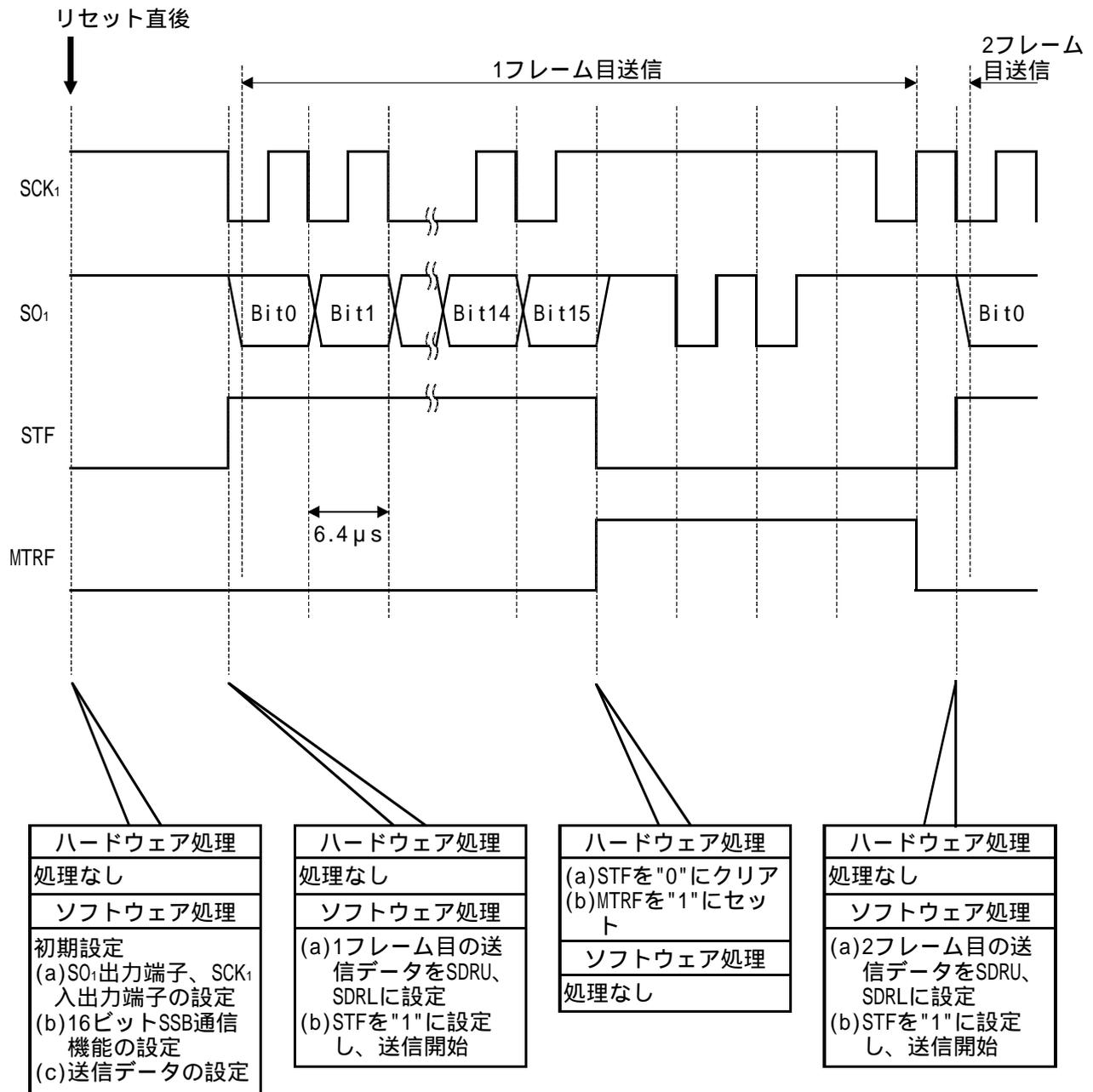


図3 SSB通信の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、転送データの設定、SSB通信モードの設定、割込みの許可、4フレームの16ビットのデータを送信したところで終了

(2) 引数の説明

表3に本タスク例で使用する引数を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0H ~ STD3H	16ビット送信データの上位8ビットを格納	メインルーチン	1バイト	入力
STD0L ~ STD3L	16ビット送信データの下位8ビットを格納	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR1	SNC1 SNC0	シリアルコントロールレジスタ1 (動作モード選択1、0) : SNC1="0"、SNC0="1"のとき、動作モードを16ビットのモードに設定	H' FFA0 ビット7 ビット6	SNC1="0" SNC0="1"
	MRKON	シリアルコントロールレジスタ1 (TAIL MARK制御) : MRKON="1"のとき、TAIL MARKを出力 (SSBモード)	H' FFA0 ビット5	1
	LATCH	シリアルコントロールレジスタ1 (LATCH TAIL選択) : LATCH="0"のとき、TAIL MARKとしてHOLD TAILを出力	H' FFA0 ビット4	0
	CKS3	シリアルコントロールレジスタ1 (クロックソース選択3) : CKS3="0"のとき、クロックソースをプリスケラSに、SCK ₁ 端子を出力に設定	H' FFA0 ビット3	0
	CKS2 CKS1 CKS0	シリアルコントロールレジスタ1 (クロック選択2、1、0) : CKS2="0"、CKS1="1"、CKS0="1"のとき、プリスケラ分周比を32分周に、転送クロック周期を6.4 μsに設定	H' FFA0 ビット2 ビット1 ビット0	CKS2="0" CKS1="1" CKS0="1"
SCSR1	SOL	シリアルコントロールステータスレジスタ1 (拡張データビット) : SOL="0"のとき、S0 ₁ 端子出力を"Low"レベルに変更 : SOL="1"のとき、S0 ₁ 端子出力を"High"レベルに変更	H' FFA1 ビット6	1
	MTRF	シリアルコントロールステータスレジスタ1 (TAIL MARK送信フラグ) : MTRF="0"のとき、転送待ち状態および8ビットまたは16ビットデータ転送中 : MTRF="1"のとき、TAIL MARK送信中	H' FFA1 ビット1	0
	STF	シリアルコントロールステータスレジスタ1 (スタートフラグ) : STF="0"のとき、転送動作の終了 : STF="1"のとき、転送動作の開始	H' FFA1 ビット0	0
SDRU	シリアルデータレジスタU : 16ビット転送時、送信データの上位8ビットを格納	H' FFA2	-	
SDRL	シリアルデータレジスタL : 16ビット転送時、送信データの下位8ビットを格納	H' FFA3	-	

ソフトウェア説明

表4 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
IENR2	IENS1	割込み許可レジスタ2 (SC11割込みイネーブル) : IENS1="0"のとき、SC11割込み要求を禁止	H'FFF5 ビット4	0
IRR2	IRRS1	割込み要求レジスタ2 (SC11割込み要求フラグ) : IRRS1="0"のとき、SC11割込みが要求されていない : IRRS1="1"のとき、SC11割込みが要求されている	H'FFF8 ビット4	0
PMR3	SO1	ポートモードレジスタ3 (P3 ₂ /SO ₁ 端子機能切り替え) : SO1="1"のとき、SO ₁ 出力端子に設定	H'FFFD ビット2	1
	SCK1	ポートモードレジスタ3 (P3 ₀ /SCK ₁ 端子機能切り替え) : SCK1="1"のとき、SCK ₁ 入出力端子として機能	H'FFFD ビット0	1
PMR7	POF1	ポートモードレジスタ3 (P3 ₂ /SO ₁ 端子PMOSコントロール) : POF1="1"のとき、NMOSオープンドレイン出力	H'FFFF ビット0	1

(4) 使用RAM説明

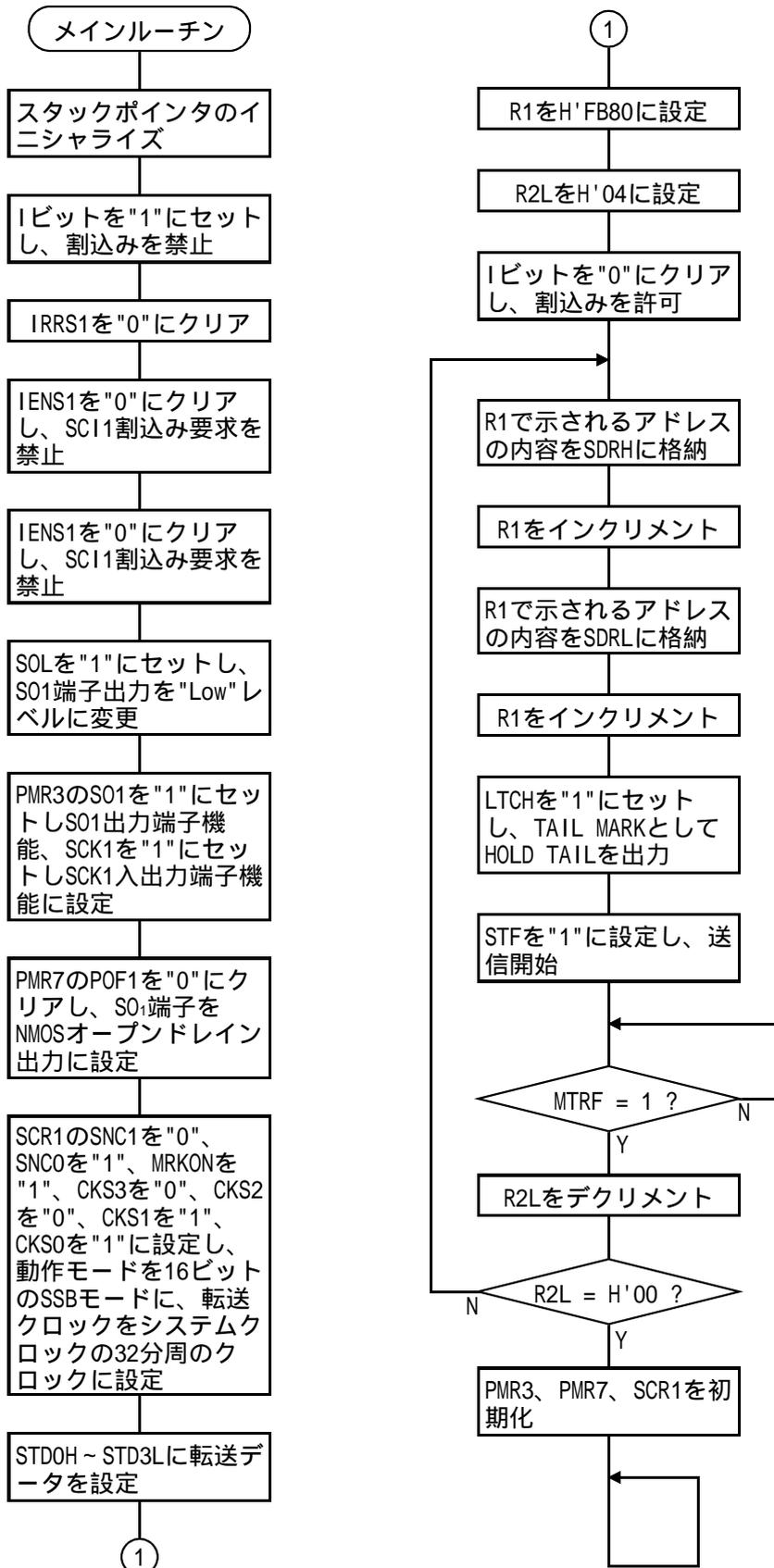
表5に本タスク例における使用RAM説明を示します。

表5 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0H	1フレーム目の16ビット送信データの上位8ビットを格納	H'FB80	メインルーチン
STD0L	1フレーム目の16ビット送信データの下位8ビットを格納	H'FB81	メインルーチン
STD1H	2フレーム目の16ビット送信データの上位8ビットを格納	H'FB82	メインルーチン
STD1L	2フレーム目の16ビット送信データの下位8ビットを格納	H'FB83	メインルーチン
STD2H	3フレーム目の16ビット送信データの上位8ビットを格納	H'FB84	メインルーチン
STD2L	3フレーム目の16ビット送信データの下位8ビットを格納	H'FB85	メインルーチン
STD3H	4フレーム目の16ビット送信データの上位8ビットを格納	H'FB86	メインルーチン
STD3L	4フレーム目の16ビット送信データの下位8ビットを格納	H'FB87	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'SSB Communications'
;
;       Function
;       : Serial Communication Interface
;       :   SSB Communication
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
SCR1      .equ        H'FFA0      ;Serial Control Register 1
SNC1      .bequ       7,SCR1      ;Select the Operation Mode 1
SNCO      .bequ       6,SCR1      ;Select the Operation Mode 0
MRKON     .bequ       5,SCR1      ;TAIL MARK Control
LTCH      .bequ       4,SCR1      ;LATCH TAIL Select
CKS3      .bequ       3,SCR1      ;Clock Source Slect 3
CKS2      .bequ       2,SCR1      ;Clock Slect 2
CKS1      .bequ       1,SCR1      ;Clock Slect 1
CKS0      .bequ       0,SCR1      ;Clock Slect 0
SCSR1     .equ        H'FFA1      ;Serial Control Status Register 1
SOL       .bequ       6,SCSR1     ;Extended Data Bit
ORER      .bequ       5,SCSR1     ;Overrun Errorr Flag
MTRF      .bequ       1,SCSR1     ;TAIL MARK Transmit Flag
STF       .bequ       0,SCSR1     ;Start Flag
SDRU      .equ        H'FFA2      ;Serial Data Register U
SDRL      .equ        H'FFA3      ;Serial Data Register L
IENR2     .equ        H'FFF5      ;Interrupt Enable Register 2
IENS1     .bequ       4,IENR2     ;SCI1 Interrupt Enable
IRR2      .equ        H'FFF8      ;Interrupt Request Register 2
IRRS1     .bequ       4,IRR2      ;SCI1 Interrupt Request Flag
PMR3      .equ        H'FFFD      ;Port Mode Register 3
S01       .bequ       2,PMR3      ;P32/S01 Pin Function Switch
SI1       .bequ       1,PMR3      ;P31/SI1 Pin Function Switch
SCK1      .bequ       0,PMR3      ;P30/SCK1 Pin Function Switch
PMR7      .equ        H'FFFF      ;Port Mode Register 7
POF1      .bequ       0,PMR7      ;P32/S01 Pin Function Switch
;
;
*****
;
;       RAM Allocation
;
*****
;
;
STACK     .equ        H'FF80      ;Stack Pointer
STD0H     .equ        H'FB80      ;Serial Transmitting Data 0 Upper
STD0L     .equ        H'FB81      ;Serial Transmitting Data 0 Lower
STD1H     .equ        H'FB82      ;Serial Transmitting Data 1 Upper

```

プログラムリスト

```

STD1L      .equ      H'FB83      ;Serial Transmitting Data 1 Lower
STD2H      .equ      H'FB84      ;Serial Transmitting Data 2 Upper
STD2L      .equ      H'FB85      ;Serial Transmitting Data 2 Lower
STD3H      .equ      H'FB86      ;Serial Transmitting Data 3 Upper
STD3L      .equ      H'FB87      ;Serial Transmitting Data 3 Lower
;
;
;*****
;
;          Vector Address
;*****
;
;
;          .org      H'0000
;          .data.w   MAIN          ;Reset Interrupt
;
;
;          .org      H'0008
;          .data.w   MAIN          ;IRQ0 Interrupt
;          .data.w   MAIN          ;IRQ1 Interrupt
;          .data.w   MAIN          ;IRQ2 Interrupt
;          .data.w   MAIN          ;IRQ3 Interrupt
;          .data.w   MAIN          ;INT0 - INT7 Interrupt
;
;
;          .org      H'0014
;          .data.w   MAIN          ;Timer A Interrupt
;          .data.w   MAIN          ;Timer B1 Interrupt
;
;
;          .org      H'0020
;          .data.w   MAIN          ;Timer X Interrupt
;          .data.w   MAIN          ;Timer V Interrupt
;
;
;          .org      H'0026
;          .data.w   MAIN          ;SCI1 Interrupt
;
;
;          .org      H'002A
;          .data.w   MAIN          ;SCI3 Interrupt
;          .data.w   MAIN          ;A/D Converter Interrupt
;          .data.w   MAIN          ;SLEEP Instruction Executed Interrupt
;
;
;*****
;
;          Main Program
;*****
;
;
;          .org      H'1000
;
;MAIN      .equ      $
;          MOV.W     #STACK,SP      ;Initialize Stack Pointer
;          ORC       #H'80,CCR      ;Interrupt Disable
;
;
;          BCLR     IRRS1          ;Clear IRRS1
;          BCLR     IENS1          ;SCI1 Interrupt Disable
;
;
;          BSET     SOL            ;Initialize S01 Terminal Output Level
;          MOV.B    #H'05,R0L      ;Initialize S01 CKS1 Terminal Function
;          MOV.B    R0L,@PMR3      ;Initialize S01 Terminal NMOS Open-Drain Output
;          BSET     POF1
;
;
;          MOV.B    #H'63,R0L      ;Initialize SSB Communication Function
;          MOV.B    R0L,@SCR1
;
;
;          MOV.W    #H'0001,R0     ;Set Serial Transmitting Data 1 Upper
;          MOV.B    R0H,@STD0H     ;Set Serial Transmitting Data 1 Lower
;          MOV.B    R0L,@STD0L

```

プログラムリスト

```

MOV.W      #H'0011,R0
MOV.B      R0H,@STD1H      ;Set Serial Transmitting Data 2 Upper
MOV.B      R0L,@STD1L      ;Set Serial Transmitting Data 2 Lower
MOV.W      #H'0111,R0
MOV.B      R0H,@STD2H      ;Set Serial Transmitting Data 3 Upper
MOV.B      R0L,@STD2L      ;Set Serial Transmitting Data 3 Lower
MOV.W      #H'1111,R0
MOV.B      R0H,@STD3H      ;Set Serial Transmitting Data 4 Upper
MOV.B      R0L,@STD3L      ;Set Serial Transmitting Data 4 Lower
;
;
MOV.W      #H'FB80,R1      ;Initialize Serial Transmitting Data Address
MOV.B      #H'04,R2L      ;Initialize Serial Transmitting Data Counter
;
MAIN1      .equ          $
MOV.B      @R1,R0H        ;Load Serial Transmitting Data Upper
MOV.B      R0H,@SDRU      ;Set Serial Transmitting Data Upper
ADDS      #1,R1          ;Increment Serial Transmitting Data Address
;
MOV.B      @R1,R0L        ;Load Serial Transmitting Data Lower
MOV.B      R0L,@SDRL      ;Set Serial Transmitting Data Lower
ADDS      #1,R1          ;Increment Serial Transmitting Data Address
;
BCLR      LTCH          ;Set HOLD TAIL
;
BSET      STF          ;Start Transmitting
;
MAIN2      .equ          $
BTST      MTRF          ;MTRF = "1" ?
BEQ      MAIN2          ;No.
;
DEC      R2L          ;Decrement Serial Transmitting Data Counter
BNE      MAIN1          ;Serial Transmitting Data Counter = H'00 ? No.
;
MOV.B      #H'00,R0L
MOV.B      R0L,@PMR3      ;Initialize S01 & SCK1 Terminal Function
BCLR      POF1          ;Initialize S01 Terminal Function
MOV.B      R0L,@SCR1      ;Initialize SCI1 Function
;
MAIN9      .equ          $
BRA      MAIN9
;
.end

```

2.17 調歩同期式シリアルデータ送信

調歩同期式シリアルデータ送信	使用機能	SCI3 : 調歩同期式シリアル転送機能
----------------	------	----------------------

仕様

- (1) 図1に示すように調歩同期式シリアル転送機能を使用して、4バイトの8ビットデータの送信を行います。
- (2) 送信データの通信フォーマットは、データ長が8ビット、奇数パリティ、ストップビット長が1ビットに設定します。
- (3) ビットレートは31250 (bit/s) で送信します。またデータ送信の終了時にブレークを出力します。

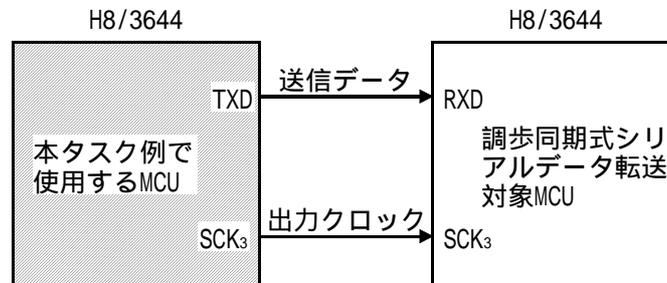
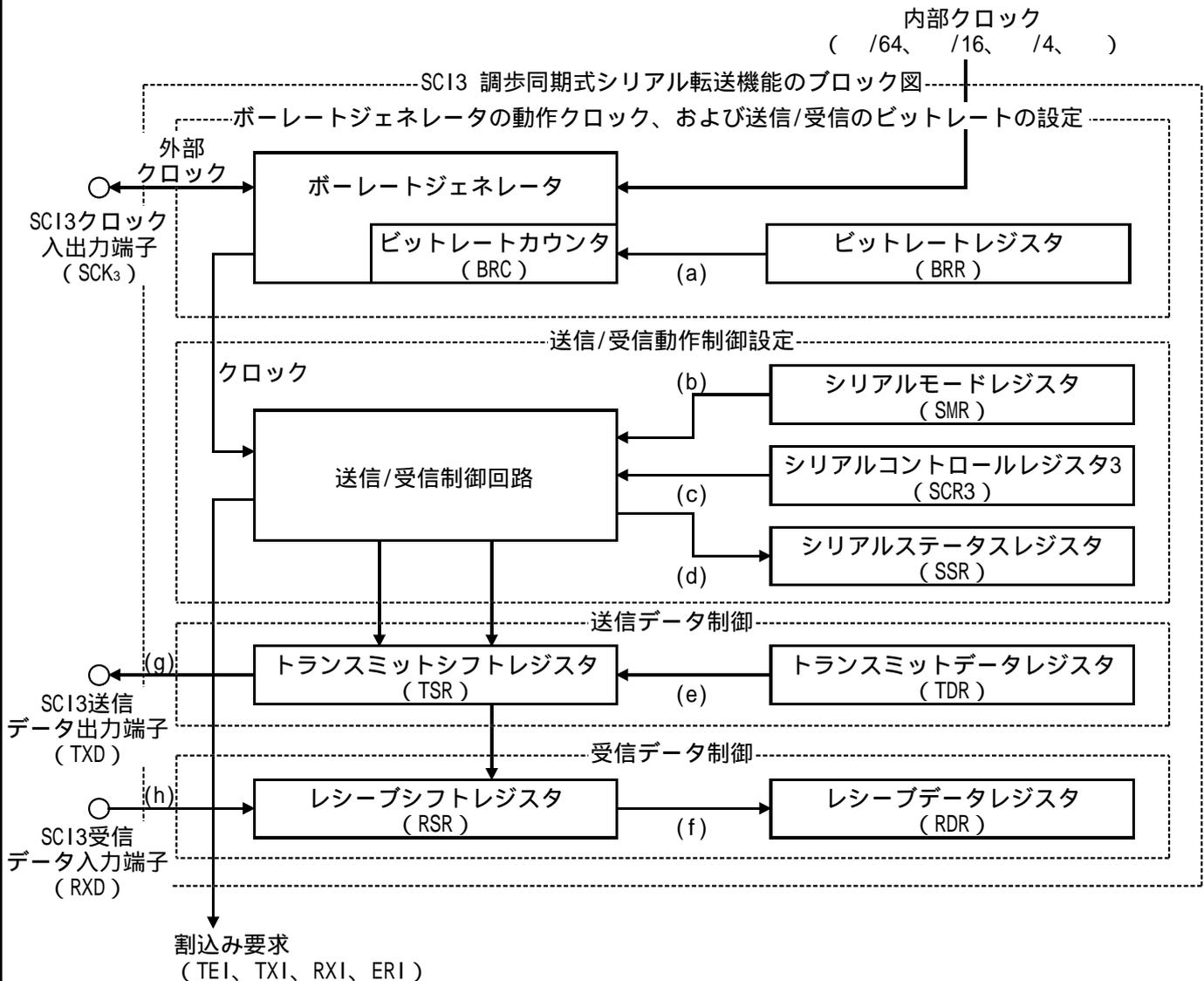


図1 調歩同期式シリアルデータ送信

使用機能説明

- (1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、調歩同期式のシリアルデータの送信を行います。図2に調歩同期式シリアルデータ送信のブロック図を示します。以下に調歩同期式シリアルデータ送信のブロック図について説明します。
 - ・調歩同期式モードは、キャラクタ単位で同期をとる調歩同期方式でシリアルデータ通信を行います。
 - ・Universal Asynchronous Receiver/Transmitter (UART) や、Asynchronous Communication Interface Adapter (ACIA) などの標準の調歩同期式通信用LSIとのシリアルデータ通信ができます。
 - ・複数のプロセッサとシリアル通信ができるマルチプロセッサ間通信機能を備えています。
 - ・通信フォーマットを12種類のフォーマットから選択できます。
 - ・独立した送信部と受信部を備えているので、送信と受信を同時に行なうことができます。また、送信部および受信部ともにダブルバッファ構造になっているため、連続送信・連続受信ができます。
 - ・内蔵のボーレートジェネレータで任意のビットレートを選択可能です。
 - ・送受信クロックソースを内部クロック、または外部クロックから選択可能です。
 - ・割り込み要因には送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割り込み要因があります。
 - ・レシーブシフトレジスタ (RSR) は、シリアルデータを受信するためのレジスタです。RSRにRXD端子から入力されたシリアルデータを、LSB (ビット0) から受信した順にセットしパラレルデータに変換します。1バイトのデータを受信すると、データは自動的にRDRへ転送されます。CPUからRSRを直接リード/ライトすることはできません。
 - ・レシーブデータレジスタ (RDR) は、受信したシリアルデータを格納する8ビットのレジスタです。1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送し、受信動作を完了します。その後、RSRは受信可能となります。RSRとRDRはダブルバッファになっているため連続した受信動作が可能です。RDRは受信専用レジスタなのでCPUからライトできません。
 - ・トランスミットシフトレジスタ (TSR) は、シリアルデータを送信するためのレジスタです。TDRから送信データをいったんTSRに転送し、LSB (ビット0) から順にTXD端子に送出することでシリアルデータ送信を行います。1バイトのデータを送信すると、自動的にTDRからTSRへ次の送信データを転送し、送信を開始します。ただし、TDRにデータが書き込まれていない (TDREに"1"がセットされている) 場合にはTDRからTSRへのデータ転送は行ないません。CPUからTSRを直接リード/ライトすることはできません。
 - ・トランスミットデータレジスタ (TDR) は、送信データを格納する8ビットのレジスタです。TSRの"空"を検出すると、TDRに書き込まれた送信データをTSRに転送し、シリアルデータ送信を開始します。TSRのシリアルデータ送信中に、TDRに次の送信データをライトしておく、連続送信が可能です。TDRは、常にCPUによるリード/ライトが可能です。
 - ・シリアルモードレジスタ (SMR) は、シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。SMRは、常にCPUによるリード/ライトが可能です。
 - ・シリアルコントロールレジスタ3 (SCR3) は、送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止、および送信/受信クロックソースの選択を行なう8ビットのレジスタです。SCR3は、常にCPUによるリード/ライトが可能です。

使用機能説明



- 【注】 (a) SMRで選択されるポレートジェネレータの動作クロックと合わせて、送信/受信のビットレートを設定します。本タスク例では、送信のビットレートを31250 (bit/s) に設定しています。
- (b) シリアルデータ通信フォーマットの設定と、ポレートジェネレータのクロックソースを選択します。本タスク例でシリアルデータ通信フォーマットは、動作モードを調歩同期式に、データ長を8ビットに、パリティビットあり、パリティモードを奇数パリティに、ストップビット長を1ビットに、内蔵ポレートジェネレータのクロックソースをクロックに設定しています。
- (c) 送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止を選択します。本タスク例では、調歩同期式モードでのクロック出力は、コミュニケーションモードを調歩同期式に、クロックソースを内部クロックに、SCK₃端子機能をクロック出力に設定しています。また、割り込み要求の許可/禁止は送信データエンpty割り込みを禁止に、受信データフル割り込み要求を禁止に設定しています。
- (d) ステータスフラグ (トランスミットデータレジスタエンpty、レシーブデータレジスタフル、オーバーランエラー、フレーミングエラー、パリティエラー、トランスミットエンド) によりSC13の動作状態を示す。
- (e) TSRの"空"を検出することにより、TDRに書き込まれた送信データをTSRに転送。
- (f) 1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送。
- (g) 送信データ。
- (h) 受信データ。

図2 調歩同期式シリアルデータ送信のブロック図

使用機能説明

- ・シリアルステータスレジスタ (SSR) は、SCI3の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8ビットのレジスタです。SSRは常にCPUからリード/ライトできます。ただし、TDRE、RDRF、OER、PER、FERへ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。また、TENDおよびMPBRはリード専用であり、ライトすることはできません。
- ・ビットレートレジスタ (BRR) は、SMRのCKS1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて、送信/受信のビットレートを設定する8ビットのレジスタです。BRRは常にCPUによるリード/ライトが可能です。
- ・表1に、調歩同期式モードのBRRの設定例を示します。表1はアクティブ (高速) モードで、OSCが10MHzのときの値を示しています。

表1 ビットレートに対するBRRの設定例 (調歩同期式モード)

Rビットレート (bit/s)	110	150	300	600	1200	2400	4800	9600	19200	31250	38400
n	2	2	1	0	0	0	0	0	0	0	0
N	88	64	129	64	129	64	32	15	7	4	3
誤差 (%)	-0.25	+0.16	+0.16	+0.16	+0.16	+0.16	-1.36	+1.73	+1.73	0	+1.73

- 【注】 1. 誤差は、1%以内となるように設定します。
2. BRRの設定値は以下の計算式で求められます。

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

B : ビットレート (bit/s)

N : ボーレートジェネレータのBRRの設定値 (0 N 255)

OSC : oscの値 (MHz) = 10MHz

n : ボーレートジェネレータの入力クロックのNo. (n=0,1,2,3)
(nとクロックの関係は表2を参照)

表2 nとクロックの関係

n	クロック	SMRの設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

3. 表1に誤差は以下の計算式で求めた値を小数点第3位を四捨五入して表示してあります。

$$\text{誤差 (\%)} = \frac{B(n, N, OSC \text{から求めたレート}) - R(\text{表1の上欄のビットレート})}{R(\text{表1の上欄のビットレート})} \times 100$$

4. OSCが10MHzのときの最大ビットレート (調歩同期式モード) は、156250 (bit/s) になります。ただし、設定値は、n=0、N=0のときです。

- ・調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信/受信し、1キャラクタ単位で同期をとりながらシリアル通信を行なうモードです。
- ・SCI3内部では、送信部と受信部は独立しているので、全二重通信を行なうことができます。また、送信部と受信部がともにダブルバッファ構造になっているので、送信中にデータのライト、受信中にデータのリードができ、連続送信/受信が可能です。
- ・図3に調歩同期式通信のデータフォーマットを示します。調歩同期式通信では、通信回線は通常マーク状態 ("High" レベル) に保たれています。SCI3では通信回線を監視し、スペース ("Low" レベル) になったところをスタートビットとみなしてシリアル通信を開始します。
- ・通信データの1キャラクタはスタートビット ("Low" レベル) から始まり、送信/受信データ (LSBファースト: 最下位ビットから)、パリティビット ("High" または "Low" レベル)、最後にストップビット ("High" レベル) の順で構成されます。
- ・調歩同期式モードでは、受信時にスタートビットの立ち上がりエッジで同期化を行いません。また、データを1ビット期間の16倍の周波数のクロックの8番目でサンプリングするので、各ビットの中央で通信データを取り込みます。

使用機能説明	
--------	--

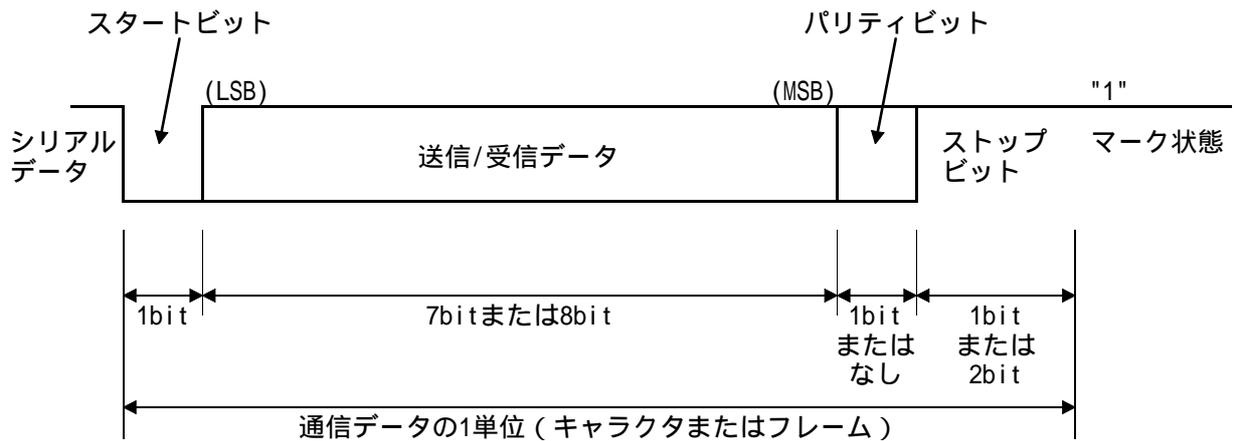


図3 調歩同期式通信のデータフォーマット

- ・ SCI3クロック (SCK₃) は、SCI3のクロック入出力端子です。
- ・ SCI3レシーブデータ入力 (RXD) は、SCI3の受信データ入力端子です。
- ・ SCI3トランスミットデータ出力 (TXD) は、SCI3の送信データ出力端子です。
- ・ SCI3の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび3種類の受信エラー (オーバーランエラー、フレーミングエラー、パリティエラー) の計6種類があり、共通のベクタアドレスが割り付けられています。
- ・ 各割込み要求は、SCR3のTIE、RIEで許可/禁止できます。
- ・ SSRのTDREが"1"にセットされるとTXIが発生します。SSRのTENDが"1"にセットされると、TEIが発生します。この2つの割込みは送信時に発生します。
- ・ SSRのTDREは初期値が"1"になっています。したがって送信データをTDRへ転送する前にSCR3のTIEを"1"にセットして送信データエンプティ割込み要求 (TXI) を許可すると、送信データが準備されていなくてもTXIが発生します。
- ・ SSRのTENDは初期値が"1"になっています。したがって、送信データをTDRへ転送する前にSCR3のTEIEを"1"にセットして送信終了割込み要求 (TEI) を許可すると、送信データが送信されていなくてもTEIが発生します。
- ・ 送信データをTDRへ転送する処理を割込み処理ルーチンの中で行なうようにすることで、これらの割込みを有効に利用できます。また、これらの割込み要求 (TXI、TEI) の発生を防ぐためには、送信データをTDRへ転送した後に、これらの割込み要求に対応する許可ビット (TIE、TEIE) を"1"にセットします。
- ・ SSRのRDRFが"1"にセットされるとRXIが発生します。OER、PER、FERのいずれかが"1"にセットされるとERIが発生します。この2つの割込み要求は受信時に発生します。

(2) 表3に本タスク例の機能割付け示します。表3に示すように機能を割り付け、調歩同期式シリアルデータ送信を行ないます。

表3 機能割付け

機能	機能割付け
TSR	シリアルデータを送信するためのレジスタ
TDR	送信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット、ボーレートジェネレータのクロックソースの設定
SSR	SCI3の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
TXI	送信データエンプティ (TDRE) による割込み要求
TEI	送信終了 (TEND) による割込み要求
PMR7	TXD出力端子設定
SCK ₃	SCI3のクロック出力端子
TXD	SCI3の送信データ出力端子

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理により調歩同期式シリアルデータ送信を行ないます。

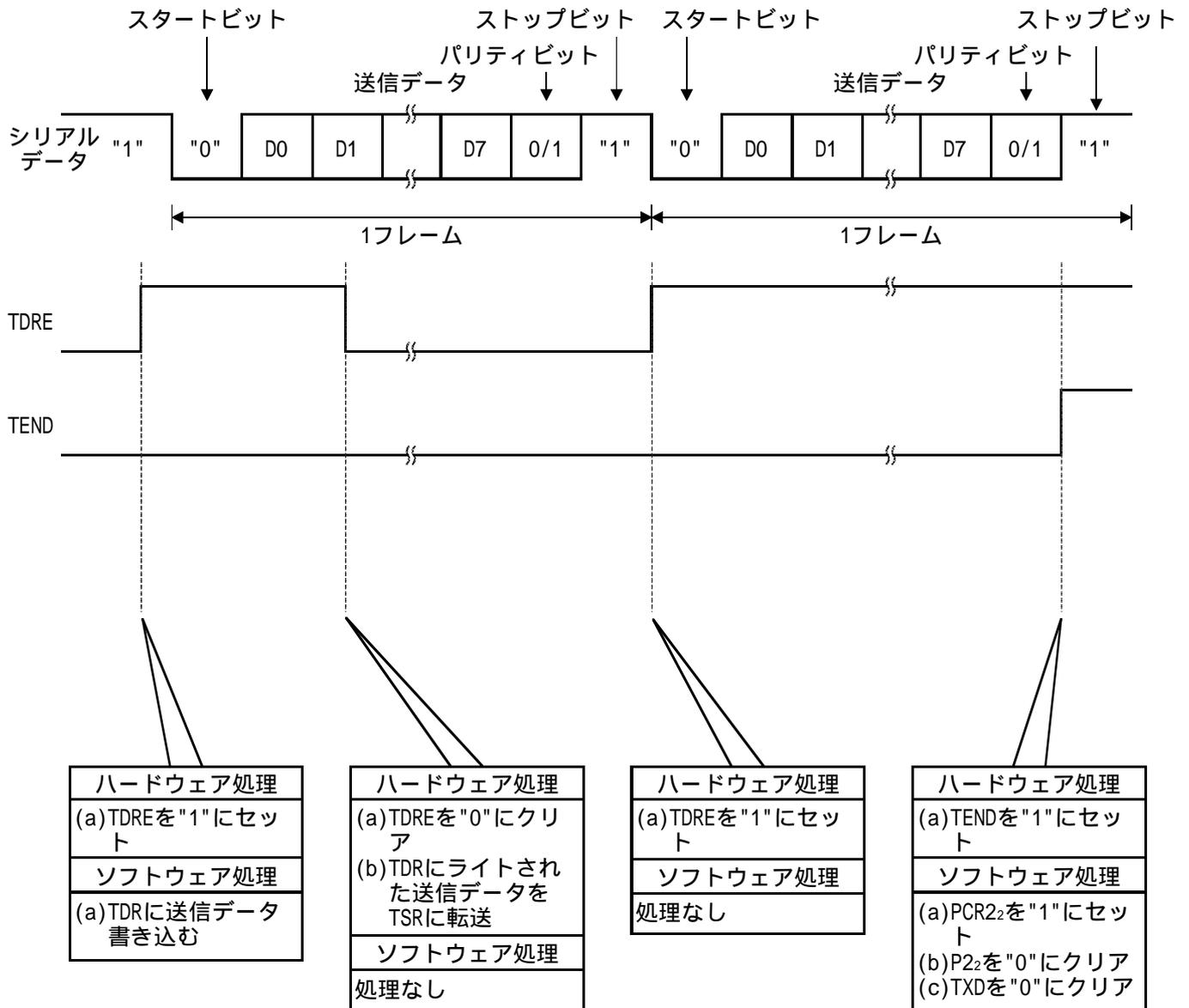


図4 調歩同期式シリアルデータ送信の動作原理

ソフトウェア説明

(1) モジュール説明

表4に本タスク例におけるモジュール説明を示します。

表4 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、転送データの設定、調歩同期式シリアルデータ送信の設定、割込みの許可、4バイトのデータを送信したところで終了

(2) 引数の説明

表5に本タスク例で使用する引数を示します。

表5 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0 ~ STD3	調歩同期式シリアル送信データ	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表6に本タスク例における使用内部レジスタ説明を示します。

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
SMR	COM シリアルモードレジスタ(コミュニケーションモード) : COM="0"のとき、コミュニケーションモードを調歩同期式モードに設定	H'FFA8 ビット7	0
	CHR シリアルモードレジスタ(キャラクタレングス) : CHR="0"のとき、調歩同期式モード時におけるデータ長を8ビットデータに設定	H'FFA8 ビット6	0
	PE シリアルモードレジスタ(パリティイネーブル) : PE="1"のとき、調歩同期式モードで、送信時にパリティビットの付加およびチェックを許可	H'FFA8 ビット5	1
	PM シリアルモードレジスタ(パリティモード) : PM="1"のとき、パリティの付加やチェックを奇数パリティに設定	H'FFA8 ビット4	1
	STOP シリアルモードレジスタ(ストップビットレングス) : STOP="0"のとき、調歩同期式モードでのストップビットの長さを1ビットに設定	H'FFA8 ビット3	0
	MP シリアルモードレジスタ(マルチプロセッサモード) : MP="0"のとき、マルチプロセッサ通信機能を禁止	H'FFA8 ビット2	0
	CKS1 CKS0 シリアルモードレジスタ(クロックセレクト1、0) : CKS1="0"、CKS0="0"のとき、内蔵ボーレートジェネレータのクロックソースをクロックに設定	H'FFA8 ビット1 ビット0	CKS1="0" CKS0="0"
BRR	ビットレートレジスタ : BRR=H'04のとき、SMRのCKS1、CLS0で選択されるボーレートジェネレータの動作クロックとあわせて送信のビットレートを31250 (bit/s) に設定	H'FFA9	H'04
SCR3 TIE	シリアルコントロールレジスタ3 (トランスミットインタラプトイネーブル) : TIE="0"のとき、送信データエンプティ割込み要求(TXI)を禁止	H'FFAA ビット7	0

ソフトウェア説明

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR3	TE	シリアルコントロールレジスタ3 (トランスミットイネーブル) : TE="0"のとき、送信動作を禁止 (TXD端子はトランスミットデータ端子)	H'FFAA ビット5	0
	TEIE	シリアルコントロールレジスタ3 (トランスミットエンドインタラプトイネーブル) : TEIE="0"のとき、送信終了割り込み要求を禁止	H'FFAA ビット2	0
	CKE1 CKE0	シリアルコントロールレジスタ3 (クロックイネーブル1、0) : CKE1="0"、CKE0="1"のとき、調歩同期式モードにおいてクロックソースを内部クロック、SCK ₃ 端子機能をクロック出力に設定	H'FFAA ビット1 ビット0	CKE1="0" CKE0="1"
TDR	トランスミットデータレジスタ : 送信データを格納する8ビットのレジスタ	H'FFAB	-	
SSR	TDRE	シリアルステータスレジスタ (トランスミットデータエンプティ) : TDRE="0"のとき、TDRにライトされた送信データがTSRに転送されていないことを示す : TSRE="1"のとき、TDRに送信データがライトされていない、またはTDRにライトされた送信データがTSRに転送されたことを示す	H'FFAC ビット7	1
	TEND	シリアルステータスレジスタ (トランスミットエンド) : TEND="0"のとき、送信中であることを示す : TEND="1"のとき、送信を終了したことを示す	H'FFAC ビット2	1
PDR2	P2 ₂	ポートデータレジスタ2 (ポートデータレジスタ22) : P2 ₂ ="0"のとき、P2 ₂ 端子の出力レベルは"Low" : P2 ₂ ="1"のとき、P2 ₂ 端子の出力レベルは"High"	H'FFD5 ビット2	0
PCR2	PCR2 ₂	ポートコントロールレジスタ2 (ポートコントロールレジスタ22) : PCR2 ₂ ="0"のとき、P2 ₂ 端子を入力端子機能に設定 : PCR2 ₂ ="1"のとき、P2 ₂ 端子を出力端子機能に設定	H'FFE5 ビット2	1
PMR7	TXD	ポートモードレジスタ7 (P2 ₂ /TXD端子機能切り替え) : TXD="1"のとき、P2 ₂ /TXD端子をTXD端子機能に設定	H'FFFF ビット2	1

(4) 使用RAM説明

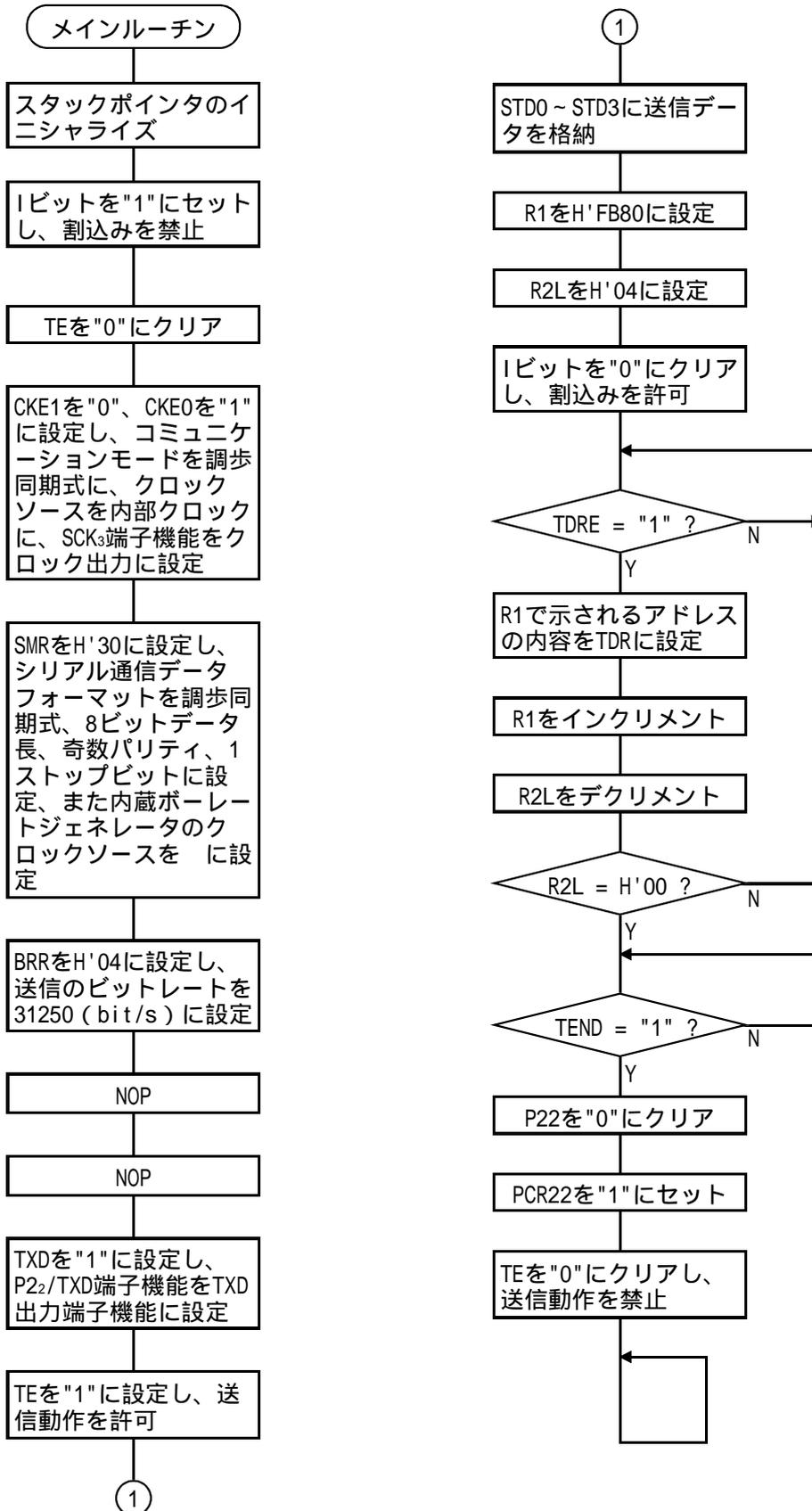
表7に本タスク例における使用RAM説明を示します。

表7 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0	調歩同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	調歩同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	調歩同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	調歩同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Asynchronous Serial Data Transmission'
;
;           Function
;           : Serial Communication Interface
;           Asynchronous Serial Interface
;           -Transmitting
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
SMR           .equ           H'FFA8           ;Serial Mode Register
COM           .bequ          7,SMR           ;Communication Mode
CHR           .bequ          6,SMR           ;Character Length
PE           .bequ          5,SMR           ;Parity Enable
PM           .bequ          4,SMR           ;Parity Mode
STOP         .bequ          3,SMR           ;Stop Bit Length
MP           .bequ          2,SMR           ;Multiprocessor Mode
CKS1         .bequ          1,SMR           ;Clock Select 1
CKS0         .bequ          0,SMR           ;Clock Select 0
BRR           .equ           H'FFA9           ;Bit Rate Register
SCR3         .equ           H'FFAA           ;Serial Control Register 3
TIE          .bequ          7,SCR3          ;Transmit Interrupt Enable
RIE          .bequ          6,SCR3          ;Receive Interrupt Enable
TE           .bequ          5,SCR3          ;Transmit Enable
RE           .bequ          4,SCR3          ;Receive Enable
MPIE         .bequ          3,SCR3          ;Multiprocessor Interrupt Enable
TEIE         .bequ          2,SCR3          ;Transmit End Interrupt Enable
CKE1         .bequ          1,SCR3          ;Clock Enable 1
CKE0         .bequ          0,SCR3          ;Clock Enable 0
TDR           .equ           H'FFAB           ;Transmit Data Register
SSR           .equ           H'FFAC           ;Serial Status Register
TDRE         .bequ          7,SSR           ;Transmit Data Register Empty
RDRF         .bequ          6,SSR           ;Receive Data Register Full
OER          .bequ          5,SSR           ;Overrun Error
FER          .bequ          4,SSR           ;Framing Error
PER          .bequ          3,SSR           ;Parity Error
TEND         .bequ          2,SSR           ;Transmit End
MPBR         .bequ          1,SSR           ;Multiprocessor Bit Receive
MPBT         .bequ          0,SSR           ;Multiprocessor Bit Transfer
RDR           .equ           H'FFAD           ;Receive Data Register
PDR2         .equ           H'FFD5           ;Port Data Register 2
P22          .bequ          2,PDR2          ;Port Data Register 22
PCR2         .equ           H'FFE5           ;Port Control Register 2
PCR22        .bequ          2,PCR2          ;Port Control Register 22

```

プログラムリスト

```

PMR7      .equ          H'FFFF      ;Port Mode Register 7
TXD       .bequ        2,PMR7      ;P22/TXD Terminal Function Switch
;
;
;*****
;
;          RAM Allocation
;*****
;
;
STACK     .equ          H'FF80      ;Stack Pointer
STD0      .equ          H'FB80      ;Serial Transmitting Data 0
STD1      .equ          H'FB81      ;Serial Transmitting Data 1
STD2      .equ          H'FB82      ;Serial Transmitting Data 2
STD3      .equ          H'FB83      ;Serial Transmitting Data 3
;
;
;*****
;
;          Vector Address
;*****
;
;          .org          H'0000
;          .data.w      MAIN        ;Reset Interrupt
;
;          .org          H'0008
;          .data.w      MAIN        ;IRQ0 Interrupt
;          .data.w      MAIN        ;IRQ1 Interrupt
;          .data.w      MAIN        ;IRQ2 Interrupt
;          .data.w      MAIN        ;IRQ3 Interrupt
;          .data.w      MAIN        ;INT0 - INT7 Interrupt
;
;          .org          H'0014
;          .data.w      MAIN        ;Timer A Interrupt
;          .data.w      MAIN        ;Timer B1 Interrupt
;
;          .org          H'0020
;          .data.w      MAIN        ;Timer X Interrupt
;          .data.w      MAIN        ;Timer V Interrupt
;
;          .org          H'0026
;          .data.w      MAIN        ;SCI1 Interrupt
;
;          .org          H'002A
;          .data.w      MAIN        ;SCI3 Interrupt
;          .data.w      MAIN        ;A/D Converter Interrupt
;          .data.w      MAIN        ;SLEEP Instruction Executed Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org          H'1000
;
;MAIN     .equ          $
;          MOV.W        #STACK,SP   ;Initialize Stack Pointer
;          ORC          #H'80,CCR    ;Interrupt Disable
;
;          MOV.W        #H'0130,R0
;          MOV.B        ROH,@SCR3   ;Clear TE & Initialize Clock Source
;          MOV.B        ROL,@SMR    ;Initialize Transmit Format
;
;          MOV.B        #H'04,R0L
;          MOV.B        ROL,@BRR    ;Initialize Transmit Bit Rate
;
;
;

```

プログラムリスト

```

NOP                                ;1bit Wait
NOP
;
BSET    TXD                        ;Initialize TXD Terminal Function
BSET    TE                          ;Transmit Enable
;
MOV.W   #H'0055,R0
MOV.B   R0H,@STD0                  ;Set Serial Transmitting Data 1
MOV.B   R0L,@STD1                  ;Set Serial Transmitting Data 2
MOV.W   #H'AAFF,R0
MOV.B   R0H,@STD2                  ;Set Serial Transmitting Data 3
MOV.B   R0L,@STD3                  ;Set Serial Transmitting Data 4
;
MOV.W   #H'FB80,R1                  ;Initialize Serial Transmitting Data Address
MOV.B   #H'04,R2L                   ;Initialize Serial Transmitting Data Counter
;
ANDC    #H'7F,CCR                   ;Interrupt Enable
;
MAIN1   .equ    $
BTST    TDRE                        ;TDRE = "1" ?
BEQ     MAIN1                       ;No.
;
MOV.B   @R1,R0L
MOV.B   R0L,@TDR                    ;Write Serial Transmit Data to TDR
;
ADDS    #1,R1                       ;Increment Serial Transmitting Data Address
DEC     R2L                          ;Decrement Serial Transmitting Data Counter
BNE     MAIN1                       ;Transmit End ? No.
;
MAIN2   .equ    $
BTST    TEND                        ;TEND = "1" ?
BEQ     MAIN2                       ;No.
;
BCLR    P22                         ;Clear P22
BSET    PCR22                       ;Set PCR22
BCLR    TXD                          ;Set P22 Terminal Function
;
BCLR    TE                          ;Clear TE
;
MAIN9   .equ    $
BRA     MAIN9
;
.end

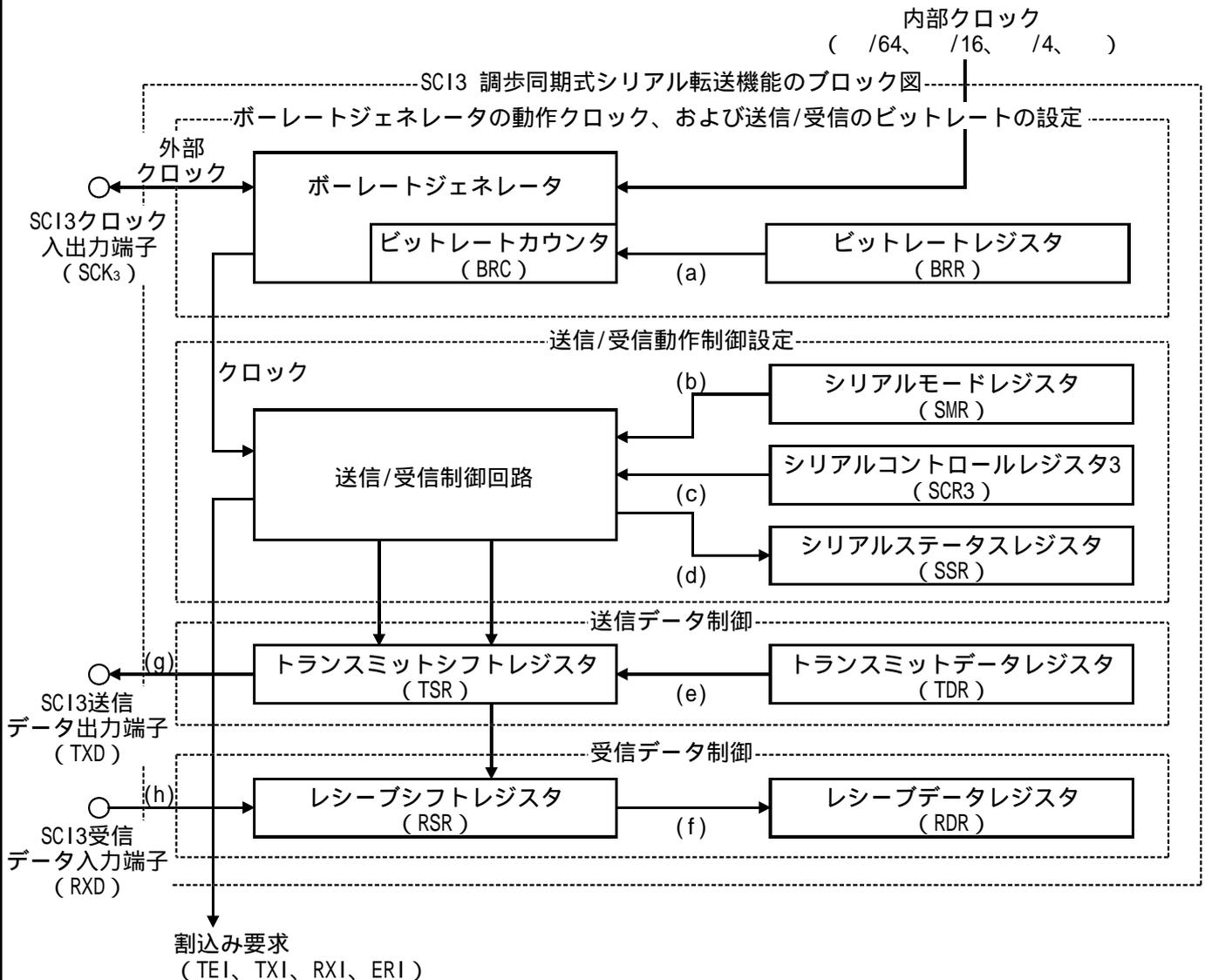
```

2.18 調歩同期式シリアルデータ受信

調歩同期式シリアルデータ受信	使用機能	SCI3 : 調歩同期式シリアル転送機能
仕様		
<p>(1) 図1に示すように調歩同期式シリアル転送機能を使用して、4バイトの8ビットデータの受信を行います。</p> <p>(2) 送信データの通信フォーマットは、データ長が8ビット、奇数パリティ、ストップビット長が1ビットに設定します。</p> <p>(3) ビットレートは31250 (bit/s) で送信します。4バイトのデータを受信すると終了します。</p>		
図1 調歩同期式シリアルデータ受信		

使用機能説明		
<p>(1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、調歩同期式のシリアルデータの受信を行います。図2に調歩同期式シリアルデータ受信のブロック図を示します。以下に調歩同期式シリアルデータ受信のブロック図について説明します。</p> <ul style="list-style-type: none"> ・調歩同期式モードは、キャラクタ単位で同期をとる調歩同期方式でシリアルデータ通信を行います。 ・Universal Asynchronous Receiver/Transmitter (UART) や、Asynchronous Communication Interface Adapter (ACIA) などの標準の調歩同期式通信用LSIとのシリアルデータ通信ができます。 ・複数のプロセッサとシリアル通信ができるマルチプロセッサ間通信機能を備えています。 ・通信フォーマットを12種類のフォーマットから選択できます。 ・独立した送信部と受信部を備えているので、送信と受信を同時に行なうことができます。また、送信部および受信部ともにダブルバッファ構造になっているため、連続送信・連続受信ができます。 ・内蔵のボーレートジェネレータで任意のビットレートを選択可能です。 ・送受信クロックソースを内部クロック、または外部クロックから選択可能です。 ・割り込み要因には送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割り込み要因があります。 ・レシーブシフトレジスタ (RSR) は、シリアルデータを受信するためのレジスタです。RSRにRXD端子から入力されたシリアルデータを、LSB (ビット0) から受信した順にセットしパラレルデータに変換します。1バイトのデータを受信すると、データは自動的にRDRへ転送されます。CPUからRSRを直接リード/ライトすることはできません。 ・レシーブデータレジスタ (RDR) は、受信したシリアルデータを格納する8ビットのレジスタです。1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送し、受信動作を完了します。その後、RSRは受信可能となります。RSRとRDRはダブルバッファになっているため連続した受信動作が可能です。RDRは受信専用レジスタなのでCPUからライトできません。 ・トランスミットシフトレジスタ (TSR) は、シリアルデータを送信するためのレジスタです。TDRから送信データをいったんTSRに転送し、LSB (ビット0) から順にTXD端子に送出することでシリアルデータ送信を行ないます。1バイトのデータを送信すると、自動的にTDRからTSRへ次の送信データを転送し、送信を開始します。ただし、TDRにデータが書き込まれていない (TDREに"1"がセットされている) 場合にはTDRからTSRへのデータ転送は行ないません。CPUからTSRを直接リード/ライトすることはできません。 ・トランスミットデータレジスタ (TDR) は、送信データを格納する8ビットのレジスタです。TSRの"空"を検出すると、TDRに書き込まれた送信データをTSRに転送し、シリアルデータ送信を開始します。TSRのシリアルデータ送信中に、TDRに次の送信データをライトしておく、連続送信が可能です。TDRは、常にCPUによるリード/ライトが可能です。 ・シリアルモードレジスタ (SMR) は、シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。SMRは、常にCPUによるリード/ライトが可能です。 ・シリアルコントロールレジスタ3 (SCR3) は、送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止、および送信/受信クロックソースの選択を行なう8ビットのレジスタです。SCR3は、常にCPUによるリード/ライトが可能です。 		

使用機能説明



- 【注】 (a) SMRで選択されるポーレートジェネレータの動作クロックと合わせて、送信/受信のビットレートを設定します。本タスク例では、送信のビットレートを31250 (bit/s) に設定しています。
- (b) シリアルデータ通信フォーマットの設定と、ポーレートジェネレータのクロックソースを選択します。本タスク例でシリアルデータ通信フォーマットは、動作モードを調歩同期式に、データ長を8ビットに、パリティビットあり、パリティモードを奇数パリティに、ストップビット長を1ビットに、内蔵ポーレートジェネレータのクロックソースをクロックに設定しています。
- (c) 送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止を選択します。本タスク例では、調歩同期式モードでのクロック出力は、コミュニケーションモードを調歩同期式に、クロックソースを内部クロックに、SCK₃端子機能を入出力ポートに設定しています。また、割り込み要求の許可/禁止は送信データエンpty割り込み要求を禁止、受信データフル割り込み要求を禁止、受信エラー割り込み要求を禁止に設定しています。
- (d) ステータスフラグ (トランスミットデータレジスタエンpty、レシーブデータレジスタフル、オーバーランエラー、フレーミングエラー、パリティエラー、トランスミットエンド) によりSC13の動作状態を示す。
- (e) TSRの"空"を検出することにより、TDRに書き込まれた送信データをTSRに転送。
- (f) 1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送。
- (g) 送信データ。
- (h) 受信データ。

図2 調歩同期式シリアルデータ受信のブロック図

使用機能説明

- ・シリアルステータスレジスタ (SSR) は、SCI3の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8ビットのレジスタです。SSRは常にCPUからリード/ライトできます。ただし、TDRE、RDRF、OER、PER、FERへ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。また、TENDおよびMPBRはリード専用であり、ライトすることはできません。
- ・ビットレートレジスタ (BRR) は、SMRのCKS1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて、送信/受信のビットレートを設定する8ビットのレジスタです。BRRは常にCPUによるリード/ライトが可能です。
- ・表1に、調歩同期式モードのBRRの設定例を示します。表1はアクティブ (高速) モードで、OSCが10MHzのときの値を示しています。

表1 ビットレートに対するBRRの設定例 (調歩同期式モード)

Rビットレート (bit/s)	110	150	300	600	1200	2400	4800	9600	19200	31250	38400
n	2	2	1	0	0	0	0	0	0	0	0
N	88	64	129	64	129	64	32	15	7	4	3
誤差 (%)	-0.25	+0.16	+0.16	+0.16	+0.16	+0.16	-1.36	+1.73	+1.73	0	+1.73

- 【注】 1. 誤差は、1%以内となるように設定します。
2. BRRの設定値は以下の計算式で求められます。

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

B : ビットレート (bit/s)

N : ボーレートジェネレータのBRRの設定値 (0 N 255)

OSC : oscの値 (MHz) = 10MHz

n : ボーレートジェネレータの入力クロックのNo. (n=0, 1, 2, 3)
(nとクロックの関係は表2を参照)

表2 nとクロックの関係

n	クロック	SMRの設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

3. 表1に誤差は以下の計算式で求めた値を小数点第3位を四捨五入して表示してあります。

$$\text{誤差 (\%)} = \frac{B(n, N, OSC \text{から求めたレート}) - R(\text{表1の上欄のビットレート})}{R(\text{表1の上欄のビットレート})} \times 100$$

4. OSCが10MHzのときの最大ビットレート (調歩同期式モード) は、156250 (bit/s) になります。ただし、設定値は、n=0、N=0のときです。

- ・調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信/受信し、1キャラクタ単位で同期をとりながらシリアル通信を行なうモードです。
- ・SCI3内部では、送信部と受信部は独立しているので、全二重通信を行なうことができます。また、送信部と受信部がともにダブルバッファ構造になっているので、送信中にデータのライト、受信中にデータのリードができ、連続送信/受信が可能です。
- ・図3に調歩同期式通信のデータフォーマットを示します。調歩同期式通信では、通信回線は通常マーク状態 ("High" レベル) に保たれています。SCI3では通信回線を監視し、スペース ("Low" レベル) になったところをスタートビットとみなしてシリアル通信を開始します。
- ・通信データの1キャラクタはスタートビット ("Low" レベル) から始まり、送信/受信データ (LSBファースト: 最下位ビットから)、パリティビット ("High" または "Low" レベル)、最後にストップビット ("High" レベル) の順で構成されます。
- ・調歩同期式モードでは、受信時にスタートビットの立ち上がりエッジで同期化を行いません。また、データを1ビット期間の16倍の周波数のクロックの8番目でサンプリングするので、各ビットの中央で通信データを取り込みます。

使用機能説明



図3 調歩同期式通信のデータフォーマット

- SCI3クロック (SCK₃) は、SCI3のクロック入出力端子です。
- SCI3レシーブデータ入力 (RXD) は、SCI3の受信データ入力端子です。
- SCI3トランスミットデータ出力 (TXD) は、SCI3の送信データ出力端子です。
- SCI3の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび3種類の受信エラー (オーバーランエラー、フレーミングエラー、パリティエラー) の計6種類があり、共通のベクタアドレスが割り付けられています。
- 各割込み要求は、SCR3のTIE、RIEで許可/禁止できます。
- SSRのTDREが"1"にセットされるとTXIが発生します。SSRのTENDが"1"にセットされると、TEIが発生します。この2つの割込みは送信時に発生します。
- SSRのTDREは初期値が"1"になっています。したがって送信データをTDRへ転送する前にSCR3のTIEを"1"にセットして送信データエンプティ割込み要求 (TXI) を許可すると、送信データが準備されていなくてもTXIが発生します。
- SSRのTENDは初期値が"1"になっています。したがって、送信データをTDRへ転送する前にSCR3のTEIEを"1"にセットして送信終了割込み要求 (TEI) を許可すると、送信データが送信されていなくてもTEIが発生します。
- 送信データをTDRへ転送する処理を割込み処理ルーチンの中で行なうようにすることで、これらの割込みを有効に利用できます。また、これらの割込み要求 (TXI、TEI) の発生を防ぐためには、送信データをTDRへ転送した後に、これらの割込み要求に対応する許可ビット (TIE、TEIE) を"1"にセットします。
- SSRのRDRFが"1"にセットされるとRXIが発生します。OER、PER、FERのいずれかが"1"にセットされるとERIが発生します。この2つの割込み要求は受信時に発生します。

(2) 表3に本タスク例の機能割付けを示します。表3に示すように機能を割り付け、調歩同期式シリアルデータ受信を行ないます。

表3 機能割付け

機能	機能割付け
RSR	シリアルデータを受信するためのレジスタ
RDR	受信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット、ボーレートジェネレータのクロックソースの設定
SSR	SCI3の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
RXI	受信データフル (RDRF) による割込み要求
ERI	受信エラー (OER、FER、PER) による割込み要求
SCK ₃	入出力ポート
RXD	SCI3の受信データ入力端子

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理により調歩同期式シリアルデータ受信を行ないます。

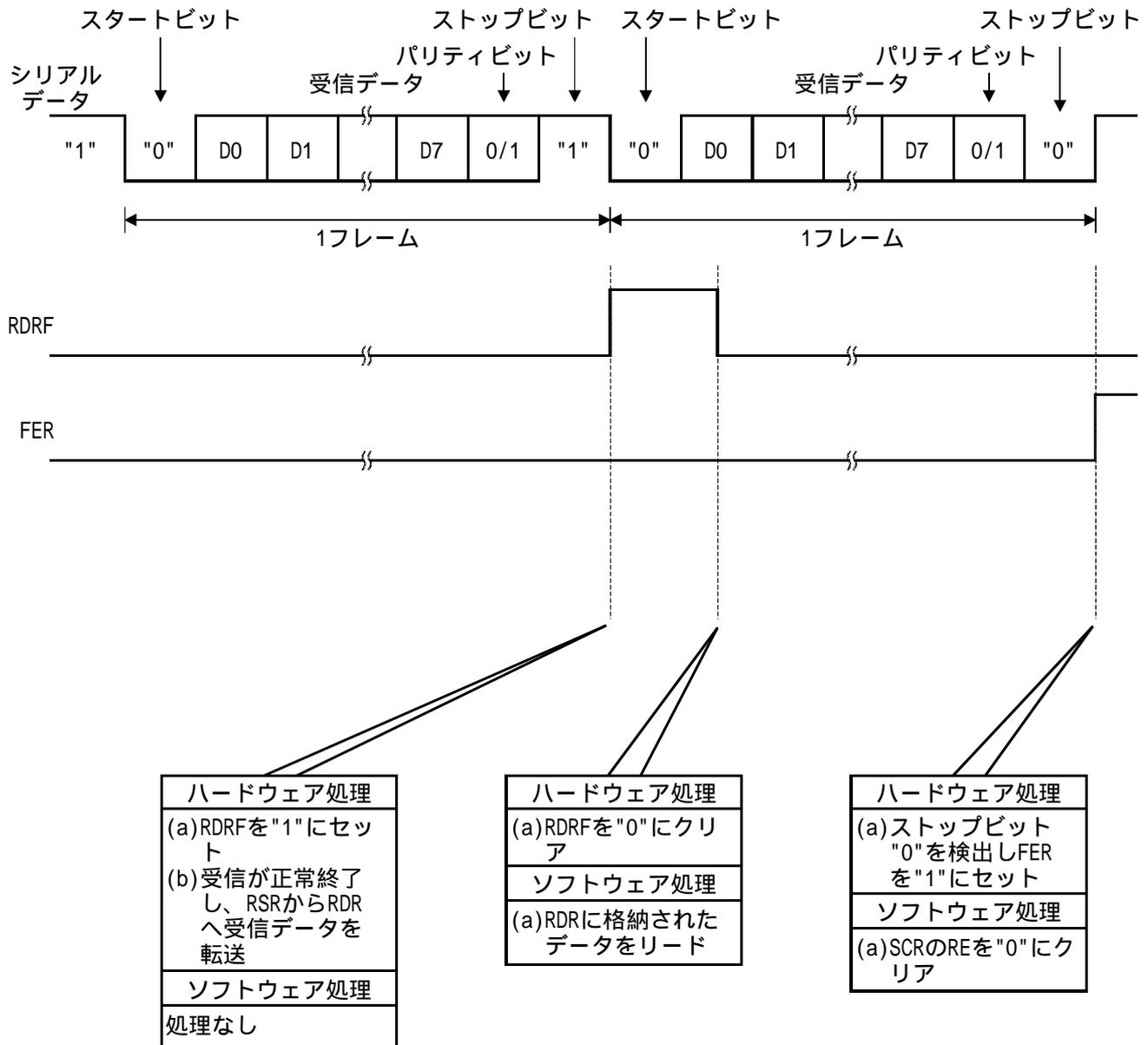


図4 調歩同期式シリアルデータ受信の動作原理

ソフトウェア説明

(1) モジュール説明

表4に本タスク例におけるモジュール説明を示します。

表4 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、調歩同期式シリアルデータ受信の設定、割込みの許可、受信エラーが発生した場合には受信エラー処理サブルーチンへ分岐、4バイトのデータを受信したところで終了
受信エラー処理	ERSUB	OER、FER、PERのどのエラーかを判定し、所定のエラー処理を行なう

(2) 引数の説明

表5に本タスク例で使用する引数を示します。

表5 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
SRD0 ~ SRD3	調歩同期式シリアル受信データ	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表6に本タスク例における使用内部レジスタ説明を示します。

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SMR	COM	シリアルモードレジスタ (コミュニケーションモード) : COM="0" のとき、コミュニケーションモードを調歩同期式モードに設定	H'FFA8 ビット7	0
	CHR	シリアルモードレジスタ (キャラクタレングス) : CHR="0" のとき、調歩同期式モード時におけるデータ長を8ビットデータに設定	H'FFA8 ビット6	0
	PE	シリアルモードレジスタ (パリティイネーブル) : PE="1" のとき、調歩同期式モードで、送信時にパリティビットの付加およびチェックを許可	H'FFA8 ビット5	1
	PM	シリアルモードレジスタ (パリティモード) : PM="1" のとき、パリティの付加やチェックを奇数パリティに設定	H'FFA8 ビット4	1
	STOP	シリアルモードレジスタ (ストップビットレングス) : STOP="0" のとき、調歩同期式モードでのストップビットの長さを1ビットに設定	H'FFA8 ビット3	0
	MP	シリアルモードレジスタ (マルチプロセッサモード) : MP="0" のとき、マルチプロセッサ通信機能を禁止	H'FFA8 ビット2	0
	CKS1 CKS0	シリアルモードレジスタ (クロックセレクト1、0) : CKS1="0"、CKS0="0" のとき、内蔵ボーレートジェネレータのクロックソースをクロックに設定	H'FFA8 ビット1 ビット0	CKS1="0" CKS0="0"
BRR	ビットレートレジスタ : BRR=H'04 のとき、SMRのCKS1、CLS0で選択されるボーレートジェネレータの動作クロックとあわせて送信のビットレートを31250 (bit/s) に設定	H'FFA9	H'04	
SCR3	RIE	シリアルコントロールレジスタ3 (レシーブインタラプトイネーブル) : RIE="0" のとき、受信データフル割込み要求 (RXI)、および受信エラー割込み要求 (ERI) を禁止	H'FFAA ビット6	0

ソフトウェア説明

表6 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
SCR3	RE	シリアルコントロールレジスタ3 (レシーブイネーブル) : RE="0"のとき、受信動作を禁止 : RE="1"のとき、受信動作を許可	H'FFAA ビット4	0
	CKE1 CKE0	シリアルコントロールレジスタ3 (クロックイネーブル1、0) : CKE1="0"、CKE0="0"のとき、調歩同期式モードにおいてク ロックソースを内部クロック、SCK ₃ 端子機能を入出力ポート に設定	H'FFAA ビット1 ビット0	CKE1="0" CKE0="0"
RDR		レシーブデータレジスタ : 受信データを格納する8ビットのレジスタ	H'FFAD	-
SSR	RDRF	シリアルステータスレジスタ (レシーブデータレジスタフル) : RDRF="0"のとき、RDRに受信データが格納されていないこと を示す : RDRF="1"のとき、RDRに受信データが格納されていることを 示す	H'FFAC ビット6	1
	OER	シリアルステータスレジスタ(オーバランエラー) : TEND="0"のとき、受信中、または受信を完了したことを示す : TEND="1"のとき、受信時にオーバランエラーが発生したことを 示す	H'FFAC ビット5	0
	FER	シリアルステータスレジスタ(フレーミングエラー) : FER="0"のとき、受信中、または受信を完了したことを示す : FER="1"のとき、受信時にフレーミングエラーが発生したことを 示す	H'FFAC ビット4	0
	PER	シリアルステータスレジスタ(パリティエラー) : PER="0"のとき、受信中、または受信を完了したことを示す : PER="1"のとき、受信時にパリティエラーが発生したことを 示す	H'FFAC ビット3	0

(4) 使用RAM説明

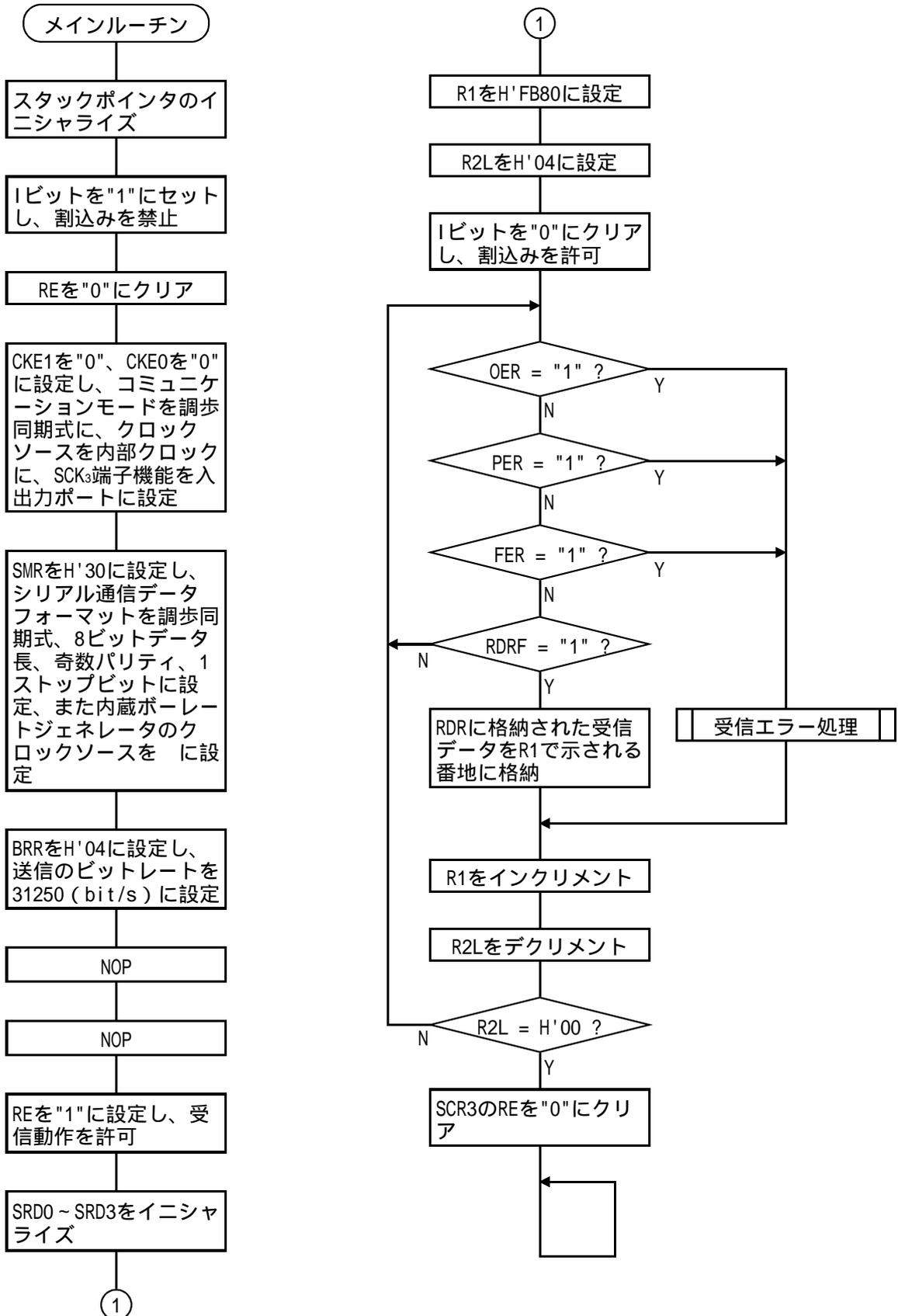
表7に本タスク例における使用RAM説明を示します。

表7 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
SRD0	調歩同期式シリアル受信データの1バイト目を格納	H'FB80	メインルーチン
SRD1	調歩同期式シリアル受信データの2バイト目を格納	H'FB81	メインルーチン
SRD2	調歩同期式シリアル受信データの3バイト目を格納	H'FB82	メインルーチン
SRD3	調歩同期式シリアル受信データの4バイト目を格納	H'FB83	メインルーチン

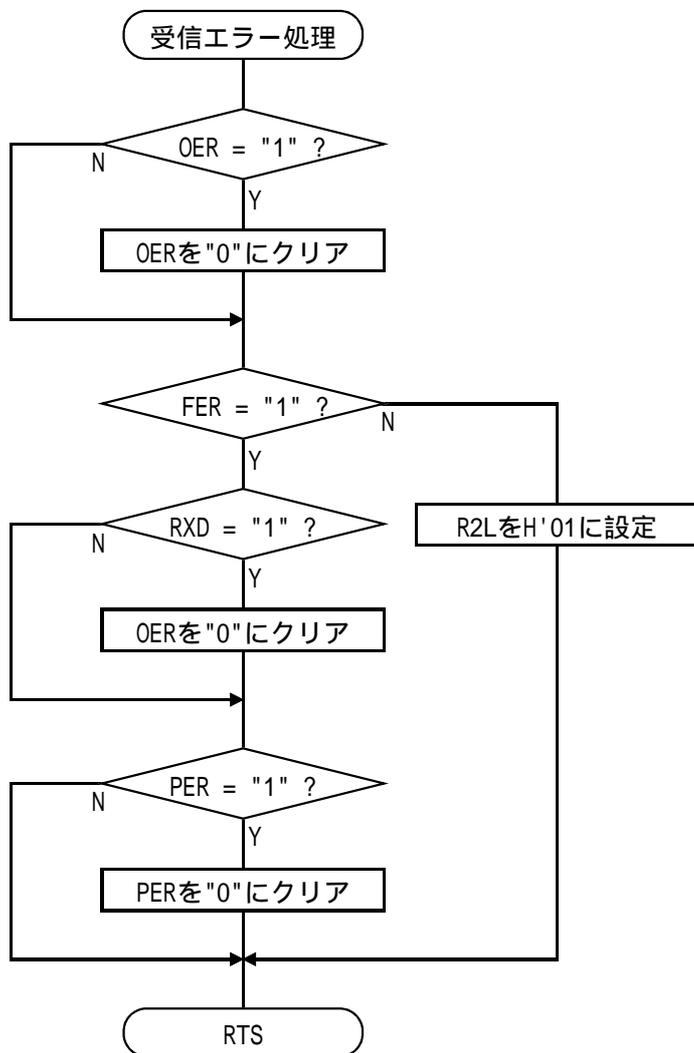
フローチャート

(a) メインルーチン



フローチャート

(b) サブルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Asynchronous Serial Data Reception'
;
;           Function
;           : Serial Communication Interface
;           :   Asynchronous Serial Interface
;           :   -Receiving
;
;           External Clock : 10MHz
;           Internal Clock :  5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
SMR           .equ           H'FFA8           ;Serial Mode Register
COM           .bequ          7,SMR           ;Communication Mode
CHR           .bequ          6,SMR           ;Character Length
PE           .bequ          5,SMR           ;Parity Enable
PM           .bequ          4,SMR           ;Parity Mode
STOP         .bequ          3,SMR           ;Stop Bit Length
MP           .bequ          2,SMR           ;Multiprocessor Mode
CKS1         .bequ          1,SMR           ;Clock Select 1
CKS0         .bequ          0,SMR           ;Clock Select 0
BRR          .equ           H'FFA9           ;Bit Rate Register
SCR3         .equ           H'FFAA           ;Serial Control Register 3
TIE          .bequ          7,SCR3          ;Transmit Interrupt Enable
RIE          .bequ          6,SCR3          ;Receive Interrupt Enable
TE           .bequ          5,SCR3          ;Transmit Enable
RE           .bequ          4,SCR3          ;Receive Enable
MPIE         .bequ          3,SCR3          ;Multiprocessor Interrupt Enable
TEIE         .bequ          2,SCR3          ;Transmit End Interrupt Enable
CKE1         .bequ          1,SCR3          ;Clock Enable 1
CKE0         .bequ          0,SCR3          ;Clock Enable 0
TDR          .equ           H'FFAB           ;Transmit Data Register
SSR          .equ           H'FFAC           ;Serial Status Register
TDRE         .bequ          7,SSR           ;Transmit Data Register Empty
RDRF         .bequ          6,SSR           ;Receive Data Register Full
OER          .bequ          5,SSR           ;Overrun Error
FER          .bequ          4,SSR           ;Framing Error
PER          .bequ          3,SSR           ;Parity Error
TEND         .bequ          2,SSR           ;Transmit End
MPBR         .bequ          1,SSR           ;Multiprocessor Bit Receive
MPBT         .bequ          0,SSR           ;Multiprocessor Bit Transfer
RDR          .equ           H'FFAD           ;Receive Data Register
PDR2         .equ           H'FFD5           ;Port Data Register 2
P22          .bequ          2,PDR2          ;Port Data Register 22
PCR2         .equ           H'FFE5           ;Port Control Register 2
PCR22        .bequ          2,PCR2          ;Port Control Register 22

```

プログラムリスト

```

PMR7      .equ          H'FFFF      ;Port Mode Register 7
TXD       .bequ        2,PMR7      ;P22/TXD Terminal Function Switch
;
;
;*****
;
;          RAM Allocation
;*****
;
;
STACK     .equ          H'FF80      ;Stack Pointer
SRD0     .equ          H'FB80      ;Serial Receiving Data 0
SRD1     .equ          H'FB81      ;Serial Receiving Data 1
SRD2     .equ          H'FB82      ;Serial Receiving Data 2
SRD3     .equ          H'FB83      ;Serial Receiving Data 3
;
;
;*****
;
;          Vector Address
;*****
;
;          .org          H'0000
;          .data.w      MAIN        ;Reset Interrupt
;
;          .org          H'0008
;          .data.w      MAIN        ;IRQ0 Interrupt
;          .data.w      MAIN        ;IRQ1 Interrupt
;          .data.w      MAIN        ;IRQ2 Interrupt
;          .data.w      MAIN        ;IRQ3 Interrupt
;          .data.w      MAIN        ;INT0 - INT7 Interrupt
;
;          .org          H'0014
;          .data.w      MAIN        ;Timer A Interrupt
;          .data.w      MAIN        ;Timer B1 Interrupt
;
;          .org          H'0020
;          .data.w      MAIN        ;Timer X Interrupt
;          .data.w      MAIN        ;Timer V Interrupt
;
;          .org          H'0026
;          .data.w      MAIN        ;SCI1 Interrupt
;
;          .org          H'002A
;          .data.w      MAIN        ;SCI3 Interrupt
;          .data.w      MAIN        ;A/D Converter Interrupt
;          .data.w      MAIN        ;SLEEP Instruction Executed Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org          H'1000
;
;MAIN     .equ          $
;          MOV.W        #STACK,SP   ;Initialize Stack Pointer
;          ORC          #H'80,CCR    ;Interrupt Disable
;
;          MOV.W        #H'0030,R0
;          MOV.B        ROH,@SCR3   ;Initialize RE & Initialize Clock Source
;          MOV.B        ROL,@SMR    ;Initialize Receive Format
;
;          MOV.B        #H'04,ROL
;          MOV.B        ROL,@BRR    ;Initialize Receive Bit Rate
;

```

プログラムリスト

```

NOP
NOP
;
;
;      BSET      RE          ;Receive Enable
;
;      MOV.B     ROH,@SRD0   ;Initialize Serial Receiving Data
;      MOV.B     ROH,@SRD1   ;Initialize Serial Receiving Data
;      MOV.B     ROH,@SRD2   ;Initialize Serial Receiving Data
;      MOV.B     ROH,@SRD3   ;Initialize Serial Receiving Data
;
;      MOV.W     #'FB80,R1   ;Initialize Serial Receiving Data Address
;      MOV.B     #'04,R2L    ;Initialize Serial Receiving Data Counter
;
;      ANDC      #'7F,CCR    ;Interrupt Enable
;
MAIN1  .equ      $
      BTST      OER          ;OER = "1" ?
      BNE      MAIN2        ;Yes.
;
;      BTST      PER          ;PER = "1" ?
;      BNE      MAIN2        ;Yes.
;
;      BTST      FER          ;FER = "1" ?
;      BNE      MAIN2        ;Yes.
;
;      BTST      RDRF         ;RDRF = "1" ?
;      BEQ      MAIN1        ;No.
;
;      BRA      MAIN3
;
MAIN2  .equ      $
      JSR      @ERSUB
      BRA      MAIN4
;
MAIN3  .equ      $
      MOV.B     @RDR,R0L     ;Load Serial Receiving Data
      MOV.B     R0L,@R1     ;Save Serial Receiving Data
;
MAIN4  .equ      $
      ADDS     #1,R1        ;Incerment Serial Receiving Data Address
      DEC      R2L          ;Decrement Serial Receiving Data Counter
      BNE      MAIN1        ;Serial Receiving Data Counter = H'00 ? No.
;
;      BCLR      RE          ;Clear RE
;
MAIN9  .equ      $
      BRA      MAIN9
;
;*****
;
;      Errorr Subroutine
;*****
;
;
ERSUB  .equ      $
      BTST      OER          ;OER = "1" ?
      BEQ      ERSUB1       ;No.
;
;      BCLR      OER          ;Clear OER
;
ERSUB1 .equ      $
      BTST      FER          ;FER = "1" ?

```

プログラムリスト

```

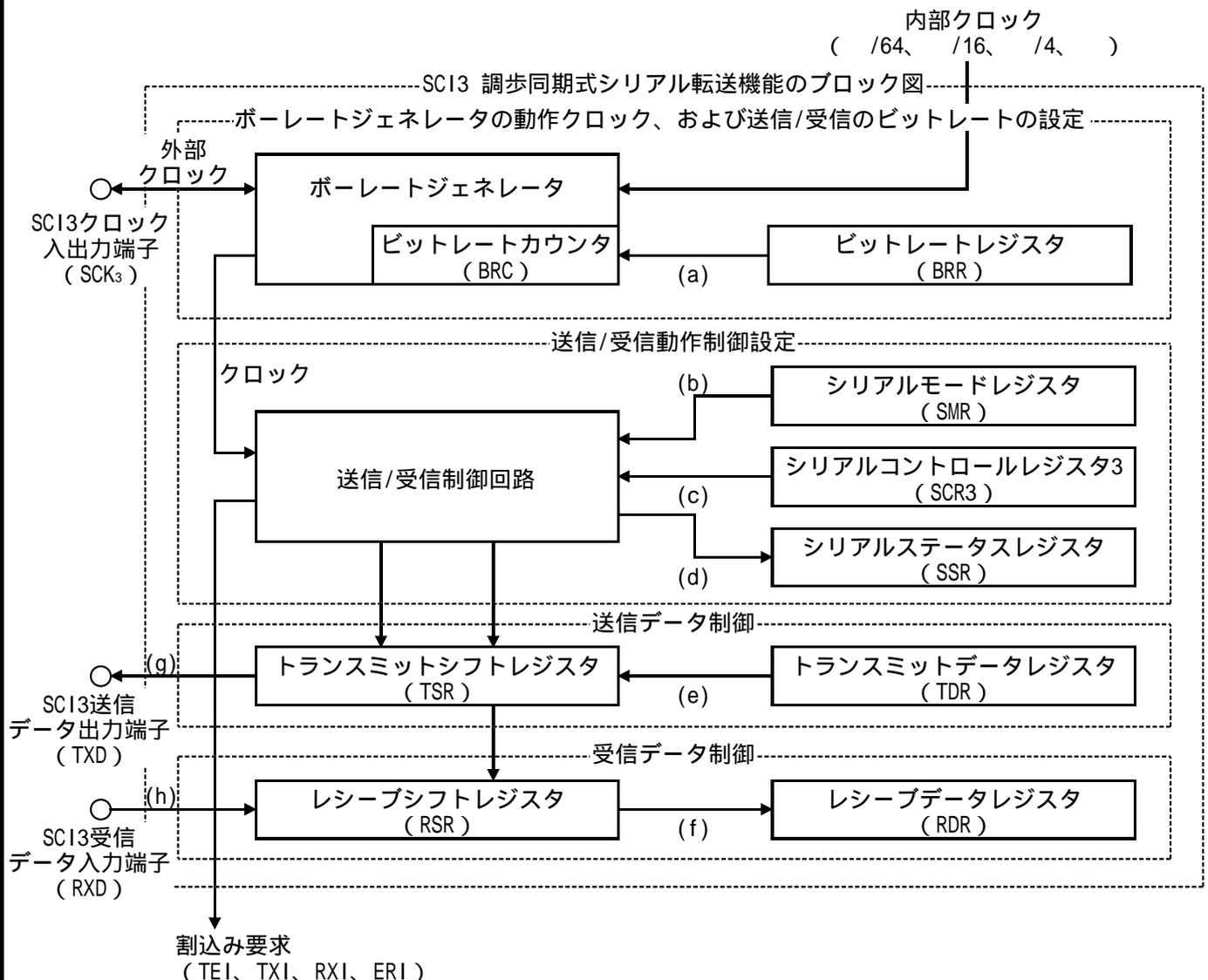
;          BEQ          ERSUB3          ;No.
;
;          BTST         P22              ;RXD = "1" ?
;          BEQ          ERSUB2          ;Yes
;
;          BCLR         FER              ;Clear FER
;          BRA          ERSUB3
;
ERSUB2     .equ         $
;          MOV.B        #H'01,R2L      ;Set Serial Receiving Data Counter
;
;
ERSUB3     .equ         $
;          BTST         PER              ;PER = "1" ?
;          BEQ          ERSUB4          ;No.
;
;          BCLR         PER              ;Clear PER
;
;
ERSUB4     .equ         $
;          RTS
;
;          .end
```

2.19 調歩同期式シリアルデータ同時送受信

調歩同期式シリアルデータ同時送受信	使用機能	SCI3 : 調歩同期式シリアル転送機能
仕様		
<p>(1) 図1に示すように調歩同期式シリアル転送機能を使用して、4バイトの8ビットデータの同時送受を行います。</p> <p>(2) 送信データの通信フォーマットは、データ長が8ビット、奇数パリティ、ストップビット長が1ビットに設定します。</p> <p>(3) ビットレートは31250 (bit/s) で送信します。4バイトのデータを送受信すると終了します。</p>		
図1 調歩同期式シリアルデータ同時送受信		

使用機能説明		
<p>(1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、調歩同期式のシリアルデータの同時送受信を行ないます。図2に調歩同期式シリアルデータ同時送受信のブロック図を示します。以下に調歩同期式シリアルデータ同時送受信のブロック図について説明します。</p> <ul style="list-style-type: none"> ・調歩同期式モードは、キャラクタ単位で同期をとる調歩同期方式でシリアルデータ通信を行ないます。 ・Universal Asynchronous Receiver/Transmitter (UART) や、Asynchronous Communication Interface Adapter (ACIA) などの標準の調歩同期式通信用LSIとのシリアルデータ通信ができます。 ・複数のプロセッサとシリアル通信ができるマルチプロセッサ間通信機能を備えています。 ・通信フォーマットを12種類のフォーマットから選択できます。 ・独立した送信部と受信部を備えているので、送信と受信を同時に行なうことができます。また、送信部および受信部ともにダブルバッファ構造になっているため、連続送信・連続受信ができます。 ・内蔵のボーレートジェネレータで任意のビットレートを選択可能です。 ・送受信クロックソースを内部クロック、または外部クロックから選択可能です。 ・割り込み要因には送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割り込み要因があります。 ・レシーブシフトレジスタ (RSR) は、シリアルデータを受信するためのレジスタです。RSRにRXD端子から入力されたシリアルデータを、LSB (ビット0) から受信した順にセットしパラレルデータに変換します。1バイトのデータを受信すると、データは自動的にRDRへ転送されます。CPUからRSRを直接リード/ライトすることはできません。 ・レシーブデータレジスタ (RDR) は、受信したシリアルデータを格納する8ビットのレジスタです。1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送し、受信動作を完了します。その後、RSRは受信可能となります。RSRとRDRはダブルバッファになっているため連続した受信動作が可能です。RDRは受信専用レジスタなのでCPUからライトできません。 ・トランスミットシフトレジスタ (TSR) は、シリアルデータを送信するためのレジスタです。TDRから送信データをいったんTSRに転送し、LSB (ビット0) から順にTXD端子に送出することでシリアルデータ送信を行ないます。1バイトのデータを送信すると、自動的にTDRからTSRへ次の送信データを転送し、送信を開始します。ただし、TDRにデータが書き込まれていない (TDREに"1"がセットされている) 場合にはTDRからTSRへのデータ転送は行ないません。CPUからTSRを直接リード/ライトすることはできません。 ・トランスミットデータレジスタ (TDR) は、送信データを格納する8ビットのレジスタです。TSRの"空"を検出すると、TDRに書き込まれた送信データをTSRに転送し、シリアルデータ送信を開始します。TSRのシリアルデータ送信中に、TDRに次の送信データをライトしておく、連続送信が可能です。TDRは、常にCPUによるリード/ライトが可能です。 ・シリアルモードレジスタ (SMR) は、シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。SMRは、常にCPUによるリード/ライトが可能です。 ・シリアルコントロールレジスタ3 (SCR3) は、送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止、および送信/受信クロックソースの選択を行なう8ビットのレジスタです。SCR3は、常にCPUによるリード/ライトが可能です。 		

使用機能説明



- 【注】 (a) SMRで選択されるボーレートジェネレータの動作クロックと合わせて、送信/受信のビットレートを設定します。本タスク例では、送信のビットレートを31250 (bit/s) に設定しています。
- (b) シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択します。本タスク例でシリアルデータ通信フォーマットは、動作モードを調歩同期式に、データ長を8ビットに、パリティビットあり、パリティモードを奇数パリティに、ストップビット長を1ビットに、内蔵ボーレートジェネレータのクロックソースをクロックに設定しています。
- (c) 送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止を選択します。本タスク例では、調歩同期式モードでのクロック出力は、コミュニケーションモードを調歩同期式に、クロックソースを内部クロックに、SCK₃端子機能を出力クロックに設定しています。また、割り込み要求の許可/禁止は送信データエンpty割り込み要求を禁止、受信データフル割り込み要求を禁止、受信エラー割り込み要求を禁止に設定しています。
- (d) ステータスフラグ (トランスミットデータレジスタエンpty、レシーブデータレジスタフル、オーバーランエラー、フレーミングエラー、パリティエラー、トランスミットエンド) によりSC13の動作状態を示す。
- (e) TSRの"空"を検出することにより、TDRに書き込まれた送信データをTSRに転送。
- (f) 1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送。
- (g) 送信データ。
- (h) 受信データ。

図2 調歩同期式シリアルデータ同時送受信のブロック図

使用機能説明

- ・シリアルステータスレジスタ (SSR) は、SCI3の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8ビットのレジスタです。SSRは常にCPUからリード/ライトできます。ただし、TDRE、RDRF、OER、PER、FERへ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。また、TENDおよびMPBRはリード専用であり、ライトすることはできません。
- ・ビットレートレジスタ (BRR) は、SMRのCKS1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて、送信/受信のビットレートを設定する8ビットのレジスタです。BRRは常にCPUによるリード/ライトが可能です。
- ・表1に、調歩同期式モードのBRRの設定例を示します。表1はアクティブ (高速) モードで、OSCが10MHzのときの値を示しています。

表1 ビットレートに対するBRRの設定例 (調歩同期式モード)

Rビットレート (bit/s)	110	150	300	600	1200	2400	4800	9600	19200	31250	38400
n	2	2	1	0	0	0	0	0	0	0	0
N	88	64	129	64	129	64	32	15	7	4	3
誤差 (%)	-0.25	+0.16	+0.16	+0.16	+0.16	+0.16	-1.36	+1.73	+1.73	0	+1.73

- 【注】 1. 誤差は、1%以内となるように設定します。
2. BRRの設定値は以下の計算式で求められます。

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

B : ビットレート (bit/s)

N : ボーレートジェネレータのBRRの設定値 (0 N 255)

OSC : oscの値 (MHz) = 10MHz

n : ボーレートジェネレータの入力クロックのNo. (n=0, 1, 2, 3)
(nとクロックの関係は表2を参照)

表2 nとクロックの関係

n	クロック	SMRの設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

3. 表1に誤差は以下の計算式で求めた値を小数点第3位を四捨五入して表示してあります。

$$\text{誤差 (\%)} = \frac{B(n, N, OSC \text{から求めたレート}) - R(\text{表1の上欄のビットレート})}{R(\text{表1の上欄のビットレート})} \times 100$$

4. OSCが10MHzのときの最大ビットレート (調歩同期式モード) は、156250 (bit/s) になります。ただし、設定値は、n=0、N=0のときです。

- ・調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信/受信し、1キャラクタ単位で同期をとりながらシリアル通信を行なうモードです。
- ・SCI3内部では、送信部と受信部は独立しているので、全二重通信を行なうことができます。また、送信部と受信部がともにダブルバッファ構造になっているので、送信中にデータのライト、受信中にデータのリードができ、連続送信/受信が可能です。
- ・図3に調歩同期式通信のデータフォーマットを示します。調歩同期式通信では、通信回線は通常マーク状態 ("High" レベル) に保たれています。SCI3では通信回線を監視し、スペース ("Low" レベル) になったところをスタートビットとみなしてシリアル通信を開始します。
- ・通信データの1キャラクタはスタートビット ("Low" レベル) から始まり、送信/受信データ (LSBファースト: 最下位ビットから)、パリティビット ("High" または "Low" レベル)、最後にストップビット ("High" レベル) の順で構成されます。
- ・調歩同期式モードでは、受信時にスタートビットの立ち上がりエッジで同期化を行いません。また、データを1ビット期間の16倍の周波数のクロックの8番目でサンプリングするので、各ビットの中央で通信データを取り込みます。

使用機能説明

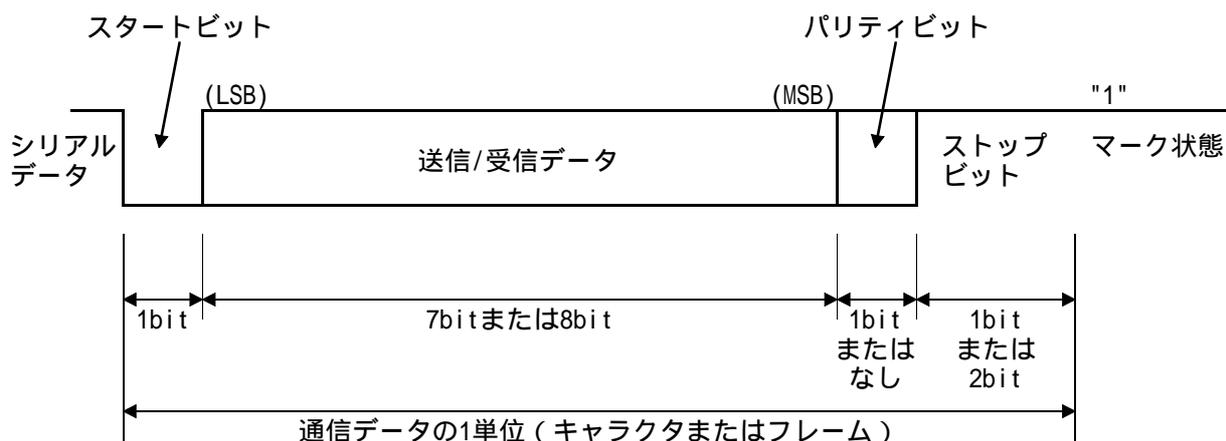


図3 調歩同期式通信のデータフォーマット

- ・SCI3クロック (SCK₃) は、SCI3のクロック入出力端子です。
- ・SCI3レシーブデータ入力 (RXD) は、SCI3の受信データ入力端子です。
- ・SCI3トランスミットデータ出力 (TXD) は、SCI3の送信データ出力端子です。
- ・SCI3の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび3種類の受信エラー (オーバランエラー、フレーミングエラー、パリティエラー) の計6種類があり、共通のベクタアドレスが割り付けられています。
- ・各割込み要求は、SCR3のTIE、RIEで許可/禁止できます。
- ・SSRのTDREが"1"にセットされるとTXIが発生します。SSRのTENDが"1"にセットされると、TEIが発生します。この2つの割込みは送信時に発生します。
- ・SSRのTDREは初期値が"1"になっています。したがって送信データをTDRへ転送する前にSCR3のTIEを"1"にセットして送信データエンプティ割込み要求 (TXI) を許可すると、送信データが準備されていなくてもTXIが発生します。
- ・SSRのTENDは初期値が"1"になっています。したがって、送信データをTDRへ転送する前にSCR3のTEIEを"1"にセットして送信終了割込み要求 (TEI) を許可すると、送信データが送信されていなくてもTEIが発生します。
- ・送信データをTDRへ転送する処理を割込み処理ルーチンの中で行なうようにすることで、これらの割込みを有効に利用できます。また、これらの割込み要求 (TXI、TEI) の発生を防ぐためには、送信データをTDRへ転送した後に、これらの割込み要求に対応する許可ビット (TIE、TEIE) を"1"にセットします。
- ・SSRのRDRFが"1"にセットされるとRXIが発生します。OER、PER、FERのいずれかが"1"にセットされるとERIが発生します。この2つの割込み要求は受信時に発生します。

(2) 表3に本タスク例の機能割付けを示します。表3に示すように機能を割り付け、調歩同期式シリアルデータ同時送受信を行ないます。

表3 機能割付け

機能	機能割付け
RSR	シリアルデータを受信するためのレジスタ
RDR	受信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット、ボーレートジェネレータのクロックソースの設定
SSR	SCI3の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
RXI	受信データフル (RDRF) による割込み要求
ERI	受信エラー (OER、FER、PER) による割込み要求
SCK ₃	SCI3のクロック出力端子
TXD	SCI3の送信データ出力端子
RXD	SCI3の受信データ入力端子

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理により調歩同期式シリアルデータ同時送受信を行います。

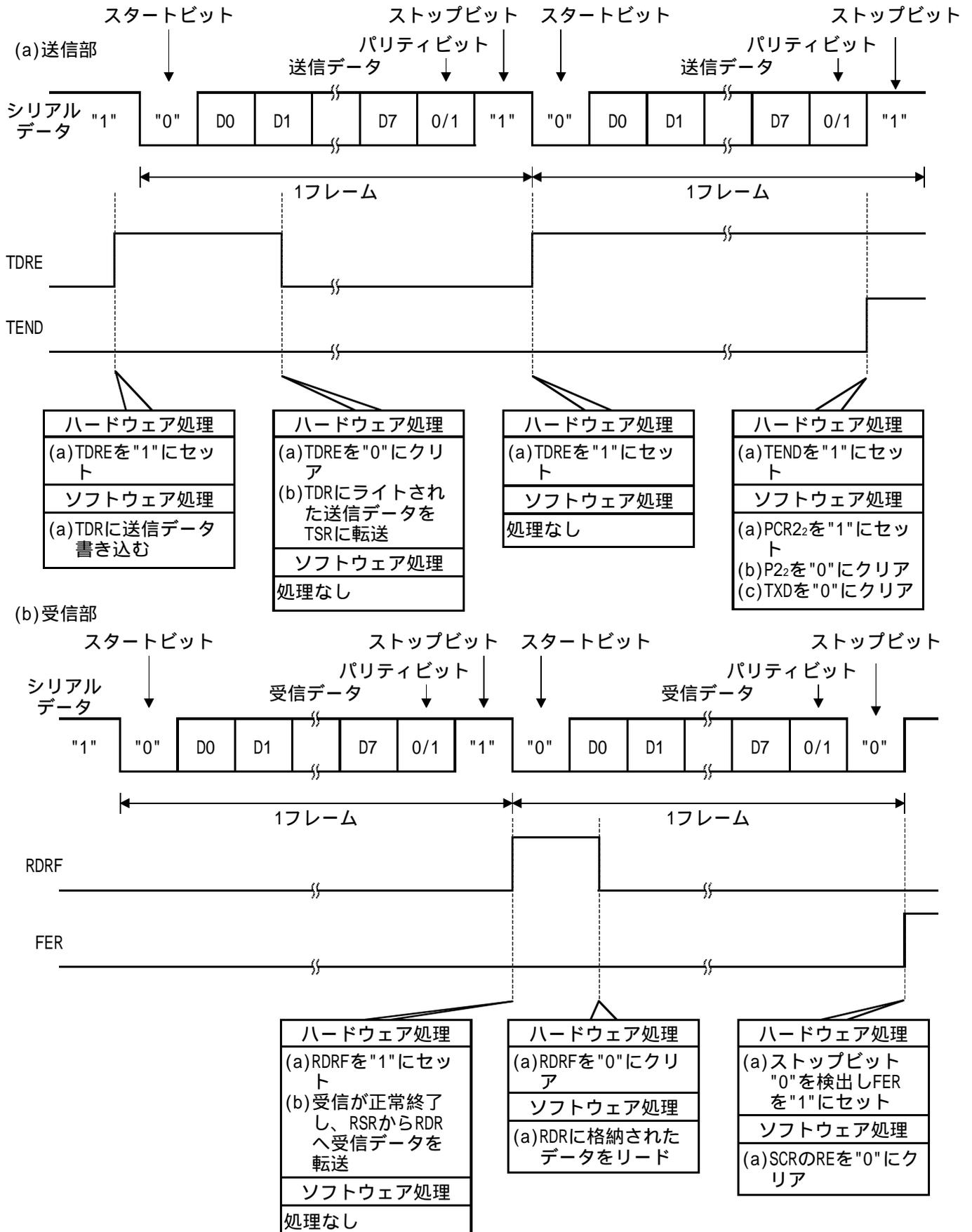


図4 調歩同期式シリアルデータ同時送受信の動作原理

ソフトウェア説明

(1) モジュール説明

表4に本タスク例におけるモジュール説明を示します。

表4 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、調歩同期式シリアルデータ送受信の設定、割込みの許可、受信エラーが発生した場合には受信エラー処理サブルーチンへ分岐、4バイトのデータを送受信すると終了
受信エラー処理	ERSUB	OER、FER、PERのどのエラーかを判定し、所定のエラー処理を行なう

(2) 引数の説明

表5に本タスク例で使用する引数を示します。

表5 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0 ~ STD3	調歩同期式シリアル送信データ	メインルーチン	1バイト	入力
SRD0 ~ SRD3	調歩同期式シリアル受信データ	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表6に本タスク例における使用内部レジスタ説明を示します。

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SMR	COM	シリアルモードレジスタ (コミュニケーションモード) : COM="0"のとき、コミュニケーションモードを調歩同期式モードに設定	H'FFA8 ビット7	0
	CHR	シリアルモードレジスタ (キャラクタレングス) : CHR="0"のとき、調歩同期式モード時におけるデータ長を8ビットデータに設定	H'FFA8 ビット6	0
	PE	シリアルモードレジスタ (パリティイネーブル) : PE="1"のとき、調歩同期式モードで、送信時にパリティビットの付加およびチェックを許可	H'FFA8 ビット5	1
	PM	シリアルモードレジスタ (パリティモード) : PM="1"のとき、パリティの付加やチェックを奇数パリティに設定	H'FFA8 ビット4	1
	STOP	シリアルモードレジスタ (ストップビットレングス) : STOP="0"のとき、調歩同期式モードでのストップビットの長さを1ビットに設定	H'FFA8 ビット3	0
	MP	シリアルモードレジスタ (マルチプロセッサモード) : MP="0"のとき、マルチプロセッサ通信機能を禁止	H'FFA8 ビット2	0
	CKS1 CKS0	シリアルモードレジスタ (クロックセレクト1、0) : CKS1="0"、CKS0="0"のとき、内蔵ボーレートジェネレータのクロックソースをクロックに設定	H'FFA8 ビット1 ビット0	CKS1="0" CKS0="0"
BRR	ビットレートレジスタ : BRR=H'04のとき、SMRのCKS1、CLS0で選択されるボーレートジェネレータの動作クロックとあわせて送信のビットレートを31250 (bit/s) に設定	H'FFA9	H'04	
SCR3	TIE	シリアルコントロールレジスタ3 (トランスミットインタラプトイネーブル) : TIE="0"のとき、送信データエンプティ割込み要求 (TXI) を禁止	H'FFAA ビット7	0

ソフトウェア説明

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR3	RIE	シリアルコントロールレジスタ3 (レシーブインタラプトイネーブル) : RIE="0"のとき、受信データフル割込み要求 (RXI)、および 受信エラー割込み要求 (ERI) を禁止	H'FFAA ビット6	0
	TE	シリアルコントロールレジスタ3 (トランスミットイネーブル) : TE="0"のとき、送信動作を禁止 (TXD端子はトランスミ ットデータ端子)	H'FFAA ビット5	0
	RE	シリアルコントロールレジスタ3 (レシーブイネーブル) : RE="0"のとき、受信動作を禁止 : RE="1"のとき、受信動作を許可	H'FFAA ビット4	0
	CKE1 CKE0	シリアルコントロールレジスタ3 (クロックイネーブル1、0) : CKE1="0"、CKE0="1"のとき、調歩同期式モードにおいてク ロックソースを内部クロック、SCK ₃ 端子機能をクロック出力 に設定	H'FFAA ビット1 ビット0	CKE1="0" CKE0="1"
TDR	トランスミットデータレジスタ : 送信データを格納する8ビットのレジスタ	H'FFAB	-	
RDR	レシーブデータレジスタ : 受信データを格納する8ビットのレジスタ	H'FFAD	-	
SSR	TDRE	シリアルステータスレジスタ (トランスミットデータエンプティ) : TDRE="0"のとき、TDRにライトされた送信データがTSRに転送 されていないことを示す : TDRE="1"のとき、TDRに送信データがライトされていない、 またはTDRIにライトされた送信データがTSRに転送されてい ないことを示す	H'FFAC ビット7	1
	RDRF	シリアルステータスレジスタ (レシーブデータレジスタフル) : RDRF="0"のとき、RDRに受信データが格納されていないこと を示す : RDRF="1"のとき、RDRに受信データが格納されていることを 示す	H'FFAC ビット6	1
	OER	シリアルステータスレジスタ (オーバランエラー) : TEND="0"のとき、受信中、または受信を完了したことを示す : TEND="1"のとき、受信時にオーバランエラーが発生したことを 示す	H'FFAC ビット5	0
	FER	シリアルステータスレジスタ (フレーミングエラー) : FER="0"のとき、受信中、または受信を完了したことを示す : FER="1"のとき、受信時にフレーミングエラーが発生したことを 示す	H'FFAC ビット4	0
	PER	シリアルステータスレジスタ (パリティエラー) : PER="0"のとき、受信中、または受信を完了したことを示す : PER="1"のとき、受信時にパリティエラーが発生したことを 示す	H'FFAC ビット3	0
	TEND	シリアルステータスレジスタ (トランスミットエンド) : TEND="0"のとき、送信中であることを示す : TEND="1"のとき、送信を終了したことを示す	H'FFAC ビット2	1
PDR2	P2 ₂	ポートデータレジスタ2 (ポートデータレジスタ22) : P2 ₂ ="0"のとき、P2 ₂ 端子の出力レベルは"Low" : P2 ₂ ="1"のとき、P2 ₂ 端子の出力レベルは"High"	H'FFD5 ビット2	0

表6 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
PCR2	PCR2 ₂ ポートコントロールレジスタ2 (ポートコントロールレジスタ22) : PCR2 ₂ ="0"のとき、P2 ₂ 端子を入力端子機能に設定 : PCR2 ₂ ="1"のとき、P2 ₂ 端子を出力端子機能に設定	H'FFE5 ビット2	0
PMR7	TXD ポートモードレジスタ7 (P2 ₂ /TXD端子機能切替え) : TXD="1"のとき、P2 ₂ /TXD端子機能をTXD端子機能に設定	H'FFFF ビット2	1

(4) 使用RAM説明

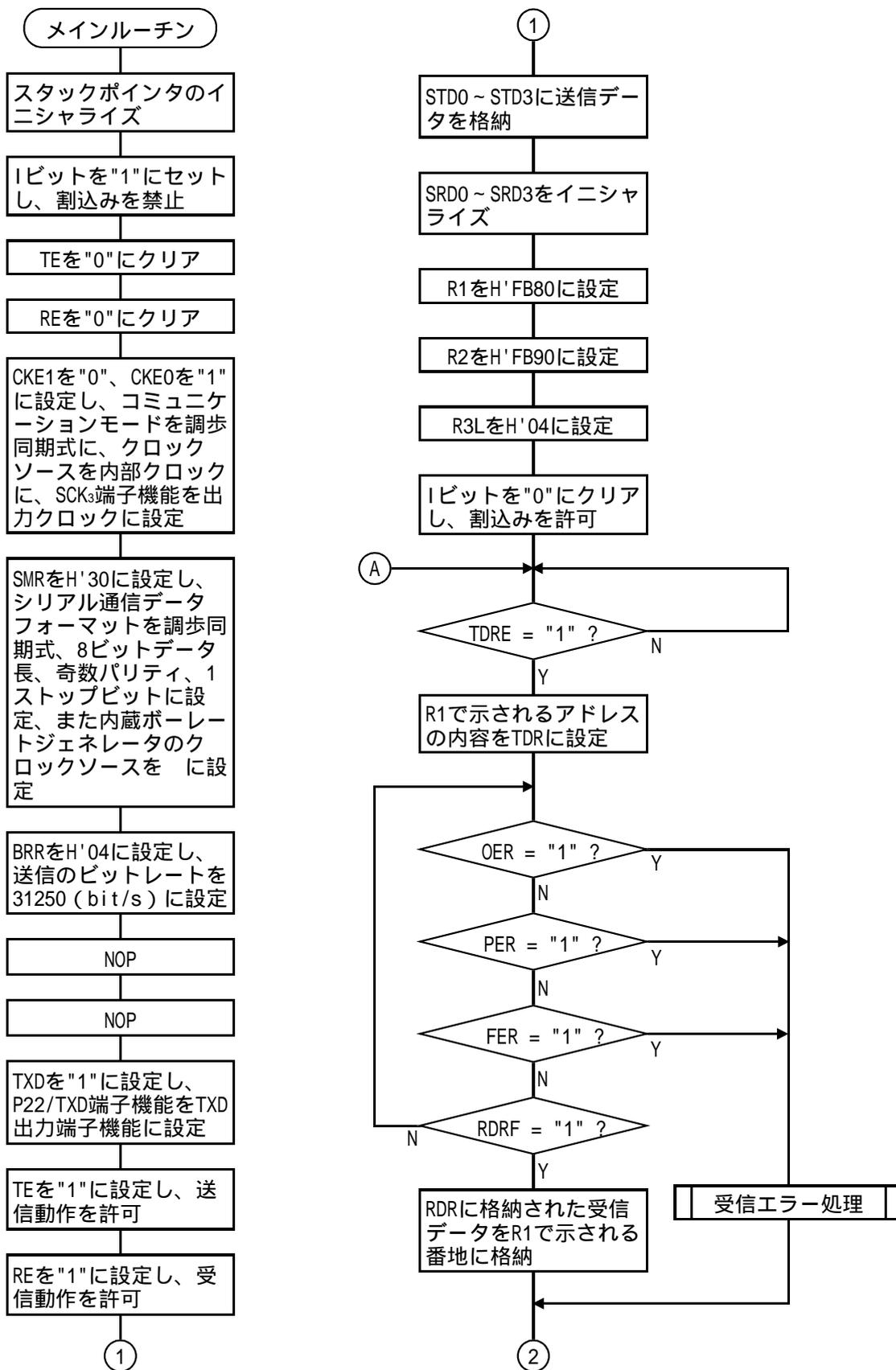
表7に本タスク例における使用RAM説明を示します。

表7 使用RAM説明

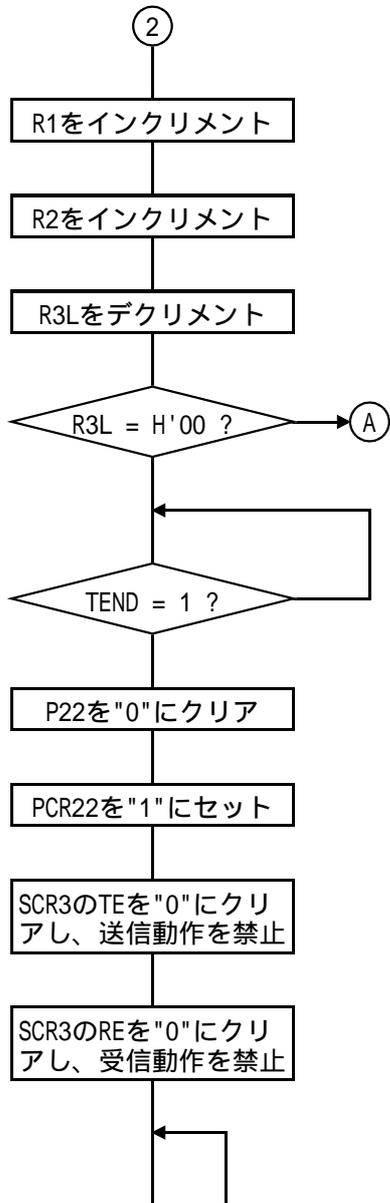
ラベル名	機能	アドレス	使用モジュール名
STD0	調歩同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	調歩同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	調歩同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	調歩同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン
SRD0	調歩同期式シリアル受信データの1バイト目を格納	H'FB90	メインルーチン
SRD1	調歩同期式シリアル受信データの2バイト目を格納	H'FB91	メインルーチン
SRD2	調歩同期式シリアル受信データの3バイト目を格納	H'FB92	メインルーチン
SRD3	調歩同期式シリアル受信データの4バイト目を格納	H'FB93	メインルーチン

フローチャート

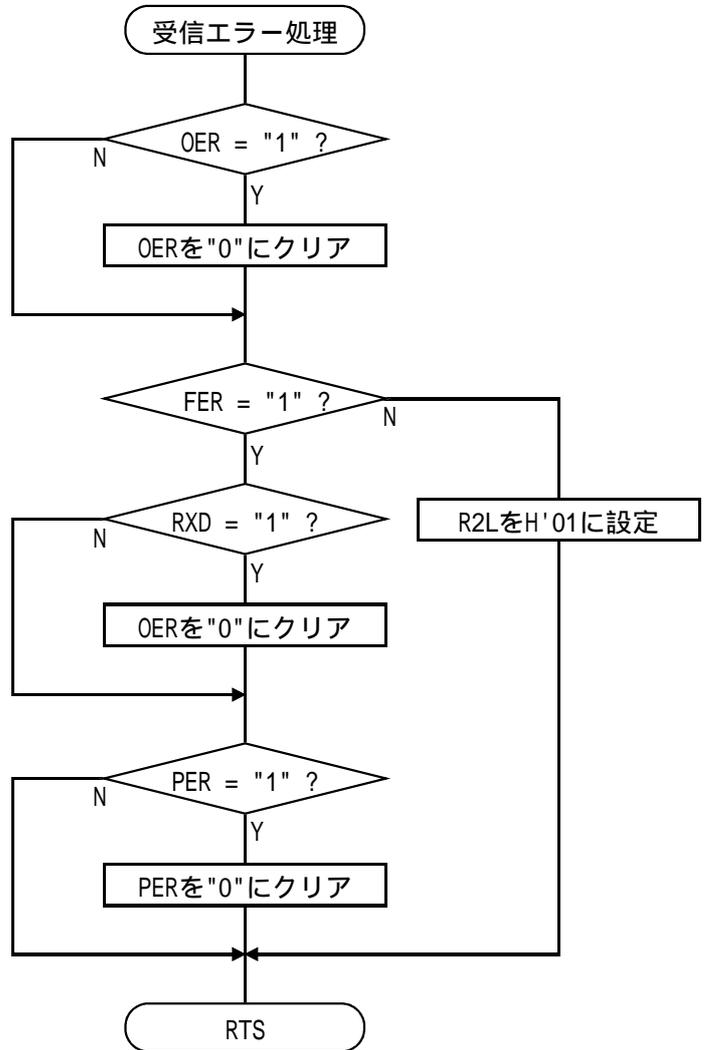
(a) メインルーチン



フローチャート



(b) サブルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Asynchronous Serial Data Simultaneous
;       Transmission and Reception'
;
;       Function
;       : Serial Communication Interface
;       :   Asynchronous Serial Interface
;       :   -Transmitting/Receiving
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
SMR          .equ      H'FFA8      ;Serial Mode Register
COM          .bequ     7,SMR       ;Communication Mode
CHR          .bequ     6,SMR       ;Character Length
PE           .bequ     5,SMR       ;Parity Enable
PM           .bequ     4,SMR       ;Parity Mode
STOP         .bequ     3,SMR       ;Stop Bit Length
MP           .bequ     2,SMR       ;Multiprocessor Mode
CKS1         .bequ     1,SMR       ;Clock Select 1
CKS0         .bequ     0,SMR       ;Clock Select 0
BRR          .equ      H'FFA9      ;Bit Rate Register
SCR3         .equ      H'FFAA      ;Serial Control Register 3
TIE          .bequ     7,SCR3      ;Transmit Interrupt Enable
RIE          .bequ     6,SCR3      ;Receive Interrupt Enable
TE           .bequ     5,SCR3      ;Transmit Enable
RE           .bequ     4,SCR3      ;Receive Enable
MPIE         .bequ     3,SCR3      ;Multiprocessor Interrupt Enable
TEIE         .bequ     2,SCR3      ;Transmit End Interrupt Enable
CKE1         .bequ     1,SCR3      ;Clock Enable 1
CKE0         .bequ     0,SCR3      ;Clock Enable 0
TDR          .equ      H'FFAB      ;Transmit Data Register
SSR          .equ      H'FFAC      ;Serial Status Register
TDRE         .bequ     7,SSR       ;Transmit Data Register Empty
RDRF         .bequ     6,SSR       ;Receive Data Register Full
OER          .bequ     5,SSR       ;Overrun Errorr
FER          .bequ     4,SSR       ;Framing Errorr
PER          .bequ     3,SSR       ;Parity Errorr
TEND         .bequ     2,SSR       ;Transmit End
MPBR         .bequ     1,SSR       ;Multiprocessor Bit Receive
MPBT         .bequ     0,SSR       ;Multiprocessor Bit Transfer
RDR          .equ      H'FFAD      ;Receive Data Register
PDR2         .equ      H'FFD5      ;Port Data Register 2
P22          .bequ     2,PDR2      ;Port Data Register 22
PCR2         .equ      H'FFE5      ;Port Control Register 2

```

プログラムリスト

```

PCR22      .bequ      2,PCR2      ;Port Control Register 22
PMR7       .equ       H'FFFF     ;Port Mode Register 7
TXD        .bequ      2,PMR7     ;P22/TXD Terminal Function Switch
;
;
;*****
;
;          RAM Allocation
;*****
;
;
STACK      .equ       H'FF80     ;Stack Pointer
STD0       .equ       H'FB80     ;Serial Transmitting Data 0
STD1       .equ       H'FB81     ;Serial Transmitting Data 1
STD2       .equ       H'FB82     ;Serial Transmitting Data 2
STD3       .equ       H'FB83     ;Serial Transmitting Data 3
SRD0       .equ       H'FB90     ;Serial Receiving Data 0
SRD1       .equ       H'FB91     ;Serial Receiving Data 1
SRD2       .equ       H'FB92     ;Serial Receiving Data 2
SRD3       .equ       H'FB93     ;Serial Receiving Data 3
;
;
;*****
;
;          Vector Address
;*****
;
;
;          .org      H'0000
;          .data.w   MAIN      ;Reset Interrupt
;
;
;          .org      H'0008
;          .data.w   MAIN      ;IRQ0 Interrupt
;          .data.w   MAIN      ;IRQ1 Interrupt
;          .data.w   MAIN      ;IRQ2 Interrupt
;          .data.w   MAIN      ;IRQ3 Interrupt
;          .data.w   MAIN      ;INT0 - INT7 Interrupt
;
;
;          .org      H'0014
;          .data.w   MAIN      ;Timer A Interrupt
;          .data.w   MAIN      ;Timer B1 Interrupt
;
;
;          .org      H'0020
;          .data.w   MAIN      ;Timer X Interrupt
;          .data.w   MAIN      ;Timer V Interrupt
;
;
;          .org      H'0026
;          .data.w   MAIN      ;SCI1 Interrupt
;
;
;          .org      H'002A
;          .data.w   MAIN      ;SCI3 Interrupt
;          .data.w   MAIN      ;A/D Converter Interrupt
;          .data.w   MAIN      ;SLEEP Instruction Executed Interrupt
;
;
;*****
;
;          Main Program
;*****
;
;
;          .org      H'1000
;
;
MAIN       .equ       $
MOV.W     #STACK,SP      ;Initialize Stack Pointer
ORC       #H'80,CCR      ;Interrupt Disable
;
;
MOV.W     #H'0130,R0
MOV.B     R0H,@SCR3      ;Initialize RE & Initialize Clock Source

```

プログラムリスト

```

;
MOV.B      ROL,@SMR      ;Initialize Receive Format
;
MOV.B      #H'04,ROL
MOV.B      ROL,@BRR      ;Initialize Receive Bit Rate
;
NOP
NOP
;
BSET      TXD            ;Initialize TXD Terminal Function
BSET      TE              ;Transmit Enable
BSET      RE              ;Receive Enable
;
MOV.W      #H'0055,R0
MOV.B      ROH,@STDO      ;Initialize Serial Transmitting Data 0
MOV.B      ROL,@STD1      ;Initialize Serial Transmitting Data 1
MOV.W      #H'AAFF,R0
MOV.B      ROH,@STD2      ;Initailize Serial Transmitting Data 2
MOV.B      ROL,@STD3      ;Initialize Serial Transmitting Data 3
;
MOV.B      #H'00,ROL
MOV.B      ROL,@SRD0      ;Initialize Serial Receiving Data 0
MOV.B      ROL,@SRD1      ;Initialize Serial Receiving Data 1
MOV.B      ROL,@SRD2      ;Initialize Serial Receiving Data 2
MOV.B      ROL,@SRD3      ;Initialize Serial Receiving Data 3
;
MOV.W      #H'FB80,R1      ;Initialize Serial Transmitting Data Address
MOV.W      #H'FB90,R2      ;Initialize Serial Receiving Data Address
MOV.B      #H'04,R3L      ;Initialize Serial Transfer Data Counter
;
ANDC      #H'7F,CCR      ;Interrupt Enable
;
MAIN1     .equ           $
BTST      TDRE            ;TDRE = "1" ?
BEQ      MAIN1            ;No.
;
MOV.B      @R1,ROL        ;Load Serial Transmitting Data
MOV.B      ROL,@TDR       ;Save Serial Transmitting Data
;
MAIN2     .equ           $
BTST      OER              ;OER = "1" ?
BNE      MAIN3            ;Yes.
;
BTST      PER              ;PER = "1" ?
BNE      MAIN3            ;Yes.
;
BTST      FER              ;FER = "1" ?
BNE      MAIN3            ;Yes.
;
BTST      RDRF            ;RDRF = "1" ?
BEQ      MAIN2            ;No.
BRA      MAIN4
;
MAIN3     .equ           $
JSR      @ERSUB
BRA      MAIN5
;
MAIN4     .equ           $
MOV.B      @RDR,ROL        ;Load Serial Receiving Data
MOV.B      ROL,@R2         ;Save Serial Receiving Data
;

```

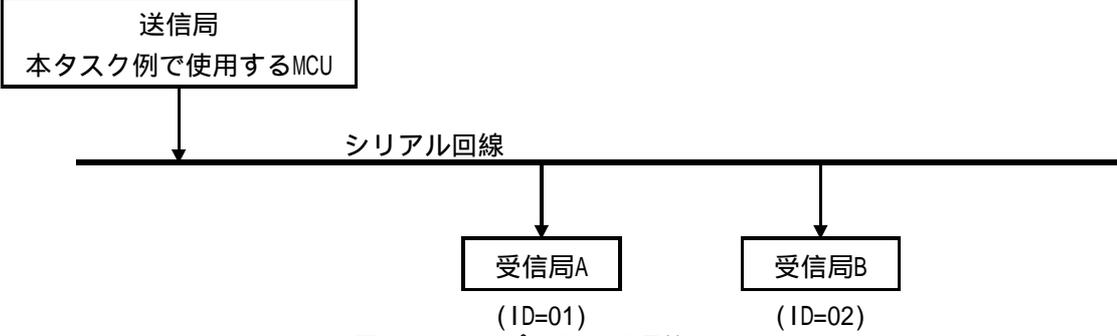
プログラムリスト

```

MAIN5      .equ          $
           ADDS          #1,R1      ;Increment Serial Transmitting Data Address
           ADDS          #1,R2      ;Increment Serial Receiving Data Address
           DEC           R3L        ;Decremrnt Serial Transfer Data Counter
           BNE          MAIN1      ;Serial Transfer Data Counter = H'00 ?
;
MAIN6      .equU         $
           BTST         TEND        ;TEND = "1" ?
           BEQ          MAIN6      ;No.
;
           BCLR         P22        ;Clear P22
           BSET         PCR22       ;Set P22 Terminal Function
;
           BCLR         TE         ;Clear TE
           BCLR         RE         ;Clear RE
;
MAIN9      .equ          $
           BRA          MAIN9
;
;*****
;          Errorr Subroutine
;*****
;
ERSUB      .equ          $
           BTST         OER         ;OER = "1" ?
           BEQ          ERSUB1     ;No.
;
           BCLR         OER        ;Clear OER
;
ERSUB1     .equ          $
           BTST         FER         ;FER = "1" ?
           BEQ          ERSUB3     ;No.
;
           BTST         P22        ;RXD = "1" ?
           BEQ          ERSUB2     ;Yes
;
           BCLR         FER        ;Clear FER
           BRA          ERSUB3
;
ERSUB2     .equ          $
           MOV.B        #H'01,R2L  ;Set Serial Receiving Data Counter
;
ERSUB3     .equ          $
           BTST         PER         ;PER = "1" ?
           BEQ          ERSUB4     ;No.
;
           BCLR         PER        ;Clear PER
;
ERSUB4     .equ          $
           RTS
;
           .end

```

2.20 マルチプロセッサ通信

マルチプロセッサ通信	使用機能	SCI3 : マルチプロセッサ通信機能
仕様		
<p>(1) 図1に示すようにマルチプロセッサ通信機能を使用して、受信局AにデータH'B8を受信局BにデータH'DEをそれぞれ送信します。</p> <p>(2) 送信データの通信フォーマットは、データ長が8ビット、マルチプロセッサビット1ビット、ストップビット長が1ビットに設定します。</p> <p>(3) ビットレートは31250 (bit/s) で送信します。またデータ送信の終了時にブレークを出力します。</p> <p style="text-align: center;">H8/3644</p> <div style="text-align: center;">  </div> <p style="text-align: center;">図1 マルチプロセッサ通信</p>		
使用機能説明		
<p>(1) 本タスク例では、シリアルコミュニケーションインタフェース (SCI : Serial Communication Interface) を使用して、マルチプロセッサ通信の送信を行ないます。図3にマルチプロセッサ送信のブロック図を示します。以下にマルチプロセッサ通信のブロック図について説明します。</p> <ul style="list-style-type: none"> ・マルチプロセッサ通信機能とは、調歩同期式モードでマルチプロセッサビットを付加したフォーマット (マルチプロセッサフォーマット) でシリアルデータ通信を行うことにより、複数のプロセッサ間で通信回線を共有してデータの送受信を行う機能です。 ・マルチプロセッサ通信を行うとき、受信局は各々固有のIDコードが割り付けられています。シリアル通信サイクルは、受信局を指定するID送信サイクルと指定された受信局へ通信データを送信するデータ送信サイクルの2つから構成されます。 ・このID送信サイクルとデータ送信サイクルの区別は、マルチプロセッサビットで行います。マルチプロセッサビットが"1"のときID送信サイクル、"0"のときデータ送信サイクルとなります。 ・送信局は、まずシリアルデータ通信を行いたい受信局のIDコードに、マルチプロセッサビット"1"を付加した通信データを送信します。続いて、送信データにマルチプロセッサビット"0"を付加した通信データを送信します。 ・受信局は、マルチプロセッサビットが"1"の通信データを、自局のIDと比較し一致した場合は続いて送信される通信データを受信します。一致しなかった場合は再びマルチプロセッサビットが"1"の通信データが送信されるまで通信データを読み飛ばします。 ・送信/受信フォーマットは4種類を選択できます。マルチプロセッサフォーマットを指定した場合は、パリティビットの指定は無効です。 ・独立した送信部と受信部を備えているので、送信と受信を同時に行なうことができます。また、送信部および受信部ともにダブルバッファ構造になっているため、連続送信・連続受信ができます。 ・内蔵のボーレートジェネレータで任意のビットレートを選択可能です。 ・送受信クロックソースを内部クロック、または外部クロックから選択可能です。 ・割り込み要因には送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割り込み要因があります。 ・レシーブシフトレジスタ (RSR) は、シリアルデータを受信するためのレジスタです。RSRにRXD端子から入力されたシリアルデータを、LSB (ビット0) から受信した順にセットしパラレルデータに変換します。1バイトのデータを受信すると、データは自動的にRDRへ転送されます。CPUからRSRを直接リード/ライトすることはできません。 ・レシーブデータレジスタ (RDR) は、受信したシリアルデータを格納する8ビットのレジスタです。1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送し、受信動作を完了します。その後、RSRは受信可能となります。RSRとRDRはダブルバッファになっているため連続した受信動作が可能です。RDRは受信専用レジスタなのでCPUからライトできません。 ・トランスミットシフトレジスタ (TSR) は、シリアルデータを送信するためのレジスタです。TDRから送信データをいったんTSRに転送し、LSB (ビット0) から順にTXD端子に送出することでシリアルデータ送信を行ないます。1バイトのデータを送信すると、自動的にTDRからTSRへ次の送信データを転送し、送信を開始します。ただし、TDRにデータが書き込まれていない (TDREに"1"がセットされている) 場合にはTDRからTSRへのデータ転送は行ないません。CPUからTSRを直接リード/ライトすることはできません。 		

使用機能説明

- ・トランスミットデータレジスタ (TDR) は、送信データを格納する8ビットのレジスタです。TSRの"空"を検出すると、TDRに書き込まれた送信データをTSRに転送し、シリアルデータ送信を開始します。TSRのシリアルデータ送信中に、TDRに次の送信データをライトしておく、連続送信が可能です。TDRは、常にCPUによるリード/ライトが可能です。
- ・シリアルモードレジスタ (SMR) は、シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。SMRは、常にCPUによるリード/ライトが可能です。
- ・シリアルコントロールレジスタ3 (SCR3) は、送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止、および送信/受信クロックソースの選択を行なう8ビットのレジスタです。SCR3は、常にCPUによるリード/ライトが可能です。
- ・図2にマルチプロセッサフォーマットを使用したシリアルデータを、表1にマルチプロセッサフォーマットを示します。

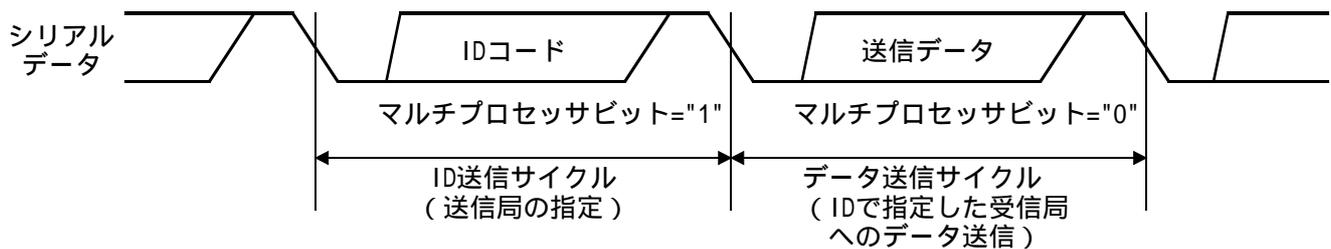


図2 マルチプロセッサビットを使用したシリアルデータ

表1 マルチプロセッサフォーマット

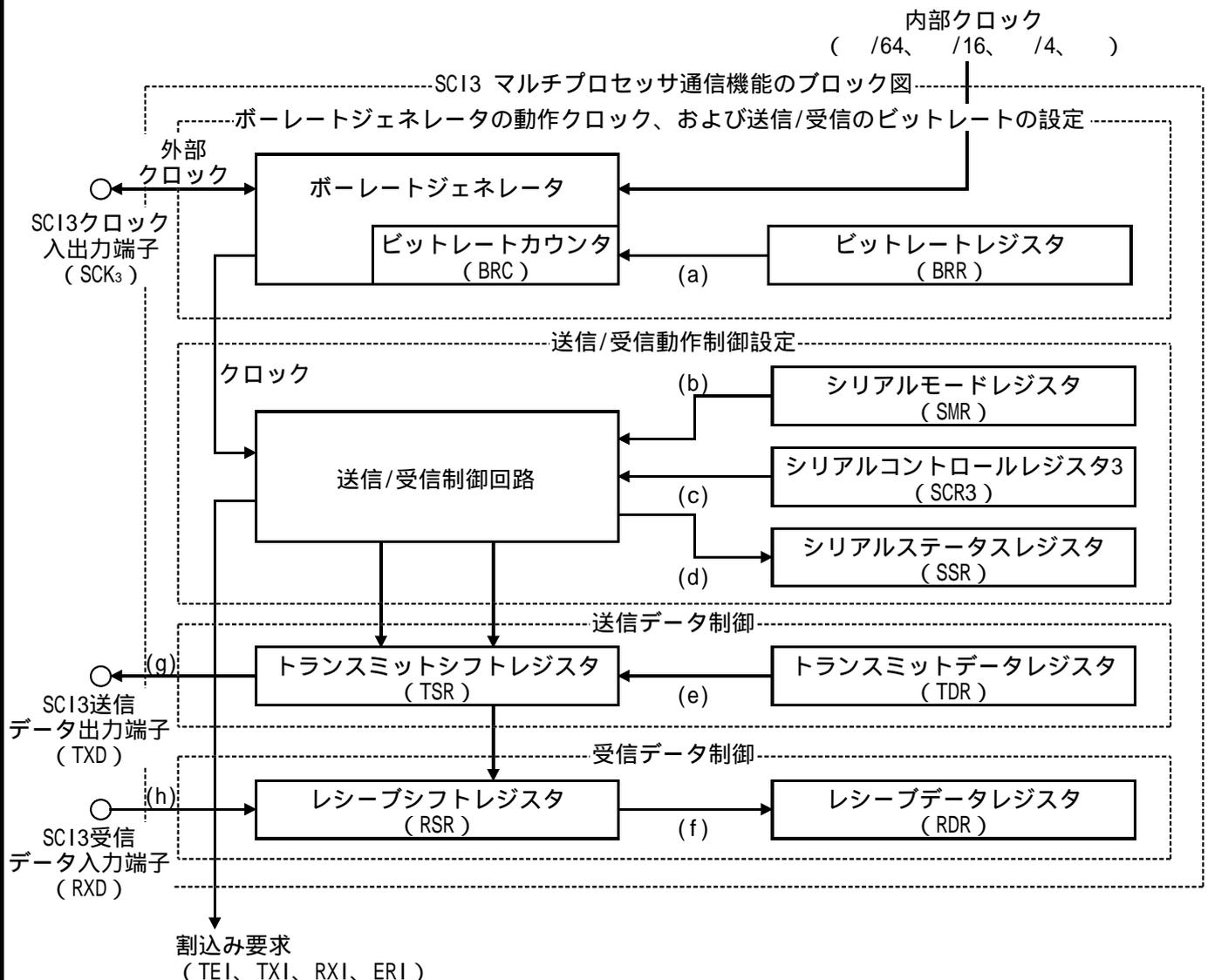
MSR				シリアル通信フォーマットとフレーム長												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	*	1	0	S	8ビットデータ								MPB	STOP		
0	*	1	1	S	8ビットデータ								MPB	STOP	STOP	
1	*	1	0	S	7ビットデータ							MPB	STOP			
1	*	1	1	S	7ビットデータ							MPB	STOP	STOP		

* : Don't care

【記号説明】

- S : スタートビット
- STOP : ストップビット
- MPB : マルチプロセッサビット

使用機能説明



- 【注】 (a) SMRで選択されるポレートジェネレータの動作クロックと合わせて、送信/受信のビットレートを設定します。本タスク例では、送信のビットレートを31250 (bit/s) に設定しています。
- (b) シリアルデータ通信フォーマットの設定と、ポレートジェネレータのクロックソースを選択します。本タスク例でシリアルデータ通信フォーマットは、動作モードを調歩同期式に、データ長を8ビットに、パリティビットの付加およびチェックを禁止、マルチプロセッサ通信機能を許可、ストップビット長を1ビットに、内蔵ポレートジェネレータのクロックソースをクロックに設定しています。
- (c) 送信/受信動作、調歩同期式モードでのクロック出力、割り込み要求の許可/禁止を選択します。本タスク例では、調歩同期式モードでのクロック出力は、コミュニケーションモードを調歩同期式に、クロックソースを内部クロックに、SCK₃端子機能をクロック出力に設定しています。また、割り込み要求の許可/禁止は送信データエンpty割り込みを禁止に、受信データフル割り込み要求を禁止に設定しています。
- (d) ステータスフラグ (トランスミットデータレジスタエンpty、レシーブデータレジスタフル、オーバーランエラー、フレーミングエラー、パリティエラー、トランスミットエンド) によりSCI3の動作状態を示す。
- (e) TSRの"空"を検出することにより、TDRに書き込まれた送信データをTSRに転送。
- (f) 1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送。
- (g) 送信データ。
- (h) 受信データ。

図3 マルチプロセッサ通信機能のブロック図

使用機能説明

- ・シリアルステータスレジスタ (SSR) は、SCI3の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8ビットのレジスタです。SSRは常にCPUからリード/ライトできません。ただし、TDRE、RDRF、OER、PER、FERへ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。また、TENDおよびMPBRはリード専用であり、ライトすることはできません。
- ・ビットレートレジスタ (BRR) は、SMRのCKS1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて、送信/受信のビットレートを設定する8ビットのレジスタです。BRRは常にCPUによるリード/ライトが可能です。
- ・表2に、調歩同期式モードのBRRの設定例を示します。表1はアクティブ (高速) モードで、OSCが10MHzのときの値を示しています。

表2 ビットレートに対するBRRの設定例 (調歩同期式モード)

Rビットレート (bit/s)	110	150	300	600	1200	2400	4800	9600	19200	31250	38400
n	2	2	1	0	0	0	0	0	0	0	0
N	88	64	129	64	129	64	32	15	7	4	3
誤差 (%)	-0.25	+0.16	+0.16	+0.16	+0.16	+0.16	-1.36	+1.73	+1.73	0	+1.73

- 【注】 1. 誤差は、1%以内となるように設定します。
2. BRRの設定値は以下の計算式で求められます。

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^6 - 1$$

B : ビットレート (bit/s)

N : ボーレートジェネレータのBRRの設定値 (0 N 255)

OSC : oscの値 (MHz) = 10MHz

n : ボーレートジェネレータの入力クロックのNo. (n=0,1,2,3)
(nとクロックの関係は表3を参照)

表3 nとクロックの関係

n	クロック	SMRの設定値	
		CKS1	CKS0
0		0	0
1	/4	0	1
2	/16	1	0
3	/64	1	1

3. 表2に誤差は以下の計算式で求めた値を小数点第3位を四捨五入して表示してあります。

$$\text{誤差 (\%)} = \frac{B(n, N, OSC \text{から求めたレート}) - R(\text{表2の上欄のビットレート})}{R(\text{表2の上欄のビットレート})} \times 100$$

4. OSCが10MHzのときの最大ビットレート (調歩同期式モード) は、156250 (bit/s) になります。ただし、設定値は、n=0、N=0のときです。

- ・調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信/受信し、1キャラクタ単位で同期をとりながらシリアル通信を行なうモードです。
- ・SCI3内部では、送信部と受信部は独立しているので、全二重通信を行なうことができます。また、送信部と受信部がともにダブルバッファ構造になっているので、送信中にデータのライト、受信中にデータのリードができ、連続送信/受信が可能です。
- ・調歩同期式通信では、通信回線は通常マーク状態 ("High" レベル) に保たれています。SCI3では通信回線を監視し、スペース ("Low" レベル) になったところをスタートビットとみなしてシリアル通信を開始します。
- ・通信データの1キャラクタはスタートビット ("Low" レベル) から始まり、送信/受信データ (LSBファースト: 最下位ビットから)、パリティビット ("High" または "Low" レベル)、最後にストップビット ("High" レベル) の順で構成されます。
- ・調歩同期式モードでは、受信時にスタートビットの立ち上がりエッジで同期化を行いません。また、データを1ビット期間の16倍の周波数のクロックの8番目でサンプリングするので、各ビットの中央で通信データを取り込みます。

使用機能説明

- SCI3クロック (SCK₃) は、SCI3のクロック入出力端子です。
- SCI3レシーブデータ入力 (RXD) は、SCI3の受信データ入力端子です。
- SCI3トランスミットデータ出力 (TXD) は、SCI3の送信データ出力端子です。
- SCI3の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび3種類の受信エラー (オーバーランエラー、フレーミングエラー、パリティエラー) の計6種類があり、共通のベクタアドレスが割り付けられています。
- 各割込み要求は、SCR3のTIE、RIEで許可/禁止できます。
- SSRのTDREが"1"にセットされるとTXIが発生します。SSRのTENDが"1"にセットされると、TEIが発生します。この2つの割込みは送信時に発生します。
- SSRのTDREは初期値が"1"になっています。したがって送信データをTDRへ転送する前にSCR3のTIEを"1"にセットして送信データエンプティ割込み要求 (TXI) を許可すると、送信データが準備されていなくてもTXIが発生します。
- SSRのTENDは初期値が"1"になっています。したがって、送信データをTDRへ転送する前にSCR3のTEIEを"1"にセットして送信終了割込み要求 (TEI) を許可すると、送信データが送信されていなくてもTEIが発生します。
- 送信データをTDRへ転送する処理を割込み処理ルーチンの中で行なうようにすることで、これらの割込みを有効に利用できます。また、これらの割込み要求 (TXI、TEI) の発生を防ぐためには、送信データをTDRへ転送した後に、これらの割込み要求に対応する許可ビット (TIE、TEIE) を"1"にセットします。
- SSRのRDRFが"1"にセットされるとRXIが発生します。OER、PER、FERのいずれかが"1"にセットされるとERIが発生します。この2つの割込み要求は受信時に発生します。

(2) 表4に本タスク例の機能割付けを示します。表3に示すように機能を割り付け、マルチプロセッサ通信機能による送信を行ないます。

表4 機能割付け

機能	機能割付け
TSR	シリアルデータを送信するためのレジスタ
TDR	送信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット、ボーレートジェネレータのクロックソースの設定
SSR	SCI3の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
TXI	送信データエンプティ (TDRE) による割込み要求
TEI	送信終了 (TEND) による割込み要求
PMR7	TXD出力端子設定
SCK ₃	SCI3のクロック出力端子
TXD	SCI3の送信データ出力端子

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理によりマルチプロセッサ通信機能による送信を行います。

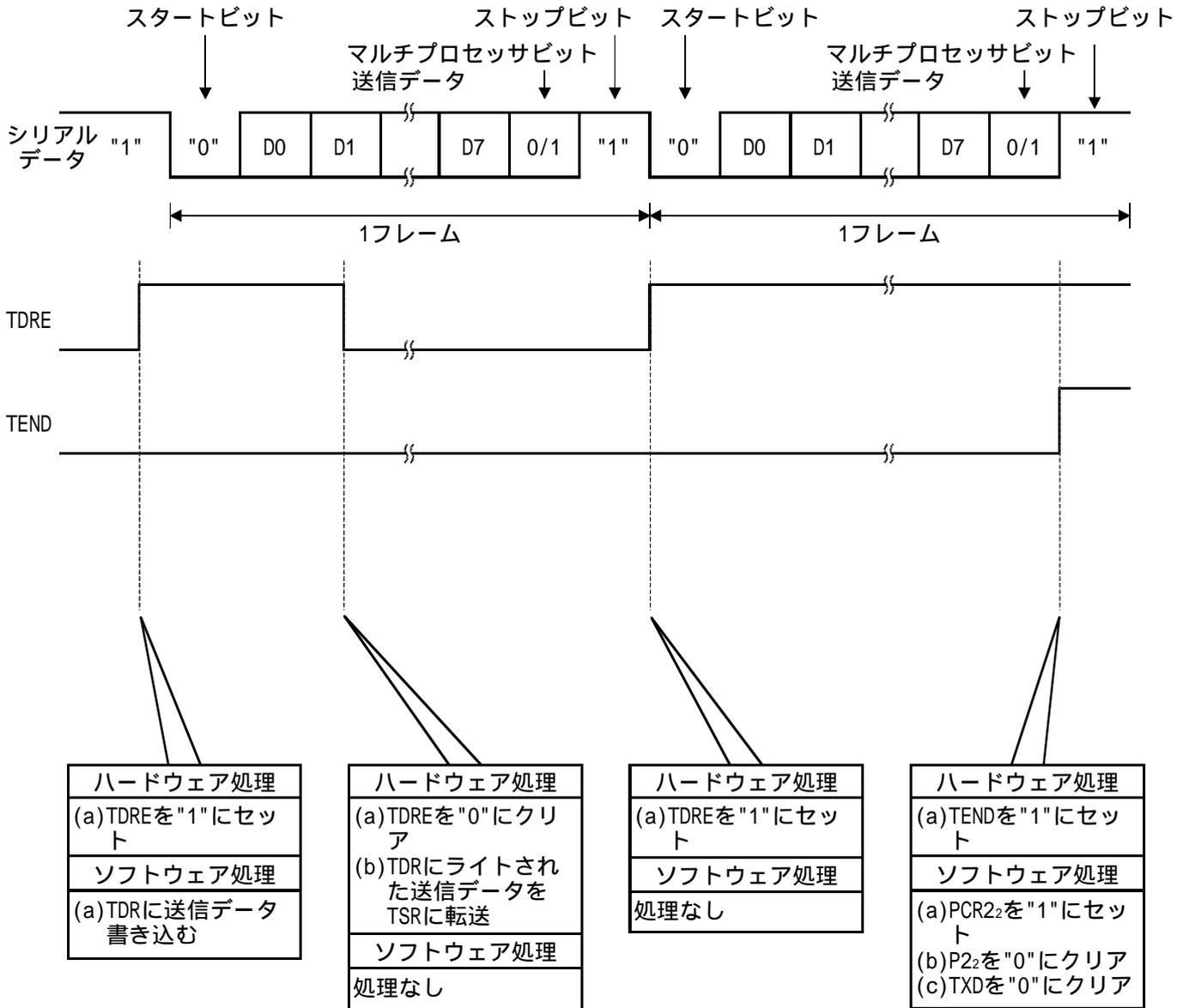


図4 マルチプロセッサ通信機能による送信の動作原理

ソフトウェア説明

(1) モジュール説明

表5に本タスク例におけるモジュール説明を示します。

表5 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、転送データの設定、マルチプロセッサ通信機能の設定、割込みの許可、4バイトのデータを送信したところで終了

(2) 引数の説明

表6に本タスク例で使用する引数を示します。

表6 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0 ~ STD3	調歩同期式シリアル送信データ	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表7に本タスク例における使用内部レジスタ説明を示します。

表7 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SMR	COM	シリアルモードレジスタ (コミュニケーションモード) : COM="0" のとき、コミュニケーションモードを調歩同期式モードに設定	H'FFA8 ビット7	0
	CHR	シリアルモードレジスタ (キャラクタレングス) : CHR="0" のとき、調歩同期式モード時におけるデータ長を8ビットデータに設定	H'FFA8 ビット6	0
	PE	シリアルモードレジスタ (パリティイネーブル) : PE="0" のとき、調歩同期式モードで、送信時にパリティビットの付加およびチェックを禁止	H'FFA8 ビット5	0
	STOP	シリアルモードレジスタ (ストップビットレングス) : STOP="0" のとき、調歩同期式モードでのストップビットの長さを1ビットに設定	H'FFA8 ビット3	0
	MP	シリアルモードレジスタ (マルチプロセッサモード) : MP="1" のとき、マルチプロセッサ通信機能を許可	H'FFA8 ビット2	1
	CKS1 CKS0	シリアルモードレジスタ (クロックセレクト1、0) : CKS1="0"、CKS0="0" のとき、内蔵ボーレートジェネレータのクロックソースをクロックに設定	H'FFA8 ビット1 ビット0	CKS1="0" CKS0="0"
BRR	ビットレートレジスタ : BRR=H'04 のとき、SMRのCKS1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて送信のビットレートを31250 (bit/s) に設定	H'FFA9	H'04	
SCR3	TIE	シリアルコントロールレジスタ3 (トランスミットインタラプトイネーブル) : TIE="0" のとき、送信データエンプティ割込み要求 (TXI) を禁止	H'FFAA ビット7	0

ソフトウェア説明

表7 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SCR3	TE	シリアルコントロールレジスタ3 (トランスミットイネーブル) : TE="0"のとき、送信動作を禁止 (TXD端子はトランスミットデータ端子)	H'FFAA ビット5	0
	TEIE	シリアルコントロールレジスタ3 (トランスミットエンドインタラプトイネーブル) : TEIE="0"のとき、送信終了割り込み要求を禁止	H'FFAA ビット2	0
	CKE1 CKE0	シリアルコントロールレジスタ3 (クロックイネーブル1、0) : CKE1="0"、CKE0="1"のとき、調歩同期式モードにおいてクロックソースを内部クロック、SCK ₃ 端子機能をクロック出力に設定	H'FFAA ビット1 ビット0	CKE1="0" CKE0="1"
TDR	トランスミットデータレジスタ : 送信データを格納する8ビットのレジスタ	H'FFAB	-	
SSR	TDRE	シリアルステータスレジスタ (トランスミットデータエンpty) : TDRE="0"のとき、TDRにライトされた送信データがTSRに転送されていないことを示す : TSRE="1"のとき、TDRに送信データがライトされていなく、またはTDRにライトされた送信データがTSRに転送されたことを示す	H'FFAC ビット7	1
	TEND	シリアルステータスレジスタ (トランスミットエンド) : TEND="0"のとき、送信中であることを示す : TEND="1"のとき、送信を終了したことを示す	H'FFAC ビット2	1
	MPBR	シリアルステータスレジスタ (マルチプロセッサビットレシーブ) : MPBR="0"のとき、マルチプロセッサビットが"0"のデータを受信した : MPBR="1"のとき、マルチプロセッサビットが"1"のデータを受信した	H'FFAC ビット1	0
	MPBT	シリアルステータスレジスタ (マルチプロセッサビットトランスファ) : MPBT="0"のとき、マルチプロセッサビット"0"を送信 : MPBT="1"のとき、マルチプロセッサビット"1"を送信	H'FFAC ビット0	0
PDR2	P ₂	ポートデータレジスタ2 (ポートデータレジスタ22) : P ₂ ="0"のとき、P ₂ 端子の出力レベルは"Low" : P ₂ ="1"のとき、P ₂ 端子の出力レベルは"High"	H'FFD5 ビット2	0
PCR2	PCR ₂	ポートコントロールレジスタ2 (ポートコントロールレジスタ22) : PCR ₂ ="0"のとき、P ₂ 端子を入力端子機能に設定 : PCR ₂ ="1"のとき、P ₂ 端子を出力端子機能に設定	H'FFE5 ビット2	1
PMR7	TXD	ポートモードレジスタ7 (P ₂ /TXD端子機能切り替え) : TXD="1"のとき、P ₂ /TXD端子をTXD端子機能に設定	H'FFFF ビット2	1

(4) 使用RAM説明

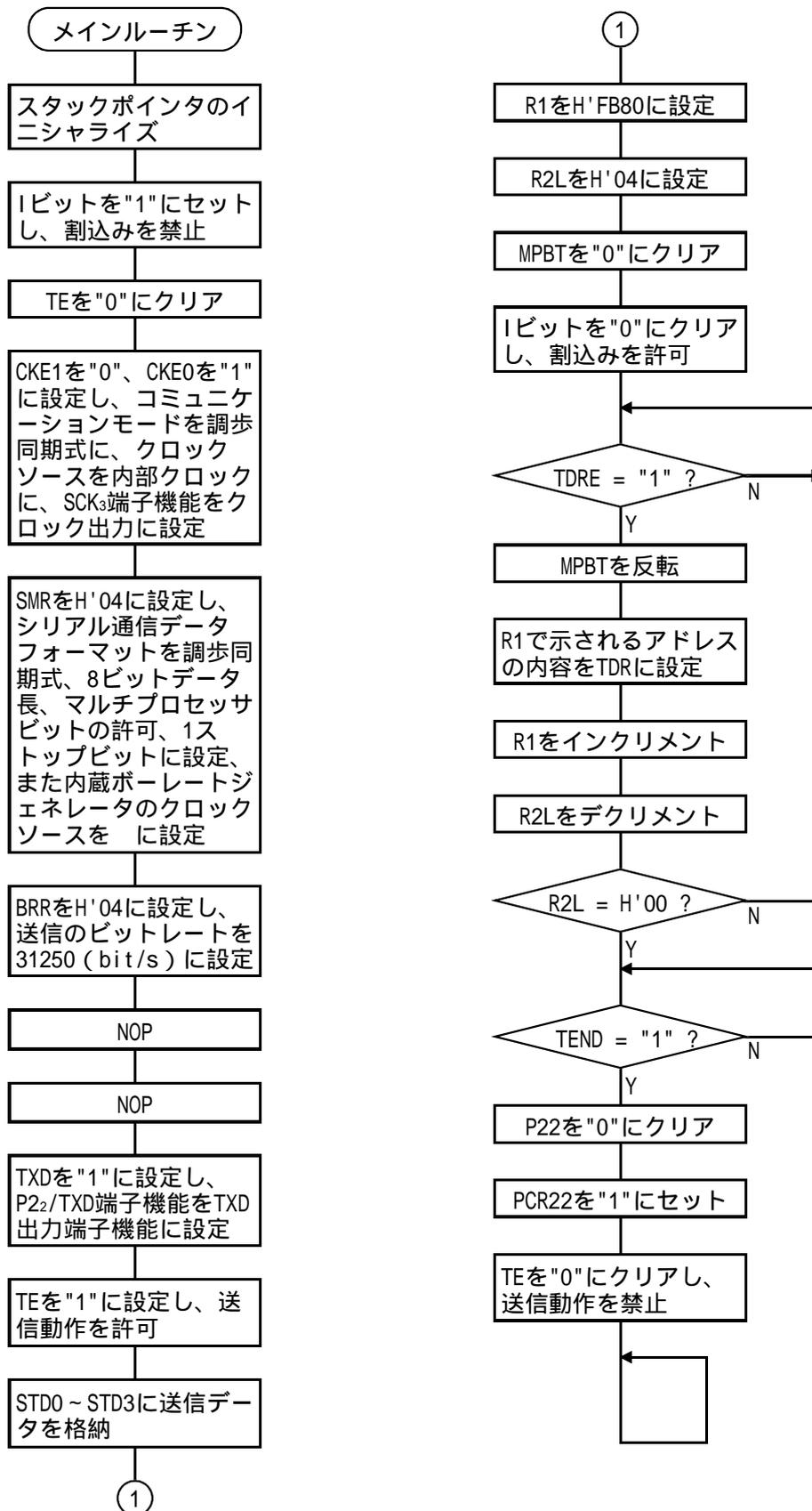
表8に本タスク例における使用RAM説明を示します。

表8 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0	調歩同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	調歩同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	調歩同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	調歩同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'MultiProcessor Communications'
;
;       Function
;       : Serial Communication Interface
;       -Multi-Processor Communication
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
SMR          .equ          H'FFA8          ;Serial Mode Register
COM          .bequ        7,SMR          ;Communication Mode
CHR          .bequ        6,SMR          ;Character Length
PE          .bequ        5,SMR          ;Parity Enable
PM          .bequ        4,SMR          ;Parity Mode
STOP        .bequ        3,SMR          ;Stop Bit Length
MP          .bequ        2,SMR          ;Multiprocesor Mode
CKS1        .bequ        1,SMR          ;Clock Select 1
CKS0        .bequ        0,SMR          ;Clock Select 0
BRR          .equ          H'FFA9          ;Bit Rate Register
SCR3        .equ          H'FFAA          ;Serial Control Register 3
TIE         .bequ        7,SCR3          ;Transmit Interrupt Enable
RIE         .bequ        6,SCR3          ;Receive Interrupt Enable
TE          .bequ        5,SCR3          ;Transmit Enable
RE          .bequ        4,SCR3          ;Receive Enable
MPIE        .bequ        3,SCR3          ;Multiprocessor Interrupt Enable
TEIE        .bequ        2,SCR3          ;Transmit End Interrupt Enable
CKE1        .bequ        1,SCR3          ;Clock Enable 1
CKE0        .bequ        0,SCR3          ;Clock Enable 0
TDR          .equ          H'FFAB          ;Transmit Data Register
SSR          .equ          H'FFAC          ;Serial Status Register
TDRE        .bequ        7,SSR          ;Transmit Data Register Empty
RDRF        .bequ        6,SSR          ;Receive Data Register Full
OER         .bequ        5,SSR          ;Overrun Errorr
FER         .bequ        4,SSR          ;Framing Errorr
PER         .bequ        3,SSR          ;Parity Errorr
TEND        .bequ        2,SSR          ;Transmit End
MPBR        .bequ        1,SSR          ;Multiprocessor Bit Receive
MPBT        .bequ        0,SSR          ;Multiprocessor Bit Transfer
RDR          .equ          H'FFAD          ;Receive Data Register
PDR2        .equ          H'FFD5          ;Port Data Register 2
P22         .bequ        2,PDR2          ;Port Data Register 22
PCR2        .equ          H'FFE5          ;Port Control Register 2
PCR22       .bequ        2,PCR2          ;Port Control Register 22
PMR7        .equ          H'FFFF          ;Port Mode Register 7

```

プログラムリスト

```

TXD          .bequ      2,PMR7          ;P22/TXD Terminal Function Switch
;
;*****
;
;          RAM Allocation
;*****
;
;
STACK        .equ       H'FF80         ;Stack Pointer
STD0         .equ       H'FB80         ;Serial Transmitting Data 0
STD1         .equ       H'FB81         ;Serial Transmitting Data 1
STD2         .equ       H'FB82         ;Serial Transmitting Data 2
STD3         .equ       H'FB83         ;Serial Transmitting Data 3
;
;*****
;
;          Vector Address
;*****
;
;
;          .org          H'0000
;          .data.w      MAIN           ;Reset Interrupt
;
;
;          .org          H'0008
;          .data.w      MAIN           ;IRQ0 Interrupt
;          .data.w      MAIN           ;IRQ1 Interrupt
;          .data.w      MAIN           ;IRQ2 Interrupt
;          .data.w      MAIN           ;IRQ3 Interrupt
;          .data.w      MAIN           ;INT0 - INT7 Interrupt
;
;
;          .org          H'0014
;          .data.w      MAIN           ;Timer A Interrupt
;          .data.w      MAIN           ;Timer B1 Interrupt
;
;
;          .org          H'0020
;          .data.w      MAIN           ;Timer X Interrupt
;          .data.w      MAIN           ;Timer V Interrupt
;
;
;          .org          H'0026
;          .data.w      MAIN           ;SCI1 Interrupt
;
;
;          .org          H'002A
;          .data.w      MAIN           ;SCI3 Interrupt
;          .data.w      MAIN           ;A/D Converter Interrupt
;          .data.w      MAIN           ;SLEEP Instruction Executed Interrupt
;
;*****
;
;          Main Program
;*****
;
;
;          .org          H'1000
;
;MAIN        .equ       $
;          MOV.W        #STACK,SP     ;Initialize Stack Pointer
;          ORC          #H'80,CCR     ;Interrupt Disable
;
;
;          MOV.W        #H'0104,R0
;          MOV.B        R0H,@SCR3    ;Clear TE & RE and Initialize Clock Souce
;          MOV.B        R0L,@SMR     ;Initialize Transmit Format
;
;
;          MOV.B        #H'04,R0L
;          MOV.B        R0L,@BRR     ;Initialize Transmit Bit Rate
;
;
;          NOP                    ;1bit Wait

```

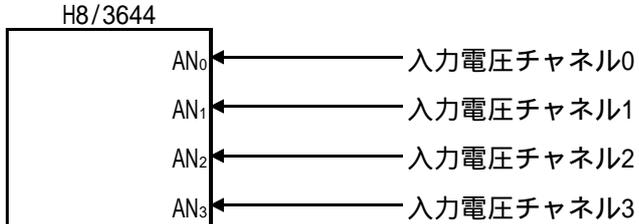
プログラムリスト

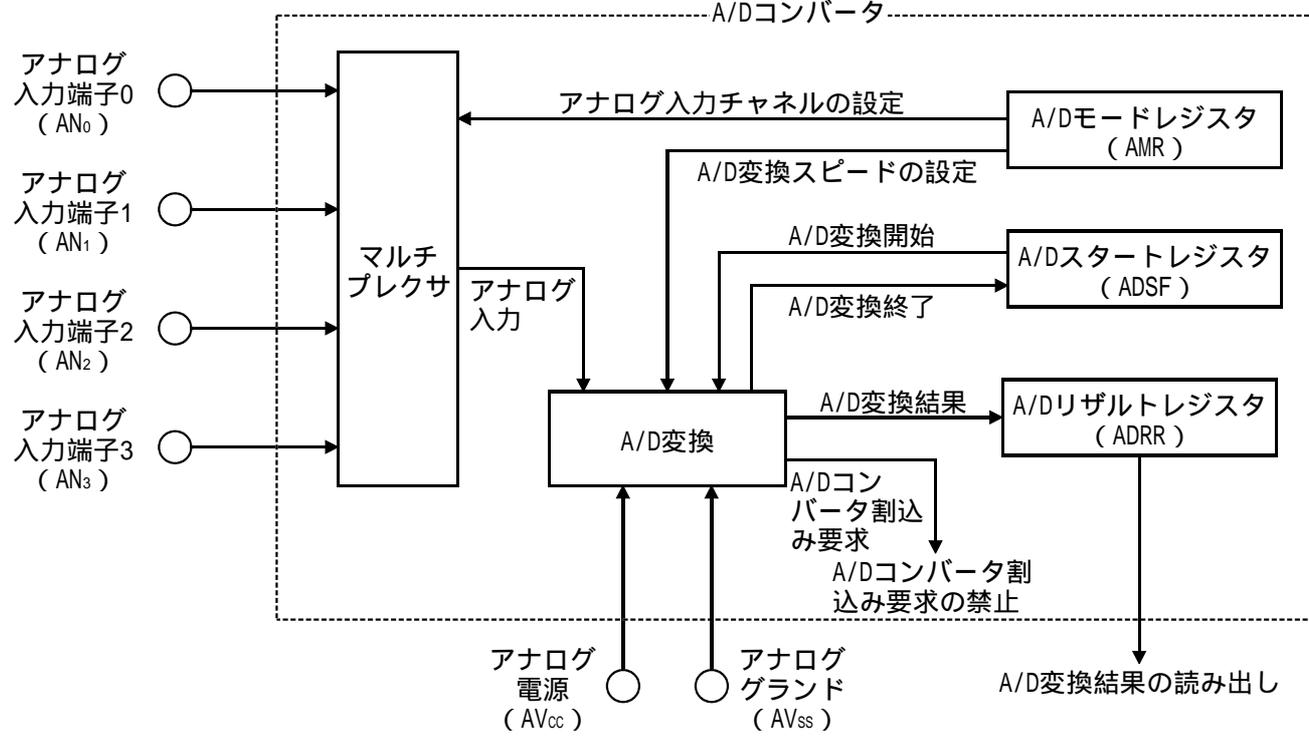
```

;
;      NOP
;
;      BSET      TXD      ;Initialize TXD Terminal Function
;      BSET      TE       ;Transmit Enable
;
;      MOV.W     #H'01B8,R0
;      MOV.B     ROH,@STD0 ;Set Serial Transmitting Data 1
;      MOV.B     ROL,@STD1 ;Set Serial Transmitting Data 2
;      MOV.W     #H'02DE,R0
;      MOV.B     ROH,@STD2 ;Set Serial Transmitting Data 3
;      MOV.B     ROL,@STD3 ;Set Serial Transmitting Data 4
;
;      MOV.W     #H'FB80,R1 ;Initialize Serial Transmitting Data Address
;      MOV.B     #H'04,R2L  ;Initialize Serial Transmitting Data Counter
;      BCLR     MPBT      ;Clear Multiprocessor Bit Transfer
;
;      ANDC     #H'7F,CCR  ;Interrupt Enable
;
;
; MAIN1
;      .equ     $
;      BTST     TDRE      ;TDRE = "1" ?
;      BEQ     MAIN1     ;No.
;
;      BNOT     MPBT      ;Invert Multiprocessor Bit Transfer
;
;      MOV.B     @R1,ROL   ;
;      MOV.B     ROL,@TDR ;Write Serial Transmit Data to TDR
;
;      ADDS     #1,R1     ;Increment Serial Transmitting Data Address
;      DEC     R2L       ;Decrement Serial Transmitting Data Counter
;      BNE     MAIN1
;
;
; MAIN2
;      .equ     $
;      BTST     TEND      ;TEND = "1" ?
;      BEQ     MAIN2     ;No.
;
;      BCLR     P22      ;Clear P22
;      BSET     PCR22    ;Set PCR22
;      BCLR     TXD      ;Set P22 Terminal Function
;
;      BCLR     TE       ;Clear TE
;
;
; MAIN9
;      .equ     $
;      BRA     MAIN9
;
;
;      .end

```

2.21 4チャンネルA/D変換による電圧測定

4チャンネルA/D変換による電圧測定	使用機能	A/Dコンバータ
仕様		
<p>(1) A/Dコンバータを使用して、4チャンネルのA/D変換による電圧測定を行ないます。</p> <p>(2) 図1に示すように、4チャンネルの電圧をH8/3644に入力し、A/D変換した結果をRAMに格納します。</p>		
		
<p align="center">図1 4チャンネルA/D変換による電圧測定</p>		

使用機能説明
<p>(1) 本タスク例では、A/Dコンバータを使用して、4チャンネルA/D変換による電圧測定を行ないます。</p> <p>(a) 図2にA/Dコンバータのブロック図を示します。以下にA/Dコンバータのブロック図について説明します。</p> <ul style="list-style-type: none"> ・A/Dモードレジスタ (AMR) は、8ビットのリード/ライト可能なレジスタで、A/D変換スピードの設定、およびアナログ入力端子の指定を行ないます。本タスク例では、A/D変換スピードを12.4 μsに設定しています。 ・A/Dリザルトレジスタ (ADRR) は、8ビットのリード専用レジスタで、A/D変換された結果を格納します。 ・A/Dスタートレジスタ (ADSR) は、8ビットのリード/ライト可能なレジスタで、A/D変換の開始、または停止を指定します。 ・アナログ入力端子0~3 (AN₀ ~ AN₃) は、入力電圧チャンネル0~3の入力端子です。 ・アナログ電源 (AV_{CC}) は、アナログ部の電源および基準電圧端子です。 ・アナロググランド (AV_{SS}) は、アナログ部のグランドおよび基準電圧端子です。

<p align="center">図2 A/Dコンバータのブロック図</p>

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、4チャンネルA/D変換による電圧測定を行ないます。

表1 機能割付け

機能	機能割付け
AMR	A/D変換スピードの設定、およびアナログ入力端子の指定
ADSR	A/D変換の開始、または停止を指定
ADSR	A/D変換された結果を格納
AN ₀ ~ AN ₃	入力電圧チャンネル0~3の入力端子
AV _{cc}	アナログ部の電源および基準電圧端子
AV _{ss}	アナログ部のグラウンドおよび基準電圧端子

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理により4チャンネルA/D変換による電圧測定を行ないます。

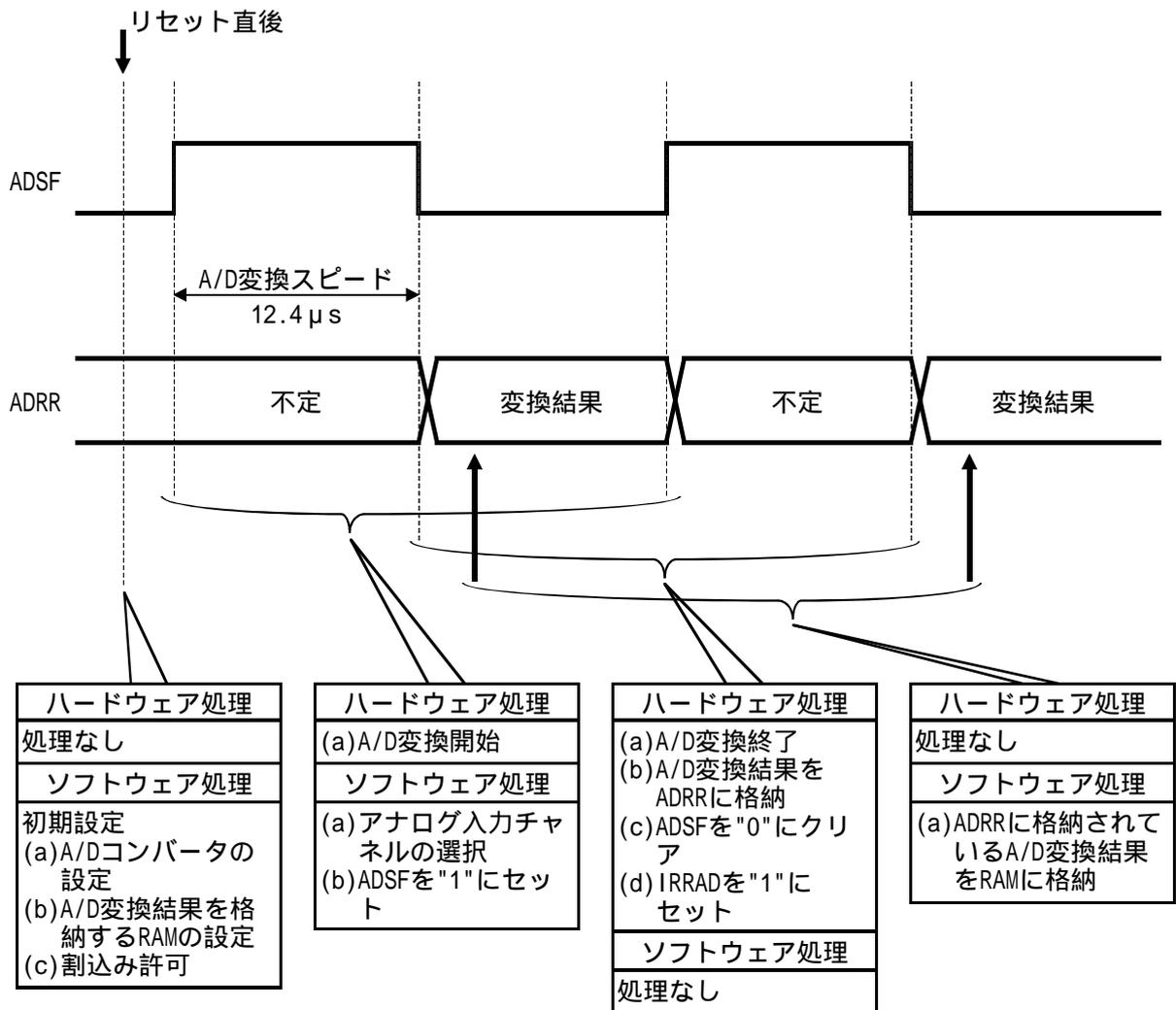


図3 4チャンネルA/D変換による電圧測定の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、A/Dコンバータの設定、割込みの許可、アナログ入力チャンネルの選択、A/D変換の開始、A/D変換終了後ADRRに格納されたA/D変換結果をRAMに格納、アナログ入力チャンネル0~3までのA/D変換を行なうと終了

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
ADCR0	アナログ入力チャンネル0のA/D変換結果を格納	メインルーチン	1バイト	出力
ADCR1	アナログ入力チャンネル1のA/D変換結果を格納	メインルーチン	1バイト	出力
ADCR2	アナログ入力チャンネル2のA/D変換結果を格納	メインルーチン	1バイト	出力
ADCR3	アナログ入力チャンネル3のA/D変換結果を格納	メインルーチン	1バイト	出力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

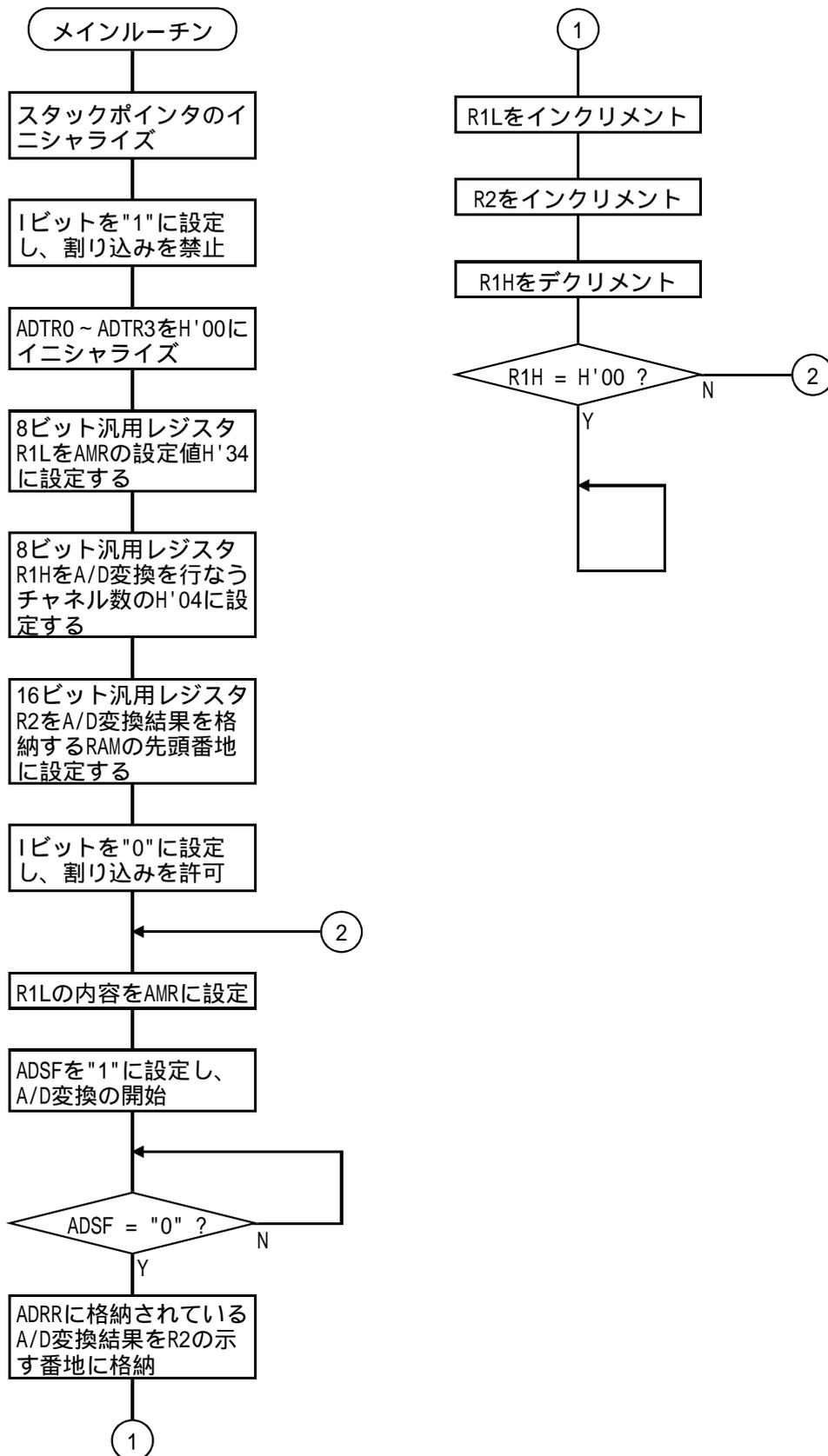
レジスタ名	機能	アドレス	設定値	
AMR	CKS	A/Dモードレジスタ(クロックセレクト) : CKS="0"のとき、A/D変換スピードを12.4 μ sに設定	H'FFC4 ビット7	0
	CH3	A/Dモードレジスタ(チャンネルセレクト3~0) : CH3="0"、CH2="1"、CH1="0"、CH0="0"のとき、AN ₀ を選択 : CH3="0"、CH2="1"、CH1="0"、CH0="1"のとき、AN ₁ を選択 : CH3="0"、CH2="1"、CH1="1"、CH0="0"のとき、AN ₂ を選択 : CH3="0"、CH2="1"、CH1="1"、CH0="1"のとき、AN ₃ を選択	H'FFC4 ビット3	CH3="0"
	CH2		ビット2	CH2="1"
	CH1		ビット1	CH1="0"
CH0	ビット0		CH0="0"	
ADSR	ADSF	A/Dスタートレジスタ(A/Dスタートフラグ) : ADSF="0"のとき、A/D変換の終了 : ADSF="1"のとき、A/D変換の開始	H'FFC6 ビット7	0
ADRR	A/Dリザルトレジスタ : A/D変換結果の8ビットデータを格納	H'FFC5	不定	

(4) 使用RAM説明

本タスク例では、引数以外のRAMは使用しません。

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Voltage Measurement by 4-Channel A/D
;       Converter'
;
;       Function
;       : A/D Converter
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
AMR      .equ        H'FFC4      ;A/D Mode Register
ADRR     .equ        H'FFC5      ;A/D Result Register
ADSR     .equ        H'FFC6      ;A/D Start Register
ADSF     .bequ       7,ADSR      ;A/D Start Flag
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;
STACK    .equ        H'FF80      ;Stack Pointer
ADCRO    .equ        H'FB80      ;A/D Convert Result Data 0
ADCR1    .equ        H'FB81      ;A/D Convert Result Data 1
ADCR2    .equ        H'FB82      ;A/D Convert Result Data 2
ADCR3    .equ        H'FB83      ;A/D Convert Result Data 3
;
;
*****
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.w       MAIN          ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w       MAIN          ;IRQ0 Interrupt
;       .data.w       MAIN          ;IRQ1 Interrupt
;       .data.w       MAIN          ;IRQ2 Interrupt
;       .data.w       MAIN          ;IRQ3 Interrupt
;       .data.w       MAIN          ;INT0 - INT7 Interrupt
;
;
;       .org          H'0014
;       .data.w       MAIN          ;Timer A Interrupt
;       .data.w       MAIN          ;Timer B1 Interrupt
;
;
;       .org          H'0020
;       .data.w       MAIN          ;Timer X Interrupt

```


2.22 14ビットPWM機能によるデューティパルス出力

14ビットPWM機能によるデューティパルス出力	使用機能	14ビットPWM機能
仕様		
<p>(1) 図1に示すように、14ビットPWM機能を使用して、PWM出力端子よりデューティパルスを出します。</p> <p>(2) 本タスク例では、パルス周期が102.4μs、パルスHigh幅が77.2μs、デューティ75.4%のデューティパルスを出します。</p>		
図1 14ビットPWM機能によるデューティパルス出力		

使用機能説明
<p>(1) 本タスク例では、14ビットPWM機能を使用して、PWM出力端子よりデューティパルスを出します。</p> <p>(a) 図2に14ビットPWM機能のブロック図を示します。以下に14ビットPWM機能のブロック図について説明します。</p> <ul style="list-style-type: none"> ・システムクロック () は、10MHzのOSCクロックを2分周した5MHzクロックで、CPUおよび周辺機能を動作させるための基準クロックです。 ・PWMコントロールレジスタ (PWCR) は、8ビットのライト専用レジスタで、入力クロックの選択を行ないます。1変換周期 32.768/、最小変化幅 2/、または1変換周期 16.384/、最小変化幅 1/ の2種の変換周期より選択が可能です。 ・リップル低減をはかったパルス分割方式を使用しています。 ・PWMデータレジスタU、L (PWDRU、PWDRL) は、ライト専用の14ビットのレジスタで、PWDRUが上位6ビット、PWDRLが下位8ビットの構成になっています。PWDRU、PWDRLに書き込まれた内容はPWM波形1周期の"High"レベル幅の合計に対応しています。PWDRU、PWDRLに14ビットのデータをライトするとPWDRU、PWDRLの内容がPWM波形生成部に取り込まれ、PWM波形生成のデータの更新行われます。また、14ビットデータの設定は必ず、PWDRUへの下位8ビットのデータのライト、PWDRLへの上位6ビットのデータのライトの順で行なわなければなりません。 ・ポートモードレジスタ1 (PMR1) は、8ビットのリード/ライト可能なレジスタで、ポート1の各端子機能の切替えを制御します。PMR1ビット4のP14/PWM端子機能切替え (PWM) により、P14/PWM端子をPWM出力端子機能に設定します。 ・PWM出力端子 (PWM) より、パルス分割方式PWM波形を出します。 <p>【注意】本タスク例で使用している14ビットPWM機能を使用してPWM波形を出力する場合、PWMデータレジスタを書き換えるタイミングによって、正常なPWM波形が出力されない場合があります。</p>
図2 14ビットPWM機能ブロック図

使用機能説明

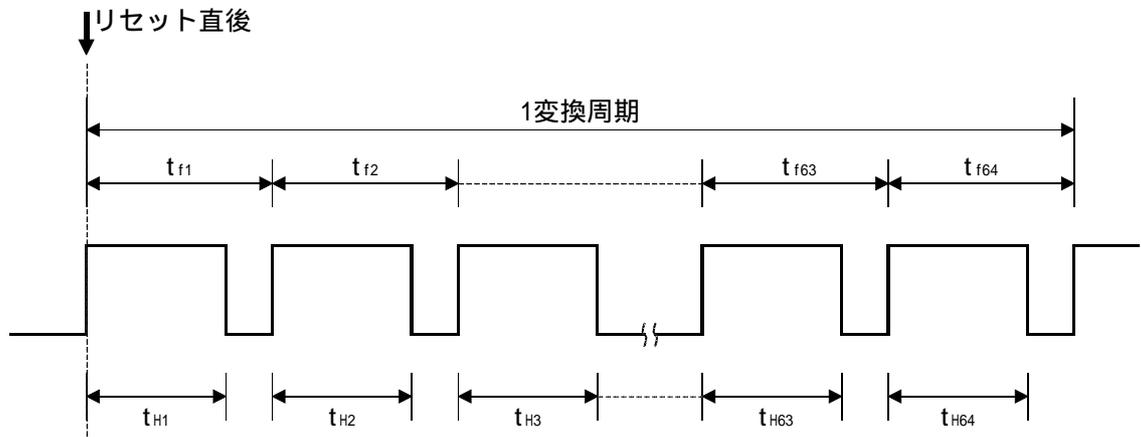
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、14ビットPWM機能によるデューティパルス出力を行ないます。

表1 機能割付け

機能	機能割付け
PWCR	14ビットPWMに供給されるクロックを選択
PWDRU	PWM出力波形データの上位6ビットの設定
PWDRL	PWM出力波形データの低位8ビットの設定
PWM	PWM波形出力端子

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理により14ビットPWM機能によるデューティパルス出力を行ないます。



$$T_H = t_{H1} + t_{H2} + t_{H3} + \dots + t_{H64}$$

$$t_{f1} = t_{f2} = t_{f3} = \dots t_{f64}$$

1変換周期 = $32768 / \dots = 6.5536\text{ms}$

最小変化幅(精度) = $2 / \dots = 0.4\mu\text{s}$

・1変換周期は上図に示すように64個のパルスで構成され、この1変換周期中の"High"レベル幅合計 (T_H) が、PWDRU、PWDRLのデータに対応しています。この関係は次式で示されます。

$$T_H = (\text{PWDRU、PWDRLのデータ値} + 64) \times t / 2$$

・ここで t は、PWM入力クロックの周期で、PWCR0=1のとき、 $4 / \dots$ となります。したがって、PWDRU、PWDRLのデータ値がH'3000のとき、 T_H 、および t_{fn} は以下ようになります。

$$T_H = t_{H1} + t_{H2} + t_{H3} + \dots + t_{H64} = 4.9408\text{ms}$$

$$t_{f1} = t_{f2} = t_{f3} = \dots t_{f64} = 102.4\mu\text{s}$$

ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a) PWM出力波形データの設定
(b) 14ビットPWM機能の設定
(c) 割込みの許可

図3 14ビットPWM機能によるデューティパルス出力動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、14ビットPWM機能の設定、割込みの許可を行なう

(2) 引数の説明

表3に本タスク例における引数の説明を示します。

表3 引数の説明

引数名	機能	使用モジュール名	データ長	入出力
R1H	PWDRUに設定するPWM出力波形データの上位6ビット	メインルーチン	1バイト	入力
R1L	PWDRLに設定するPWM出力波形データの下位8ビット	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表4に本タスク例における使用内部レジスタ説明を示します。

表4 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
PWCR	PWCRO	PWMコントロールレジスタ(クロックセレクト0) : PWCRO="1"のとき、14ビットPWMに供給されるクロックを /4 に設定	H'FFD0 ビット0	1
	PWDRU	PWMデータレジスタU : PWM出力波形データの上位6ビットを設定	H'FFD1	H'30
	PWDRL	PWMデータレジスタL : PWM出力波形データの下位8ビットを設定	H'FFD2	H'00
PMR1	PWM	ポートモードレジスタ1(P14/PWM端子機能切替え) : PWM="1"のとき、P14/PWM端子をPWM出力端子機能に設定	H'FFFC ビット4	1

(4) 使用RAM説明

本タスク例では、RAMは使用しません。

フローチャート

(a) メインルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Duty Pulse Output by 14-bit PWM Function'
;
;       Function
;       : 14bit PWM
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
;       PWCR          .equ          H'FFD0          ;PWM Control Register
;       PWCRO         .bequ         0,PWCR         ;Clock Select
;       PWDRU         .equ          H'FFD1         ;PWM Data Register U
;       PWDRL         .equ          H'FFD2         ;PWM Data Register L
;       PMR1          .equ          H'FFFC         ;Port Mode Register 1
;       PWM           .bequ         4,PMR1         ;P14/PWM Terminal Function Change
;
;
;
;       Ram Allocation
;
*****
;
;
;       STACK        .equ          H'FF80          ;Stack Pointer
;
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .DATA.W      MAIN          ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w      MAIN          ;IRQ0 Interrupt
;       .data.w      MAIN          ;IRQ1 Interrupt
;       .data.w      MAIN          ;IRQ2 Interrupt
;       .data.w      MAIN          ;IRQ3 Interrupt
;       .data.w      MAIN          ;INT0 - INT7 Interrupt
;
;
;       .org          H'0014
;       .data.w      MAIN          ;Timer A Interrupt
;       .data.w      MAIN          ;Timer B1 Interrupt
;
;
;       .org          H'0020
;       .data.w      MAIN          ;Timer X Interrupt
;       .data.w      MAIN          ;Timer V Interrupt
;
;
;       .org          H'0026

```

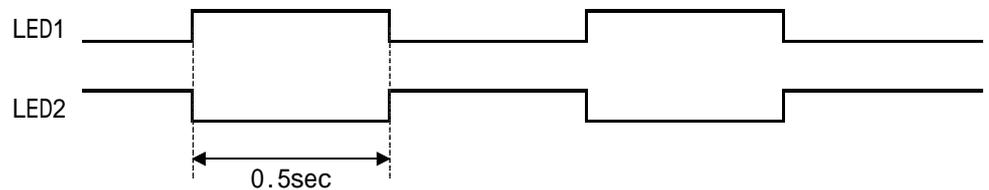
プログラムリスト

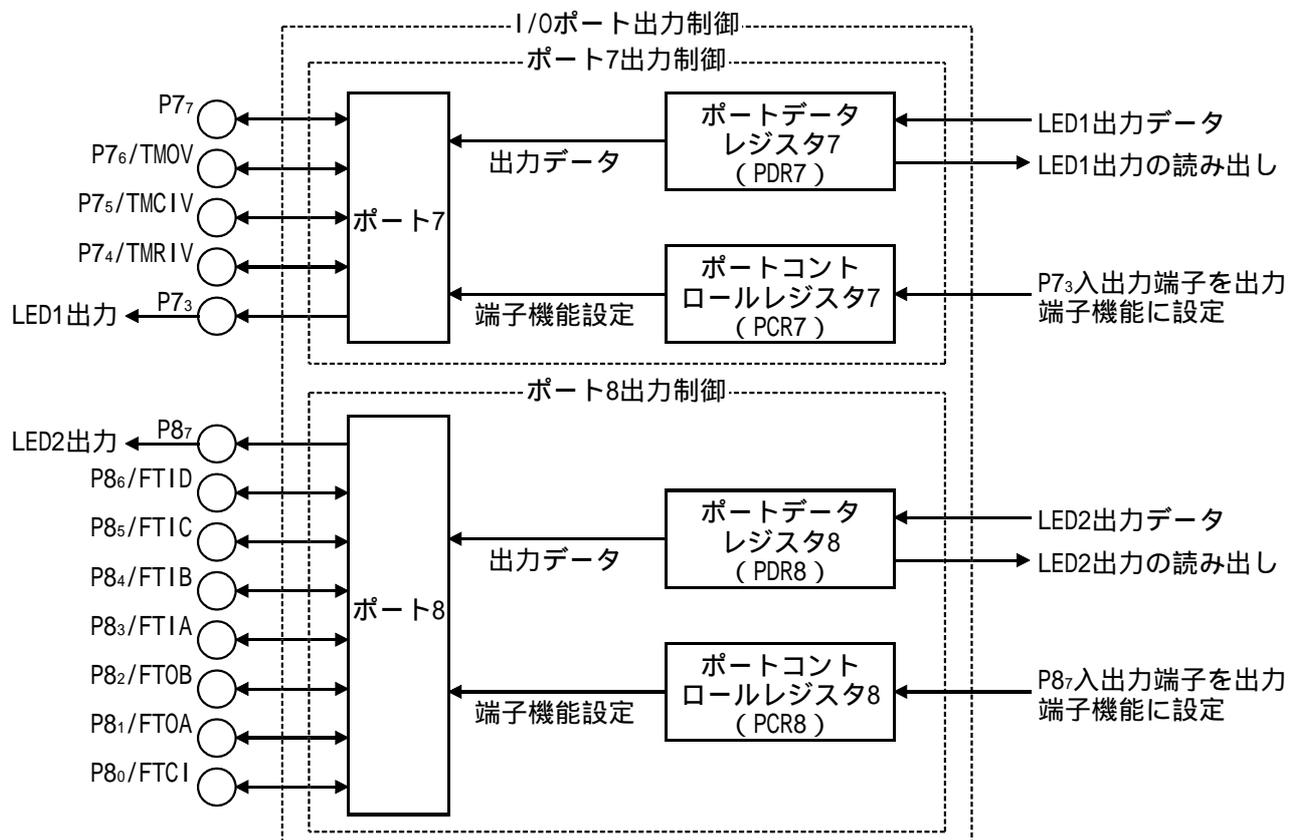
```

;
;      .data.w      MAIN      ;Sci1 Interrupt
;
;      .org        H'002A
;      .data.w      MAIN      ;Sci3 Interrupt
;      .data.w      MAIN      ;A/D Converter Interrupt
;      .data.w      MAIN      ;Sleep Interrupt
;
;*****
;
;      Main Program
;*****
;
;      .org        H'1000
;
MAIN   .equ         $
      MOV.W       #STACK,SP   ;Initialize Stack Pointer
      ORC        #H'80,CCR    ;Interrupt Disable
;
      MOV.W       #H'3000,R1   ;Set 14-Bits PWM Output Pulse Data
;
      MOV.W       #H'14FF,R0
      MOV.B      R0H,@PMR1    ;Initialize PWM Output Terminal Function
      MOV.B      R0L,@PWCR    ;Initialize PWM Input Clock
;
      MOV.B      R1L,@PWDRL   ;Initialize PWM Output Pulse Data Higher
      MOV.B      R1H,@PWDRU   ;Initialize PWM Output Pulse Data Lower
;
      ANDC       #H'7F,CCR    ;Interrupt Enable
;
MAIN9  .equ         $
      BRA        MAIN9
;
      .end

```

2.23 I/Oポートに接続したLEDの点滅動作

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
<p>仕様</p> <p>(1) 図1に示すように、ポートに接続した2つのLEDを交互に点灯/消灯させます。 (2) 点灯、および消灯の時間はタイマA時計用タイムベース機能を使用して0.5secに設定します。 (3) LED1はポート7のP7₃出力端子に、LED2はポート8のP8₇出力端子に接続端子に接続されているものとします。</p>  <p style="text-align: center;">図1 LED点滅動作</p>		

使用機能説明
<p>(1) 本タスク例では、I/Oポートに接続したLEDの点滅動作を行いません。</p> <p>(a) 図2にI/Oポートのブロック図を示します。以下にI/Oポートのブロック図について説明します。</p> <ul style="list-style-type: none"> ・ポートコントロールレジスタ7 (PCR7) は、ポート7の各端子P7₇~P7₃の入出力をビットごとに制御します。PCR7₃に"1"をセットするとP7₃端子は出力端子となり、"0"にクリアすると入力端子となります。 ・ポートデータレジスタ7 (PDR7) は、ポート7の各端子P7₇~P7₃のデータを格納する8ビットのレジスタです。PCRが"1"のとき、ポート7のリードを行なうと、PDR7の値を直接リードします。そのため端子状態の影響を受けません。PCRが"0"のとき、ポート7のリードを行なうと、端子状態が読み出されます。 ・ポートコントロールレジスタ8 (PCR8) は、ポート8の各端子P8₇~P8₀の入出力をビットごとに制御します。PCR8₇に"1"をセットするとP8₇端子は出力端子となり、"0"にクリアすると入力端子となります。 ・ポートデータレジスタ8 (PDR8) は、ポート8の各端子P8₇~P8₀のデータを格納する8ビットのレジスタです。PCRが"1"のとき、ポート8のリードを行なうと、PDR8の値を直接リードします。そのため端子状態の影響を受けません。PCRが"0"のとき、ポート8のリードを行なうと、端子状態が読み出されます。  <p style="text-align: center;">図2 I/Oポート機能のブロック図</p>

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
---------------------	------	--------

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、I/Oポートに接続したLEDの点滅動作を行ないます。

表1 機能割付け

機能	機能割付け
PDR7	ポート7のP7 ₇ ～P7 ₃ のデータを格納
PDR8	ポート8のP8 ₇ ～P8 ₀ のデータを格納
PCR7	ポート7のP7 ₇ ～P7 ₃ 入出力端子機能の設定
PCR8	ポート8のP8 ₇ ～P8 ₀ 入出力端子機能の設定
P7 ₃	LED1の出力端子
P8 ₇	LED2の出力端子
PSW	31.768kHzを4分周したクロックを入力とする5ビットのカウンタ
TCA	PSW出力クロックを入力とする8ビットのカウンタ
TMA	タイマAを時計用タイムベース機能に設定、TCAオーバフロー周期を0.5secに設定

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりI/Oポートに接続したLEDの点滅動作を行ないます。

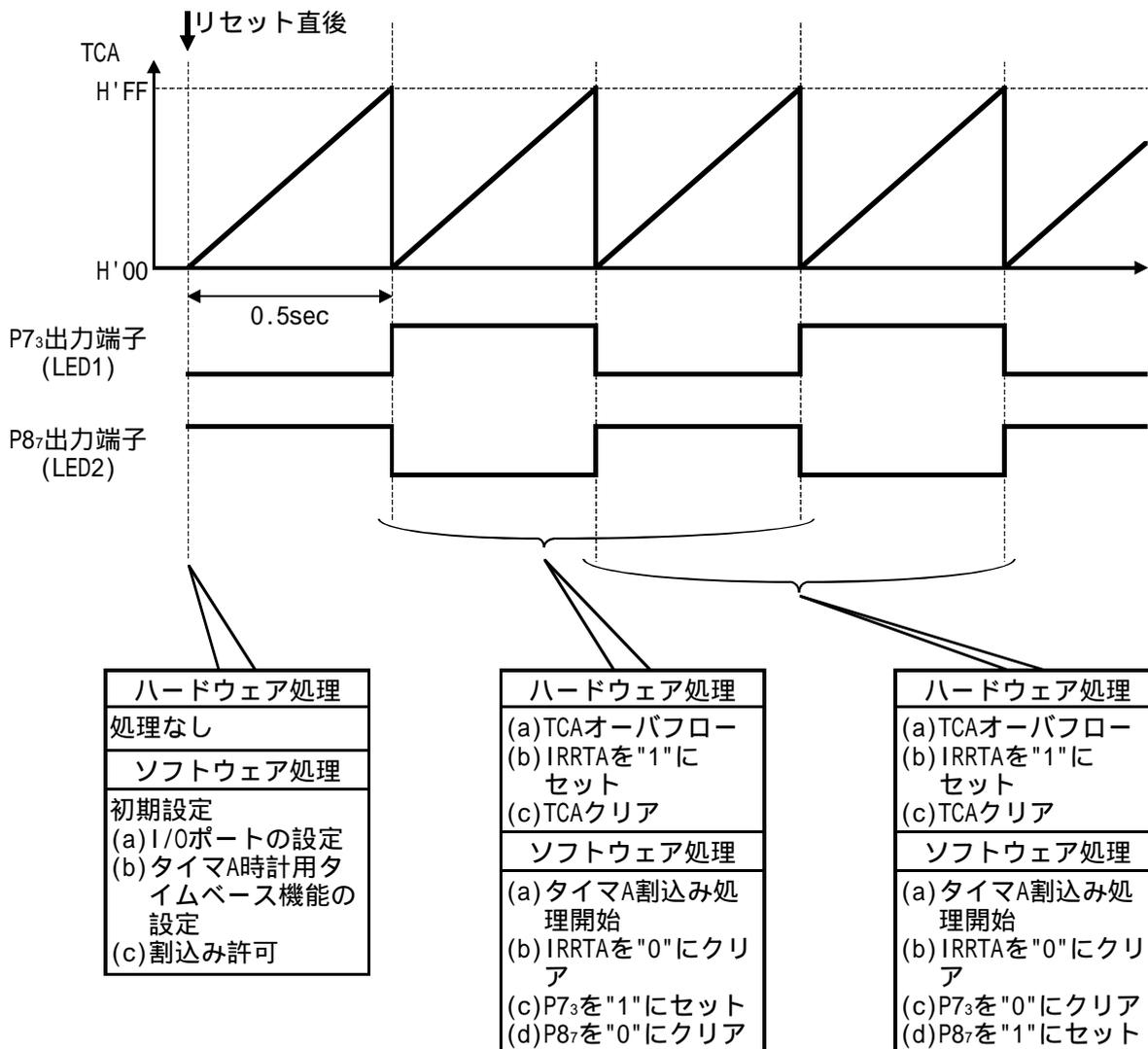


図3 I/Oポートに接続したLEDの点滅動作の動作原理

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
---------------------	------	--------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、タイマA時計用タイムベース機能の設定、I/Oポートの設定、および割込みの許可を行なう
ポート出力	TAINT	タイマA割込み処理ルーチンで、LED1、LED2の出力の判定、および出力の制御を行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMA TMA3 TMA2 TMA1 TMA0	タイマモードレジスタA (内部クロックセレクト3~0) : TMA3="1"、TMA2="0"、TMA1="0"、TMA0="1"のとき、タイマA機能を時計用タイムベース機能に設定、プリスケアラをPSWに設定、およびTCAオーバフロー周期を0.5secに設定	H'FFB0 ビット3 ビット2 ビット1 ビット0	TMA3="1" TMA2="0" TMA1="0" TMA0="1"
TCA	タイマカウンタA : PSWの出力するクロックを入力とし、0.5secでオーバフローする8ビットのアップカウンタ	H'FFB1	H'00
PDR7 P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PDR8 P8 ₇	ポートデータレジスタ8 (ポートデータレジスタ8 ₇) : P8 ₇ =0のとき、P8 ₇ 端子の出力レベルは"Low" : P8 ₇ =1のとき、P8 ₇ 端子の出力レベルは"High"	H'FFDB ビット7	1
PCR7 PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ ="1"のとき、P7 ₃ 入出力端子は、P7 ₃ 出力端子として機能	H'FFEA ビット3	1
PCR8 PCR8 ₇	ポートコントロールレジスタ8 (ポートコントロールレジスタ8 ₇) : PCR8 ₇ ="1"のとき、P8 ₇ 入出力端子は、P8 ₇ 出力端子として機能	H'FFEB ビット7	1
IENR1 IENTA	割込み許可レジスタ1 (タイマA割込みイネーブル) : IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IRR1 IRRTA	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRRTA="0"のとき、タイマA割込みが要求されていない : IRRTA="1"のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0

(4) 使用RAM説明

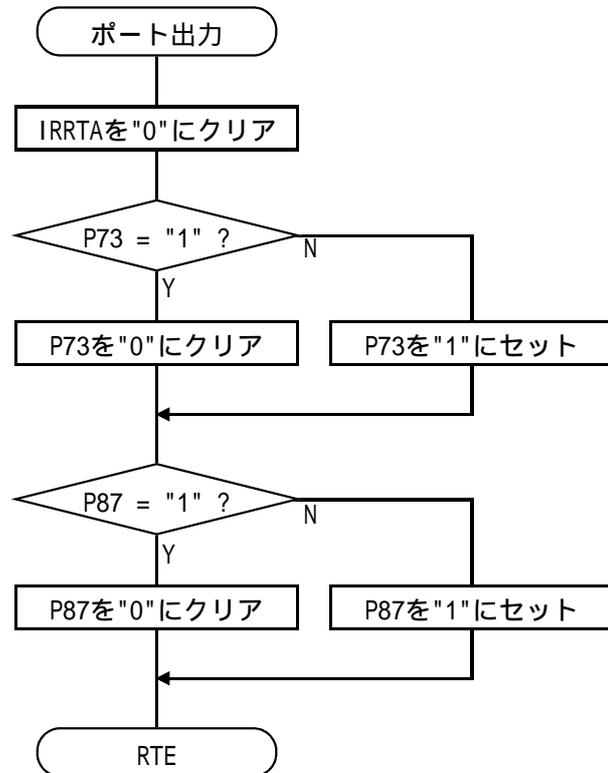
本タスク例では、引数は使用していません。

フローチャート

(a) メインルーチン



(b) タイマA割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Flickering of LEDs Connected to I/O Port'
;
;       Function
;       : I/O Port
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
;
TMA      .equ        H'FFB0      ;Timer Mode Register A
TCA      .equ        H'FFB1      ;Timer Counter A
PDR7     .equ        H'FFDA      ;Port Data Register 7
P73      .bequ       3,PDR7      ;Port Data Register 73
PDR8     .equ        H'FFDB      ;Port Data Register 8
P87      .bequ       7,PDR8      ;Port Data Register 87
PCR7     .equ        H'FFEA      ;Port Control Register 7
PCR73    .bequ       3,PCR7      ;Port Control Register 73
PCR8     .equ        H'FFEB      ;Port Control Register 8
PCR87    .bequ       7,PCR8      ;Port Control Register 87
IENR1    .equ        H'FFF4      ;Interrupt Enable Register 1
IENTA    .bequ       6,IENR1     ;Timer A Interrupt Enable
IRR1     .equ        H'FFF7      ;Interrupt Request Register 1
IRRTA    .bequ       6,IRR1      ;Timer A Interrupt Request Flag
;
;
;
*****
;
;       Ram Allocation
;
*****
;
;
STACK    .equ        H'FF80      ;Stack Pointer
;
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.w       MAIN        ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w       MAIN        ;IRQ0 Interrupt
;       .data.w       MAIN        ;IRQ1 Interrupt
;       .data.w       MAIN        ;IRQ2 Interrupt
;       .data.w       MAIN        ;IRQ3 Interrupt
;       .data.w       MAIN        ;INT0 - INT7 Interrupt
;
;

```

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
プログラムリスト		
;	.org	<pre> H'0014 .data.w TAIN ;Timer A Interrupt .data.w MAIN ;Timer B1 Interrupt ; .org H'0020 .data.w MAIN ;Timer X Interrupt .data.w MAIN ;Timer V Interrupt ; .org H'0026 .data.w MAIN ;Sci1 Interrupt ; .org H'002A .data.w MAIN ;Sci3 Interrupt .data.w MAIN ;A/D Converter Interrupt .data.w MAIN ;Sleep Interrupt ; ***** ; ; Main Program ***** ; ; .org H'1000 ; MAIN .equ \$ MOV.W #STACK,SP ;Initialize Stack Pointer ORC #H'80,CCR ;Interrupt Disable ; MOV.B #H'19,ROL MOV.B ROL,@TMA ;Initialize TCA Overflow Period ; BCLR IRRTA ;Clear IRRTA BSET IENTA ;Timer A Interrupt Enable ; BSET PCR73 ;Initialize P73 Output Terminal Function BSET PCR87 ;Initialize P87 Output Terminal Function ; BCLR P73 ;Initialize P73 Terminal Output BSET P87 ;Initialize P87 Terminal Output ; ANDC #H'7F,CCR ;Interrupt Enable ; MAIN9 .equ \$ BRA MAIN9 ; ***** ; ; Timer A Interrupt ***** ; ; TAIN .equ \$ BCLR IRRTA ;Clear IRRTA ; BTST P73 ;Turn on LED1 ? BEQ TAIN1 ;NO. ; BCLR P73 ;Turn off LED1 BRA TAIN2 ; TAIN1 .equ \$ BSET P73 ;Turn on LED1 ; TAIN2 .equ \$ </pre>

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
プログラムリスト		
<pre> ; ; ; TAINT3 ; TAINT9 ; </pre>	<pre> BTST P87 BEQ TAIN3 ; BCLR P87 BRA TAIN9 ; .equ \$ BSET P87 ; .equ \$ RTE ; .end </pre>	<pre> ;Turn on LED2 ? ;NO ;Turn off LED2 ;Turn on LED2 </pre>

2.24 外部割込みによるカウントスタート

外部割込みによるカウントスタート	使用機能	外部割込み
仕様		
<p>(1) $\overline{\text{IRQ}}_0$端子に接続したスイッチ入力のオンにより$\overline{\text{IRQ}}_0$割込みを発生させ、16ビット汎用レジスタR1に設定した16ビットカウンタのカウントアップを開始します。</p> <p>(2) $\overline{\text{IRQ}}_0$割込みは、$\overline{\text{IRQ}}_0$端子入力の立ち下がりエッジの検出により要求されます。</p> <p>(3) R1に設定した16ビットカウンタがオーバフローするたびにLEDの点灯、または消灯を行いません。</p> <p>(4) 図1に$\overline{\text{IRQ}}_0$入力端子のスイッチ接続例を示します。</p> <div data-bbox="470 448 1093 705" style="text-align: center;"> </div> <p style="text-align: center;">図1 $\overline{\text{IRQ}}_0$入力端子のスイッチ接続例</p>		

使用機能説明
<p>(1) 本タスク例では、外部割込みを使用して$\overline{\text{IRQ}}_0$外部割込みによるカウントスタートを行いません。</p> <p>(a) 図2に割込み受け付けまでのフローを示します。以下に外部割込みについて説明します。</p> <ul style="list-style-type: none"> ・外部割込みには、$\overline{\text{IRQ}}_3 \sim \overline{\text{IRQ}}_0$、$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$割込みの12要因があります。 ・$\overline{\text{IRQ}}_3 \sim \overline{\text{IRQ}}_0$割込みは、$\overline{\text{IRQ}}_3 \sim \overline{\text{IRQ}}_0$端子の入力信号により要求されます。$\overline{\text{IRQ}}_3 \sim \overline{\text{IRQ}}_0$割込みは、立ち上がり/立ち下がりエッジセンスを割込みエッジセレクトレジスタ1 (IEGR1)のIEG3 ~ IEG0により指定できます。 ・ポートモードレジスタ1 (PMR1)により、端子機能が$\overline{\text{IRQ}}_3 \sim \overline{\text{IRQ}}_0$端子に選択された状態で指定されたエッジが入力されると割込み要求レジスタ1 (IRR1)のIRR13 ~ IRR10の対応するビットが"1"にセットされ、割込み要求を発生します。 ・割込み要求の受け付けは、割込み許可レジスタ1 (IENR1)のIEN3 ~ IEN0を"0"にクリアすることにより禁止できます。 ・$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$割込みは、$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$端子の入力信号により要求されます。$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$割込みは、立ち上がり/立ち下がりエッジセンスを割込みエッジセレクトレジスタ2 (IEGR2)のINTEG7 ~ INTEG0により指定できます。 ・$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$端子に指定されたエッジが入力されると、割込み要求レジスタ3 (IRR3)のINTF7 ~ INTF0の対応するビットが"1"にセットされ、割込み要求を発生します。 ・割込み要求の受け付けは、割込み許可レジスタ3 (IENR3)のINTEN7 ~ INTEN0を"0"にクリアすることにより禁止できます。 ・$\overline{\text{INT}}_7 \sim \overline{\text{INT}}_0$端子は、ポート5との兼用端子です。本端子をポートとして使用する場合でも、端子にエッジが入力されればIRR3の対応するビットは"1"にセットされます。 ・コンディションコードレジスタ (CCR)のIビットを"1"にすることによりすべての割り込みを禁止できます。 ・以下に割込みの動作を示します。 <ol style="list-style-type: none"> (1) 割込み許可レジスタの対応するビットが"1"にセットされている状態で、割込み要因が発生したとき、割込みコントローラに対して割込み要求信号が送られます。 (2) 割込みコントローラに割込み要求信号が送られると、割込み要求フラグがセットされます。 (3) 割込み要求フラグが"1"にセットされている割込みの中で、優先順位に従って最高位の割込み要求が選択され、その他は保留となります。 (4) CCRのIビットを参照し、Iビットが"0"にクリアされている場合は、割込み要求は受け付けられませんが、Iビットが"1"にセットされている場合は割込み要求は保留となります。 (5) 割込みが受け付けられると、そのとき実行中の命令の処理が終了した後、プログラムカウンタ (PC)とCCRがスタック領域に退避されます。スタックされるPCは、リターン後に実行する最初のアドレスを示しています。 (6) CCRのIビットが"1"にセットされます。これにより、すべての割込みはマスクされます。 (7) 受け付けた割込みに対応するベクタアドレスを生成し、そのアドレスの内容によって示されるアドレスから、割込み処理ルーチンの実行を開始します。 <ul style="list-style-type: none"> ・割込み許可レジスタをクリアすることにより割込みをディスエーブルにする場合、または割込み要求レジスタをクリアする場合は、必ず割込みをマスクした状態 (I="1")で行ないます。I="0"の状態で行なうと、当該操作命令の実行と当該割込みの発生が競合した場合には、当該操作命令の実行終了時に発生した割込みに対応する例外処理を実行します。

使用機能説明

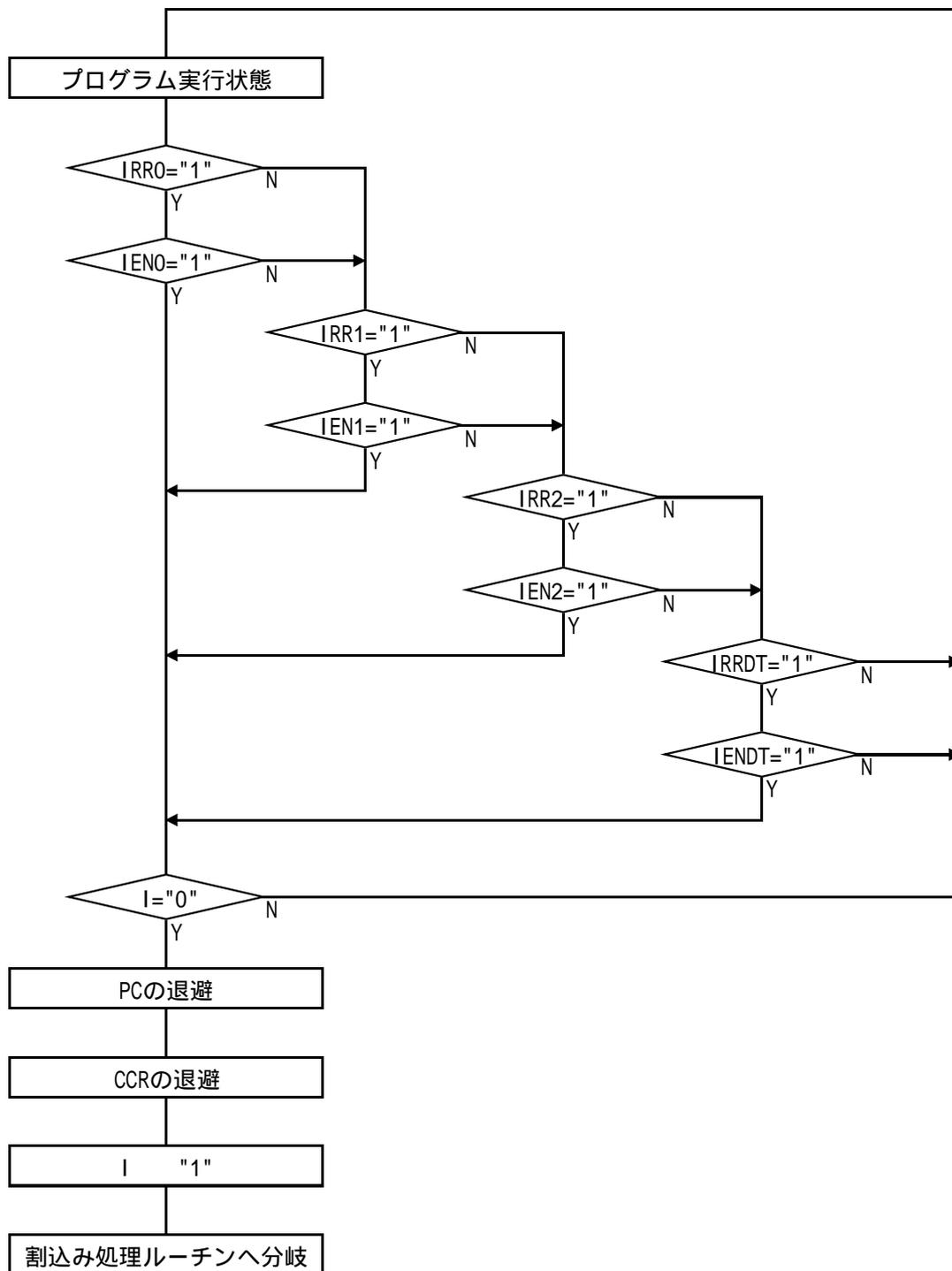


図2 割り込み受け付けまでのフロー

使用機能説明

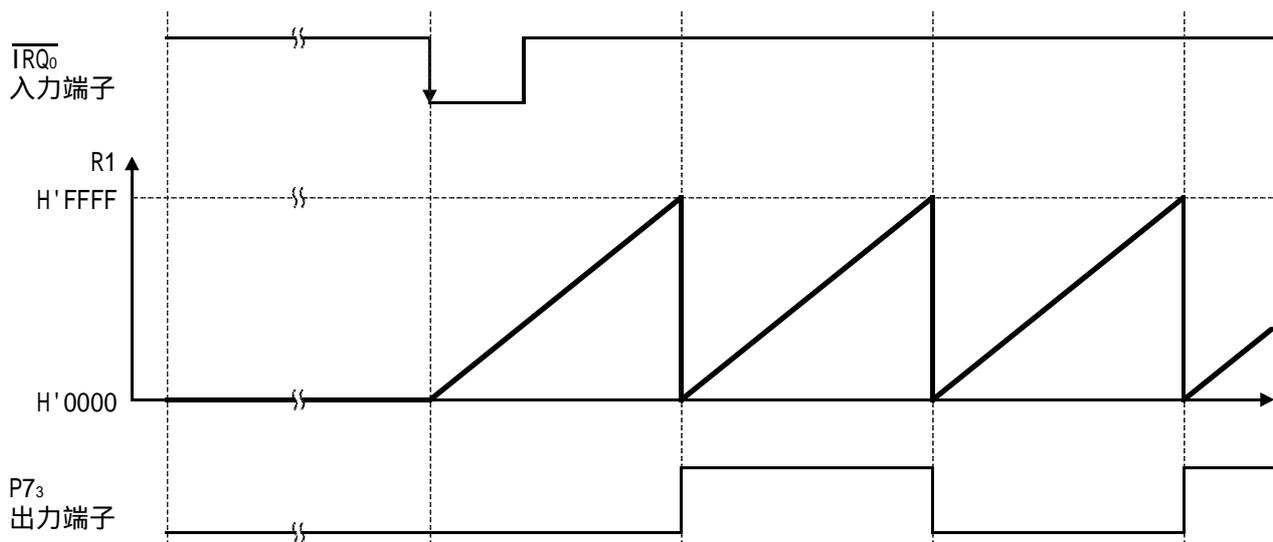
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、I/Oポートに接続したLEDの点滅動作を行ないます。

表1 機能割付け

機能	機能割付け
IRR10	IRQ ₀ 割込みの有無を反映
IEN0	$\overline{\text{TRQ}}_0$ 端子の割込み要求を許可
IEG0	$\overline{\text{TRQ}}_0$ 端子の入力エッジの選択
$\overline{\text{TRQ}}_0$	スイッチ入力の入力端子
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力の出力端子

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理により外部割込みによるカウントスタート動作を行ないます。



ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a) $\overline{\text{TRQ}}_0$ 入力端子の設定
(b) R1に設定する16ビットカウンタの設定
(c) P73出力端子の設定
(d) 割込み許可

ハードウェア処理
(a) IRR10を"1"にセット
ソフトウェア処理
(a) IRQ0割込み処理の開始
(b) IRR10を"0"にクリア
(c) R1に設定した16ビットカウンタのカウントアップ開始
(d) IEN0を"0"にクリア

ハードウェア処理
処理なし
ソフトウェア処理
(a) R1に設定した16ビットカウンタのオーバーフロー
(b) P73を"1"にセット

ハードウェア処理
処理なし
ソフトウェア処理
(a) R1に設定した16ビットカウンタのオーバーフロー
(b) P73を"0"にクリア

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、IRQ ₀ 割込みの設定、LED出力端子の設定、割込みの許可、16ビットカウンタのインクリメント、およびLED出力を行なう
スイッチオン	IRINT	IRQ ₀ 割込み処理ルーチンで、SWONFを"1"にセット

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
PDR7 P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7 PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ ="1"のとき、P7 ₃ 入出力端子は、P7 ₃ 出力端子として機能	H'FFEA ビット3	1
IEGR1 IEG0	割込みエッジセレクトレジスタ1 (IRQ ₀ エッジセレクト) : IEG0="0"のとき、TRQ ₀ 端子入力の立ち上がりエッジを検出	H'FFF2 ビット0	0
IENR1 IEN0	割込み許可レジスタ1 (IRQ ₀ 割込みイネーブル) : IEN0="1"のとき、TRQ ₀ 端子の割込み要求を許可	H'FFF4 ビット0	1
IRR1 IRR10	割込み要求レジスタ1 (IRQ ₀ 割込み要求フラグ) : IRR10="0"のとき、IRQ ₀ 割込みが要求されていない : IRR10="1"のとき、IRQ ₀ 割込みが要求されている	H'FFF7 ビット0	0

(4) 使用RAM説明

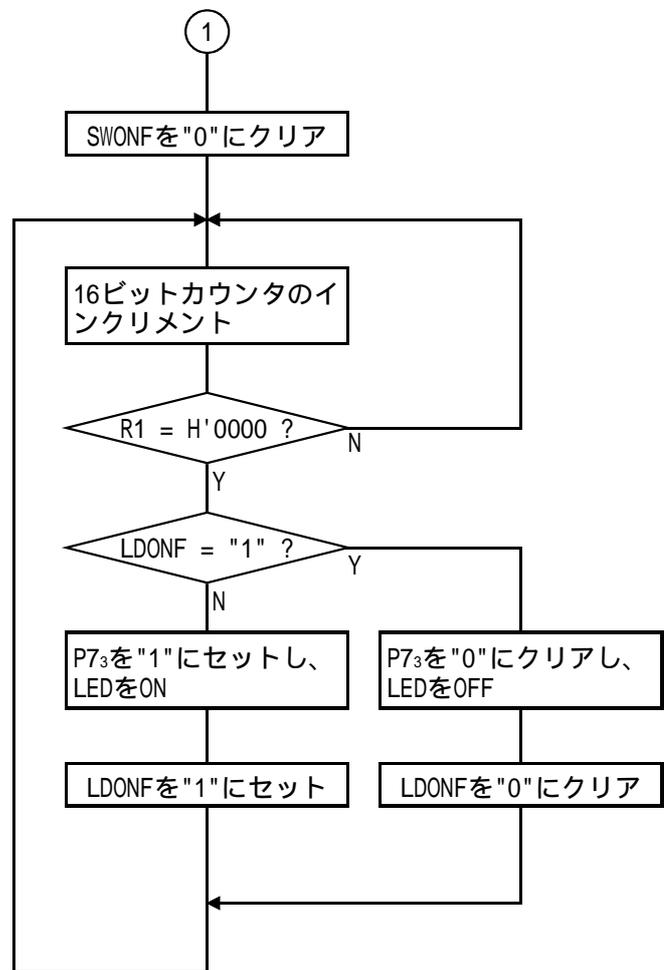
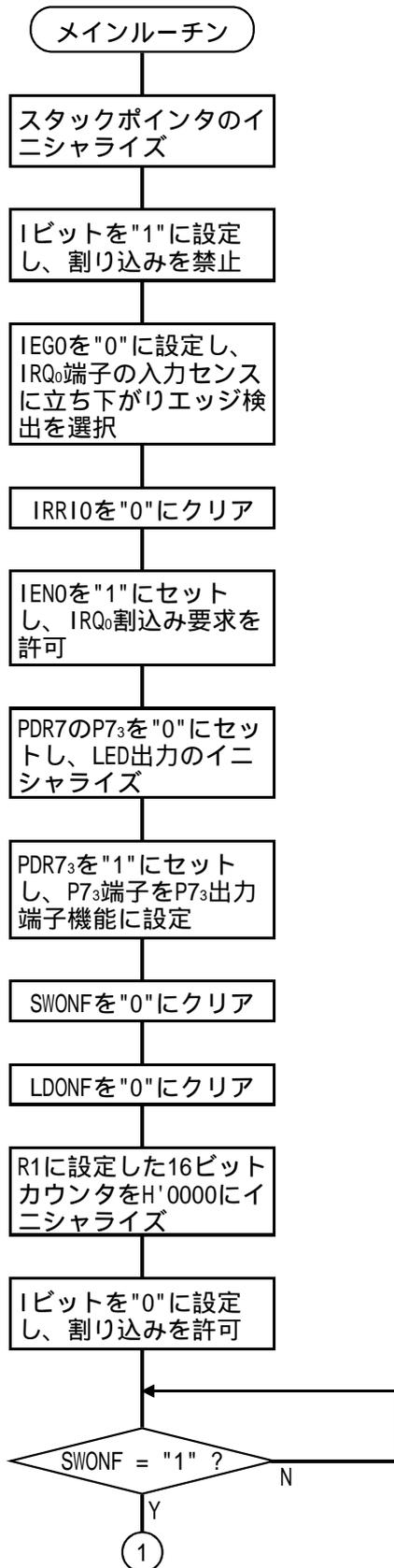
表4に本タスク例で使用するRAMの説明を示します。

表4 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
USRF	SWONF	スイッチ入力のON/OFFを判定するフラグ	H'FB80 ビット0 メインルーチン スイッチオン
	LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット1 メインルーチン

フローチャート

(a) メインルーチン



(b) IRQ0割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Count Start by External Interrupt'
;
;       Function
;       : External Interrupt
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
PDR7      .equ        H'FFDA      ;Port Data Register 7
P73       .bequ       3,PDR7      ;Port Data Register 73
PCR7      .equ        H'FFEA      ;Port Control Register 7
PCR73     .bequ       3,PCR7      ;Port Control Register 73
IEGR1     .equ        H'FFF2      ;Interrupt Edge Select Register 1
IEG0      .bequ       0,IEGR1     ;IRQ0 Edge Select
IENR1     .equ        H'FFF4      ;Interrupt Enable Register 1
IENO      .bequ       0,IENR1     ;IRQ0 Interrupt Enable
IRR1      .equ        H'FFF7      ;Interrupt Request Register 1
IRR10     .bequ       0,IRR1      ;IRQ0 Interrupt Request Flag
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;
STACK     .equ        H'FF80      ;Stack Pointer
USRF      .equ        H'FFB0      ;User Flag Area
SWONF     .bequ       0,USRF      ;Switch On Flag
LDONF     .bequ       1,USRF      ;LED On Flag
;
;
*****
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.w       MAIN        ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w       IRINT       ;IRQ0 Interrupt
;       .data.w       MAIN        ;IRQ1 Interrupt
;       .data.w       MAIN        ;IRQ2 Interrupt
;       .data.w       MAIN        ;IRQ3 Interrupt
;       .data.w       MAIN        ;INT0 - INT7 Interrupt
;
;
;       .org          H'0014
;       .data.w       MAIN        ;Timer A Interrupt

```

外部割込みによるカウントスタート	使用機能	外部割込み
プログラムリスト		
;	.data.w	MAIN ;Timer B1 Interrupt
;	.org	H'0020
;	.data.w	MAIN ;Timer X Interrupt
;	.data.w	MAIN ;Timer V Interrupt
;	.org	H'0026
;	.data.w	MAIN ;Sci1 Interrupt
;	.org	H'002A
;	.data.w	MAIN ;Sci3 Interrupt
;	.data.w	MAIN ;A/D Converter Interrupt
;	.data.w	MAIN ;Sleep Interrupt
;	*****	
;	Main Program	
;	*****	
;	.org	H'1000
MAIN	.equ	\$
;	MOV.W	#STACK,SP ;Initialize Stack Pointer
;	ORC	#H'80,CCR ;Interrupt Disable
;	BCLR	IEG0 ;Initialize IRQ0 Terminal Input Edge
;	BCLR	IRRI0 ;Initialize IRRIO
;	BSET	IENO ;IRQ0 Interrupt Request Enable
;	BCLR	P73 ;Initialzie LED Output
;	BSET	PCR73 ;Initialize P73 Input-Output Terminal Function
;	MOV.B	#H'00,R0L
;	MOV.B	R0L,@USRF ;Initialize SWONF & LDONF
;	SUB.W	R1,R1 ;Initialize 16bit Counter
;	ANDC	#H'7F,CCR ;Interrupt Enable
MAIN1	.equ	\$
;	MOV.B	@USRF,R0L
;	BTST	#0,R0L ;SWONF = 1 ?
;	BEQ	MAIN1 ;No.
;	BCLR	#0,R0L
;	MOV.B	R0L,@USRF ;Clear SWONF
MAIN2	.equ	\$
;	ADDS	#1,R1 ;Increment 16bit Counter
;	MOV.W	R1,R1 ;16bit Counter = H'0000 ?
;	BNE	MAIN2 ;No.
;	MOV.B	@USRF,R0L
;	BTST	#1,R0L ;LDONF = 1 ?
;	BNE	MAIN3 ;Yes.
;	BSET	P73 ;Turn On LED
;	BSET	#1,R0L
;	MOV.B	R0L,@USRF ;Set LDONF
;	BRA	MAIN2
MAIN3	.equ	\$

プログラムリスト

```

        BCLR      P73          ;Turn Off LED
        BCLR      #1,ROL
        MOV.B     ROL,@USRF   ;Clear LDONF
        BRA       MAIN2
;
;
;*****
;
;      IRQ0 Interrupt
;*****
;
;
;IRINT      .equ      $
;          BCLR      IRR10      ;Clear IRR10
;
;
;          MOV.B     @USRF,ROL
;          BSET      #0,ROL
;          MOV.B     ROL,@USRF   ;Sst SWONF
;
;          BSET      IENO      ;IRQ0 Interrupt Disable
;
;          RTE
;
;          .end
```

2.25 内部割込みによる多重割込み動作

内部割込みによる多重割込み動作	使用機能	内部割込み
<p data-bbox="188 181 245 212">仕様</p> <p data-bbox="180 259 1406 407"> (1) タイマA、およびタイマB1割込みを使用して、多重割込み処理を行ないます。 (2) タイマA割込み処理の中で、タイマB割込み要求を受け付けることにより、ソフトウェアでタイマB割込み要求の優先順位をタイマA割込み要求の優先順位より高くなるように設定します。 (3) タイマA割込み要求は、インターバル機能により104.8576msごとに発生するように設定します。 (4) タイマB1割込み要求は、インターバル機能により3.2768msごとに発生するように設定します。 </p>		
<p data-bbox="135 445 296 477">使用機能説明</p> <p data-bbox="180 533 1425 1379"> (1) 本タスク例では、内部割込みを使用してタイマAおよびタイマB1割込みの多重割込み動作を行ないません。 (a) 図1に割込み受け付けまでのフローを示します。以下に内部割込みについて説明します。 <ul style="list-style-type: none"> ・内蔵周辺モジュールからの割込みによる内部割込み要因は、21要因あります。 ・内蔵周辺モジュールからの割込み要求が発生すると、IRR1、IRR2の対応するビットが"1"にセットされます。IENR1、IENR2の各ビットを"0"にクリアすることにより、各割込み要求の受け付けは禁止できます。 ・CCRのIビットを"1"にセットすることにより、すべての割込みをマスクできます。 ・内蔵周辺モジュールからの割込み例外処理が受け付けられるとCCRのIビットは"1"にセットされます。 ・以下に割込みの動作を示します。 <ol style="list-style-type: none"> (1) 割込み許可レジスタの対応するビットが"1"にセットされている状態で、割込み要因が発生したとき、割込みコントローラに対して割込み要求信号が送られます。 (2) 割込みコントローラに割込み要求信号が送られると、割込み要求フラグが"1"にセットされます。 (3) 割込み要求フラグが"1"にセットされている割込みの中で、優先順位に従って最高位の割込み要求が選択され、その他は保留となります。 (4) CCRのIビットを参照し、Iビットが"0"にクリアされている場合は、割込み要求は受け付けられませんが、Iビットが"1"にセットされている場合は割込み要求は保留となります。 (5) 割込みが受け付けられると、そのとき実行中の命令の処理が終了した後、プログラムカウンタ(PC)とCCRがスタック領域に退避されます。スタックされるPCは、リターン後に実行する最初のアドレスを示しています。 (6) CCRのIビットが"1"にセットされます。これにより、すべての割込みはマスクされます。 (7) 受け付けた割込みに対応するベクタアドレスを生成し、そのアドレスの内容によって示されるアドレスから、割込み処理ルーチンの実行を開始します。 ・割込み許可レジスタを"0"にクリアすることにより割込みを禁止にする場合、または割込み要求レジスタをクリアする場合は、必ず割込みをマスクした状態(I="1")で行ないます。I="0"の状態でのような操作を行なうと、当該操作命令の実行と当該割込みの発生が競合した場合に、当該操作命令の実行終了時に発生した割込みに対応する例外処理を実行します。 </p>		

使用機能説明

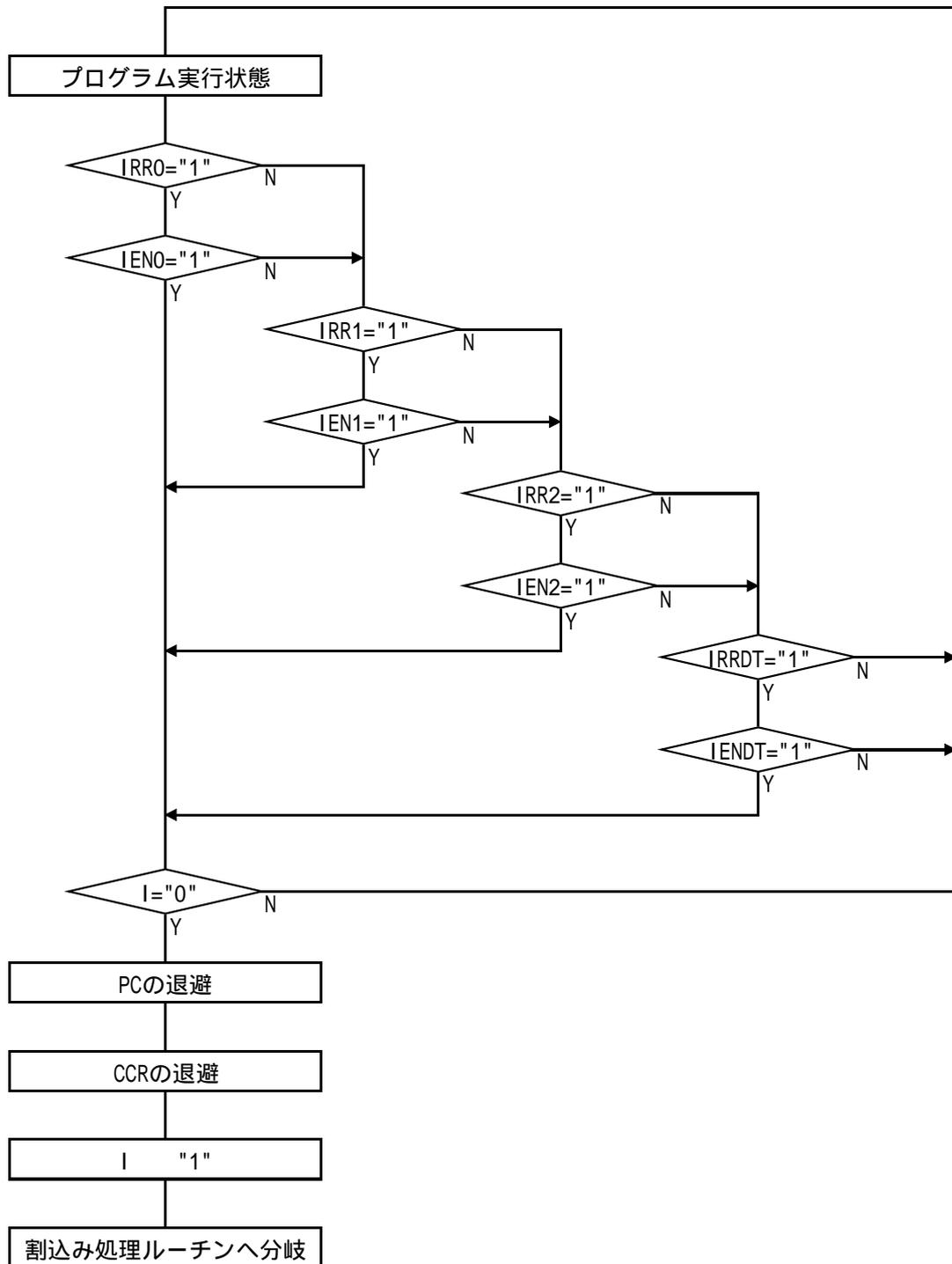


図1 割込み受け付けまでのフロー

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、内部割込みによる多重割込み動作を行ないます。

表1 機能割付け

機能	機能割付け
IENB1	タイマB1割込み要求の許可を行なう
IENTA	タイマA割込み要求の許可を行なう
IRRTB1	タイマB1割込み要求の有無を反映
IRRTA	タイマA割込み要求の有無を反映
CCR 1ビット	すべての割込み要求の許可/禁止を行なう

動作説明

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理により内部割込みによる多重割込み動作を行ないます。

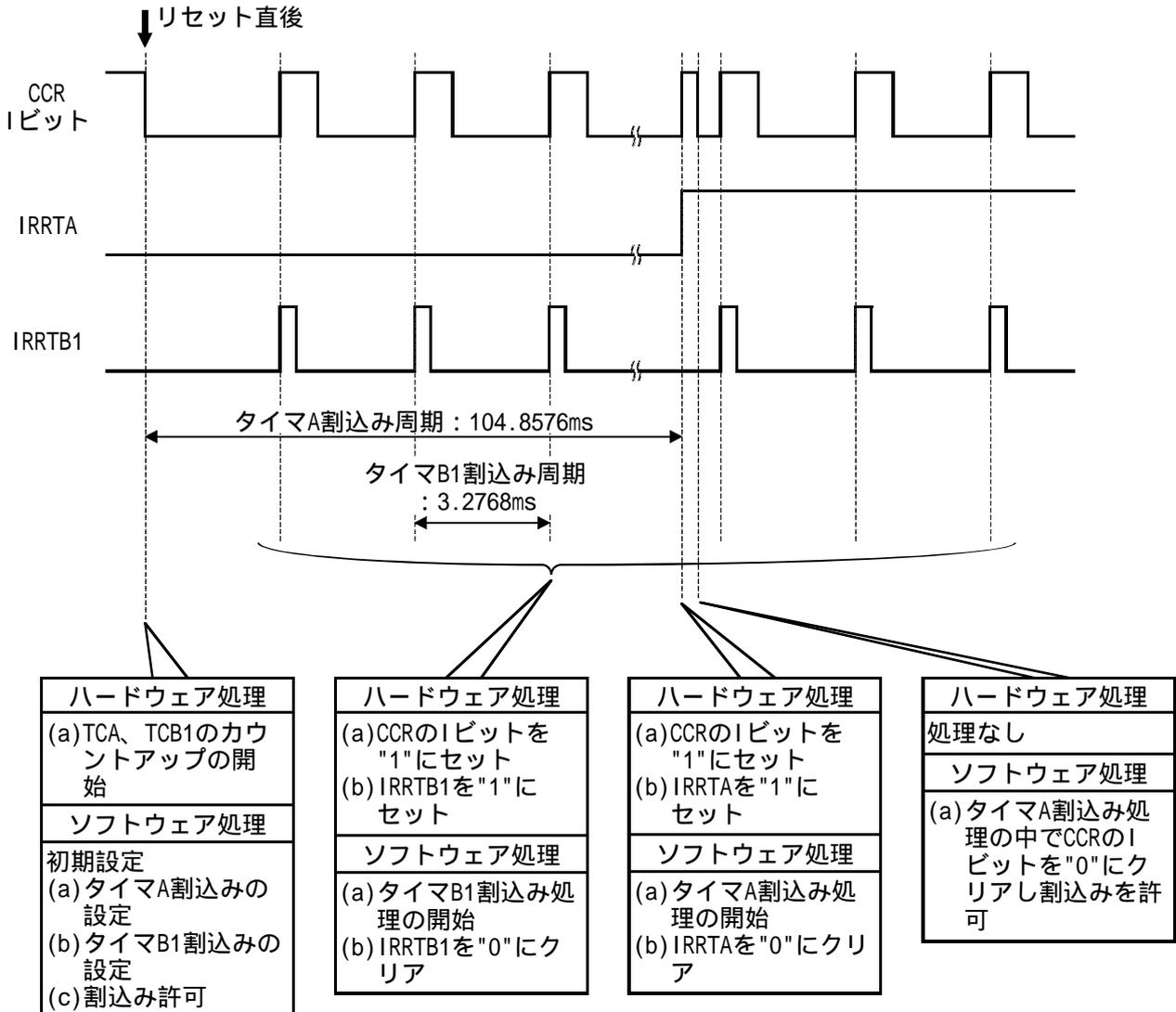


図2 内部割込みによる多重割込み動作の動作原理

内部割込みによる多重割込み動作	使用機能	内部割込み
-----------------	------	-------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、タイマA割込みの設定、タイマB1割込みの設定、ポート7の設定、割込みの許可を行なう
カウント	TAINT	タイマA割込み処理ルーチンで、割込みの許可、16ビットカウンタをインクリメントし、H'4000になったら終了
LED制御	TBINT	タイマB1割込み処理ルーチンで、LEDの点灯/消灯を行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMA	タイマモードレジスタA : TMA=H'12のとき、タイマA機能をインターバル機能に、TCA入力クロックソースをPSSに、プリスケアラ分周比を2048分周に設定	H'FFB0	H'12	
TCA	タイマカウンタA : システムクロックを2048分周したクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00	
TMB1	タイマモードレジスタB1 : TMB1=H'7Cのとき、タイマB1機能をインターバル機能に、TCB1入力クロックソースをPSSに、プリスケアラ分周比を64分周に設定	H'FFB2	H'7C	
TCB1	タイマカウンタB1 : システムクロックを64分周したクロックを入力とする8ビットのアップカウンタ	H'FFB3	H'00	
PDR7	P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7	PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 入出力端子は、P7 ₃ 出力端子として機能	H'FFEA ビット3	1
IENR1	IENB1	割込み許可レジスタ1 (タイマB1割込み要求フラグ) : IENB1="1"のとき、タイマB1割込み要求を許可	H'FFF4 ビット7	1
	IENTA	割込み許可レジスタ1 (タイマA割込み要求フラグ) : IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IRR1	IRRTB1	割込み要求レジスタ1 (タイマB1割込み要求フラグ) : IRRTB1="0"のとき、タイマB1割込みが要求されていない : IRRTB1="1"のとき、タイマB1割込みが要求されている	H'FFF7 ビット7	0
	IRRTA	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRRTA="0"のとき、タイマA割込みが要求されていない : IRRTA="1"のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0

(4) 使用RAM説明

表4に本タスク例で使用するRAMの説明を示します。

表4 使用RAM説明

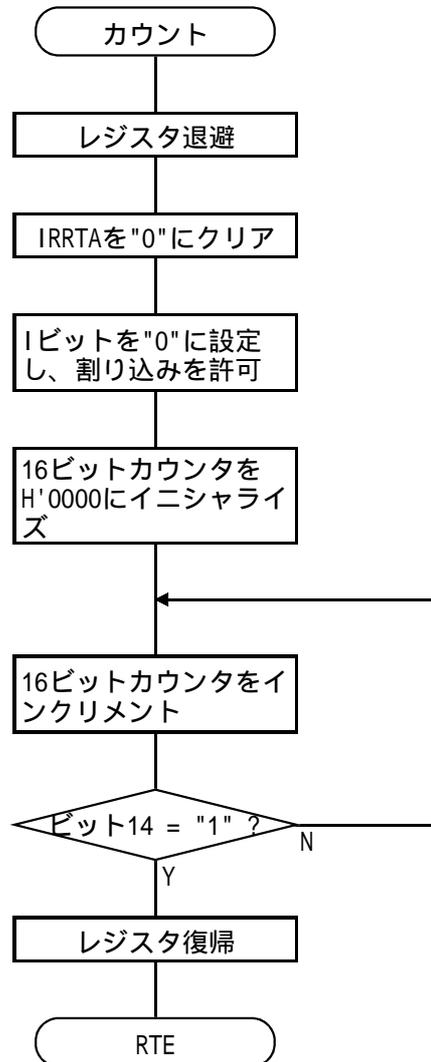
ラベル名	機能	アドレス	使用モジュール名	
USRF	LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット0	LED制御

フローチャート

(a) メインルーチン

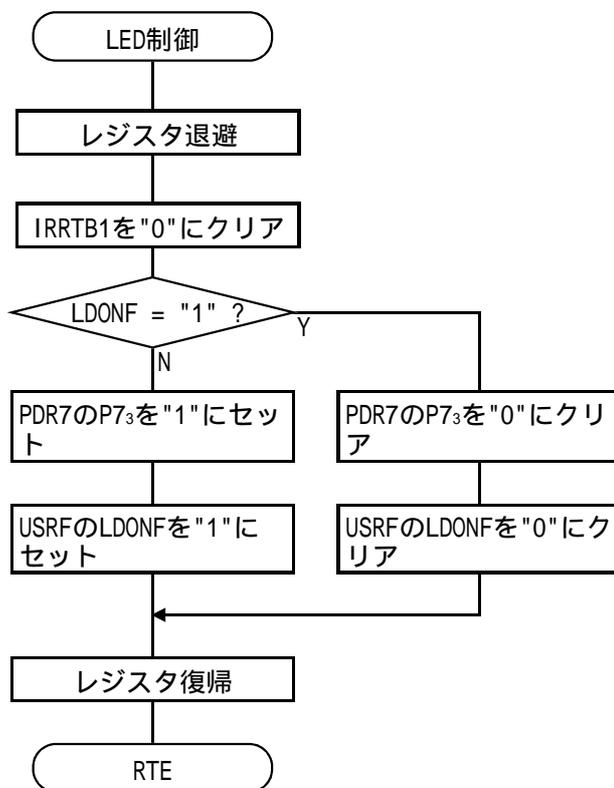


(b) タイマA割込み処理ルーチン



プログラムリスト

(c) タイマB1割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Multiple Interrupt Operation by Internal
;       Interrupt'
;
;       Function
;       : Internal Interrupt
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
TMA      .equ        H'FFB0      ;Timer Mode Register
TCA      .equ        H'FFB1      ;Timer Counter A
TMB1     .equ        H'FFB2      ;Timer Mode Register B1
TCB1     .equ        H'FFB3      ;Timer Counter B1
PDR7     .equ        H'FFDA      ;Port Data Register 7
P73      .bequ       3,PDR7      ;Port Data Register 73
PCR7     .equ        H'FFEA      ;Port Control Register 7
PCR73    .bequ       3,PCR7      ;Port Control Register 73
IENR1    .equ        H'FFF4      ;Interrupt Enable Register 1
IENTB1   .bequ       7,IENR1     ;Timer B1 Interrupt Enable
IENTA    .bequ       6,IENR1     ;Timer A Interrupt Enable
IRR1     .equ        H'FFF7      ;Interrupt Request Register 1
IRRTB1   .bequ       7,IRR1      ;Timer B1 Interrupt Request Flag
IRRTA    .bequ       6,IRR1      ;Timer A Interrupt Request Flag
;
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;
STACK    .equ        H'FF80      ;Stack Pointer
USRF     .equ        H'FB80      ;User Flag Area
LDONF    .bequ       0,USRF      ;LED on Flag
;
;
;
*****
;
;
;       Vector Address
;
*****
;
;
;       .org          H'0000
;       .data.w       MAIN        ;Reset Interrupt
;
;
;       .org          H'0008
;       .data.w       MAIN        ;IRQ0 Interrupt
;       .data.w       MAIN        ;IRQ1 Interrupt
;       .data.w       MAIN        ;IRQ2 Interrupt
;       .data.w       MAIN        ;IRQ3 Interrupt

```


プログラムリスト

```

;
;      ANDC      #H'7F,CCR      ;Interrupt Enable
;
;      SUB.W     R1,R1          ;Initialize 16bit Counter
;
TAINT1 .equ      $
;      ADDS     #1,R1          ;Increment 16bit Counter
;      BTST    #6,R1H         ;R1 Bit 14 = 1 ?
;      BEQ     TAINT1         ;No.
;
;      POP     R0              ;Restore R0
;
;      RTE
;
;*****
;
;      Timer B1 Interrupt
;*****
;
TBINT .equ      $
;      PUSH    R0              ;Store R0
;
;      BCLR    IRRTB1         ;Clear IRRTB1
;
;      MOV.B   @USRF,R0L      ;
;      BTST    #0,R0L         ;LDONF = 1 ?
;      BNE     TBINT1         ;Yes.
;
;      BSET    P73            ;Turn on LED
;      BSET    #0,R0L         ;Set LDONF
;      BRA     TBINT9
;
TBINT1 .equ      $
;      BCLR    P73            ;Turn off LED
;      BCLR    #0,R0L         ;Clear LDONF
;
TBINT9 .equ      $
;      MOV.B   R0L,@USRF
;
;      POP     R0              ;Restore R0
;
;      RTE
;
;      .end

```

2.26 スリープ（中速）モードへの遷移

スリープ（中速）モードへの遷移	使用機能	低消費電力モード：スリープ（中速）モード
-----------------	------	----------------------

仕様

- (1) スリープ（中速）モードへの遷移を行ないます。
- (2) アクティブモードでプログラム実行中に、INT₀端子に接続したスイッチ入力によりINT₀割込みを発生させ、INT₀割込み処理終了後にSLEEP命令の実行によりスリープ（中速）モードへ遷移させます。
- (3) スリープ（中速）モードに遷移してから、1sec後にタイマA割込みによりスリープ（中速）モードを解除し、アクティブ（中速）モードへ遷移させます。
- (4) プログラム起動と同時にLEDを点灯し、スリープ（中速）モードへ遷移するとLEDを消灯します。そして、アクティブ（中速）モードへ遷移すると再びLEDを点灯します。
- (5) アクティブ（中速）モードから直接遷移によりアクティブ（高速）モードへ遷移します。
- (6) 図1にINT₀端子に接続するスイッチの接続例を示します。

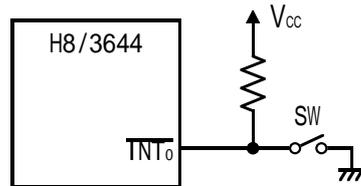


図1 スリープ（中速）モードへの遷移におけるスイッチ接続例

使用機能説明

- (1) 本タスク例では、低消費電力モードのスリープ（中速）モードへの遷移を行ないます。図2にスリープ（中速）モードへのモード遷移図を示します。以下にスリープ（中速）モードの機能の説明を示します。
 - ・アクティブモードで、SYSCR1のSSBYが"0"、LSONが"0"、SYSCR2のMSONが"1"、DTONが"0"のときSLEEP命令を実行すると、スリープ（中速）モードに遷移します。
 - ・スリープ（中速）モードでは、CPUの動作は停止し、PWMを除く内蔵周辺モジュールは動作します。
 - ・スリープ（中速）モードでは、SYSCR1のMA1、MA0で設定した周波数のクロックによって動作します。
 - ・スリープ（中速）モードの解除は、すべての割込み（タイマA、タイマB1、タイマX、タイマV、IRQ₃～IRQ₀、INT₇～INT₀、SCI3、SCI1、A/D変換器）、RES端子入力によって行われます。
 - ・割込みによる解除において、割込み要求が発生すると、スリープ（中速）モードは解除され、割込み例外処理を開始します。スリープ（中速）モードからはアクティブ（中速）モードに遷移します。
 - ・CCRのIビットが"1"のとき、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合はスリープ（中速）モードは解除されません。
 - ・RES端子による解除において、RES端子を"Low"レベルにするとリセット状態に遷移し、スリープ（中速）モードは解除されます。
 - ・本タスク例では、スリープ（中速）モードの解除にタイマA割込みを使用します。
 - ・スリープ（中速）モードでは、SYSCR1のMA1、MA0で設定した周波数のクロックによって動作します。動作クロックは、 $\frac{osc}{128}$ 、 $\frac{osc}{64}$ 、 $\frac{osc}{32}$ 、 $\frac{osc}{16}$ より選択します。
 - ・ osc はOSCクロックで、システムクロック発振器の出力クロックです。
 - ・本タスク例では、スリープ（中速）モードの動作クロックを $\frac{osc}{128}$ に設定します。

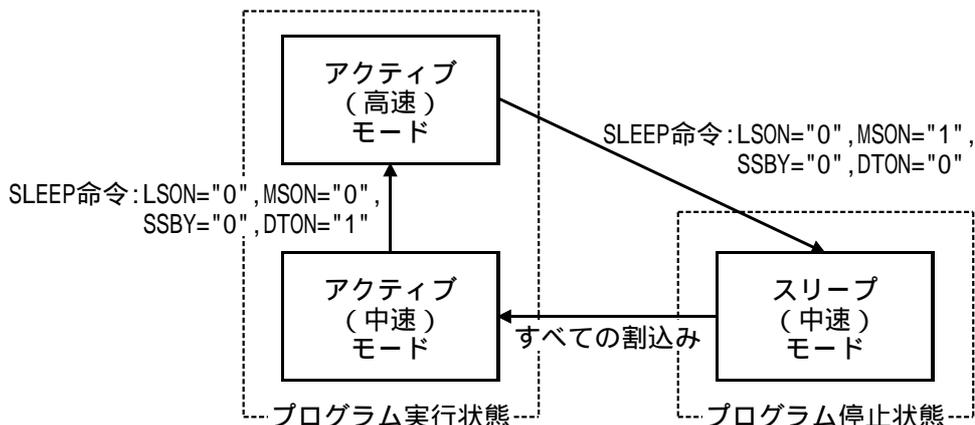


図2 スリープ（中速）モードへの遷移におけるモード遷移図

スリープ（中速）モードへの遷移	使用機能	低消費電力モード：スリープ（中速）モード
-----------------	------	----------------------

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、スリープ（中速）モードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
$\overline{\text{INT}}_0$	スイッチ入力

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりスリープ（中速）モードへの遷移を行ないます。

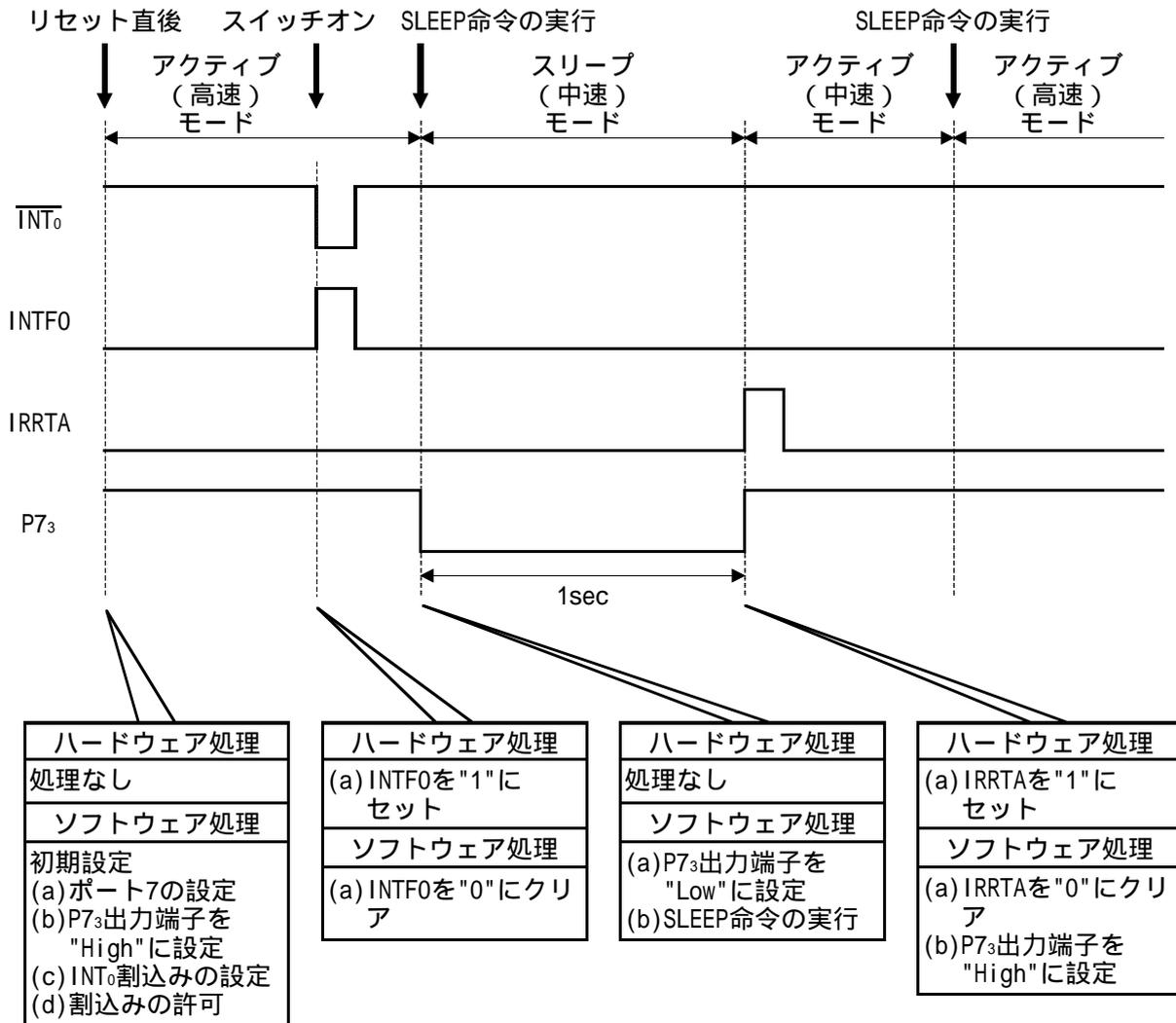


図3 スリープ（中速）モードへの遷移の動作原理

スリープ（中速）モードへの遷移	使用機能	低消費電力モード：スリープ（中速）モード
-----------------	------	----------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、INT ₀ 割込みの設定、ポート7の設定、割込みの許可、LEDの制御、タイマA割込みの設定、スリープ（中速）モード、アクティブ（高速）モードへの遷移を行なう
スイッチオン	ITINT	INT ₀ 割込み処理ルーチンで、SWONFのセット、INT ₀ 割込みの禁止を行なう
スリープ（中速）モード解除	TAINT	タイマA割込み処理ルーチンで、タイマA割込みの禁止を行なう
直接遷移	DTINT	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMA	タイマモードレジスタA : TMA=H'18のとき、タイマA機能を時計用タームベース機能に、TCA入力クロックソースをPSWに、TCAオーバフロー周期を1secに設定	H'FFB0	H'18	
TCA	タイマカウンタA : 32.768kHzを128分周したクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00	
PDR7	P7 ₃	ポートデータレジスタ7（ポートデータレジスタ7 ₃ ） : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7	PCR7 ₃	ポートコントロールレジスタ7（ポートコントロールレジスタ7 ₃ ） : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1
SYSCR1	SSBY	システムコントロールレジスタ1（ソフトウェアスタンバイ） : SSBY="0"のとき、アクティブモードでSLEEP命令実行後、スリープモードに遷移。サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移	H'FFF0 ビット7	0
	LSON	システムコントロールレジスタ1（ロースピードオンフラグ） : LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定	H'FFF0 ビット3	0
	MA1 MA0	システムコントロールレジスタ1（モードクロックセレクト1,0） : MA1="1"、MA0="1"のとき、アクティブ（中速）モードまたはスリープ（中速）モードの動作クロックを $\frac{osc}{128}$ に設定	H'FFF0 ビット1 ビット0	MA1="1" MA0="1"

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
SYSCR2	DTON	システムコントロールレジスタ2 (ダイレクトトランスファオンフラグ) : DTON="0"のとき、アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移 : DTON="1"のとき、アクティブ(中速)モードでSLEEP命令を実行したとき、アクティブ(高速)モード(SSBY="0", MSON="0", LSON="0"のとき)、またはサブアクティブモード(SSBY="1", MSON="1", LSON="1"のとき)に直接遷移	H'FFF1 ビット3	0
	MSON	システムコントロールレジスタ2 (ミドルスピードオンフラグ) : MSON="0"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ(高速)モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ(高速)モードで動作 : MSON="1"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ(中速)モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ(中速)モードで動作	H'FFF1 ビット2	1
IEGR2	INTEG0	割り込み許可エッジセレクトレジスタ2 (INT ₀ エッジセレクト) : INTEG0="1"のとき、INT ₀ 端子入力の立上りエッジを検出	H'FFF3 ビット0	1
IENR1	IENTA	割り込み許可レジスタ1 (タイマA割り込みイネーブル) : IENTA="0"のとき、タイマA割り込み要求を禁止 : IENTA="1"のとき、タイマA割り込み要求を許可	H'FFF4 ビット6	0
IENR2	IENDT	割り込み許可レジスタ2 (直接遷移割り込みイネーブル) : IENDT="0"のとき、直接遷移による割り込み要求を禁止 : IENDT="1"のとき、直接遷移による割り込み要求を許可	H'FFF5 ビット7	1
IENR3	INTENO	割り込み許可レジスタ3 (INT ₀ 割り込みイネーブル) : INTENO="0"のとき、INT ₀ 割り込み要求を禁止 : INTENO="1"のとき、INT ₀ 割り込み要求を許可	H'FFF6 ビット0	1
IRR1	IRRTA	割り込み要求レジスタ1 (タイマA割り込み要求フラグ) : IRRTA="0"のとき、タイマA割り込みが要求されていない : IRRTA="1"のとき、タイマA割り込みが要求されている	H'FFF7 ビット6	0
IRR2	IRRDT	割り込み要求レジスタ2 (直接遷移割り込み要求フラグ) : IRRDT="0"のとき、直接遷移による割り込みが要求されていない : IRRDT="1"のとき、直接遷移による割り込みが要求されている	H'FFF8 ビット7	0
IRR3	INTFO	割り込み要求レジスタ3 (INT ₀ 割り込み要求フラグ) : INTFO="0"のとき、INT ₀ 割り込みが要求されていない : INTFO="1"のとき、INT ₀ 割り込みが要求されている	H'FFF9 ビット0	0

(4) 使用RAM説明

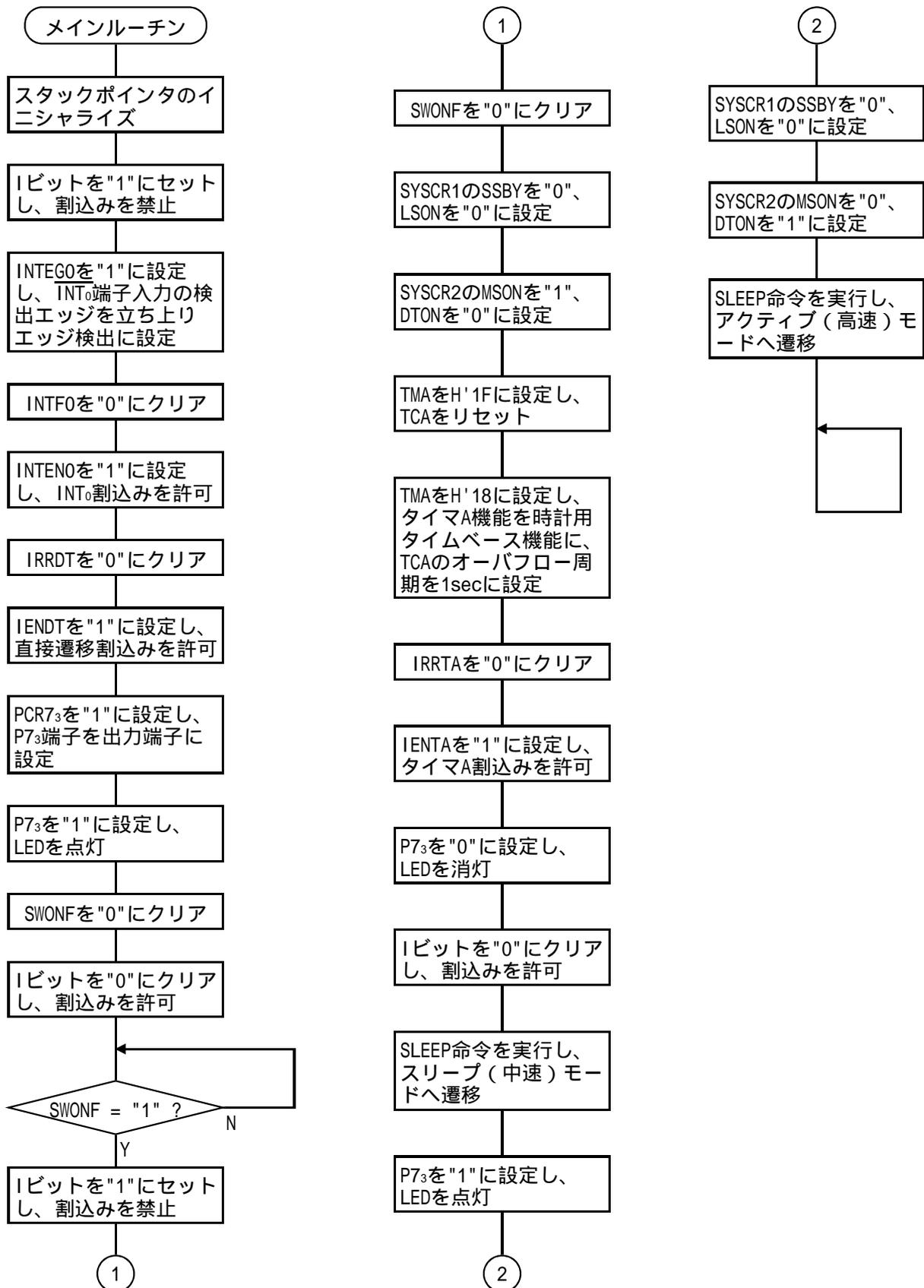
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

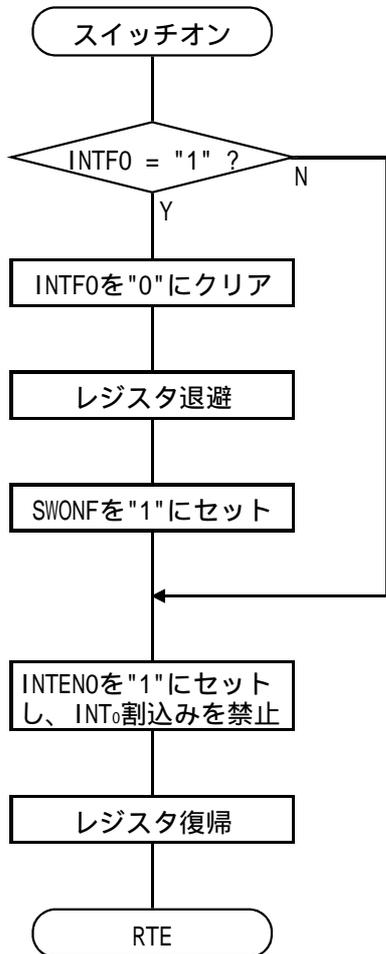
ラベル名	機能	アドレス	使用モジュール名	
USRF	SWONF	スイッチ入力のON/OFFを判定するフラグ	H'FB80 ビット0	メインルーチン スイッチオン

フローチャート

(a) メインルーチン



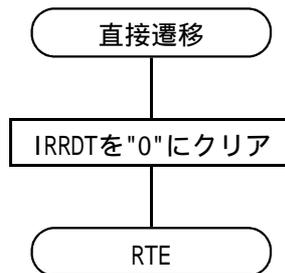
フローチャート

(b) INT₀割込み処理ルーチン

(c) タイマA割込み処理ルーチン



(d) 直接遷移割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Transition to Sleep (Middle-Speed) Mode'
;
;       Function
;       : Power-Down Mode
;       : Sleep (Middle-Speed) Mode
;
;       External Clock : 10MHz
;       Internal Clock : 5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
*****
;
;       .cpu          300L
;
*****
;
;       Symbol Definition
;
*****
;
TMA      .equ        H'FFB0      ;Timer Mode Register A
TCA      .equ        H'FFB1      ;Timer Counter A
PDR7     .equ        H'FFDA      ;Port Data Register 7
P73      .bequ       3,PDR7      ;Port Data Register 73
PCR7     .equ        H'FFEA      ;Port Control Register 7
PCR73    .bequ       3,PCR7      ;Port Control Register 73
SYSCR1   .equ        H'FFF0      ;System Control Register 1
SSBY     .bequ       7,SYSCR1    ;Software Standby
STS2     .bequ       6,SYSCR1    ;Standby Timer Select 2
STS1     .bequ       5,SYSCR1    ;Standby Timer Select 1
STS0     .bequ       4,SYSCR1    ;Standby Timer Select 0
LSON     .bequ       3,SYSCR1    ;Low Speed On Flag
MA1      .bequ       1,SYSCR1    ;Active (Middle-Speed) Mode Clock Select 1
MA0      .bequ       0,SYSCR1    ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2   .equ        H'FFF1      ;System Control Register 2
NESEL    .bequ       4,SYSCR2    ;Noise Elimination Sampling Frequency Select
DTON     .bequ       3,SYSCR2    ;Direct Transfer On Flag
MSON     .bequ       2,SYSCR2    ;Middle Speed On Flag
SA1      .bequ       1,SYSCR2    ;Subactive Mode Clock Select 1
SA0      .bequ       0,SYSCR2    ;Subactive Mode Clock Select 0
IEGR2    .equ        H'FFF3      ;Interrupt Edge Select Register 2
INTEGO   .bequ       0,IEGR2     ;INT0 Edge Select
IENR1    .equ        H'FFF4      ;Interrupt Enable Register 1
IENTA    .bequ       6,IENR1     ;Timer A Interrupt Enable
IENR2    .equ        H'FFF5      ;Interrupt Enable Register 2
IENDT    .bequ       7,IENR2     ;Direct Transfer Interrupt Enable
IENR3    .equ        H'FFF6      ;Interrupt Enable Register 3
INTENO   .bequ       0,IENR3     ;INT0 Interrupt Enable
IRR1     .equ        H'FFF7      ;Interrupt Request Register 1
IRRTA    .bequ       6,IRR1      ;Timer A Interrupt Request Flag
IRR2     .equ        H'FFF8      ;Interrupt Request Register 2
IRRDT    .bequ       7,IRR2      ;Direct Transfer Interrupt Request Flag
IRR3     .equ        H'FFF9      ;Interrupt Request Register 3
INTFO    .bequ       0,IRR3      ;INT0 Interrupt Request Flag
;

```

プログラムリスト

```

*****
;
;
;          Ram Allocation
;
*****
;
;
STACK      .equ      H'FF80      ;Stack Pointer
USRF       .equ      H'FB80      ;User Flag Area
SWONF     .bequ     0,USRF      ;Switch On Flag
;
;
*****
;
;
;          Vector Address
;
*****
;
;
;          .org      H'0000
;          .data.w   MAIN          ;Reset Interrupt
;
;
;          .org      H'0008
;          .data.w   MAIN          ;IRQ0 Interrupt
;          .data.w   MAIN          ;IRQ1 Interrupt
;          .data.w   MAIN          ;IRQ2 Interrupt
;          .data.w   MAIN          ;IRQ3 Interrupt
;          .data.w   ITINT        ;INT0 - INT7 Interrupt
;
;
;          .org      H'0014
;          .data.w   TAINT        ;Timer A Interrupt
;          .data.w   MAIN          ;Timer B1 Interrupt
;
;
;          .org      H'0020
;          .data.w   MAIN          ;Timer X Interrupt
;          .data.w   MAIN          ;Timer V Interrupt
;
;
;          .org      H'0026
;          .data.w   MAIN          ;Sci1 Interrupt
;
;
;          .org      H'002A
;          .data.w   MAIN          ;Sci3 Interrupt
;          .data.w   MAIN          ;A/D Converter Interrupt
;          .data.w   DTINT        ;Sleep Interrupt
;
;
*****
;
;
;          Main Program
;
*****
;
;
;          .org      H'1000
;
MAIN      .equ      $
MOV.W    #STACK,SP      ;Initialize Stack Pointer
ORC      #H'80,CCR      ;Interrupt Disable
;
;
;          BSET      INTEGO      ;Set Rising Edge Of INT0 Terminal Input
;          BCLR      INTF0      ;Clear INTF0
;          BSET      INTENO     ;INT0 Interrupt Enable
;
;
;          BCLR      IRRDT      ;Clear IRRDT
;          BSET      IENDT      ;Direct Transfer Interrupt Enable
;
;
;          BSET      PCR73      ;Initialize P73 Output Terminal Function
;          BSET      P73        ;Turn On LED
;
;
;          MOV.B     #H'00,ROL   ;Initialize SWONF
;          MOV.B     ROL,@USRF

```

プログラムリスト

```

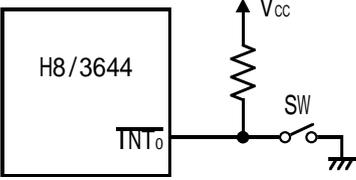
;
;
;
MAIN1    .equ          $
MOV.B   @USRF,ROL
BTST    #0,ROL      ;SWONF = "1" ?
BEQ     MAIN1      ;No.
;
;
;
ORC     #H'80,CCR   ;Interrupt Disable
;
;
;
BCLR    #0,ROL
MOV.B   ROL,@USRF  ;Clear SWONF
;
;
;
MOV.W   #H'07E4,R0
MOV.B   R0H,@SYSCR1 ;Initialize Function of Sleep Mode 1
MOV.B   R0L,@SYSCR2 ;Initialize Function of Sleep Mode 2
;
;
;
MOV.W   #H'1F18,R0
MOV.B   R0H,@TMA    ;Initialize TCA
MOV.B   R0L,@TMA    ;Initialize TCA Overflow Period
BCLR    IRRTA       ;Clear IRRTA
BSET    IENTA       ;Timer A Interrupt Enable
;
;
;
BCLR    P73         ;Turn Off LED
;
;
;
ANDC    #H'7F,CCR   ;Interrupt Enable
;
;
;
SLEEP                    ;Transition to Sleep (Middle-Speed) Mode
;
;
;
BSET    P73         ;Turn On LED
;
;
;
MOV.W   #H'07E8,R0
MOV.B   R0H,@SYSCR1 ;Initialize Function of Active Mode 1
MOV.B   R0L,@SYSCR2 ;Initialize Function of Active Mode 2
;
;
;
SLEEP                    ;Transition to Active (High-Speed) Mode
;
MAIN9    .equ          $
BRA     MAIN9
;
;
;
*****
;
;
;
INTO Interrupt
;
;
;
*****
;
;
;
ITINT    .equ          $
BTST    INTF0        ;INTF0 = "1" ?
BEQ     ITINT9      ;No.
;
;
;
BCLR    INTF0       ;Clear INTF0
;
;
;
PUSH    R0          ;Store R0
;
;
;
MOV.B   @USRF,ROL
BSET    #0,ROL
MOV.B   ROL,@USRF  ;Set SWONF
;
;
;
ITINT9   .equ          $
BCLR    INTENO      ;INT0 Interrupt Disable
;
;
;
POP     R0          ;Restore R0

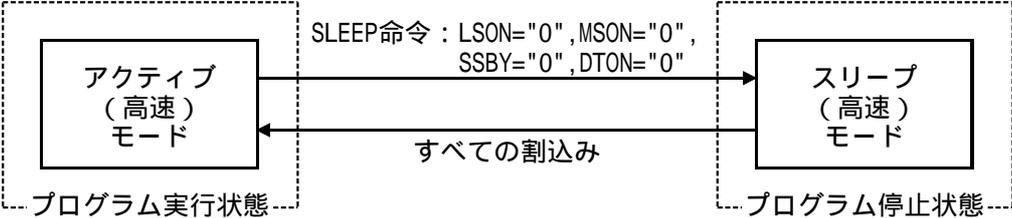
```

プログラムリスト

```
;  
; RTE  
;  
;*****  
; Timer A Interrupt  
;*****  
;  
; TAINT .equ $  
; BCLR IRRTA ;Clear IRRTA  
;  
; BCLR IENTA ;Timer A Interrupt Disable  
;  
; RTE  
;  
;*****  
; Direct Transfer Interrupt  
;*****  
;  
; DTINT .equ $  
; BCLR IRRDT ;Clear IRRDT  
;  
; RTE  
;  
; .end
```

2.27 スリープ（高速）モードへの遷移

スリープ（高速）モードへの遷移	使用機能	低消費電力モード：スリープ（高速）モード
仕様		
<p>(1) スリープ（高速）モードへの遷移を行いません。</p> <p>(2) アクティブモードでプログラム実行中に、INT₀端子に接続したスイッチ入力によりINT₀割込みを発生させ、INT₀割込み処理終了後にSLEEP命令の実行によりスリープ（高速）モードへ遷移させます。</p> <p>(3) スリープ（高速）モードに遷移してから、1sec後にタイマA割込みによりスリープ（高速）モードを解除し、アクティブ（高速）モードへ遷移させます。</p> <p>(4) プログラム起動と同時にLEDを点灯し、スリープ（高速）モードへ遷移するとLEDを消灯します。そして、アクティブ（高速）モードへ遷移すると再びLEDを点灯します。</p> <p>(5) 図1にINT₀端子に接続するスイッチの接続例を示します。</p>		
		
<p>図1 スリープ（高速）モードへの遷移におけるスイッチ接続例</p>		

使用機能説明
<p>(1) 本タスク例では、低消費電力モードのスリープ（高速）モードへの遷移を行いません。図2にスリープ（高速）モードへのモード遷移図を示します。以下にスリープ（高速）モードの機能の説明を示します。</p> <ul style="list-style-type: none"> ・アクティブモードで、SYSCR1のSSBYが"0"、LSONが"0"、SYSCR2のMSONが"0"、DTONが"0"のときSLEEP命令を実行すると、スリープ（高速）モードに遷移します。 ・スリープ（高速）モードでは、CPUの動作は停止し、PWMを除く内蔵周辺モジュールは動作します。なお、CPUのレジスタの内容は保持されます。 ・スリープ（高速）モードの解除は、すべての割込み（タイマA、タイマB1、タイマX、タイマV、IRQ₃～IRQ₀、INT₇～INT₀、SCI3、SCI1、A/D変換器）、RES端子入力によって行われます。 ・割込みによる解除において、割込み要求が発生すると、スリープ（高速）モードは解除され、割込み例外処理を開始します。スリープ（高速）モードからはアクティブ（高速）モードに遷移します。 ・CCRのIビットが"1"のとき、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合はスリープ（高速）モードは解除されません。 ・RES端子による解除において、RES端子を"Low"レベルにするとリセット状態に遷移し、スリープ（高速）モードは解除されます。 ・本タスク例では、スリープ（高速）モードの解除にタイマA割込みを使用します。

<p>図2 スリープ（高速）モードへの遷移におけるモード遷移図</p>

使用機能説明

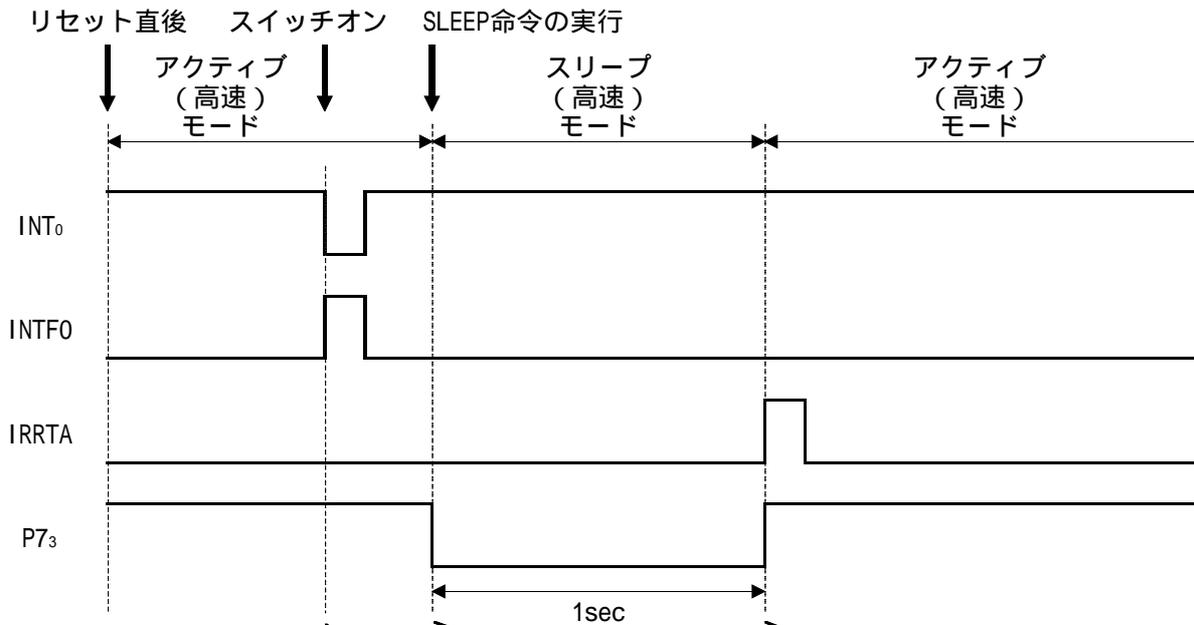
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、スリープ（高速）モードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
$\overline{\text{INT}}_0$	スイッチ入力

動作原理

(1) 図3に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりスリープ（高速）モードへの遷移を行ないます。



ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a) ポート7の設定
(b) P7 ₃ 出力端子を "High" に設定
(c) INT ₀ 割込みの設定
(d) 割込みの許可

ハードウェア処理
(a) INTF0を "1" にセット
ソフトウェア処理
(a) INTF0を "0" にクリア

ハードウェア処理
処理なし
ソフトウェア処理
(a) P7 ₃ 出力端子を "Low" に設定
(b) SLEEP命令の実行

ハードウェア処理
(a) IRRTAを "1" にセット
ソフトウェア処理
(a) IRRTAを "0" にクリア
(b) P7 ₃ 出力端子を "High" に設定

図3 スリープ（高速）モードへの遷移の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、INT ₀ 割込みの設定、ポート7の設定、割込みの許可、LEDの制御、タイマA割込みの設定、スリープ（高速）モードへの遷移を行なう
スイッチオン	ITINT	INT ₀ 割込み処理ルーチンで、SWONFのセット、INT ₀ 割り込みの禁止を行なう
スリープ（高速）モード解除	TAINT	タイマA割込み処理ルーチンで、タイマA割り込みの禁止を行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMA	タイマモードレジスタA ：TMA=H'18のとき、タイマA機能を時計用タームベース機能に、TCA入力クロックソースをPSWに、TCAオーバフロー周期を1secに設定	H'FFB0	H'18
TCA	タイマカウンタA ：32.768kHzを128分周したクロックを入力とする8ビットのアップカウンタ	H'FFB1	H'00
PDR7	P7 ₃	ポートデータレジスタ7（ポートデータレジスタ7 ₃ ） ：P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" ：P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3 0
PCR7	PCR7 ₃	ポートコントロールレジスタ7（ポートコントロールレジスタ7 ₃ ） ：PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3 1
SYSCR1	SSBY	システムコントロールレジスタ1（ソフトウェアスタンバイ） ：SSBY="0"のとき、アクティブモードでSLEEP命令実行後、スリープモードに遷移。サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移	H'FFF0 ビット7 0
	LSON	システムコントロールレジスタ1（ロースピードオンフラグ） ：LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定	H'FFF0 ビット3 0
SYSCR2	DTON	システムコントロールレジスタ2（ダイレクトトランスファオンフラグ） ：DTON="0"のとき、アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移	H'FFF1 ビット3 0
	MSON	システムコントロールレジスタ2（ミドルスピードオンフラグ） ：MSON="0"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ（高速）モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ（高速）モードで動作	H'FFF1 ビット2 0

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
I EGR2	INTEG0	割込み許可エッジセレクトレジスタ2 (INT ₀ エッジセレクト) : INTEG0="1"のとき、INT ₀ 端子入力の立上りエッジを検出	H'FFF3 ビット0	1
I ENR1	I ENTA	割込み許可レジスタ1 (タイマA割込みイネーブル) : I ENTA="0"のとき、タイマA割込み要求を禁止 : I ENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	0
I ENR3	I NTENO	割込み許可レジスタ3 (INT ₀ 割込みイネーブル) : I NTENO="0"のとき、INT ₀ 割込み要求を禁止 : I NTENO="1"のとき、INT ₀ 割込み要求を許可	H'FFF6 ビット0	1
I RR1	I RR TA	割込み要求レジスタ1 (タイマA割込み要求フラグ) : I RR TA="0"のとき、タイマA割込みが要求されていない : I RR TA="1"のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0
I RR3	I NTF0	割込み要求レジスタ3 (INT ₀ 割込み要求フラグ) : I NTF0="0"のとき、INT ₀ 割込みが要求されていない : I NTF0="1"のとき、INT ₀ 割込みが要求されている	H'FFF9 ビット0	0

(4) 使用RAM説明

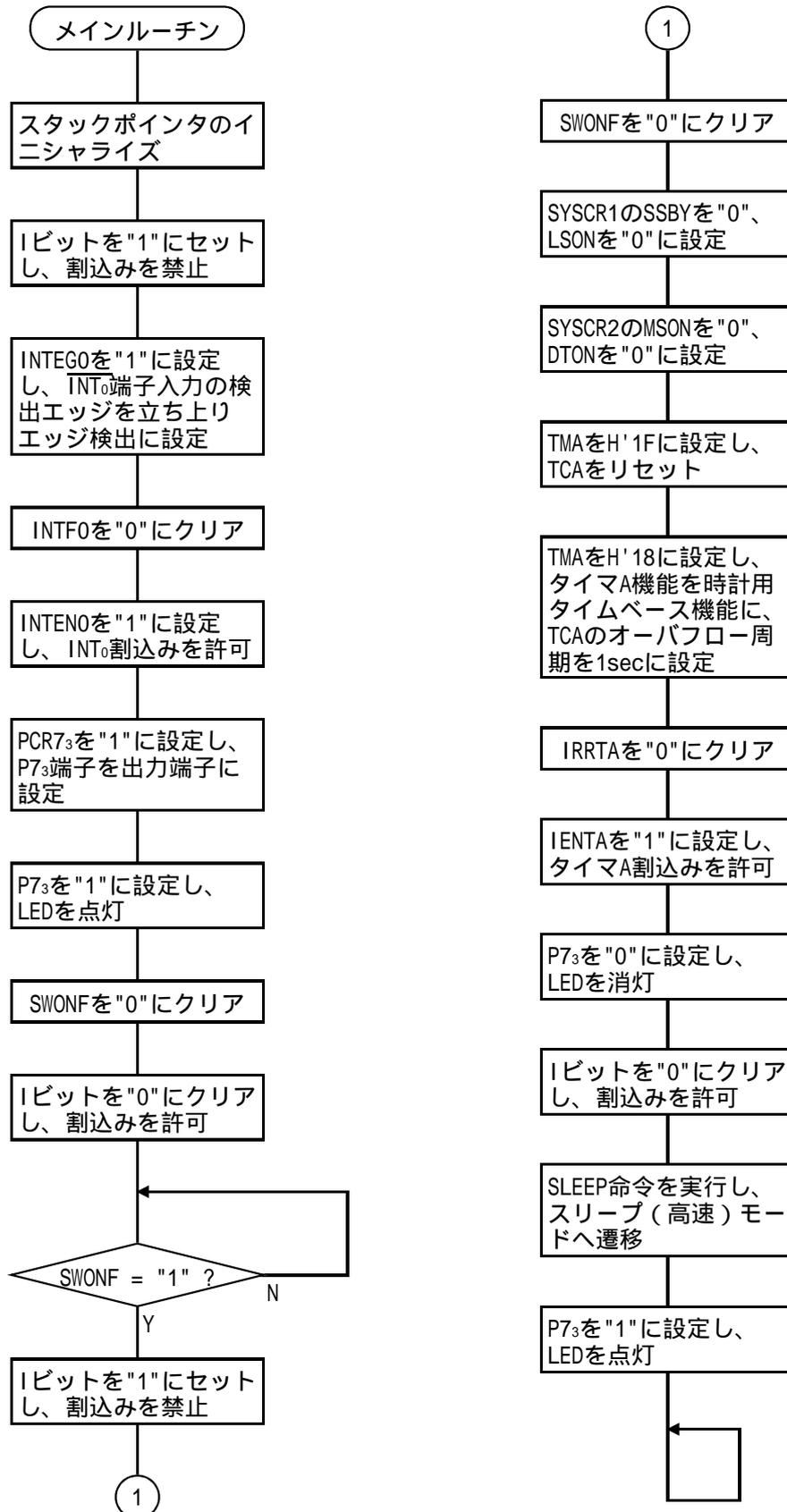
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

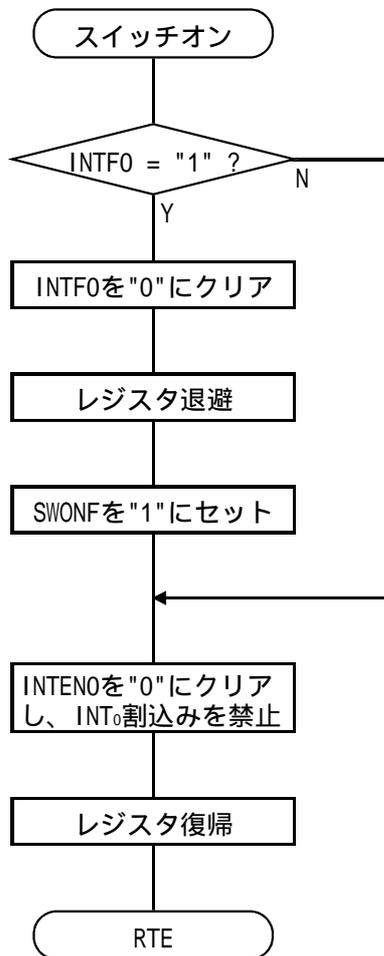
ラベル名		機能	アドレス	使用モジュール名
USRF	SWONF	スイッチ入力のON/OFFを判定するフラグ	H'FB80 ビット0	メインルーチン スイッチオン

フローチャート

(a) メインルーチン



フローチャート

(b) INT₀割込み処理ルーチン

(c) タイマA割込み処理ルーチン



プログラムリスト

```

*****
;
;
;       H8/300L Series -H8/3644,H8/3657-
;       Application Note
;
;       'Transition to Sleep (High-Speed) Mode'
;
;       Function
;       : Power-Down Mode
;       :   Sleep (High-Speed) Mode
;
;       External Clock : 10MHz
;       Internal Clock :  5MHz
;       Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;
;       .cpu          300L
;
*****
;
;
;       Symbol Definition
;
*****
;
;
TMA      .equ        H'FFB0      ;Timer Mode Register A
TCA      .equ        H'FFB1      ;Timer Counter A
PDR7     .equ        H'FFDA      ;Port Data Register 7
P73      .bequ       3,PDR7      ;Port Data Register 73
PCR7     .equ        H'FFEA      ;Port Control Register 7
PCR73    .bequ       3,PCR7      ;Port Control Register 73
SYSCR1   .equ        H'FFF0      ;System Control Register 1
SSBY     .bequ       7,SYSCR1    ;Software Standby
STS2     .bequ       6,SYSCR1    ;Standby Timer Select 2
STS1     .bequ       5,SYSCR1    ;Standby Timer Select 1
STS0     .bequ       4,SYSCR1    ;Standby Timer Select 0
LSON     .bequ       3,SYSCR1    ;Low Speed On Flag
MA1      .bequ       1,SYSCR1    ;Active (Middle-Speed) Mode Clock Select 1
MA0      .bequ       0,SYSCR1    ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2   .equ        H'FFF1      ;System Control Register 2
NESEL    .bequ       4,SYSCR2    ;Noise Elimination Sampling Frequency Select
DTON     .bequ       3,SYSCR2    ;Direct Transfer On Flag
MSON     .bequ       2,SYSCR2    ;Middle Speed On Flag
SA1      .bequ       1,SYSCR2    ;Subactive Mode Clock Select 1
SA0      .bequ       0,SYSCR2    ;Subactive Mode Clock Select 0
IEGR2    .equ        H'FFF3      ;Interrupt Edge Select Register 2
INTEGO   .bequ       0,IEGR2     ;INT0 Edge Select
IENR1    .equ        H'FFF4      ;Interrupt Enable Register 1
IENTA    .bequ       6,IENR1     ;Timer A Interrupt Enable
IENR3    .equ        H'FFF6      ;Interrupt Enable Register 3
INTENO   .bequ       0,IENR3     ;INT0 Interrupt Enable
IRR1     .equ        H'FFF7      ;Interrupt Request Register 1
IRR3     .bequ       6,IRR1      ;Timer A Interrupt Request Flag
IRR3     .equ        H'FFF9      ;Interrupt Request Register 3
INTFO    .bequ       0,IRR3      ;INT0 Interrupt Request Flag
;
;
*****
;
;
;       Ram Allocation
;
*****
;
;

```

プログラムリスト

```

STACK      .equ      H'FF80      ;Stack Pointer
USRF       .equ      H'FB80      ;User Flag Area
SWONF     .bequ     0,USRF      ;Switch On Flag
;
;
;*****
;
;          Vector Address
;*****
;
;          .org      H'0000
;          .data.w   MAIN        ;Reset Interrupt
;
;          .org      H'0008
;          .data.w   MAIN        ;IRQ0 Interrupt
;          .data.w   MAIN        ;IRQ1 Interrupt
;          .data.w   MAIN        ;IRQ2 Interrupt
;          .data.w   MAIN        ;IRQ3 Interrupt
;          .data.w   ITINT       ;INT0 - INT7 Interrupt
;
;          .org      H'0014
;          .data.w   TAINT       ;Timer A Interrupt
;          .data.w   MAIN        ;Timer B1 Interrupt
;
;          .org      H'0020
;          .data.w   MAIN        ;Timer X Interrupt
;          .data.w   MAIN        ;Timer V Interrupt
;
;          .org      H'0026
;          .data.w   MAIN        ;Sci1 Interrupt
;
;          .org      H'002A
;          .data.w   MAIN        ;Sci3 Interrupt
;          .data.w   MAIN        ;A/D Converter Interrupt
;          .data.w   MAIN        ;Sleep Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org      H'1000
;
;MAIN      .equ      $
;          MOV.W    #STACK,SP    ;Initialize Stack Pointer
;          ORC     #H'80,CCR     ;Interrupt Disable
;
;          BSET    INTEGO       ;Set Rising Edge Of INT0 Terminal Input
;          BCLR    INTFO        ;Clear INTFO
;          BSET    INTENO       ;INT0 Interrupt Enable
;
;          BSET    PCR73        ;Initialize P73 Output Terminal Function
;          BSET    P73          ;Turn On LED
;
;          MOV.B   #H'00,ROL     ;
;          MOV.B   ROL,@USRF    ;Initialize SWONF
;
;          ANDC    #H'7F,CCR     ;Interrupt Enable
;
;MAIN1     .equ      $
;          MOV.B   @USRF,ROL    ;
;          BTST   #0,ROL        ;SWONF = "1" ?
;          BEQ    MAIN1         ;No.

```

プログラムリスト

```

;
;       ORC          #H'80,CCR      ;Interrupt Disable
;
;       BCLR         #0,ROL
;       MOV.B        ROL,@USRF      ;Clear SWONF
;
;       MOV.W        #H'07E0,R0
;       MOV.B        ROH,@SYSCR1    ;Initialize Function of Sleep Mode 1
;       MOV.B        ROL,@SYSCR2    ;Initialize Function of Sleep Mode 2
;
;       MOV.W        #H'1F18,R0
;       MOV.B        ROH,@TMA        ;Initialize TCA
;       MOV.B        ROL,@TMA        ;Initialize TCA Overflow Period
;       BCLR         IRRTA          ;Clear IRRTA
;       BSET         IENTA          ;Timer A Interrupt Enable
;
;       BCLR         P73            ;Turn Off LED
;
;       ANDC         #H'7F,CCR      ;Interrupt Enable
;
;       SLEEP                            ;Transition to Sleep (High-Speed) Mode
;
;       BSET         P73            ;Turn On LED
;
MAIN9   .equ          $
        BRA          MAIN9
;
;*****
;
;       INTO Interrupt
;*****
;
;       ITINT        .equ          $
;       BTST         INTF0          ;INTF0 = "1" ?
;       BEQ          ITINT9         ;No.
;
;       BCLR         INTF0          ;Clear INTF0
;
;       PUSH         R0             ;Store R0
;
;       MOV.B        @USRF,R0L
;       BSET         #0,R0L
;       MOV.B        ROL,@USRF      ;Set SWONF
;
;       ITINT9       .equ          $
;       BCLR         INTENO         ;INT0 Interrupt Disable
;
;       POP          R0             ;Restore R0
;
;       RTE
;
;*****
;
;       Timer A Interrupt
;*****
;
;       TAIN          .equ          $
;       BCLR         IRRTA          ;Clear IRRTA
;
;       BCLR         IENTA          ;Timer A Interrupt Disable
;
;       RTE
;
;       .end

```

2.28 スタンバイモードへの遷移

スタンバイモードへの遷移	使用機能	低消費電力モード：スタンバイモード
仕様		
<p>(1) スタンバイモードへの遷移を行ないます。</p> <p>(2) アクティブ（高速）モードでプログラム実行中に、TNT_0端子に接続したスイッチ入力のオンによりINT_0割込みを発生させ、INT_0割込み処理終了後にSLEEP命令の実行によりスタンバイモードへ遷移させます。アクティブ（高速）モードではLEDの点滅動作を行ない、スタンバイモードへ遷移するとLEDは点灯します（$P7_3$端子はスタンバイモード時、ハイインピーダンス状態）。</p> <p>(3) スタンバイモードで、TRQ_0端子に接続したスイッチ入力のオンによりIRQ_0割込みを発生させ、スタンバイモードを解除し、アクティブ（高速）モードへ遷移します。アクティブ（高速）モードへ遷移すると、再びLEDの点滅動作を行ないます。</p> <p>(4) 図1にTNT_0、TRQ_0端子に接続するスイッチの接続例を示します。</p> <div data-bbox="571 537 1021 862" style="text-align: center;"> </div> <p style="text-align: center;">図1 スタンバイモードへの遷移におけるスイッチ接続例</p>		
使用機能説明		
<p>(1) 本タスク例では、低消費電力モードのスタンバイモードへの遷移を行ないます。図2にスタンバイモードへのモード遷移図を示します。以下にスタンバイモードの機能の説明を示します。</p> <ul style="list-style-type: none"> アクティブモードで、SYSCR1のSSBYが"1"、LSONが"0"、およびTMAのTMA3が"0"のときSLEEP命令を実行すると、スタンバイモードに遷移します。 スタンバイモードではクロック発振回路からのクロック供給を停止するため、CPUおよび内蔵周辺機能が停止します。規定の電圧が与えられている限り、CPUのレジスタと一部の内蔵周辺機能の内部レジスタ、および内蔵RAMのデータは保持されています。さらに、RAMデータ保持電圧で規定した電圧が与えられている限り、内蔵RAMのデータは保持されています。このとき、I/Oポートはハイインピーダンス状態となります。 スタンバイモードの解除は、割込み（IRQ_1、IRQ_0）、\overline{RES}端子入力によって行われます。 割込みによるスタンバイモードの解除は、割込み要求が発生するとシステムクロックの発振が開始され、SYSCR1のSTS2～STS0により設定された時間が経過した後、安定したシステムクロックがLSI全体に供給されて、スタンバイモードは解除され、割込み例外処理を開始します。SYSCR2のMSONが"0"のときはアクティブ（高速）モードに、"1"のときはアクティブ（中速）モードに遷移します。 CCRのIビットが"1"のとき、あるいは、割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、スタンバイモードは解除されません。 \overline{RES}端子によるスタンバイモードの解除は、\overline{RES}端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、\overline{RES}端子を"High"レベルにすると、CPUはリセット例外処理を開始します。なお、システムクロックの発振開始と同時にLSI全体にシステムクロックが供給されます。\overline{RES}端子は、必ずシステムクロックの発振が安定するまで、"Low"レベルを保持しなければなりません。 本タスク例では、IRQ_0割込みによりスタンバイモードを解除します。 スタンバイモード解除後の発振安定時間は、SYSCR1のSTS2～STS0により設定します。発振安定時間は、10ms以上となるように設定しなければなりません。 本タスク例では、動作周波数に5MHzを使用しているため、発振安定時間を13.1072msに設定します。 <div data-bbox="287 1825 1308 2049" style="text-align: center;"> </div> <p style="text-align: center;">図2 スタンバイモードへの遷移におけるモード遷移図</p>		

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、スタンバイモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
$\overline{\text{TNT}}_0$	スイッチ入力1
$\overline{\text{TRQ}}_0$	スイッチ入力2

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりスタンバイモードへの遷移を行います。

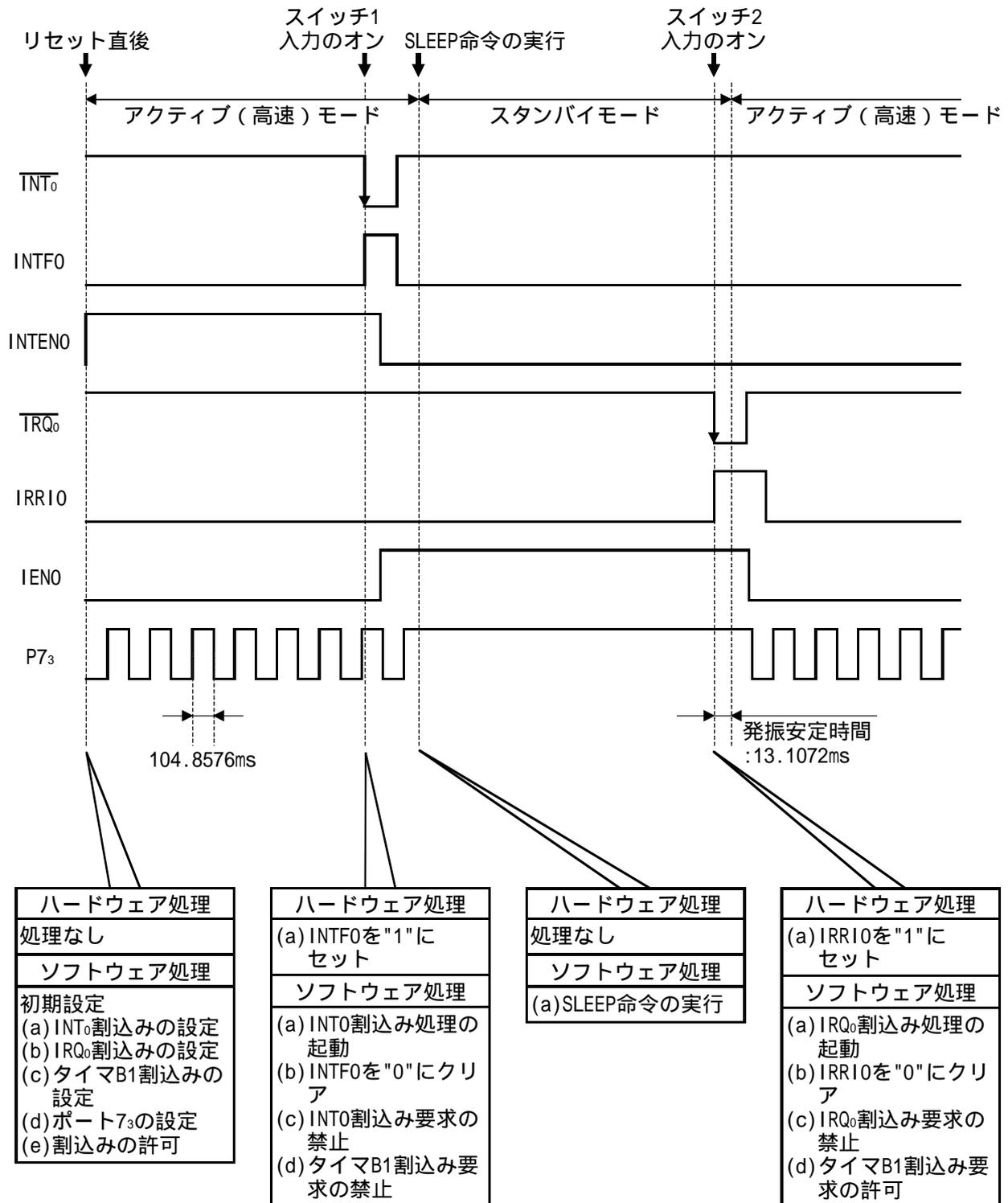


図3 スタンバイモードへの遷移の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、INT ₀ 割込みの設定、ポート7の設定、割込みの許可、LEDの制御、タイマA割込みの設定、スリープ（高速）モードへの遷移を行なう
スイッチ1オン	ITINT	INT ₀ 割込み処理ルーチンで、SWONFのセット、INT ₀ 割込みの禁止、タイマB1割込みの禁止を行なう
スタンバイモード解除	IQINT	IRQ ₀ 割込み処理ルーチンで、IRQ ₀ 割込みの禁止、タイマB1割込みの許可を行なう。
LED制御	TBINT	タイマB1割込み処理ルーチンで、LEDの制御を行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMA	タイマモードレジスタA : TMA=H'10のとき、タイマA機能をインターバルタイマ機能に、TCA入力クロックをシステムクロックの8192分周のクロックに設定	H'FFB0	H'10	
TMB1	タイマモードレジスタB1 : TMB1=H'79のとき、タイマB1機能をインターバルタイマ機能に、TCB1入力クロックをシステムクロックの2048分周のクロックに設定	H'FFB2	H'79	
TCB1	タイマカウンタB1 : システムクロックの2048分周のクロックを入力とする8ビットのアップカウンタ	H'FFB3	H'00	
PDR7	P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7	PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1
SYSCR1	SSBY	システムコントロールレジスタ1 (ソフトウェアスタンバイ) : SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモードあるいはウォッチモードに遷移。サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移	H'FFF0 ビット7	1
	STS2 STS1 STS0	システムコントロールレジスタ1 (スタンバイタイマセレクト2、1、0) : STS2="0", STS1="1", STS0="1"のとき、スタンバイモード解除後の発振安定時間を13.1072msに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="0" STS1="1" STS0="1"
	LSON	システムコントロールレジスタ1 (ロースピードオンフラグ) : LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定	H'FFF0 ビット3	0

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
SYSCR2	MSON	システムコントロールレジスタ2 (ミドルスピードオンフラグ) : MSON="0"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ(高速)モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ(高速)モードで動作	H'FFF1 ビット2	0
IEGR1	IEG0	割り込み許可エッジセレクトレジスタ1 (IRQ ₀ エッジセレクト) : IEG0="1"のとき、 $\overline{TRQ_0}$ 端子入力の立上りエッジを検出	H'FFF2 ビット0	1
IEGR2	INTEG0	割り込み許可エッジセレクトレジスタ2 (INT ₀ エッジセレクト) : INTEG0="1"のとき、 $\overline{INT_0}$ 端子入力の立上りエッジを検出	H'FFF3 ビット0	1
IENR1	IENRB1	割り込み許可レジスタ1 (タイマB1割り込みイネーブル) : IENRB1="0"のとき、タイマB1割り込み要求を禁止 : IENRB1="1"のとき、タイマB1割り込み要求を許可	H'FFF4 ビット7	1
	IRR10	割り込み許可レジスタ1 (IRQ ₀ 割り込みイネーブル) : IRR10="0"のとき、IRQ ₀ 割り込み要求を禁止 : IRR10="1"のとき、IRQ ₀ 割り込み要求を許可	H'FFF4 ビット0	0
IENR3	INTEN0	割り込み許可レジスタ3 (INT ₀ 割り込みイネーブル) : INTEN0="0"のとき、INT ₀ 割り込み要求を禁止 : INTEN0="1"のとき、INT ₀ 割り込み要求を許可	H'FFF6 ビット0	1
IRR1	IRRTB1	割り込み要求レジスタ1 (タイマB1割り込み要求フラグ) : IRRTB1="0"のとき、タイマB1割り込みが要求されていない : IRRTB1="1"のとき、タイマB1割り込みが要求されている	H'FFF7 ビット7	0
	IRR10	割り込み要求レジスタ1 (IRQ ₀ 割り込み要求フラグ) : IRR10="0"のとき、IRQ ₀ 割り込みが要求されていない : IRR10="1"のとき、IRQ ₀ 割り込みが要求されている	H'FFF7 ビット0	0
IRR3	INTF0	割り込み要求レジスタ3 (INT ₀ 割り込み要求フラグ) : INTF0="0"のとき、INT ₀ 割り込みが要求されていない : INTF0="1"のとき、INT ₀ 割り込みが要求されている	H'FFF9 ビット0	0

(4) 使用RAM説明

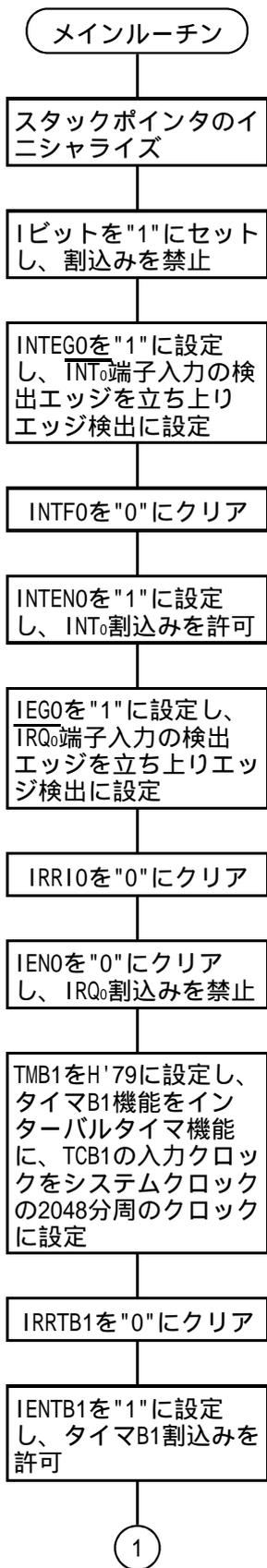
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

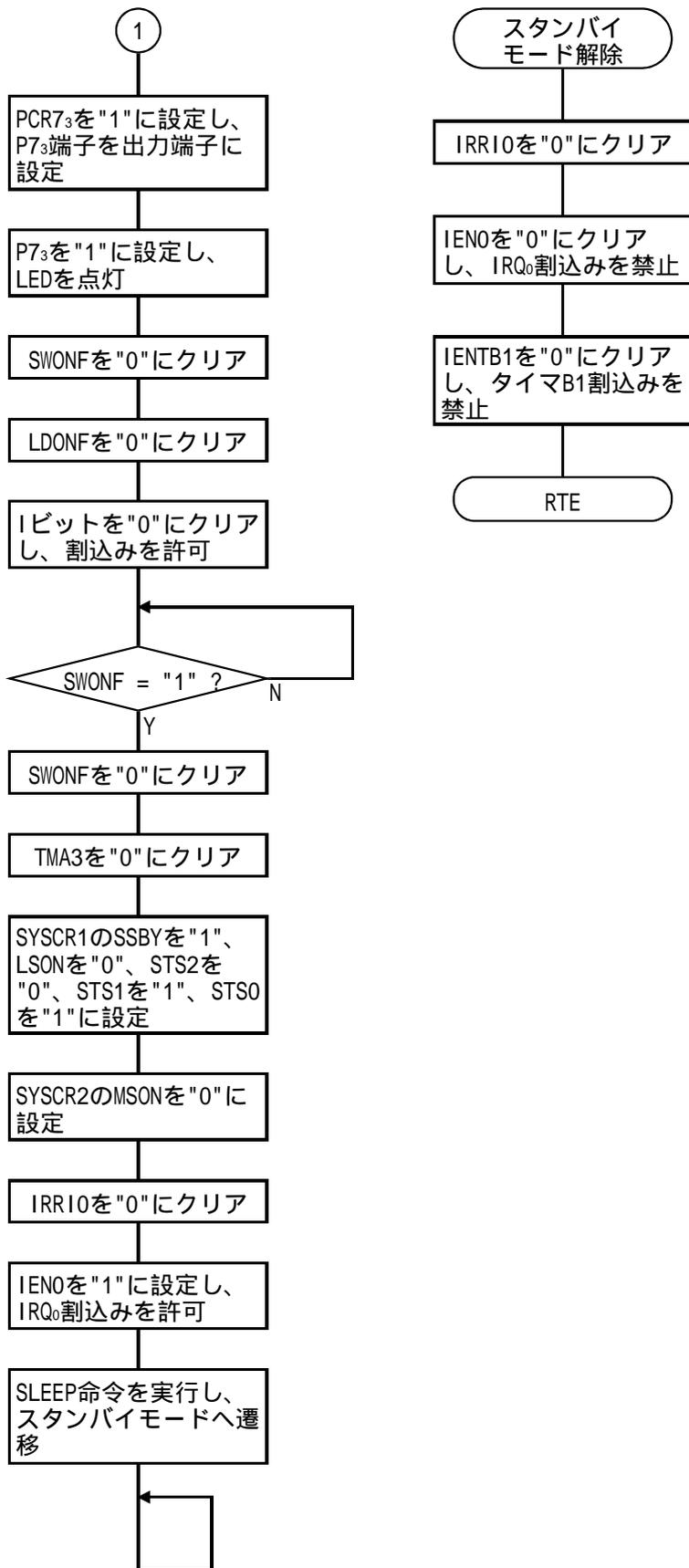
ラベル名		機能	アドレス	使用モジュール名
USRF	SWONF	スイッチ1入力のON/OFFを判定するフラグ	H'FB80 ビット0	メインルーチン スイッチ1オン
	LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット1	LED制御

フローチャート

(a) メインルーチン

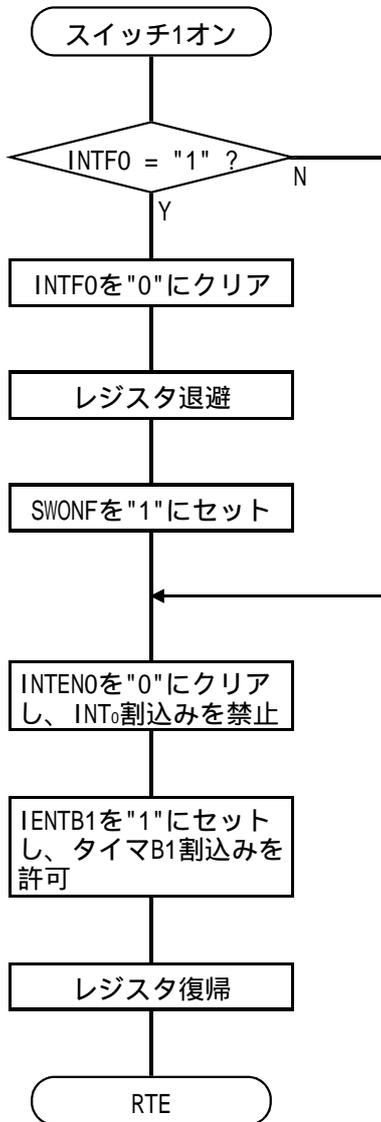


(b) IRQ0割り込み処理ルーチン

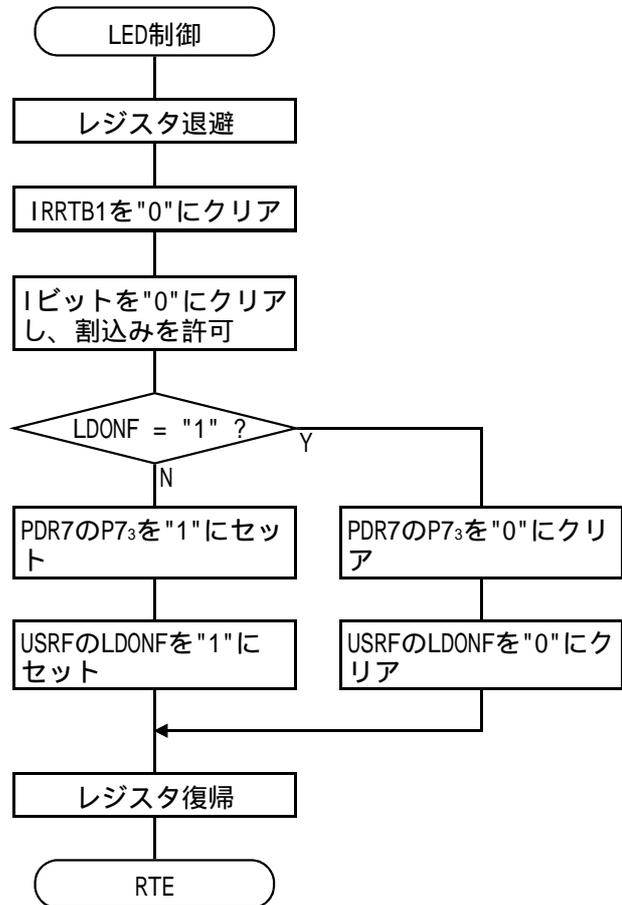


フローチャート

(c) INT₀割込み処理ルーチン



(d) タイマB1割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Transition to Standby Mode'
;
;           Function
;           : Power-Down Mode
;           Standby Mode
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu          300L
;
*****
;
;           Symbol Definition
;
*****
TMA          .equ          H'FFB0          ;Timer Mode Register A
TMB1         .equ          H'FFB2          ;Timer Mode Register B1
TCB1         .equ          H'FFB3          ;Timer Counter B1
PDR7         .equ          H'FFDA          ;Port Data Register 7
P73          .bequ         3,PDR7          ;Port Data Register 73
PCR7         .equ          H'FFEA          ;Port Control Register 7
PCR73        .bequ         3,PCR7          ;Port Control Register 73
SYSCR1       .equ          H'FFF0          ;System Control Register 1
SSBY         .bequ         7,SYSCR1        ;Software Standby
STS2         .bequ         6,SYSCR1        ;Standby Timer Select 2
STS1         .bequ         5,SYSCR1        ;Standby Timer Select 1
STS0         .bequ         4,SYSCR1        ;Standby Timer Select 0
LSON         .bequ         3,SYSCR1        ;Low Speed On Flag
MA1          .bequ         1,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 1
MA0          .bequ         0,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2       .equ          H'FFF1          ;System Control Register 2
NESEL        .bequ         4,SYSCR2        ;Noise Elimination Sampling Frequency Select
DTON         .bequ         3,SYSCR2        ;Direct Transfer On Flag
MSON         .bequ         2,SYSCR2        ;Middle Speed On Flag
SA1          .bequ         1,SYSCR2        ;Subactive Mode Clock Select 1
SA0          .bequ         0,SYSCR2        ;Subactive Mode Clock Select 0
IEGR1        .equ          H'FFF2          ;Interrupt Edge Select Register 1
IEG0         .bequ         0,IEGR1         ;IRQ0 Edge Select
IEGR2        .equ          H'FFF3          ;Interrupt Edge Select Register 2
INTEG0       .bequ         0,IEGR2         ;INT0 Edge Select
IENR1        .equ          H'FFF4          ;Interrupt Enable Register 1
IENTB1       .bequ         7,IENR1         ;Timer B1 Interrupt Enable
IENO         .bequ         0,IENR1         ;IRQ0 Interrupt Enable
IENR3        .equ          H'FFF6          ;Interrupt Enable Register 3
INTENO       .bequ         0,IENR3         ;INT0 Interrupt Enable
IRR1         .equ          H'FFF7          ;Interrupt Request Register 1
IRRTB1       .bequ         7,IRR1         ;Timer B1 Interrupt Request Flag
IRR10        .bequ         0,IRR1         ;IRQ0 Interrupt Request Flag
IRR3         .equ          H'FFF9          ;Interrupt Request Register 3
INTFO        .bequ         0,IRR3         ;INT0 Interrupt Request Flag

```

プログラムリスト

```

;
;
;*****
;
;          Ram Allocation
;*****
;
;
STACK      .equ      H'FF80      ;Stack Pointer
USRF       .equ      H'FB80      ;User Flag Area
SWONF     .bequ     0,USRF      ;Switch On Flag
LDONF     .bequ     1,USRF      ;LED On Flag
;
;*****
;
;          Vector Address
;*****
;
;
;          .org      H'0000
;          .data.w   MAIN          ;Reset Interrupt
;
;          .org      H'0008
;          .data.w   IQINT         ;IRQ0 Interrupt
;          .data.w   MAIN          ;IRQ1 Interrupt
;          .data.w   MAIN          ;IRQ2 Interrupt
;          .data.w   MAIN          ;IRQ3 Interrupt
;          .data.w   ITINT        ;INT0 - INT7 Interrupt
;
;          .org      H'0014
;          .data.w   MAIN          ;Timer A Interrupt
;          .data.w   TBINT        ;Timer B1 Interrupt
;
;          .org      H'0020
;          .data.w   MAIN          ;Timer X Interrupt
;          .data.w   MAIN          ;Timer V Interrupt
;
;          .org      H'0026
;          .data.w   MAIN          ;Sci1 Interrupt
;
;          .org      H'002A
;          .data.w   MAIN          ;Sci3 Interrupt
;          .data.w   MAIN          ;A/D Converter Interrupt
;          .data.w   MAIN          ;Sleep Interrupt
;
;*****
;
;          Main Program
;*****
;
;
;          .org      H'1000
;
MAIN      .equ      $
MOV.W    #STACK,SP      ;Initialize Stack Pointer
ORC      #H'80,CCR      ;Interrupt Disable
;
;          BSET      INTEGO        ;Set Rising Edge of INT0 Terminal Input
;          BCLR      INTFO         ;Clear INTFO
;          BSET      INTENO        ;INT0 Interrupt Enable
;
;          BSET      IEGO         ;Set Rising Edge of IRQ0 Terminal Input
;          BCLR      IRRIO        ;Clear IRRIO
;          BCLR      IENO         ;IRQ0 Interrupt Disable
;
;          MOV.B     #H'79,ROL     ;Initialize Timer B1 Function
;          MOV.B     ROL,@TMB1

```


プログラムリスト

```

MOV.B      ROL,@USRF
;
BCLR      INTENO      ;INT0 Interrupt Disable
BCLR      IENTB1      ;Timer B1 Interrupt Disable
;
POP       R0          ;Restore R0
;
ITINT9    .equ        $
RTE
;
;*****
;
;          Timer B1 Interrupt
;*****
;
TBINT     .equ        $
PUSH      R0          ;Store R0
BCLR      IRRTB1      ;Clear IRRTB1
;
ANDC      #'7F,CCR    ;Interrupt Enable
;
MOV.B     @USRF,ROL   ;
BTST     #1,ROL       ;LDONF = "1" ?
BNE      TBINT1       ;Yes.
;
BSET     P73          ;Turn On LED
BSET     #1,ROL       ;Set LDONF
BRA      TBINT9
;
TBINT1    .equ        $
BCLR     P73          ;Turn Off LED
BCLR     #1,ROL       ;Clear LDONF
;
TBINT9    .equ        $
MOV.B    ROL,@USRF
;
POP       R0          ;Restore R0
;
RTE
;
.end

```

2.29 ウォッチモードへの遷移

ウォッチモードへの遷移	使用機能	低消費電力モード：ウォッチモード
仕様		
	<p>(1) ウォッチモードへの遷移を行ないます。</p> <p>(2) アクティブモードでSYSCR1のSSBYが"1"、TMAのTMA3が"1"のときSLEEP命令を実行することによりウォッチモードへ遷移します。</p> <p>(3) ウォッチモードでSYSCR1のLSONが"1"のとき、タイマA割込み要求が発生することによりサブアクティブモードへ遷移します。</p> <p>(4) サブアクティブモードで、タイマA割込み要求の回数をカウントし、再びウォッチモードへ遷移します。</p> <p>(5) タイマA割込み要求は、0.5secごとに発生します。また、タイマA割込み処理で0.5secごとにLEDを交互に点灯、消灯させます。</p> <p>(6) タイマA割込みが120回要求され、60sec経過するとサブアクティブモードからアクティブ（高速）モードへ直接遷移させて終了します。</p>	
使用機能説明		
	<p>(1) 本タスク例では、低消費電力モードのウォッチモードへの遷移を行ないます。図2にウォッチモードへのモード遷移図を示します。以下にウォッチモードの機能の説明を示します。</p> <ul style="list-style-type: none"> ・アクティブモード、サブアクティブモードでSYSCR1のSSBYが"1"、TMAのTMA3が"1"のときSLEEP命令を実行すると、ウォッチモードへ遷移します。 ・ウォッチモードでは、タイマA以外の内蔵周辺機能は動作を停止します。 ・規定の電圧が与えられている限り、CPUと一部の内蔵周辺機能の内部レジスタ、および内蔵RAMの内容は保持され、I/Oポートは遷移前の状態を保持します。 ・ウォッチモードの解除は、割込み（IRQ₀、タイマA）、RES端子入力によって行われます。 ・割込みによる解除は、タイマA割込み、またはIRQ₀割込み要求が発生するとウォッチモードで解除され、SYSCR1のLSONとSYSCR2のMSONの組み合わせで、LSON="0"かつMSON="0"のときはアクティブ（高速）モードに、LSON="0"かつMSON="1"のときはアクティブ（中速）モードに、LSON="1"のときはサブアクティブモードに遷移します。 ・アクティブモードに遷移するときは、SYSCR1のSTS2～STS0により設定した時間が経過した後、安定したクロックがLSI全体に供給され、割込み例外処理を開始します。 ・CCRのIビットが"1"の場合、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、ウォッチモードは解除されません。 ・RES端子による解除は、RES端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、RES端子を"High"レベルにすると、CPUはリセット例外処理を開始します。 ・システムクロックの発振開始と同時にLSI全体にシステムクロックが供給されます。 ・RES端子は、必ずシステムクロックの発振が安定するまで、"Low"レベルを保持しなければなりません。 ・本タスク例では、ウォッチモードの解除にタイマA割込みを使用します。なお、ウォッチモードの解除後はサブアクティブモードへ遷移させます。 ・サブアクティブモードでSYSCR1のSSBYを"1"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由し、SYSCR1のSTS2～STS0により設定された時間を経過した後、直接、アクティブ（高速）モードに遷移します。 ・サブアクティブモード解除後の発振安定時間は、SYSCR1のSTS2～STS0により設定します。発振安定時間は、10ms以上となるように設定しなければなりません。 ・本タスク例では、動作周波数に5MHzを使用しているため、発振安定時間を13.1072msに設定しています。 	

使用機能説明	
--------	--

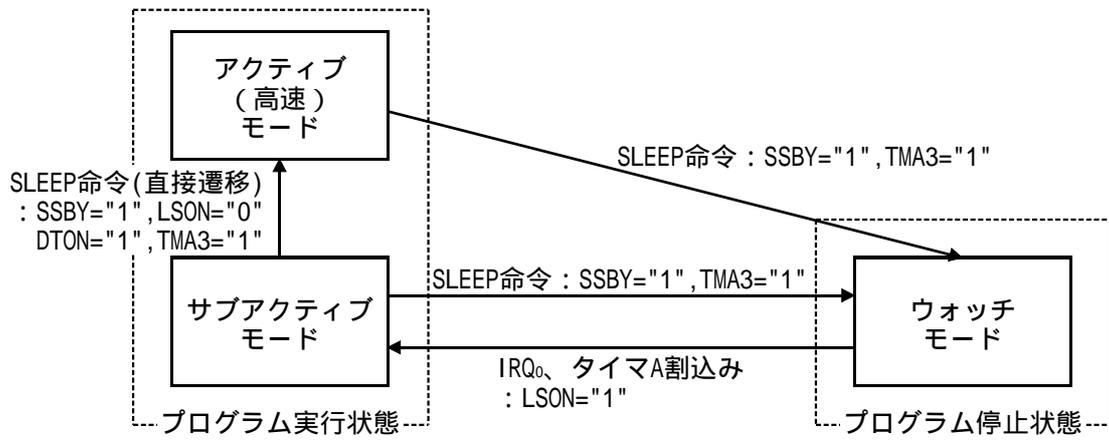


図2 ウォッチモードへの遷移におけるモード遷移図

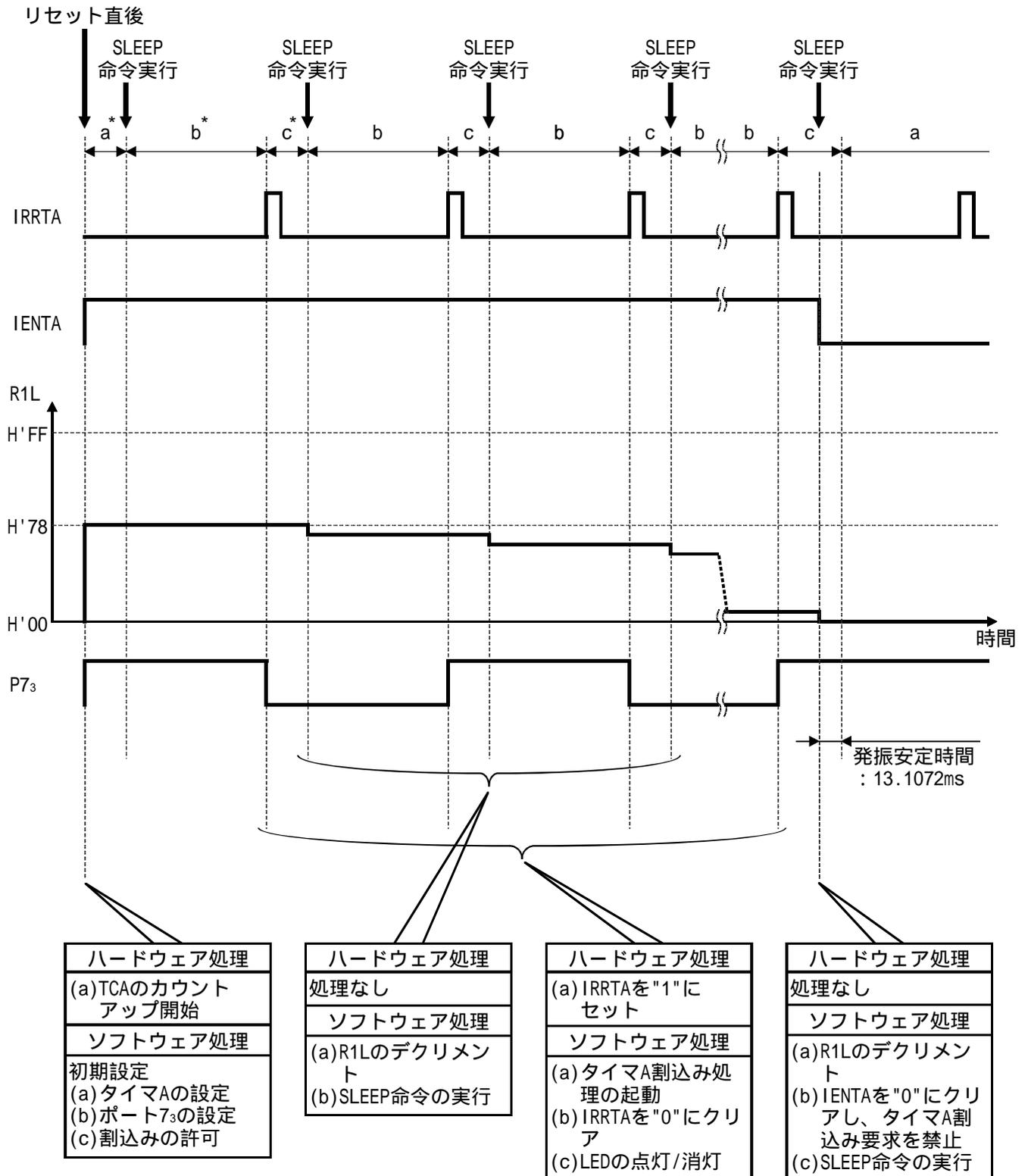
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、ウォッチモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
TMA	タイマA時計用タイムベース機能、およびTCAオーバフロー周期の設定
TCA	時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ
IRRТА	タイマA割込み要求の有無を反映
IENTA	タイマA割込み要求の許可、または禁止を設定

動作原理

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりウォッチモードへの遷移を行います。



【注】* a : アクティブ (高速) モード
 b : ウォッチモード
 c : サブアクティブモード

図3 ウォッチモードへの遷移の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、タイマA割込みの設定、ポート7の設定、割込みの許可、ウォッチモードへの遷移、タイマA割込みの禁止を行なう
LED制御	TAINT	タイマA割込み処理ルーチンで、LEDの制御を行なう
直接遷移	DTINT	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMA	タイマモードレジスタA ：TMA=H'19のとき、タイマA機能を時計用タイムベース機能に、TCAのオーバフロー周期を0.5secに設定	H'FFB0	H'19
TCA	タイマカウンタA ：PSW出力クロックを入力とし、時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ	H'FFB1	H'00
PDR7	P7 ₃ ポートデータレジスタ7 (ポートデータレジスタ7 ₃) ：P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" ：P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	1
PCR7	PCR7 ₃ ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) ：PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1
SYSCR1	SSBY システムコントロールレジスタ1 (ソフトウェアスタンバイ) ：SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモード、あるいはウォッチモードに遷移。サブアクティブモードでSLEEP命令実行後サブスリープモードに遷移	H'FFF0 ビット7	1
	STS2 STS1 STS0 システムコントロールレジスタ1 (スタンバイタイムセレクト2、1、0) ：STS2="0", STS1="1", STS0="1"のとき、スタンバイモード、ウォッチモード解除後の発振安定時間を13.1072msに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="0" STS1="1" STS0="1"
	LSON システムコントロールレジスタ1 (ロースピードオンフラグ) ：LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定 ：LSON="1"のとき、ウォッチモードを解除時、CPUの動作クロックをサブシステムクロックに設定	H'FFF0 ビット3	1
SYSCR2	DTON システムコントロールレジスタ2 (ダイレクトトランスファオンフラグ) ：DTON="0"のとき、アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移。サブアクティブモードでSLEEP命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移 ：DTON="1"のとき、サブアクティブモードでSLEEP命令を実行したとき、アクティブ (高速) モード (SSBY="1", TMA3="1", LSON="0", MSON="0"のとき)、またはアクティブ (中速) モード (SSBY="1", TMA="1", LSON="0", MSON="1"のとき) に直接遷移	H'FFF1 ビット3	0

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
SYSCR2	MSON	システムコントロールレジスタ2 (ミドルスピードオンフラグ) : MSON="0"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ(高速)モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ(高速)モードで動作	H'FFF1 ビット2	0
	SA1 SA0	システムコントロールレジスタ2 (サブアクティブモードクロックセレクト1、0) : SA1="0"、SA0="0"のとき、サブアクティブモードのCPUの動作クロックを $\omega/8$ に設定	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"
IENR1	IENTA	割込み許可レジスタ1(タイマA割込みイネーブル) : IENTA="0"のとき、タイマA割込み要求を禁止 : IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IENR2	IENDT	割込み許可レジスタ2(直接遷移割込みイネーブル) : IENDT="0"のとき、直接遷移による割込み要求を禁止 : IENDT="1"のとき、直接遷移による割込み要求を許可	H'FFF5 ビット7	1
IRR1	IRRTA	割込み要求レジスタ1(タイマA割込み要求フラグ) : IRRTA="0"のとき、タイマA割込みが要求されていない : IRRTA="1"のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0
IRR2	IRRDT	割込み要求レジスタ2(直接遷移割込み要求フラグ) : IRRDT="0"のとき、直接遷移による割込みが要求されていない : IRRDT="1"のとき、直接遷移による割込みが要求されている	H'FFF8 ビット7	0

(4) 使用RAM説明

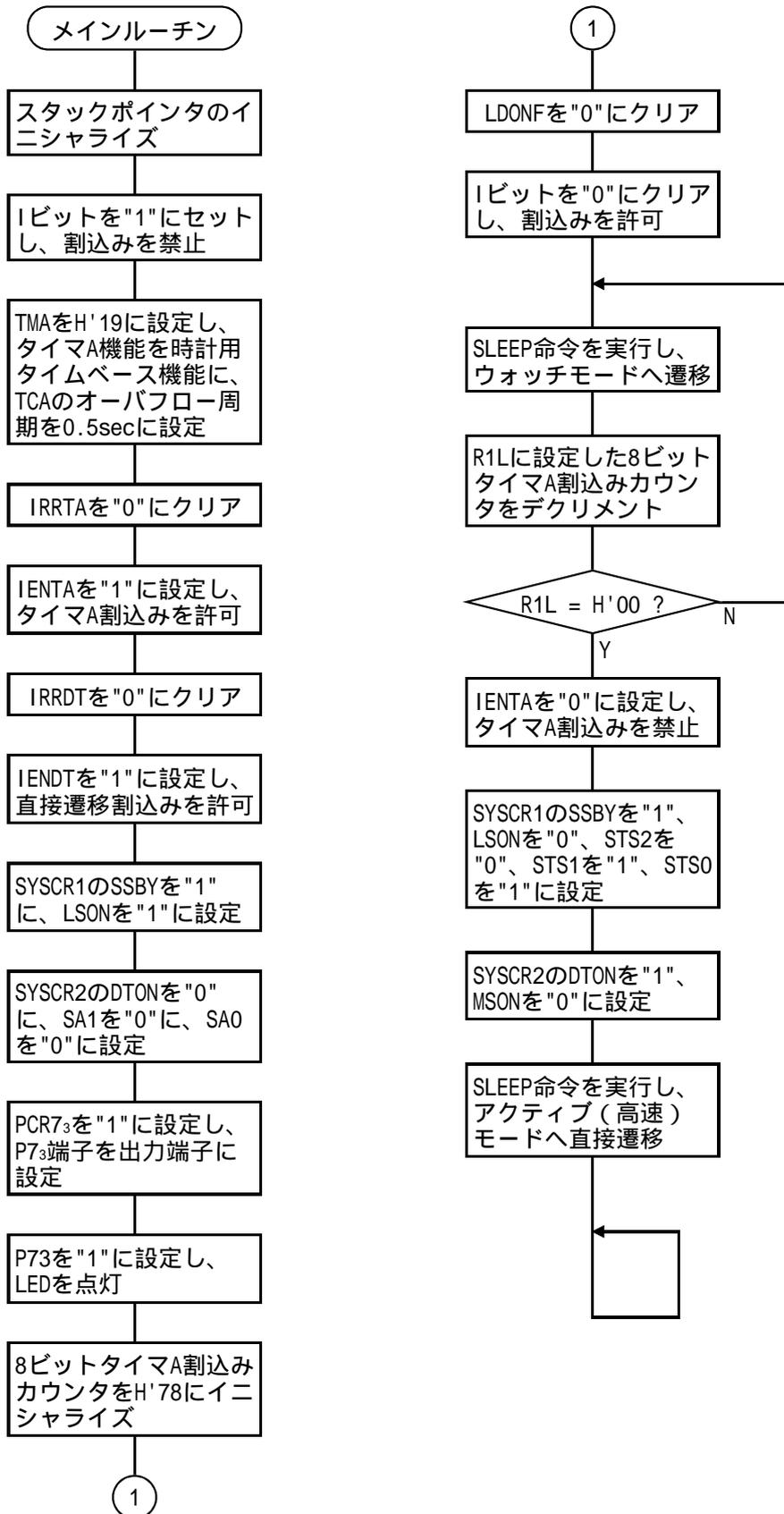
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

ラベル名		機能	アドレス	使用モジュール名
USRF	LDONF	LEDのON/OFFを判定するフラグ	H'FB80 ビット0	LED制御

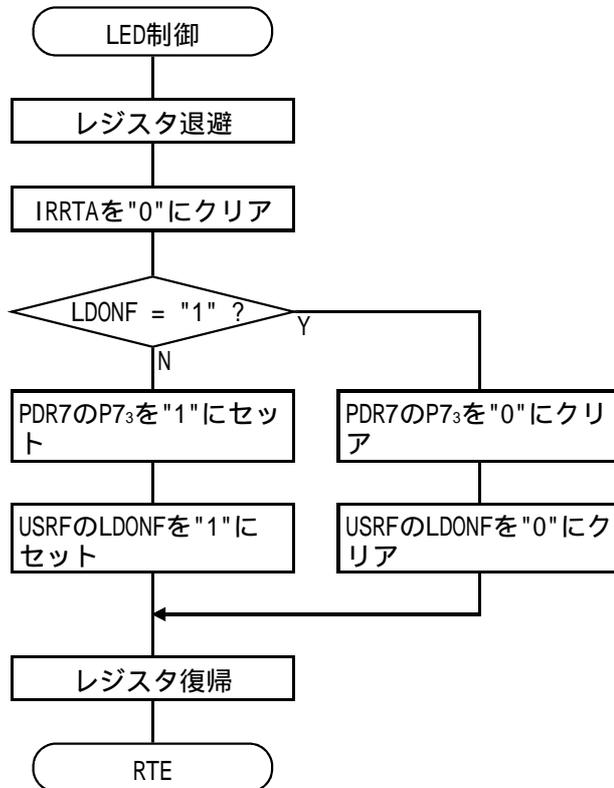
フローチャート

(a) メインルーチン

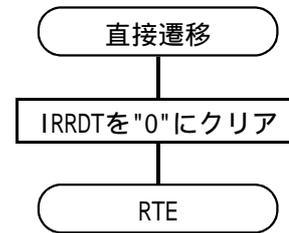


フローチャート

(b) タイマA割り込み処理ルーチン



(c) 直接遷移割り込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Transition to Watch Mode'
;
;           Function
;           : Power-Down Mode
;           Watch Mode
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
*****
;
;           .cpu          300L
;
*****
;
;           Symbol Definition
;
*****
TMA          .equ          H'FFB0          ;Timer Mode Register A
TCA          .equ          H'FFB1          ;Timer Counter A
PDR7         .equ          H'FFDA          ;Port Data Register 7
P73         .bequ         3,PDR7          ;Port Data Register 73
PCR7         .equ          H'FFEA          ;Port Control Register 7
PCR73        .bequ         3,PCR7         ;Port Control Register 73
SYSCR1       .equ          H'FFF0          ;System Control Register 1
SSBY         .bequ         7,SYSCR1       ;Software Standby
STS2         .bequ         6,SYSCR1       ;Standby Timer Select 2
STS1         .bequ         5,SYSCR1       ;Standby Timer Select 1
STS0         .bequ         4,SYSCR1       ;Standby Timer Select 0
LSON         .bequ         3,SYSCR1       ;Low Speed On Flag
MA1          .bequ         1,SYSCR1       ;Active (Middle-Speed) Mode Clock Select 1
MA0          .bequ         0,SYSCR1       ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2       .equ          H'FFF1          ;System Control Register 2
NESEL        .bequ         4,SYSCR2       ;Noise Elimination Sampling Frequency Select
DTON         .bequ         3,SYSCR2       ;Direct Transfer On Flag
MSON         .bequ         2,SYSCR2       ;Middle Speed On Flag
SA1          .bequ         1,SYSCR2       ;Subactive Mode Clock Select 1
SA0          .bequ         0,SYSCR2       ;Subactive Mode Clock Select 0
IENR1        .equ          H'FFF4          ;Interrupt Enable Register 1
IENTA        .bequ         6,IENR1        ;Timer A Interrupt Enable
IENR2        .equ          H'FFF5          ;Interrupt Enable Register 2
IENDT        .bequ         7,IENR2        ;Direct Transfer Interrupt Enable
IRR1         .equ          H'FFF7          ;Interrupt Request Register 1
IRRTA        .bequ         6,IRR1         ;Timer A Interrupt Request Flag
IRR2         .equ          H'FFF8          ;Interrupt Request Register 2
IRRDT        .bequ         7,IRR2         ;Direct Transfer Interrupt Request Flag
;
;
;           Ram Allocation
;
*****
STACK        .equ          H'FF80          ;Stack Pointer
USRF         .equ          H'FB80          ;User Flag Area

```

プログラムリスト

```

LDONF      .bequ      0,USRF      ;LED On Flag
;
;*****
;
;          Vector Address
;*****
;
;          .org      H'0000
;          .data.w   MAIN        ;Reset Interrupt
;
;          .org      H'0008
;          .data.w   MAIN        ;IRQ0 Interrupt
;          .data.w   MAIN        ;IRQ1 Interrupt
;          .data.w   MAIN        ;IRQ2 Interrupt
;          .data.w   MAIN        ;IRQ3 Interrupt
;          .data.w   MAIN        ;INT0 - INT7 Interrupt
;
;          .org      H'0014
;          .data.w   TAINT       ;Timer A Interrupt
;          .data.w   MAIN        ;Timer B1 Interrupt
;
;          .org      H'0020
;          .data.w   MAIN        ;Timer X Interrupt
;          .data.w   MAIN        ;Timer V Interrupt
;
;          .org      H'0026
;          .data.w   MAIN        ;Sci1 Interrupt
;
;          .org      H'002A
;          .data.w   MAIN        ;Sci3 Interrupt
;          .data.w   MAIN        ;A/D Converter Interrupt
;          .data.w   DTINT       ;Sleep Interrupt
;
;*****
;
;          Main Program
;*****
;
;          .org      H'1000
;
;MAIN      .equ      $
;          MOV.W     #STACK,SP   ;Initialize Stack Pointer
;          ORC       #H'80,CCR    ;Interrupt Disable
;
;          MOV.B     #H'19,R0L    ;Initialize Timer A Function
;          MOV.B     R0L,@TMA     ;Clear IRRTA
;          BCLR      IRRTA        ;Timer A Interrupt Enable
;          BSET      IENTA
;
;          BCLR      IRRDT        ;Clear IRRDT
;          BSET      IENDT       ;Direct Transfer Interrupt Enable
;
;          MOV.W     #H'8FE0,R0   ;Initialize Function of Watch Mode 1
;          MOV.B     R0H,@SYSCR1  ;Initialize Function of Watch Mode 2
;          MOV.B     R0L,@SYSCR2
;
;          BSET      P73         ;Initialize P73
;          BSET      PCR73       ;Initialize P73 Output Port
;
;          MOV.B     #H'78,R1L    ;Initialize 8bit Timer A Interrupt Counter
;
;          MOV.B     #H'01,R0L

```

プログラムリスト

```

;
;      MOV.B      ROL,@USRF      ;Initialize LDONF
;
;      ANDC      #H'7F,CCR      ;Interrupt Enable
;
MAIN1  .equ      $
      SLEEP      ;Transition to Watch Mode
;
;      DEC      R1L      ;Decrement 8bit Timer A Interrupt Counter
;      BNE      MAIN1    ;8bit Timer A Interrupt Counter = H'00 ? No.
;
;      BCLR     IENTA     ;Timer A Interrupt Disable
;
;      MOV.W    #H'B7E8,R0
;      MOV.B    ROH,@SYSCR1  ;Initialize Function of Active (High-Speed) Mode 1
;      MOV.B    ROL,@SYSCR2  ;Initialize Function of Active (High-Speed) Mode 2
;
;      SLEEP      ;Transition to Active (High-Speed) Mode
;
MAIN9  .equ      $
      BRA      MAIN9
;
;*****
;
;      Timer A Interrupt
;*****
;
TAINT  .equ      $
      BCLR     IRRTA     ;Clear IRRTA
      PUSH    R0        ;Store R0
;
;      MOV.B    @USRF,ROL
;      BTST    #0,ROL    ;LDONF = "1" ?
;      BNE     TAINT1    ;Yes.
;
;      BSET    P73      ;Turn On LED
;      BSET    #0,ROL   ;Set LDONF
;      BRA     TAINT9
;
TAINT1 .equ      $
      BCLR     P73      ;Turn Off LED
      BCLR     #0,ROL   ;Clear LDONF
;
TAINT9 .equ      $
      MOV.B    ROL,@USRF
      POP     R0        ;Restore R0
;
;      RTE
;
;*****
;
;      Direct Transfer Interrupt
;*****
;
DTINT  .equ      $
      BCLR     IRRDT    ;Clear IRRDT
;
;      RTE
;
;      .end

```

2.30 サブスリープモードへの遷移

サブスリープモードへの遷移	使用機能	低消費電力モード：サブスリープモード
<p>仕様</p> <p>(1) サブスリープモードへの遷移を行ないます。 (2) サブアクティブモードで、SYSCR1のSSBYが"0"、LSONが"1"、TMAのTMA3が"1"のときSLEEP命令を実行することにより、サブスリープモードへ遷移します。 (3) サブスリープモードは、タイマA割込みにより解除しサブアクティブモードへ復帰します。 (4) タイマA割込み処理の中で、LEDの制御とタイマA割込み発生回数のカウントを行ないます。タイマA割込みは0.5secごとに発生し、タイマA割込みが120回発生したところで、タイマA割込みを要求を禁止し、終了します。LEDは0.5secごとに点灯/消灯を繰り返します。 (5) タイマA割込みが発生し、サブアクティブモードへ遷移すると、タイマA割込み回数の判定を行い、再びウォッチモードへ遷移させます。この処理を、タイマA割込みが120回発生するまで繰り返します。</p>		
<p>使用機能説明</p> <p>(1) 本タスク例では、低消費電力モードのサブスリープモードへの遷移を行ないます。図1にサブスリープモードへのモード遷移図を示します。以下にサブスリープモードの機能の説明を示します。</p> <ul style="list-style-type: none"> サブアクティブモードでSYSCR1のSSBYが"0"、LSONが"1"、TMAのTMA3が"1"のときSLEEP命令を実行すると、サブスリープモードに遷移します。 サブスリープモードでは、タイマA、タイマC以外の内蔵周辺機能は動作を停止します。 規定の電圧が与えられている限り、CPUと一部の内蔵周辺機能の内部レジスタ、内蔵RAMの内容は保持され、I/Oポートは遷移前の状態を保持します。 サブスリープモードの解除は、タイマA、IRQ₃～IRQ₀、INT₇～INT₀割込み、RES端子入力によって行われます。 割込みによる解除は、割込み要求が発生するとサブスリープモードは解除され、割込み例外処理を開始します。 CCRのIビットが"1"の場合、あるいは割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、サブスリープモードは解除されません。 RES端子による解除は、RES端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、RES端子を"High"レベルにするとCPUはリセット例外処理を開始します。システムクロックの発振開始と同時にLSI全体にシステムクロックが供給されます。 RES端子は、必ずシステムクロックの発振が安定するまで、"Low"レベルを保持しなければなりません。 本タスク例では、サブスリープモードの解除にタイマA割込みを使用します。なお、サブスリープモードの解除後は、サブアクティブモードに遷移します。 アクティブ（高速）モードで、SYSCR1のSSBYを"1"、LSONを"1"、SYSCR2のDTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由してサブアクティブモードに遷移します。 サブアクティブモードでSYSCR1のSSBYを"1"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由し、SYSCR1のSTS2～STS0により設定された時間を経過した後、直接、アクティブ（高速）モードに遷移します。 サブアクティブモード解除後の発振安定時間は、SYSCR1のSTS2～STS0により設定します。発振安定時間は、10ms以上となるように設定しなければなりません。 本タスク例では、動作周波数に5MHzを使用しているため、発振安定時間を13.1072msに設定しています。 <p>図1 サブスリープモードへの遷移におけるモード遷移図</p> <pre> graph TD subgraph "プログラム実行状態" AM[アクティブ (高速) モード] SAM[サブアクティブモード] end subgraph "プログラム停止状態" SSM[サブスリープモード] end AM -- "SLEEP命令 (直接遷移) : LSON='1', SSBY='1', TMA3='1', DTON='1'" --> SAM SAM -- "SLEEP命令 (直接遷移) : LSON='0', MSON='0', SSBY='1', TMA3='1', DTON='1'" --> AM SAM -- "SLEEP命令 : SSBY='0', LSON='1', TMA3='1'" --> SSM SSM -- "IRQ3 ~ IRQ0, INT, タイマA 割込み" --> SAM </pre>		

使用機能説明

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、サブスリープモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
TMA	タイマA時計用タイムベース機能、およびTCAオーバフロー周期の設定
TCA	時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ
IRRTA	タイマA割込み要求の有無を反映
IENTA	タイマA割込み要求の許可、または禁止を設定

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりサブスリープモードへの遷移を行ないます。

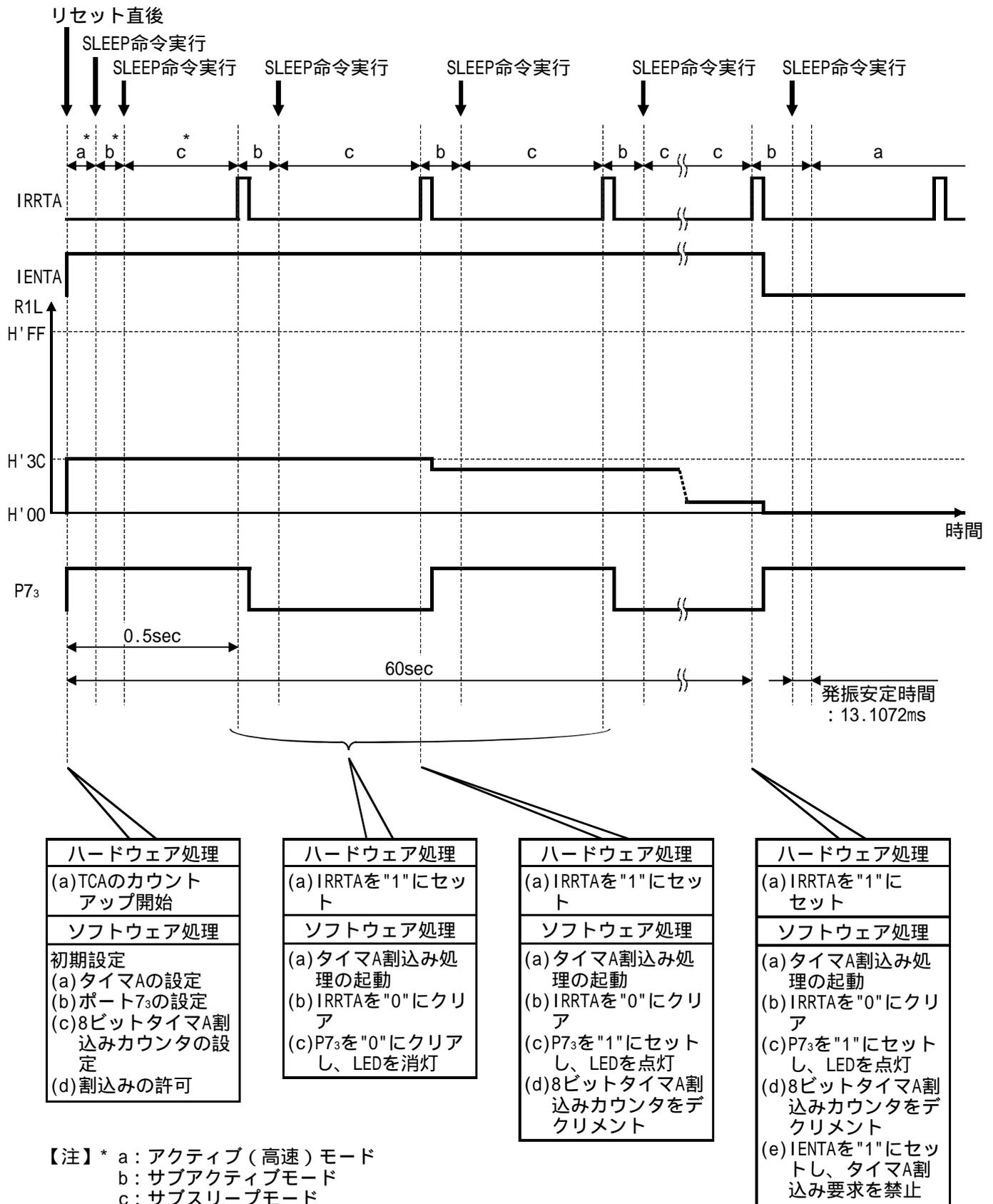


図2 サブスリープモードへの遷移の動作原理

サブスリープモードへの遷移	使用機能	低消費電力モード：サブスリープモード
---------------	------	--------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、タイマA割込みの設定、ポート7の設定、8ビットタイマA割込みカウンタの設定、割込みの許可、サブアクティブモードへの遷移、サブスリープモードへの遷移、アクティブ(高速)モードへの遷移を行なう
LED制御	TAINT	タイマA割込み処理ルーチンで、LEDの制御、8ビットタイマA割込みカウンタのデクリメント、60sec経過後、タイマA割込み要求の禁止を行なう
直接遷移	DTINT	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMA	タイマモードレジスタA : TMA=H'19のとき、タイマA機能を時計用タイムベース機能に、TCAのオーバフロー周期を0.5secに設定	H'FFB0	H'19
TCA	タイマカウンタA : PSW出力クロックを入力とし、時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ	H'FFB1	H'00
PDR7	P7 ₃ ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	1
PCR7	PCR7 ₃ ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1
SSBY	システムコントロールレジスタ1 (ソフトウェアスタンバイ) : SSBY="0"のとき、アクティブモードでSLEEP命令実行後、スリープモードに遷移。サブアクティブモードでSLEEP命令実行後、サブスリープモードに遷移 : SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモード、あるいはウォッチモードに遷移。サブアクティブモードでSLEEP命令実行後サブスリープモードに遷移	H'FFF0 ビット7	1
SYSCR1	STS2 STS1 STS0 システムコントロールレジスタ1 (スタンバイタイムセレクト2、1、0) : STS2="0", STS1="1", STS0="1"のとき、スタンバイモード、サブスリープ解除後の発振安定時間を13.1072msに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="0" STS1="1" STS0="1"
LSON	システムコントロールレジスタ1 (ロースピードオンフラグ) : LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定 : LSON="1"のとき、ウォッチモードを解除時、CPUの動作クロックをサブシステムクロックに設定	H'FFF0 ビット3	1

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
SYSCR2	DTON	H'FFF1 ビット3	1
	MSON	H'FFF1 ビット2	0
	SA1 SA0	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"
IENR1	IENTA	H'FFF4 ビット6	1
IENR2	IENDT	H'FFF5 ビット7	1
IRR1	IRRTA	H'FFF7 ビット6	0
IRR2	IRRDT	H'FFF8 ビット7	0

(4) 使用RAM説明

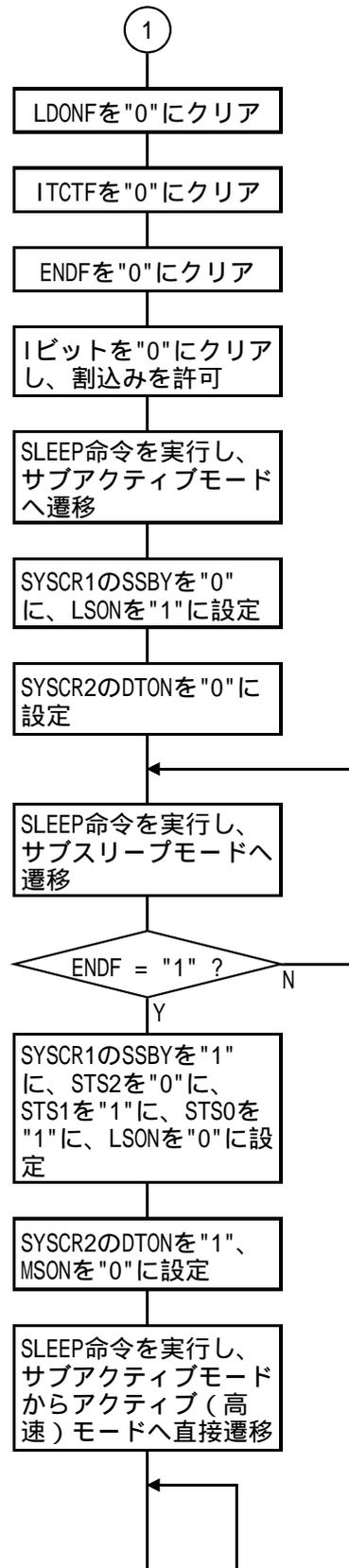
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
USRF	LDONF	H'FB80 ビット0	LED制御
	ITCNF	H'FB80 ビット1	LED制御
	ENDF	H'FB80 ビット2	メインルーチン

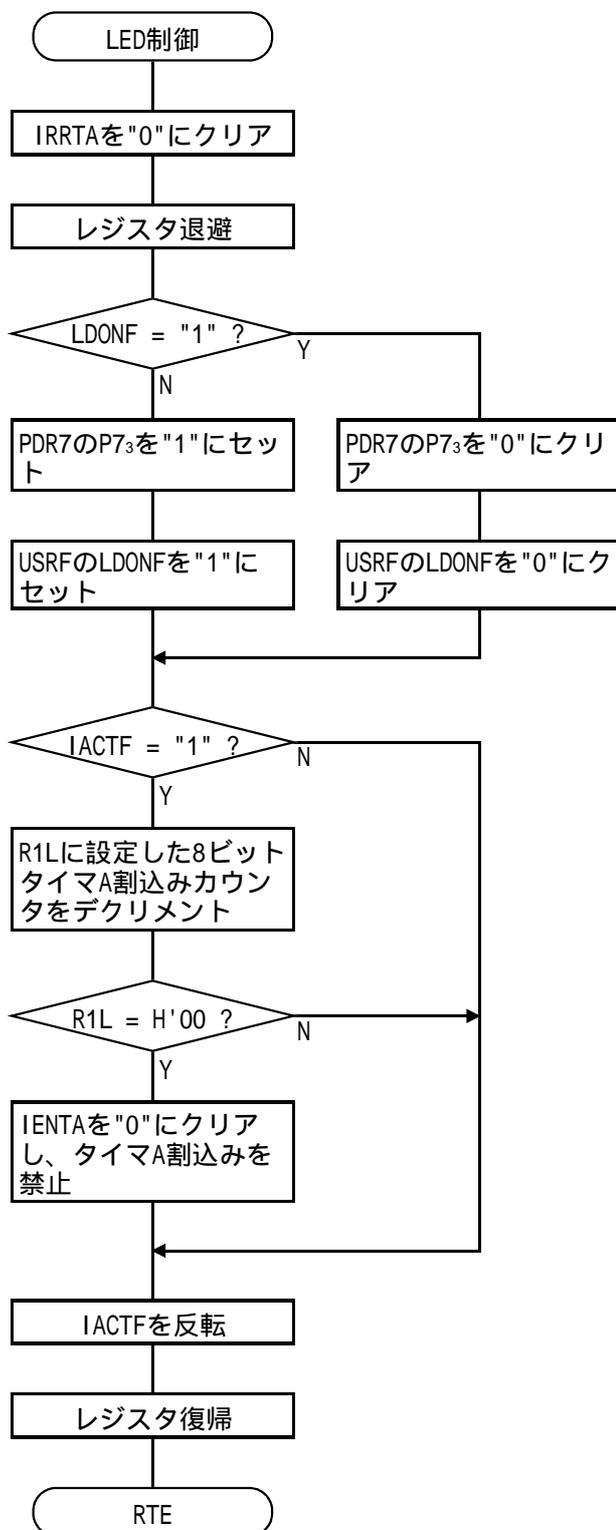
フローチャート

(a) メインルーチン

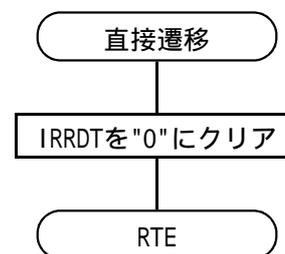


フローチャート

(b) タイマA割込み処理ルーチン



(c) 直接遷移割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Transition to Sucsleep Mode'
;
;           Function
;           : Power-Down Mode
;           : Subsleep Mode
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu           300L
;
*****
;
;           Symbol Definition
;
*****
TMA           .equ           H'FFB0           ;Timer Mode Register A
TCA           .equ           H'FFB1           ;Timer Counter A
PDR7          .equ           H'FFDA           ;Port Data Register 7
P73          .bequ          3,PDR7           ;Port Data Register 73
PCR7          .equ           H'FFEA           ;Port Control Register 7
PCR73        .bequ          3,PCR7           ;Port Control Register 73
SYSCR1        .equ           H'FFF0           ;System Control Register 1
SSBY          .bequ          7,SYSCR1        ;Software Standby
STS2          .bequ          6,SYSCR1        ;Standby Timer Select 2
STS1          .bequ          5,SYSCR1        ;Standby Timer Select 1
STS0          .bequ          4,SYSCR1        ;Standby Timer Select 0
LSON          .bequ          3,SYSCR1        ;Low Speed On Flag
MA1           .bequ          1,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 1
MA0           .bequ          0,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2        .equ           H'FFF1           ;System Control Register 2
NESEL         .bequ          4,SYSCR2        ;Noise Elimination Sampling Frequency Select
DTON          .bequ          3,SYSCR2        ;Direct Transfer On Flag
MSON          .bequ          2,SYSCR2        ;Middle Speed On Flag
SA1           .bequ          1,SYSCR2        ;Subactive Mode Clock Select 1
SA0           .bequ          0,SYSCR2        ;Subactive Mode Clock Select 0
IENR1         .equ           H'FFF4           ;Interrupt Enable Register 1
IENTA         .bequ          6,IENR1         ;Timer A Interrupt Enable
IENR2         .equ           H'FFF5           ;Interrupt Enable Register 2
IENDT         .bequ          7,IENR2         ;Direct Transfer Interrupt Enable
IRR1          .equ           H'FFF7           ;Interrupt Request Register 1
IRRTA         .bequ          6,IRR1          ;Timer A Interrupt Request Flag
IRR2          .equ           H'FFF8           ;Interrupt Request Register 2
IRRDT         .bequ          7,IRR2          ;Direct Transfer Interrupt Request Flag
;
;
;           Ram Allocation
;
*****
STACK         .equ           H'FF80           ;Stack Pointer
USRF          .equ           H'FB80           ;User Flag Area

```

プログラムリスト

```

LDONF      .bequ      0,USRF      ;LED On Flag
IACF       .bequ      1,USRF      ;Timer A Interrupt Counter Flag
ENDF       .bequ      2,USRF      ;End Flag
;
;
;*****
;
;          Vector Address
;*****
;
;
;          .org      H'0000
;          .data.w   MAIN      ;Reset Interrupt
;
;
;          .org      H'0008
;          .data.w   MAIN      ;IRQ0 Interrupt
;          .data.w   MAIN      ;IRQ1 Interrupt
;          .data.w   MAIN      ;IRQ2 Interrupt
;          .data.w   MAIN      ;IRQ3 Interrupt
;          .data.w   MAIN      ;INT0 - INT7 Interrupt
;
;
;          .org      H'0014
;          .data.w   TAINT      ;Timer A Interrupt
;          .data.w   MAIN      ;Timer B1 Interrupt
;
;
;          .org      H'0020
;          .data.w   MAIN      ;Timer X Interrupt
;          .data.w   MAIN      ;Timer V Interrupt
;
;
;          .org      H'0026
;          .data.w   MAIN      ;Sci1 Interrupt
;
;
;          .org      H'002A
;          .data.w   MAIN      ;Sci3 Interrupt
;          .data.w   MAIN      ;A/D Converter Interrupt
;          .data.w   DTINT     ;Sleep Interrupt
;
;*****
;
;          Main Program
;*****
;
;
;          .org      H'1000
;
;MAIN      .equ      $
;          MOV.W    #STACK,SP    ;Initialize Stack Pointer
;          ORC     #H'80,CCR      ;Interrupt Disable
;
;          MOV.B    #H'19,ROL     ;Initialize Timer A Function
;          MOV.B    ROL,@TMA      ;Clear IRRTA
;          BCLR    IRRTA          ;Timer A Interrupt Enable
;          BSET    IENTA
;
;          BCLR    IRRDT          ;Clear IRRDT
;          BSET    IENDT         ;Direct Transfer Interrupt Enable
;
;          MOV.W    #H'BFE8,R0    ;Initialize Function of Subactive Mode 1
;          MOV.B    ROH,@SYSCR1  ;Initialize Function of Subactive Mode 2
;          MOV.B    ROL,@SYSCR2
;
;          BSET    P73           ;Initialize P73
;          BSET    PCR73        ;Initialize P73 Output Port
;
;          MOV.B    #H'3C,R1L    ;Initialize 8bit Timer A Interrupt Counter

```

プログラムリスト

```

;
;      MOV.B      #H'01,R0L
;      MOV.B      R0L,@USRF      ;Initialize LDONF & ITCTF & ENDF
;
;      ANDC       #H'7F,CCR      ;Interrupt Enable
;
;      SLEEP      ;Transition to Subactive Mode
;
;      MOV.W      #H'0FE0,R0
;      MOV.B      R0H,@SYSCR1    ;Initialize Function of Subsleeep Mode 1
;      MOV.B      R0L,@SYSCR2    ;Initialize Function of Subsleeep Mode 2
;
MAIN1  .equ        $
;      SLEEP      ;Transion to Subsleeep Mode
;
;      MOV.B      @USRF,R0L
;      BTST      #2,R0L          ;ENDF = "1" ?
;      BEQ       MAIN1          ;No.
;
;      MOV.W      #H'B7E8,R0
;      MOV.B      R0H,@SYSCR1    ;Initialize Function of Active-H Mode 1
;      MOV.B      R0L,@SYSCR2    ;Initialize Function of Active-H Mode 2
;
;      SLEEP      ;Transition to Active (High-Speed) Mode
;
MAIN9  .equ        $
;      BRA       MAIN9
;
;*****
;
;      Timer A Interrupt
;*****
;
TAINT  .equ        $
;      BCLR      IRRTA          ;Clear IRRTA
;      PUSH     R0              ;Store R0
;
;      MOV.B      @USRF,R0H
;      BTST      #0,R0H          ;LDONF = "1" ?
;      BNE      TAINT1          ;Yes.
;
;      BSET      P73            ;Turn On LED
;      BSET      #0,R0H          ;Set LDONF
;      BRA      TAINT2
;
TAINT1 .equ        $
;      BCLR      P73            ;Turn Off LED
;      BCLR      #0,R0H          ;Clear LDONF
;
TAINT2 .equ        $
;      BTST      #1,R0H          ;IACTF = "1" ?
;      BEQ      TAINT3          ;No.
;
;      DEC      R1L             ;Decrement 8bit Timer A Interrupt Counter
;      BNE      TAINT3          ;8bit Timer A Interrupt Counter = H'00 ? Yes.
;
;      BCLR      IENTA          ;Timer A Interrupt Disable
;      BSET      #2,R0H          ;Set ENDF
;
TAINT3 .equ        $
;      BNOT     #1,R0H          ;Invert IACTF

```

プログラムリスト

```
;  
    MOV.B    ROH,@USRF    ;Save LDONF & IACNF  
    POP     RO            ;Restore RO  
;  
    RTE  
;  
;*****  
; Direct Transfer Interrupt  
;*****  
;  
DTINT    .equ    $  
    BCLR    IRRDT    ;Clear IRRDT  
;  
    RTE  
;  
    .end
```

2.31 サブアクティブモードへの遷移

サブアクティブモードへの遷移	使用機能	低消費電力モード：サブアクティブモード
仕様		
<p>(1) サブアクティブモードへの遷移を行ないます。</p> <p>(2) アクティブ（高速）モードで、SLEEP命令を実行することによりウォッチモードに遷移します。</p> <p>(3) ウォッチモードで、TRQ₀端子に接続したスイッチ入力が入オンになるとIRQ₀割込みが発生し、ウォッチモードは解除され、サブアクティブモードに遷移します。</p> <p>(4) サブアクティブモードでは、0.5secごとにタイマA割込み要求が発生し、タイマA割込み処理の中でLEDの制御を行ないます。LEDは0.5secごとに点灯、消灯を繰り返します。</p> <p>(5) サブアクティブモードで、INT₀端子に接続したスイッチ入力が入オンになるとINT₀割込みが発生し、INT₀割込み終了後に、SLEEP命令を実行してアクティブ（高速）モードに直接遷移します。</p> <p>(6) 図1にTRQ₀、INT₀端子に接続するスイッチの接続例を示します。</p> <div data-bbox="534 537 1061 884" style="text-align: center;"> </div> <p style="text-align: center;">図1 サブアクティブモードへの遷移におけるスイッチ接続例</p>		
使用機能説明		
<p>(1) 本タスク例では、低消費電力モードのサブアクティブモードへの遷移を行ないます。図2にサブアクティブモードへのモード遷移図を示します。以下にサブアクティブモードの機能の説明を示します。</p> <ul style="list-style-type: none"> ・ウォッチモードで割込み（タイマA、IRQ₀）が発生したとき、SYSCR1のLSONが"1"ならば、サブアクティブモードに遷移します。また、サブスリープモードで割込み（タイマA、IRQ₃～IRQ₀、INT₇～INT₀）が発生したとき、サブアクティブモードに遷移します。 ・CCRのIビットが"1"の場合、または割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、サブアクティブモードに遷移しません。 ・サブアクティブモードの解除は、SLEEP命令またはRES端子入力により行われます。 ・SLEEP命令による解除は、SYSCR1のSSBYが"1"、TMAのTMA3が"1"の状態で行うと、サブアクティブモードは解除され、ウォッチモードに遷移します。 ・SYSCR1のSSBYが"0"、LSONが"1"、TMAのTMA3が"1"の状態で行うと、サブスリープモードに遷移します。 ・直接遷移によってアクティブ（高速）モードへ遷移します。 ・RES端子による解除は、RES端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、RES端子を"High"レベルにすると、CPUはリセット例外処理を開始します。なお、システムクロックの発振開始と同時にLSI全体にシステムクロックが供給されます。 ・RES端子は、必ずシステムクロックの発振が安定するまで、"Low"レベルを保持してください。 ・サブアクティブモード解除後の発振安定時間は、SYSCR1のSTS2～STS0により設定します。発振安定時間は、10ms以上となるように設定しなければなりません。 ・本タスク例では、動作周波数に5MHzを使用しているため、発振安定時間を13.1072msに設定しています。 ・CPUがプログラムを実行している動作モードにはアクティブ（高速）モード、アクティブ（中速）モード、サブアクティブモードの3つのモードがあります。この3つの動作モードの間で、プログラムを停止する事なく遷移することを直接遷移と呼びます。直接遷移はSYSCR2のDTONを"1"にセットし、SLEEP命令を実行することにより可能です。遷移後は直接遷移割込み例外処理を開始します。なお、割込み許可レジスタ2により直接遷移割込みが禁止されている場合は、スリープモードまたはウォッチモードへ遷移します。また、CCRのIビットを"1"の状態で行うとスリープモードまたはウォッチモードに遷移し、遷移後のモードから割込みによる解除は不可能となります。 ・サブアクティブモードからアクティブ（高速）モードへの直接遷移は、サブアクティブモードでSYSCR1のSSBYを"1"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"、TMAのTMA3を"1"にセットした状態でSLEEP命令を実行すると、ウォッチモードを経由し、SYSCR1のSTS2～STS0により設定された時間を経過した後、直接、アクティブ（高速）モードに遷移します。 		

使用機能説明

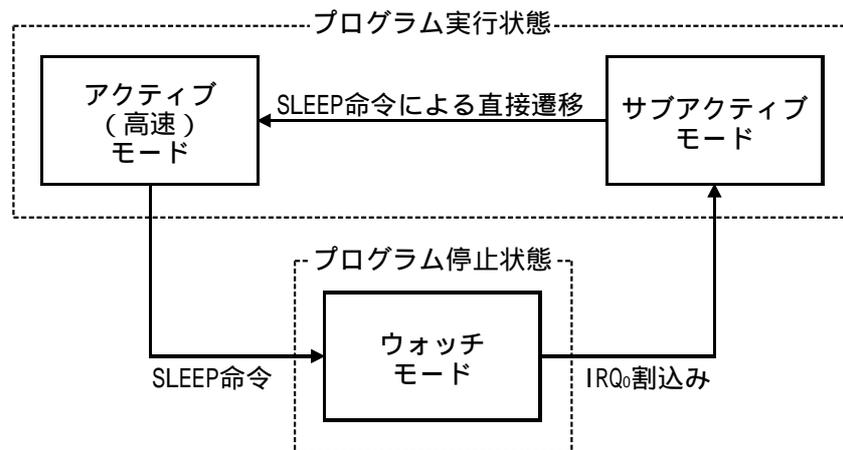


図2 サブアクティブモードへの遷移におけるモード遷移図

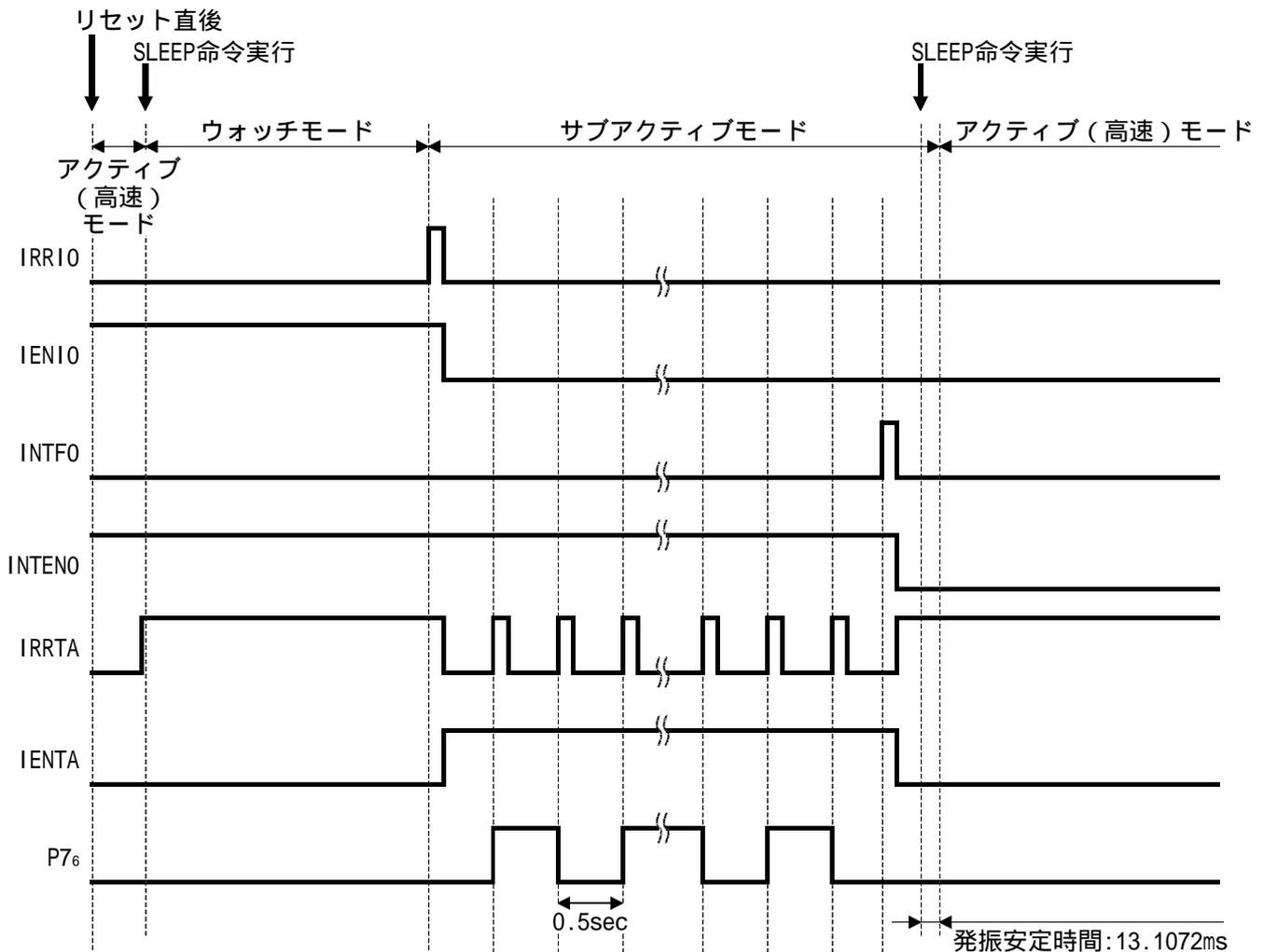
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、サブアクティブモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
$\overline{\text{TRQ}}_0$	スイッチ1入力
IRR10	タイマA割込み要求の有無を反映
IEN10	タイマA割込み要求の許可、または禁止を設定
$\overline{\text{TNT}}_0$	スイッチ2入力

動作原理

(1) 図3に動作原理を示します。図3に示すようにハードウェア処理、およびソフトウェア処理によりサブアクティブモードへの遷移を行ないます。



ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a) IRQ ₀ 割込みの設定
(b) IRQ ₀ 割込み要求の許可
(c) INT ₀ 割込みの設定
(d) INT ₀ 割込み要求の禁止
(e) タイマAの設定
(f) タイマA割込み要求の禁止
(g) ポート7 ₃ の設定
(h) 割込みの許可

ハードウェア処理
(a) IRR10を"1"にセット
(b) サブアクティブモードへ遷移
ソフトウェア処理
(a) IRQ ₀ 割込み処理の起動
(b) IRR10を"0"にクリア
(c) IRQ ₀ 割込み要求の禁止
(d) TCAをリセット
(e) タイマA割込み要求の許可
(f) INT ₀ 割込み要求の許可

ハードウェア処理
(a) IRRTAを"1"にセット
ソフトウェア処理
(a) タイマA割込み処理の起動
(b) IRRTAを"0"にクリア
(c) IEを"1"にセットし、割込み要求を許可
(c) P7 ₃ を反転し、LEDを点灯または消灯

ハードウェア処理
(a) INTF0を"1"にセット
ソフトウェア処理
(a) INT ₀ 割込み処理の起動
(b) INTF0を"0"にクリア
(c) タイマA割込み要求の禁止
(d) INT ₀ 割込み要求の禁止

図3 サブアクティブモードへの遷移の動作原理

サブアクティブモードへの遷移	使用機能	低消費電力モード：サブアクティブモード
----------------	------	---------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、IRQ ₀ 割込みの設定、INT ₀ 割込みの設定、タイマA割込みの設定、ポート7の設定、割込みの許可、ウォッチモードへの遷移、アクティブ(高速)モードへの遷移を行なう
スイッチ1オン	IQINT	IRQ ₀ 割込み処理ルーチンで、IRQ ₀ 割込み要求の禁止を行なう
スイッチ2オン	ITINT	INT ₀ 割込み処理ルーチンで、SWONFのセット、タイマA割込み要求の禁止、INT ₀ 割込み要求の禁止を行なう
LED制御	TAINT	タイマA割込み処理ルーチンで、割込み要求の許可、LEDの制御を行なう
直接遷移	DTINT	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMA	タイマモードレジスタA : TMA=H'19のとき、タイマA機能を時計用タイムベース機能に、TCAのオーバフロー周期を0.5secに設定	H'FFB0	H'19
TCA	タイマカウンタA : PSW出力クロックを入力とし、時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ	H'FFB1	H'00
PDR7	P7 ₃ ポートデータレジスタ7(ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	0
PCR7	PCR7 ₃ ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	1
SYSCR1	SSBY システムコントロールレジスタ1(ソフトウェアスタンバイ) : SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモード、あるいはウォッチモードに遷移。サブアクティブモードでSLEEP命令実行後サブスリープモードに遷移	H'FFF0 ビット7	1
	STS2 STS1 STS0 システムコントロールレジスタ1 (スタンバイタイムセレクト2、1、0) : STS2="0", STS1="1", STS0="1"のとき、スタンバイモード、ウォッチモード解除後の発振安定時間を13.1072msに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="0" STS1="1" STS0="1"
	LSON システムコントロールレジスタ1(ロースピードオンフラグ) : LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定 : LSON="1"のとき、ウォッチモードを解除時、CPUの動作クロックをサブシステムクロックに設定	H'FFF0 ビット3	1
IEGR1	IEG0 割込みエッジセレクトレジスタ1(IRQ ₀ エッジセレクト) : IEG0="1"のとき、IRQ ₀ 端子入力の立上りエッジを検出	H'FFF2 ビット0	1
IEGR2	INTEG0 割込みエッジセレクトレジスタ2(INT ₀ エッジセレクト) : INTEG0="1"のとき、INT ₀ 端子入力の立上りエッジを検出	H'FFF3 ビット0	1

ソフトウェア説明

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
SYSCR2	DTON	H'FFF1 ビット3	0
	MSON	H'FFF1 ビット2	0
	SA1 SA0	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"
IENR1	IENTA	H'FFF4 ビット6	0
	IENO	H'FFF4 ビット0	1
IENR2	IENDT	H'FFF5 ビット7	1
IENR3	INTENO	H'FFF6 ビット0	0
IRR1	IRRTA	H'FFF7 ビット6	0
	IRRI0	H'FFF7 ビット0	0
IRR2	IRRDT	H'FFF8 ビット7	0
IRR3	IRRI0	H'FFF9 ビット0	0

(4) 使用RAM説明

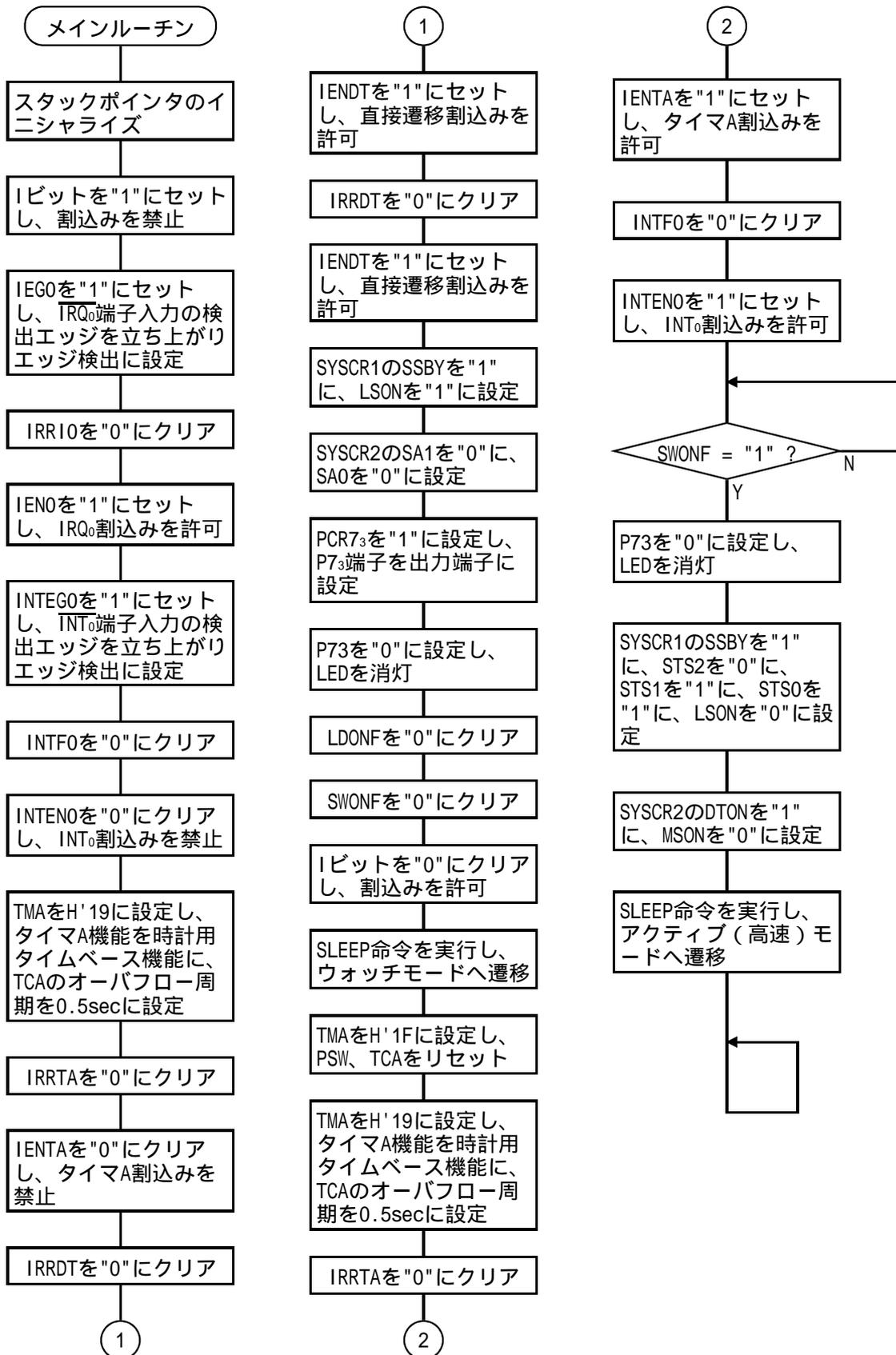
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

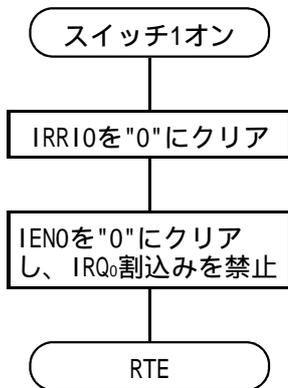
ラベル名	機能	アドレス	使用モジュール名
USRF	LDONF	H'FB80 ビット0	LED制御
	SWONF	H'FB80 ビット1	メインルーチン スイッチ2オン

フローチャート

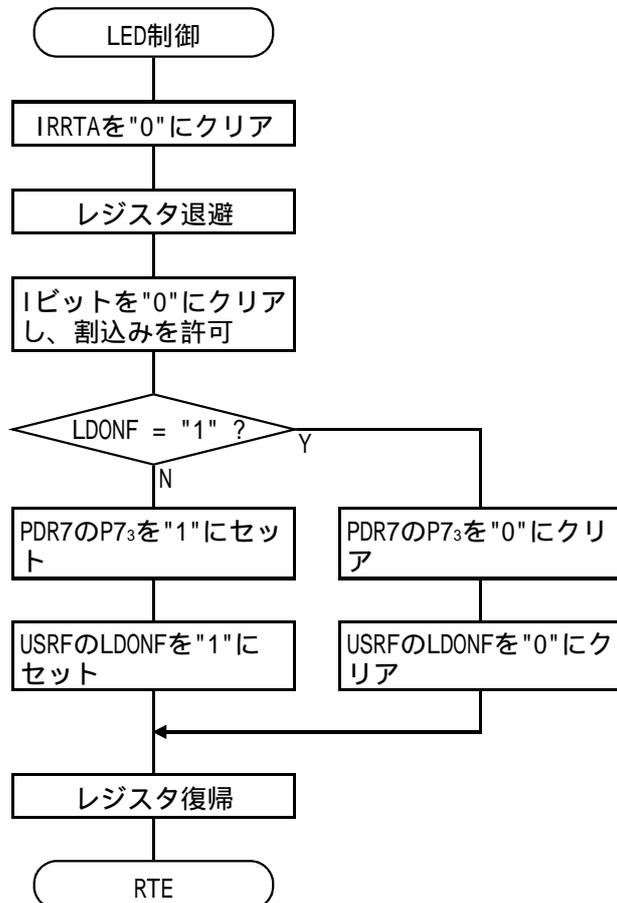
(a) メインルーチン



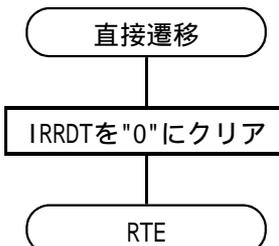
フローチャート

(b) IRQ₀割込み処理ルーチン(c) INT₀割込み処理ルーチン

(d) タイマA割込み処理ルーチン



(e) 直接遷移割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series -H8/3644,H8/3657-
;           Application Note
;
;           'Transition to Subactive Mode'
;
;           Function
;           : Power-Down Mode
;           Subactive Mode
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
;
*****
;
;           .cpu          300L
;
*****
;
;           Symbol Definition
;
*****
TMA          .equ          H'FFB0          ;Timer Mode Register A
TCA          .equ          H'FFB1          ;Timer Counter A
PDR7         .equ          H'FFDA          ;Port Data Register 7
P73         .bequ         3,PDR7          ;Port Data Register 73
PCR7         .equ          H'FFEA          ;Port Control Register 7
PCR73        .bequ         3,PCR7         ;Port Control Register 73
SYSCR1       .equ          H'FFF0          ;System Control Register 1
SSBY        .bequ         7,SYSCR1        ;Software Standby
STS2         .bequ         6,SYSCR1        ;Standby Timer Select 2
STS1         .bequ         5,SYSCR1        ;Standby Timer Select 1
STS0         .bequ         4,SYSCR1        ;Standby Timer Select 0
LSON         .bequ         3,SYSCR1        ;Low Speed On Flag
MA1          .bequ         1,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 1
MA0          .bequ         0,SYSCR1        ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2       .equ          H'FFF1          ;System Control Register 2
NESEL        .bequ         4,SYSCR2        ;Noise Elimination Sampling Frequency Select
DTON         .bequ         3,SYSCR2        ;Direct Transfer On Flag
MSON         .bequ         2,SYSCR2        ;Middle Speed On Flag
SA1          .bequ         1,SYSCR2        ;Subactive Mode Clock Select 1
SA0          .bequ         0,SYSCR2        ;Subactive Mode Clock Select 0
IEGR1        .equ          H'FFF2          ;Interrupt Edge Select Register 1
IEG0         .bequ         0,IEGR1         ;IRQ0 Edge Select
IEGR2        .equ          H'FFF3          ;Interrupt Edge Select Register 2
INTEG0       .bequ         0,IEGR2         ;INT0 Edge Select
IENR1        .equ          H'FFF4          ;Interrupt Enable Register 1
IENTA        .bequ         6,IENR1         ;Timer A Interrupt Enable
IENO         .bequ         0,IENR1         ;IRQ0 Interrupt Enable
IENR2        .equ          H'FFF5          ;Interrupt Enable Register 2
IENDT        .bequ         7,IENR2         ;Direct Transfer Interrupt Enable
IENR3        .equ          H'FFF6          ;Interrupt Enable Register 3
INTENO       .bequ         0,IENR3         ;INT0 Interrupt Enable
IRR1         .equ          H'FFF7          ;Interrupt Request Register 1
IRR1A        .bequ         6,IRR1          ;Timer A Interrupt Request Flag
IRR10        .bequ         0,IRR1          ;IRQ0 Interrupt Request Flag
IRR2         .equ          H'FFF8          ;Interrupt Request Register 2

```

プログラムリスト

```

IRRDT      .bequ      7,IRR2      ;Direct Interrupt Request Flag
IRR3       .equ       H'FFF9     ;Interrupt Request Register 3
INTFO      .bequ      0,IRR3     ;INT0 Interrupt Request Flag
;
;
;*****
;
;          Ram Allocation
;*****
;
;
STACK      .equ       H'FF80     ;Stack Pointer
USRF       .equ       H'FB80     ;User Flag Area
LDONF      .bequ      0,USRF     ;LED On Flag
SWONF      .bequ      1,USRF     ;Switch On Flag
;
;
;*****
;
;          Vector Address
;*****
;
;
;          .org      H'0000
;          .data.w   MAIN        ;Reset Interrupt
;
;
;          .org      H'0008
;          .data.w   IQINT       ;IRQ0 Interrupt
;          .data.w   MAIN        ;IRQ1 Interrupt
;          .data.w   MAIN        ;IRQ2 Interrupt
;          .data.w   MAIN        ;IRQ3 Interrupt
;          .data.w   ITINT       ;INT0 - INT7 Interrupt
;
;
;          .org      H'0014
;          .data.w   TAINT       ;Timer A Interrupt
;          .data.w   MAIN        ;Timer B1 Interrupt
;
;
;          .org      H'0020
;          .data.w   MAIN        ;Timer X Interrupt
;          .data.w   MAIN        ;Timer V Interrupt
;
;
;          .org      H'0026
;          .data.w   MAIN        ;Sci1 Interrupt
;
;
;          .org      H'002A
;          .data.w   MAIN        ;Sci3 Interrupt
;          .data.w   MAIN        ;A/D Converter Interrupt
;          .data.w   DTINT       ;Sleep Interrupt
;
;
;*****
;
;          Main Program
;*****
;
;
;          .org      H'1000
;
;MAIN      .equ       $
;          MOV.W     #STACK,SP   ;Initialize Stack Pointer
;          ORC       #H'80,CCR    ;Interrupt Disable
;
;
;          BSET      IEG0        ;Set Rising Edge of IRQ0 Terminal Input
;          BCLR      IRR10       ;Clear IRR10
;          BSET      IENO        ;IRQ0 Interrupt Enable
;
;
;          BSET      INTEGO      ;Set Rising Edge of INT0 Terminal Input
;          BCLR      INTFO       ;Clear INTFO
;          BCLR      INTENO      ;INT0 Interrupt Disable

```

プログラムリスト

```

;
;
MOV.B      #H'19,R0L
MOV.B      R0L,@TMA      ;Set TMA3
BCLR      IRRTA      ;Clear IRRTA
BCLR      IENTA      ;Timer A Interrupt Disable
;
;
BCLR      IRRDT      ;Clear IRRTA
BSET      IENDT      ;Direct Transfer Interrupt Enable
;
;
MOV.W      #H'BFE0,R0
MOV.B      R0H,@SYSCR1  ;Initialize Function of Watch Mode 1
MOV.B      R0L,@SYSCR2  ;Initiakize Function of Watch Mode 2
;
;
BCLR      P73      ;Initialize P73
BSET      PCR73     ;Initialize P73 Output Port
;
;
MOV.B      #H'00,R0L
MOV.B      R0L,@USRF   ;Initialize LDONF & SWONF
;
;
ANDC      #H'7F,CCR   ;Interrupt Enable
;
;
SLEEP      ;Transition to Watch Mode
;
;
MOV.W      #H'1F19,R0
MOV.B      R0H,@TMA    ;Reset PSW & TCA
MOV.B      R0L,@TMA    ;Initialize Timer A Function
BCLR      IRRTA      ;Clear IRRTA
BSET      IENTA      ;Timer A Interrupt Enable
;
;
BCLR      INTF0      ;Clear INTF0
BSET      INTENO     ;INT0 Interrupt Enable
;
MAIN1      .equ      $
MOV.B      @USRF,R0L
BTST      #1,R0L      ;SWONF = "1" ?
BEQ       MAIN1      ;No.
;
;
BCLR      P73      ;Turn off LED
;
;
MOV.W      #H'B7E8,R0
MOV.B      R0H,@SYSCR1 ;Initialize Fubction of Active-H Mode 1
MOV.B      R0L,@SYSCR2 ;Initialize Function of Active-H Mode 2
;
;
SLEEP      ;Transition to Active (High-Speed) Mode
;
MAIN9      .equ      $
BRA       MAIN9
;
;
*****
;
;
IRQ0 Interrupt
;
*****
;
;
IQINT      .equ      $
BCLR      IRRIO      ;Clear IRRIO
;
;
BCLR      IENO      ;IRQ0 Interrupt Disable
;
;
RTE
;
;
*****
;
;

```

プログラムリスト

```

;
;          INTO Interrupt
;*****
;
;
ITINT      .equ          $
           BTST          INTFO          ;INTFO = "1" ?
           BEQ           ITINT9        ;No.
;
           BCLR          INTFO          ;Clear INTFO
           PUSH          R0             ;Store R0
;
           MOV.B         @USRF,R0L
           BSET          #1,R0L        ;Set SWONF
           MOV.B         R0L,@USRF
;
           BCLR          IENTA         ;Timer A Interrupt Disable
           BCLR          INTENO        ;INTO Interrupt Disable
;
           POP           R0            ;Restore R0
;
ITINT9     RTE
;
;*****
;
;          Timer A Interrupt
;*****
;
;
TAINT      .equ          $
           BCLR          IRRTA         ;Clear IRRTA
           PUSH          R0            ;Store R0
;
           ANDC          #'7F,CCR      ;Interrupt Enable
;
           MOV.B         @USRF,R0L
           BTST          #0,R0L        ;LDONF = "1" ?
           BNE          TAINT1         ;Yes.
;
           BSET          P73           ;Turn on LED
           BSET          #0,R0L        ;Set LDONF
           BRA           TAINT2
;
TAINT1     .equ          $
           BCLR          P73           ;Turn off LED
           BCLR          #0,R0L        ;Clear LDONF
;
TAINT2     .equ          $
           MOV.B         R0L,@USRF
           POP           R0            ;Restore R0
;
           RTE
;
;*****
;
;          Direct Transfer Interrupt
;*****
;
;
DTINT      .equ          $
           BCLR          IRRDT         ;Clear IRRDT
;
           RTE
;
           .end

```

2.32 アクティブ（中速）モードへの遷移

アクティブ（中速）モードへの遷移	使用機能	低消費電力モード：アクティブ（中速）モード
<p data-bbox="188 181 245 215">仕様</p> <p data-bbox="180 266 1426 528"> (1) アクティブ（中速）モードへの遷移を行いません。 (2) アクティブ（高速）モードで、SLEEP命令を実行することによりウォッチモードに遷移します。 (3) ウォッチモードで、タイマA割込み要求が発生することにより、アクティブ（中速）モードに遷移します。 (4) タイマA割込み要求は、時計用タイムベース機能により0.5secごとに発生します。タイマA割込み処理の中で、LEDの点灯または消灯を行ない、タイマA割込み処理の終了後、再びSLEEP命令の実行によりウォッチモードに遷移します。LEDは0.5secごとに交互に点灯/消灯します。 (5) タイマA割込みが60回発生し、30sec経過するとアクティブ（中速）モードからアクティブ（高速）モードに直接遷移して終了します。 </p>		
<p data-bbox="137 573 300 607">使用機能説明</p> <p data-bbox="180 663 1426 1800"> (1) 本タスク例では、低消費電力モードのアクティブ（中速）モードへの遷移を行いません。図1にアクティブ（中速）モードへのモード遷移図を示します。以下にアクティブ（中速）モードの機能の説明を示します。 <ul style="list-style-type: none"> ・スタンバイモードで割込み（IRQ₁、IRQ₀）が発生したとき、ウォッチモードで割込み（タイマA、IRQ₀）が発生したとき、あるいはスリープ（中速）モードですべての割込みが発生したとき、SYSCR1のLSONが"0"かつSYSCR2のMSONが"1"ならば、アクティブ（中速）モードに遷移します。 ・CCRのIビットが"1"の場合、または割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、アクティブ（中速）モードに遷移しません。 ・アクティブ（中速）モードの解除は、SLEEP命令またはRES端子入力により行われます。 ・SLEEP命令による解除は、SYSCR1のSSBYが"1"、LSONが"0"、TMAのTMA3が"0"の状態ですLEEP命令を実行すると、スタンバイモードに遷移します。SYSCR1のSSBYが"1"、TMAのTMA3が"1"の状態ですLEEP命令を実行すると、ウォッチモードに遷移します。SYSCR1のSSBYが"0"、LSONが"0"、の状態です、SLEEP命令を実行すると、SYSCR2のMSONが"0"のときスリープ（高速）モードに、MSONが"1"のときスリープ（中速）モードに遷移します。直接遷移によってアクティブ（高速）モード、またはサブアクティブモードへ遷移します。 ・RES端子による解除は、RES端子を"Low"レベルにすると、リセット状態に遷移し、アクティブ（中速）モードは解除されます。 ・アクティブ（中速）モードの動作周波数は、SYSCR1のMA1、MA0で設定した周波数のクロックによって動作します。 ・CPUがプログラムを実行している動作モードにはアクティブ（高速）モード、アクティブ（中速）モード、サブアクティブモードの3つのモードがあります。この3つの動作モードの間で、プログラムを停止する事なく遷移することを直接遷移と呼びます。直接遷移はSYSCR2のDTONを"1"にセットし、SLEEP命令を実行することにより可能です。遷移後は直接遷移割込み例外処理を開始します。なお、割込み許可レジスタ2により直接遷移割り込みが禁止されている場合は、スリープモードDまたはウォッチモードへ遷移します。また、CCRのIビットを"1"の状態です直接遷移を行なうとスリープモードまたはウォッチモードに遷移し、遷移後のモードから割込みによる解除は不可能となります。 ・アクティブ（中速）モードからアクティブ（高速）モードへの直接遷移は、アクティブ（中速）モードでSYSCR1のSSBYを"0"、LSONを"0"、SYSCR2のMSONを"0"、DTONを"1"にセットした状態でSLEEP命令を実行すると、スリープモードを経由してアクティブ（高速）モードに遷移します。 ・タイマA割込みにより、ウォッチモードを解除しアクティブ（中速）モードに遷移する場合に、クロックが安定するまでCPUと周辺機器が待機する時間をSYSCR1のSTS2～STS0により指定します。動作周波数に応じて待機時間が10ms以上になるように指定しなければなりません。 ・本タスク例では、動作周波数に5MHzを使用しているため、STS2を"0"、STS1を"1"、STS0を"1"に設定し、待機時間を65,536ステート、13.1072msに設定しています。 ・アクティブ（中速）モードの動作クロックはSYSCR1のMA1、MA0により設定します。本タスク例では、アクティブ（中速）モードの動作クロックを $\frac{osc}{64}$、78.125kHzに設定しています。 </p>		

使用機能説明

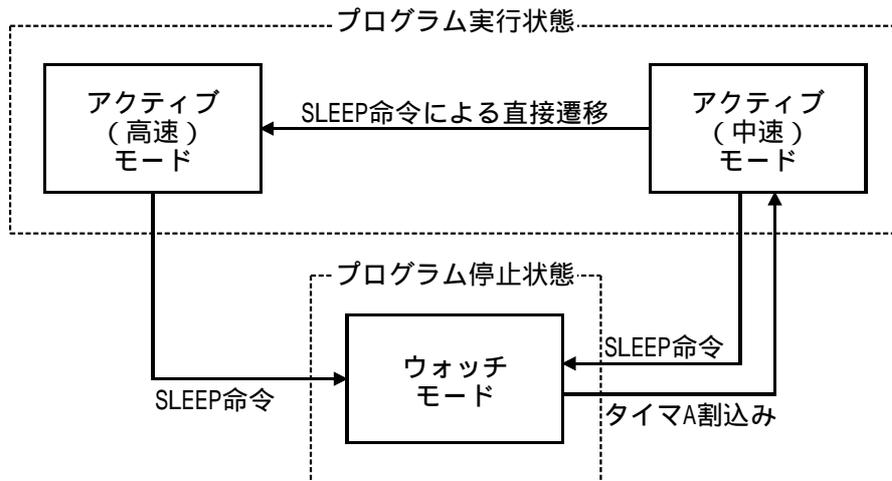


図1 アクティブ（中速）モードへの遷移におけるモード遷移図

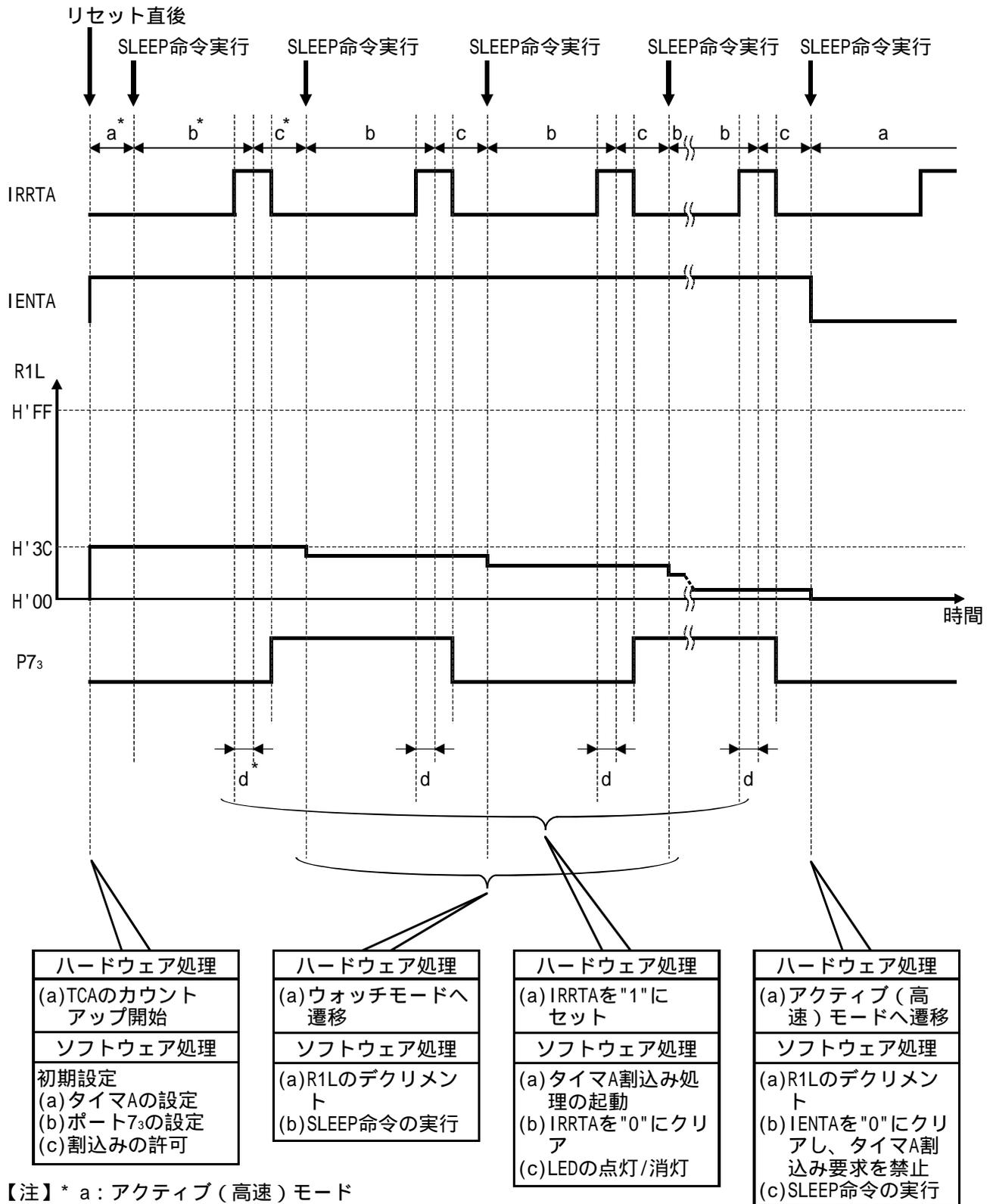
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、アクティブ（中速）モードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 ₃ 出力端子機能の設定
PDR7	P7 ₃ 出力端子のデータの格納
P7 ₃	LED出力
TMA	タイマA時計用タイムベース機能、およびTCAオーバフロー周期の設定
TCA	時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ
IRRTA	タイマA割込み要求の有無を反映
IENTA	タイマA割込み要求の許可、または禁止を設定

動作原理

(1) 図2に動作原理を示します。図2に示すようなハードウェア処理、およびソフトウェア処理によりアクティブ（中速）モードへの遷移を行ないます。



【注】* a：アクティブ（高速）モード
 b：ウォッチモード
 c：アクティブ（中速）モード
 d：発振安定時間（13.1072ms）

図3 アクティブ（中速）モードへの遷移の動作原理

アクティブ(中速)モードへの遷移	使用機能	低消費電力モード：アクティブ(中速)モード
------------------	------	-----------------------

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	MAIN	スタックポインタのイニシャライズ、タイマA割込みの設定、ポート7の設定、割込みの許可、ウォッチモードへの遷移、8ビットタイマカウンタのデクリメント、タイマA割込みの禁止を行なう
LED制御	TAINT	タイマA割込み処理ルーチンで、LEDの制御を行なう
直接遷移	DTINT	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TMA	タイマモードレジスタA : TMA=H'19のとき、タイマA機能を時計用タイムベース機能に、TCAのオーバフロー周期を0.5secに設定	H'FFB0	H'19	
TCA	タイマカウンタA : PSW出力クロックを入力とし、時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ	H'FFB1	H'00	
PDR7	P7 ₃	ポートデータレジスタ7 (ポートデータレジスタ7 ₃) : P7 ₃ =0のとき、P7 ₃ 端子の出力レベルは"Low" : P7 ₃ =1のとき、P7 ₃ 端子の出力レベルは"High"	H'FFDA ビット3	1
PCR7	PCR7 ₃	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 ₃) : PCR7 ₃ =1のとき、P7 ₃ 端子を出力端子に設定	H'FFEA ビット3	0
SYSCR1	SSBY	システムコントロールレジスタ1 (ソフトウェアスタンバイ) : SSBY="0"のとき、アクティブモードでSLEEP命令実行後スリープモードに遷移。サブアクティブモードでSLEEP命令実行後サブスリープモードに遷移 : SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモード、あるいはウォッチモードに遷移。サブアクティブモードでSLEEP命令実行後サブスリープモードに遷移	H'FFF0 ビット7	1
	STS2 STS1 STS0	システムコントロールレジスタ1 (スタンバイタイムセレクト2、1、0) : STS2="0", STS1="1", STS0="1"のとき、スタンバイモード、ウォッチモード解除後の発振安定時間を13.1072msに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="0" STS1="1" STS0="1"
	LSON	システムコントロールレジスタ1 (ロースピードオンフラグ) : LSON="0"のとき、ウォッチモードを解除時、CPUの動作クロックをシステムクロックに設定	H'FFF0 ビット3	0
	MA1 MA0	システムコントロールレジスタ1 (アクティブ(中速)モードクロックセレクト1、0) : MA1="1", MA0="0"のとき、アクティブ(中速)モードの動作クロックを $\frac{osc}{64}$ に設定	H'FFF0 ビット1 ビット0	MA1="1" MA0="0"

ソフトウェア説明

表3 使用内部レジスタ説明

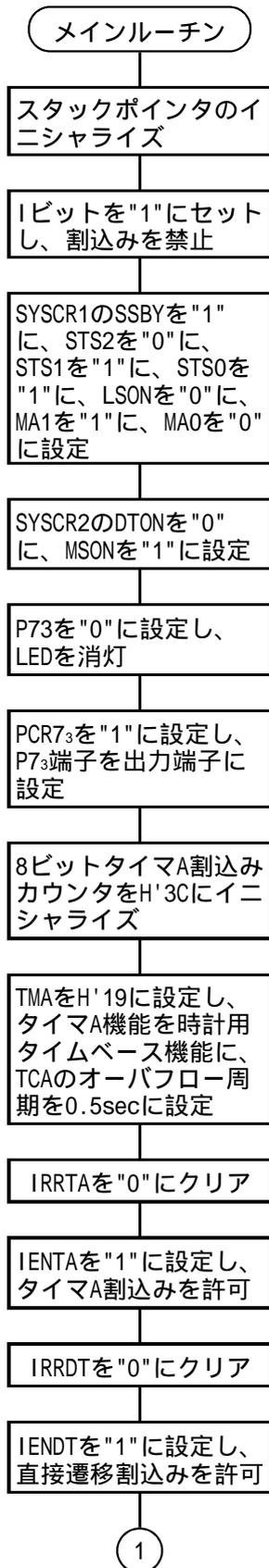
レジスタ名		機能	アドレス	設定値
SYSCR2	DTON	システムコントロールレジスタ2 （ダイレクトトランスファオンフラグ） ：DTON="0"のとき、アクティブモードでSLEEP命令を実行したとき、スタンバイモード、ウォッチモード、またはスリープモードに遷移。サブアクティブモードでSLEEP命令を実行したとき、ウォッチモード、またはサブスリープモードに遷移 ：DTON="1"のとき、アクティブ（中速）モードでSLEEP命令を実行したとき、アクティブ（高速）モード（SSBY="0"、MSON="0"、LSON="0"のとき）、またはサブアクティブモード（SSBY="1"、TMA3="1"、LSON="1"のとき）に直接遷移	H'FFF1 ビット3	0
	MSON	システムコントロールレジスタ2 （ミドルスピードオンフラグ） ：MSON="1"のとき、スタンバイモード、ウォッチモード、スリープモード解除後、アクティブ（中速）モードで動作。アクティブモードでSLEEP命令を実行したとき、スリープ（中速）モードで動作	H'FFF1 ビット2	0
IENR1	IENTA	割込み許可レジスタ1（タイマA割込みイネーブル） ：IENTA="0"のとき、タイマA割込み要求を禁止 ：IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IENDT	IENDT	割込み許可レジスタ2（直接遷移割込みイネーブル） ：IENDT="0"のとき、直接遷移割込み要求を禁止 ：IENDT="1"のとき、直接遷移割込み要求を許可	H'FFF5 ビット7	1
IRR1	IRRRTA	割込み要求レジスタ1（タイマA割込み要求フラグ） ：IRRRTA="0"のとき、タイマA割込みが要求されていない ：IRRRTA="1"のとき、タイマA割込みが要求されている	H'FFF7 ビット6	0

(4) 使用RAM説明

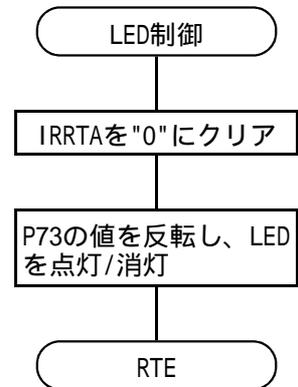
本タスク例ではRAMは使用しません。

フローチャート

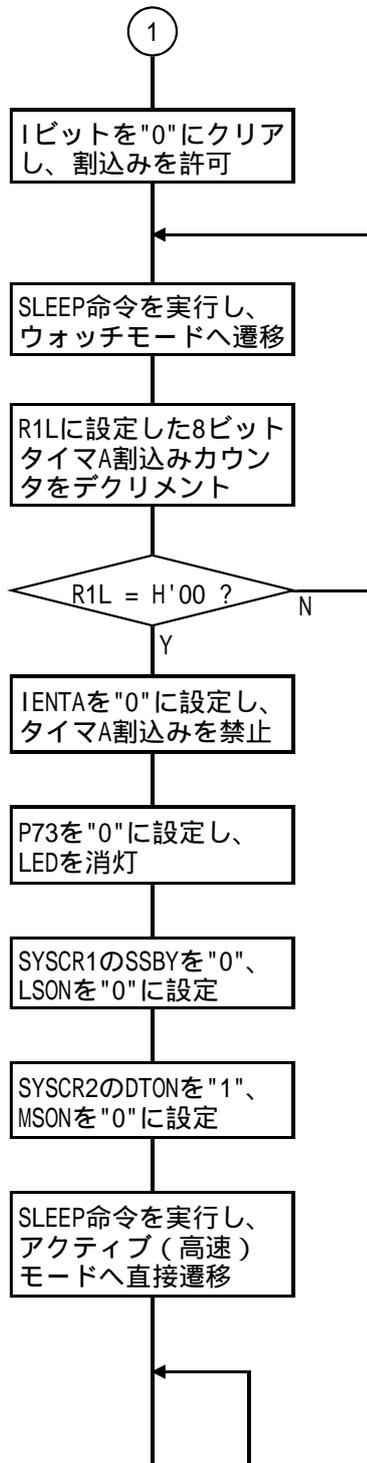
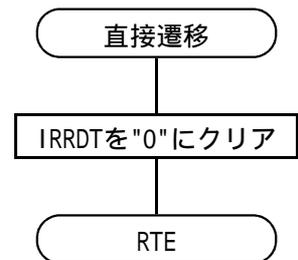
(a) メインルーチン



(b) タイマA割込み処理ルーチン



(c) 直接遷移割込み処理ルーチン



プログラムリスト

```

*****
;
;
;           H8/300L Series H8/3644,H8/3657-
;           Application Note
;
;           'Transition to Active (Middle-Speed) Mode'
;
;           Function
;           : Power-Down Mode
;           : Active (Middle-Speed) Mode
;
;           External Clock : 10MHz
;           Internal Clock : 5MHz
;           Sub Clock      : 32.768kHz
;
*****
;
*****
;
;           .cpu          300L
;
*****
;
;           Symbol Definition
;
*****
;
TMA          .equ          H'FFB0          ;Timer Mode Register A
TCA          .equ          H'FFB1          ;Timer Counter A
PDR7         .equ          H'FFDA          ;Port Data Register 7
P73         .bequ         3,PDR7          ;Port Data Register 73
PCR7         .equ          H'FFEA          ;Port Control Register 7
PCR73        .bequ         3,PCR7         ;Port Control Register 73
SYSCR1       .equ          H'FFF0          ;System Control Register 1
SSBY         .bequ         7,SYSCR1       ;Software Standby
STS2         .bequ         6,SYSCR1       ;Standby Timer Select 2
STS1         .bequ         5,SYSCR1       ;Standby Timer Select 1
STS0         .bequ         4,SYSCR1       ;Standby Timer Select 0
LSON         .bequ         3,SYSCR1       ;Low Speed On Flag
MA1          .bequ         1,SYSCR1       ;Active (Middle-Speed) Mode Clock Select 1
MA0          .bequ         0,SYSCR1       ;Active (Middle-Speed) Mode Clock Select 0
SYSCR2       .equ          H'FFF1          ;System Control Register 2
NESEL        .bequ         4,SYSCR2       ;Noise Elimination Sampling Frequency Select
DTON         .bequ         3,SYSCR2       ;Direct Transfer On Flag
MSON         .bequ         2,SYSCR2       ;Middle Speed On Flag
SA1          .bequ         1,SYSCR2       ;Subactive Mode Clock Select 1
SA0          .bequ         0,SYSCR2       ;Subactive Mode Clock Select 0
IENR1        .equ          H'FFF4          ;Interrupt Enable Register 1
IENTA        .bequ         6,IENR1        ;Timer A Interrupt Enable
IENR2        .equ          H'FFF5          ;Interrupt Enable Register 2
IENDT        .bequ         7,IENR2        ;Direct Transfer Interrupt Enable
IRR1         .equ          H'FFF7          ;Interrupt Request Register 1
IRRTA        .bequ         6,IRR1         ;Timer A Interrupt Request Flag
IRR2         .equ          H'FFF8          ;Interrupt Request Register 2
IRRDT        .bequ         7,IRR2         ;Direct Transfer Interrupt Request Register
;
;
*****
;
;           Ram Allocation
;
*****
;
STACK        .equ          H'FF80          ;Stack Pointer
;
;

```

プログラムリスト

```

*****
;
;
;          Vector Address
;
*****
;
;
;          .org          H'0000
;          .data.w      MAIN          ;Reset Interrupt
;
;
;          .org          H'0008
;          .data.w      MAIN          ;IRQ0 Interrupt
;          .data.w      MAIN          ;IRQ1 Interrupt
;          .data.w      MAIN          ;IRQ2 Interrupt
;          .data.w      MAIN          ;IRQ3 Interrupt
;          .data.w      MAIN          ;INT0 - INT7 Interrupt
;
;
;          .org          H'0014
;          .data.w      TAINT         ;Timer A Interrupt
;          .data.w      MAIN          ;Timer B1 Interrupt
;
;
;          .org          H'0020
;          .data.w      MAIN          ;Timer X Interrupt
;          .data.w      MAIN          ;Timer V Interrupt
;
;
;          .org          H'0026
;          .data.w      MAIN          ;Sci1 Interrupt
;
;
;          .org          H'002A
;          .data.w      MAIN          ;Sci3 Interrupt
;          .data.w      MAIN          ;A/D Converter Interrupt
;          .data.w      DTINT         ;Sleep Interrupt
;
*****
;
;
;          Main Program
;
*****
;
;
;          .org          H'1000
;
MAIN      .equ          $
;          MOV.W        #STACK,SP    ;Initialize Stack Pointer
;          ORC          #H'80,CCR     ;Interrupt Disable
;
;          MOV.W        #H'B6E4,R0
;          MOV.B        R0H,@SYSCR1  ;Initialize Function of Watch Mode 1
;          MOV.B        R0L,@SYSCR2  ;Initialize Function of Watch Mode 2
;
;          BCLR         P73          ;Initialize P73
;          BSET         PCR73        ;Initialize P73 Output Port
;
;          MOV.B        #H'3C,R1L    ;Initialize 8bit Timer A Interrupt Counter
;
;          MOV.B        #H'19,R0L
;          MOV.B        R0L,@TMA     ;Initialize Timer A Function
;          BCLR         IRRTA        ;Clear IRRTA
;          BSET         IENTA        ;Timer A Interrupt Enable
;
;          BCLR         IRRDT        ;Clear IRRDT
;          BSET         IENDT        ;Direct Transfer Interrupt Enable
;
;          ANDC         #H'7F,CCR     ;Interrupt Enable
;
MAIN1     .equ          $

```

プログラムリスト

```

;
; SLEEP ;Transition to Watch Mode
;
; DEC R1L ;Decrement 8bit Timer A Counter
; BNE MAIN1 ;8bit Timer A Counter = H'00 ? No.
;
; BCLR IENTA ;Timer A Interrupt Disable
;
; BCLR P73 ;Turn off LED
;
; MOV.W #H'07EC,R0
; MOV.B R0H,@SYSCR1 ;Initialize Function of Acitve-H Mode 1
; MOV.B R0L,@SYSCR2 ;Initialize Function of Active-H Mode 2
;
; SLEEP ;Transition to Active (Middle-Speed) Mode
;
MAIN9 .equ $
      BRA MAIN9
;
; *****
; Timer A Interrupt
; *****
;
; TAINTE .equ $
; BCLR IRRTA ;Clear IRRTA
;
; BNOT P73 ;Turn on or Turn off LED
;
; RTE
;
; *****
; Direct Transfer Interrupt
; *****
;
; DTINTE .equ $
; BCLR IRRDT ;Claer IRRDT
;
; RTE
;
; .end

```

H8/300L シリーズ アプリケーションノート

発行年月 平成 10 年 10 月 第 1 版

発 行 株式会社 日立製作所
半導体事業本部統括営業本部

編 集 株式会社 超Lメディア
技術ドキュメントグループ

©株式会社 日立製作所

H8/300L シリーズ アプリケーションノート



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

ADJ-502-072 (H)