

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。

H8/300L SLP シリーズ

非同期イベントカウンタ動作

要旨

非同期イベントカウンタ(AEC:Asynchronous Event Counter)を使用して、サブアクティブ、アクティブモード間の遷移、およびポートの反転を行ないます。

動作確認デバイス

H8/38024

目次

1. 仕様	2
2. 使用機能説明	2
3. 動作説明	7
4. ソフトウェア説明	8
5. フローチャート	14
6. プログラムリスト	17

1. 仕様

- (1) 非同期イベントカウンタを使用して、524.288ms 毎にサブアクティブモードからアクティブ(高速)モードへ遷移させ、アクティブ(高速)モードでポート出力を反転させ、再びサブアクティブモードへ遷移します。
- (2) 2MHz のイベント入力を非同期イベント入力 L(AEVL)端子へ入力します。
- (3) 本タスク例では、16 ビット非同期イベントカウンタとして使用します。

2. 使用機能説明

- (1) 本タスク例では、非同期イベントカウンタ(AEC:Asynchronous Event Counter)を使用して、サブアクティブ、アクティブモード間の遷移、およびポートの反転を行いません。以下に、非同期カウンタの特徴を示します。
 - 基本クロックの動作とは無関係に、非同期に入力される外部イベントをカウント可能です。
 - カウンタは 16 ビット構成になっており、65536 回以内のイベントのカウントが可能です。
 - 2 チャンネルの独立した 8 ビットイベントカウンタとしても使用可能です。
 - ソフトウェアによるカウンタのリセット、カウントアップ機能の停止が制御可能です。
 - イベントカウンタのオーバーフローを検出し、自動的に割込みを発生します。
 - モジュールスタンバイモードにより、未使用時はモジュール単体でスタンバイモードに設定可能です。
- (2) 図 1 に本タスク例で使用する 16 ビット非同期イベントカウンタのブロック図を示します。

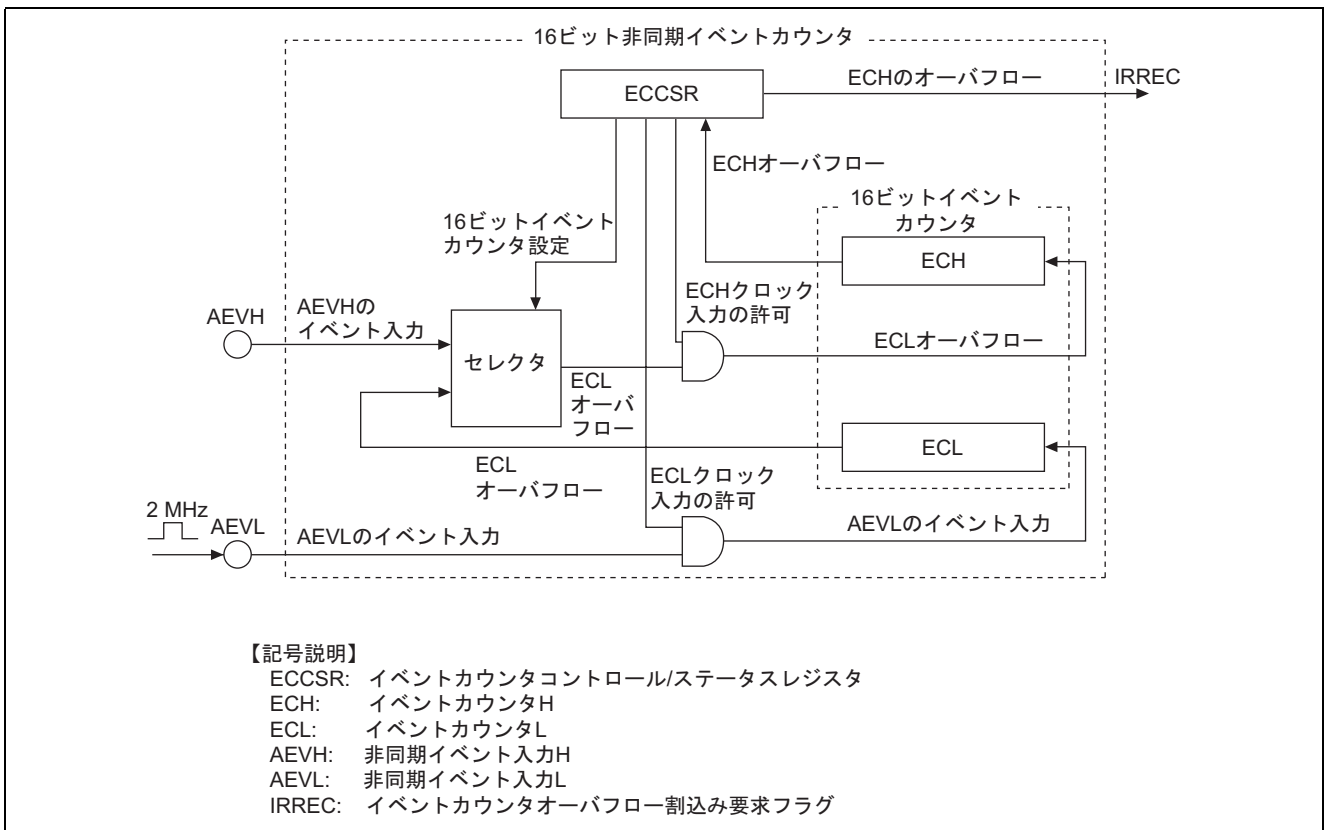


図 1 非同期イベントカウンタブロック図

(3) 表 1 に 16 ビット非同期イベントカウンタの各機能について説明します。

表 1 16 ビット非同期イベントカウンタ機能

入力端子エッジセレクトレジスタ (AEGSR)	
機能	AEGSR は 8 ビットのリード/ライト可能なレジスタで, AEVH, AEVL, IRQAEC 端子の立ち上がり, 立ち下り, 両エッジセンスを選択します。
イベントカウンタコントロールレジスタ (ECCR)	
機能	ECCR は 8 ビットのリード/ライト可能なレジスタで, カウンタの入カクロック, IRQAEC / IECPWM の制御を行ないます。
イベントカウンタコントロール/ステータスレジスタ (ECCSR)	
機能	ECCSR は 8 ビットのリード/ライト可能なレジスタで, カウンタのオーバフローの検出, カウンタのリセット, カウントアップ機能の停止の制御を行ないます。リセット時, ECCSR は, H'00 にイニシャライズされます。
イベントカウンタ H (ECH)	
機能	ECH は 8 ビットのリード可能なアップカウンタで, 独立した 8 ビットのイベントカウンタとして, または ECL と組み合わせることで 16 ビットのイベントカウンタの上位 8 ビットのアップカウンタとして動作します。入力クロックは外部非同期イベント AEVH 端子, または下位の 8 ビットカウンタ ECL からのオーバフロー信号のいずれかを選択可能です。ECH はソフトウェアで H'00 にクリア可能です。リセット時, EC は H'00 にイニシャライズされます。
イベントカウンタ L (ECL)	
機能	ECL は 8 ビットのリード可能なアップカウンタで, 独立した 8 ビットのイベントカウンタとして, または ECH と組み合わせることで 16 ビットのイベントカウンタの下位 8 ビットのアップカウンタとして動作します。入力クロックは外部非同期イベント AEVL 端子からのイベントクロックを使用します。ECL はソフトウェアで H'00 にクリア可能です。リセット時, ECL は H'00 にイニシャライズされます。
非同期イベント入力 H (AEVH)	
機能	AEVH はイベントカウンタ H (ECH) に入力するイベント入力端子です。
非同期イベント入力 L (AEVL)	
機能	AEVL はイベントカウンタ L (ECL) に入力するイベント入力端子です。
非同期イベントカウンタ割込み要求フラグ (IRREC)	
機能	非同期イベントカウンタ割込み要求が発生すると, IRREC は "1" にセットされます。また, IRREC は割込みが受け付けられてもオートクリアされません。IRREC をクリアする場合は, ソフトウェアで "0" をライトしてクリアして下さい。
非同期イベントカウンタ割込みイネーブル (IENEC)	
機能	非同期イベントカウンタ割込み要求の許可/禁止を制御します。

(4) 16 ビット非同期イベントカウンタとして使用する場合の設定方法の例を図 2 に示します。

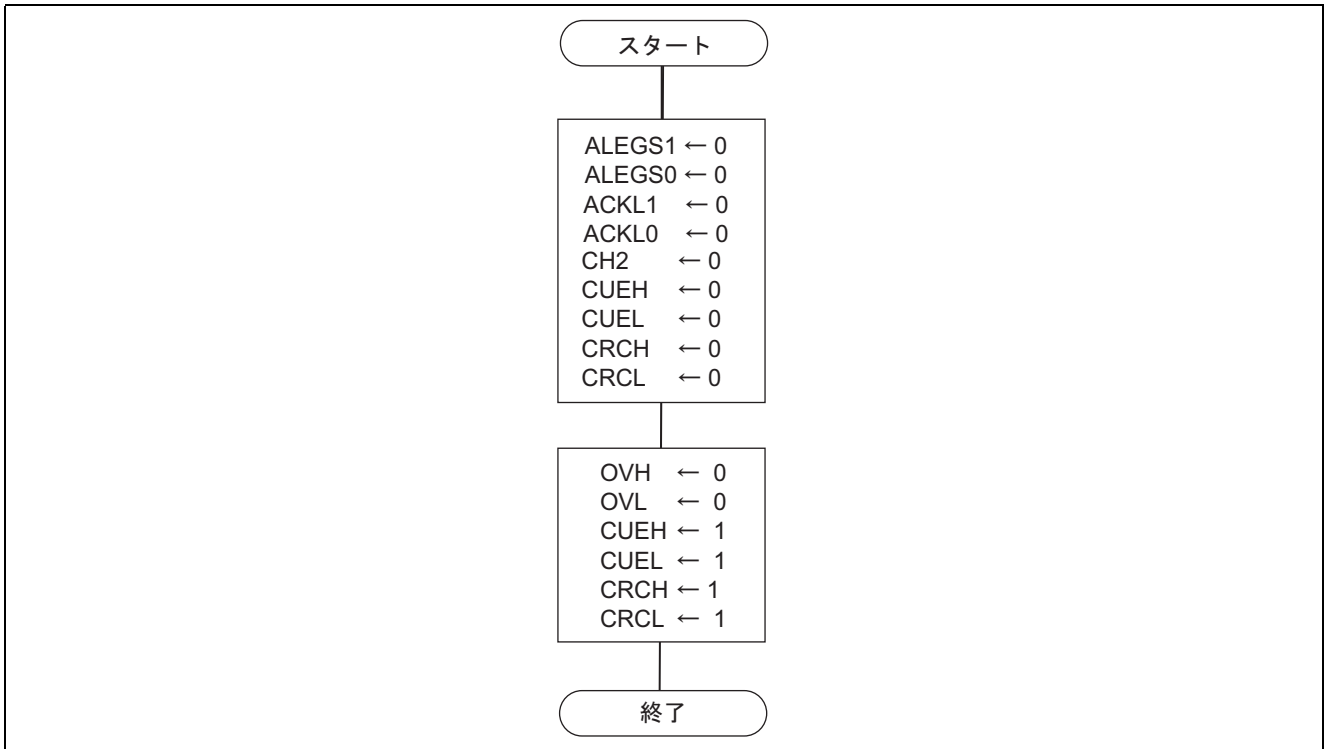


図 2 16 ビット非同期イベントカウンタの設定方法例

リセット時、CH2 は"0"にクリアされるため、リセット後は ECH、ECL は 16 ビットイベントカウンタとして動作します。また、図 2 に示すように設定することによっても、16 ビットイベントカウンタとして動作します。動作クロックは AEVL 端子からの非同期イベント入力です。ECH、ECL のカウント値とともに H'FF になった後に、クロックが入力されると ECH、ECL はオーバフローし、ECCSR の OVH フラグが"1"にセットされ、ECH、ECL のカウント値はそれぞれ H'00 に戻り、再びカウントアップを再開します。オーバフロー発生時には、IRR2 の IRREC が"1"にセットされます。このとき、IENR2 の IENEC が"1"ならば CPU に割り込みを要求します。

(5) 非同期イベントカウンタの動作モードを表 2 に示します。

表 2 非同期イベントカウンタの動作モード

動作モード	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
AEGSR	リセット	動作	動作	保持 ^{*1}	動作	動作	保持 ^{*1}	保持
ECCR	リセット	動作	動作	保持 ^{*1}	動作	動作	保持 ^{*1}	保持
ECCSR	リセット	動作	動作	保持 ^{*1}	動作	動作	保持 ^{*1}	保持
ECH	リセット	動作	動作	動作 ^{*1*2}	動作 ^{*2}	動作 ^{*2}	動作 ^{*1*2}	停止
ECL	リセット	動作	動作	動作 ^{*1*2}	動作 ^{*2}	動作 ^{*2}	動作 ^{*1*2}	停止

【注】 1. 非同期外部イベントが入力されるとカウンタはカウントアップしますが、カウンタオーバフロー H/L フラグは影響を受けません。

2. 非同期外部イベントを選択した場合に動作します。その他は停止して保持します。

(6) 16 ビット非同期イベントカウンタにおける注意事項

- (a) ECH, ECL の値をリードする場合には, リードする前に ECCSR の CUEH, CUEL を "0" にクリアして非同期イベント入力をカウンタに入力しないようにしてください。リードしている際にイベントカウンタがカウントアップすると正しい値がリードできません。
- (b) AEVH, AEVL 端子に入力するクロックの周波数は, 最大 16MHz までの範囲として下さい。またクロックの High 幅, Low 幅は, 最小 30ns となるようにして下さい。デューティ比はいくつでもかまいません。
- (c) 16 ビットモードで使用する際, ECCSR の設定は, CUEH を "1" にセットしてから CRCH を "1" にセットするか, CUEH と CRCH を同時にセットしてからクロックを入力してください。その後 16 ビットモードで使用中は CUEH の値を変更しないで下さい。16 ビットモード設定中に CUEH を変化させると ECH が誤カウントアップすることがあります。
- (d) AEGSR の ECPWME が 1 のとき, イベントカウンタ PWM は動作中ですので ECPWCRH, ECPWCRL, ECPWDRH, ECPWDRL を書き換えしないでください。データを変更する際は必ず AEGSR の ECPWME を 0 にしてイベントカウンタ PWM を停止させて書き換えてください。
- (e) イベントカウンタ PWM データレジスタとイベントカウンタ PWM コンペアレジスタは必ずイベントカウンタ PWM データレジスタ < イベントカウンタ PWM コンペアレジスタの関係で設定してください。それ以外の設定のときに AEGSR の ECPWME を 1 にセットしないでください。
- (f) IRQAEC は内部で同期をとって割り込みを発生しているためクロックの停止と割り込み受け付けまでは最大 1t_{cy} の誤差が生じます。
- (g) 表 3 に動作モードとイベント入力周波数について示します。

表 3 動作モードと AEVH, AEVL 端子イベント入力の周波数の関係

モード	AEVL, AEVH 端子に入力する 最大クロック周波数
アクティブ(高速), スリープ(高速)	16MHz
アクティブ(中速), スリープ(中速)	2 · f _{osc}
(/16)	f _{osc}
(/32)	1/2 · f _{osc}
(/64)	1/4 · f _{osc}
f _{osc} = 1MHz ~ 4MHz	(/128)
ウォッチ, サブアクティブ, サブスリープ,	(w /2)
スタンバイ	(w /4)
w = 32.768kHz または 38.4kHz	(w /8)
	1000kHz
	500kHz
	250kHz

(7) 表 4 に本タスク例の機能割付けを示します。

表 4 機能割付け

機能	機能割付け
AECSR	AEVL の立ち上がり, 立ち下り, 両エッジセンスを選択します。
ECCR	AEVL カウンタの入力クロックを制御します。
ECCSR	16 ビット非同期イベントカウンタ機能の設定, カウンタオーバフローの検出, ECH, ECL に入力するイベントクロックの入力の許可/禁止を行ないます。
ECH	ECL のオーバフロー信号を入力クロックとする 16 ビットイベントカウンタの上位 8 ビットのアップカウンタとして機能します。
ECL	AEVL 端子の外部非同期イベント入力を入力クロックとする 16 ビットイベントカウンタの下位 8 ビットのアップカウンタとして機能します。
COM	P40/SCK32 端子機能の設定
SCR3	P40/SCK32 端子機能の設定
TMA3	サブアクティブモードへの遷移を設定
AEVL	2MHz の外部非同期イベント入力の入力端子として機能します。
P40	ECH,ECL のオーバフロー 10 回をトリガとして, P40 出力を反転します。
PCR40	P40 端子を出力機能に設定
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
IENDT	直接遷移割込み要求の許可/禁止を制御します。
IENEC	非同期イベントカウンタ割込み要求の許可/禁止を制御します。
IRRDT	直接遷移割込み要求の有無を反映します。
IRREC	非同期イベントカウンタ割込み要求の有無を反映します。

3. 動作説明

(1) 図3に動作説明を示します。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非同期イベントカウンタ動作を行ないます。

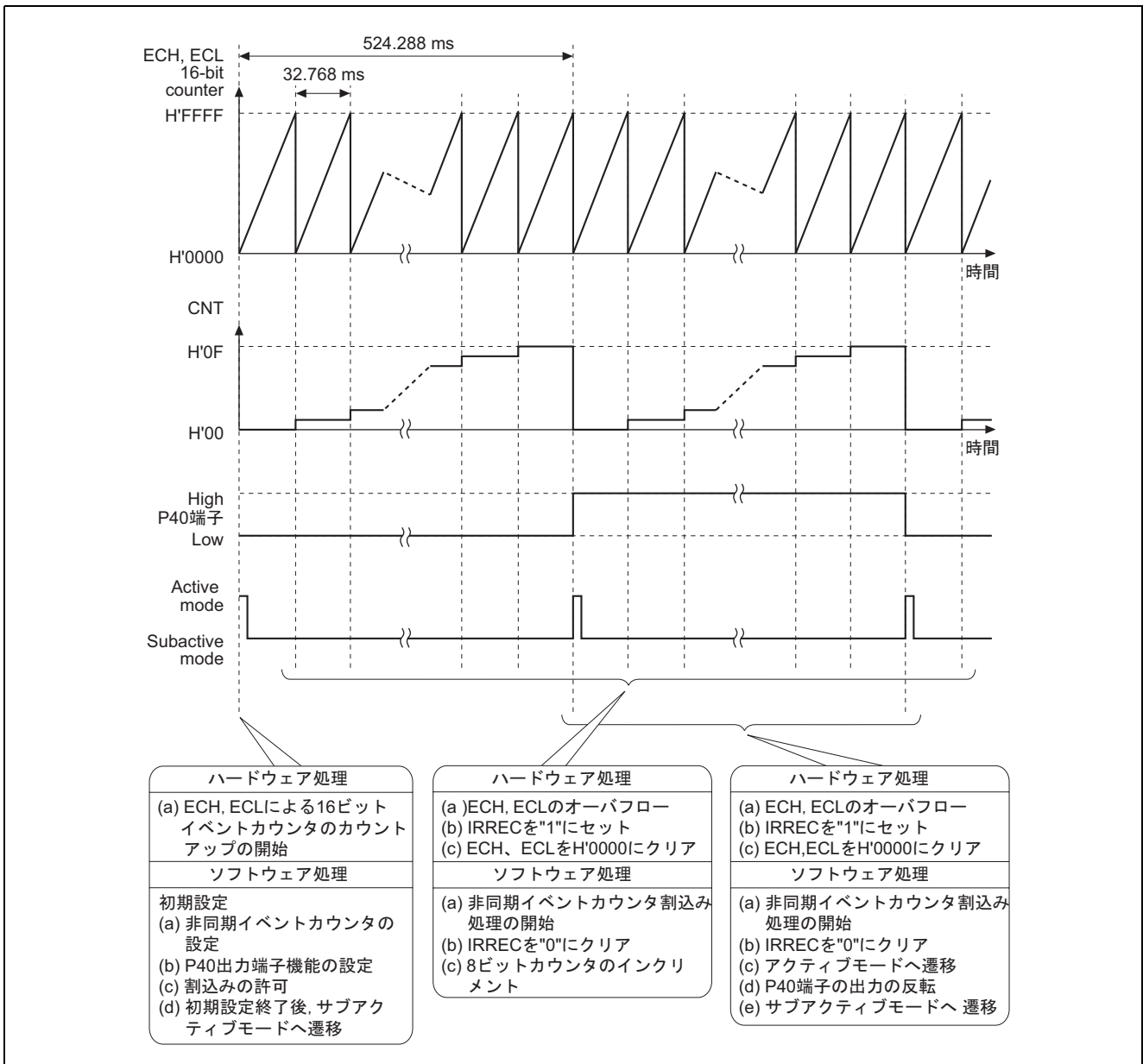


図3 非同期イベントカウンタ動作の動作説明

4. ソフトウェア説明

(1) モジュール説明

本タスク例のモジュールを表 5 に示します。

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	使用RAM, ポート40, 非同期イベントカウンタ, システムコントロールレジスタの初期設定, 割込みの許可, サブアクティブモードへの直接遷移, 524.288ms 経過後, ポート出力制御, アクティブ(高速)モード, サブアクティブモードへの直接遷移を行なう。
非同期イベントカウンタ割込み処理ルーチン	aecint	非同期イベントカウンタ割込み処理ルーチンで, 割込み要求フラグのクリア, 8ビットカウンタのインクリメントおよびイニシャライズ, 524.299ms 経過後, RAM に設定したフラグを立てる。
直接遷移割込み処理ルーチン	dtint	直接遷移割込み処理ルーチンで, 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, 引数を使用しません。

(3) 使用内部レジスタ説明

本タスク例の使用内部レジスタを表 6 に示します。

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
AECSR	ALEGS1 ALEGS0	入力端子エッジセレクトレジスタ(AEC エッジ選択 L) : ALEGS1="0", ALEGS0="0"のとき, AEVL 立ち上がりエッジを検出します。	H'FF92 ビット 5 ビット 4	ALEGS1=0 ALEGS0=0
ECCR	ACKL1 ACKL0	イベントカウンタコントロールレジスタ (AEC クロック選択 L) : ACKL1="0", ACKL0="0"のとき, ECL 側で使用するクロックを AEVL 端子で入力します。	H'FF94 ビット 5 ビット 4	ACKL1=0 ACKL0=0
ECCSR	OVH	イベントカウンタコントロール/ステータスレジスタ (カウンタオーバーフローH) ~ ECH がオーバーフローしたことを示すステータスフラグ。 : OVH="0"のとき, ECH がオーバーフローしていないことを示す : OVH="1"のとき, ECH がオーバーフローしたことを示す	H'FF95 ビット 7	0
	OVL	イベントカウンタコントロール/ステータスレジスタ (カウンタオーバーフローL) ~ ECL がオーバーフローしたことを示すステータスフラグ。 : OVL="0"のとき, ECL がオーバーフローしていないことを示す : OVL="1"のとき, ECL がオーバーフローしたことを示す	H'FF95 ビット 6	0

表 6 使用内部レジスタ説明(つづき)

レジスタ名	機能	アドレス	設定値	
ECCSR	CH2	イベントカウンタコントロール/ステータスレジスタ (チャンネル選択) ~ ECH, ECL を 1 チャンネルの 16 ビットイベントカウンタとして使用するか, 2 チャンネルの独立した 8 ビットのイベントカウンタとして使用するかを選択。 : CH2="0" のとき, ECH, ECL を連結した 1 チャンネルの 16 ビットイベントカウンタとして使用します : CH2="1" のとき, ECH, ECL を独立した 2 チャンネルの 8 ビットイベントカウンタとして使用します	H'FF95 ビット 4	0
	CUEH	イベントカウンタコントロール/ステータスレジスタ (カウントアップイネーブル H) ~ ECH に入力されるイベントクロック入力の許可/禁止を制御。 : CUEH="0" のとき, ECH のイベントクロックの入力を禁止 : CUEH="1" のとき, ECH のイベントクロックの入力を許可	H'FF95 ビット 3	0
	CUEL	イベントカウンタコントロール/ステータスレジスタ (カウントアップイネーブル L) ~ ECL に入力されるイベントクロック入力の許可/禁止を制御。 : CUEL="0" のとき, ECL のイベントクロックの入力を禁止 : CUEL="1" のとき, ECL のイベントクロックの入力を許可	H'FF95 ビット 2	0
	CRCH	イベントカウンタコントロール/ステータスレジスタ (カウンタリセット制御 H) ~ ECH のリセットを制御。 : CRCH="0" のとき, ECH をリセット : CRCH="1" のとき, ECH リセットを解除しカウントアップ機能を許可	H'FF95 ビット 1	0
	CRCL	イベントカウンタコントロール/ステータスレジスタ (カウンタリセット制御 L) ~ ECL のリセットを制御。 : CRCL="0" のとき, ECL をリセット : CRCL="1" のとき, ECL リセットを解除しカウントアップ機能を許可	H'FF95 ビット 0	0
ECH	イベントカウンタ H ~ 8 ビットのリード可能なアップカウンタで, ECL と組み合わせることにより 16 ビットイベントカウンタの上位 8 ビットのアップカウンタとして動作。	H'FF96	H'00	
ECL	イベントカウンタ L ~ 8 ビットのリード可能なアップカウンタで, ECH と組み合わせることにより 16 ビットイベントカウンタの下位 8 ビットのアップカウンタとして動作。	H'FF97	H'00	

表 6 使用内部レジスタ説明(つづき)

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, P40/SCK32 端子を P40 端子に設定 : COM="1"のとき, P40/SCK32 端子を SCK32 出力端子に設定	H'FFA8 ビット 7	0
SCR3	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"で COM="0", PCR40="0"のとき P40/SCK32 端子を P40 入力端子に設定	H'FFAA ビット 1 ビット 0	CKE1=0 CKE0=0
TMA	TMA3	タイマモードレジスタ A(内部クロックセレクト 3) ~ TCA に入力するクロックソースを選択。 : TMA3="0"のとき, TCA の入力クロックソースに PSS を, タイマ A 機能にインターバルタイマ機能を選択 : TMA3="1"のとき, TCA の入力クロックソースに PSW を, タイマ A 機能に時計用タイムベース機能を選択	H'FFB0 ビット 3	1
PMR3	AEVL	ポートモードレジスタ 3(P37/AEVL 端子機能切り替え) ~ P37 端子の機能を設定します。 : AEVL="0"のとき, P37 入出力端子として機能 : AEVL="1"のとき, AEVL 入力端子として機能	H'FFCA ビット 7	1
PDR4	P40	ポートデータレジスタ 4(P40) ~ P40 端子のデータを格納します。 : P40="0"のとき, P40 端子の出力レベルは"Low" : P40="1"のとき, P40 端子の出力レベルは"High"	H'FFD7 ビット 0	0
PCR4	PCR40	ポートコントロールレジスタ 4 (ポートコントロールレジスタ 40) ~ P40 端子の入出力を制御。 : PCR40="0"のとき, P40 端子は入力端子機能 : PCR40="1"のとき, P40 端子は出力端子機能	H'FFE7 ビット 0	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) ~ スタンバイモード, ウォッチモードへの遷移を指定し ます。 : 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, ウォッチモードに遷移	H'FFF0 ビット 7	1

表 6 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
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 2~0) ~ 特定の割込みにより、スタンバイモード、ウォッチ モードを解除し、アクティブモードに遷移する場合に、 クロックが安定するまで CPU と周辺機能が待機する時 間を指定。 : STS2 ~ STS1="000" のとき、待機時間は 8,192 ステート : STS2 ~ STS1="001" のとき、待機時間は 16,384 ステート : STS2 ~ STS1="010" のとき、待機時間は 1,024 ステート : STS2 ~ STS1="011" のとき、待機時間は 2,048 ステート : STS2 ~ STS1="100" のとき、待機時間は 4,096 ステート : STS2 ~ STS1="101" のとき、待機時間は 2 ステート : STS2 ~ STS1="110" のとき、待機時間は 8 ステート : STS2 ~ STS1="111" のとき、待機時間は 16 ステート	H'FFF0 ビット 6 ビット 5 ビット 4	STS2=0 STS1=0 STS0=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) ~ ウォッチモード解除時に、CPU の動作クロックをシス テムクロック()にするか、サブクロック(_{SUB})にす るかを選択。 : LSON=0 のとき、CPU の動作クロックはシステムクロッ ク() : LSON=1 のとき、CPU の動作クロックはサブクロック (_{SUB})	H'FFF0 ビット 3	1
SYSCR2	NESEL	システムコントロールレジスタ 2 (ノイズ除去サンプリング周波数選択) ~ サブクロック発振器より生成されたウォッチクロック (_w)を、システムクロック発振器より生成された OSC クロック (_{osc})によりサンプリングする周波数を選 択。 : NESEL=0 のとき、 _{osc} の 16 分周クロックでサンプリ ング : NESEL=1 のとき、 _{osc} の 4 分周クロックでサンプリ ング	H'FFF1 ビット 4	1

表 6 使用内部レジスタ説明(つづき)

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) ~ 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各モード間を, SLEEP 命令を実 行することにより直接遷移するか否かを指定。 : 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(高速)モードで SLEEP 命 令を実行したとき, アクティブ(中速)モード (SSBY="1", MSON="1", 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, 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, 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", TMA3="1", LSON="0", MSON="1"のとき)に直接遷移	H'FFF1 ビット 3	1
	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) ~ 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)モードで動作させるか, アク ティブ(中速)モードで動作させるかを選択 : MSON="0"のとき, アクティブ(高速)モードで動作 : MSON="1"のとき, アクティブ(中速)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) ~ サブアクティブモードの CPU の動作クロック ($w/8$, $w/4$, $w/2$)を選択。 : SA1="0", SA0="0"のとき, $w/8$ を選択 : SA1="0", SA0="1"のとき, $w/4$ を選択 : SA1="1", SA0="*"のとき, $w/2$ を選択	H'FFF1 ビット 1 ビット 0	SA1=0 SA0=0

表 6 使用内部レジスタ説明(つづき)

レジスタ名		機能	アドレス	設定値
IRR2	IRRDT	割込み要求レジスタ 2(直接遷移割込み要求フラグ) ~ 直接遷移割込み要求の有無を反映。 : IRRDT="0"のとき、直接遷移割込みが要求されていないことを示す : IRRDT="1"のとき、直接遷移割込みが要求されていることを示す	H'FFF7 ビット 7	0
	IRREC	割込み要求レジスタ 2 (非同期イベントカウンタ割込み要求フラグ) ~ 非同期イベントカウンタ割込み要求の有無を反映。 : IRREC="0"のとき、非同期イベントカウンタ割込みが要求されていないことを示す : IRREC="1"のとき、非同期イベントカウンタ割込みが要求されていることを示す	H'FFF7 ビット 0	0
IENR2	IENDT	割込み許可レジスタ 2(直接遷移割込みイネーブル) ~ 直接遷移割込み要求の許可/禁止を制御。 : 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'FFF4 ビット 7	1
	IENEC	割込み許可レジスタ 2 (非同期イベントカウンタ割込みイネーブル) ~ 非同期イベントカウンタ割込み要求の許可/禁止を制御。 : IENEC="0"のとき、非同期イベントカウンタ割込み要求を禁止 : IENEC="1"のとき、非同期イベントカウンタ割込み要求を許可	H'FFF4 ビット 0	1

【注】 *: Don't Care

(4) 使用 RAM 説明

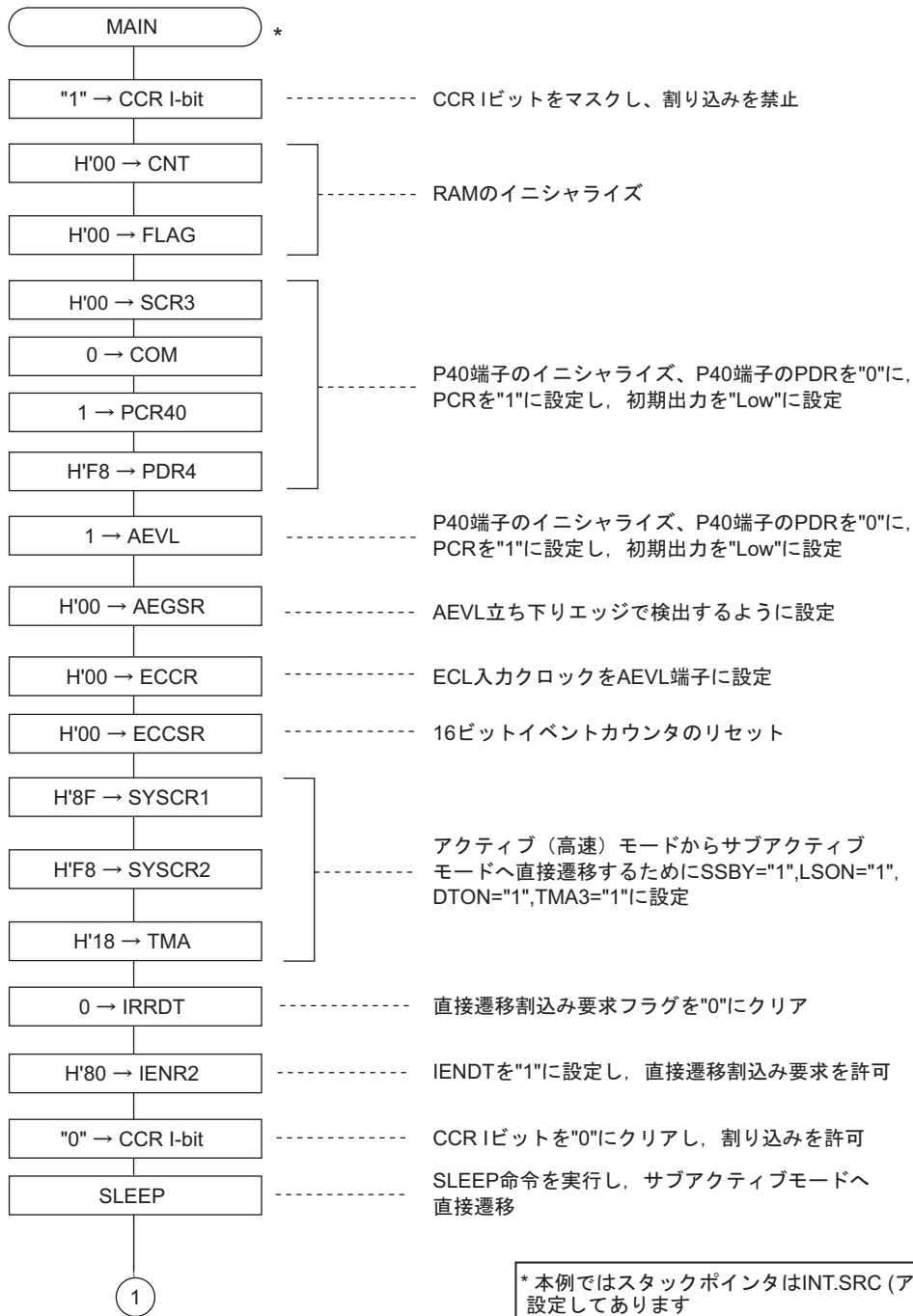
本タスク例の使用 RAM を表 7 に示します。

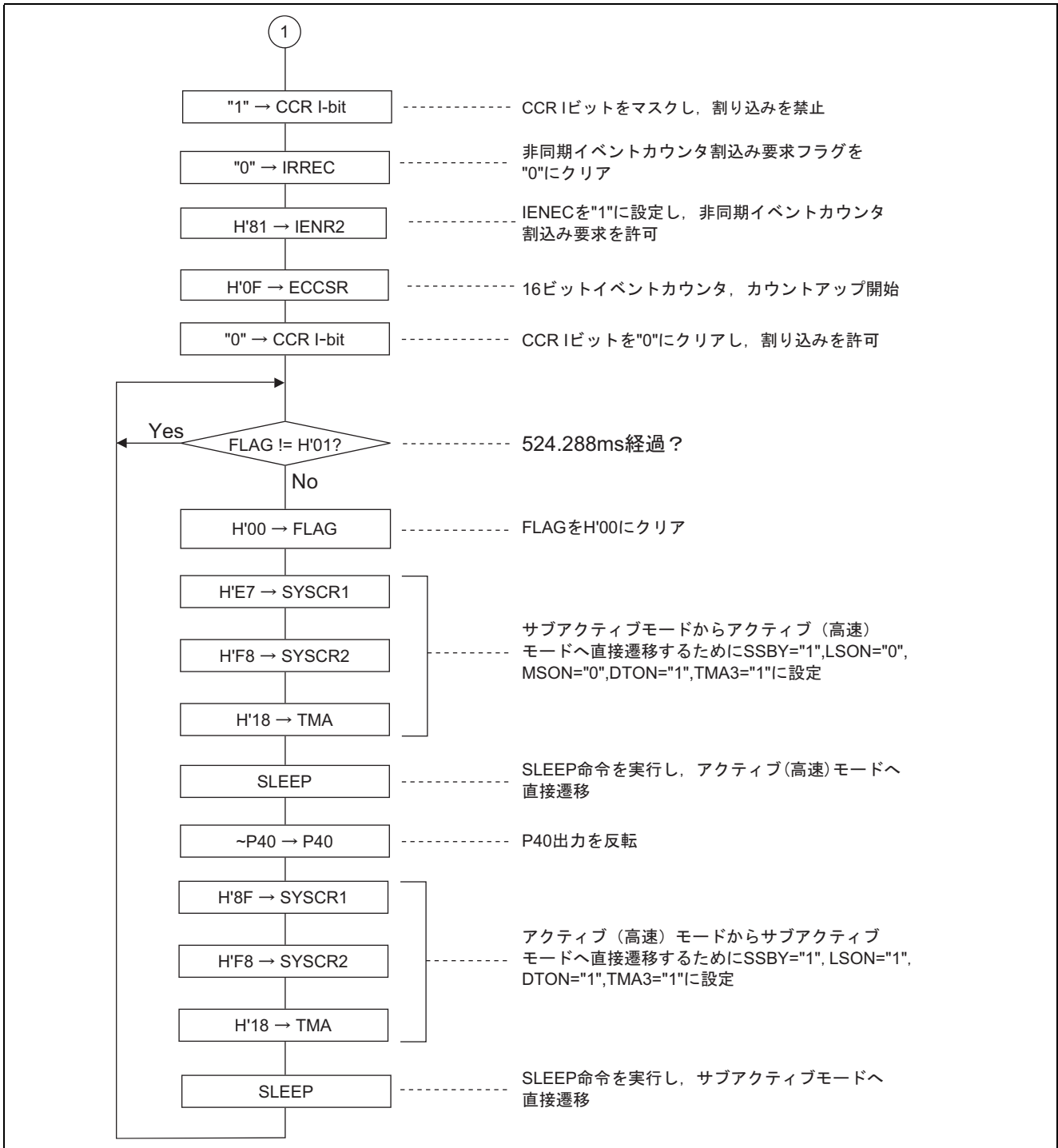
表 7 使用 RAM 説明

ラベル名	機能	アドレス	使用モジュール名
FLAG	524.288ms 経過を示すフラグ	H'FB80	main, aecint
CNT	タイマ F 割込み要求の回数をカウントする 8 ビットカウンタ	H'FB81	main, aecint

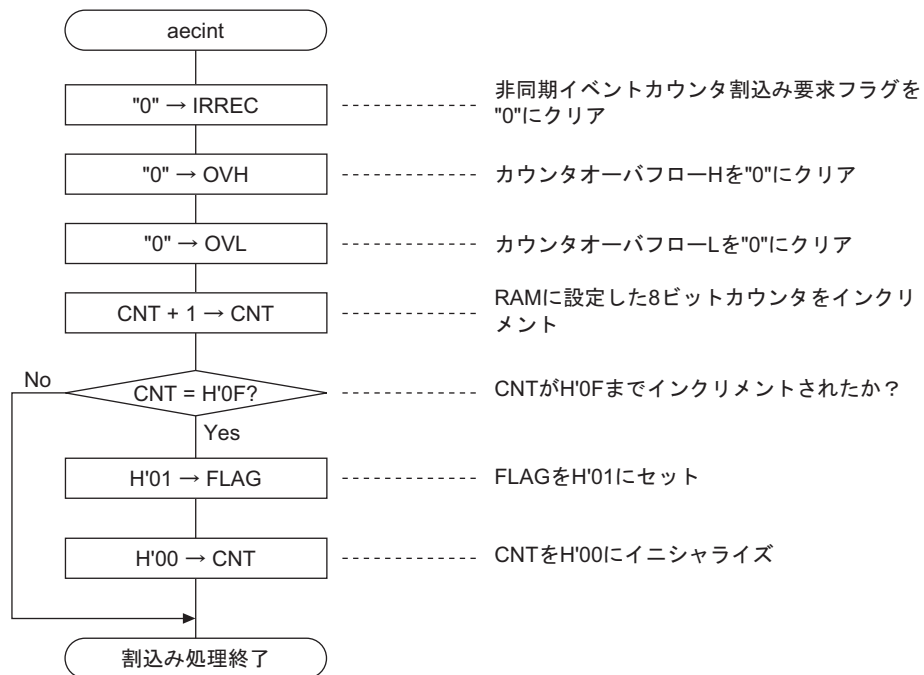
5. フローチャート

(1) メインルーチン

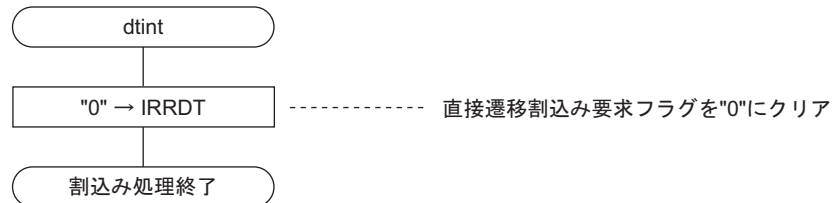




(2) 非同期イベントカウンタ割込み処理ルーチン



(3) 直接遷移割込み処理ルーチン



6. プログラムリスト

6.1 INIT.SRC(プログラムリスト)

```

        .EXPORT  _INIT
        .IMPORT  _main
;
        .SECTION P, CODE
        _INIT:
        MOV.W   #'FF80,R7
        LDC.B   #'10000000,CCR
        JMP     @_main
;
        .END

```

```

/*****/
/*                                     */
/* H8/300L Super Low Power Series      */
/*   -H8/38024 Series-                 */
/* Application Note                     */
/*                                     */
/* 'Asynchronous Event Counter Control' */
/*                                     */
/* Function                             */
/* : AEC(Asynvchronous Event Counter)  */
/*                                     */
/* External Clock : 10MHz               */
/* Internal Clock : 5MHz                */
/* Sub Clock      : 32.768kHz           */
/*                                     */
/*****/

#include <machine.h>

/*****/
/* Symbol Definition                   */
/*****/

struct BIT {
    unsigned char  b7:1;    /* bit7 */
    unsigned char  b6:1;    /* bit6 */
    unsigned char  b5:1;    /* bit5 */
    unsigned char  b4:1;    /* bit4 */
    unsigned char  b3:1;    /* bit3 */
    unsigned char  b2:1;    /* bit2 */
    unsigned char  b1:1;    /* bit1 */
    unsigned char  b0:1;    /* bit0 */
};

#define  AEGSR      *(volatile unsigned char *)0xFF92 /* Input Edge Select Register */
#define  ECCR       *(volatile unsigned char *)0xFF94 /* Event Counter Control Register */
#define  ECCSR      *(volatile unsigned char *)0xFF95 /* Event Counter ControlStatus Register */
#define  ECCSR_BIT  (*(struct BIT *)0xFF95)           /* Event Counter ControlStatus Register */
#define  OVH        ECCSR_BIT.b7                     /* Counter Over Flow H */
#define  OVL        ECCSR_BIT.b6                     /* Counter Over Flow L */
#define  ECH        *(volatile unsigned char *)0xFF96 /* Event Counter H */
#define  ECL        *(volatile unsigned char *)0xFF97 /* Event Counter L */
#define  SMR        *(volatile unsigned char *)0xFFA8 /* Serial Mode Register */
#define  SMR_BIT    (*(struct BIT *)0xFFA8)          /* Serial Mode Register */

```

```

#define COM          SMR_BIT.b7          /* Communication Mode          */
#define CHR          SMR_BIT.b6          /* Character Length            */
#define PE           SMR_BIT.b5          /* Parity Enable               */
#define PM           SMR_BIT.b4          /* Parity Mode                 */
#define STOP         SMR_BIT.b3          /* Stop Bit Length            */
#define MP           SMR_BIT.b2          /* Multiprocessor Mode         */
#define CKS1         SMR_BIT.b1          /* Clock Select 1             */
#define CKS0         SMR_BIT.b0          /* Clock Select 0             */
#define SCR3         *(volatile unsigned char *)0xFFAA /* Serial Control Register 3 */
#define SCR3_BIT     (*(struct BIT *)0xFFAA) /* Serial Control Register 3 */
#define TIE          SCR3_BIT.b7         /* Transmit Interrupt Enable   */
#define RIE          SCR3_BIT.b6         /* Receive Interrupt Enable    */
#define TE           SCR3_BIT.b5         /* Transmit Enable            */
#define RE           SCR3_BIT.b4         /* Receive Enable             */
#define MPIE         SCR3_BIT.b3         /* Multiprocessor Interrupt Enable */
#define TEIE         SCR3_BIT.b2         /* Transmit End Interrupt Enable */
#define CKE1         SCR3_BIT.b1         /* Clock Enable 1             */
#define CKE0         SCR3_BIT.b0         /* Clock Enable 0             */
#define TMA          *(volatile unsigned char *)0xFFB0 /* Timer Mode Register A      */
#define PMR3_BIT     (*(struct BIT *)0xFFCA) /* Port Mode Register 3       */
#define AEVL         PMR3_BIT.b7         /* P37/AEVL Select           */
#define PDR4         *(volatile unsigned char *)0xFFD7 /* Port Data Register 4      */
#define PDR4_BIT     (*(struct BIT *)0xFFD7) /* Port Data Register 4      */
#define P40          PDR4_BIT.b0        /* Port 40                    */
#define PCR4         *(volatile unsigned char *)0xFFE7 /* Port Control Register4    */
#define PCR4_BIT     (*(struct BIT *)0xFFE7) /* Port Control Register4    */
#define PCR40        PCR4_BIT.b0        /* Port Control Register40    */
#define SYSCR1       *(volatile unsigned char *)0xFFF0 /* System Control Register 1 */
#define SYSCR1_BIT   (*(struct BIT *)0xFFF0) /* System Control Register 1 */
#define SSBY         SYSCR1_BIT.b7      /* Software Standby          */
#define STS2         SYSCR1_BIT.b6      /* Standby Timer Select 2    */
#define STS1         SYSCR1_BIT.b5      /* Standby Timer Select 1    */
#define STS0         SYSCR1_BIT.b4      /* Standby Timer Select 0    */
#define LSON         SYSCR1_BIT.b3      /* Low Speed On Flag         */
#define MA1          SYSCR1_BIT.b1      /* Active Mode Clock Select 1 */
#define MA0          SYSCR1_BIT.b0      /* Active Mode Clock Select 0 */
#define SYSCR2       *(volatile unsigned char *)0xFFF1 /* System Control Register 2 */
#define SYSCR2_BIT   (*(struct BIT *)0xFFF1) /* System Control Register 2 */
#define NESEL        SYSCR2_BIT.b4      /* Noise Elimination Sampling Frequency Select */
#define DTON         SYSCR2_BIT.b3      /* Direct Transfer On Flag   */
#define MSON         SYSCR2_BIT.b2      /* Middle Speed On Flag      */
#define SA1          SYSCR2_BIT.b1      /* Subactive Mode Clock Select 1 */
#define SA0          SYSCR2_BIT.b0      /* Subactive Mode Clock Select 0 */
#define IENR2        *(volatile unsigned char *)0xFFF4 /* Interrupt Enable Register 2 */
#define IENR2_BIT    (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 2 */
#define IENDT        IENR2_BIT.b7      /* Timer FH Interrupt Enable  */
#define IENEC        IENR2_BIT.b0      /* Timer FH Interrupt Enable  */
#define IRR2_BIT     (*(struct BIT *)0xFFF7) /* Interrupt Request Register 2 */
#define IRRDT        IRR2_BIT.b7      /* Timer FH Interrupt Request Flag */
#define IRREC        IRR2_BIT.b0      /* Timer FH Interrupt Request Flag */

#pragma interrupt (aecint)
#pragma interrupt (dtint)
/*****
/* Function define
/*****
extern void INIT ( void ); /* SP Set

```

```

void      main ( void );
void      aecint ( void );
void      dtint ( void );

/*****/
/* RAM define */
/*****/
unsigned char  FLAG;          /* User Flag Area */
unsigned char  CNT;          /* User Flag Area */

/*****/
/* Vector Address */
/*****/
#pragma section  V1          /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
    INIT          /* 00 Reset */
};
#pragma section  V2          /* VECTOR SECTOIN SET */
void (*const VEC_TBL2[])(void) = {
    aecint        /* 18 Timer F Interrupt */
};
#pragma section  V3          /* VECTOR SECTOIN SET */
void (*const VEC_TBL3[])(void) = {
    dtint         /* 28 Timer F Interrupt */
};

#pragma section              /* P */
/*****/
/* Main Program */
/*****/
void  main ( void )
{
    set_imask_ccr(1);        /* Interrupt Disable */

    CNT = 0;                /* Initialize 8-bit Counter */
    FLAG = 0;               /* Initialize Event Flag */

    SCR3 = 0;
    COM = 0;
    PCR40 = 1;              /* Initialize P40 Terminall Function */
    PDR4 = 0xF8;           /* Initialize P40 PDR */

    AEVL = 1;
    AEGSR = 0x00;
    ECCR = 0x00;
    ECCSR = 0;              /* Reset 16-bit Event Counter */

    SYSCR1 = 0x8F;          /* Set SYSCR 1 */
    SYSCR2 = 0xF8;          /* Set SYSCR 2 */
    TMA = 0x18;             /* Initialize TCA Overflow Period */

    IRRDT = 0;
    IENR2 = 0x80;          /* Timer A Interrupt Enable */

    set_imask_ccr(0);      /* Interrupt Enable */

    sleep();                /* Transition to Sleep Mode */
}
    
```

```

set_imask_ccr(1);                /* Interrupt Disable          */

IRREC = 0;
IENR2 = 0x81;                   /* Timer A Interrupt Enable   */
ECCSR = 0x0F;

set_imask_ccr(0);                /* Interrupt Enable           */

while(1){
    while (!FLAG);

    FLAG = 0;
    SYSCR1 = 0xE7;
    SYSCR2 = 0xF8;
    TMA = 0x18;                  /* TMA3="1"                   */

    sleep();                     /* Transition to Sleep Mode   */
    P40 = ~P40;

    SYSCR1 = 0x8F;               /* Set SYSCR 1                */
    SYSCR2 = 0xF8;               /* Set SYSCR 2                */
    TMA = 0x18;                  /* TMA3="1"                   */
    sleep();
}
}

/*****/
/* AEC Interrupt                */
/*****/
void aecint ( void )
{
    IRREC = 0;                   /* Clear IRREC                 */

    OVH = 0;                     /* Clear OVH                   */
    OVL = 0;                     /* Clear OVL                   */
    CNT++;                       /* Increment CNT                */

    if ( CNT > 0x0F ){
        FLAG = 1;                /* Set Event Flag              */
        CNT = 0;                 /* Initialize 8-bit Counter    */
    }
}

/*****/
/* Direct Transfer Interrupt     */
/*****/
void dtint ( void )
{
    IRRDT = 0;                   /* Clear IRRDT                 */
}

```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0018
CV3	H'0028
P	H'0100
B	H'FB80

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.12.19	—	初版発行

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留意ください。

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、慎重に制作したものです。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、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、その他お気づ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。