

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い申し上げます。

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。

32172/32173 グループ スタートアッププログラム例

RJJ05B0105-0112Z
Rev.1.12
2003.10.29

1.0 要約

この資料は 32172/32173 グループで使用する汎用スタートアッププログラム例を掲載しています。

2.0 はじめに

この資料で説明する応用例は次のマイコン、条件での利用に適用されます。

- ・マイコン : 32172/32173 グループ
- ・動作周波数 : 20 ~ 40 MHz

3.0 スタートアッププログラムの概要

組み込み用アプリケーションプログラムをターゲットシステム上で動作させるためには、ユーザプログラム、EIT (Exception, Interrupt, Trap) ハンドラ処理とともに、ターゲットシステムの初期設定を行うスタートアッププログラムが必要です。

EIT ハンドラ処理については「4.0 EIT の概要」以降で説明します。

3.1 スタートアッププログラムの作成方法

組み込み用アプリケーションプログラムをターゲットシステム上で動作させるには、ユーザプログラム (main 関数以降) を呼び出す「スタートアッププログラム」が必要です。本参考プログラム例では、以下に示す(1)~(12)の設定または処理を行っています。

- (1) EIT ベクタエントリ
- (2) プロテクト ID の設定
- (3) ICU ベクタテーブル
- (4) SFR 領域の確保
- (5) EIT ハンドラ処理
- (6) スタック領域の確保
- (7) マイクロプロセッサ動作モードの設定
- (8) スタックポインタの設定
- (9) データ領域 (D、B) セクションの設定
- (10) ベースレジスタの設定
- (11) main 関数の呼び出し
- (12) IE ビット操作関数

なお、実際にはアプリケーションプログラムの内容によって、これ以外にも追加すべき処理がある場合、または、この中から削除すべき処理がある場合があります。

C 標準ライブラリを使用する際は、C 標準ライブラリの初期設定、および終了処理を組み込む必要がある場合があります。

詳しくは、「CC32R ユーザーズマニュアル C コンパイラ編」をご参照ください。

3.1.1 EIT ベクタエントリ (5.0 参考プログラム例 10 ~ 63 行目)

EIT ベクタエントリは、内蔵 ROM 領域 / 拡張領域の先頭に置かれ、各 EIT 事象の処理ハンドラ先頭アドレス (分岐先アドレスではないことに注意) への分岐命令を配置します。

EIT 事象に関しては、「4.0 EIT の概要」以降で説明します。

3.1.2 プロテクト ID の設定 (5.0 参考プログラム例 65 ~ 67 行目)

汎用シリアルプログラマ、エミュレータ等の内蔵フラッシュメモリ書き込み / 消去ツール使用時は、ツールで入力した ID と、内蔵フラッシュメモリ内の ID と照合を行います。正しい ID を入力し一致しないと書き込み / 消去を実行できません (一部ツールでは、全領域消去した後に、ツール動作が実行可能になり、内蔵フラッシュメモリの書き込みも可能になります。)

内蔵フラッシュメモリのプロテクトが不要な場合は、内蔵フラッシュメモリのプロテクト用 ID 照合領域 (H'0000 0084 ~ H'0000 0093) すべてに H'FF を設定してください。本参考プログラム例では、プロテクト ID 不要の場合を想定しています。

3.1.3 ICU ベクタテーブル (5.0 参考プログラム例 70 ~ 108 行目)

割り込みコントローラ (ICU) ベクタテーブルに、各内蔵周辺 I/O の割り込みハンドラ先頭アドレスを設定します。割り込み要求受付時には、受け付けた割り込み要求要因に対応する ICU ベクタテーブルのアドレス下位 16 ビットが、割り込みベクタレジスタ (IVECT) にセットされます。EIT ハンドラでは、この IVECT レジスタの内容を「LDH 命令」で読み出すことで、ICU ベクタテーブルのアドレスを取得します。

3.1.4 SFR 領域の確保 (5.0 参考プログラム例 110 ~ 115 行目)

H'0080 0000 番地より内蔵周辺 I/O のレジスタが配置されています。ここでは SFR (Special Function Register) 領域を 16K バイト確保しています。

3.1.5 EI ハンドラ処理 (5.0 参考プログラム例 119 ~ 196 行目)

EIT 事象が発生すると、それまで実行していたプログラムを中断して、ハードウェア前処理 (後述) を行った後 EIT ハンドラ処理に分岐します。

本参考プログラム例の場合、外部割り込み (EI) の EIT ハンドラを実現しています。

詳細を「4.3 割り込みハンドラ処理手順」で説明します。

システムブレーク割り込み、例外、トラップ発生時のハンドラ処理は無限ループ処理としています。

システムブレーク割り込み、例外、トラップハンドラ処理組み込み時には、その処理の先頭アドレス (ラベル名) を EIT ベクタエントリの対応する箇所に記述してください。

3.1.6 スタック領域の確保 (5.0 参考プログラム例 217 ~ 226 行目)

アプリケーションの動作に必要なスタック領域を確保します。一般的には、アセンブリプログラムにおいて、疑似命令 [.SECTION] でスタック領域セクションを宣言し、疑似命令 [.RES] でスタック領域を確保します。

アプリケーションの動作に必要なスタックサイズは、関数の呼び出し関係や、各関数を使用するスタックサイズなどを考慮して求めます。ただし、厳密に算出するのは困難であるため、一般的には、デバッグ行程や評価行程で実際にプログラムを実行してスタックの使用量を調べる方法 (初期段階で十分なスタック領域を確保しておき、デバッガを用いてスタックの使用量を調べる等) を用います。

この他に、スタックサイズ算出ユーティリティ (stk32R) を使用する方法があります。stk32R は、コンパイル時に"-stack"オプションを付加し、出力されたスタック使用量表示ファイルを処理してプログラムの動作に必要なスタックサイズを求めることができます。詳細は「CC32R ユーザーズマニュアル C コンパイラ編」をご参照ください。スタックはワード (4 バイト) 単位で使用するため、領域確保も 4 バイト単位で行います。

本参考プログラム例では、2048 バイトのスタック領域を確保しています。(実際には、プログラムによって、適切なスタック領域を確保する必要があります。)

3.1.7 マイクロプロセッサ動作モードの設定 (5.0 参考プログラム例 233,234,296 ~ 298 行目)

ターゲットのマイクロプロセッサに対して、使用スタックの指定や割り込みの許可 / 禁止の指定を、プロセッサ状態語レジスタ (PSW) により設定します。

PSW は、CPU のステータスを表示するレジスタで、通常使用する PSW フィールドと、EIT 発生時に PSW フィールドを退避するための BPSW フィールドからなります。

本参考プログラム例では、プログラム実行直後に「割り込み禁止」に設定し、main 関数の呼び出し直前に「割り込み用スタック使用」、「割り込み許可」に設定しています。

3.1.8 スタックポインタの設定 (5.0 参考プログラム例 239 ~ 242 行目)

確保したスタック領域の最上位アドレスをスタックポインタに設定します。

本参考プログラム例では、割り込み用スタック領域のみを使用する設定 (スタックモードビット (SM) が常時"0") を前提に記述してあるため、ユーザ用スタックの設定は行っていません。

3.1.9 データ領域 (D、B) セクションの設定

組み込み用アプリケーションでは、初期値ありデータ領域 (D セクション)、初期値なしデータ領域 (B セクション) の初期設定が必要です。

アプリケーションリンク時、および、スタートアッププログラム実行時に以下の作業を行います。

(1) 組み込み用アプリケーションリンク時の作業

- ・ D、B セクション領域を RAM 領域に配置する (領域の確保のみ行い、データの出力は行わない)
- ・ D セクション用の初期値データ領域 (ROM_D) を ROM 領域に配置する。

(2) スタートアッププログラムでの初期化処理

- ・ RAM 領域内の B セクション領域を、ゼロクリア ("0"データで埋める) によって初期化する。
(5.0 参考プログラム例 245 ~ 256 行目)
- ・ ROM 領域の ROM_D セクションにあるデータを、RAM 領域内の D セクションの領域に転送することによって初期化する。
(5.0 参考プログラム例 259 ~ 272 行目)

3.1.10 ベースレジスタの設定 (5.0 参考プログラム例 275 ~ 291 行目)

コンパイル時に"-access"オプションを付加すると、D、B セクションのシンボルのアクセスに R11、R12、R13 をベースとするレジスタ相対間接の命令が出力されます。ここでは、その際に必要な R12 レジスタの設定と"_REL_BASE12"シンボルの定義を行っています。本参考プログラム例では"_REL_BASE12"に H'0080 8000 を定義していますので、H'0080 8000 から ±32KB の領域に対しレジスタ相対間接の命令が出力されます。また、"-access"オプションを付加する場合、上記のベースレジスタの設定以外にアクセス制御ファイルが必要です。詳細は「CC32R ユーザーズマニュアル C コンパイラ編」をご参照ください。

3.1.11 main 関数の呼び出し (5.0 参考プログラム例 300 行目)

main 関数の先頭アドレスにサブルーチン呼び出しを行います。main 関数から戻ってこない構成の場合は、単純なジャンプ命令でも構いません。

3.1.12 IE ビット操作関数 (5.0 参考プログラム例 308 ~ 322 行目)

- ・ 割り込み許可関数 (EnInt)
PSW の割り込みイネーブルビット (IE) を "1" にセットすることによって割り込みを許可します。
- ・ 割り込み禁止関数 (DisInt)
PSW の IE ビットを "0" にクリアすることによって割り込みを禁止します。

4.0 EIT の概要

CPU が通常のプログラムを実行している途中で、ある事象の発生によりそのプログラムの実行を中断し、別のプログラムを実行する必要がある場合があります。このような事象を総称して、EIT 事象と呼びます。

4.1 EIT の要因

M32R/ECU の EIT 事象には、次に示す要因があります。

(1) 例外 (Exception)

命令実行に伴うエラーや違反などによって発生するもので、次のものが該当します。

- ・予約命令例外 (RIE : Reserved Instruction Exception)
予約命令 (インプリメントされていない命令) の実行を検出した場合に発生します。
- ・アドレス例外 (AE : Address Exception)
ロード命令やストア命令でアライメントのとれていないアドレスにアクセスしようとした場合に発生します。

(2) 割り込み (Interrupt)

外部からのハードウェア的な信号によって発生するもので、次のものが該当します。

- ・リセット割り込み (RI : Reset Interrupt)
リセット信号を入力することにより常に受け付けられます。リセット割り込みは最高位の優先度をもちます。
- ・システムブレーク割り込み (SBI : System Brake Interrupt)
電源断の検出時や外部ウォッチドックタイマからの異常検出時に使用される緊急用割り込みです。割り込み処理後、原則として割り込み発生時に実行していた元のプログラムに復帰しない場合にのみ使用できます。
- ・外部割り込み (EI : External Interrupt)
ICU で管理される各内蔵周辺 I/O からの割り込み要求です。ICU では、割り込み禁止を含めて 8 レベルの優先順位で割り込み管理します。

(3) トラップ (Trap)

ソフトウェア割り込みのことでプログラム中の「TRAP 命令」の実行によって発生します。

4.2 EIT 処理の概要

EIT 事象が発生したとき、EIT 処理が行われます。EIT 処理には、ハードウェアが自動的に処理する部分とユーザが記述したプログラム (EIT ハンドラ) によって処理される部分があります。リセット割り込みを除く EIT 受付時の EIT 処理手順は次の通りです。

(1) EIT 受付時のハードウェア前処理

- ・レジスタ転送：PC BPC
- ・レジスタ転送：PSW BPSW

(2) EIT ハンドラ

- ・発生要因の対象となる EIT ベクタエントリ中の命令 (対象となる EIT ハンドラへの分岐命令：5.0 参考プログラム例 10～63 行目) を実行します。
- ・SBI を除く対象となる EIT ハンドラ処理を実行後、「RTE 命令」で EIT ハンドラからぬけます。

(3) 「RTE 命令」実行時のハードウェア後処理

- ・レジスタ転送：BPSW PSW
- ・レジスタ転送：BPC PC

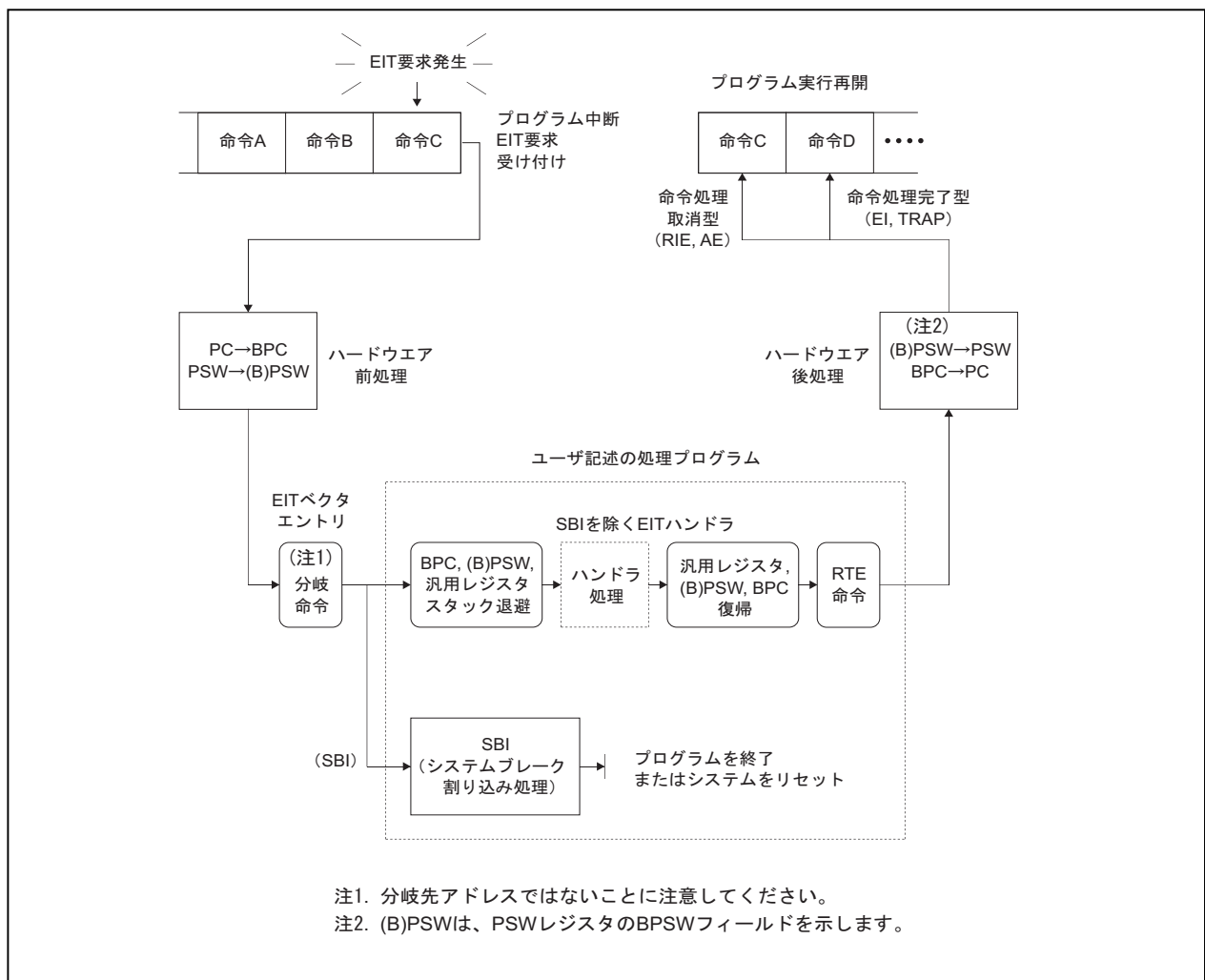


図 4.2.1 EIT 処理手順の概要

4.3 割り込みハンドラ処理手順

外部割り込み (EI) の EIT ハンドラ処理手順を下記に示します。また、「5.0 参考プログラム例」の 119 ~ 196 行目に外部割り込み (EI) ハンドラ処理のプログラム例を記述しています。本参考プログラム例の割り込みハンドラ処理は、パイプラインの動き等を考慮して最高速で実行できるように最適化してあります。

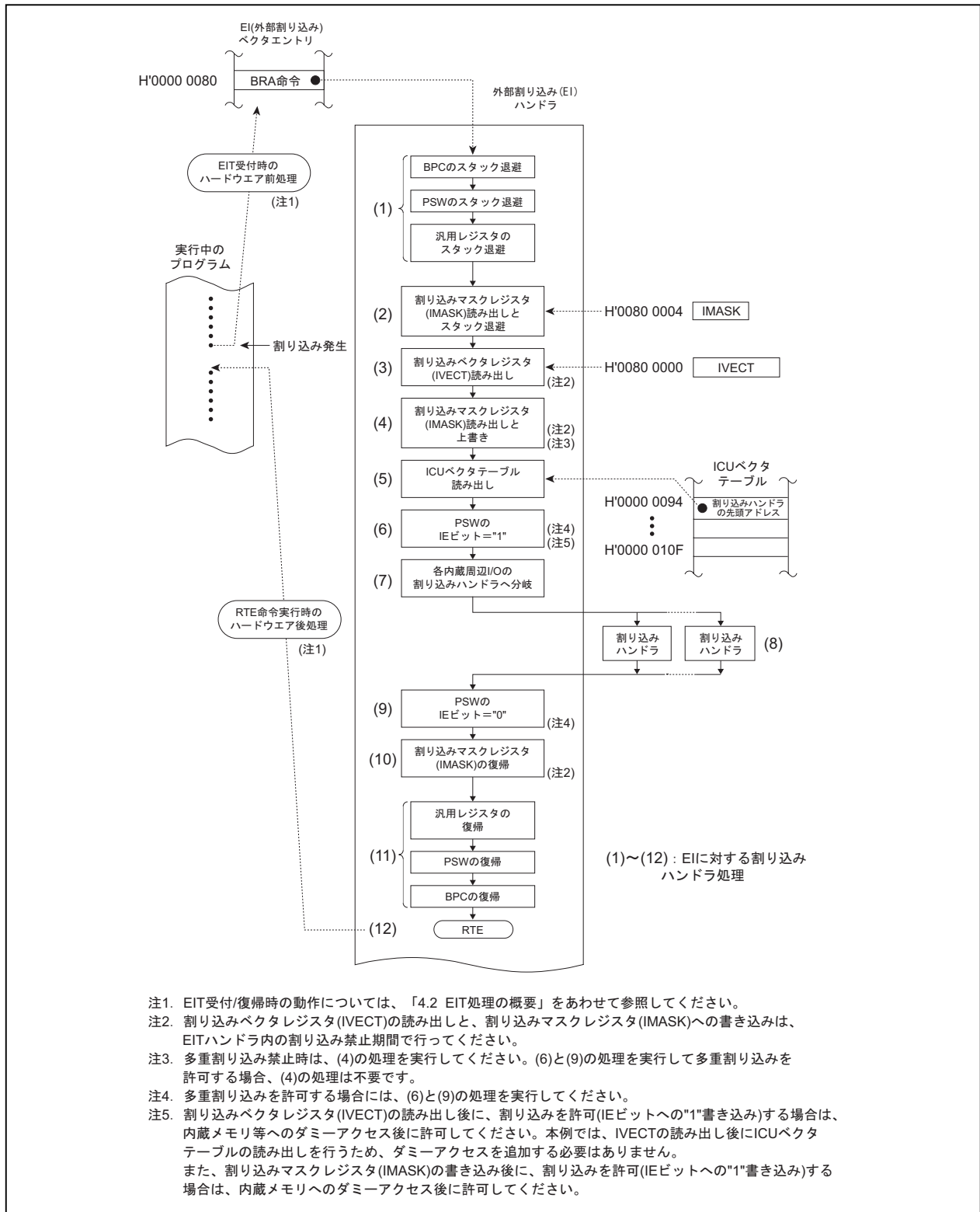


図 4.3.1 M32R/ECU の内蔵周辺 I/O からの EI ハンドラ処理動作例

ここでは、前ページ図中の外部割り込み (EI) ハンドラ処理の説明をします。

- (1) 各レジスタのスタックへの退避
(バックアップ PC、プロセッサ状態語レジスタ、汎用レジスタ、アキュムレータ)
 - ・バックアップ PC (BPC) は、EIT 発生時にプログラムカウンタ (PC) の値を退避します。EIT 発生時には EIT 発生直前の PC の値または次命令の値がセットされ「RTE 命令」実行時に BPC の値は PC に戻されます。
(5.0 参考プログラム例 146,161 行目)
 - ・PSW は、M32R のステータスを表示するレジスタで、通常使用する PSW フィールドと、EIT 発生時に PSW フィールドを退避するための BPSW フィールドからなります。EIT 発生時に PSW レジスタ内の SM,IE,C をそれぞれ BPSW に退避します。
(5.0 参考プログラム例 144,160 行目)
 - ・汎用レジスタ (R0 ~ R15) は、EIT ハンドラ内で使用するレジスタは必ず退避します。また、EIT ハンドラから実行されるユーザ処理内で使用するレジスタも退避が必要です。
(5.0 参考プログラム例 128~131,133,143,145,147,155,156 行目)
 - ・アキュムレータ (ACC) は、DSP 機能用命令実行中に外部割り込みが実行される可能性があり、その EIT ハンドラ内で DSP 機能用命令、または、乗算命令「MUL」を使用する場合は必ず退避します。
(5.0 参考プログラム例 136~139 行目)
- (2) 割り込みマスクレジスタの読み出しとスタックへの退避
(5.0 参考プログラム例 132,141,162 行目)
 - ・割り込みマスクレジスタ (IMASK) を読み出してスタックに退避します。
- (3) 割り込みベクタレジスタの読み出し
(5.0 参考プログラム例 142,149 行目)
 - ・割り込みベクタレジスタを読み出します。
 - ・この割り込みベクタレジスタ読み出しにより、ハードウェアは自動的に受け付けた割り込み要求要因の割り込み優先レベル (ILEVEL) を、新しい IMASK 値として IMASK レジスタにセットします。
- (4) 割り込み要求マスクレジスタの読み出しと上書き
(5.0 参考プログラム例 151 ~ 153 行目)
 - ・割り込み要求マスクレジスタを読み出し、読み出した値で上書きします。
 - ・この処理は、多重割り込みを許可に設定する場合には不要です。
- (5) ICU ベクタテーブルの読み出し
(5.0 参考プログラム例 158 行目)
 - ・受け付けた割り込み要求要因の ICU ベクタテーブルを読み出します。該当する ICU ベクタテーブルのアドレスは、(3)で読み出した割り込みベクタレジスタの内容 (受け付けた割り込み要求要因の ICU ベクタテーブルのアドレス下位 16 ビット) をゼロ拡張することで得られます。
- (6) 多重割り込みの許可
(5.0 参考プログラム例 134,164 行目)
 - ・割り込みの処理中にさらに優先レベルの高い割り込みを許可 (多重割り込みの許可) する場合には、PSW の IE ビットを"1"にセットします。
本参考プログラム例では、多重割り込みを許可に設定しています。

- (7) 各内蔵周辺 I/O の割り込みハンドラへの分岐
(5.0 参考プログラム例 166 行目)
・ (5) で読み出した割り込みハンドラ先頭番地へ分岐します。
- (8) 各内蔵周辺 I/O の割り込みハンドラ処理
- (9) 割り込み禁止
(5.0 参考プログラム例 168,169 行目)
・ PSW の IE ビットを "0" にクリアして、割り込みを禁止します。
- (10) 割り込みマスクレジスタの復帰
(5.0 参考プログラム例 171,172,177 行目)
・ (2) で退避した割り込みマスクレジスタを復帰します。
- (11) スタックからの各レジスタの復帰
(アキュムレータ、汎用レジスタ、プロセッサ状態語レジスタ、バックアップ PC)
(5.0 参考プログラム例 168,169,173 ~ 176,178 ~ 194 行目)
・ (1) で退避したレジスタを復帰します。
- (12) 外部割り込み処理の完了
(5.0 参考プログラム例 196 行目)
・ 「 RTE 命令 」 を実行し、外部割り込み処理を完了します。

4.4 内蔵周辺 I/O からの割り込み

内蔵周辺 I/O からの割り込みは、システムブレーク割り込み (SBI) と共に割り込みコントローラ (ICU) で管理され、外部割り込み (EI) として M32R CPU に伝えられます。

内蔵周辺 I/O からの割り込みは、全部で 31 要因あり、割り込み禁止を含めて 8 レベルの優先順位をつけて管理します。同一レベルの割り込み要求が複数同時に発生した場合は、あらかじめハードウェアで固定された優先順位が適用されます。内蔵周辺 I/O 内での割り込み要求発生元の特定は、内蔵周辺 I/O の割り込み要求ステータスレジスタを読むことで行います。

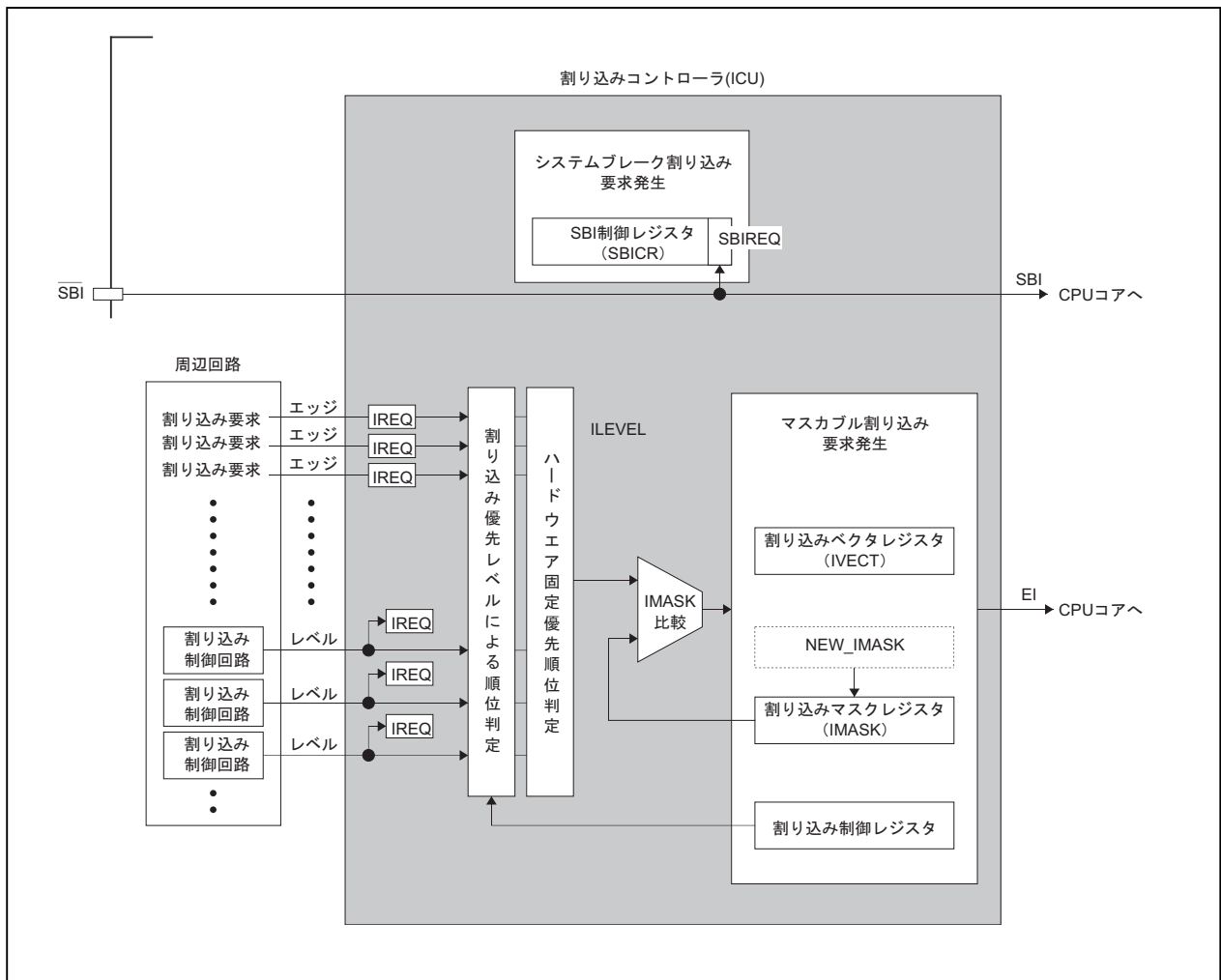


図 4.4.1 割り込みコントローラブロック図

4.4.1 内蔵周辺 I/O の割り込み要因

割り込みコントローラには、マルチジャンクションタイマ (MJT)、DMAC、シリアル I/O、A-D 変換器、RTD、CAN からの割り込み要求が入力されます。割り込みの詳細については、ユーザーズマニュアルの各内蔵周辺 I/O の章をご覧ください。

表 4.4.1 内蔵周辺 I/O 割り込み要因 (1/2)

| 割り込み要因 | 内容 | 入力要因数 | ICU 入力要因タイプ (注 1) |
|-------------------------|---|-------|-------------------|
| PDC 入力&エラー検出 割り込み | PDC 入力 0~3(TINA0,TINA1,TINB0,TINB1)検出 PDC 入力 0~3 エラー検出 | 8 | レベル |
| PWM オフ入力割り込み | PWM オフ入力 0,1 割り込み | 2 | レベル |
| A-D0 変換器割り込み | A-D0 変換器のスキャンモードのワンショット終了, 単一モード終了,コンパレータモード終了 | 1 | エッジ |
| A-D1 変換器割り込み | A-D1 変換器のスキャンモードのワンショット終了, 単一モード終了,コンパレータモード終了 | 1 | エッジ |
| SIO0 送信割り込み | SIO0 の送信バッファエンプティ割り込み | 1 | エッジ |
| SIO0 受信割り込み | SIO0 の受信完了,または受信エラー割り込み | 1 | エッジ |
| SIO1 送信割り込み | SIO1 の送信バッファエンプティ割り込み | 1 | エッジ |
| SIO1 受信割り込み | SIO1 の受信完了,または受信エラー割り込み | 1 | エッジ |
| SIO2,3 送受信割り込み | SIO2,3 の受信完了または受信エラー割り込み, 送信バッファエンプティ割り込み | 2 | レベル |
| SIO4 送信割り込み | SIO4 の送信バッファエンプティ割り込み | 1 | エッジ |
| SIO4 受信割り込み | SIO4 の受信完了,または受信エラー割り込み | 1 | エッジ |
| SIO5 送信割り込み | SIO5 の送信バッファエンプティ割り込み | 1 | エッジ |
| SIO5 受信割り込み | SIO5 の受信完了,または受信エラー割り込み | 1 | エッジ |
| SIO6,7 送受信割り込み | SIO6,7 の受信完了または受信エラー割り込み, 送信バッファエンプティ割り込み | 2 | レベル |
| RTD 割り込み | RTD 割り込み発生コマンド | 1 | エッジ |
| DMA 転送割り込み 0 | DMA0~4 の転送終了 | 5 | レベル |
| DMA 転送割り込み 1 | DMA5~9 の転送終了 | 5 | レベル |
| PDC コンペアマッチ& エラー割り込み | PDC コンペアマッチ、S エラー割り込み | 8 | レベル |
| CAN0 送受信&エラー 割り込み | CAN0 送受信完了,CAN0 受信完了,CAN0 エラーパッシブ, CAN0 エラーパスオフ,CAN0 パスエラー | 19 | レベル |
| CAN1 送受信&エラー 割り込み | CAN1 送受信完了,CAN1 受信完了,CAN1 エラーパッシブ, CAN1 エラーパスオフ,CAN1 パスエラー | 19 | レベル |

注 1. ICU 入力要因タイプ

- ・エッジ：ICU に入力される割り込み信号の立ち上がりエッジで、割り込み要求が発生します。
- ・レベル：ICU に入力される割り込み信号の"L"レベルの期間中、割り込み要求が発生します。レベルタイプの場合、ICU の割り込み制御レジスタ中 IRQ ビットに対するソフトウェアによるセット/クリアはできません。

表 4.4.2 内蔵周辺 I/O 割り込み要因 (2/2)

| 割り込み要因 | 内容 | 入力要因数 | ICU 入力要因 タイプ (注 1) |
|-------------|-----------------------------------|-------|-----------------------|
| TOM0 出力割り込み | TOM0_0 ~ TOM0_7 出力 | 8 | レベル |
| TOM1 出力割り込み | TOM1_0 ~ TOM1_7 出力 | 8 | レベル |
| TMS0 出力割り込み | TMS0 出力 | 1 | エッジ |
| TID0 出力割り込み | TID0 出力 | 1 | エッジ |
| TID1 出力割り込み | TID1 出力 | 1 | エッジ |
| タイマ入力割り込み 5 | タイマ入力割り込みグループ 5 (TIN8, TIN9 入力) | 2 | レベル |
| タイマ入力割り込み 4 | タイマ入力割り込みグループ 4 (TIN10, TIN11 入力) | 2 | レベル |
| タイマ入力割り込み 3 | タイマ入力割り込みグループ 3 (TIN20, TIN21 入力) | 3 | レベル |
| タイマ入力割り込み 2 | タイマ入力割り込みグループ 2 (TIN22, TIN23 入力) | 3 | レベル |
| タイマ入力割り込み 1 | タイマ入力割り込みグループ 1 (TIN16, TIN17 入力) | 3 | レベル |
| タイマ入力割り込み 0 | タイマ入力割り込みグループ 0 (TIN18, TIN19 入力) | 3 | レベル |

注 1. ICU 入力要因タイプ

- ・エッジ：ICU に入力される割り込み信号の立ち下がりエッジで、割り込み要求が発生します。
- ・レベル：ICU に入力される割り込み信号の"L"レベルの期間中、割り込み要求が発生します。レベルタイプの場合、ICU の割り込み制御レジスタ中 IRQ ビットに対するソフトウェアによるセット/クリアはできません。

ICU 入力要因タイプには、エッジタイプとレベルタイプがあります。それぞれのタイプの構成を示します。

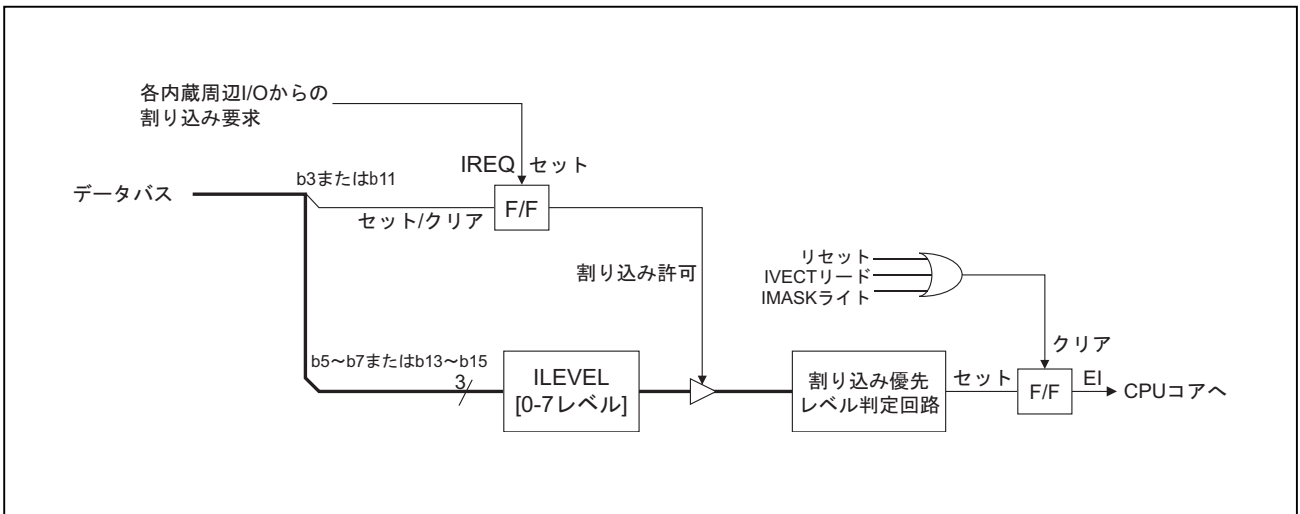


図 4.4.2 ICU の割り込み要因構成（エッジタイプ）

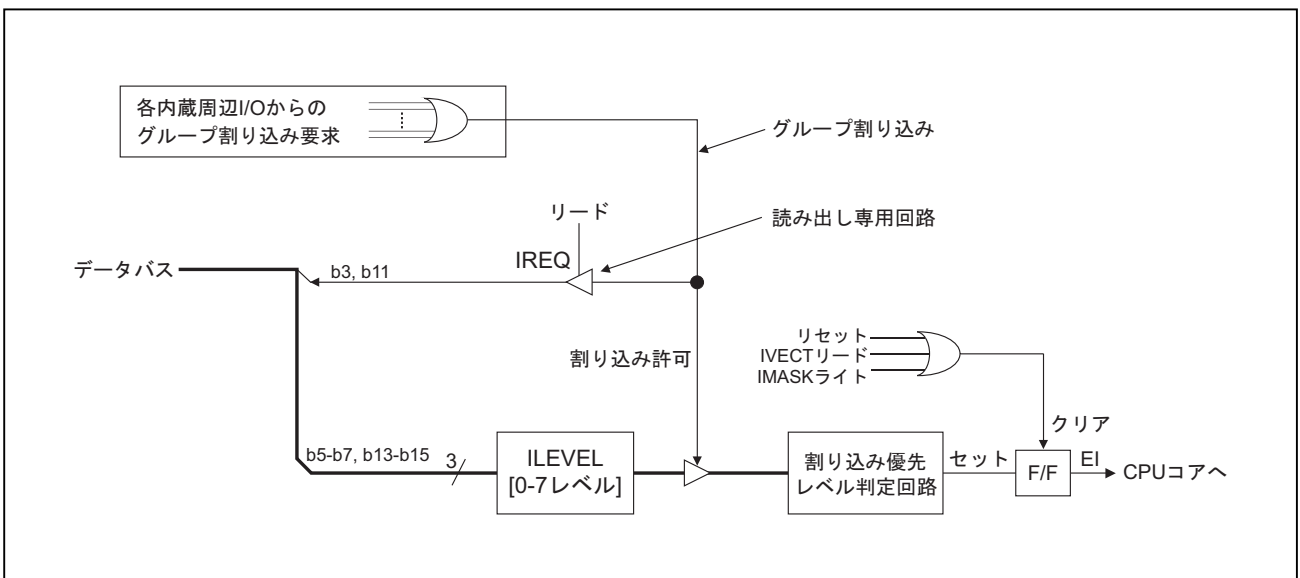


図 4.4.3 ICU の割り込み要因構成（レベルタイプ）

(1) エッジタイプの割り込み要因

エッジタイプの割り込み要因は ICU 内の割り込み制御レジスタの割り込み要求ビットに対し、ソフトウェアで書き込むことによって、セット/クリアすることができます。また、IVECT レジスタを読み出すことによってクリアされます。

(2) レベルタイプの割り込み要因

レベルタイプの割り込みは、1つの割り込みベクタテーブルに対して、割り込み要因が2つ以上グルーピングされ、割り当てられています。それら要因は、それぞれの内蔵周辺 I/O で割り込み制御レジスタを使用した割り込み要求制御と割り込み入力の判定を行った後に、ICU に入力されます。このため ICU 内の割り込み要求ビットは割り込み許可された割り込み要求の判定ビットとしてのみ機能し、書き込み処理は出来ません。下図に割り込み要求ステータスレジスタとマスクレジスタの構成例を示します。

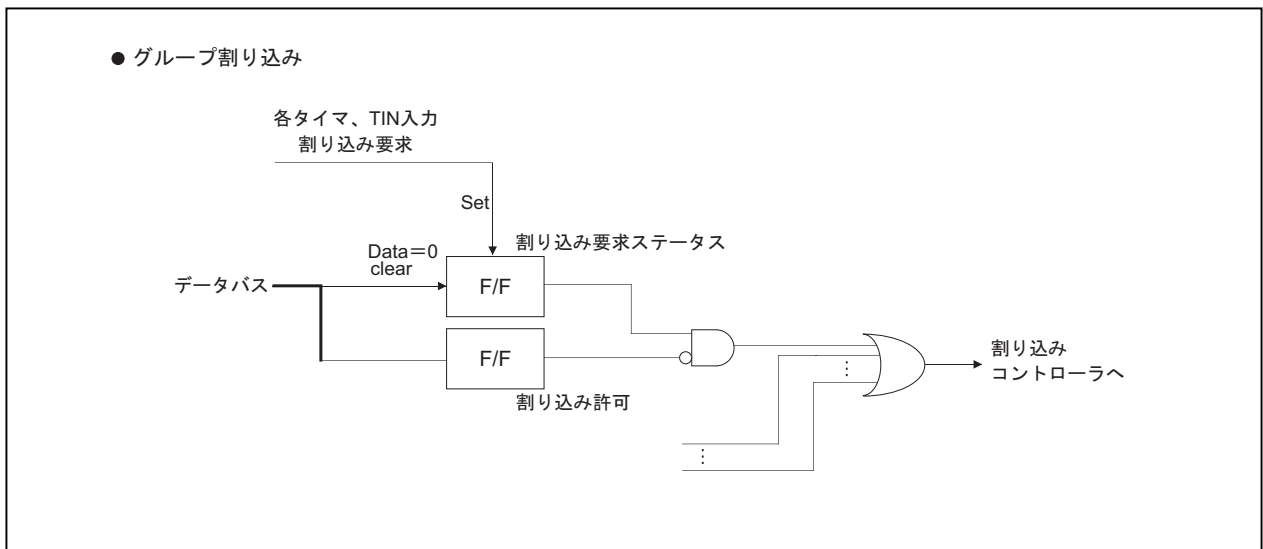
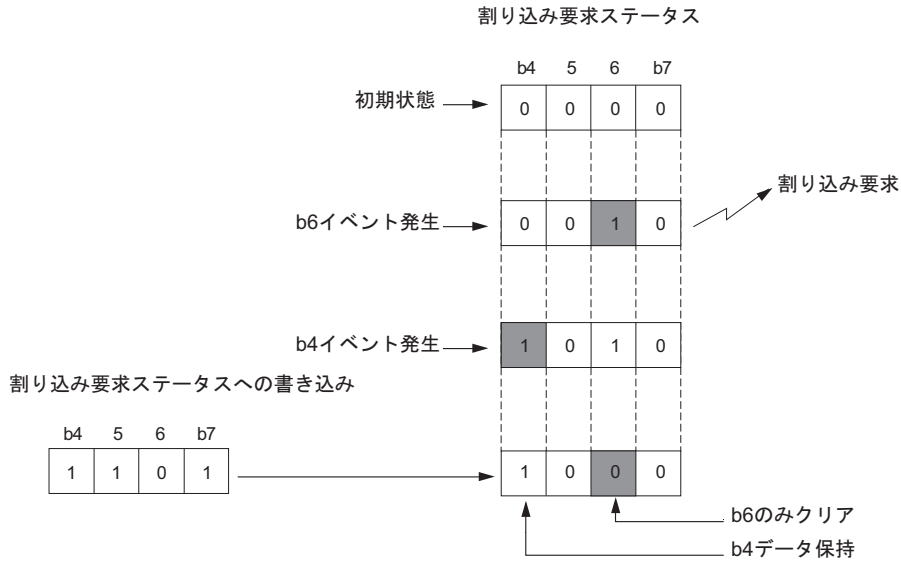


図 4.4.4 割り込み要求ステータスレジスタとマスクレジスタの構成例 (MJT)

・内蔵周辺 I/O の割り込み要求ステータスビット

割り込み要求を判別するためのステータスビットで、割り込み要求が発生するとハードウェア的にセットされ、ソフトウェア的にセットすることは出来ません。ステータスビットは、“0”を書き込むことによりクリアされ、“1”を書き込むとステータスビットの状態を保持します。なお、割り込みマスクビットの影響を受けず動作しますので、周辺機能の動作確認用にも使用することが出来ます。割り込み処理時には、グルーピングされた割り込み要求ステータスビットの内、割り込み処理を行ったステータスビットのみクリアしてください。割り込み処理を行っていないステータスビットをクリアすると未実行の割り込み要求もクリアされますので注意してください。グルーピングされている割り込みの割り込み要求ステータスビットをすべてクリアすると、その割り込みグループに対応したレベルタイプの割り込み要求がクリアされます。

● 割り込み要求ステータスクリア例



● プログラム例

- ・ 割り込み要求ステータスレジスタ (ISTREG) の割り込み要求ステータス1 : ISTAT1 (0x02ビット) をクリアする場合



ISTREG = 0xfd; /*ISTAT1 (0x02ビット) のみ クリア*/

割り込み要求ステータスをクリアする場合は、必ず他のステータスには"1"を書き込んで下さい。その際、下のように論理演算を用いるとISTREGの読み出し、論理演算、書き込みの3段階の手順となるため、読み出しから書き込みの間に他の割り込み要求が発生した場合に、誤ってクリアする場合があります。



ISTREG &= 0xfd; /*ISTAT1 (0x02ビット) のみ クリア*/

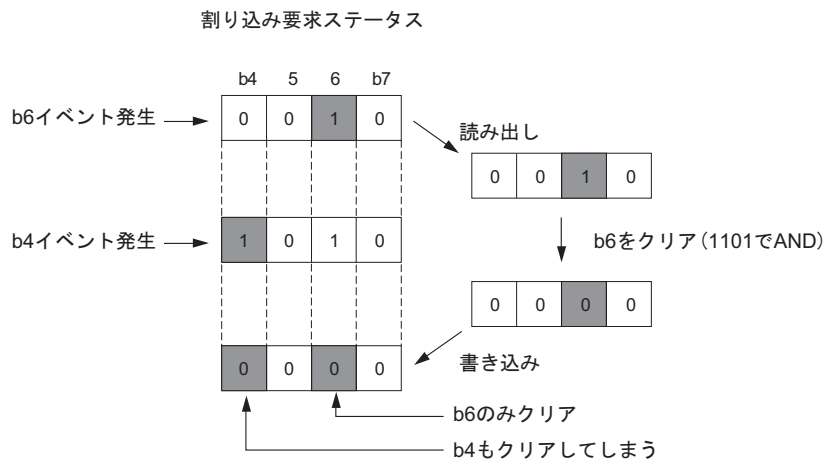


図 4.4.5 割り込み要求ステータスクリア例

5.0 参考プログラム例

スタートアッププログラムのプログラム例 (startup.ms) を示します。アプリケーションプログラムの内容によっては、以下の処理以外にも追加すべき処理がある場合、または、この中から削除すべき処理がある場合があります。

```

1 ;"FILE COMMENT" *****
2 ;* M32R C Programming Rev. 1.11
3 ;* < Sample startup Program for 32172/32173 >
4 ;*
5 ;* Copyright (c) 2003 Renesas Technology Corporation
6 ;* And Renesas Solutions Corporation
7 ;* All Rights Reserved
8 ;*****
9 ;
10 ;*****
11 ; EIT Vector Entry
12 ;*****
13 ;
14 .SECTION EITVECT, CODE, ALIGN=4
15 ;
16 .EXPORT reset, EIT_reset, EIT_loop
17 ;
18 reset:
19 BRA EIT_reset:24 ; H'0000 0000 Reset Interrupt (RI)
20 NOP
21 NOP
22 NOP
23 NOP
24 NOP
25 NOP
26 BRA EIT_loop:24 ; H'0000 0010 System Break Interrupt (SBI)
27 NOP
28 NOP
29 NOP
30 NOP
31 NOP
32 NOP
33 BRA EIT_loop:24 ; H'0000 0020 Reserved Instruction Exception (RIE)
34 NOP
35 NOP
36 NOP
37 NOP
38 NOP
39 NOP
40 BRA EIT_loop:24 ; H'0000 0030 Address Exception (AE)
41 NOP
42 NOP
43 NOP
44 NOP
45 NOP
46 NOP
47 BRA EIT_loop:24 ; H'0000 0040 Trap 0
48 BRA EIT_loop:24 ; H'0000 0044 Trap 1
49 BRA EIT_loop:24 ; H'0000 0048 Trap 2
50 BRA EIT_loop:24 ; H'0000 004C Trap 3
51 BRA EIT_loop:24 ; H'0000 0050 Trap 4
52 BRA EIT_loop:24 ; H'0000 0054 Trap 5
53 BRA EIT_loop:24 ; H'0000 0058 Trap 6
54 BRA EIT_loop:24 ; H'0000 005C Trap 7
55 BRA EIT_loop:24 ; H'0000 0060 Trap 8
56 BRA EIT_loop:24 ; H'0000 0064 Trap 9
57 BRA EIT_loop:24 ; H'0000 0068 Trap 10
58 BRA EIT_loop:24 ; H'0000 006C Trap 11
59 BRA EIT_loop:24 ; H'0000 0070 Trap 12
60 BRA EIT_loop:24 ; H'0000 0074 Trap 13
61 BRA EIT_loop:24 ; H'0000 0078 Trap 14
62 BRA EIT_loop:24 ; H'0000 007C Trap 15
63 BRA EIT_ei:24 ; H'0000 0080 External Interrupt (EI)
64 ;
65 .SECTION PROTECTID, DATA, ALIGN=1
66 .DATA.B H'FF,H'FF,H'FF,H'FF,H'FF,H'FF,H'FF,H'FF ; H'0000 0084 Protect ID

```

```

67  .DATA.B          H'FF,H'FF,H'FF,H'FF,H'FF,H'FF,H'FF,H'FF ;
68 ;
69 ;
70 ;*****
71 ; ICU Vector Table
72 ;*****
73 ;
74  .SECTION         ICUVECT, DATA, ALIGN=4
75 ;
76 ;
77 vectbl:
78  .DATA.W          EIT_reset          ; H'0000 0094   PDC Input & Error Detection Interrupt
79  .DATA.W          EIT_reset          ; H'0000 0098   PWM Off Input Interrupt
80  .DATA.W          EIT_reset          ; H'0000 009C   Timer Input Interrupt 5:TIN8,TIN9
81  .DATA.W          EIT_reset          ; H'0000 00A0   Timer Input Interrupt 4:TIN10,TIN11
82  .DATA.W          EIT_reset          ; H'0000 00A4   Timer Input Interrupt 3:TIN20,TIN21
83  .DATA.W          EIT_reset          ; H'0000 00A8   Timer Input Interrupt 2:TIN22,TIN23
84  .DATA.W          EIT_reset          ; H'0000 00AC   Timer Input Interrupt 1:TIN16,TIN17
85  .DATA.W          EIT_reset          ; H'0000 00B0   Timer Input Interrupt 0:TIN18,TIN19
86  .DATA.W          EIT_reset          ; H'0000 00B4   TOM0 Output Interrupt:TOM0_0-TOM0_7
87  .DATA.W          EIT_reset          ; H'0000 00B8   TOM1 Output Interrupt:TOM1_0-TOM1_7
88  .DATA.W          EIT_reset          ; H'0000 00BC   TMS0 Output Interrupt
89  .DATA.W          EIT_reset          ; H'0000 00C0   TID0 Output Interrupt
90  .DATA.W          EIT_reset          ; H'0000 00C4   TID1 Output Interrupt
91  .DATA.W          EIT_reset          ; H'0000 00C8   DMA0-4 Interrupt
92  .DATA.W          EIT_reset          ; H'0000 00CC   A-D0 Converter Interrupt
93  .DATA.W          EIT_reset          ; H'0000 00D0   SIO0 Receive Interrupt
94  .DATA.W          EIT_reset          ; H'0000 00D4   SIO0 Transmit Interrupt
95  .DATA.W          EIT_reset          ; H'0000 00D8   SIO1 Receive Interrupt
96  .DATA.W          EIT_reset          ; H'0000 00DC   SIO1 Transmit Interrupt
97  .DATA.W          EIT_reset          ; H'0000 00E0   A-D1 Converter Interrupt
98  .DATA.W          EIT_reset          ; H'0000 00E4   DMA5-9 Interrupt
99  .DATA.W          EIT_reset          ; H'0000 00E8   SIO2,3 Transmit/Receive Interrupt
100 .DATA.W          EIT_reset          ; H'0000 00EC   SIO4 Receive Interrupt
101 .DATA.W          EIT_reset          ; H'0000 00F0   SIO4 Transmit Interrupt
102 .DATA.W          EIT_reset          ; H'0000 00F4   SIO5 Receive Interrupt
103 .DATA.W          EIT_reset          ; H'0000 00F8   SIO5 Transmit Interrupt
104 .DATA.W          EIT_reset          ; H'0000 00FC   SIO6,7 Transmit/Receive Interrupt
105 .DATA.W          EIT_reset          ; H'0000 0100   RTD Interrupt
106 .DATA.W          EIT_reset          ; H'0000 0104   PDC Compare Match & Error Interrupt
107 .DATA.W          EIT_reset          ; H'0000 0108   CAN0 Transmit/Receive & Error Interrupt
108 .DATA.W          EIT_reset          ; H'0000 010C   CAN1 Transmit/Receive & Error Interrupt
109 ;
110 ;*****
111 ; Set SFR Address
112 ;*****
113 ;
114  .SECTION         SFR, DATA, ALIGN=1
115  .RES.B           H'4000
116 ;
117 IMASK             .EQU    H'00800004          ; IMASK Address
118 ;
119 ;*****
120 ; External Interrupt Handler
121 ;*****
122 ;
123  .SECTION         EIT_P, CODE, ALIGN=4
124 ;
125  .EXPORT          EIT_ei
126 ;
127 EIT_ei:
128  ST               R0,@-R15          ; PUSH R0              (R0)
129 ;
130  ST               R1,@-R15          ; PUSH R1              (R1)
131  ST               R2,@-R15          ; PUSH R2              (R2)
132  LD24             R0,#IMASK         ;
133  ST               R3,@-R15          ; PUSH R3              (R3)
134  LDI              R3,#H'40         ;
135 ;
136  MVFACHI          R1                ; PUSH Accumulator
137  MVFACLO          R2                ;
138  ST               R1,@-R15          ;                      (AccH)
139  ST               R2,@-R15          ;                      (AccL)
140 ;
141  LDB              R2,@R0            ; Read IMASK(H'0080 0004) Register

```

32172/32173 グループ スタートアッププログラム例

```

142  ADDI      R0,#-4          ;
143  ST        R4,@-R15       ; PUSH  R4          (R4)
144  MVFC     R4,PSW          ;
145  ST        R5,@-R15       ; PUSH  R5          (R5)
146  MVFC     R5,BPC          ;
147  ST        R6,@-R15       ; PUSH  R6          (R6)
148  ;
149  LDH      R1,@R0          ; Read IVECT(H'0080 0000) Register
150  ;
151  LD24     R0,#IMASK        ; Overwrite IMASK Register
152  LDB      R6,@R0          ;
153  STB      R6,@R0          ;
154  ;
155  ST        R7,@-R15       ; PUSH  R7          (R7)
156  ST        R14,@-R15      ; PUSH  Link Register (R14)
157  ;
158  LD        R1,@R1         ; Read ICU Vector Table
159  ;
160  ST        R4,@-R15       ; PUSH  PSW          (PSW)
161  ST        R5,@-R15       ; PUSH  BPC          (BPC)
162  ST        R2,@-R15       ; PUSH  IMASK Register (IMASK)
163  ;
164  MVTC     R3,PSW          ; Enable Interrupt
165  ;
166  JL        R1              ; Call Interrupt Handler
167  ;
168  LDI      R5,#H'00         ;
169  MVTC     R5,PSW          ; Disable Interrupt
170  ;
171  LD24     R3,#IMASK        ;
172  LD        R2,@R15+        ; POP   IMASK          (IMASK)
173  LD        R1,@R15+        ; POP   BPC           (BPC)
174  LD        R0,@R15+        ; POP   PSW           (PSW)
175  ;
176  LD        R14,@R15+       ; POP   Link Register (R14)
177  STB      R2,@R3          ;
178  LD        R7,@R15+       ; POP   R7            (R7)
179  MVTC     R1,BPC          ;
180  LD        R6,@R15+       ; POP   R6            (R6)
181  LD        R5,@R15+       ; POP   R5            (R5)
182  LD        R4,@R15+       ; POP   R4            (R4)
183  ;
184  LD        R2,@R15+       ; POP   Accumulator  (AccL)
185  LD        R1,@R15+       ;                               (AccH)
186  MVTC     R0,PSW          ;
187  MVTACLO R2              ;
188  MVTACHI R1              ;
189  ;
190  LD        R3,@R15+       ; POP   R3            (R3)
191  LD        R2,@R15+       ; POP   R2            (R2)
192  LD        R1,@R15+       ; POP   R1            (R1)
193  ;
194  LD        R0,@R15+       ; POP   R0            (R0)
195  ;
196  RTE
197  ;
198  ;*****
199  ; EIT_loop
200  ;*****
201  ;
202  EIT_loop:
203  BRA      EIT_loop
204  ;
205  ;*****
206  ; Start Up Program
207  ;*****
208  ;
209  .IMPORT  $main
210  ;
211  .SECTION C, DATA, ALIGN=4
212  .SECTION D, DATA, ALIGN=4
213  .SECTION B, DATA, ALIGN=4
214  .SECTION ROM_D, DATA, ALIGN=4
215  ;
216  ;*****

```

```

217 ; Set Interrupt Stack
218 ;
219     .SECTION          SPINT, DATA, ALIGN=4
220     .RES.B           2048                ; Interrupt Stack Area
221 ;
222 ;+++++
223 ; Set User Stack
224 ;
225     .SECTION          SPUSR, DATA, ALIGN=4
226     .RES.B           2048                ; User Stack Area
227 ;
228 ;+++++
229 ; Startup & Exit
230 ;
231     .SECTION          P, CODE, ALIGN=4
232 EIT_reset:
233     LDI               R0, #'00           ;
234     MVTC              R0, PSW            ; Disable Interrupt
235     LDI               R0, #7             ;
236     LD24              R1, #IMASK        ;
237     STB               R0, @R1           ;
238 ;
239 ; LD24              R1, #(SPUSR+sizeof(SPUSR)) ;
240     LD24              R2, #(SPINT+sizeof(SPINT)) ;
241 ; MVTC              R1, SPU              ; Set User Stack Pointer
242     MVTC              R2, SPI            ; Set Interrupt Stack Pointer
243 ;
244 ;+++++
245 ; Clear B Section
246 ;
247     LD24              R5, #sizeof(B)
248     BLEZ              R5, loop_cnt0
249     LD24              R4, #B
250     LDI               R0, #0
251 loop0:
252     STB               R0, @R4
253     ADDI              R4, #1
254     ADDI              R5, #-1
255     BNEZ              R5, loop0
256 loop_cnt0:
257 ;
258 ;+++++
259 ; Data Set (ROM_D Section => D Section)
260 ;
261     LD24              R6, #sizeof(ROM_D)
262     BLEZ              R6, loop_cnt1
263     LD24              R4, #D
264     LD24              R5, #ROM_D
265 loop1:
266     LDB               R0, @R5
267     STB               R0, @R4
268     ADDI              R4, #1
269     ADDI              R5, #1
270     ADDI              R6, #-1
271     BNEZ              R6, loop1
272 loop_cnt1:
273 ;
274 ;+++++
275 ; Set Base Register
276 ;
277 ;     .EXPORT          __REL_BASE13
278 ;     .EXPORT          __REL_BASE12
279 ;     .EXPORT          __REL_BASE11
280 ; __REL_BASE13      .EQU    0x00808000
281 __REL_BASE12      .EQU    0x00808000
282 ; __REL_BASE11      .EQU    0x00808000
283 ;
284 ;     SETH             R13, #HIGH(__REL_BASE13)
285 ;     OR3              R13, R13, #LOW(__REL_BASE13)
286 ;
287 ;     SETH             R12, #HIGH(__REL_BASE12)
288 ;     OR3              R12, R12, #LOW(__REL_BASE12)
289 ;
290 ;     SETH             R11, #HIGH(__REL_BASE11)
291 ;     OR3              R11, R11, #LOW(__REL_BASE11)

```

```
292 ;
293 ;+++++
294 ; Call main()
295 ;
296 ;   LDI           R0, #'C0           ; Enable Interrupt , Use User Stack
297 ;   LDI           R0, #'40           ; Enable Interrupt , Use Interrupt Stack
298 ;   MVTC          R0, PSW
299 ;
300 ;   BL            $main              ; Call C(main) Routine
301 ;
302 ;+++++
303 ;
304 endless:
305 ;   BRA           endless            ; Dummy
306 ;
307 ;+++++
308 ; Set/Clear IE Flag Routine
309 ;
310 ;   .EXPORT       $EnInt
311 ;   .EXPORT       $DisInt
312 $EnInt:
313 ;   MVFC          R0, PSW
314 ;   OR3           R0, R0, #'0040
315 ;   MVTC          R0, PSW
316 ;   JMP           R14
317 ;
318 $DisInt:
319 ;   MVFC          R0, PSW
320 ;   AND3          R0, R0, #'FFBF
321 ;   MVTC          R0, PSW
322 ;   JMP           R14
323 ;
324 ;   .END
```

6.0 参考ドキュメント

データシート

- ・ 32172/32173 グループデータシート Rev.0.20

(最新版をルネサス テクノロジホームページから入手してください。)

ユーザーズマニュアル

- ・ 32172/32173 グループ ユーザーズマニュアル Rev.1.20
- ・ M3T-CC32R V.4.10 ユーザーズマニュアル (C コンパイラ編)
- ・ M3T-AS32R V.4.10 ユーザーズマニュアル (アセンブラ編)

(最新版をルネサス テクノロジホームページから入手してください。)

ソフトウェアマニュアル

- ・ M32R ファミリ ソフトウェアマニュアル Rev.1.10

(最新版をルネサス テクノロジホームページから入手してください。)

7.0 ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://www.renesas.com/>

ルネサス製品全般に関するお問合せ先

カスタマ・サポート・センター : csc@renesas.com

M32R ファミリに関する技術的なお問合せ先

M32R ファミリ MCU 技術サポート窓口 : support_apl@renesas.com

安全設計に関するお願い

- ・ 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

- ・ 本資料は、お客様が用途に応じた適切なルネサス テクノロジー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 テクノロジー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。
- ・ 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 テクノロジーは責任を負いません。
- ・ 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 テクノロジーは、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 テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、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 テクノロジーホームページ (<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。
- ・ 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 テクノロジーはその責任を負いません。
- ・ 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 テクノロジーは、適用可否に対する責任を負いません。
- ・ 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、海底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 テクノロジー、ルネサス販売または特約店へご照会ください。
- ・ 本資料の転載、複製については、文書によるルネサス テクノロジーの事前の承諾が必要です。
- ・ 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 テクノロジー、ルネサス販売または特約店までご照会ください。