

SH7254R グループ

R01AN1171JJ0100

Rev.1.00

アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2012.05.08

要旨

本アプリケーションノートは、アドバンスドタイマユニット-III (ATU-III) を用いた、タイマ動作例についてまとめたものです。

ATU-III は、各タイマブロックは、それぞれが異なる機能を備えており、互いに独立して動作することができます。また、クロックバスを介して複数のタイマを連動して動作させることもできます。タイマブロックは、同一の機能を持った 1 個以上のタイマサブブロックによって構成され、各サブブロックはさらに 1 個以上のチャンネルを備えています。

本アプリケーションノートに掲載されているタスク例は動作確認済みですが、実際にご使用になる場合には、必ず動作環境を確認の上ご使用くださいますようお願いいたします。

動作確認デバイス

SH72546R

目次

1.	はじめに	3
1.1	仕様	3
1.2	使用機能	3
1.3	適用条件	3
2.	応用例の説明	3
2.1	使用機能の動作概要	3
2.2	動作例 1 パルスの周期測定[タイマ A]	4
2.3	動作例 2 パルスの High 幅測定[タイマ A]	10
2.4	動作例 3 ノイズキャンセル(後続エッジキャンセル)[タイマ A]	17
2.5	動作例 4 ノイズキャンセル(先行エッジキャンセル)[タイマ A]	23
2.6	動作例 5 イベント周期測定[タイマ C]	29
2.7	動作例 6 パルス出力(トグル出力、タイマカウントクリア)[タイマ C]	36
2.8	動作例 7 パルス出力(トグル出力、コンペア値加算)[タイマ C]	42
2.9	動作例 8 パルス出力(High/Low 切替出力、コンペア値加算)[タイマ C]	48
2.10	動作例 9 ワンショットパルス出力(オフセット付)[タイマ D]	55
2.11	動作例 10 ワンショットパルス出力(オフセット、ターミナ付)[タイマ D]	64
2.12	動作例 11 PWM 波形出力[タイマ E]	74
2.13	動作例 12 有効エッジ入力間隔[タイマ F]	81
2.14	動作例 13 一定時間内エッジカウント[タイマ F]	87
2.15	動作例 14 入力 High/Low 期間計測[タイマ F]	93
2.16	動作例 15 PWM 入力波形計測[タイマ F]	99
2.17	動作例 16 回転速度/パルス計測[タイマ F]	105
2.18	動作例 17 アップダウンイベントカウント[タイマ F]	112
2.19	動作例 18 4 遷倍イベントカウント[タイマ F]	118
2.20	動作例 19 コンペアマッチ割り込み発生[タイマ G]	124
2.21	動作例 20 コンペアマッチ発生による他モジュールの起動[タイマ G]	129
2.22	動作例 21 コンペアマッチ発生による他モジュールの起動[タイマ G]	136
2.23	動作例 22 コンペアマッチ割り込み発生[タイマ H]	145
2.24	動作例 23 FIFO レジスタへの値の格納[タイマ J]	151
2.25	動作例 24 FIFO レジスタへの値の格納[タイマ J]	159
2.26	動作例 25 倍周クロックの生成[タイマ B]	167
2.27	動作例 26 倍周クロックを用いた入力信号の欠け歯の検出[タイマ B]	179
2.28	動作例 27 倍周補正クロック信号の生成[タイマ B]	186
2.29	動作例 28 倍周補正クロック信号の生成[タイマ B]	198

1. はじめに

1.1 仕様

本アプリケーションノートでは、SH72546R のアドバンスドタイマユニット-III (ATU-III) の使用方法およびファームウェアの作成例を掲載しています。

1.2 使用機能

- アドバンスドタイマユニット-III (ATU-III)
- ダイレクトメモリアクセスコントローラ (DMAC)
- A/D 変換器 (ADC)
- 割り込みコントローラ (INTC)
- ピンファンクションコントローラ (PFC)

1.3 適用条件

- マイコン : SH72546R
- 動作周波数 : 内部クロック(ϕ) 200MHz
周辺クロック(P ϕ) 40MHz
- 統合開発環境 : ルネサスエレクトロニクス(株)
High-performance Embedded Workshop バージョン=V.4.09.00.007
- C コンパイラ : ルネサスエレクトロニクス(株)
SuperH RISC engine family V.9.04 Release 00
- コンパイルオプション : `-cpu=sh2afpu -object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -gbr=auto
-chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0
-del_vacant_loop=0 -struct_alloc=1 -nologo`

2. 応用例の説明

本応用例では、ATU-IIIのタイマ A~J の各種機能の動作例をそれぞれ纏めたものです。インプットキャプチャ機能などで必要となる外部入力信号は任意に生成する必要が有ります。

2.1 使用機能の動作概要

ATU-III は、タイマ A~J の 9 種類のタイマブロックと、プリスケアラ、および共通制御部から構成されています。各タイマブロックは、それぞれが異なる機能を備えており、互いに独立して動作することができます。また、クロックバスを介して複数のタイマを連動して動作させることもできます。タイマブロックは、同一の機能を持った 1 個以上のタイマサブブロックによって構成され、各サブブロックはさらに 1 個以上のチャンネルを備えています。

2.2 動作例 1 パルスの周期測定[タイマ A]

2.2.1 概要

- 1) 図 2.1に示すようにパルスの周期を測定し、結果を RAM に保存します。
- 2) パルスの周期は次の式で求められます。

$$[\text{パルスの周期(ns)} = \text{タイマ値} \times \text{P}\phi \text{周期(40MHz 動作時 25ns)} \times 2 \text{ (プリスケーラの分周比)}]$$

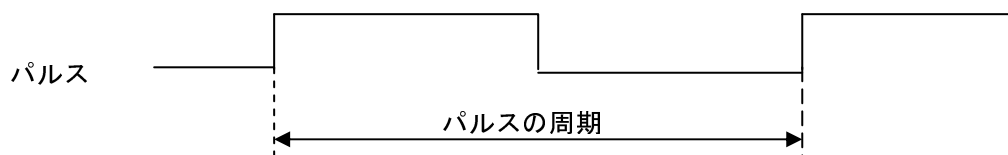


図 2.1 パルス周期測定タイミング

2.2.2 使用機能説明

表 2.1に使用端子及び関連レジスタの機能割り付けを示します。

表 2.1 ATU-III機能割り付け

使用端子		機能
端子	TIA00	測定するパルスはこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケーラの動作を設定します。
	PSCR0	プリスケーラ 0 の分周比を設定します。
タイマ A レジスタ	TIOR1A	外部入力のキャプチャするエッジを設定します。
	TIERA	TIA00 のインプットキャプチャ割り込みを許可/禁止を設定します
	TCNTA	選択したクロックバスでタイマカウントを行います。
	TSRA	インプットキャプチャが発生した場合にフラグがセットされます。
PFC レジスタ	PECR1	端子機能を設定します。
INTC レジスタ	IPR06	TIA00 のインプットキャプチャ割り込みの優先度を設定します。

2.2.3 動作説明

図 2.2に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルスの周期測定を行います。外部入力パルスの立ち上がりに応じてインプットキャプチャ割り込みによるタイマカウントを開始し現在のキャプチャ値を保存します。次の立ち上がりで再度インプットキャプチャによる割り込みで、前回のキャプチャ値と今回のキャプチャ値との差分を取得することで、パルス周期のカウンタ値を取得します。

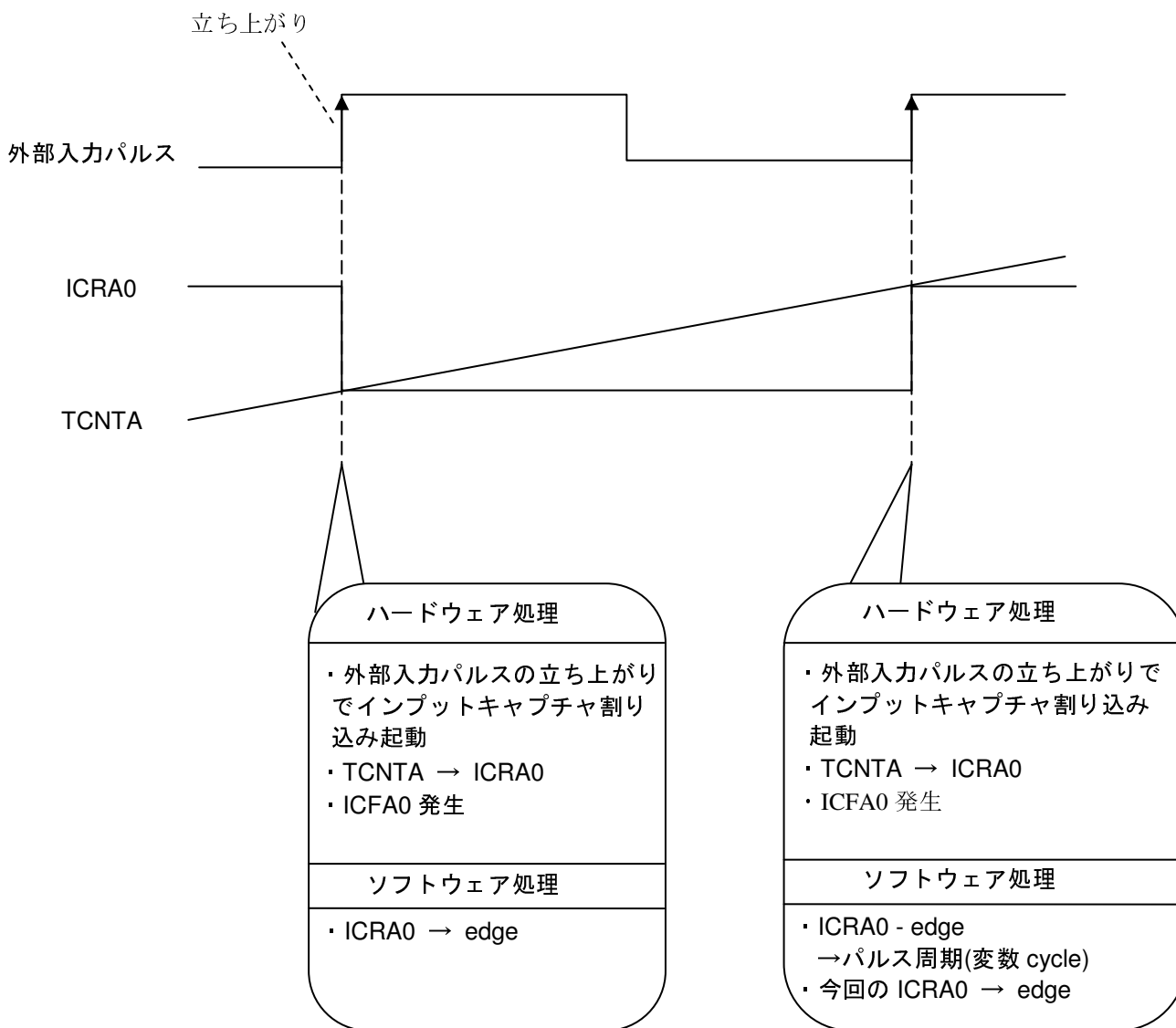


図 2.2 パルス周期計測動作原理

2.2.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
パルスの周期測定	ICIA0	ICFA0によって起動し、ICRA0の値によってパルスの周期を測定します。

● 使用変数の説明

ラベル名	機能	データ長	使用モジュール
cycle	パルスの周期に相当するタイマ値を設定します。 パルスの周期は次の式で求められます。 パルスの周期(ns)=cycleの値×Pφ周期(40MHz動作時25ns)×2(プリスケアラの分周比)	unsigned long	パルスの周期測定
edge	入力キャプチャ発生時のTCNT0の値を保存しません。		

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポートE PE0端子を入力端子TIA00に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ0の分周比を2に設定します。	0x0001	
ATUENR	ATU-IIIタイマA及びプリスケアラのカウント動作を設定します。	0x0003	
TIOR1A	TIA00からの入力信号の立ち上がりエッジで入力キャプチャをするように設定します。	0x0001	
TIERA	TIA00の入力キャプチャ割り込み要求を許可します。	0x01	
ICRA0	測定パルスのエッジ検出時のTCNTAの値を格納します。	0x0000	
IPR06	TIA00の入力キャプチャ割り込み要求の優先度を設定します。	0xF000	
TSRA	タイマAのフラグをクリアします。	0x00	パルスの周期測定
ICRA0	入力キャプチャ信号検出時に、TCNTA0の値を格納します。	-	

2.2.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH725xEVB04-C
 * Description     : This is the main tutorial code.
 * Operation      : Measurement of Pulse period.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 25.08.2011 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long cycle, edge;
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
void main(void)
{
    cycle = edge = 0;          /* 変数初期化 */
                                */

    /* ポートの設定 */
    /* Configure PECCR1
b15:14 PE7MD = 0    PE7 入出力 (ポート)
b13:12 PE6MD = 0    PE6 入出力 (ポート)
b11     Reserved
b10     PE5MD = 0    PE5 入出力 (ポート)
b9      Reserved
b8      PE4MD = 0    PE4 入出力 (ポート)
b7      Reserved
b6      PE3MD = 0    PE3 入出力 (ポート)
b5:4 PE2MD = 0     PE2 入出力 (ポート)
b3:2 PE1MD = 0     PE1 入出力 (ポート)
b1      Reserved
b0      PE0MD = 1    TIA00 入力 (ATU-III) */
PORTE.CR1.WORD = 0x0001;     /* PE0 を TIA00 入力に設定 */
                                */

    /* ATU-III の設定 */
    /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00    TIA のインプットキャプチャ禁止
b9:8 IOA4 = 00     TIA のインプットキャプチャ禁止
b7:6 IOA3 = 00     TIA のインプットキャプチャ禁止
b5:4 IOA2 = 00     TIA のインプットキャプチャ禁止
b3:2 IOA1 = 00     TIA のインプットキャプチャ禁止
b1:0 IOA0 = 01     TIA の立ち上がりで ICRA にキャプチャ */
ATUA.TIOR1A.WORD = 0x0001;   /* TIA00 の立上りエッジでインプットキャプチャ */
                                */

    /* Configure TSRA
b7      OVFA = 0    インプットキャプチャなし
b6      Reserved
b5      ICFA5 = 0   インプットキャプチャなし
b4      ICFA4 = 0   インプットキャプチャなし
b3      ICFA3 = 0   インプットキャプチャなし
b2      ICFA2 = 0   インプットキャプチャなし
b1      ICFA1 = 0   インプットキャプチャなし
b0      ICFA0 = 0   インプットキャプチャなし */
ATUA.TSRA.BYTE &= 0x00;     /* インプットキャプチャフラグをクリア */
                                */

    /* Configure TIERA
b7      OVEA = 0    オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0   インプットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0   インプットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0   インプットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0   インプットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0   インプットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 1   インプットキャプチャ割り込み A0 要求の出力を許可 */
ATUA.TIERA.BYTE = 0x01;     /* TIA00 のインプットキャプチャ割り込みを許可 */
                                */

    /* Configure PSCR0
b15:10 Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b9:0 PSCn = H'1      プリスケーラの分周比を 2 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0001; /* プリスケーラの分周比を 2 に設定 */

/* 割り込み設定 */
/* Configure IPR06
b15:12 ATU-A(ICIA0,1) = H'F 対応する割り込みの優先順位を 15 に設定
b11:8  ATU-A(ICIA2,3) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4  ATU-A(ICIA4,5) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0  ATU-A(OVIA)    = H'0  対応する割り込みの優先順位を 0 に設定 */
INTC.IPR06.WORD = 0xF000; /* TIA00 の割り込み(ICIA0)の優先順位を 15 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0      タイマ J のカウント動作を停止
b8     THE = 0      タイマ H のカウント動作を停止
b7     TGE = 0      タイマ G のカウント動作を停止
b6     TFE = 0      タイマ F のカウント動作を停止
b5     TEE = 0      タイマ E のカウント動作を停止
b4     TDE = 0      タイマ D のカウント動作を停止
b3     TCE = 0      タイマ C のカウント動作を停止
b2     TBE = 0      タイマ B のカウント動作を停止
b1     TAE = 1      タイマ A のカウント動作を許可
b0     PSCE = 1     プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0003; /* タイマAおよびプリスケーラのカウントをスタート */

while(1); /* 無限ループで待機 */
} /* End of function main() */

/*****
* Function Name : ICIA0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value : none
*****/
void ICIA0(void)
{
  /* Configure TSRA
b7     OVFA = 0     インพุットキャプチャフラグをクリア
b6     Reserved
b5     ICFA5 = 0    インพุットキャプチャフラグをクリア
b4     ICFA4 = 0    インพุットキャプチャフラグをクリア
b3     ICFA3 = 0    インพุットキャプチャフラグをクリア
b2     ICFA2 = 0    インพุットキャプチャフラグをクリア
b1     ICFA1 = 0    インพุットキャプチャフラグをクリア
b0     ICFA0 = 0    インพุットキャプチャフラグをクリア */
  ATUA.TSRA.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */
  cycle = ATUA.ICRA0 - edge; /* パルス周期演算 */
  edge = ATUA.ICRA0; /* キャプチャ値保存 */
} /* End of function ICIA0() */
```

2.3 動作例 2 パルスの High 幅測定[タイマ A]

2.3.1 概要

- 1) 図 2.3に示すようにパルスの周期を測定し、結果を RAM に保存します。
- 2) パルスの周期は次の式で求められます。

[パルスの High 幅(ns)=タイマ値×Pφ周期(40MHz 動作時 25ns)×2 (プリスケアラの分周比)]

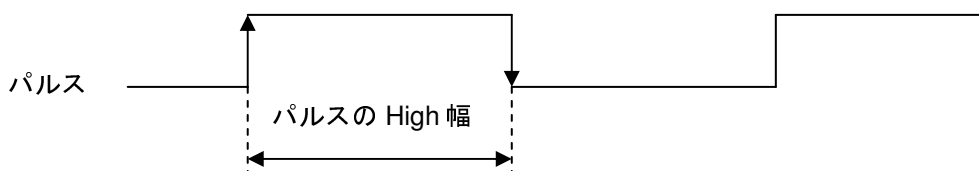


図 2.3 パルスの High 幅計測

2.3.2 使用機能説明

表 2.2に使用端子及び関連レジスタの機能割り付けを示します。

表 2.2 ATU-III機能割り付け

使用端子		機能
端子	TIA00	測定するパルスはこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ A レジスタ	TIOR1A	外部入力のキャプチャ対象エッジを設定します。
	TIERA	TIA00 のインプットキャプチャ割り込みを許可/禁止を設定します
	TCNTA	選択したクロックバスでタイマカウントを行います。
	TSRA	インプットキャプチャが発生した場合にフラグがセットされます。
PFC レジスタ	PECR1	端子機能を設定します。
INTC レジスタ	IPR06	TIA00 のインプットキャプチャ割り込みの優先度を設定します。

2.3.3 動作説明

図 2.4に動作原理を示します。これに従って SH72546 のハードウェアおよびソフトウェア処理によってパルスの周期測定を行います。外部入力パルスの立ち上がりに応じて入力キャプチャ割り込みによるタイマカウントを開始し現在のキャプチャ値を保存します。次は立ち下がりで入力キャプチャによる割り込みを行います。立ち上がり時と立ち下がり時のキャプチャ値の差分を取得することで、パルス High 幅のカウント値を取得します。

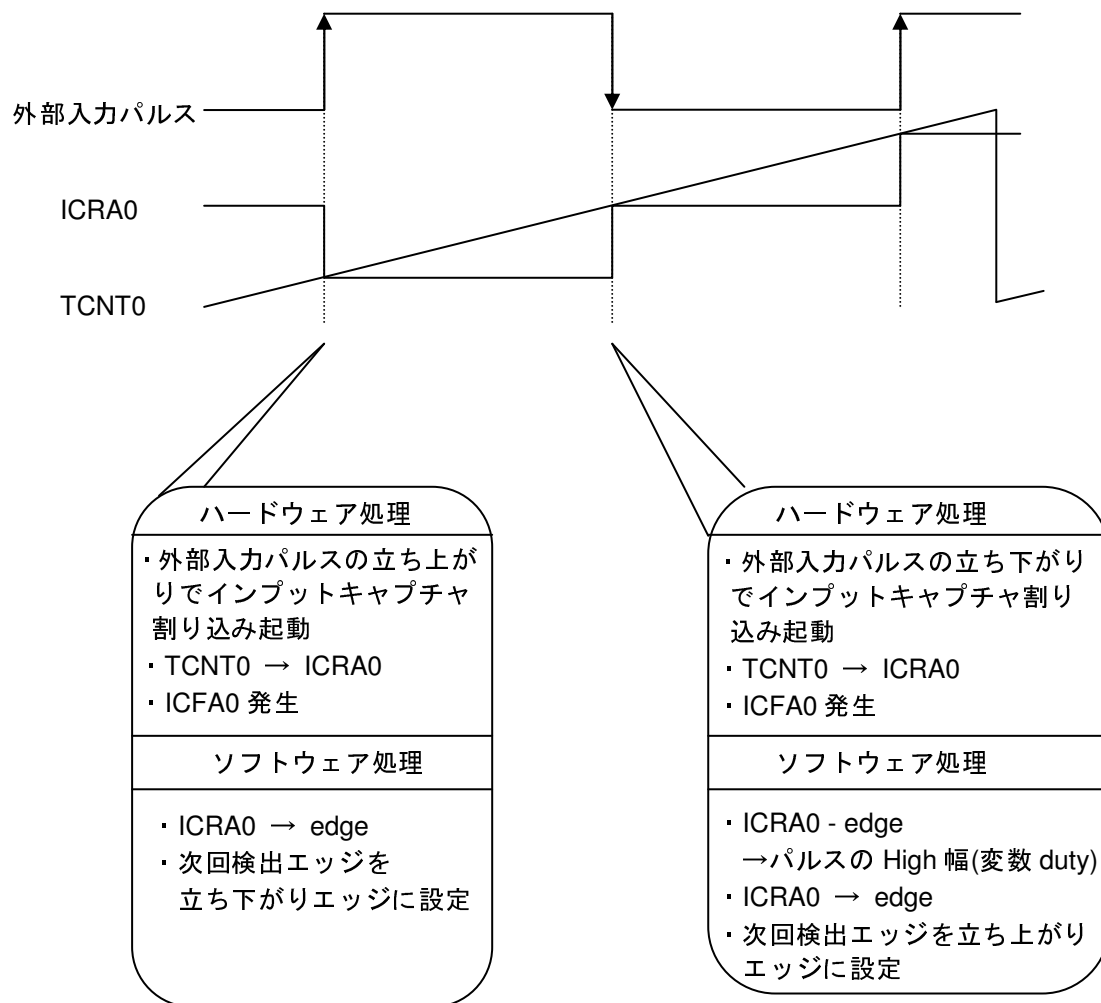


図 2.4 パルスの High 幅計測動作原理

2.3.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
パルスの High 幅測定	ICIA0	ICFA0 によって起動し、ICRA0 の値からパルスの幅を測定します。

● 使用変数の説明

ラベル名	機能	データ長	使用モジュール
duty	パルスの High 幅に相当するタイマ値を設定します。 パルスの High 幅は次の式で求められます。 パルスの High 幅(ns)=duty の値 × Pφ 周期(40MHz 動作時 25ns) × 2 (プリスケアラの分周比)	unsigned long	パルスの周期測定
edge	インプットキャプチャ値を保存します。		

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE0 端子を入力端子 TIA00 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 2 に設定します。	0x0001	
ATUENR	ATU-IIIチャンネル A 及びプリスケアラのカウント動作を設定します。	0x0003	
TIOR1A	TIA00 からの入力信号の立ち上がりエッジでインプットキャプチャをするように設定します。	0x0001	
	TIA00 からの入力信号の立ち下がりエッジでインプットキャプチャをするように設定します。	0x0002	
TIERA	TIA00 のインプットキャプチャ割り込み要求を許可します。	0x01	
ICRA0	測定パルスのエッジ検出時の TCNTA の値を格納します。	0x0000	
IPR06	TIA00 のインプットキャプチャ割り込み要求の優先度を設定します。	0xF000	
TSRA	タイマ A のフラグをクリアします。	0x00	パルスの HIGH 幅測定
TIOR1A	TIA00 からの入力信号の立ち上がり/立ち下がりインプットキャプチャを行うように設定します。 前回の TIOR1A の値で変わります。	0x0001 0x0002	

2.3.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform  : SH725xEVB04-C
* Description    : This is the main tutorial code.
* Operation     : Measurement of Pulse high width.
*****/
/*****
* History : DD.MM.YYYY Version Description
*          : 23.12.2011 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"       /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
unsigned long duty, edge;
void main(void);           /* メインルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value : none
*****/

```

```
void main(void)
{
    duty = edge = 0;          /* 変数初期化 */

    /* ポートの設定 */
    /* Configure PECR1
    b15:14 PE7MD = 0PE7 入出力 (ポート)
    b13:12 PE6MD = 0PE6 入出力 (ポート)
    b11     Reserved
    b10     PE5MD = 0PE5 入出力 (ポート)
    b9      Reserved
    b8      PE4MD = 0PE4 入出力 (ポート)
    b7      Reserved
    b6      PE3MD = 0PE3 入出力 (ポート)
    b5:4 PE2MD = 0PE2 入出力 (ポート)
    b3:2 PE1MD = 0PE1 入出力 (ポート)
    b1      Reserved
    b0      PE0MD = 1TIA00 入力 (ATU-III) */
    PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定 */

    /* ATU-III の設定 */
    /* Configure TIOR1A
    b15:12 Reserved
    b11:10 IOA5 = 0 TIA のインプットキャプチャ禁止
    b9:8 IOA4 = 0 TIA のインプットキャプチャ禁止
    b7:6 IOA3 = 0 TIA のインプットキャプチャ禁止
    b5:4 IOA2 = 0 TIA のインプットキャプチャ禁止
    b3:2 IOA1 = 0 TIA のインプットキャプチャ禁止
    b1:0 IOA0 = 1 TIA の立ち上がりで ICRA にキャプチャ */
    ATUA.TIOR1A.WORD = 0x0001; /* TIA00 の立ち上がりエッジで ICRA にキャプチャ */

    /* Configure TSRA
    b7     OVFA = 0 インプットキャプチャフラグをクリア
    b6     Reserved
    b5     ICFA5 = 0 インプットキャプチャフラグをクリア
    b4     ICFA4 = 0 インプットキャプチャフラグをクリア
    b3     ICFA3 = 0 インプットキャプチャフラグをクリア
    b2     ICFA2 = 0 インプットキャプチャフラグをクリア
    b1     ICFA1 = 0 インプットキャプチャフラグをクリア
    b0     ICFA0 = 0 インプットキャプチャフラグをクリア */
    ATUA.TSRA.BYTE &= 0x00; /* インプットキャプチャフラグをクリア */

    /* Configure TIERA
    b7     OVEA = 0 オーバフロー割り込み A 要求の出力を禁止
    b6     Reserved
    b5     ICEA5 = 0 インプットキャプチャ割り込み A5 要求の出力を禁止
    b4     ICEA4 = 0 インプットキャプチャ割り込み A4 要求の出力を禁止
    b3     ICEA3 = 0 インプットキャプチャ割り込み A3 要求の出力を禁止
    b2     ICEA2 = 0 インプットキャプチャ割り込み A2 要求の出力を禁止
    b1     ICEA1 = 0 インプットキャプチャ割り込み A1 要求の出力を禁止
    b0     ICEA0 = 1 インプットキャプチャ割り込み A0 要求の出力を許可 */
    ATUA.TIERA.BYTE = 0x01; /* TIA00 のインプットキャプチャ割り込みを許可 */

    /* Configure PSCR0
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:10 Reserved
b9:0 PSCn = H'1 プリスケーラの分周比を 2 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0001; /* プリスケーラの分周比を 2 に設定 */

/* 割り込み設定 */
/* Configure IPR06
b15:12 ATU-A(ICIA0,1) = H'F 対応する割り込みの優先順位を 15 に設定
b11:8 ATU-A(ICIA2,3) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4 ATU-A(ICIA4,5) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 ATU-A(OVIA) = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR06.WORD = 0xF000; /* TIA00 の割り込み(ICIA0)の優先順位を 15 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9 TJE = 0 タイマ J のカウント動作を停止
b8 THE = 0 タイマ H のカウント動作を停止
b7 TGE = 0 タイマ G のカウント動作を停止
b6 TFE = 0 タイマ F のカウント動作を停止
b5 TEE = 0 タイマ E のカウント動作を停止
b4 TDE = 0 タイマ D のカウント動作を停止
b3 TCE = 0 タイマ C のカウント動作を停止
b2 TBE = 0 タイマ B のカウント動作を停止
b1 TAE = 1 タイマ A のカウント動作を許可
b0 PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0003; /* タイマ A およびプリスケーラのカウントをスタート */

while(1); /* 無限ループで待機 */
} /* End of function main() */

/*****
* Function Name : ICIA0
* Description : インพุットキャプチャ割り込み
* Arguments : none
* Return Value : none
*****/
void ICIA0(void) /* インพุットキャプチャ割り込み */
{
/* Configure TSRA
b7 OVFA = 0 インพุットキャプチャフラグをクリア
b6 Reserved
b5 ICFA5 = 0 インพุットキャプチャフラグをクリア
b4 ICFA4 = 0 インพุットキャプチャフラグをクリア
b3 ICFA3 = 0 インพุットキャプチャフラグをクリア
b2 ICFA2 = 0 インพุットキャプチャフラグをクリア
b1 ICFA1 = 0 インพุットキャプチャフラグをクリア
b0 ICFA0 = 0 インพุットキャプチャフラグをクリア */
ATUA.TSRA.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */

/* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00 TIA のインพุットキャプチャ禁止
b9:8 IOA4 = 00 TIA のインพุットキャプチャ禁止
b7:6 IOA3 = 00 TIA のインพุットキャプチャ禁止
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b5:4 IOA2 = 00    TIA のインプットキャプチャ禁止
b3:2 IOA1 = 00    TIA のインプットキャプチャ禁止
b1:0 IOA0 = 01    TIA の立ち上がりで ICRA にキャプチャ  */
if(ATUA.TIOR1A.WORD == 0x0001) /* 今回読み込んだ値が立ち上がりエッジならば  */
{
    edge = ATUA.ICRA0; /* インプットキャプチャが発生した時点でのカウントの値を保存  */

    /* Configure TIOR1A
    b15:12Reserved
    b11:10IOA5 = 00TIA のインプットキャプチャ禁止
    b9:8 IOA4 = 00TIA のインプットキャプチャ禁止
    b7:6 IOA3 = 00TIA のインプットキャプチャ禁止
    b5:4 IOA2 = 00TIA のインプットキャプチャ禁止
    b3:2 IOA1 = 00TIA のインプットキャプチャ禁止
    b1:0 IOA0 = 10TIA の立ち下がりがりで ICRA にキャプチャ  */
    ATUA.TIOR1A.WORD = 0x0002; /* 次回読み込むのは立下がりエッジ  */
}
else /* 今回読み込んだ値が立下がりエッジならば  */
{
    duty = ATUA.ICRA0 - edge; /* 今回のインプットキャプチャ時のカウントの値から
                               前回のインプットキャプチャ時の値を引き、Low 期間の値を求める*/

    /* Configure TIOR1A
    b15:12Reserved
    b11:10IOA5 = 00TIA のインプットキャプチャ禁止
    b9:8 IOA4 = 00TIA のインプットキャプチャ禁止
    b7:6 IOA3 = 00TIA のインプットキャプチャ禁止
    b5:4 IOA2 = 00TIA のインプットキャプチャ禁止
    b3:2 IOA1 = 00TIA のインプットキャプチャ禁止
    b1:0 IOA0 = 01TIA の立ち上がりがりで ICRA にキャプチャ  */
    ATUA.TIOR1A.WORD = 0x0001; /* 次回読み込むのは立ち上がりエッジ  */
}
} /* End of function ICIA0() */
```

2.4 動作例 3 ノイズキャンセル(後続エッジキャンセル)[タイマ A]

2.4.1 概要

図 2.5に示すように後続エッジキャンセル機能は、入力される信号に混ざるノイズを除去する為に、入力から一定期間内のレベル変化を無視する機能です。タイマとのコンペアマッチでノイズキャンセル期間を制御します。

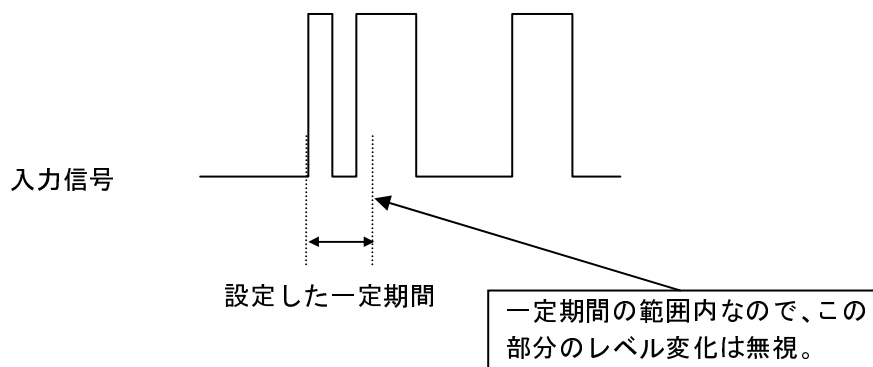


図 2.5 後続ノイズキャンセルの概念図

2.4.2 使用機能説明

表 2.3に使用端子及び関連レジスタの機能割り付けを示します。

表 2.3 ATU-III機能割り付け

使用端子		機能
端子	TIA00	測定する入力信号をこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケーラの動作を設定します。
	PSCR0	プリスケーラ0の分周比を設定します。
	NCMR	各タイマのノイズキャンセル動作モードおよびカウントクロックを設定します。
タイマA レジスタ	TIOR1A	外部入力のキャプチャするエッジを設定します。
	TIOR2A	ノイズキャンセルに関する内容を設定します。
	NCNTA	エッジ入力と同時にカウントを開始し、ノイズキャンセルの期間を測定します。
	NCRA0	ノイズキャンセルの期間を設定します。
PFC レジスタ	PECR1	端子機能を設定します。

2.4.3 動作説明

図 2.6に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって後続ノイズキャンセルを行います。ノイズキャンセル後の信号は、タイマカウントの開始後はコンペアマッチが発生するまでは更新されません。(図は立ち上がりエッジを検出する場合の例)

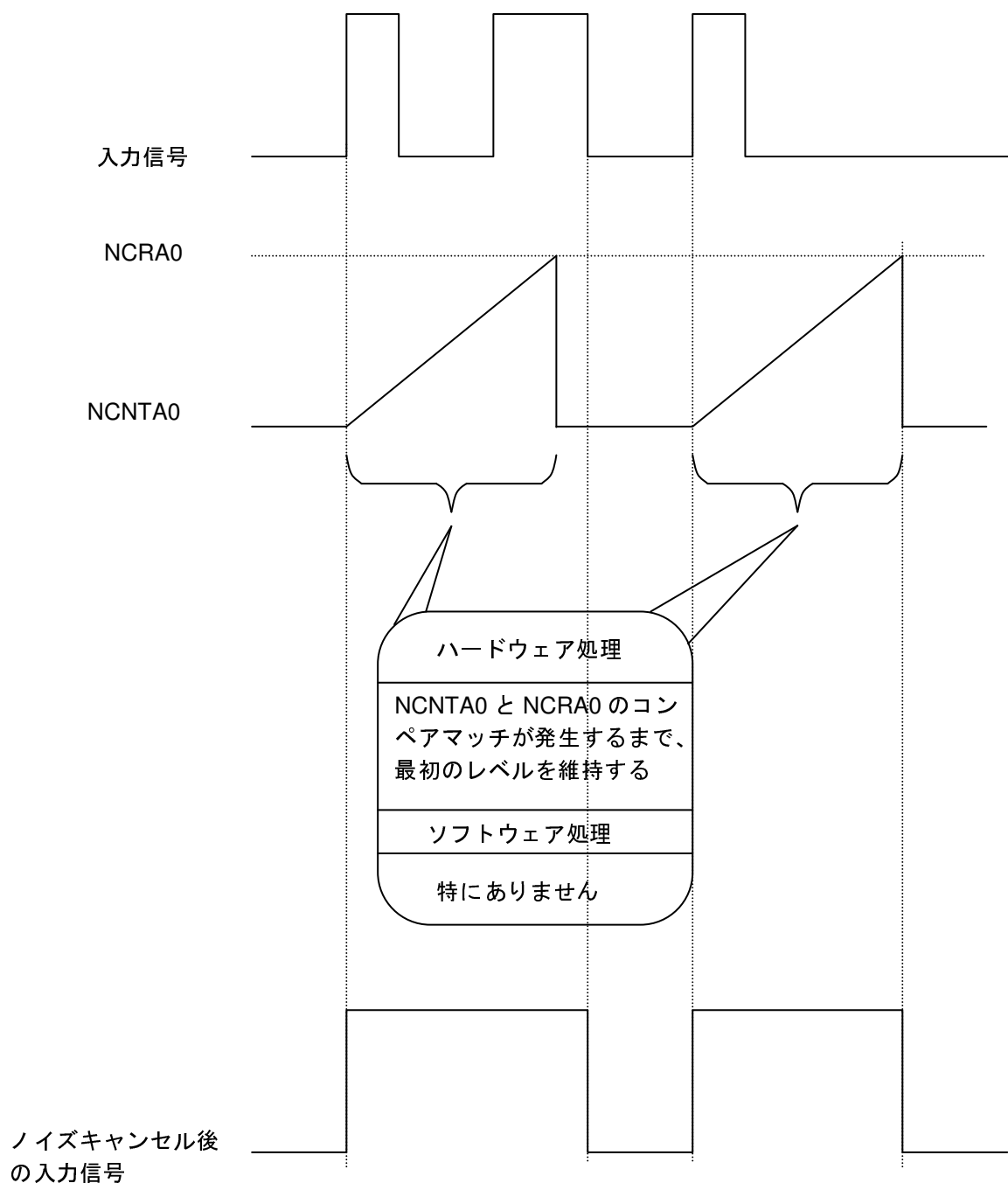


図 2.6 後続ノイズキャンセル動作原理

2.4.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE0 端子を入力端子 TIA00 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 2 に設定します。	0x0001	
ATUENR	ATU-IIIチャンネル A 及びプリスケアラのカウンタ動作を設定します。	0x0003	
NCMR	タイマ A のノイズキャンセル機能を後続ノイズキャンセルに設定します。	0x00	
TIOR1A	TIA00 からの入力信号の立ち上がりエッジでインプットキャプチャをするように設定します。	0x0001	
TIOR2A	ノイズキャンセルカウンタのクロックバス及び動作許可を設定します。	0x0001	
NCRA0	ノイズキャンセルカウンタの上限を 0x12 に設定します。	0x12	

2.4.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH725xEVB04-C
 * Description     : This is the main tutorial code.
 * Operation      : Subsequent edge noise cancellation.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 23.12.2011 1.00 First Release
 */

/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PECR1
  b15:14 PE7MD = 0PE7 入出力 (ポート)
  b13:12 PE6MD = 0PE6 入出力 (ポート)
  b11     Reserved
  b10     PE5MD = 0PE5 入出力 (ポート)
  b9      Reserved
  b8      PE4MD = 0PE4 入出力 (ポート)
  b7      Reserved
  b6      PE3MD = 0PE3 入出力 (ポート)
  b5:4 PE2MD = 0PE2 入出力 (ポート)
  b3:2 PE1MD = 0PE1 入出力 (ポート)
  b1      Reserved
  b0      PE0MD = 1TIA00 入力 (ATU-III) */
  PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定 */

  /* ATU-III の設定 */
  /* Configure TIOR1A
  b15:12 Reserved
  b11:10 IOA5 = 0 TIA のインプットキャプチャ禁止
  b9:8 IOA4 = 0 TIA のインプットキャプチャ禁止
  b7:6 IOA3 = 0 TIA のインプットキャプチャ禁止
  b5:4 IOA2 = 0 TIA のインプットキャプチャ禁止
  b3:2 IOA1 = 0 TIA のインプットキャプチャ禁止
  b1:0 IOA0 = 1 TIA の立ち上がりで ICRA にキャプチャ */
  ATUA.TIOR1A.WORD = 0x0001; /* TIA00 の立ち上がりエッジで ICRA にキャプチャ */

  /* Configure TSRA
  b7      OVFA = 0 インプットキャプチャフラグをクリア
  b6      Reserved
  b5      ICFA5 = 0 インプットキャプチャフラグをクリア
  b4      ICFA4 = 0 インプットキャプチャフラグをクリア
  b3      ICFA3 = 0 インプットキャプチャフラグをクリア
  b2      ICFA2 = 0 インプットキャプチャフラグをクリア
  b1      ICFA1 = 0 インプットキャプチャフラグをクリア
  b0      ICFA0 = 0 インプットキャプチャフラグをクリア */
  ATUA.TSRA.BYTE &= 0x00; /* インプットキャプチャフラグをクリア */

  /* Configure NCMR
  b7      NCCSEL = 0 内部周辺クロックの 128 分周でカウントアップ
  b6:4 Reserved
  b3      NCMJ = 0 タイマ J 後続エッジキャンセルモード
  b2      NCMF = 0 タイマ F 後続エッジキャンセルモード
  b1      NCMC = 0 タイマ C 後続エッジキャンセルモード
  b0      NCMA = 0 タイマ A 後続エッジキャンセルモード */
  ATUCTRL.NCMR.BYTE = 0x00; /* タイマ A のノイズキャンセラを後続エッジキャンセルに設定 */

  /* Configure TIOR2A
  b15:14 Reserved
  b13     NCKA5 = 0NCNTA5 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b12     NCKA4 = 0NCNTA4 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b11     NCKA3 = 0NCNTA3 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b10     NCKA2 = 0NCNTA2 のカウントソースクロックにノイズキャンセラカウントクロックを選択
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b9      NCKA1 = 0NCNTA1 のカウントソースクロックにノイズキャンセラカウントクロックを選択
b8      NCKA0 = 0NCNTA0 のカウントソースクロックにノイズキャンセラカウントクロックを選択
b7:6 Reserved
b5      NCEA5 = 0TIA05 入力のノイズキャンセル機能を無効
b4      NCEA4 = 0TIA04 入力のノイズキャンセル機能を無効
b3      NCEA3 = 0TIA03 入力のノイズキャンセル機能を無効
b2      NCEA2 = 0TIA02 入力のノイズキャンセル機能を無効
b1      NCEA1 = 0TIA01 入力のノイズキャンセル機能を無効
b0      NCEA0 = 1TIA00 入力のノイズキャンセル機能を有効 */
ATUA.TIOR2A.WORD = 0x0001; /* TIA00 のノイズキャンセラを有効に設定
                           カウンタのクロックをクロックバス 0 に設定 */

/* Configure NCRA0
b7:0 NCCNTAn = H'12 ノイズキャンセルタイムの上限値設定 */
ATUA.NCRA0 = 0x12; /* TIA00 のノイズキャンセルカウンタの上限を 18 に設定 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'1 プリスケーラの分周比を 2 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0001; /* プリスケーラの分周比を 2 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0 タイマ J のカウント動作を停止
b8      THE = 0 タイマ H のカウント動作を停止
b7      TGE = 0 タイマ G のカウント動作を停止
b6      TFE = 0 タイマ F のカウント動作を停止
b5      TEE = 0 タイマ E のカウント動作を停止
b4      TDE = 0 タイマ D のカウント動作を停止
b3      TCE = 0 タイマ C のカウント動作を停止
b2      TBE = 0 タイマ B のカウント動作を停止
b1      TAE = 1 タイマ A のカウント動作を許可
b0      PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0003; /* タイマ A およびプリスケーラのカウントをスタート */

while(1);
} /* End of function main() */
```

2.5 動作例 4 ノイズキャンセル(先行エッジキャンセル)[タイマ A]

2.5.1 概要

図 2.7に示すように先行エッジキャンセル機能は、入力される信号に混ざるノイズを除去するために、期間が一定未満のレベル変化を無視する機能です。

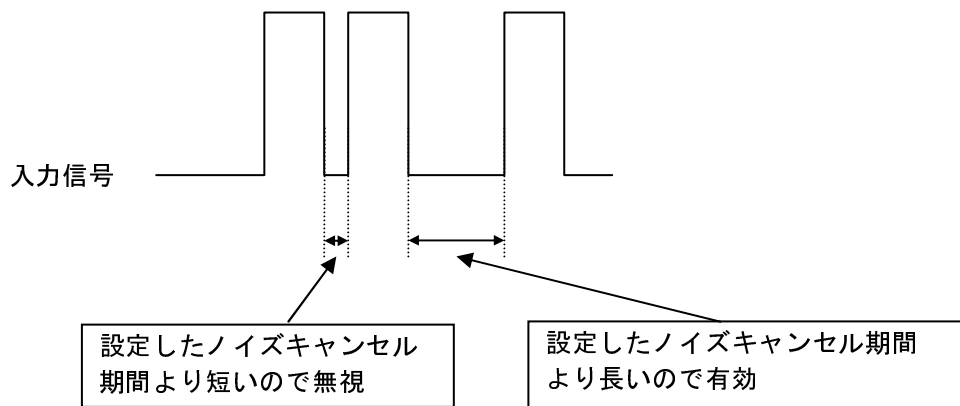


図 2.7 ノイズキャンセルの概念図

2.5.2 使用機能説明

表 2.4に使用端子及び関連レジスタの機能割り付けを示します。

表 2.4 ATU-III機能割り付け

使用端子		機能
端子	TIA0	測定する入力信号をこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ0の分周比を設定します。
	NCMR	各タイマのノイズキャンセル動作モードおよびカウントクロックを設定します。
タイマ A レジスタ	TIOR1A	外部入力のキャプチャするエッジを設定します。
	TIOR2A	ノイズキャンセルに関する内容を設定します。
	NCNTA	エッジ入力と同時にカウントを開始し、ノイズキャンセルの期間を測定します。
	NCRA0	ノイズキャンセルの期間を設定します。

2.5.3 動作説明

図 2.8に動作原理を示します。信号レベルが指定期間以上 High でないものをノイズと見なし無視する為に、コンペアマッチ発生後の入力信号をノイズキャンセル済の信号として取り込みます。

これに従って SH72546R のハードウェアおよびソフトウェア処理によって先行ノイズキャンセル動作を行います。（図は立ち上がりエッジを検出する場合の例）

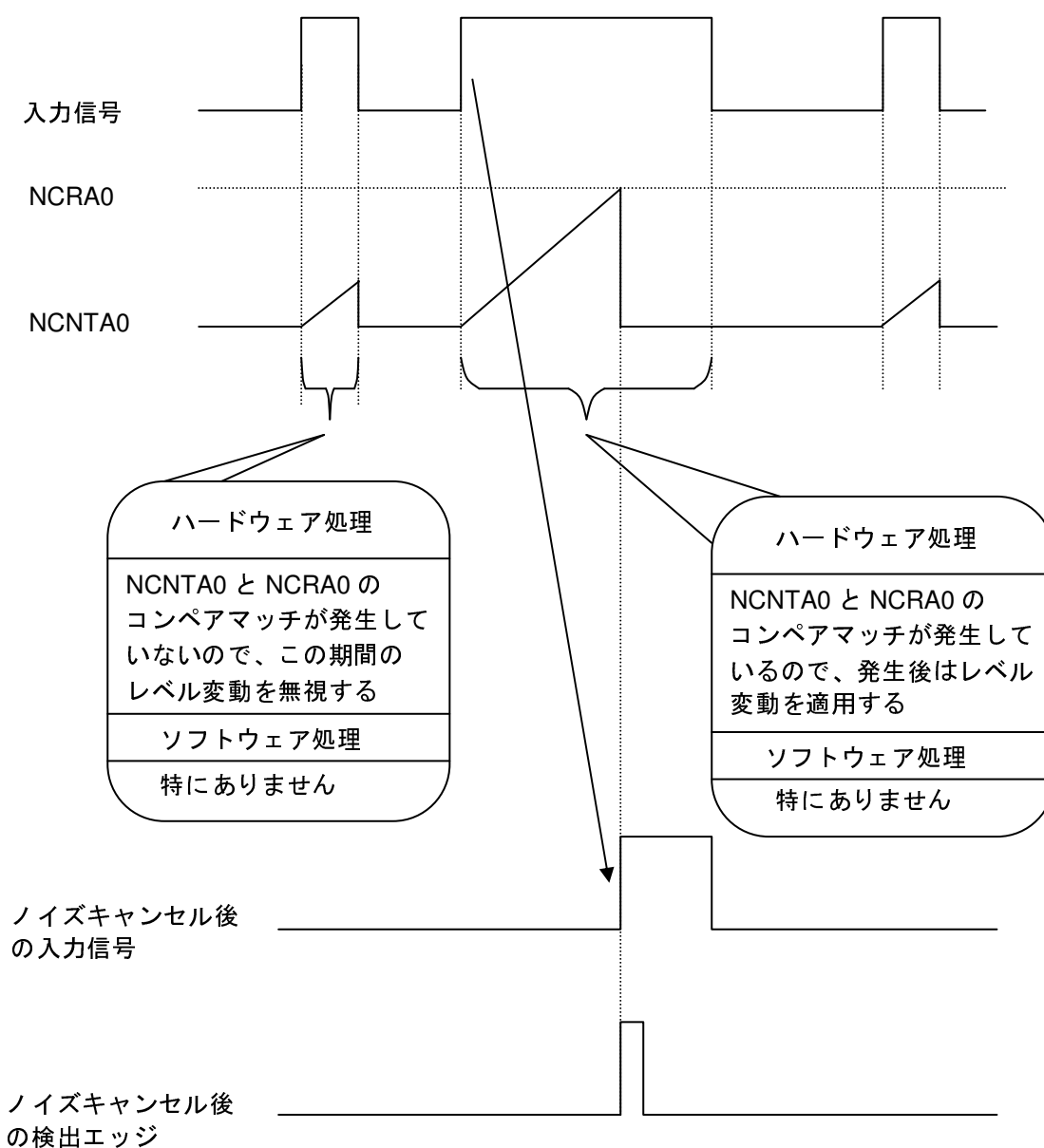


図 2.8 先行ノイズキャンセル動作原理

2.5.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE0 端子を入力端子 TIA00 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 2 に設定します。	0x0001	
ATUENR	ATU-IIIチャンネル A 及びプリスケアラのカウント動作を設定します。	0x0003	
NCMR	タイマ A のノイズキャンセル機能を先行ノイズキャンセルに設定します。	0x01	
TIOR1A	TIA00 からの入力信号の立ち上がりエッジでインプットキャプチャをするように設定します。※	0x0001	
TIOR2A	ノイズキャンセルカウンタのクロックバス及び動作許可を設定します。	0x0001	
NCRA0	ノイズキャンセルカウンタの上限を 0x12 に設定します	0x12	

※キャプチャ後のレベル変動が監視されるので、このレジスタでノイズキャンセルの対象になるレベルを設定します。

2.5.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Leading-edge noise cancellation.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PECR1
  b15:14 PE7MD = 0PE7 入出力 (ポート)
  b13:12 PE6MD = 0PE6 入出力 (ポート)
  b11     Reserved
  b10     PE5MD = 0PE5 入出力 (ポート)
  b9      Reserved
  b8      PE4MD = 0PE4 入出力 (ポート)
  b7      Reserved
  b6      PE3MD = 0PE3 入出力 (ポート)
  b5:4 PE2MD = 0PE2 入出力 (ポート)
  b3:2 PE1MD = 0PE1 入出力 (ポート)
  b1      Reserved
  b0      PE0MD = 1TIA00 入力 (ATU-III) */
  PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定 */

  /* ATU-IIIの設定 */
  /* Configure TIOR1A
  b15:12 Reserved
  b11:10 IOA5 = 0 TIA のインプットキャプチャ禁止
  b9:8 IOA4 = 0 TIA のインプットキャプチャ禁止
  b7:6 IOA3 = 0 TIA のインプットキャプチャ禁止
  b5:4 IOA2 = 0 TIA のインプットキャプチャ禁止
  b3:2 IOA1 = 0 TIA のインプットキャプチャ禁止
  b1:0 IOA0 = 1 TIA の立ち上がりで ICRA にキャプチャ */
  ATUA.TIOR1A.WORD = 0x0001; /* TIA00 の立ち上がりエッジで ICRA にキャプチャ */
  /* Configure TSRA
  b7      OVFA = 0 インプットキャプチャフラグをクリア
  b6      Reserved
  b5      ICFA5 = 0 インプットキャプチャフラグをクリア
  b4      ICFA4 = 0 インプットキャプチャフラグをクリア
  b3      ICFA3 = 0 インプットキャプチャフラグをクリア
  b2      ICFA2 = 0 インプットキャプチャフラグをクリア
  b1      ICFA1 = 0 インプットキャプチャフラグをクリア
  b0      ICFA0 = 0 インプットキャプチャフラグをクリア */
  ATUA.TSRA.BYTE &= 0x00; /* インプットキャプチャフラグをクリア */

  /* Configure NCMR
  b7      NCCSEL = 0 内部周辺クロックの 128 分周でカウントアップ
  b6:4 Reserved
  b3      NCMJ = 0 タイマ J 後続エッジキャンセルモード
  b2      NCMF = 0 タイマ F 後続エッジキャンセルモード
  b1      NCMC = 0 タイマ C 後続エッジキャンセルモード
  b0      NCMA = 1 タイマ A 先行エッジキャンセルモード */
  ATUCTRL.NCMR.BYTE = 0x01; /* タイマ A のノイズキャンセラを先行エッジキャンセルに設定 */

  /* Configure TIOR2A
  b15:14 Reserved
  b13     NCKA5 = 0NCNTA5 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b12     NCKA4 = 0NCNTA4 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b11     NCKA3 = 0NCNTA3 のカウントソースクロックにノイズキャンセラカウントクロックを選択
  b10     NCKA2 = 0NCNTA2 のカウントソースクロックにノイズキャンセラカウントクロックを選択
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b9      NCKA1 = 0NCNTA1 のカウントソースクロックにノイズキャンセラカウントクロックを選択
b8      NCKA0 = 0NCNTA0 のカウントソースクロックにノイズキャンセラカウントクロックを選択
b7:6 Reserved
b5      NCEA5 = 0TIA05 入力のノイズキャンセル機能を無効
b4      NCEA4 = 0TIA04 入力のノイズキャンセル機能を無効
b3      NCEA3 = 0TIA03 入力のノイズキャンセル機能を無効
b2      NCEA2 = 0TIA02 入力のノイズキャンセル機能を無効
b1      NCEA1 = 0TIA01 入力のノイズキャンセル機能を無効
b0      NCEA0 = 1TIA00 入力のノイズキャンセル機能を有効 */
ATUA.TIOR2A.WORD = 0x0001; /* TIA00 のノイズキャンセラを有効に設定
                           カウンタのクロックをクロックバス 0 に設定 */

/* Configure NCRA0
b7:0 NCCNTAn = H'12 ノイズキャンセルタイムの上限値設定 */
ATUA.NCRA0 = 0x12; /* TIA00 のノイズキャンセルカウンタの上限を 18 に設定 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'1 プリスケーラの分周比を 2 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0001; /* プリスケーラの分周比を 2 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0 タイマ J のカウント動作を停止
b8      THE = 0 タイマ H のカウント動作を停止
b7      TGE = 0 タイマ G のカウント動作を停止
b6      TFE = 0 タイマ F のカウント動作を停止
b5      TEE = 0 タイマ E のカウント動作を停止
b4      TDE = 0 タイマ D のカウント動作を停止
b3      TCE = 0 タイマ C のカウント動作を停止
b2      TBE = 0 タイマ B のカウント動作を停止
b1      TAE = 1 タイマ A のカウント動作を許可
b0      PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0003; /* タイマ A およびプリスケーラのカウントをスタート */

while(1);
} /* End of function main() */
```

2.6 動作例 5 イベント周期測定[タイマ C]

2.6.1 概要

図 2.9に示すように、イベントの周期を測定し結果を RAM に格納します。外部からのイベントトリガを 2 回受けて、その間のカウンタ値を記録することでイベント間の周期を測定します。

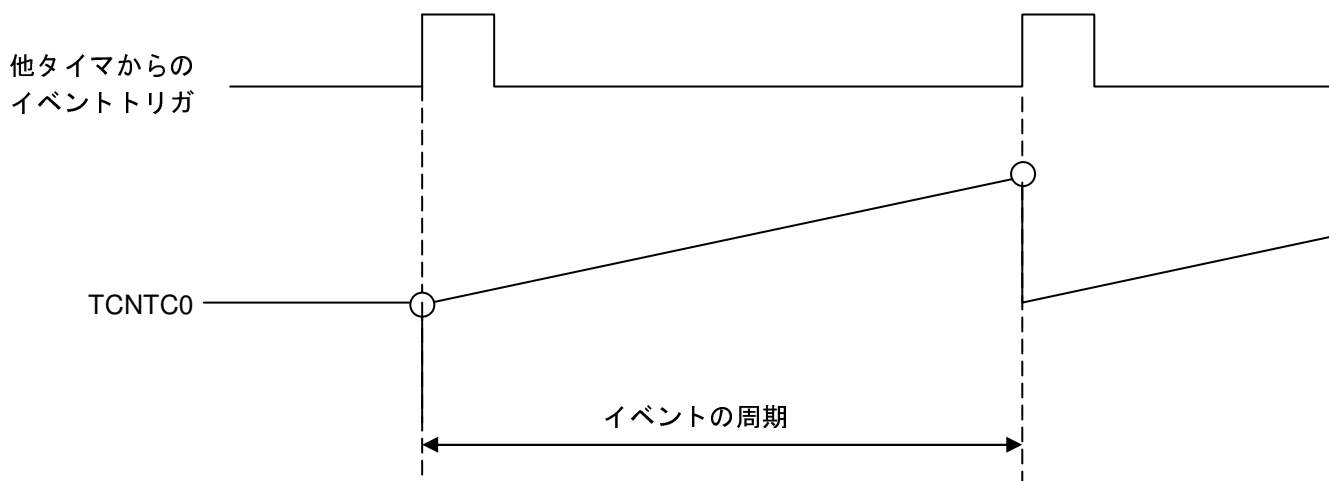


図 2.9 イベント周期測定タイミング

2.6.2 使用機能説明

表 1 に使用端子及び関連レジスタの機能割り付けを示します。

表 2.5 ATU-III機能割り付け

使用端子		機能
端子	TCLKA	測定するパルスをこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
	CBCNT	クロックバス 4,5 に供給する外部入力信号の設定をします。
タイマ C レジスタ	TIORC1	タイマ C の I/O コントロールを設定します。
	TIERC1	TIC1 のインプットキャプチャ割り込みを許可/禁止を設定します
	TSRC0	インプットキャプチャが発生した場合にフラグがセットされます。
	GRC10	コンペアマッチの値を設定します。
	TSTRC	タイマ C のカウント動作を許可/停止します。
	TCNTC0	タイマ C の周期計算用のカウンタです。
	TCRC0	タイマ C の動作を設定します。
PFC レジスタ	PDCR2 PDCR1	端子機能を設定します。
INTC レジスタ	IPR08	IMIC10 のインプットキャプチャ割り込みの優先度を設定します。

2.6.3 動作説明

図 2.10に動作原理を示します。SH72546R のハードウェアおよびソフトウェア処理によってイベントの周期測定を行います。TIOC10 は外部からの入力で、立ち上がり/立下りは任意に行われます。

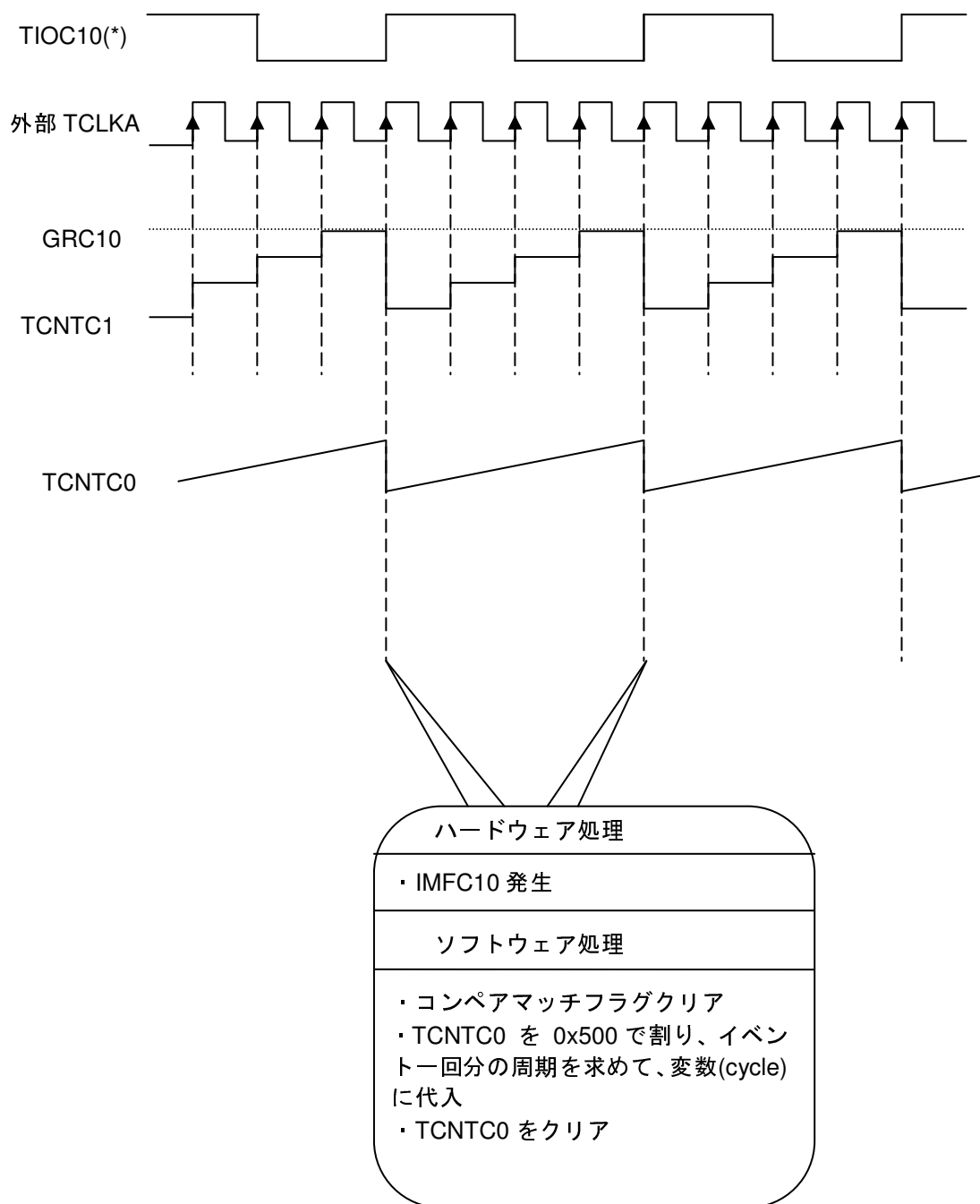


図 2.10 イベント周期計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.6.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
イベント周期測定	IMIC10	ICFA0によって起動し、ICRA0の値によってパルスの周期を測定します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
cycle	イベントの周期に相当するタイマ値を設定します。 イベントの周期は次の式で求められます。 イベントの周期(ns)=タイマ値×Pφ周期(40MHz動作時 25ns)×10 (プリスケアラの分周比)	unsigned long	パルスの周期測定

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール
PDCR2	PD12 端子を TCLKA 機能(クロックバス 4 用外部入力)に設定します。	0x0100	メインルーチン
PDCR1	PD4 を TIOC10 入出力に設定します。	0x0100	
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIタイマ C 及びプリスケアラのカウンタ動作を設定します。	0x0009	
TIORC1	汎用レジスタ 10 のコンペアマッチ及び、コンペアマッチ発生時の動作を設定します。	0x0003	
TIERC1	TIOC10 に対応するコンペアマッチ割り込みを許可します。	0x01	
GRC10	TCNTC1 に対応するコンペアマッチの値を設定します。	0x00005000	
TCRC1	サブブロック 1 をクロックバス 0 で動作させます。	0x0C	
TSTRC	タイマ C のサブブロック 0、1 のカウンタの動作を許可します。	0x03	
IPR08	TIOC00,01 の割り込み要求の優先度を設定します。	0xF000	イベント周期測定
TSRC1	タイマ C のコンペアマッチフラグをクリアします。	IMFC10 = 0	
TCNTC0	タイマカウンタをクリアします。	0x00000000	

2.6.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform   : SH7254R
* Description    : This is the main tutorial code.
* Operation      : Periodic measurements of the event.
*****/
/*****
* History : DD.MM.YYYY Version Description
*         : 12.01.2012 1.00   First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル      */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル      */

/*****
Private global variables and functions
*****/
volatile unsigned long cycle;
void main(void);           /* メインルーチン                      */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value : none
*****/

```

```
void main(void)
{
    cycle = 0;

    /* ポート設定 */
    /* Configure PDCR2
    b15:12 Reserved
    b11:10 PD13MD = 00    PD13 入出力 (ポート)
    b9:8 PD12MD = 01    PD12 TCLKA 入力 (ATU-III)
    b7:6 PD11MD = 00    PD11 入出力 (ポート)
    b5:4 PD10MD = 00    PD10 入出力 (ポート)
    b3:2 PD9MD = 00     PD9 入出力 (ポート)
    b1:0 PD8MD = 00     PD8 入出力 (ポート) */
    PORTD.CR2.WORD = 0x0100;    /* PD12 端子を TCLKA 機能(クロックバス 4 用外部入力)に設定
    */

    /* Configure PDCR1
    b15:14 PD7MD = 00    PD7 入出力 (ポート)
    b13:12 PD6MD = 00    PD6 入出力 (ポート)
    b11:10 PD5MD = 00    PD5 入出力 (ポート)
    b9:8 PD4MD = 01     PD4 TIIOC10 入出力 (ATU-III)
    b7:6 PD3MD = 00     PD3 入出力 (ポート)
    b5:4 PD2MD = 00     PD2 入出力 (ポート)
    b3:2 PD1MD = 00     PD1 入出力 (ポート)
    b1:0 PD0MD = 00     PD0 入出力 (ポート) */
    PORTD.CR1.WORD = 0x0100;    /* PD4 端子を TIIOC10 機能(コンペアマッチ入出力)に設定
    */

    /* ATU-III の設定 */
    /* Configure TCRC
    b7    FCMCn3= 0    汎用レジスタ GRCn3 での強制コンペアマッチ禁止
    b6    FCMCn2= 0    汎用レジスタ GRCn2 での強制コンペアマッチ禁止
    b5    FCMCn1= 0    汎用レジスタ GRCn1 での強制コンペアマッチ禁止
    b4    FCMCn0= 0    汎用レジスタ GRCn0 での強制コンペアマッチ禁止
    b3    PWNn0 = 1    PWM モードで動作
    b2:0 CKSELCn = 100 クロックバス 4 でカウント */
    ATUC.SUBBLOCK[1].TCRC.BYTE = 0x0C; /* サブブロック 1 をクロックバス 4 (外部クロック入力
    A) で動作 */

    /* Configure TIERC
    b7:5 Reserved
    b4    OVECn = 0    OVFCn による割り込み要求を禁止
    b3    IMECn3 = 0    IMFCn3 による割り込み要求を禁止
    b2    IMECn2 = 0    IMFCn2 による割り込み要求を禁止
    b1    IMECn1 = 0    IMFCn1 による割り込み要求を禁止
    b0    IMECn0 = 1    IMFCn0 による割り込み要求を許可 */
    ATUC.SUBBLOCK[1].TIERC.BYTE = 0x01; /* TIIOC10 に対応するコンペアマッチ割り込みを許
    可 */

    /* Configure TIORC
    b11    Reserved
    b10:8 IOCN2 = 000    コンペアマッチ禁止
    b7     Reserved
    b6:4 IOCN1 = 000    コンペアマッチ禁止
    b3     Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b2:0 IOCn0 = 011      コンペアマッチでトグル出力 */
ATUC.SUBBLOCK[1].TIORC.WORD = 0x003; /* 汎用レジスタ 10 のコンペアマッチでトグル出力
*/

/* Configure GRC
b31:0 汎用レジスタデータ*/
ATUC.SUBBLOCK[1].GRC[0] = 0x00000500; /* コンペアマッチの値を 0x500 に設定
*/

/* Configure TSTRC
b7:5 Reserved
b4      STRC4 = 0      TCNTC4 のカウント動作を停止
b3      STRC3 = 0      TCNTC3 のカウント動作を停止
b2      STRC2 = 0      TCNTC2 のカウント動作を停止
b1      STRC1 = 1      TCNTC1 のカウント動作を許可
b0      STRC0 = 1      TCNTC0 のカウント動作を許可 */
ATUC.TSTRC.BYTE = 0x03; /* タイマ C サブブロック 0,1 のカウントをスタート
*/

/* Configure CBCNT
b7:6 Reserved
b5:4 CB4EG = 01      クロックバス 4:外部クロックの立ち上がりエッジ選択
b3      Reserved
b2      CB5SEL = 0      クロックバス 5:外部クロック入力 B (TCLKB)
b1:0 CB5EG = 00      クロックバス 5:外部クロックのエッジ検出をしない*/
ATUCTRL.CBCNT.BYTE = 0x10; /* クロックバス 4 は外部入力の立ち上がりエッジを元にカウン
ト */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'9      プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラの分周比を 10 に設定 */

/* 割り込み設定 */
/* Configure IPR08
b15:12 ATU-C1 (IMIC10-13) = H'F      対応する割り込みの優先順位を 15 に設定
b11:8  ATU-C1 (OVIC1) = H'0          対応する割り込みの優先順位を 0 に設定
b7:4  ATU-C2 (IMIC20-23) = H'0      対応する割り込みの優先順位を 0 に設定
b3:0  ATU-C2 (OVIC2) = H'0          対応する割り込みの優先順位を 0 に設定 */
INTC.IPR08.WORD = 0xF000; /* TIIOC0,01 の割り込み優先順位を 15 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0      タイマ J のカウント動作を停止
b8      THE = 0      タイマ H のカウント動作を停止
b7      TGE = 0      タイマ G のカウント動作を停止
b6      TFE = 0      タイマ F のカウント動作を停止
b5      TEE = 0      タイマ E のカウント動作を停止
b4      TDE = 0      タイマ D のカウント動作を停止
b3      TCE = 1      タイマ C のカウント動作を許可
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 0      タイマ A のカウント動作を停止
b0      PSCE = 1     プリスケーラのカウント動作を許可 */
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUCTRL.ATUENR.WORD = 0x0009;      /* タイマ c およびプリスケーラのカウントをスタート
*/

while(1);                          /* 無限ループに入り、待機 */
} /* End of function main() */

/*****
* Function Name : IMIC10
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value : none
*****/
void IMIC10(void)
{
    /* Configure TSRC
    b7:5 Reserved
    b4      OVFCn  &= 1      オーバフロー発生
    b3      IMFCn3 &= 1      インพุットキャプチャ、コンペアマッチフラグ維持
    b2      IMFCn2 &= 1      インพุットキャプチャ、コンペアマッチフラグ維持
    b1      IMFCn1 &= 1      インพุットキャプチャ、コンペアマッチフラグ維持
    b0      IMFCn0 &= 0      インพุットキャプチャ、コンペアマッチフラグをクリア */
    ATUC.SUBBLOCK[1].TSRC.BYTE &= 0xFE;      /* コンペアマッチフラグを
    クリア */

    /* 0x500 で割り、イベント一回の周期を求める */
    cycle = (ATUC.SUBBLOCK[0].TCNTC / ATUC.SUBBLOCK[1].GRC[0]);

    /* Configure TCNTC
    b31:0 タイマカウンタ */
    ATUC.SUBBLOCK[0].TCNTC ^= ATUC.SUBBLOCK[0].TCNTC;
} /* End of function IMIC10() */
```

2.7 動作例 6 パルス出力(トグル出力、タイマカウントクリア)[タイマ C]

2.7.1 概要

図 2.11に示すように、一定の周期でパルスをトグル出力します。本機能を実現する為に、タイマ動作モードを PWM モードに設定します。

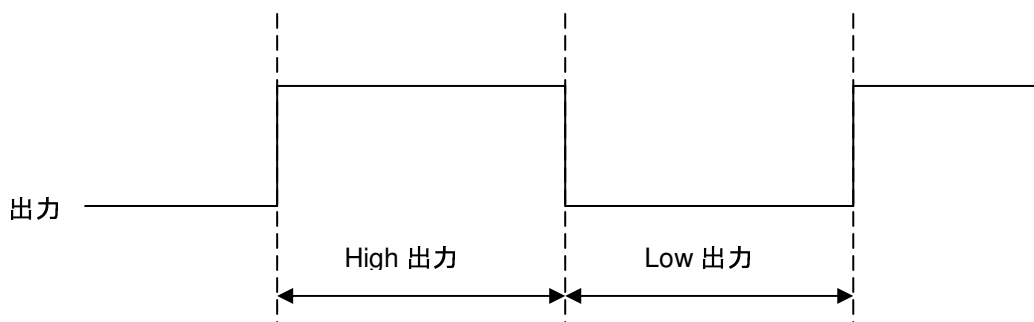


図 2.11 トグル出力

2.7.2 使用機能説明

表 2.6に使用端子及び関連レジスタの機能割り付けを示します。

表 2.6 ATU-III機能割り付け

使用端子		機能
端子	TIOC00	パルスを出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ C レジスタ	TIORC0	タイマ C の機能を設定します。
	TCRC0	タイマ C の動作モード、クロックバスを設定します。
	GRC00	周期を設定します。
	TSTRC	タイマ C 各サブブロックのカウンタの動作を設定します。
PFC レジスタ	PD CR1	端子機能を設定します。

2.7.3 動作説明

図 2.12に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルス出力を行います。PWM モードでは、TCNTC0 と GCR00 とのコンペアマッチ発生時に TCNTC0 の値は 0 にクリアされ、TCNTC0 は再度 0 からカウンタ動作を行います。

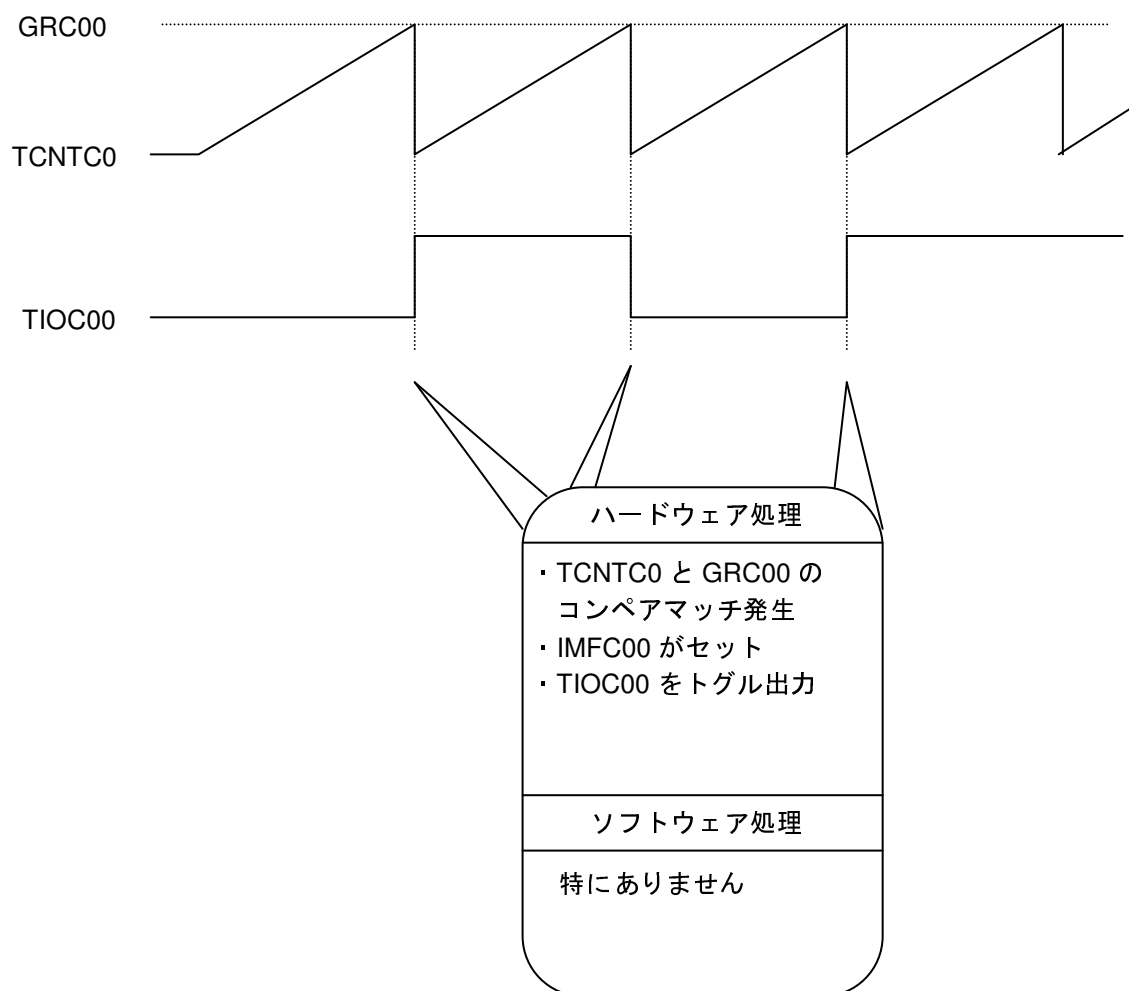


図 2.12 パルス出力動作原理

2.7.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール
PDCR1	ポート D0 端子を入力端子 TIOC00 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	タイマ C 及びプリスケアラのカウンタ動作を設定します。	0x0009	
TIORC0	汎用レジスタ 00 のコンペアマッチ及び、コンペアマッチ発生時の動作を設定します。	0x0003	
TCRC0	サブブロック C0 のクロック選択、動作モードの設定、強制コンペアマッチの設定を行います。	0x08	
GRC00	TCNTC0 に対応するコンペアマッチの値を設定する。	0x00005000	
TSTRC	サブブロック 0、1 のカウンタの動作許可をします。	0x01	

2.7.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Pulse output (toggle output, timer counter clear).
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments     : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PDCR1
  b15:14 PD7MD = 00      PD7 入出力 (ポート)
  b13:12 PD6MD = 00      PD6 入出力 (ポート)
  b11:10 PD5MD = 00      PD5 入出力 (ポート)
  b9:8 PD4MD = 01       PD4 入出力 (ポート)
  b7:6 PD3MD = 00       PD3 入出力 (ポート)
  b5:4 PD2MD = 00       PD2 入出力 (ポート)
  b3:2 PD1MD = 00       PD1 入出力 (ポート)
  b1:0 PD0MD = 01       PD0 TI0C00 入出力 (ATU-III) */
  PORTD.CR1.WORD = 0x0001; /* PD0 を TI0C00 の入出力に設定 (出力波形)
  */

  /* ATU-III の設定 */
  /* Configure TSTRC
  b7:5 Reserved
  b4      STRC4 = 0      TCNTC4 のカウント動作を停止
  b3      STRC3 = 0      TCNTC3 のカウント動作を停止
  b2      STRC2 = 0      TCNTC2 のカウント動作を停止
  b1      STRC1 = 0      TCNTC1 のカウント動作を停止
  b0      STRC0 = 1      TCNTC0 のカウント動作を許可 */
  ATUC.TSTRC.BYTE = 0x01; /* タイマ C サブブロック 0 のカウントをスタート
  */

  /* Configure TCRC
  b7      FCMCn3= 0      汎用レジスタ GRCn3 での強制コンペアマッチ禁止
  b6      FCMCn2= 0      汎用レジスタ GRCn2 での強制コンペアマッチ禁止
  b5      FCMCn1= 0      汎用レジスタ GRCn1 での強制コンペアマッチ禁止
  b4      FCMCn0= 0      汎用レジスタ GRCn0 での強制コンペアマッチ禁止
  b3      PWNn0 = 1      PWM モードで動作
  b2:0 CKSELCn = 000     クロックバス 0 でカウント */
  ATUC.SUBBLOCK[0].TCRC.BYTE = 0x08; /* サブブロック 1 をクロックバス 0 (外部クロック入力
  A) で動作 */

  /* Configure TIORC
  b15     Reserved
  b14:12 IOcn3 = 000     汎用レジスタ 03 コンペアマッチ禁止
  b11     Reserved
  b10:8   IOcn2 = 000     汎用レジスタ 02 コンペアマッチ禁止
  b7      Reserved
  b6:4   IOcn1 = 000     汎用レジスタ 01 コンペアマッチ禁止
  b3      Reserved
  b2:0   IOcn0 = 011     汎用レジスタ 00 コンペアマッチでトグル出力 */
  ATUC.SUBBLOCK[0].TIORC.WORD = 0x0003; /* 汎用レジスタ 00 のコンペアマッチでトグル出力
  */

  /* Configure GRC
  b31:0   汎用レジスタデータ (コンペアマッチに使用) */
  ATUC.SUBBLOCK[0].GRC[0] = 0x00005000; /* コンペアマッチの値を 0x50 に設定
  */

  /* Configure PSCR0
  b15:10  Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b9:0 PSCn = H'9      プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009;      /* プリスケーラの分周比を 10 に設定
*/

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0      タイマ J のカウント動作を停止
b8      THE = 0      タイマ H のカウント動作を停止
b7      TGE = 0      タイマ G のカウント動作を停止
b6      TFE = 0      タイマ F のカウント動作を停止
b5      TEE = 0      タイマ E のカウント動作を停止
b4      TDE = 0      タイマ D のカウント動作を停止
b3      TCE = 1      タイマ C のカウント動作を許可
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 0      タイマ A のカウント動作を停止
b0      PSCE = 1     プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0009;      /* タイマ C およびプリスケーラのカウントを動作
*/

while(1);
} /* End of function main() */
```

2.8 動作例 7 パルス出力(トグル出力、コンペア値加算)[タイマ C]

2.8.1 概要

図 2.13に示すように、一定の周期でパルスをとグル出力します。本動作例では HIGH と Low の切り替えタイミングをコンペアマッチ値の加算で更新することにより、パルス出力を実現します。

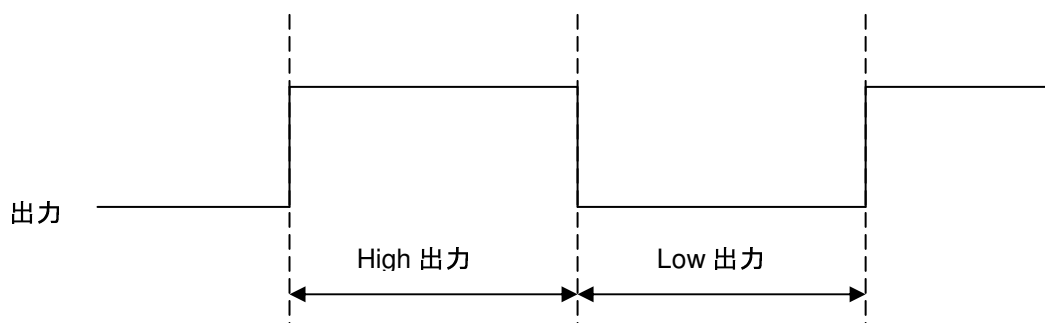


図 2.13 トグル出力

2.8.2 使用機能説明

表 2.7に使用端子及び関連レジスタの機能割り付けを示します。

表 2.7 ATU-III機能割り付け

使用端子		機能
端子	TIOC00	パルスを出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ C レジスタ	TIORC0	タイマ C の機能を設定します。
	TIERC	インプットキャプチャ/コンペアマッチ割り込みの許可/禁止を設定します
	TSRC	インプットキャプチャ/コンペアマッチ発生時にフラグがセットされます。
	GRC	周期を設定します。
	TSTRC	タイマ C 各サブブロックのカウンタの動作を設定します。
PFC レジスタ	PDCR1	端子機能を設定します。
INTC レジスタ	IPR07	IMIC00 のコンペアマッチ割り込みの優先度を設定します。

2.8.3 動作説明

図 2.14に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルス出力を行います。タイマカウンタ TCNTC0 とコンペアマッチ用レジスタ GRC00 とのコンペアマッチが発生すると、GRC00 は自身を加算します。

本動作例では GRC00 の値は常に一定に増加するので、一定の High/Low 幅を持つパルスが出力されます。

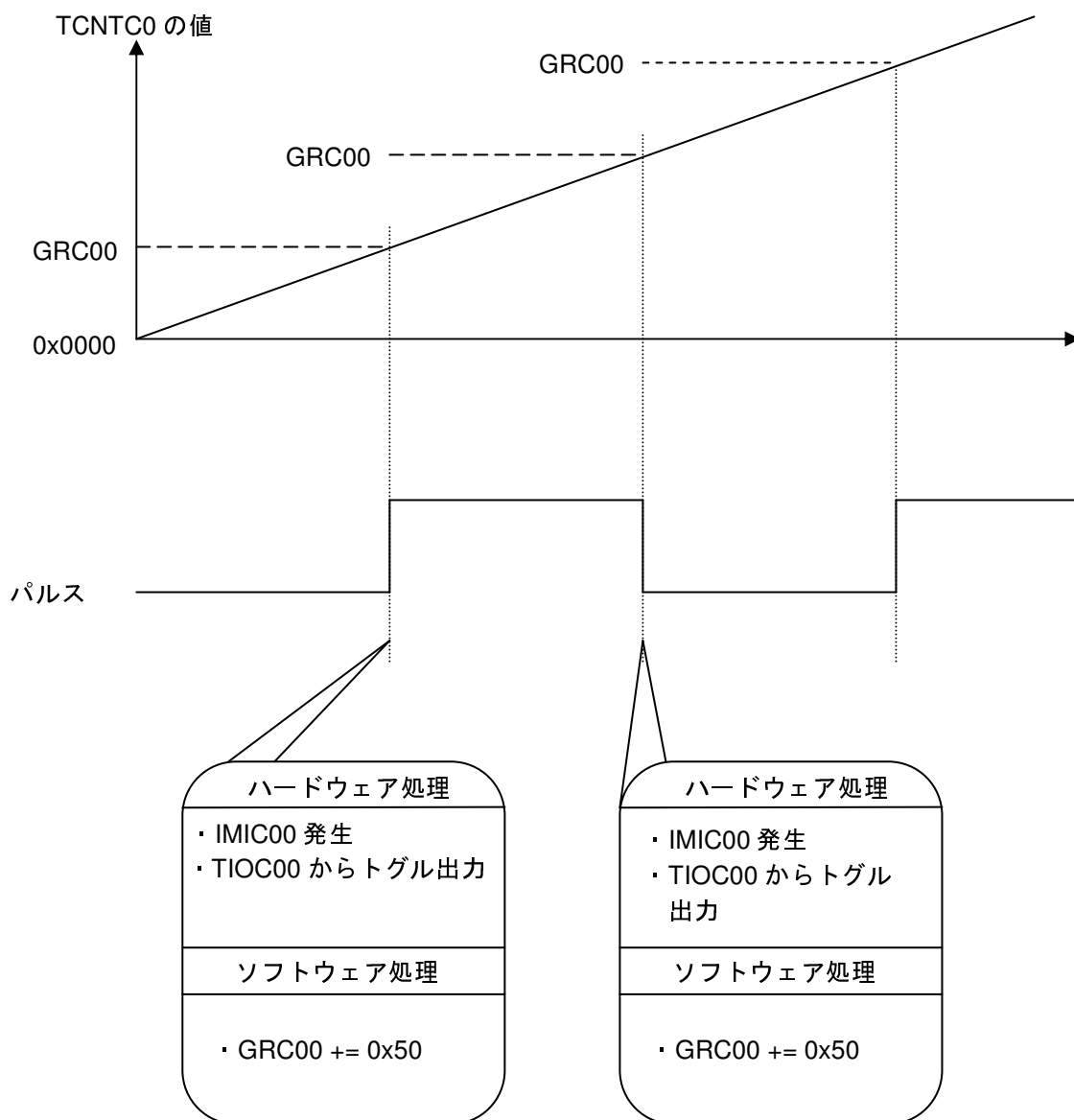


図 2.14 パルス出力動作原理

2.8.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
再コンペアマッチルーチン	IMIC00	ICFC00によって起動し、TSRCをクリアし、汎用レジスタ00の値に加算を行う事によってコンペアマッチを再び発生可能にします。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール
PDCR1	ポート D0 端子を入出力端子 TIOC00 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	タイマ C 及びプリスケアラのカウンタ動作を設定します。	0x0009	
TIORC0	汎用レジスタ 00 のコンペアマッチ及び、コンペアマッチ発生時の動作を設定します。	0x0003	
TIERC0	TIOC10 に対応するコンペアマッチ割り込みを許可します。	0x01	
GRC00	TCNTC1 に対応するコンペアマッチの値を設定する。	0x00005000	
TSTRC0	サブブロック 0 のカウンタの動作許可をします。	0x01	
IPR07	TCNTC0 のコンペアマッチ割り込み要求の優先度を設定します。	0x00FF	

2.8.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description    : This is the main tutorial code.
 * Operation      : Measurement of Pulse period.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PDCR1
  b15:14 PD7MD = 00      PD7 入出力 (ポート)
  b13:12 PD6MD = 00      PD6 入出力 (ポート)
  b11:10 PD5MD = 00      PD5 入出力 (ポート)
  b9:8 PD4MD = 01       PD4 入出力 (ポート)
  b7:6 PD3MD = 00       PD3 入出力 (ポート)
  b5:4 PD2MD = 00       PD2 入出力 (ポート)
  b3:2 PD1MD = 00       PD1 入出力 (ポート)
  b1:0 PD0MD = 01       PD0 TIOC00 入出力 (ATU-III) */
  PORTD.CR1.WORD = 0x0001;          /* PD0 を TIOC00 の入出力に設定 (出力波形) */

  /* ATU-III の設定 */
  /* Configure TSTRC
  b7:5 Reserved
  b4      STRC4 = 0      TCNTC4 のカウント動作を停止
  b3      STRC3 = 0      TCNTC3 のカウント動作を停止
  b2      STRC2 = 0      TCNTC2 のカウント動作を停止
  b1      STRC1 = 0      TCNTC1 のカウント動作を停止
  b0      STRC0 = 1      TCNTC0 のカウント動作を許可 */
  ATUC.TSTRC.BYTE = 0x01;          /* タイマ C サブブロック 0 のカウントをスタート */

  /* Configure TIERC
  b7:5 Reserved
  b4      OVECn = 0      OVFCn による割り込み要求を禁止
  b3      IMECn3 = 0     IMFCn3 による割り込み要求を禁止
  b2      IMECn2 = 0     IMFCn2 による割り込み要求を禁止
  b1      IMECn1 = 0     IMFCn1 による割り込み要求を禁止
  b0      IMECn0 = 1     IMFCn0 による割り込み要求を許可 */
  ATUC.SUBBLOCK[0].TIERC.BYTE = 0x01; /* コンペアマッチフラグ 00 によるコンペアマッチ割り込みを許可 */

  /* Configure TIORC
  b15      Reserved
  b14:12  IOCN3 = 000   汎用レジスタ 03 コンペアマッチ禁止
  b11      Reserved
  b10:8   IOCN2 = 000   汎用レジスタ 02 コンペアマッチ禁止
  b7       Reserved
  b6:4   IOCN1 = 000   汎用レジスタ 01 コンペアマッチ禁止
  b3      Reserved
  b2:0   IOCN0 = 011   汎用レジスタ 00 コンペアマッチでトグル出力 */
  ATUC.SUBBLOCK[0].TIORC.WORD = 0x0003; /* 汎用レジスタ 00 のコンペアマッチでトグル出力 */

  /* Configure GRC
  b31:0   汎用レジスタデータ (コンペアマッチに使用) */
  ATUC.SUBBLOCK[0].GRC[0] = 0x00005000; /* コンペアマッチの値を H'50 に設定 */

  /* Configure PSCR0
  b15:10  Reserved
  b9:0    PSCn = H'9     プリスケアラの分周比を 10 (設定値は-1) に設定 */
  ATUCTRL.PSCR0.WORD = 0x0009;          /* プリスケアラの分周比を 10 に設定 */
}
```

```

/* Configure IPR07
b15:12 ATU-B(CMIB0,1) = H'0      対応する割り込みの優先順位を 0 に設定
b11:8  ATU-B(CMIB6,ICIB0) = H'0  対応する割り込みの優先順位を 0 に設定
b7:4  ATU-C0(IMIC00-03) = H'F    対応する割り込みの優先順位を 15 に設定
b3:0  ATU-C0(OVIC0) = H'F       対応する割り込みの優先順位を 15 に設定 */
INTC.IPR07.WORD = 0x00FF;      /* IMIC00、OVIC0 の割り込み優先順位を 15 に設定
*/

set_imask(0);                  /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0                タイマ J のカウント動作を停止
b8     THE = 0                タイマ H のカウント動作を停止
b7     TGE = 0                タイマ G のカウント動作を停止
b6     TFE = 0                タイマ F のカウント動作を停止
b5     TEE = 0                タイマ E のカウント動作を停止
b4     TDE = 0                タイマ D のカウント動作を停止
b3     TCE = 1                タイマ C のカウント動作を許可
b2     TBE = 0                タイマ B のカウント動作を停止
b1     TAE = 0                タイマ A のカウント動作を停止
b0     PSCE = 1              プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0009; /* タイマ C およびプリスケアラのカウントを動作
*/

while(1);
} /* End of function main() */

/*****
* Function Name : IMIC00
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value  : none
*****/
void IMIC00(void)
{
/* Configure TSRC
b7:5 Reserved
b4     OVFCn &= 1            オーバフロー発生
b3     IMFCn3 &= 1          インพุットキャプチャ、コンペアマッチフラグを維持
b2     IMFCn2 &= 1          インพุットキャプチャ、コンペアマッチフラグを維持
b1     IMFCn1 &= 1          インพุットキャプチャ、コンペアマッチフラグを維持
b0     IMFCn0 &= 0          インพุットキャプチャ、コンペアマッチフラグをクリア */
ATUC.SUBBLOCK[0].TSRC.BYTE &= 0xFE; /* コンペアマッチフラグをクリア
*/

/* Configure GRC
b31:0  汎用レジスタデータ (コンペアマッチに使用) */
ATUC.SUBBLOCK[0].GRC[0] += 0x5000; /* 汎用レジスタのコンペアマッチの値を 0x50 増加
*/
} /* End of function IMIC00() */

```

2.9 動作例 8 パルス出力(High/Low 切替出力、コンペア値加算)[タイマ C]

2.9.1 概要

図 2.15に示すように、一定の周期でパルスをトグル出力します。

本動作例ではポート出力の HIGH と Low の切り替えタイミングを、タイマのコンペアマッチ値加算で更新します。コンペアマッチ発生毎にポートの High/Low を切り替えることで、パルス出力を行います。

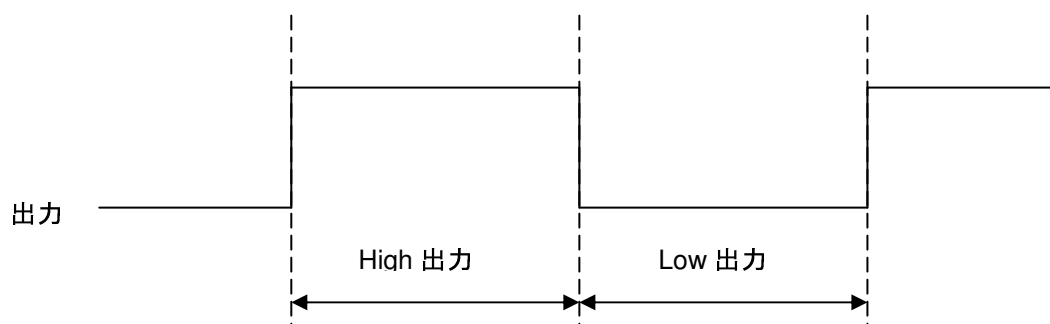


図 2.15 トグル出力

2.9.2 使用機能説明

表 2.8に使用端子及び関連レジスタの機能割り付けを示します。

表 2.8 ATU-III機能割り付け

使用端子		機能
端子	PE0	設定されたレベルのパルスを出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ C レジスタ	TIORC	タイマ C の機能を設定します。
	TIERC	インプットキャプチャ/コンペアマッチ割り込みの許可/禁止を設定します
	TSRC	インプットキャプチャ/コンペアマッチ発生時にフラグがセットされます。
	GRC00	周期を設定します。
	TSTRC	タイマ C 各サブブロックのカウンタの動作を設定します。
PFC レジスタ	PEIOR	PE0 端子の入出力方向を設定します。
	PEDR	PE0 の出力レベルを設定します。
INTC レジスタ	IPR07	IMIC00 のコンペアマッチ割り込みの優先度を設定します。

2.9.3 動作説明

図 2.16に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルス出力を行います。タイマカウンタ TCNTC0 とコンペアマッチ用レジスタ GRC00 とのコンペアマッチが発生すると、GRC00 は自身を加算します。

本動作例では GRC00 の値は常に一定に増加するので、一定の High/Low 幅を持つパルスが出力されます。

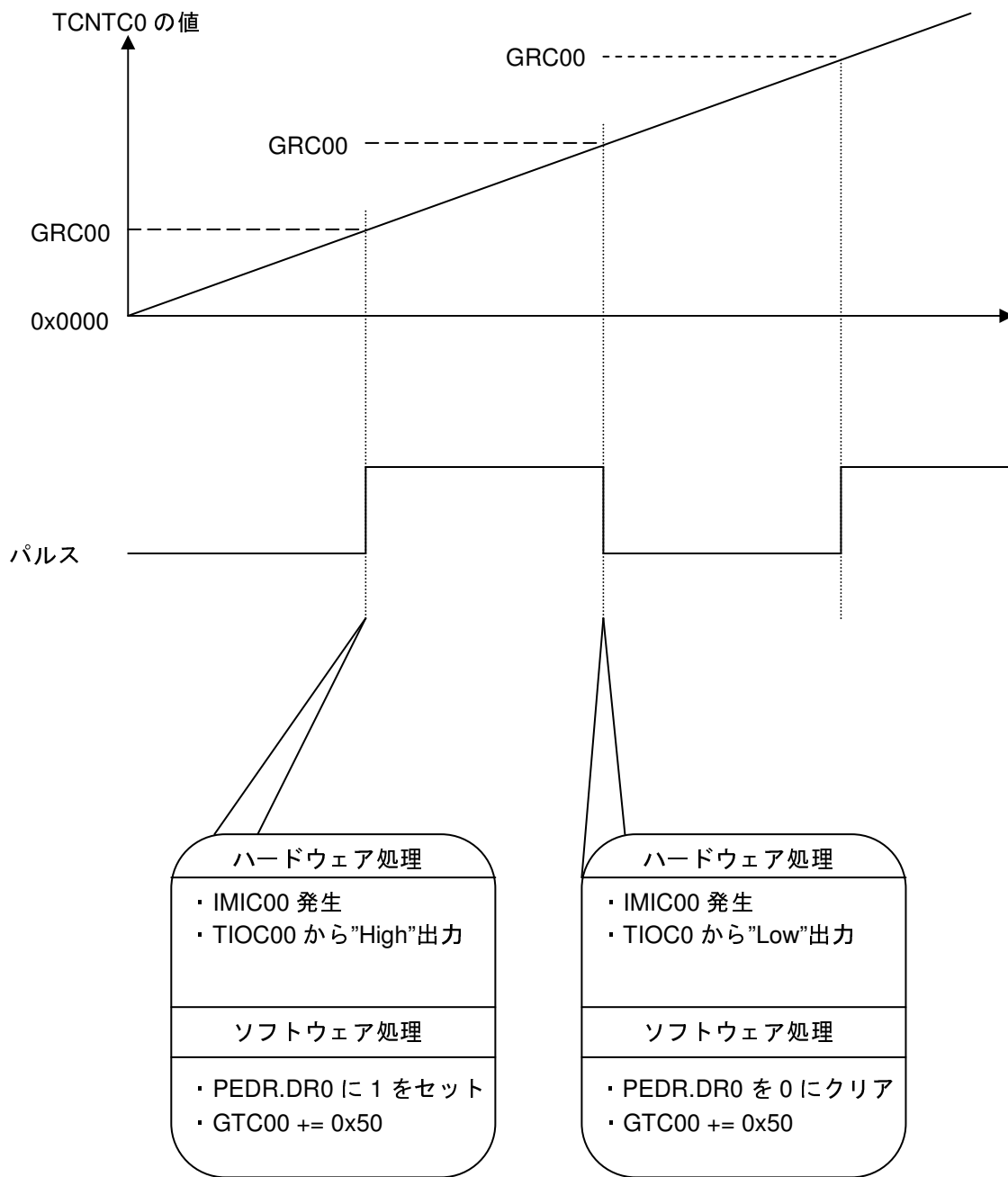


図 2.16 パルス出力動作原理

2.9.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
出力の切り替えルーチン	IMIC00	ICFC00によって起動し、PE0の出力レベルの切り替え、および再コンペアマッチへの処理を行います

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール
PEIOR	ポート PE0 端子を汎用出力に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIタイマ C 及びプリスケアラのカウント動作を設定します。	0x0009	
TIORC0	汎用レジスタ 00 のコンペアマッチ及び、コンペアマッチ発生時の動作を設定します。	0x0001	
TIERC0	TIOC10 に対応するコンペアマッチ割り込みを許可します。	0x01	
GRC00	TCNTC1 に対応するコンペアマッチの値を設定する。	0x00005000	
TSTRC	サブブロック 0、1 のカウントの動作の許可を設定します。	0x01	
IPR07	TCNTC0 のコンペアマッチ割り込み要求の優先度を設定します。	0x00FF	
PEDR	PE0 の出力初期値を Low に設定します。	0x00	

2.9.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Pulse output. (Switchable output High / Low, compare the added value)
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments     : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PEIOR
  b15:14 Reserved
  b13:0  PEnIOR  対応する端子の入出力設定 0:Input 1:Output */
  PORTE.IOR.WORD = 0x0001;          /* PE0 を出力に設定          */

  /* Configure PEDR
  b15:14 Reserved
  b13:0  REnDR = H'001  対応する端子から "1" を出力 */
  PORTE.DR.BIT.DR0 = 1;            /* PE0 の初期値を High に設定          */

  /* ATU-III の設定 */
  /* Configure TSTRC
  b7:5 Reserved
  b4      STRC4 = 0    TCNTC4 のカウント動作を停止
  b3      STRC3 = 0    TCNTC3 のカウント動作を停止
  b2      STRC2 = 0    TCNTC2 のカウント動作を停止
  b1      STRC1 = 0    TCNTC1 のカウント動作を停止
  b0      STRC0 = 1    TCNTC0 のカウント動作を許可 */
  ATUC.TSTRC.BYTE = 0x01;          /* サブブロック 0 のカウンタを動作          */

  /* Configure TIERC
  b7:5 Reserved
  b4      OVECn = 0    OVFCn による割り込み要求を禁止
  b3      IMECn3 = 0    IMFCn3 による割り込み要求を禁止
  b2      IMECn2 = 0    IMFCn2 による割り込み要求を禁止
  b1      IMECn1 = 0    IMFCn1 による割り込み要求を禁止
  b0      IMECn0 = 1    IMFCn0 による割り込み要求を許可 */
  ATUC.SUBBLOCK[0].TIERC.BYTE = 0x01; /* コンペアマッチフラグ 00 によるコンペアマッ
  チ割り込みを許可 */

  /* Configure TIORC
  b15      Reserved
  b14:12  IOcn3 = 000  コンペアマッチ禁止
  b11      Reserved
  b10:8   IOcn2 = 000  コンペアマッチ禁止
  b7       Reserved
  b6:4   IOcn1 = 000  コンペアマッチ禁止
  b3       Reserved
  b2:0   IOcn0 = 001  コンペアマッチで 0 出力 */
  ATUC.SUBBLOCK[0].TIORC.WORD = 0x0001; /* 汎用レジスタ 00 のインプットキャプチャを禁止
  */

  /* Configure GRC
  b31:0   汎用レジスタデータ */
  ATUC.SUBBLOCK[0].GRC[0] = 0x00005000; /* コンペアマッチの値を 0x5000 に設定
  */

  /* Configure PSCR0
  b15:10  Reserved
  b9:0    PScn = H'9    プリスケアラの分周比を 10 (設定値は-1) に設定 */
  ATUCTRL.PSCR0.WORD = 0x0009;          /* プリスケアラの分週比を 10 に設定          */

  /* Configure IPR07
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:12 ATU-B(CMIB0,1) = H'0    対応する割り込みの優先順位を 15 に設定
b11:8  ATU-B(CMIB6,ICIB0) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4  ATU-C0(IMIC00-03) = H'F    対応する割り込みの優先順位を 0 に設定
b3:0  ATU-C0(OVIC0) = H'F      対応する割り込みの優先順位を 0 に設定 */
INTC.IPR07.WORD = 0x00FF;      /* IMIC00、OVIC0 の割り込み優先順位を 15 に設定
*/

set_imask(0);                  /* 割り込みマスクレベルを 0 に設定          */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0                タイマ J のカウント動作を停止
b8     THE = 0                タイマ H のカウント動作を停止
b7     TGE = 0                タイマ G のカウント動作を停止
b6     TFE = 0                タイマ F のカウント動作を停止
b5     TEE = 0                タイマ E のカウント動作を停止
b4     TDE = 0                タイマ D のカウント動作を停止
b3     TCE = 1                タイマ C のカウント動作を許可
b2     TBE = 0                タイマ B のカウント動作を停止
b1     TAE = 0                タイマ A のカウント動作を停止
b0     PSCE = 1              プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0009;  /* タイマ C およびプリスケアラのカウントを動作
*/

while(1);
} /* End of function main() */

/*****
* Function Name : IMIC00
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value : none
*****/
void IMIC00(void)
{
    /* Configure TSRC
    b7:5 Reserved
    b4     OVFCn &= 1          オーバフローフラグを維持
    b3     IMFCn3 &= 1         インพุットキャプチャ、コンペアマッチフラグを維持
    b2     IMFCn2 &= 1         インพุットキャプチャ、コンペアマッチフラグを維持
    b1     IMFCn1 &= 1         インพุットキャプチャ、コンペアマッチフラグを維持
    b0     IMFCn0 &= 0         インพุットキャプチャ、コンペアマッチフラグをクリア */
    ATUC.SUBBLOCK[0].TSRC.BYTE &= 0xFE; /* コンペアマッチフラグをクリア          */

    /* Configure GRC
    b31:0  汎用レジスタデータ (コンペアマッチに使用) */
    ATUC.SUBBLOCK[0].GRC[0] += 0x5000; /* 汎用レジスタのコンペアマッチの値を 0x50 増加 */

    /* Configure PEDR
    b15:14 Reserved
    b13:0  RENDR              対応する端子から "1" を出力 (b0 = DR.BIT.DR0) */
    if(PORTE.DR.BIT.DR0 == 0) /* 出力が Low ならば          */
    {
        PORTE.DR.BIT.DR0 = 1; /* 出力を High に設定          */
    }
    else /* 出力を High にならば          */

```

```
{  
    PORTE.DR.BIT.DR0 = 0;      /* 出力を Low に設定          */  
}  
} /* End of function IMIC00() */
```

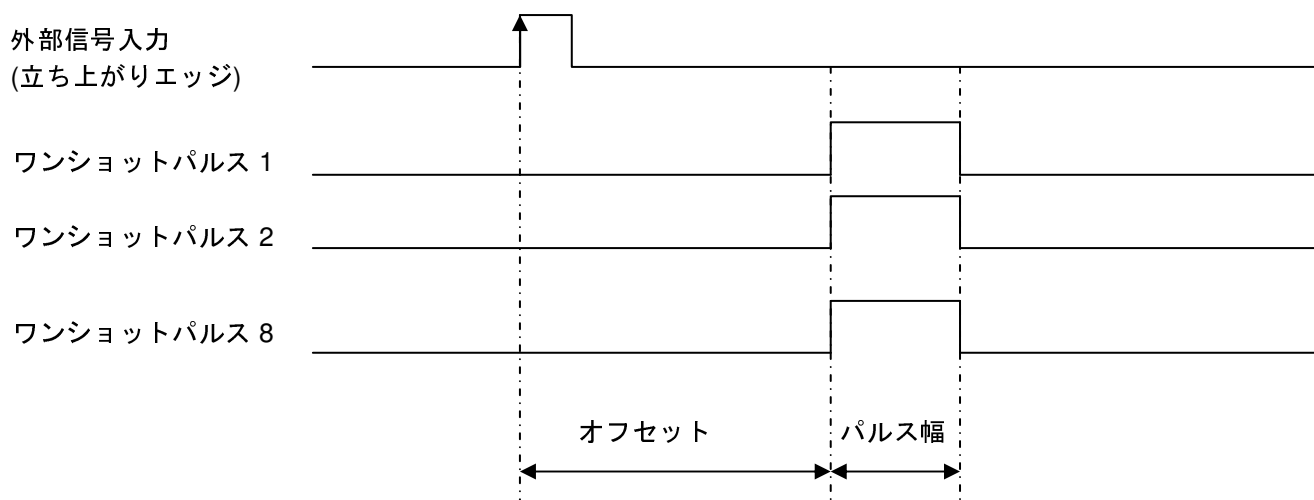
2.10 動作例 9 ワンショットパルス出力(オフセット付)[タイマ D]

2.10.1 概要

- 1) 図 2.17に示すように外部信号の立ち上がり同期してワンショットパルスを出力します。オフセット、パルス幅は内部クロックカウンタ値を設定します。
- 2) 本タスク例では、外部信号の立ち上がりからのオフセット及びパルス幅は以下に示す範囲で可変可能です。

$$4\mu\text{s} < \text{オフセット} < P\phi \text{周期} \times 65536$$

$$\text{*注: } 200\text{ns} \leq \text{パルス幅} < 13\text{ms}$$



- *注: オフセットは外部信号の周期より小さく、
オフセット設定ルーチンの実行時間(約 $4\mu\text{s}$)より大きいこと。

図 2.17 ワンショットパルス出力

2.10.2 使用機能説明

表 2.9に使用端子及び関連レジスタの機能割り付けを示します。

表 2.9 ATU-III機能割り付け

使用端子		機能
端子	TIA00	開始用の外部信号を入力します。
	TOD00B~02B	ワンショットパルスを出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ0の分周比を設定します。
タイマ A レジスタ	TIOR1A	キャプチャする外部入力のエッジの向きを設定します。
	TSRA0	外部入力のインプットキャプチャが発生した場合、フラグがセットされます。
	TIERA0	TIA00のインプットキャプチャ割り込みを許可します。
タイマ D レジスタ	TIOR1D	OCRDに対応するコンペアマッチの許可、および発生時の動作の設定を行います。
	DCRD0	ダウンカウンタの開始、停止条件を設定します。
	TSRD0	コンペアマッチ及び、オーバー、アンダー両フローが発生した場合、フラグがセットされます。
	TSTRD	タイマDのサブブロックのカウント動作を設定します。
	OCRD0	タイマDのアウトプットコンペアレジスタです。
	DCNTD0	タイマDのダウンカウンタです。
INTCレジスタ	IPR06	TIA00のコンペアマッチ割り込みの優先度を設定します。
PFCレジスタ	PECR1	ポートE端子の機能を設定します。
	PFCE1	ポートF端子の機能を設定します。

2.10.3 動作説明

図 2.18に動作原理を示します。図に示すように SH72546R のハードウェア及びソフトウェアの処理によりワンショットパルスを出力します。タイマに対する3つのコンペアレジスタ設定と、ダウンカウント長設定を行うことで、それぞれ異なった波形のワンショットパルスを生成します。

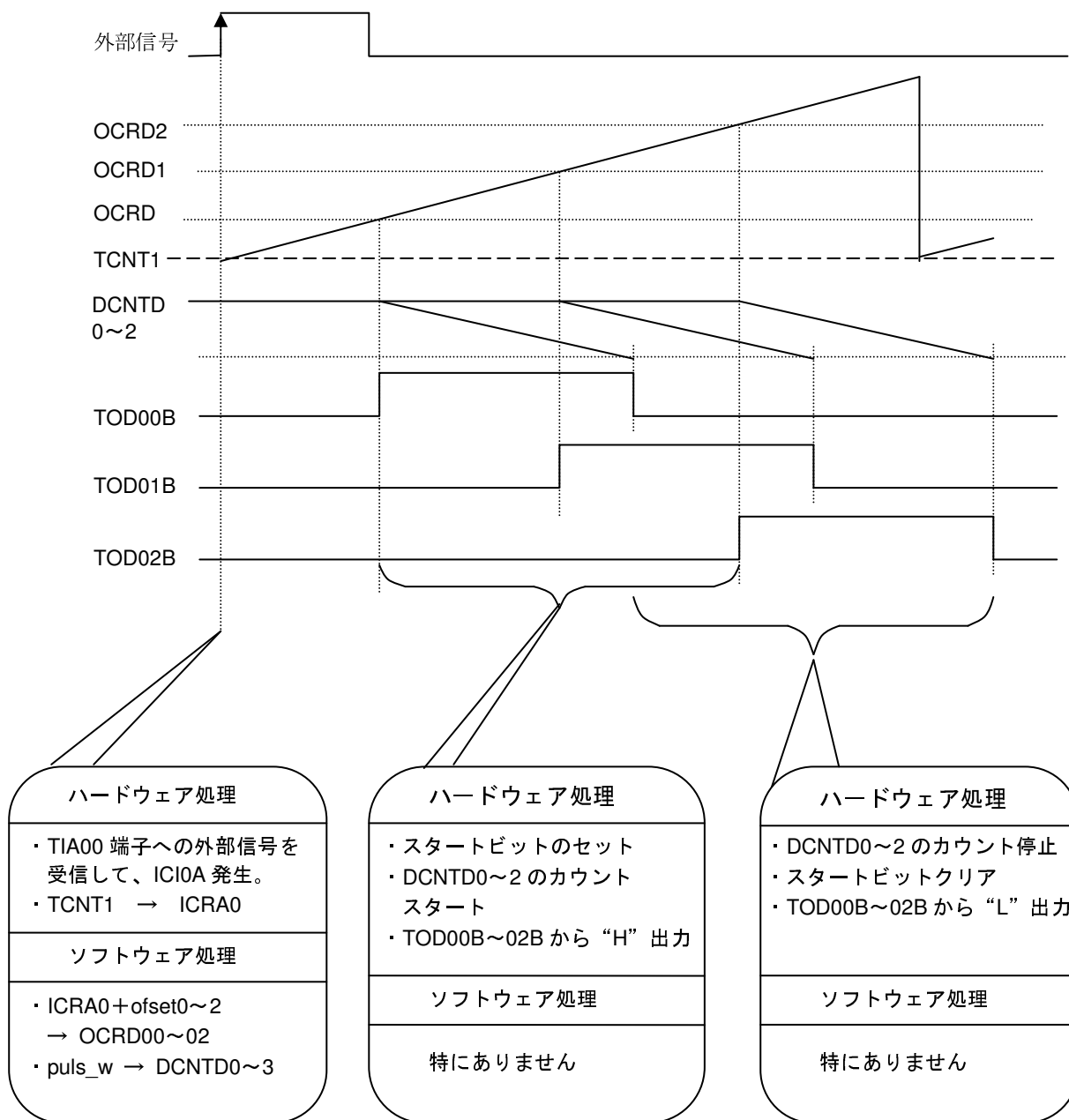


図 2.18 ワンショットパルス出力動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.10.4 ソフトウェア説明

- 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	ATU-III等の初期設定を行います。
ワンショットパルス出カルーチン	ICIA00	ICFA0によって起動し、オフセット及びパルス幅を設定し、ワンショットパルスを出力します。

- 使用定数の説明

ラベル名	機能	データ長	使用モジュール名
offset0~2	ワンショットパルスのオフセットに相当するタイマ値を設定します。オフセットは以下の式で求められます。 オフセット(ns)=タイマ値×Pφ周期(40MHz動作時 25ns)×10(プリスケアラの分周比)	unsigned short	メインルーチン ワンショットパルス出力
puls_w	ワンショットパルスのパルス幅に相当するタイマ値を設定します。パルス幅は以下の式で求められます。 パルス幅(ns)=タイマ値×Pφ周期×10	unsigned short	

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	PE0 端子を TIA0 に設定します。	0x0001	メインルーチン
PFCR1	PF00、01,02 を TOD 出力 00B、01B、02B に設定します。	0x0015	
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのタイマ A、D およびプリスケアラのカウント開始を設定します。	0x0013	
TIOR1A	TIA0 の立ち上がりエッジで ICRA0 へインプットキャプチャするように設定します。	0x0001	
TIERA	TIA0 による割り込み要求を許可します。	0x01	
ICRA0	TIA0 のインプットキャプチャで TCNTA の値を転送します。	—	
TIOR1D0	タイマ D の OCRD00~02 をコンペアマッチ要因に設定します。	0x0153F	
IPR06	TIA0 の割り込み優先レベルを 15 に設定します。	0xF000	
DCRD0	OCRD00~02 のコンペアマッチをダウンカウントスタートトリガに設定します。	0x0222	
OCRD00~02	ワンショットパルスのオフセットを設定します。	—	ワンショットパルス出力
DCNTD00~02	ワンショットパルスのパルス幅を設定します。	—	

2.10.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : One-shot pulse output. (With offset)
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Macro definitions
 */
#define offset0 0x2000        /* オフセット */
#define offset1 0x2500        /* オフセット */
#define offset2 0x2A00        /* オフセット */
#define puls_w  0x4500        /* パルス幅 */

/*
 * Private global variables and functions
 */
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
void main(void); /* メインルーチン */

/*****
* Function Name : main
* Description : The main loop
* Arguments : none
* Return Value : none
*****/
void main(void)
{
    /* PORT の設定 */
    /* Configure PECR1
b15:14 PE7MD = 0 PE7 入出力 (ポート)
b13:12 PE6MD = 0 PE6 入出力 (ポート)
b11 Reserved
b10 PE5MD = 0 PE5 入出力 (ポート)
b9 Reserved
b8 PE4MD = 0 PE4 入出力 (ポート)
b7 Reserved
b6 PE3MD = 0 PE3 入出力 (ポート)
b5:4 PE2MD = 0 PE2 入出力 (ポート)
b3:2 PE1MD = 0 PE1 入出力 (ポート)
b1 Reserved
b0 PE0MD = 1 TIA00 入力 (ATU-III) */
PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定 */

    /* Configure PFCR1
b15:14 PF7MD[1:0] = 00 PF7 入出力 (ポート)
b13:12 PF6MD[1:0] = 00 PF6 入出力 (ポート)
b11:10 PF5MD[1:0] = 00 PF5 入出力 (ポート)
b9:8 PF4MD[1:0] = 00 PF4 入出力 (ポート)
b7:6 PF3MD[1:0] = 00 PF3 入出力 (ポート)
b5:4 PF2MD[1:0] = 01 TOD02B 出力 (ATU-III)
b3:2 PF1MD[1:0] = 01 TOD01B 出力 (ATU-III)
b1:0 PF0MD[1:0] = 01 TOD00B 出力 (ATU-III) */
PORTF.CR1.WORD = 0x0015; /* PF0,1,2 を TOD00B,01B,02B 出力に設定 */

    /* ATU-III Timer A の設定 */
    /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 0 TIA のインプットキャプチャ禁止
b9:8 IOA4 = 0 TIA のインプットキャプチャ禁止
b7:6 IOA3 = 0 TIA のインプットキャプチャ禁止
b5:4 IOA2 = 0 TIA のインプットキャプチャ禁止
b3:2 IOA1 = 0 TIA のインプットキャプチャ禁止
b1:0 IOA0 = 1 TIA の立ち上がりで ICRA にキャプチャ */
ATUA.TIOR1A.WORD = 0x0001; /* TIA00 の立ち上がりエッジでインプットキャプチャを行う */

    /* Configure TSRA
b7 OVFA = 0 インプットキャプチャなし
b6 Reserved
b5 ICFA5 = 0 インプットキャプチャなし
b4 ICFA4 = 0 インプットキャプチャなし
b3 ICFA3 = 0 インプットキャプチャなし
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b2      ICFA2 = 0   インพุットキャプチャなし
b1      ICFA1 = 0   インพุットキャプチャなし
b0      ICFA0 = 0   インพุットキャプチャなし */
ATUA.TSRA.BYTE &= 0x00;          /* インพุットキャプチャフラグをクリア          */

/* Configure TIERA
b7      OVEA = 0   オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0   インพุットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0   インพุットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0   インพุットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0   インพุットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0   インพุットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 1   インพุットキャプチャ割り込み A0 要求の出力を許可 */
ATUA.TIERA.BYTE = 0x01;          /* TIA00 のインพุットキャプチャ割り込みを許可          */

/* ATU-III Timer D の設定 */
/* Configure TIOR1D0
b15:14 OSSD03[1:0] = 00   TOD03A 端子出力なし
b13:12 OSSD02[1:0] = 01   コンペアマッチ A 発生時、TOD02A 端子に I/O コントロールビット
A での設定値を出力
b11:10 OSSD01[1:0] = 01   コンペアマッチ A 発生時、TOD01A 端子に I/O コントロールビット
A での設定値を出力
b9:8 OSSD00[1:0] = 01   コンペアマッチ A 発生時、TOD00A 端子に I/O コントロールビット A での
設定値を出力
b7:6 IOAD03[1:0] = 00   コンペアマッチ禁止
b5:4 IOAD02[1:0] = 11   コンペアマッチでトグル出力
b3:2 IOADn1[1:0] = 11   コンペアマッチでトグル出力
b1:0 IOADn0[1:0] = 11   コンペアマッチでトグル出力 */
ATUD.SUBBLOCKB[0].TIOR1D.WORD = 0x153F; /* OCRD00,01,02 のコンペアマッチ A 発生で
TOD00A,01A,02A はトグル出力 */

/* Configure DCRD0
b15      Reserved
b14:12 TRGSEL03[2:0] = 000 カウントスタートトリガ = カウントストップトリガ = なし
b11      Reserved
b10:8   TRGSEL02[2:0] = 010 カウントスタートトリガ = コンペアマッチ A / カウントストップ
トリガ = なし
b7      Reserved
b6:4   TRGSEL01[2:0] = 010 カウントスタートトリガ = コンペアマッチ A / カウントストップト
リガ = なし
b3      Reserved
b2:0   TRGSEL00[2:0] = 010 カウントスタートトリガ = コンペアマッチ A / カウントストップト
リガ = なし */
ATUD.SUBBLOCKB[0].DCRD.WORD = 0x0222; /* コンペアマッチ A 発生時にダウンカウントスター
ト          */

/* Configure TSRD0
b15:14 Reserved
b13      OVFD2D0 = 0   TCNT2D0 でオーバフローなし
b12      OVFD1D0 = 0   TCNT1D0 でオーバフローなし
b11:8   UDFD03:0 = 0   アンダフローなし
b7:4   CMFAD03:0 = 0   コンペアマッチ A なし
b3:0   CMFBD03:0 = 0   コンペアマッチ B なし */
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダーフローフラグをク  
リア */

/* Configure TSTRD
b7:4 Reserved
b3 STRD3 = 0 TCNT1D3、TCNT2D3、DCNTD30 のカウント動作を停止
b2 STRD2 = 0 TCNT1D2、TCNT2D2、DCNTD20 のカウント動作を停止
b1 STRD1 = 0 TCNT1D1、TCNT2D1、DCNTD10 のカウント動作を停止
b0 STRD0 = 1 TCNT1D0、TCNT2D0、DCNTD00 のカウント動作を許可 */
ATUD.TSTRD.BYTE = 0x01; /* タイマ D のサブブロック 0 のカウントをスタート
*/

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCR0[9:0] = H'9 プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラ 0 の分周比を 10 に設定
*/

/* Configure IPR06
b15:12 ATU-A(ICIA0,1) = H'F 対応する割り込みの優先順位を 15 に設定
b11:8 ATU-A(ICIA2,3) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4 ATU-A(ICIA4,5) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 ATU-A(OVIA) = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR06.WORD = 0xF000; /* TIA00 の割り込み(ICIA0)の優先順位を 15 に設定
*/

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9 TJE = 0 タイマ J のカウント動作を停止
b8 THE = 0 タイマ H のカウント動作を停止
b7 TGE = 0 タイマ G のカウント動作を停止
b6 TFE = 0 タイマ F のカウント動作を停止
b5 TEE = 0 タイマ E のカウント動作を停止
b4 TDE = 0 タイマ D のカウント動作を許可
b3 TCE = 0 タイマ C のカウント動作を停止
b2 TBE = 0 タイマ B のカウント動作を停止
b1 TAE = 1 タイマ A のカウント動作を許可
b0 PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0013; /* タイマ A,D およびプリスケーラのカウントをス  
タート */

while(1);
} /* End of function main() */

/*****
* Function Name : ICIA0
* Description : インพุットキャプチャ割り込み
* Arguments : none
* Return Value : none
*****/
void ICIA0(void)
{
/* Configure TSRA
b7 OVFA = 0 インพุットキャプチャフラグをクリア
```

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b6      Reserved
b5      ICFA5 = 0インプットキャプチャフラグをクリア
b4      ICFA4 = 0インプットキャプチャフラグをクリア
b3      ICFA3 = 0インプットキャプチャフラグをクリア
b2      ICFA2 = 0インプットキャプチャフラグをクリア
b1      ICFA1 = 0インプットキャプチャフラグをクリア
b0      ICFA0 = 0インプットキャプチャフラグをクリア */
ATUA.TSRA.BYTE &= 0x00; /* インプットキャプチャフラグをク
リア */

/* Configure TIERA
b7      OVEA = 0 オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0インプットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0インプットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0インプットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0インプットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0インプットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 0インプットキャプチャ割り込み A0 要求の出力を禁止 */
ATUA.TIERA.BYTE = 0x00; /* TIA00 のインプットキャプチャ
割り込みを禁止 */

/* インプットキャプチャ時のカウントに一定数を加えてオフセットの値を決定 */
ATUD.SUBBLOCKB[0].OCRD[0] = (ATUA.ICRA0 + offset0) << 8; /* OCRD00 に TCNT1D0 に
対応するコンペアマッチ A の条件を設定 */
ATUD.SUBBLOCKB[0].OCRD[1] = (ATUA.ICRA0 + offset1) << 8; /* OCRD01 に TCNT1D1 に
対応するコンペアマッチ A の条件を設定 */
ATUD.SUBBLOCKB[0].OCRD[2] = (ATUA.ICRA0 + offset2) << 8; /* OCRD02 に TCNT1D2 に
対応するコンペアマッチ A の条件を設定 */

/* ダウンカウントの長さを決定し、パルスの High 幅を設定 */
ATUD.SUBBLOCKB[0].DCNTD[0] = puls_w << 8;
ATUD.SUBBLOCKB[0].DCNTD[1] = puls_w << 8;
ATUD.SUBBLOCKB[0].DCNTD[2] = puls_w << 8;
} /* End of function ICIA0() */
```

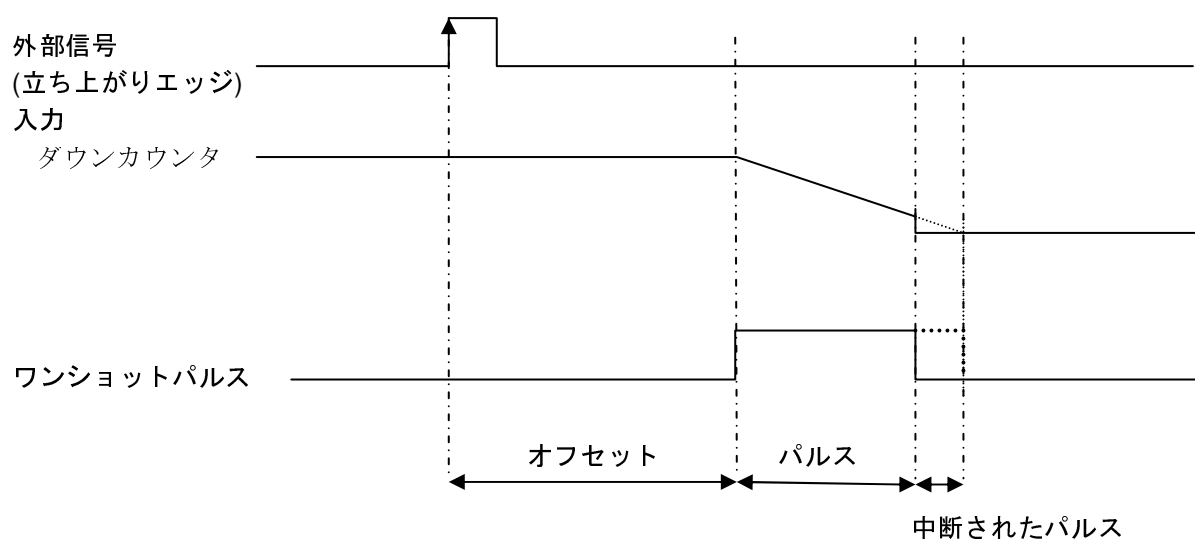
2.11 動作例 10 ワンショットパルス出力(オフセット、ターミネート付)[タイマ D]

2.11.1 概要

- 1) 図 2.19に示すように外部信号の立ち上がりに同期してワンショットパルスを出力します。オフセット、パルス幅は内部クロックカウンタ値を設定します。
- 2) DCRD への設定によりパルス出力を強制的に終了しパルス幅を制御します。
- 3) 本タスク例では、外部信号の立ち上がりからのオフセット及びパルス幅は以下に示す範囲で可変可能です。

$$4\mu\text{s} < \text{オフセット} < P\phi \text{周期} \times 65536$$

$$\text{*注: } 200\text{ns} \leq \text{パルス幅} < 13\text{ms}$$



*注：オフセットは外部信号の周期より小さく、
オフセット設定ルーチンの実行時間(約 $4\mu\text{s}$)より大きいこと。

図 2.19 ワンショットパルス出力

2.11.2 使用機能説明

表 2.10に使用端子及び関連レジスタの機能割り付けを示します。

表 2.10 ATU-III機能割り付け

使用端子		機能
端子	TIA00	開始用の外部信号を入力します。
	TOD00~02B	ワンショットパルスを出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ0の分周比を設定します。
タイマA レジスタ	TIOR1A	キャプチャする外部入力のエッジの向きを設定する。
	TSRA	外部入力のインプットキャプチャが発生した場合、フラグがセットされます。
	TIERA	TIA00のインプットキャプチャ割り込みを許可します。
タイマD レジスタ	TIOR1D	OCRDに対応するコンペアマッチの許可、および発生時の動作の設定を行います。
	TIOR2D	GRDの機能の設定を行います。
	DCRD	ダウンカウントの開始、停止条件を設定します。
	TSRD	コンペアマッチ及び、オーバー、アンダー両フローが発生した場合、フラグがセットされます。
	TSTRD	タイマDのサブブロックのカウント動作を設定します。
INTCレジスタ	IPR06	TIA00のコンペアマッチ割り込みの優先度を設定します。

2.11.3 動作説明

図 2.20に動作原理を示します。図に示すように SH72546R のハードウェア及びソフトウェアの処理によりワンショットパルスを出力します。ダウンカウンタの長さを調整することで、ワンショットパルス出力の強制終了(L レベル出力)までの期間を設定します。

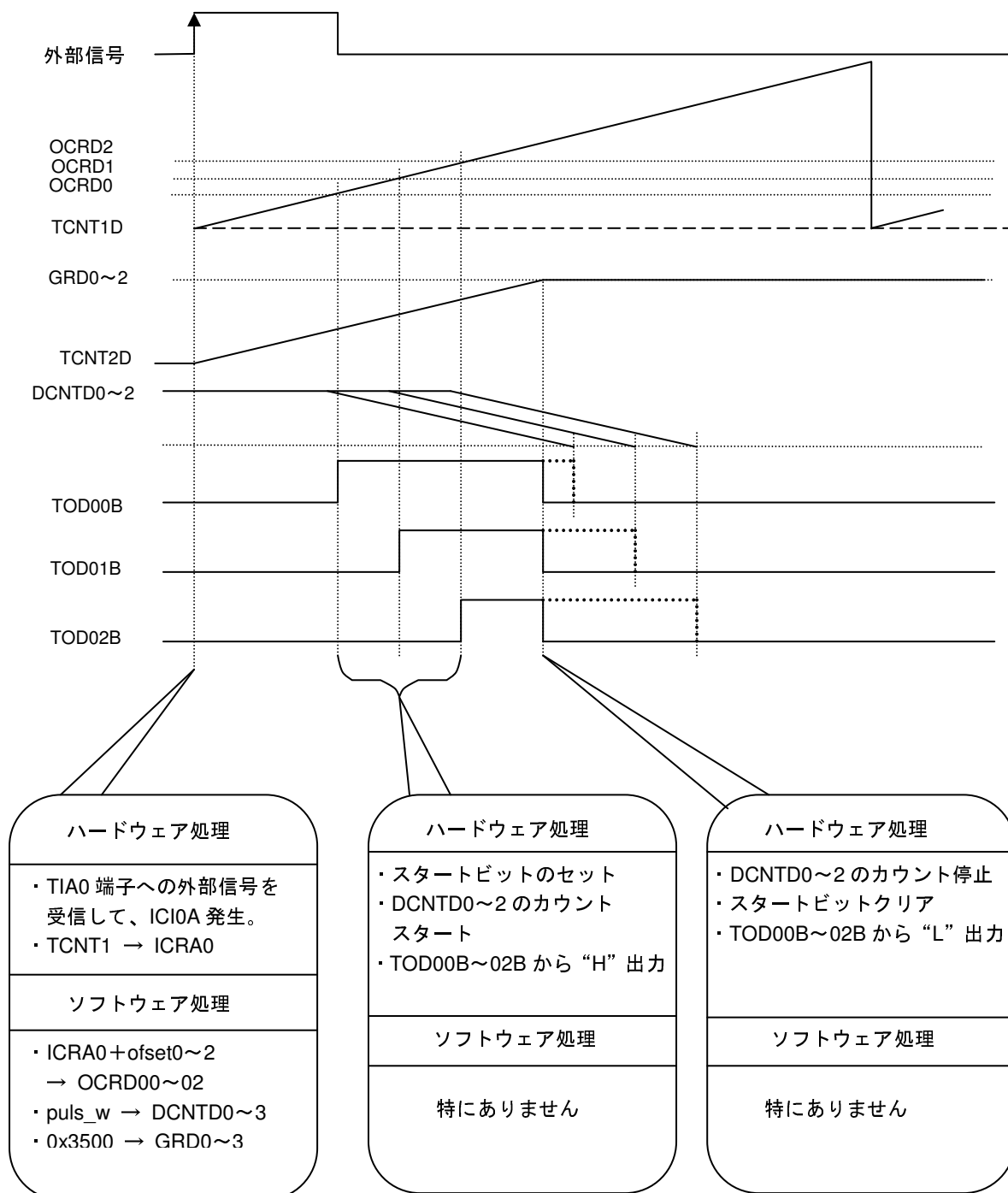


図 2.20 ワンショットパルス出力動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.11.4 ソフトウェア説明

- 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	ATU-III等の初期設定を行います。
ワンショットパルス出力	ICIA00	ICFA0によって起動し、オフセット期間、ターミネート期間、およびパルス幅を設定し、ワンショットパルスを出力します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
Ofset0~2	ワンショットパルスのオフセットに相当するタイマ値を設定します。オフセットは以下の式で求められます。 オフセット(ns)=タイマ値×Pφ周期(40MHz動作時25ns)×10(プリスケアラの分周比)	unsigned short	全モジュールで使用
Puls_w	ワンショットパルスのパルス幅に相当するタイマ値を設定します。パルス幅は以下の式で求められます。 パルス幅(ns)=タイマ値×Pφ周期×10	unsigned short	

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	PE0端子をTIA0に設定します。	0x0001	メインルーチン
PFCR1	PF00~02端子をTOD出力00B、01B、02Bに設定します。	0x0015	
PSCR0	プリスケアラ0の分周比を10に設定します。	0x0009	
ATUENR	ATU-IIIのタイマA、Dおよびプリスケアラのカウント開始を設定します。	0x0013	
TIOR1A	TIA0の立ち上がりエッジでICRA0へインプットキャプチャするように設定します。	0x0001	
TIERA	TIA0による割り込み要求を許可します。	0x01	
ICRA0	TIA0のインプットキャプチャでTCNTAの値を転送します。	—	
TIOR1D0	TCNT1DとOGRD00~02のコンペアマッチを許可します。	0x153F	
TIOR2D0	TCNT2DとGRD00~02のコンペアマッチを許可します。	0x0333	
DCRD0	OGRD00~02のコンペアマッチをダウンカウントスタートトリガに、GED00~02のコンペアマッチをダウンカウントストップトリガに、それぞれ設定します。	0x0333	
OGRD00~02	ワンショットパルスのオフセットを設定します。	—	ワンショットパルス出力
GRD00~02	ワンショットパルスのターミネートまでの期間を設定します。	—	
DCNTD00~02	ワンショットパルスのパルス幅を設定します。	—	
IPR06	TIA0の割り込み優先レベルを15に設定します。	0xF000	メインルーチン

2.11.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform   : SH7254R
* Description     : This is the main tutorial code.
* Operation      : One-shot pulse output. (Offset, with termination)
*****/
/*****
* History : DD.MM.YYYY Version Description
*          : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Macro definitions
*****/
#define Offset0 0x2000        /* オフセット */
#define Offset1 0x2500        /* オフセット */
#define Offset2 0x2A00        /* オフセット */
#define Puls_w 0x2500         /* パルス幅 */

/*****
Private global variables and functions
*****/

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
void main(void);          /* メインルーチン          */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value  : none
*****/
void main(void)
{
  /* PORT の設定 */
  /* Configure PECR1
b15:14 PE7MD = 0      PE7 入出力 (ポート)
b13:12 PE6MD = 0      PE6 入出力 (ポート)
b11     Reserved
b10     PE5MD = 0      PE5 入出力 (ポート)
b9      Reserved
b8      PE4MD = 0      PE4 入出力 (ポート)
b7      Reserved
b6      PE3MD = 0      PE3 入出力 (ポート)
b5:4 PE2MD = 0      PE2 入出力 (ポート)
b3:2 PE1MD = 0      PE1 入出力 (ポート)
b1      Reserved
b0      PE0MD = 1      TIA00 入力 (ATU-III) */
  PORTE.CR1.WORD = 0x0001; /* PE0 を TIA0 入力に設定
                          */

  /* Configure PFCR1
b15:14 PF7MD[1:0] = 00  PF7 入出力 (ポート)
b13:12 PF6MD[1:0] = 00  PF6 入出力 (ポート)
b11:10 PF5MD[1:0] = 00  PF5 入出力 (ポート)
b9:8 PF4MD[1:0] = 00  PF4 入出力 (ポート)
b7:6 PF3MD[1:0] = 00  PF3 入出力 (ポート)
b5:4 PF2MD[1:0] = 01  TOD02B 出力 (ATU-III)
b3:2 PF1MD[1:0] = 01  TOD01B 出力 (ATU-III)
b1:0 PF0MD[1:0] = 01  TOD00B 出力 (ATU-III) */
  PORTF.CR1.WORD = 0x0015; /* PF0,1,2 を TOD00B,01B,02B 出力に設定
                          */

  /* ATU-III Timer A の設定 */
  /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 0      TIA のインプットキャプチャ禁止
b9:8 IOA4 = 0      TIA のインプットキャプチャ禁止
b7:6 IOA3 = 0      TIA のインプットキャプチャ禁止
b5:4 IOA2 = 0      TIA のインプットキャプチャ禁止
b3:2 IOA1 = 0      TIA のインプットキャプチャ禁止
b1:0 IOA0 = 1      TIA の立ち上がりで ICRA にキャプチャ */
  ATUA.TIOR1A.WORD = 0x0001; /* TIA00 の立ち上がりエッジで ICRA にインプットキャプチャ
  を行う          */

  /* Configure TSRA
b7      OVFA = 0      インプットキャプチャフラグをクリア
b6      Reserved
b5      ICFA5 = 0     インプットキャプチャフラグをクリア
b4      ICFA4 = 0     インプットキャプチャフラグをクリア
```

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b3      ICFA3 = 0   インพุットキャプチャフラグをクリア
b2      ICFA2 = 0   インพุットキャプチャフラグをクリア
b1      ICFA1 = 0   インพุットキャプチャフラグをクリア
b0      ICFA0 = 0   インพุットキャプチャフラグをクリア */
ATUA.TSRA.BYTE  &= 0x00;      /* インพุットキャプチャフラグをクリア
*/

/* Configure TIERA
b7      OVEA = 0    オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0   インพุットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0   インพุットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0   インพุットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0   インพุットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0   インพุットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 1   インพุットキャプチャ割り込み A0 要求の出力を許可 */
ATUA.TIERA.BYTE = 0x01;      /* TIA00 のインพุットキャプチャ割り込みを許可
*/

/* ATU-III Timer D の設定 */
/* Configure TIOR1D0
b15:14  OSSD03[1:0] = 00      TOD03A 端子出力なし
b13:12  OSSD02[1:0] = 01      コンペアマッチ A 発生時、TOD02A 端子に I/O コントロールビット A
での設定値を出力
b11:10  OSSD01[1:0] = 01      コンペアマッチ A 発生時、TOD01A 端子に I/O コントロールビット A
での設定値を出力
b9:8    OSSD00[1:0] = 01      コンペアマッチ A 発生時、TOD00A 端子に I/O コントロールビット A
での設定値を出力
b7:6    IOAD03[1:0] = 00      コンペアマッチ禁止
b5:4    IOAD02[1:0] = 11      コンペアマッチでトグル出力
b3:2    IOADn1[1:0] = 11      コンペアマッチでトグル出力
b1:0    IOADn0[1:0] = 11      コンペアマッチでトグル出力 */
ATUD.SUBBLOCKB[0].TIOR1D.WORD = 0x153F; /* OCRD00,01,02 のコンペアマッチ A 発生で
TOD00A,01A,02A はトグル出力 */

/* Configure TIOR2D0
b15      Reserved
b14:12  IOBDn3[2:0] = 000     コンペアマッチ禁止 (GRDn3 をアウトプットコンペアレジスタと
して利用するとき)
b11      Reserved
b10:8   IOBDn2[2:0] = 011     コンペアマッチでトグル出力 (GRDn2 をアウトプットコンペアレ
ジスタとして利用するとき)
b7       Reserved
b6:4    IOBDn1[2:0] = 011     コンペアマッチでトグル出力 (GRDn1 をアウトプットコンペアレ
ジスタとして利用するとき)
b3       Reserved
b2:0    IOBDn0[2:0] = 011     コンペアマッチでトグル出力 (GRDn0 をアウトプットコンペアレ
ジスタとして利用するとき) */
ATUD.SUBBLOCKB[0].TIOR2D.WORD = 0x0333; /* GRD00,01,02 のコンペアマッチ B 発生で
TOD00A はトグル出力 */

/* Configure DCRD0
b15      Reserved
b14:12  TRGSELD03[2:0] = 000  カウントスタートトリガ = カウントストップトリガ = なし
b11      Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b10:8 TRGSELD02[2:0] = 011 カウントスタートトリガ = コンペアマッチ A / カウントストップトリガ = コンペアマッチ B
b7 Reserved
b6:4 TRGSELD01[2:0] = 011 カウントスタートトリガ = コンペアマッチ A / カウントストップトリガ = コンペアマッチ B
b3 Reserved
b2:0 TRGSELD00[2:0] = 011 カウントスタートトリガ = コンペアマッチ A / カウントストップトリガ = コンペアマッチ B */
ATUD.SUBBLOCKB[0].DCRD.WORD = 0x0333; /* コンペアマッチ A 発生時にダウンカウントスタート、コンペアマッチ B 発生時にダウンカウントストップ */

/* Configure TSRD0
b15:14 Reserved
b13 OVF2D0 = 0 オーバフローフラグをクリア
b12 OVF1D0 = 0 オーバフローフラグをクリア
b11:8 UDFD03:0 = 0 アンダフローフラグをクリア
b7:4 CMFAD03:0 = 0 コンペアマッチ A フラグをクリア
b3:0 CMFBD03:0 = 0 コンペアマッチ B フラグをクリア */
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダーフローフラグをクリア */

/* Configure TSTRD
b7:4 Reserved
b3 STRD3 = 0 TCNT1D3、TCNT2D3、DCNTD30 のカウント動作を停止
b2 STRD2 = 0 TCNT1D2、TCNT2D2、DCNTD20 のカウント動作を停止
b1 STRD1 = 0 TCNT1D1、TCNT2D1、DCNTD10 のカウント動作を停止
b0 STRD0 = 1 TCNT1D0、TCNT2D0、DCNTD00 のカウント動作を許可 */
ATUD.TSTRD.BYTE = 0x01; /* タイマ D のサブブロック 0 のカウントをスタート */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCR0[9:0] = H'9 プリスケアラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケアラ 0 の分周比を 10 に設定 */

/* Configure IPR06
b15:12 ATU-A(ICIA0,1) = H'F 対応する割り込みの優先順位を設定
b11:8 ATU-A(ICIA2,3) = H'0 対応する割り込みの優先順位を設定
b7:4 ATU-A(ICIA4,5) = H'0 対応する割り込みの優先順位を設定
b3:0 ATU-A(OVIA) = H'0 対応する割り込みの優先順位を設定 */
INTC.IPR06.WORD = 0xF000; /* TIA00 の割り込み(ICIA0)の優先順位を 15 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9 TJE = 0 タイマ J のカウント動作を停止
b8 THE = 0 タイマ H のカウント動作を停止
b7 TGE = 0 タイマ G のカウント動作を停止
b6 TFE = 0 タイマ F のカウント動作を停止
b5 TEE = 0 タイマ E のカウント動作を停止
b4 TDE = 0 タイマ D のカウント動作を許可
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b3      TCE = 0      タイマ C のカウント動作を停止
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 1      タイマ A のカウント動作を許可
b0      PSCE = 1     プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0013;          /* タイマ A,D およびプリスケアラのカウントをス
タート */

while(1);
} /* End of function main() */

/*****
* Function Name : ICIA0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value  : none
*****/
void ICIA0(void)
{
    /* Configure TSRA
b7      OVFA = 0     インพุットキャプチャなし
b6      Reserved
b5      ICFA5 = 0    インพุットキャプチャなし
b4      ICFA4 = 0    インพุットキャプチャなし
b3      ICFA3 = 0    インพุットキャプチャなし
b2      ICFA2 = 0    インพุットキャプチャなし
b1      ICFA1 = 0    インพุットキャプチャなし
b0      ICFA0 = 0    インพุットキャプチャなし */
    ATUA.TSRA.BYTE &= 0x00;          /* インพุットキャプチャをクリア */

    /* Configure TIERA
b7      OVEA = 0     オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0    インพุットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0    インพุットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0    インพุットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0    インพุットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0    インพุットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 0    インพุットキャプチャ割り込み A0 要求の出力を禁止 */
    ATUA.TIERA.BYTE = 0x00;          /* TIA00 のインพุットキャプチャ割り込みを禁止 */

    /* インพุットキャプチャ時のカウントに一定数を加えてオフセットの値を決定 */
    ATUD.SUBBLOCKB[0].OCRDRD[0] = (ATUA.ICRA0 + Ofset0) << 8; /* OCRDRD00、TCNT1D0 に対
    応するコンペアマッチ A の条件を設定*/
    ATUD.SUBBLOCKB[0].OCRDRD[1] = (ATUA.ICRA0 + Ofset1) << 8; /* OCRDRD01、TCNT1D1 に対
    応するコンペアマッチ A の条件を設定*/
    ATUD.SUBBLOCKB[0].OCRDRD[2] = (ATUA.ICRA0 + Ofset2) << 8; /* OCRDRD02、TCNT1D2 に対
    応するコンペアマッチ A の条件を設定*/

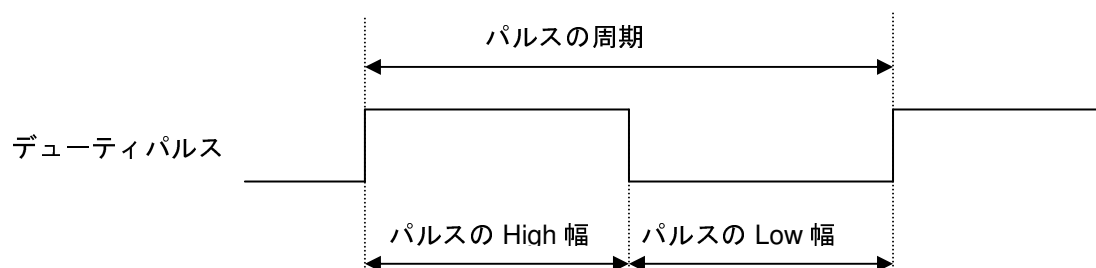
    /* ダウンカウントの長さを決定し、ターミネートの実行時間を設定 */
    ATUD.SUBBLOCKB[0].GRDRD[0] = (ATUA.ICRA0 + 0x3500) << 8; /* GRDRD00、TCNT2D0 に対
    応するコンペアマッチ B の条件を設定 */
    ATUD.SUBBLOCKB[0].GRDRD[1] = (ATUA.ICRA0 + 0x3500) << 8; /* GRDRD01、TCNT2D1 に対
    応するコンペアマッチ B の条件を設定 */
    ATUD.SUBBLOCKB[0].GRDRD[2] = (ATUA.ICRA0 + 0x3500) << 8; /* GRDRD02、TCNT2D2 に対
    応するコンペアマッチ B の条件を設定 */
}
```

```
/* ダウンカウントの長さを決定し、パルスの High 幅を設定 */  
ATUD.SUBBLOCKB[0].DCNTD[0] = Puls_w << 8;  
ATUD.SUBBLOCKB[0].DCNTD[1] = Puls_w << 8;  
ATUD.SUBBLOCKB[0].DCNTD[2] = Puls_w << 8;  
} /* End of function ICIA0() */
```

2.12 動作例 11 PWM 波形出力[タイマ E]

2.12.1 概要

図 2.21に示すように、デューティ及び周期を変化できるパルスを出力します。タイマ E のサイクルレジスタとデューティレジスタの設定を行うことで、パルス幅を設定します。



$$\text{デューティ} = \frac{\text{パルスの High 幅}}{\text{パルスの周期}} \times 100 (\%)$$

図 2.21 PWM 出力の概要図

2.12.2 使用機能説明

表 2.11に使用端子及び関連レジスタの機能割り付けを示します。

表 2.11 ATU-III機能割り付け

使用端子		機能
端子	TOE00	PWM 波形を出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ E レジスタ	DTRE00	デューティレジスタの値を設定します。
	DRLDE00	デューティリロードレジスタの値を設定します。
	CYLMRE00	サイクルレジスタの値を設定します。
	CRLDE00	サイクルリロードレジスタの値を設定します。
	RLDCRE0	DTRE のリロード機能を有効に設定します。
	TIERE0	サイクルマッチ割り込みを設定します。
	TSTRE	タイマ E 各サブブロックのカウンタの動作を設定します。
PFC レジスタ	PECR1	端子機能を設定します。
INTC レジスタ	IPR14	サイクルマッチ割り込みの優先度を設定します。

2.12.3 動作説明

図 2.22に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって PWM 波形を出力します。コンペアマッチの度にデューティリロードレジスタの値を更新することで、PWM 波形の Duty 比を変化させています。

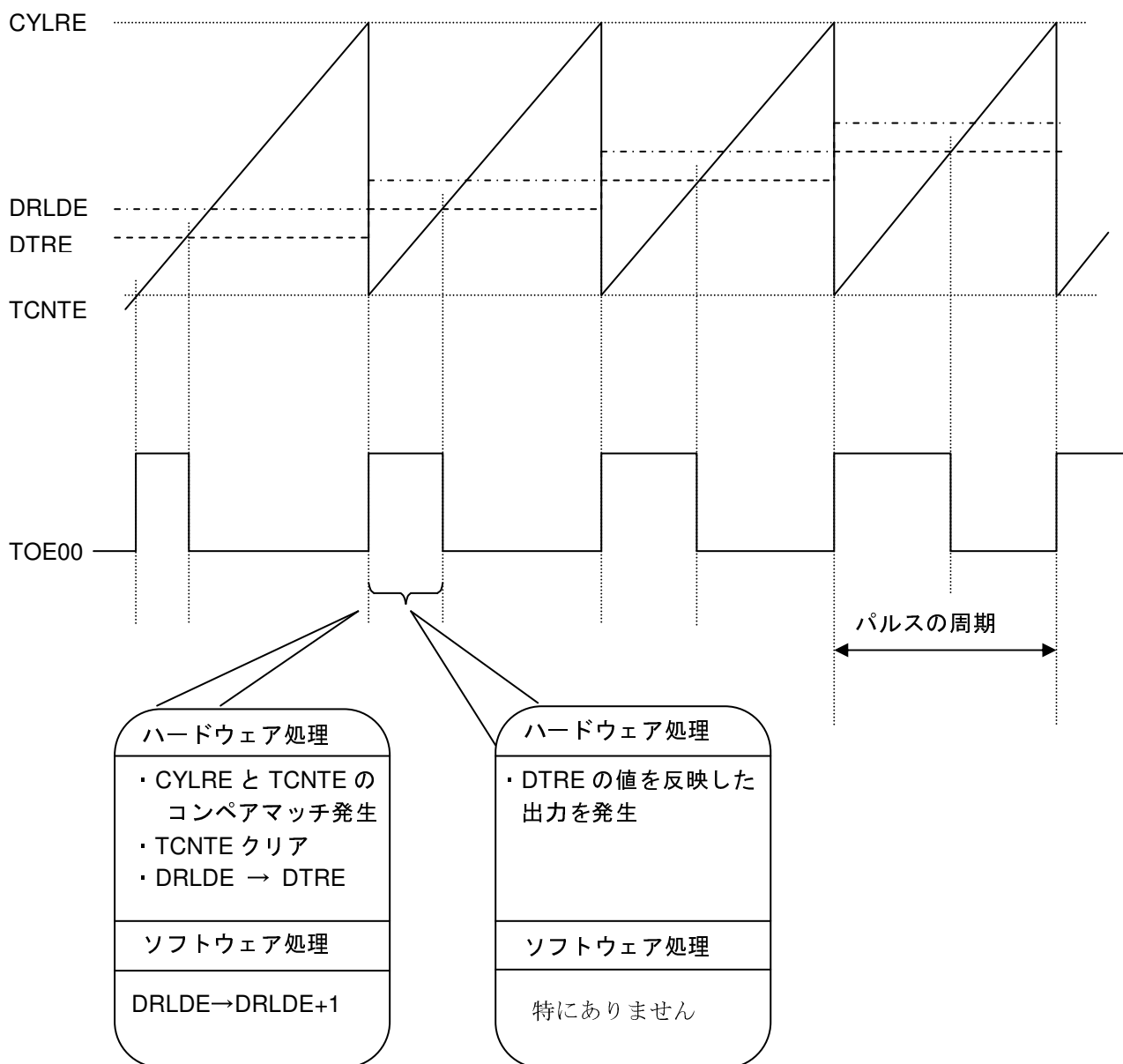


図 2.22 PWM 出力動作原理

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

2.12.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
サイクルマッチ 割り込み	CMIE00	CMFE00によって起動し、DRLDEの値を増減します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
up	サイクルマッチ割り込み発生時、デューティリロードレジスタの値が一定 以下/以上でセット/クリアされ、デューティリロードレジスタ値の増減方向を設定するフラグになります。	unsigned char	サイクルマッチ 割り込み

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	PE6 端子を TOE00 に設定します。	0x1000	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 2 に設定します。	0x0001	
ATUENR	ATU-IIIのタイマ E およびプリスケアラのカウント開始を許可します。	0x0021	
TSTRE	タイマ E サブブロック 0 の動作を許可します。	0x01	
SSTRE0	タイマ E サブブロック 0 タイマカウンタ 00 の動作を許可します。	0x01	
TIERE0	サイクルレジスタ 00 によるサイクルマッチ割り込みを許可する。	0x01	
RLDCRE0	DTRE00 のリロード機能を有効に設定する。	0x01	
DTRE00	PWM 波形のデューティを設定します。	0x0001	
CYLRE00	PWM 波形の周期を設定します。	0x0100	
DRLDE00	サイクルマッチ割り込み発生時にこのレジスタの値を DTRE にセットします。	0x0001	
CRLDE00	サイクルマッチ割り込み発生時にこのレジスタの値を CYLRE にセットします。	0x0100	
IPR14	CMIE0 の割り込み優先レベルを 15 に設定します。	0x00F0	メインルーチン

2.12.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : PWM wave output.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned char up;
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
    up = 0;

    /* ポートの設定 */
    /* Configure PECCR1
b15:14 PE7MD = 00      PE7 入出力 (ポート)
b13:12 PE6MD = 01      TOE00 出力 (ATU-III)
b11      Reserved
b10      PE5MD = 0      PE5 入出力 (ポート)
b9       Reserved
b8       PE4MD = 0      PE4 入出力 (ポート)
b7       Reserved
b6       PE3MD = 0      PE3 入出力 (ポート)
b5:4 PE2MD = 00      PE2 入出力 (ポート)
b3:2 PE1MD = 00      PE1 入出力 (ポート)
b1       Reserved
b0       PE0MD = 0      PE0 入出力 (ポート) */
    PORTE.CR1.WORD = 0x1000; /* PE6 を TOE00 出力に設定
    */

    /* ATU-IIIの設定 */
    /* Configure DTRE00
b15-0 DTREn[15:0] PWM のデューティ */
    ATUE.SUBBLOCK[0].DTRE[0] = 0x0001; /* デューティレジスタの値を 0x0001 に設定
    */

    /* Configure DRLDE00
b15-0 DRLDE [15:0] PWM のデューティ (リロード用) */
    ATUE.SUBBLOCK[0].DRLDE[0] = 0x0001; /* デューティリロードレジスタ値を 0x01 に設定
    */

    /* Configure CYLRE00
b15-0 CRLDE00[15:0] PWM の周期 */
    ATUE.SUBBLOCK[0].CYLRE[0] = 0x0100; /* サイクルレジスタの値を 0x00FF に設定
    */

    /* Configure DRLDE00
b15-0 CYLRE00[15:0] PWM の周期 (リロード用) */
    ATUE.SUBBLOCK[0].CRLDE[0] = 0x0100; /* サイクルリロードレジスタの値を 0x00FF に設定
    */

    /* Configure RLDCRE0
b7:4 Reserved
b3      RLDEn[3] = 0 サイクルマッチ時のデューティレジスタ、サイクルレジスタのリロード機能を
無効に設定
b2      RLDEn[2] = 0 サイクルマッチ時のデューティレジスタ、サイクルレジスタのリロード機能を
無効に設定
b1      RLDEn[1] = 0 サイクルマッチ時のデューティレジスタ、サイクルレジスタのリロード機能を
無効に設定
b0      RLDEn[0] = 1 サイクルマッチ時のデューティレジスタ、サイクルレジスタのリロード機能を
有効に設定 */
    ATUE.SUBBLOCK[0].RLDCRE.BYTE = 0x01; /* DTRE00 のリロード機能を有効に設定
    */

    /* Configure TIERE0
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b7:4 Reserved
b3      CMEE03 = 0      CMFE03 による割り込み要求を禁止
b2      CMEE02 = 0      CMFE02 による割り込み要求を禁止
b1      CMEE01 = 0      CMFE01 による割り込み要求を禁止
b0      CMEE00 = 1      CMFE00 による割り込み要求を許可 */
    ATUE.SUBBLOCK[0].TIERE.BYTE = 0x01; /* サイクルレジスタ 00 のサイクルマッチ発生による
    割り込みを許可 */

/* Configure TSTRE
b7:6 Reserved
b5      STRE5 = 0      サブブロック E5 の動作を禁止
b4      STRE4 = 0      サブブロック E4 の動作を禁止
b3      STRE3 = 0      サブブロック E3 の動作を禁止
b2      STRE2 = 0      サブブロック E2 の動作を禁止
b1      STRE1 = 0      サブブロック E1 の動作を禁止
b0      STRE0 = 1      サブブロック E0 の動作を許可 */
ATUE.TSTRE.BYTE = 0x01;          /* タイマ E サブブロック 0 の動作を許可
    */

/* Configure SSTORE0
b7:4 Reserved
b3      SSTORE03 = 0    サブブロック E0 のチャンネル 3 のカウント動作を停止
b2      SSTORE02 = 0    サブブロック E0 のチャンネル 2 のカウント動作を停止
b1      SSTORE01 = 0    サブブロック E0 のチャンネル 1 のカウント動作を停止
b0      SSTORE00 = 1    サブブロック E0 のチャンネル 0 のカウント動作を許可 */
ATUE.SUBBLOCK[0].SSTORE.BYTE = 0x01; /* タイマ E サブブロック 0 カウンタ E0 の動作を許
    可 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCR[9:0] = H'1    プリスケアラの分周比に 2 (設定値は-1) を設定 */
ATUCTRL.PSCR0.WORD = 0x0001;      /* プリスケアラの分周比を 2 に設定
    */

/* Configure IPR14
b15:12 Reserved
b11:8  Reserved
b7:4 ATU-E0 (CMIE00:03) = H'F    対応する割り込みの優先順位を 15 に設定
b3:0 ATU-E1 (CMIE10-13) = H'0    対応する割り込みの優先順位を 0 に設定 */
INTC.IPR14.WORD = 0x00F0;        /* CMIE00 の割り込み優先順位を 15 に設定
    */

set_imask(0);                    /* 割り込みマスクレベルを 0 に設定
    */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0          タイマ J のカウント動作を停止
b8      THE = 0          タイマ H のカウント動作を停止
b7      TGE = 0          タイマ G のカウント動作を停止
b6      TFE = 0          タイマ F のカウント動作を停止
b5      TEE = 1          タイマ E のカウント動作を許可
b4      TDE = 0          タイマ D のカウント動作を停止
b3      TCE = 0          タイマ C のカウント動作を停止
b2      TBE = 0          タイマ B のカウント動作を停止
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b1      TAE = 0          タイマ A のカウント動作を停止
b0      PSCE = 1       プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0021;      /* タイマ E およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : CMIE00
* Description   : サイクルマッチ割り込み
* Arguments    : none
* Return Value  : none
*****/
void CMIE00(void)
{
  /* Configure TSRE0
b7      OVFE03 = 0      オーバフローフラグをクリア
b6      OVFE02 = 0      オーバフローフラグをクリア
b5      OVFE01 = 0      オーバフローフラグをクリア
b4      OVFE00 = 0      オーバフローフラグをクリア
b3      CMFE03 = 0      サイクルマッチフラグをクリア
b2      CMFE02 = 0      サイクルマッチフラグをクリア
b1      CMFE01 = 0      サイクルマッチフラグをクリア
b0      CMFE00 = 0      サイクルマッチフラグをクリア */
  ATUE.SUBBLOCK[0].TSRE.BYTE &= 0x00;      /* サイクルマッチフラグをクリア
*/

  if(up == 0)          /* up フラグがセットされていないならば
*/
  {
    ATUE.SUBBLOCK[0].DRLDE[0]--;          /* デューティサイクルレジスタの値を減少
*/
  }
  else                /* up フラグがセットされているならば          */
  {
    ATUE.SUBBLOCK[0].DRLDE[0]++;          /* デューティサイクルレジスタの値を増加
*/
  }

  if(ATUE.SUBBLOCK[0].DRLDE[0] >= 0x00FF) /* デューティサイクルレジスタの値が
0x00FC 以上ならば */
  {
    up = 0;          /* up フラグをクリア          */
  }
  if(ATUE.SUBBLOCK[0].DRLDE[0] <= 0x0001) /* デューティサイクルレジスタの値が
0x0005 以下ならば */
  {
    up = 1;          /* up フラグをセット          */
  }
} /* End of function CMIE00() */
```

2.13 動作例 12 有効エッジ入力間隔[タイマ F]

2.13.1 概要

図 2.23に示すように、設定回数の有効なエッジが入力されるまでの時間を計測します。タイマの動作モードを有効エッジ入力間隔計測に設定することで、有効エッジ数を設定します。

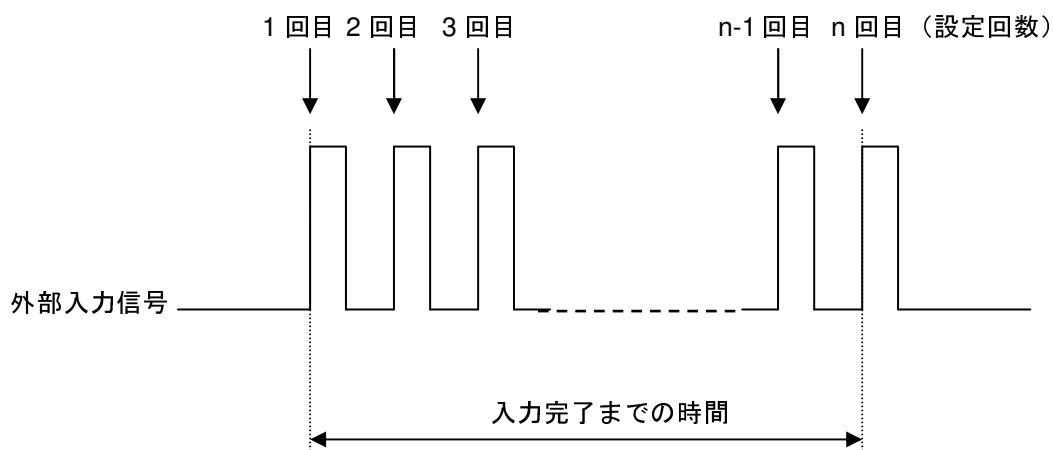


図 2.23 有効エッジ入力間隔のタイミング（立ち上がりエッジのカウント時）

2.13.2 使用機能説明

表 2.12に使用端子及び関連レジスタの機能割り付けを示します。

表 2.12 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ0の分周比を設定します。
タイマF レジスタ	TIERF0	ICIF0 割り込みを許可します。
	TCRF0	タイマF各サブブロックの動作を設定します。
	GRBF0	検出するエッジの回数を設定します。
	ECNTBF0	有効な外部信号の回数を記録します。
	TSRF0	タイマFのフラグ状態が格納されます。
	TSTRF	タイマF各サブブロックのカウンタの動作を設定します。
INTCレジスタ	IPR16	ICIF0の割り込みの優先度を設定します。
PFCレジスタ	PHCR	端子機能を設定します。

2.13.3 動作説明

図 2.24に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって有効エッジ入力の時間を計測します。本例では検出するエッジの回数を 5 回に設定し、5 回検出後にカウンタの値を記録しています。

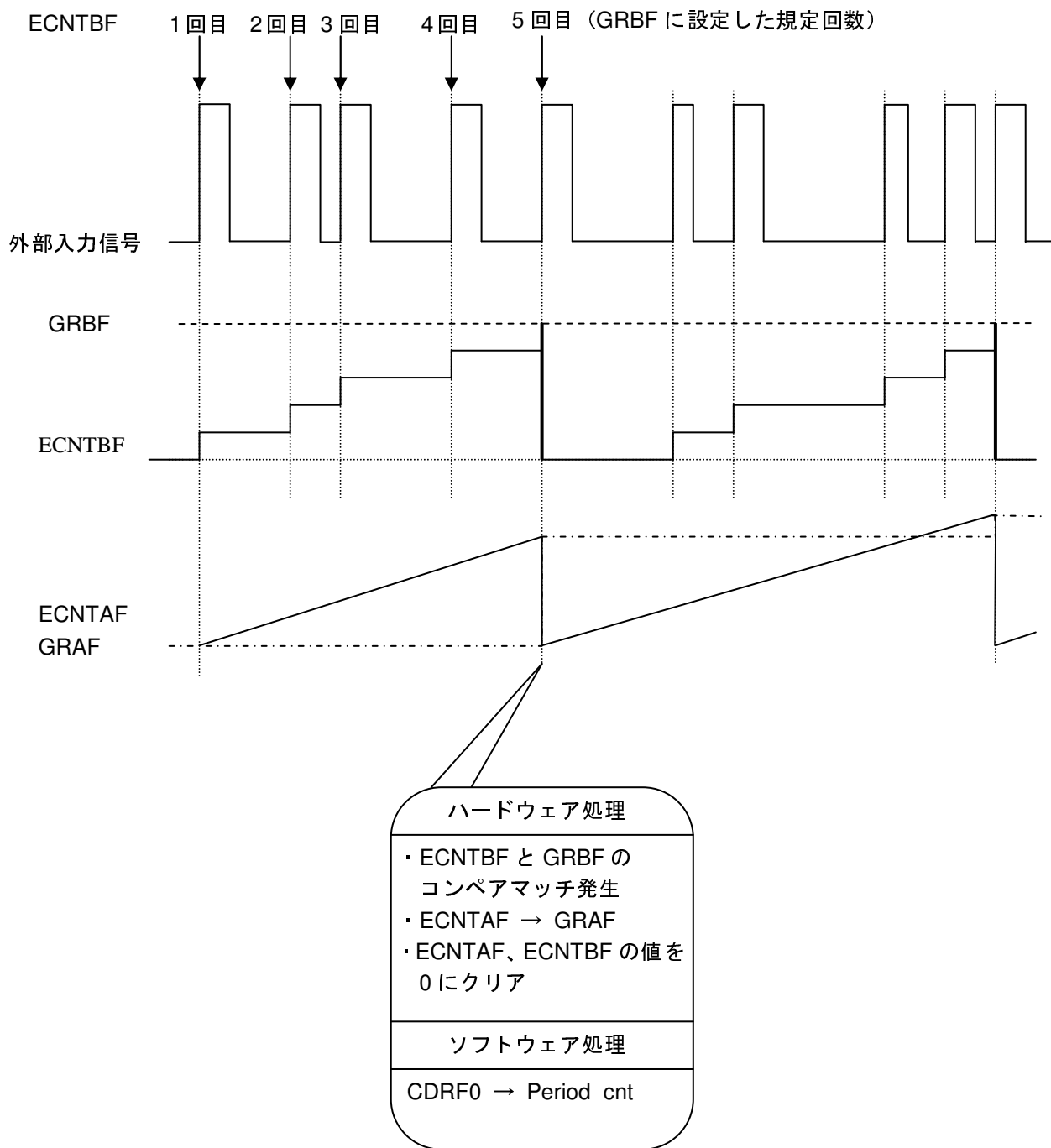


図 2.24 有効エッジ入力間隔計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.13.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
計測値の保存ルーチン	ICIF0	計測した値を変数に保存します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
Period_cnt	計測した規定数のエッジ入力に要した期間を保存します。	unsigned long	計測値の保存ルーチン

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0 端子を TIF0A に設定します。	0x0003	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウント開始を許可します。	0x0041	
TIERF0	割り込みを許可します。	0x01	
IPR16	割り込みの優先順位を設定します。	0xF000	
TSTRF	タイマ F サブブロック 0 のカウントをスタートさせる。	0x00000001	
TCRF0	サブブロック 0 を「クロックバス 0 でカウント」、「有効エッジ入力間隔計測」「計測エッジは立ち上がり」に設定します。	0x05	
GRBF0	検出するエッジの回数を 5 回に設定します。	0x0005	
ECNTAF0	TCRF で設定したクロックバスでアップカウントを行います。	—	
ECNTBF0	外部入力の立ち上がりエッジの個数をカウントします。	0x0000	

2.13.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Active edge input interval.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long Period_cnt;
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments     : none
 * Return Value  : none
 */

```

```

void main(void)
{
  /* ポートの設定 */
  /* Configure PHCR
  b15:12 Reserved
  b11:10 PH5MD[1:0] = 00   PH5 入出力 (ポート)
  b9:8 PH4MD[1:0] = 00   PH4 入出力 (ポート)
  b7:6 PH3MD[1:0] = 00   PH3 入出力 (ポート)
  b5:4 PH2MD[1:0] = 00   PH2 入出力 (ポート)
  b3:2 PH1MD[1:0] = 00   PH1 入出力 (ポート)
  b1:0 PH0MD[1:0] = 11   TIF0A 入力 (ATU-III) */
  PORTH.CR.WORD = 0x0003;          /* PH0 を TIF0A(エッジ入力用端子)に設定
  */

  /* ATU-III (有効エッジ入力間隔計測) の設定 */
  /* Configure TCRF0
  b7:5 CKSELF0[2:0] = 000   クロックバス 0
  b4:2 MDF0[2:0] = 001     有効エッジ入力間隔計測
  b1:0 EGSELFn[1:0] = 01   立ち上がりエッジ */
  ATUF.SUBBLOCK[0].TCRF.BYTE= 0x05; /* サブブロック 0 の時間計測カウンタのクロックソース
  を「クロックバス 0」,
                                     動作モードを「有効エッジ入力間隔計測」,
                                     計測エッジを「立上りエッジ」に それぞれ設定      */

  /* Configure GRBF0
  b0      GRBF0[15:0]      アウトプットコンペア値 */
  ATUF.SUBBLOCK[0].GRBF = 0x0005; /* 検出するエッジの回数を 5 回に設定      */

  /* Configure ECNTBF0
  b0      ECNTBF0[15:0]   イベントカウンタ */
  ATUF.SUBBLOCK[0].ECNTBF = 0x0000; /* イベントカウンタの初期値を 0 に設定      */

  /* Configure TIERF0
  b7:4 Reserved
  b3      OVECF0 = 0      OVFCF0 による割り込みを禁止
  b2      OVEBF0 = 0      OVFBF0 による割り込みを禁止
  b1      OVEAF0 = 0      OVFAF0 による割り込みを禁止
  b0      ICEF0 = 1      ICFF0 による割り込みを許可 */
  ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01; /* TIF0 のインプットキャプチャ割り込み(ICIF0)
  を許可      */

  /* Configure TSTRF
  b0      STRF0 = 1      ECNTAF0、ECNTBF0、および ECNTCF0 のカウント動作を許可 */
  ATUF.TSTRF.BIT.STRF0 = 1;          /* サブブロック 0 のカウントをスタート      */

  /* Configure PSCR0
  b15:10 Reserved
  b9:0 PSC0 = H'9      プリスケーラの分周比を 10 (設定値は-1)に設定 */
  ATUCTRL.PSCR0.WORD = 0x0009;      /* プリスケーラの分周比を 10 に設定      */

  /* Configure IPR16
  b15:12 ATU-F (ICIF0~ICIF3) = H'F      対応する割り込みの優先順位を 15 に設定
  b11:8  ATU-F (ICIF4~ICIF7) = H'0      対応する割り込みの優先順位を 0 に設定
  b7:4  ATU-F (ICIF8~ICIF11) = H'0      対応する割り込みの優先順位を 0 に設定
  b3:0  ATU-F (ICIF12~ICIF15) = H'0     対応する割り込みの優先順位を 0 に設定 */

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
INTC.IPR16.WORD = 0xF000;          /* ICIF0 の割り込み優先順位を 15 に設定          */

set_imask(0);                      /* 割り込みマスクレベルを 0 に設定          */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 0      タイマ J のカウント動作を停止
b8      THE = 0      タイマ H のカウント動作を停止
b7      TGE = 0      タイマ G のカウント動作を停止
b6      TFE = 1      タイマ F のカウント動作を許可
b5      TEE = 0      タイマ E のカウント動作を停止
b4      TDE = 0      タイマ D のカウント動作を停止
b3      TCE = 0      タイマ C のカウント動作を停止
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 0      タイマ A のカウント動作を停止
b0      PSCE = 1     プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0041;      /* タイマ F およびプリスケアラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value : none
*****/
void ICIF0(void)
{
    Period_cnt = ATUF.SUBBLOCK[0].CDRF; /* 計測した値を変数 Period_cnt に保存(GRAF0)
*/

    /* Configure TSRF0
b7:4 Reserved
b3      OVFCF0 = 0      ECNTCFn にオーバーフローなし
b2      OVFBF0 = 0      ECNTBFn にオーバーフロー/アンダフローなし
b1      OVFAF0 = 0      ECNTAFn にオーバーフローなし
b0      ICF0 = 0       サブブロック Fn にインพุットキャプチャの検出なし */
    ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア
*/
} /* End of function ICIF0() */
```

2.14 動作例 13 一定時間内エッジカウント[タイマ F]

2.14.1 概要

図 2.25に示すように、一定時間内の有効な入力エッジの回数を記録します。タイマ F の動作モードは、一定期間内エッジカウントに設定します。

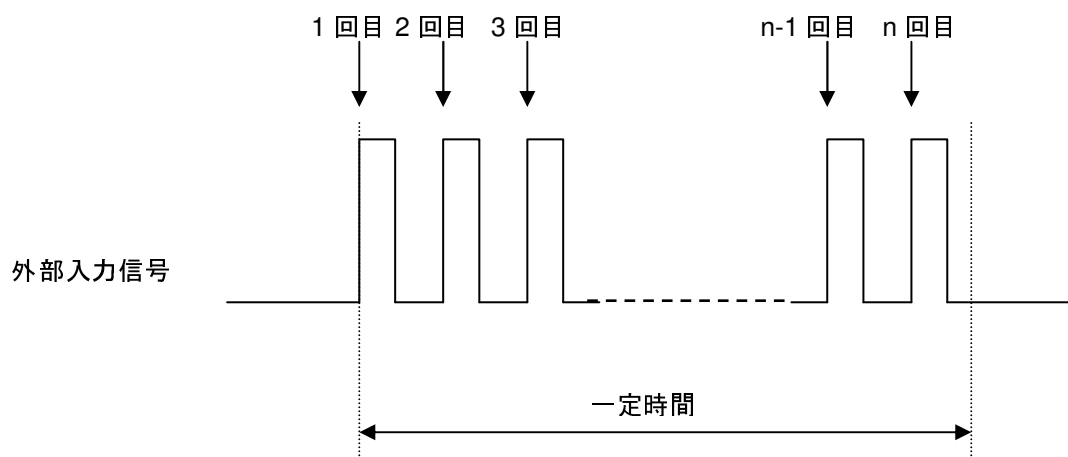


図 2.25 一定時間内エッジカウントの概要（立ち上がりエッジのカウント時）

2.14.2 使用機能説明

表 2.13に使用端子及び関連レジスタの機能割り付けを示します。

表 2.13 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケーラの動作を設定します。
	PSCR0	プリスケーラ 0 の分周比を設定します。
タイマ F レジスタ	TIERF0	割り込みの許可をします。
	TCRF0	タイマ F 各サブブロックの動作を設定します。
	GRAF0	エッジを検出する期間を設定します。
	ECNTAF0	TCRF で設定したクロックバスでアップカウントを行います。
	ECNTBF0	有効な外部信号の回数を記録します。
	TSRF0	タイマ F サブブロック 0 の状態が格納されます。
	TSTRF	タイマ F 各サブブロックのカウンタの動作を設定します。
INTC レジスタ	IPR16	割り込みの優先順位を設定します。
PFC レジスタ	PHCR	端子機能を設定します。

2.14.3 動作説明

図 2.26に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって一定時間内の有効エッジ入力の回数を計測します。

エッジ回数の計測期間は、時間計測カウンタレジスタで設定します。

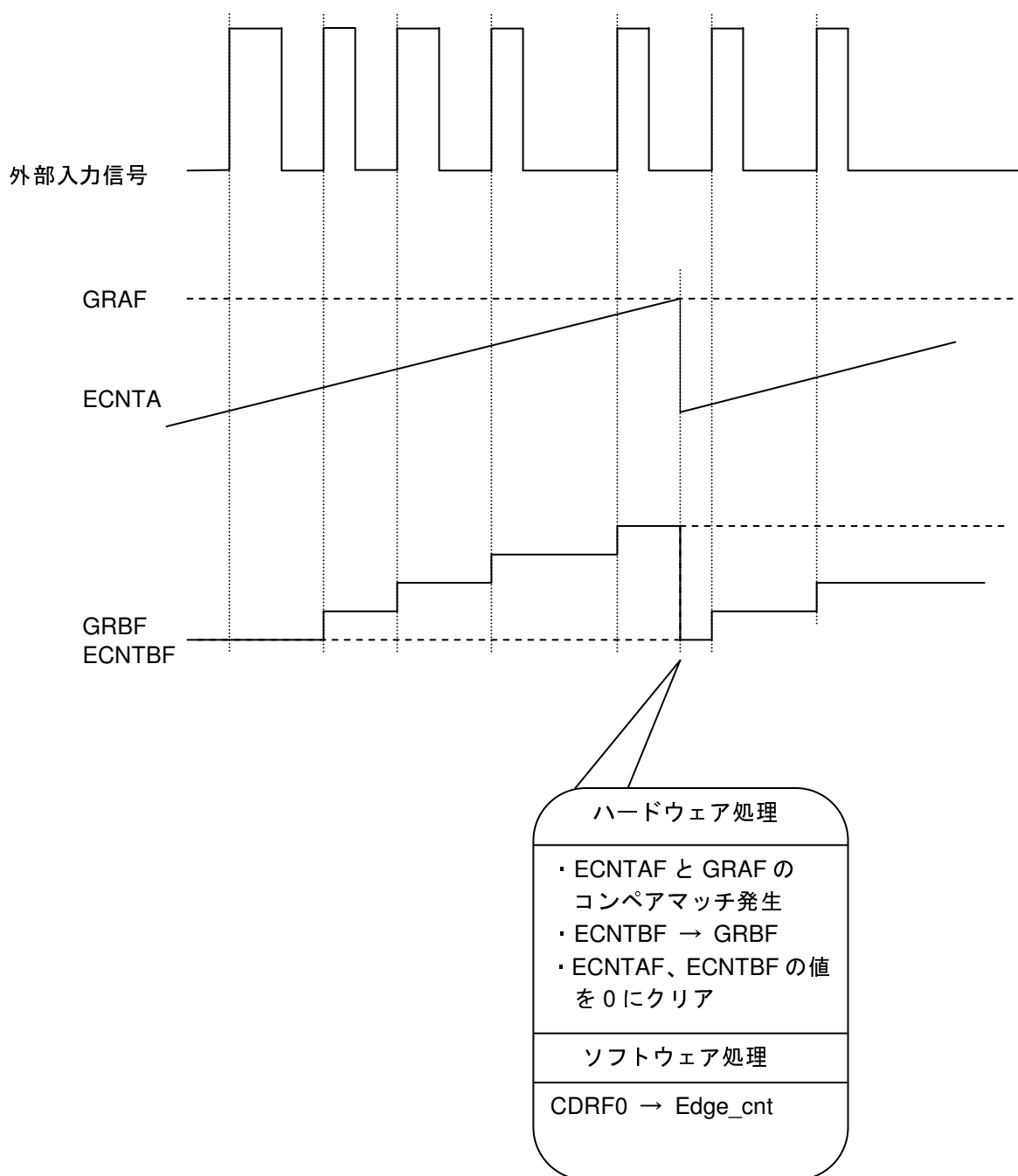


図 2.26 有効エッジ入力間隔計測動作原理

2.14.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
計測値の保存ルーチン	ICIF0	計測した値を変数に保存します。

● 使用変数の説明

ラベル名	機能	データ長	使用モジュール
Edge_cnt	計測したエッジ数を保存します	unsigned long	計測値の保存ルーチン

● 使用レジスタの設定

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0 端子を TIF0A に設定します。	0x0003	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウンタ開始を許可します。	0x0041	
IPR16	割り込みの優先順位を設定します。	0xF000	
TIERF0	割り込みの許可をします。	0x01	
TSTRF	タイマ F サブブロック 0 のカウンタをスタートさせる。	0x00000001	
TCRF0	サブブロック 0 を「クロックバス 0 でカウンタ」、「一定期間内エッジカウンタ」「計測エッジは立ち上がり」に設定します。	0x01	
GRAF0	エッジを計測する期間を 0x50 カウンタに設定します。	0x00005000	計測値の保存ルーチン
TSRF0	タイマ F のインプットキャプチャフラグをクリアします。	0x00	

2.14.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Edge counts within a certain period .
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long Edge_cnt;
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```

void main(void)
{
    /* ポートの設定 */
    /* Configure PHCR
    b15:12 Reserved
    b11:10 PH5MD[1:0] = 00    PH5 入出力 (ポート)
    b9:8 PH4MD[1:0] = 00    PH4 入出力 (ポート)
    b7:6 PH3MD[1:0] = 00    PH3 入出力 (ポート)
    b5:4 PH2MD[1:0] = 00    PH2 入出力 (ポート)
    b3:2 PH1MD[1:0] = 00    PH1 入出力 (ポート)
    b1:0 PH0MD[1:0] = 11    TIF0A 入力 (ATU-III) */
    PORTH.CR.WORD = 0x0003;          /* PH0 を TIF0A(エッジ入力用端子)に設定      */

    /* ATU-III (一定期間内エッジカウント)の設定 */
    /* Configure TCRF0
    b7:5 CKSELF0[2:0] = 000    クロックバス 0
    b4:2 MDF0[2:0] = 000      一定時間内エッジカウント
    b1:0 EGSELFn[1:0] = 01    立ち上がりエッジ */
    ATUF.SUBBLOCK[0].TCRF.BYTE= 0x01; /* サブブロック 0 の時間計測カウンタの設定: クロック
    バス 0
                                動作モード: B'000 = 一定期間内エッジカウント
                                計測エッジ: B'01 = 立上りエッジ      */

    /* Configure GRAF0
    b31:8  GRAF0[23:0]    時間計測カウンタ A のインプットキャプチャ値
    b7:0 Reserved */
    ATUF.SUBBLOCK[0].GRAF = 0x00005000; /* エッジ計測期間を 0x50 カウントに設定      */

    /* Configure TIERF0
    b7:4 Reserved
    b3      OVECF0 = 0      OVFCF0 による割り込みを禁止
    b2      OVEBF0 = 0      OVFBF0 による割り込みを禁止
    b1      OVEAF0 = 0      OVFAF0 による割り込みを禁止
    b0      ICEF0 = 1      ICFF0 による割り込みを許可 */
    ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01; /* ICEF0 セットによる割り込み許可      */

    /* Configure TSTRF
    b0      STRF0 = 1      ECNTAF0、ECNTBF0、および ECNTCF0 のカウント動作を許可 */
    ATUF.TSTRF.BIT.STRF0 = 1;          /* サブブロック 0 のカウントをスタート      */

    /* Configure PSCR0
    b15:10 Reserved
    b9:0 PSCn = H'9    プリスケーラの分周比を 10 (設定値は-1)に設定 */
    ATUCTRL.PSCR0.WORD = 0x0009;      /* プリスケーラの分周比を 10 に設定      */

    /* Configure IPR16
    b15:12 ATU-F (ICIF0~ICIF3) = H'F    対応する割り込みの優先順位を 15 に設定
    b11:8  ATU-F (ICIF4~ICIF7) = H'0    対応する割り込みの優先順位を 0 に設定
    b7:4  ATU-F (ICIF8~ICIF11) = H'0    対応する割り込みの優先順位を 0 に設定
    b3:0  ATU-F (ICIF12~ICIF15) = H'0    対応する割り込みの優先順位を 0 に設定 */
    INTC.IPR16.WORD = 0xF000;          /* ICIF0 の割り込み優先順位を 15 に設定      */

    set_imask(0);                      /* 割り込みマスクレベルを 0 に設定      */

    /* Configure ATUENR

```

```
b15:10 Reserved
b9      TJE = 0      タイマ J のカウント動作を停止
b8      THE = 0      タイマ H のカウント動作を停止
b7      TGE = 0      タイマ G のカウント動作を停止
b6      TFE = 1      タイマ F のカウント動作を許可
b5      TEE = 0      タイマ E のカウント動作を停止
b4      TDE = 0      タイマ D のカウント動作を停止
b3      TCE = 0      タイマ C のカウント動作を停止
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 0      タイマ A のカウント動作を停止
b0      PSCE = 1     プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0041;      /* タイマ F およびプリスケーラのカウントをスタート*/

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value : none
*****/
void ICIF0(void)
{
    /* Configure CDRF0
    b31:8  CDRFn[23:0]      GRBF0 のデータ
    b7:0 Reserved */
    Edge_cnt = ATUF.SUBBLOCK[0].CDRF; /* 計測した値を変数 Edge_cnt に保存 (GRBF0) */

    /* Configure TSRF0
    b7:4 Reserved
    b3      OVFCF0 = 0      オーバフローフラグをクリア
    b2      OVFBF0 = 0      オーバフロー/アンダフローフラグをクリア
    b1      OVFAF0 = 0      オーバフローフラグをクリア
    b0      ICFF0 = 0      インพุットキャプチャの検出フラグをクリア */
    ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */
} /* End of function ICIF0() */
```

2.15 動作例 14 入力 High/Low 期間計測[タイマ F]

2.15.1 概要

図 2.27に示すように、入力パルス内規定回数内での High（または Low）レベル出力の時間を記録します。本動作例では、タイマ F の動作モードは入力 High/Low 期間計測モードに設定し、入力パルスの High 期間を計測対象とします。

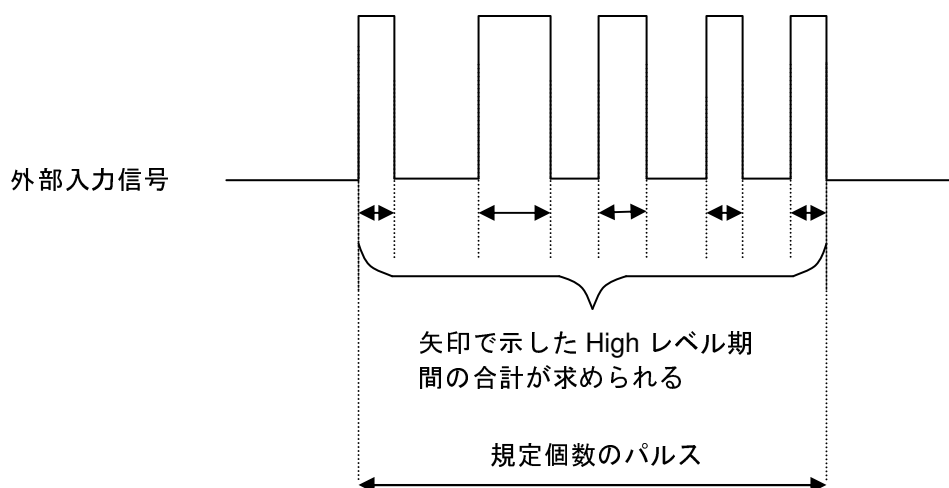


図 2.27 High/Low 期間計測の概要

2.15.2 使用機能説明

表 2.14に使用端子及び関連レジスタの機能割り付けを示します。

表 2.14 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ F レジスタ	TIERF0	割り込みの許可をします。
	TCRF0	タイマ F 各サブブロックの動作を設定します。
	GRBF0	検出するパルスの個数を設定します。
	ECNTAF0	設定したレベルが入力されている期間をカウントします。
	ECNTBF0	有効な外部信号の回数を記録します。
INTC レジスタ	TSTRF	タイマ F 各サブブロックのカウンタの動作を設定します。
INTC レジスタ	IPR16	割り込み優先順位を設定します。
PFC レジスタ	PHCR	端子機能を設定します。

2.15.3 動作説明

図 2.28に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって一定時間内の有効エッジ入力の回数を計測します。入力ハイ/ローレベル期間計測モードでは、検出するパルスの回数を設定し、設定回数検出完了後にインプットキャプチャ割り込みを行います。この時のカウンタの値を用いることで期間計測します。

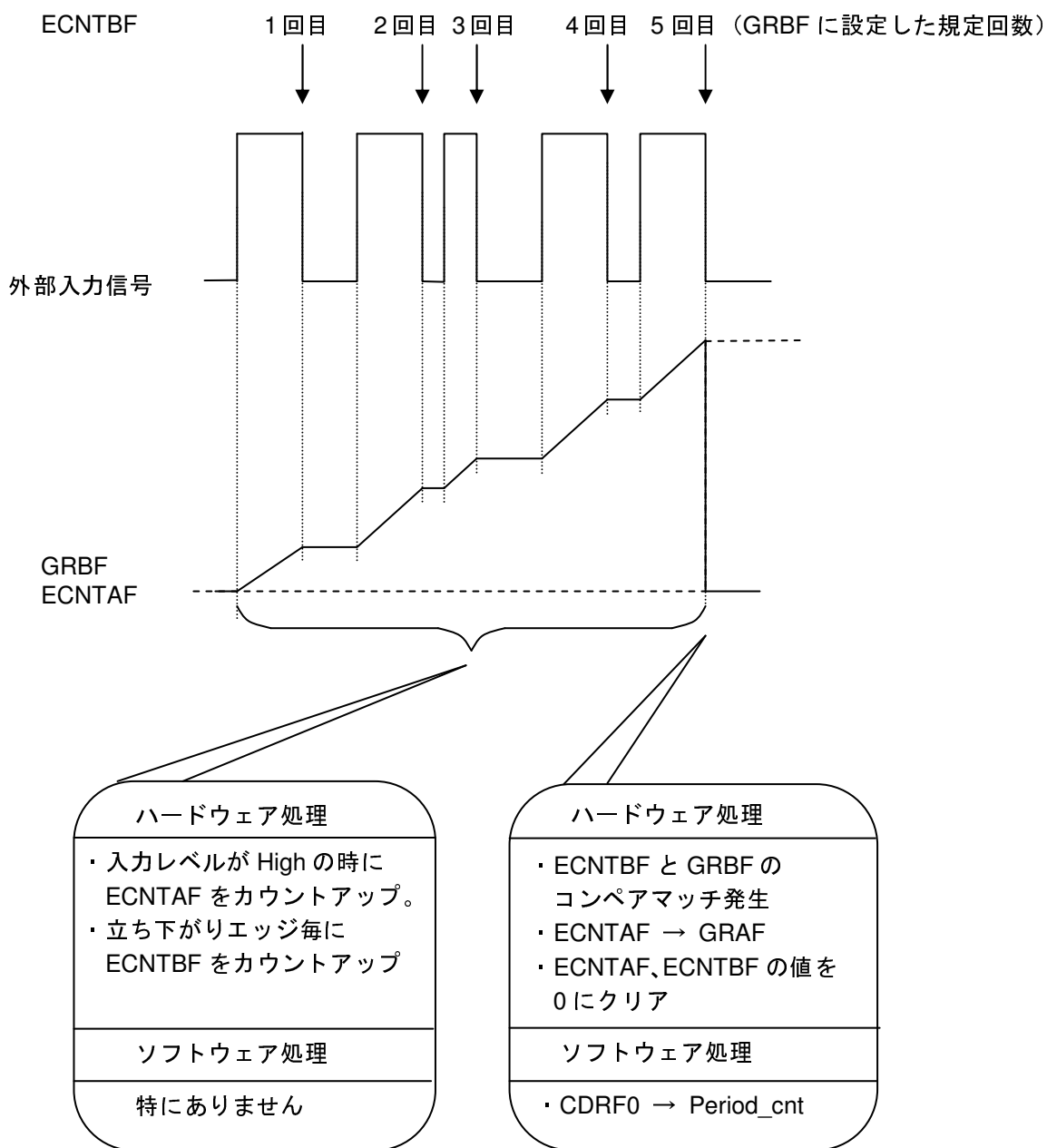


図 2.28 有効エッジ入力間隔計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.15.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
計測値の保存ルーチン	ICIF0	ICIF0 セットで割り込みを発生し、計測した値を変数で保存します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
Period_cnt	計測した High(または Low)期間の合計値を保存します	unsigned long	計測値の保存ルーチン

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0 端子を TIF0A に設定します。	0x0003	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
IPR16	割り込みの優先順位を設定します。	0xF000	
TIERF0	割り込みの許可をします。	0x01	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウンタ開始を許可します。	0x0041	
TSTRF	タイマ F サブブロック 0 のカウンタ動作を許可します。	STRF0 = 1	
TCRF0	サブブロック 0 を「クロックバス 0 でカウンタ」、「入力 High/Low 期間計測」「計測期間は High」に設定します。	0x09	
GRBF0	計測パルスの回数を 5 回に設定します。	0x0005	
TSRF0	タイマ F サブブロック 0 のインプットキャプチャフラグをクリアします。	0x00	

2.15.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Measurement of High / Low Input period.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long Period_cnt;
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
    /* ポートの設定 */
    /* Configure PHCR
    b15:12 Reserved
    b11:10 PH5MD[1:0] = 00    PH5 入出力 (ポート)
    b9:8 PH4MD[1:0] = 00    PH4 入出力 (ポート)
    b7:6 PH3MD[1:0] = 00    PH3 入出力 (ポート)
    b5:4 PH2MD[1:0] = 00    PH2 入出力 (ポート)
    b3:2 PH1MD[1:0] = 00    PH1 入出力 (ポート)
    b1:0 PH0MD[1:0] = 11    TIF0A 入力 (ATU-III) */
    PORTH.CR.WORD = 0x0003;          /* PH0 を TIF0A に設定 (外部信号の入力端子)
    */

    /* ATU-III (入力ハイ/ロー期間計測) の設定 */
    /* Configure TCRF0
    b7:5 CKSELF0[2:0] = 000    クロックバス 0
    b4:2 MDF0[2:0] = 010    入力ハイ/ローレベル期間計測
    b1:0 EGSELFn[1:0] = 01    立ち上がりエッジ */
    ATUF.SUBBLOCK[0].TCRF.BYTE= 0x09;    /* 時間計測カウンタのクロックソースを「クロック
    バス 0」,
    サブブロック 0 の動作モードを「入力ハイ/ロー期間計測」,
    計測出力を「ハイ期間計測」に それぞれ設定    */

    /* Configure GRBF0
    b15:0 GRBF0[15:0]    イベントカウンタのインプットキャプチャ値 */
    ATUF.SUBBLOCK[0].GRBF = 0x0005;    /* 検出するパルスの回数を 5 回に設定
    */

    /* Configure TIERF0
    b7:4 Reserved
    b3 OVECF0 = 0    OVFCF0 による割り込みを禁止
    b2 OVEBF0 = 0    OVFBF0 による割り込みを禁止
    b1 OVEAF0 = 0    OVFAF0 による割り込みを禁止
    b0 ICEF0 = 1    ICFF0 による割り込みを許可 */
    ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01;    /* ICEF0 セットによる割り込みを許可
    */

    /* Configure TSTRF
    b0 STRF0 = 1    ECNTAF0、ECNTBF0、および ECNTCF0 のカウント動作を許可 */
    ATUF.TSTRF.BIT.STRF0 = 1;    /* サブブロック 0 のカウントをスタート    */

    /* Configure PSCR0
    b15:10 Reserved
    b9:0 PSC0 = H'9    プリスケーラの分周比を 10 (設定値は-1) に設定 */
    ATUCTRL.PSCR0.WORD = 0x0009;    /* プリスケーラの分周比を 10 に設定    */

    /* Configure IPR16
    b15:12 ATU-F (ICIF0~ICIF3) = H'F    対応する割り込みの優先順位を 15 に設定
    b11:8 ATU-F (ICIF4~ICIF7) = H'0    対応する割り込みの優先順位を 0 に設定
    b7:4 ATU-F (ICIF8~ICIF11) = H'0    対応する割り込みの優先順位を 0 に設定
    b3:0 ATU-F (ICIF12~ICIF15) = H'0    対応する割り込みの優先順位を 0 に設定 */
    INTC.IPR16.WORD = 0xF000;    /* ICIF0 の割り込み優先順位を 15 に設定    */

    set_imask(0);    /* 割り込みマスクレベルを 0 に設定    */
}
```

```
/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0     タイマ J のカウント動作を停止
b8     THE = 0     タイマ H のカウント動作を停止
b7     TGE = 0     タイマ G のカウント動作を停止
b6     TFE = 1     タイマ F のカウント動作を許可
b5     TEE = 0     タイマ E のカウント動作を停止
b4     TDE = 0     タイマ D のカウント動作を停止
b3     TCE = 0     タイマ C のカウント動作を停止
b2     TBE = 0     タイマ B のカウント動作を停止
b1     TAE = 0     タイマ A のカウント動作を停止
b0     PSCE = 1   プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0041;      /* タイマ F およびプリスケーラのカウントをスター
ト */

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value : none
*****/
void ICIF0(void)
{
    /* Configure CDRF0
b31:8  CDRFn[23:0]   GRAF0 のデータ
b7:0 Reserved */
    Period_cnt = ATUF.SUBBLOCK[0].CDRF;      /* 変数 Period_cnt に計測したエッジ数を保存
    */

    /* Configure TSRF0
b7:4 Reserved
b3     OVFCF0 = 0     オーバフローフラグをクリア
b2     OVFBF0 = 0     オーバフロー/アンダフローフラグをクリア
b1     OVFAF0 = 0     オーバフローフラグをクリア
b0     ICFF0 = 0     インพุットキャプチャの検出フラグをクリア */
    ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00;      /* ICEF0 を 0 にクリア
    */
} /* End of function ICIF0() */
```

2.16 動作例 15 PWM 入力波形計測[タイマ F]

2.16.1 概要

図 2.29に示すように、入力される一定回数の PWM 波形についてオフデューティと PWM サイクルを計測します。タイマ F の動作モードは、PWM 入力波形計測モードに設定します。

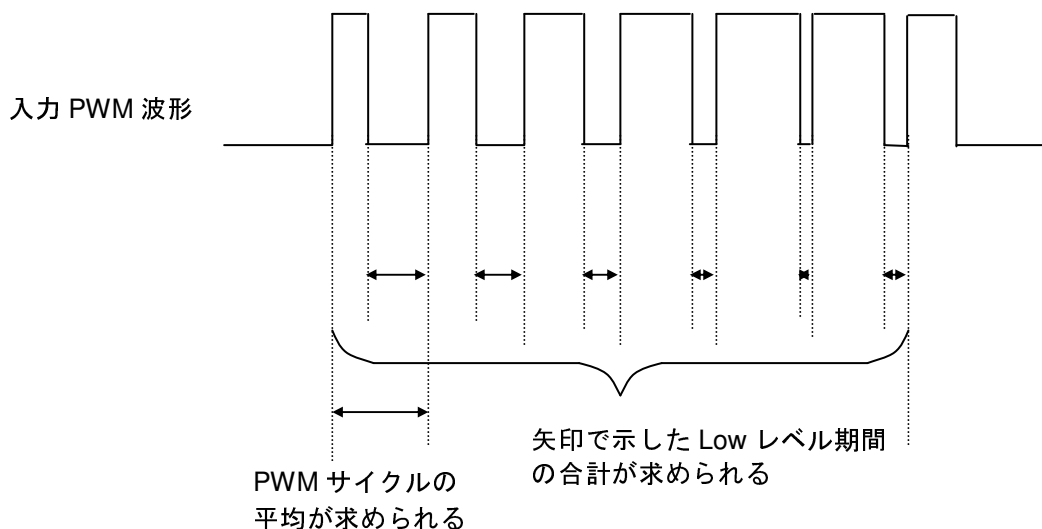


図 2.29 PWM 入力波形計測の概要

2.16.2 使用機能説明

表 2.15に使用端子及び関連レジスタの機能割り付けを示します。

表 2.15 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ F レジスタ	TCRF0	タイマ F 各サブブロックの動作を設定します。
	GRBF0	検出するパルスの個数を設定します。
	ECNTAF0	設定したレベルが入力されている期間をカウントします。
	ECNTBF0	有効な外部信号の回数を記録します。
	ECNTCF0	外部入力のエッジの個数をカウントします。
	TSTRF	タイマ F 各サブブロックのカウンタの動作を設定します。
	TSRF0	タイマ F サブブロック 0 の状態が格納されます。
	CDRF0	タイマ F キャプチャ出力レジスタです。
	GRCF0	PWM 波形のサイクルの値を保存
	TIERF0	タイマステータスレジスタ F (TSRF) のステータスフラグに対応する割り込みを許可するか禁止するか指定します。
INTC レジスタ	IPR16	割り込み優先順位を設定します。
PFC レジスタ	PHCR	端子機能を設定します。

2.16.3 動作説明

図 2.30に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって PWM 波形のオフデューティと、PWM サイクルを計測します。

設定された数の入力 PWM 波形を検出するとインプットキャプチャ割り込みが発生し、計測開始からのカウンタ値と入力 PWM 波形 OFF 時のみ更新されたカウンタ値を得ます。

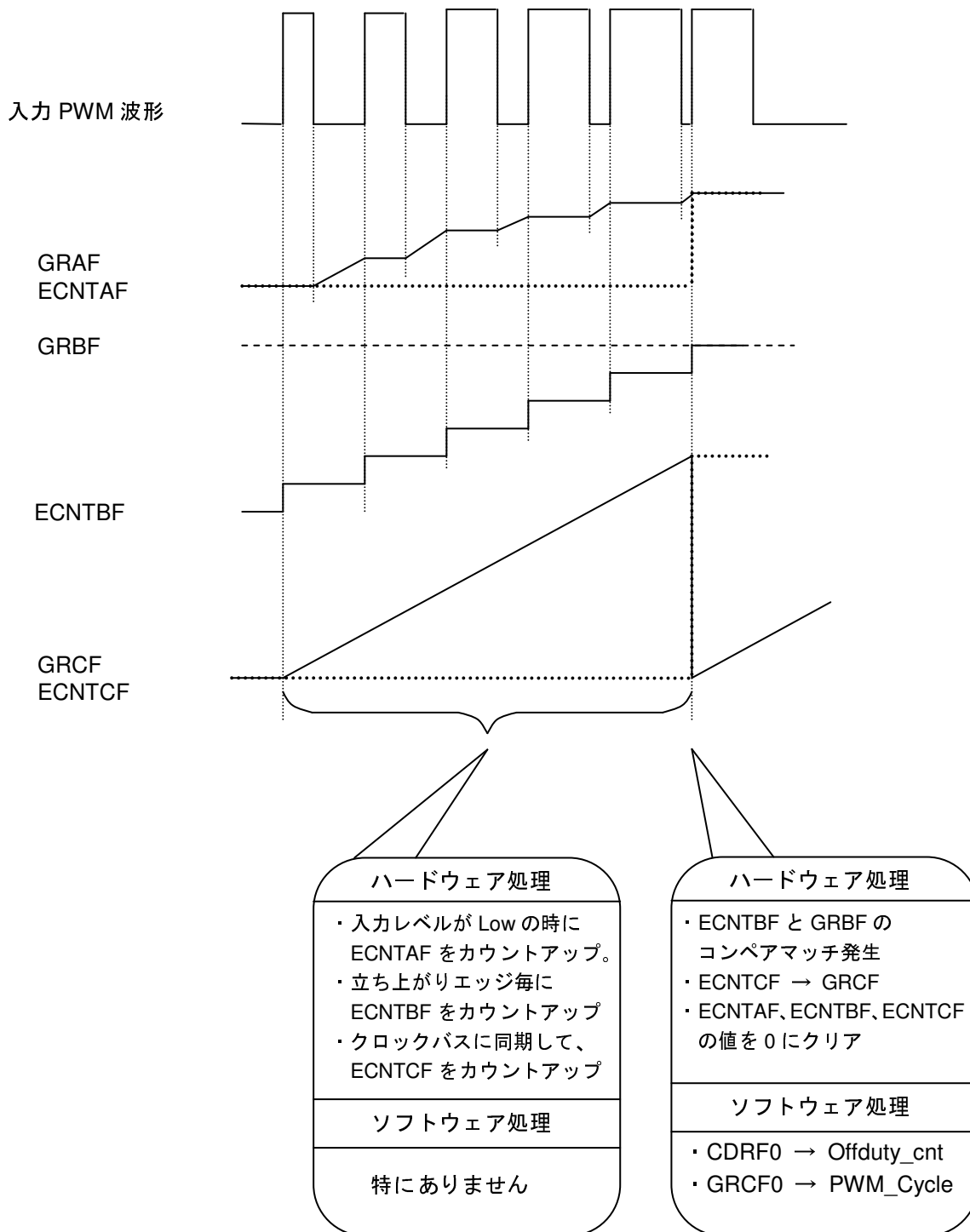


図 2.30 有効エッジ入力間隔計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.16.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
計測値の保存ルーチン	ICIF0	計測した値を変数に保存します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
Offduty_cnt	計測期間内のオフデューティの期間の合計値を保存します。	unsigned long	計測値の保存ルーチン
PWM_Cycle	直前に入力されたPWM波のサイクルを保存します。		

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0 端子を TIF0A に設定します。	0x0003	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウンタ開始を許可します。	0x0061	
TSTRF	タイマ F サブブロック 0 のカウンタをスタートさせます。	0x00000001	
TCRF0	サブブロック 0 を「クロックバス 0 でカウンタ」、「PWM 入力波形計測」「計測エッジは立ち上がり」に設定します。	0x11	
GRBF0	計測パルスの回数を 6 回に設定します。	0x0006	
TIERF0	TIF0 のインプットキャプチャ割り込み(ICIF0)を許可します。	0x01	
CDRF0	計測したオフデューティ合計を保存します。	—	計測値の保存ルーチン
GRCF0	直前の PWM サイクルの値を保存します。	—	
TSRF0	タイマ F サブブロック 0 のステータスレジスタをクリアします。	0x00	

2.16.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Measurement of PWM input waveform.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h> /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h" /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long Offduty_cnt, PWM_Cycle;
void main(void); /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```

void main(void)
{
    /* ポートの設定 */
    /* Configure PHCR
    b15:12 Reserved
    b11:10 PH5MD[1:0] = 00    PH5 入出力 (ポート)
    b9:8 PH4MD[1:0] = 00    PH4 入出力 (ポート)
    b7:6 PH3MD[1:0] = 00    PH3 入出力 (ポート)
    b5:4 PH2MD[1:0] = 00    PH2 入出力 (ポート)
    b3:2 PH1MD[1:0] = 00    PH1 入出力 (ポート)
    b1:0 PH0MD[1:0] = 11    TIF0A 入力 (ATU-III) */
    PORTH.CR.WORD = 0x0003;          /* PH0 を TIF0A(エッジ入力用端子)に設定
    */

    /* ATU-III (PWM 入力波形計測) の設定 */
    /* Configure TCRF0
    b7:5 CKSELF0[2:0] = 000    クロックバス 0
    b4:2 MDF0[2:0] = 100      PWM 入力波形計測
    b1:0 EGSELEFn[1:0] = 01    立ち上がりエッジ */
    ATUF.SUBBLOCK[0].TCRF.BYTE= 0x11;    /* サブブロック 0 の時間計測カウンタのクロック
    ソースを「クロックバス 0」,
    動作モードを「PWM 入力波形計測」,
    計測エッジを「立上りエッジ」に それぞれ設定    */

    /* Configure GRBF0
    b15:0 GRBF0[15:0]    イベントカウンタのインプットキャプチャ値 */
    ATUF.SUBBLOCK[0].GRBF = 0x0006;    /* 検出するエッジの回数を 6 周期分に設定
    */

    /* Configure TIERF0
    b7:4 Reserved
    b3 OVECF0 = 0    OVFCF0 による割り込みを禁止
    b2 OVEBF0 = 0    OVFBF0 による割り込みを禁止
    b1 OVEAF0 = 0    OVFAF0 による割り込みを禁止
    b0 ICEF0 = 1    ICFF0 による割り込みを許可 */
    ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01;    /* TIF0 のインプットキャプチャ割り込み
    (ICIF0)を許可 */

    /* Configure TSTRF
    b0 STRF0 = 1    ECNTAF0、ECNTBF0、および ECNTCF0 のカウント動作を許可 */
    ATUF.TSTRF.BIT.STRF0 = 1;    /* サブブロック 0 のカウントをスタート    */

    /* Configure PSCR0
    b15:10 Reserved
    b9:0 PSC0 = H'9    プリスケーラの分周比を 10 (設定値は-1)に設定 */
    ATUCTRL.PSCR0.WORD = 0x0009;    /* プリスケーラの分周比を 10 に設定    */

    /* Configure IPR16
    b15:12 ATU-F (ICIF0~ICIF3) = H'F    対応する割り込みの優先順位を 15 に設定
    b11:8 ATU-F (ICIF4~ICIF7) = H'0    対応する割り込みの優先順位を 0 に設定
    b7:4 ATU-F (ICIF8~ICIF11) = H'0    対応する割り込みの優先順位を 0 に設定
    b3:0 ATU-F (ICIF12~ICIF15) = H'0    対応する割り込みの優先順位を 0 に設定 */
    INTC.IPR16.WORD = 0xF000;    /* ICIF0 の割り込み優先順位を 15 に設定    */

    set_imask(0);    /* 割り込みマスクレベルを 0 に設定    */
}

```

```
/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0     タイマ J のカウント動作を停止
b8     THE = 0     タイマ H のカウント動作を停止
b7     TGE = 0     タイマ G のカウント動作を停止
b6     TFE = 1     タイマ F のカウント動作を許可
b5     TEE = 1     タイマ E のカウント動作を許可
b4     TDE = 0     タイマ D のカウント動作を停止
b3     TCE = 0     タイマ C のカウント動作を停止
b2     TBE = 0     タイマ B のカウント動作を停止
b1     TAE = 0     タイマ A のカウント動作を停止
b0     PSCE = 1   プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0061;      /* タイマ E, F およびプリスケーラのカウントをス
タート */

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value  : none
*****/
void ICIF0(void)
{
    /* Configure CDRF0
b31:8  CDRF0[23:0]   GRAF0 のデータ
b7:0 Reserved */
    Offduty_cnt = ATUF.SUBBLOCK[0].CDRF; /* 変数 Edge_cnt に計測したオフデューティ合計を
保存 */

    /* Configure GRCF0
b31:8  GRCF0[23:0]   ECNTCF0 のデータ
b7:0 Reserved */
    PWM_Cycle = ATUF.SUBBLOCK[0].GRCF; /* 変数 PWM_Cycle に直前の PWM サイクルの値を保
存 */

    /* Configure TSRF0
b7:4 Reserved
b3     OVFCF0 = 0     オーバフローフラグをクリア
b2     OVFBF0 = 0     オーバフロー/アンダフローフラグをクリア
b1     OVFAF0 = 0     オーバフローフラグをクリア
b0     ICFF0 = 0     インพุットキャプチャの検出フラグをクリア */
    ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア
*/
} /* End of function ICIF0() */
```

2.17 動作例 16 回転速度/パルス計測[タイマ F]

2.17.1 概要

図 2.31に示すように、入力されたエッジ数とエッジ入力時間を計測し、さらに直前の入力エッジとの間に現れる PWM 波形のオフデューティと PWM サイクルを測定します。動作モードは回転速度/パルス計測モードに設定します。

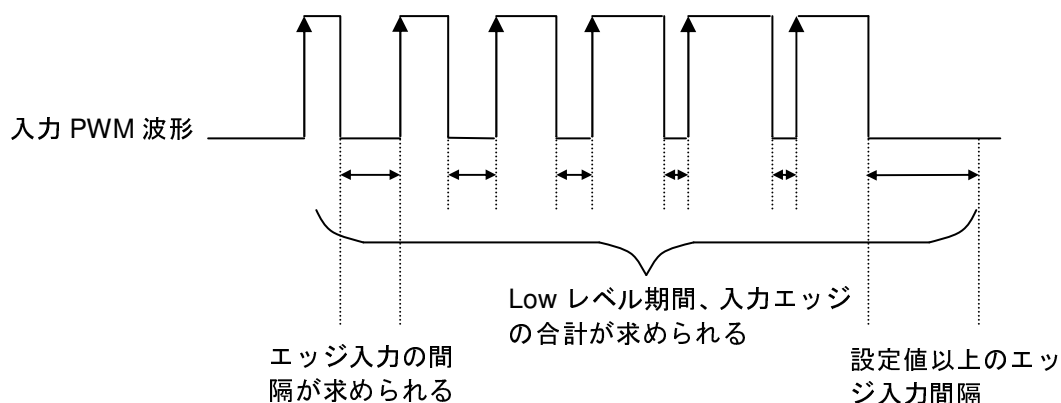


図 2.31 回転速度/パルス計測の概要

2.17.2 使用機能説明

表 2.16に使用端子及び関連レジスタの機能割り付けを示します。

表 2.16 ATU-III機能割り付け

使用端子		機能
端子	TIF12	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケーラの動作を設定します。
	PSCR0	プリスケーラ0の分周比を設定します。
タイマF レジスタ	TCRF12	タイマF各サブブロックの動作を設定します。
	ECNTAF12	設定したレベルが入力されている期間をカウントします。
	ECNTBF12	有効な外部信号の回数を記録します。
	ECNTCF12	エッジの間隔をカウントします。
	GRAF12	ECNTAF12の値をキャプチャします。
	GRBF12	エッジの最大間隔を設定します。
	GRCF12	ECNTCF12の値をキャプチャします。
	GRDF12	動作開始からのカウント数をキャプチャします。
	CDRF12	ECNTBF12の値をキャプチャします。
	TSRF12	タイマFサブブロック12の状態が格納されます。
TSTRF	タイマF各サブブロックのカウンタの動作を設定します。	
PFCレジスタ	PHCR	端子機能を設定します。
INTCレジスタ	IPR16	TIF12のインプットキャプチャ割り込みの優先度を設定します。

2.17.3 動作説明

図 2.32に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって PWM 波形のオフデューティ、PWM サイクルを計測します。下記では、カウンタ値 ECNTCF がオーバーフローフラグ設定値である GRBF にセットされるとインプットキャプチャ割り込みが発生し、計測に必要なカウンタ値を得ます。

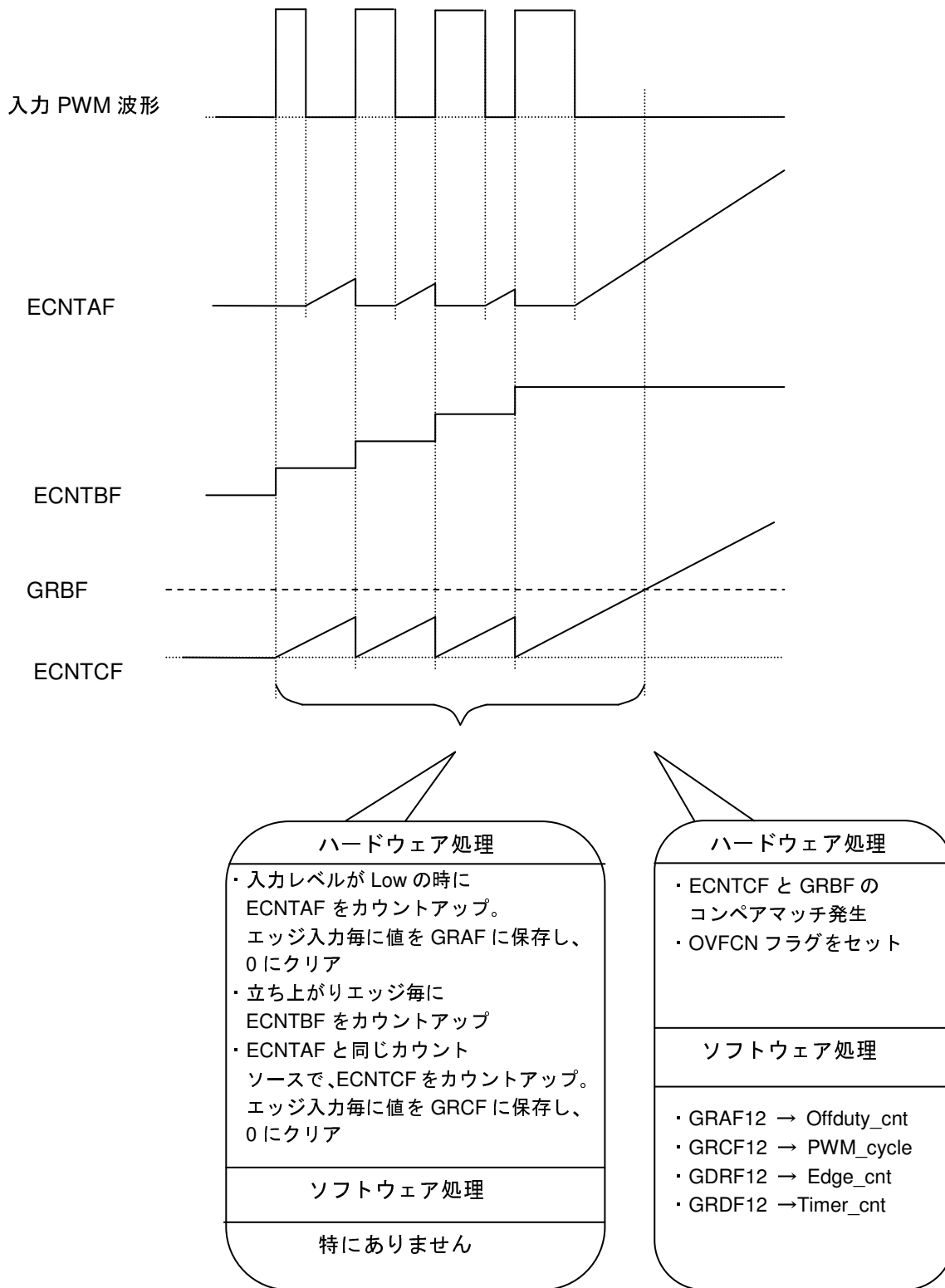


図 2.32 有効エッジ入力間隔計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.17.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
計測値の保存ルーチン	ICIF12	計測した値を変数に保存します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
Edge_cnt	動作開始からその時点までの入力されたエッジ数を保存します。	unsigned long	計測値の保存ルーチン
Timer_cnt	動作開始からその時点までの設定したクロックバスでのカウント数を保存します。		
Offduty_cnt	直前に入力された PWM 波形のオフデューティの値を保存します。		
PWM_Cycle	直前に入力された PWM 波のサイクルを保存します。		

- 使用レジスタの設定

レジスタ名	機能	設定値	使用モジュール名
PGCR2	PG14 端子を TIF12 に設定します。	0x3000	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウント開始を許可します。	0x0061	
TSTRF	タイマ F サブブロック 12 のカウントをスタートさせます。	STRF12 = 1	
TCRF12	サブブロック 12 を「クロックバス 0 でカウント」、「回転速度/パルス計測」「計測エッジは立ち上がり」に設定します。	0x15	
GRBF12	オーバーフローを発生させるエッジ間隔を 0x110 に設定します。	0x0110	
TIERF12	ICFF12 のセットによるインプットキャプチャ割り込みを許可します。	0x0001	
IPR16	ICIF12 の割り込み優先度を 14 に設定します。	0x000E	

2.17.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Measurement of rotation speed and pulse .
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long Offduty_cnt, PWM_cycle, Edge_cnt, Timer_cnt;
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  Offduty_cnt = 0;
  PWM_cycle   = 0;
  Edge_cnt    = 0;
  Timer_cnt   = 0;

  /* ポートの設定 */
  /* Configure PGCR2
b15:14    PG15MD[1:0] = 00  PG15 入出力 (ポート)
b13:12    PG14MD[1:0] = 11  TIF12 入力 (ATU-III)
b11:10    PG13MD[1:0] = 00  PG13 入出力 (ポート)
b9:8      PG12MD[1:0] = 00  PG12 入出力 (ポート)
b7:6      PG11MD[1:0] = 00  PG11 入出力 (ポート)
b5:4      PG10MD[1:0] = 00  PG10 入出力 (ポート)
b3:2      PG9MD[1:0]  = 00  PG9  入出力 (ポート)
b1:0      PG8MD[1:0]  = 00  PG8  入出力 (ポート) */
  PORTG.CR2.WORD = 0x3000; /* PG14 を TIF12 (エッジ入力用端子) に設定
  */

  /* ATU-III (回転速度/パルス計測) の設定 */
  /* Configure TCRF12
b7:5 CKSELF12[2:0] = 000   クロックバス 0
b4:2 MDF12[2:0]   = 101   回転速度/パルス計測
b1:0 EGSELF12[1:0] = 01   立ち上がりエッジ */
  ATUF.SUBBLOCK[12].TCRF.BYTE= 0x15; /*サブブロック 12 の時間計測カウンタのクロックソースを「ク
  ロックバス 0」,
                                     動作モードを「回転速度/パルス計測」,
                                     計測エッジを「立上りエッジ」に それぞれ設定      */

  /* Configure GRBF0
b15:0  GRBF0[15:0]   イベントカウンタのインプットキャプチャ値 */
  ATUF.SUBBLOCK[12].GRBF = 0x0110; /* オーバフローフラグをセットするエッジ間隔を 0x0110
  に設定 */

  /* Configure TSRF12
b7:4 Reserved
b3      OVFCF12 = 0   ECNTCF12 にオーバーフローなし
b2      OVFBF12 = 0   ECNTBF12 にオーバーフロー/アンダフローなし
b1      OVFAF12 = 0   ECNTAF12 にオーバーフローなし
b0      ICFF12  = 0   サブブロック F12 にインプットキャプチャの検出なし */
  ATUF.SUBBLOCK[12].TSRF.BYTE &= 0x00; /* インプットキャプチャフラグをクリア
  */

  /* Configure TIERF12
b7:4 Reserved
b3      OVECF12 = 0   OVFCF12 による割り込みを禁止
b2      OVEBF12 = 0   OVFBF12 による割り込みを禁止
b1      OVEAF12 = 0   OVFAF12 による割り込みを禁止
b0      ICEF12  = 1   ICFF12 による割り込みを許可 */
  ATUF.SUBBLOCK[12].TIERF.BYTE = 0x01; /* ICEF12 セットによる割り込みを許可
  */

  /* Configure TSTRF
b12     STRF12 = 1    ECNTAF12、ECNTBF12、および ECNTCF12 のカウント動作を許可 */
}
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUF.TSTRF.BIT.STRF12 = 1;          /* サブブロック 12 のカウントをスタート
*/

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCR0 = H'9      プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009;      /* プリスケーラの分周比を 10 に設定 */

/* Configure IPR16
b15:12 ATU-F (ICIF0~ICIF3) = H'0      対応する割り込みの優先順位を 0 に設定
b11:8  ATU-F (ICIF4~ICIF7) = H'0      対応する割り込みの優先順位を 0 に設定
b7:4  ATU-F (ICIF8~ICIF11) = H'0      対応する割り込みの優先順位を 0 に設定
b3:0  ATU-F (ICIF12~ICIF15) = H'E     対応する割り込みの優先順位を 14 に設定 */
INTC.IPR16.WORD = 0x000E;          /* ICIF12 の割り込み優先順位を 14 に設定
*/

set_imask(0);                      /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0      タイマ J のカウント動作を停止
b8     THE = 0      タイマ H のカウント動作を停止
b7     TGE = 0      タイマ G のカウント動作を停止
b6     TFE = 1      タイマ F のカウント動作を許可
b5     TEE = 1      タイマ E のカウント動作を許可
b4     TDE = 0      タイマ D のカウント動作を停止
b3     TCE = 0      タイマ C のカウント動作を停止
b2     TBE = 0      タイマ B のカウント動作を停止
b1     TAE = 0      タイマ A のカウント動作を停止
b0     PSCE = 1     プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0061;      /* タイマ E, F およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF12
* Description   : インพุットキャプチャ割り込み
* Arguments    : none
* Return Value  : none
*****/
void ICIF12(void)
{
    /* Configure GRAF12
b31:8  GRAF12[23:0] ECNTAF12 のデータ
b7:0 Reserved */
    Offduty_cnt = (ATUF.SUBBLOCK[12].GRAF >> 8); /* 変数 Offduty_cnt に直前の PWM 波形
のオフデューティの値を保存 */

    /* Configure GRCF12
b31:8  GRCF12[23:0] ECNTCF12 のデータ
b7:0 Reserved */
    PWM_cycle = (ATUF.SUBBLOCK[12].GRCF >> 8); /* 変数 PWM_cycle に直前の PWM 波形
のサイクルの値を保存 */
}
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
/* Configure CDRF12
b31:8 CDRF0[23:0] ECNTBF12 のデータ
b7:0 Reserved */
Edge_cnt = (ATUF.SUBBLOCK[12].CDRF >> 16); /* 変数 Edge_cnt に動作開始からのエッジ
数を保存 */

/* Configure GRDF12
b31:8 GRDF12[23:0] ECNTAF12 の累積データ
b7:0 Reserved */
Timer_cnt = (ATUF.SUBBLOCK[12].GRDF >> 8); /* 変数 Timer_cnt に動作開始からのカ
ウント数を保存 */

/* Configure TSRF0
b7:4 Reserved
b3 OVFCF12 = 0 オーバフローフラグをクリア
b2 OVFBF12 = 0 オーバフロー/アンダフローフラグをクリア
b1 OVFAF12 = 0 オーバフローフラグをクリア
b0 ICFF12 = 0 インพุットキャプチャの検出フラグをクリア */
ATUF.SUBBLOCK[12].TSRF.BYTE &= 0x00; /* ICEF0 を 0 にクリア
*/
} /* End of function ICIF12() */
```

2.18 動作例 17 アップダウンイベントカウント[タイマ F]

2.18.1 概要

- 1) 二つの入力端子を用い、一方への入力エッジでカウントを、もう一方への入力レベルでそのカウントがアップカウントか、ダウンカウントかを切り替え、カウントを行います。
- 2) 一定期間毎にその時点でのカウントの値を保存し、カウントの値をクリアします。

2.18.2 使用機能説明

表 2.17に使用端子及び関連レジスタの機能割り付けを示します。

表 2.17 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号（カウント用エッジ）を入力します。
	TIF0B	外部から信号（カウント方向用レベル）を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ F レジスタ	TCRF0	タイマ F 各サブブロックの動作を設定します。
	ECNTAF0	指定したクロックバスで時間を計測します。
	ECNTBF0	二端子への入力に従ってアップ/ダウンカウント動作を行います。
	GRAF0	ECNTAF0 に対する、コンペアマッチの値を設定します。
	GRBF0	ECNTAF0 のコンペアマッチ発生時に、ECNTBF0 の値をキャプチャします。
	TSRF0	タイマ F サブブロック 0 の状態が格納されます。
PFC レジスタ	TSTRF	タイマ F 各サブブロックのカウントの動作を設定します。
	PHCR	ポート H の端子機能を設定します。
PFC レジスタ	PDCR2	ポート D の端子機能を設定します。
	INTC レジスタ	IPR16

2.18.3 動作説明

表 2.18、図 2.33に二端子の入力に対応した動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってアップ/ダウンカウントを行います。

表 2.18 アップ/ダウンイベントカウントモード時のカウント方向

入力	カウント方向	
	アップカウント	ダウンカウント
TIFnA (入力エッジ)		
TIFnB (入力レベル)	ローレベル	ハイレベル

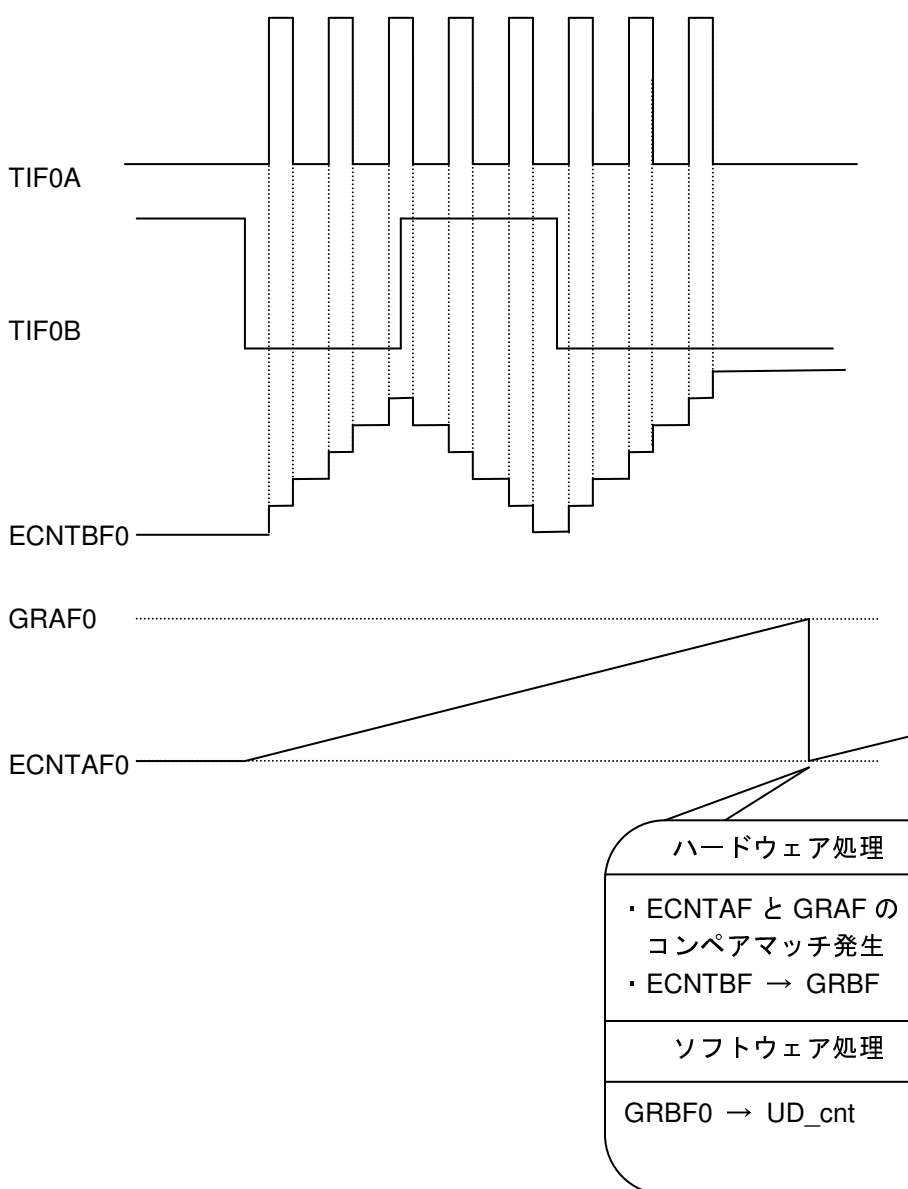


図 2.33 アップダウンイベントカウント動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.18.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
コンペアマッチ 割り込みルーチン	ICIF0	ECNTAF0、GRAF0のコンペアマッチ時に割り込み、ECNTBF0の値をGRBF0にキャプチャします。

- 使用変数の説明

変数名	機能
UD_cnt	コンペアマッチ割り込み発生時の ECNTBF0 の値を保存します。

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0 端子を TIF0A に設定します。	0x0003	メインルーチン
PDCR2	PD9 端子を TIF0B に設定します。	0x0008	
PSCR0	プリスケアラ 0 の分周比を 50 に設定します。	0x0031	
ATUENR	ATU-IIIのタイマ F およびプリスケアラのカウント動作を許可します。	0x0043	
TSTRF	タイマ F サブブロック 0 のカウントをスタートさせます。	0x00000001	
TSRF0	サブブロック 0 のフラグ管理を行います。	0x00	
TCRF0	サブブロック 0 を「クロックバス 0 でカウント」「アップ/ダウンイベントカウント」「計測エッジは両エッジ」に設定します。	0x1F	
TIERF0	ECNTAF0、GRAF0 のコンペアマッチ発生時の割り込みを許可します。	0x01	
GRAF0	検出するエッジの回数を FF(256)回に設定します。	0x0000FF00	
IPR16	ICIF0 の割り込み優先順位を 0xE に設定します。	0xE000	
TSRF0	タイマ F サブブロック 0 のステータスレジスタをクリアします。	0x00	コンペアマッチ割り込みルーチン

2.18.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)     : SH72546R
 * Tool-Chain    : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                : C/C++ compiler package for the SuperH RISC engine family
 *                : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform  : SH7254R
 * Description    : This is the main tutorial code.
 * Operation     : Up/Down event count.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned short UD_cnt;
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PHCR
  b15:12 Reserved
  b11:10 PH5MD[1:0] = 00   PH5 入出力 (ポート)
  b9:8 PH4MD[1:0] = 00   PH4 入出力 (ポート)
  b7:6 PH3MD[1:0] = 00   PH3 入出力 (ポート)
  b5:4 PH2MD[1:0] = 00   PH2 入出力 (ポート)
  b3:2 PH1MD[1:0] = 00   PH1 入出力 (ポート)
  b1:0 PH0MD[1:0] = 11   TIF0A 入力 (ATU-III) */
  PORTH.CR.WORD = 0x0003;      /* PH0 を TIF0A(エッジ入力用端子)に設定
  */

  /* Configure PDCR2
  b15:12 Reserved
  b11:10 PD13MD[1:0] = 00   PD13 入出力 (ポート)
  b9:8 PD12MD[1:0] = 00   PD12 入出力 (ポート)
  b7:6 PD11MD[1:0] = 00   PD11 入出力 (ポート)
  b5:4 PD10MD[1:0] = 00   PD10 入出力 (ポート)
  b3:2 PD9MD[1:0] = 10   TIF0B 入力 (ATU-III)
  b1:0 PD8MD[1:0] = 00   PD8 入出力 (ポート) */
  PORTD.CR2.WORD = 0x0008;    /* PD9 を TIF0B(エッジ入力用端子)に設定
  */

  /* ATU-III (4 通倍イベントカウント) の設定 */
  /* Configure TCRF0
  b7:5 CKSELF0[2:0] = 000   クロックバス 0
  b4:2 MDF0[2:0] = 111     4 通倍イベントカウント
  b1:0 EGSELEFn[1:0] = 11   両エッジ */
  ATUF.SUBBLOCK[0].TCRF.BYTE= 0x1F; /* サブブロック 0 の時間計測カウンタのクロックソース
  を「クロックバス 0」,
  動作モードを「4 通倍イベントカウント」,
  計測エッジを「両エッジ」に それぞれ設定      */

  /* Configure GRAF0
  b31:8 GRAF0[23:0]   アウトプットコンペア値
  b7:0 Reserved */
  ATUF.SUBBLOCK[0].GRAF = 0x0000FF00; /* 計測する期間をクロックバス 0 0xFF 回(254 周
  期分)に設定 */

  /* Configure TIERF0
  b7:4 Reserved
  b3   OVECF0 = 0   OVFCF0 による割り込みを禁止
  b2   OVEBF0 = 0   OVFBF0 による割り込みを禁止
  b1   OVEAF0 = 0   OVFAF0 による割り込みを禁止
  b0   ICEF0 = 1   ICCF0 による割り込みを許可 */
  ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01; /* ECNTAF0 のコンペアマッチ割り込み(ICIF0)を
  許可      */

  /* Configure TSTRF
  b0   STRF0 = 1   ECNTAF0、ECNTBF0、および ECNTCF0 のカウント動作を許可 */
  ATUF.TSTRF.BIT.STRF0 = 1;      /* サブブロック 0 のカウントをスタート      */

  /* Configure PSCR0
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:10 Reserved
b9:0 PSC0 = H'31 プリスケーラの分周比を 50 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x00031; /* プリスケーラの分周比を 50 に設定
*/

/* Configure IPR16
b15:12 ATU-F (ICIF0~ICIF3) = H'E 対応する割り込みの優先順位を 14 に設定
b11:8 ATU-F (ICIF4~ICIF7) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4 ATU-F (ICIF8~ICIF11) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 ATU-F (ICIF12~ICIF15) = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR16.WORD = 0xE000; /* ICIF0 の割り込み優先順位を 14 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9 TJE = 0 タイマ J のカウント動作を停止
b8 THE = 0 タイマ H のカウント動作を停止
b7 TGE = 0 タイマ G のカウント動作を停止
b6 TFE = 1 タイマ F のカウント動作を許可
b5 TEE = 0 タイマ E のカウント動作を停止
b4 TDE = 0 タイマ D のカウント動作を停止
b3 TCE = 0 タイマ C のカウント動作を停止
b2 TBE = 0 タイマ B のカウント動作を停止
b1 TAE = 1 タイマ A のカウント動作を許可
b0 PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0043; /* タイマ A, F およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description : コンペアマッチ割り込み
* Arguments : none
* Return Value : none
*****/
void ICIF0(void)
{
/* Configure TSRF0
b7:4 Reserved
b3 OVFCF0 = 0 オーバフローフラグをクリア
b2 OVFBF0 = 0 オーバフロー/アンダフローフラグをクリア
b1 OVFAF0 = 0 オーバフローフラグをクリア
b0 ICFF0 = 0 インพุットキャプチャの検出フラグをクリア */
ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */

/* Configure GRBF0
b15:0 CDRFn[23:0] ECNTBF0 のデータ */
UD_cnt = ATUF.SUBBLOCK[0].GRBF; /* キャプチャした ECNTBF の値を変数に保存 */
} /* End of function ICIF0() */
```

2.19 動作例 18 4 通倍イベントカウント[タイマ F]

2.19.1 概要

二つの入力端子を用い、両方の入力エッジ及び入力レベルによってアップカウントか、ダウンカウントかを切り替え、カウントを行います。

2.19.2 使用機能説明

表 2.19に使用端子及び関連レジスタの機能割り付けを示します。

表 2.19 ATU-III機能割り付け

使用端子		機能
端子	TIF0A	外部から信号を入力します。
	TIF0B	外部から信号を入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ F レジスタ	TCRF0	タイマ F 各サブブロックの動作を設定します。
	TIERF0	タイマ F 割り込み許可の設定を行います。
	ECNTAF0	指定したクロックバスで時間を計測します。
	ECNTBF0	二端子への入力に従って 4 通倍イベントカウント動作を行います。
	GRAF0	ECNTAF0 に対する、コンペアマッチの値を設定します。
	GRBF0	ECNTAF0 のコンペアマッチ発生時に、ECNTBF0 の値をキャプチャします。
PFC レジスタ	TSTRF	タイマ F 各サブブロックのカウンタの動作を設定します。
	PHCR	ポート H の端子機能を設定します。
INTC レジスタ	PDCR2	ポート D の端子機能を設定します。
	IPR16	コンペアマッチ割り込み ICIF0 の優先度を設定します

2.19.3 動作説明

表 2.20に二端子の入力に対応した動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって4 通倍イベントカウントを行います。

表 2.20 アップ/ダウンイベントカウントモード時のカウント方向

入力	カウント方向							
	アップカウント				ダウンカウント			
TIFnA (入力エッジ)	ハイレベル		ローレベル		ハイレベル		ローレベル	
TIFnB (入力レベル)		ハイレベル		ローレベル		ローレベル		ハイレベル

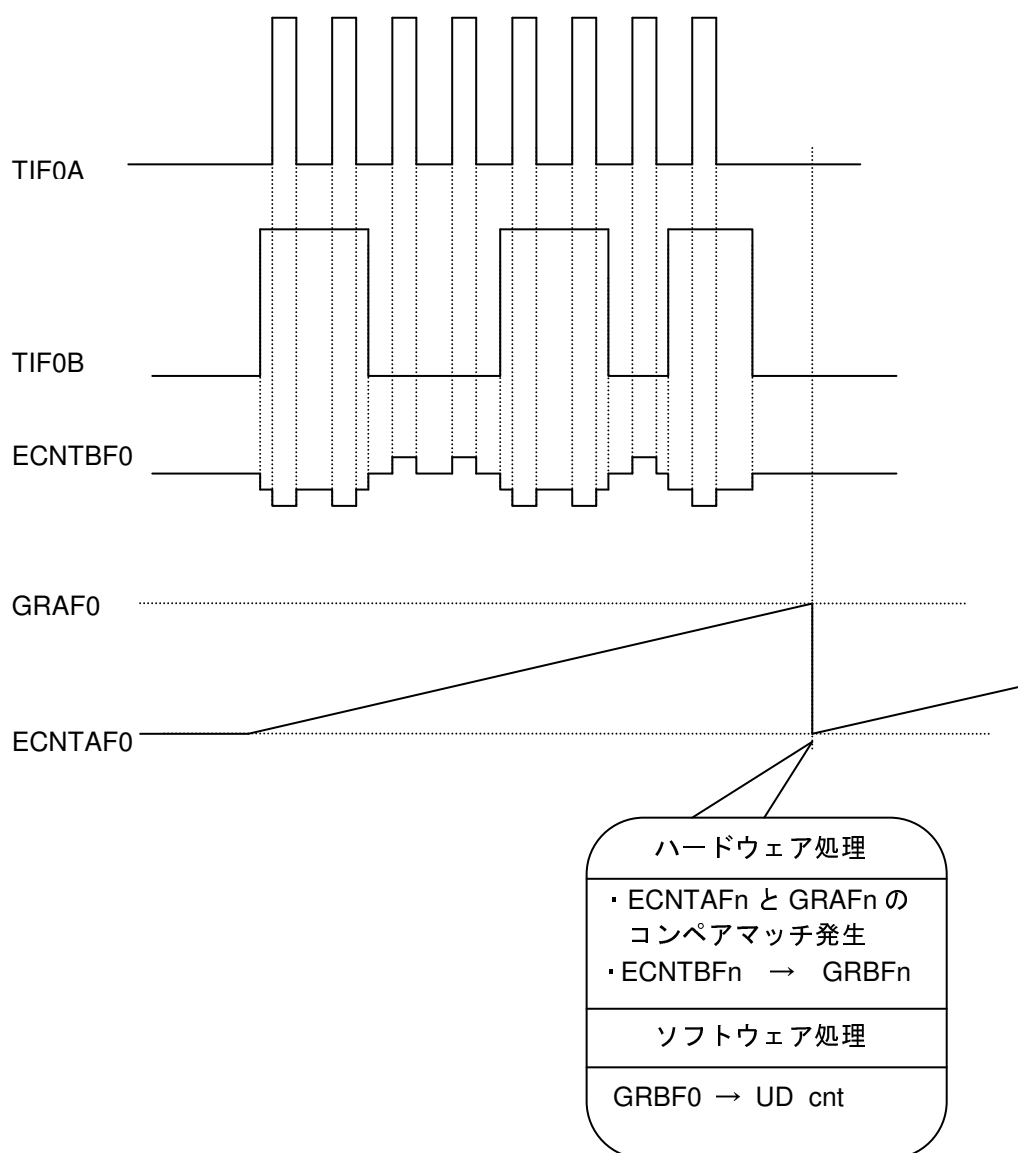


図 2.34 アップダウンイベントカウント動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.19.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
コンペアマッチ 割り込みルーチン	ICIF0	ECNTAF0、GRAF0のコンペアマッチ時に割り込みを発生し、ECNTBF0の値をGRBF0にキャプチャします。

- 使用変数の説明

変数名	機能
UD_cnt	コンペアマッチ割り込み発生時のECNTBF0の値を保存します。

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PHCR	PH0端子をTIF0Aに設定します。	0x0003	メインルーチン
PDCR2	PD9端子をTIF0Bに設定します。	0x0008	
PSCR0	プリスケアラ0の分周比を50に設定します。	0x031	
ATUENR	ATU-IIIのタイマFおよびプリスケアラのカウンタ動作を許可します。	0x0043	
TSTRF	タイマFサブブロック0のカウンタをスタートさせます。	0x00000001	
TSRF0	サブブロック0のフラグ管理を行います。	0x00	
TCRF0	サブブロック0を「クロックバス0でカウンタ」「アップ/ダウンイベントカウンタ」「計測エッジは両エッジ」に設定します。	0x1B	
TIERF0	ECNTAF0、GRAF0のコンペアマッチ発生時の割り込みを許可します。	0x01	
GRAF0	検出するエッジの回数をFF(256)回に設定します。	0x0000FF00	
ECNTAF0	TCRF0で設定したクロックバスでアップカウンタを行います。	—	
ECNTBF0	二端子への入力に対応したアップ/ダウンカウンタを行います。	—	
GRBF0	コンペアマッチ発生時にECNTBF0の値を保存します。	—	

2.19.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Event count multiplied by 4.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long UD_cnt;
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments     : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PHCR
  b15:12 Reserved
  b11:10 PH5MD[1:0] = 00   PH5 入出力 (ポート)
  b9:8 PH4MD[1:0] = 00   PH4 入出力 (ポート)
  b7:6 PH3MD[1:0] = 00   PH3 入出力 (ポート)
  b5:4 PH2MD[1:0] = 00   PH2 入出力 (ポート)
  b3:2 PH1MD[1:0] = 00   PH1 入出力 (ポート)
  b1:0 PH0MD[1:0] = 11   TIF0A 入力 (ATU-III) */
  PORTH.CR.WORD = 0x0003;      /* PH0 を TIF0A(エッジ入力用端子)に設定
  */

  /* Configure PDCR2
  b15:12 Reserved
  b11:10 PD13MD[1:0] = 00   PD13 入出力 (ポート)
  b9:8 PD12MD[1:0] = 00   PD12 入出力 (ポート)
  b7:6 PD11MD[1:0] = 00   PD11 入出力 (ポート)
  b5:4 PD10MD[1:0] = 00   PD10 入出力 (ポート)
  b3:2 PD9MD[1:0] = 10   TIF0B 入力 (ATU-III)
  b1:0 PD8MD[1:0] = 00   PD8 入出力 (ポート) */
  PORTD.CR2.WORD = 0x0008;    /* PD9 を TIF0B(エッジ入力用端子)に設定
  */

  /* ATU-III (4 通倍イベントカウンタ)の設定 */
  /* Configure TCRF0
  b7:5 CKSELF0[2:0] = 000 クロックバス 0
  b4:2 MDF0[2:0] = 111   4 通倍イベントカウンタ
  b1:0 EGSELEn[1:0] = 11 両エッジ */
  ATUF.SUBBLOCK[0].TCRF.BYTE= 0x1F; /* サブブロック 0 の時間計測カウンタのクロックソースを「ク
  ロックバス 0」,
  動作モードを「4 通倍イベントカウンタ」,
  計測エッジを「両エッジ」に それぞれ設定      */

  /* Configure GRAF0
  b31:8 GRAF0[23:0]   アウトプットコンペア値
  b7:0 Reserved */
  ATUF.SUBBLOCK[0].GRAF = 0x0000FF00; /* 計測する期間をクロックバス 0 0xFF 回(254 周
  期分)に設定 */

  /* Configure TIERF0
  b7:4 Reserved
  b3   OVECF0 = 0   OVFCF0 による割り込みを禁止
  b2   OVEBF0 = 0   OVFBF0 による割り込みを禁止
  b1   OVEAF0 = 0   OVFAF0 による割り込みを禁止
  b0   ICEF0 = 1   ICCF0 による割り込みを許可 */
  ATUF.SUBBLOCK[0].TIERF.BYTE = 0x01; /* ECNTAF0 のコンペアマッチ割り込み(ICIF0)を
  許可      */

  /* Configure TSTRF
  b0   STRF0 = 1   ECNTAF0、ECNTBF0、および ECNTCF0 のカウンタ動作を許可 */
  ATUF.TSTRF.BIT.STRF0 = 1;      /* サブブロック 0 のカウンタをスタート      */

  /* Configure PSCR0
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:10 Reserved
b9:0 PSC0 = H'31 プリスケーラの分周比を 50 (設定値は-1)に設定 */
ATUCTRL.PSCR0.WORD = 0x00031; /* プリスケーラの分周比を 50 に設定
*/

/* Configure IPR16
b15:12 ATU-F (ICIF0~ICIF3) = H'E 対応する割り込みの優先順位を 14 に設定
b11:8 ATU-F (ICIF4~ICIF7) = H'0 対応する割り込みの優先順位を 0 に設定
b7:4 ATU-F (ICIF8~ICIF11) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 ATU-F (ICIF12~ICIF15) = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR16.WORD = 0xE000; /* ICIF0 の割り込み優先順位を 14 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9 TJE = 0 タイマ J のカウント動作を停止
b8 THE = 0 タイマ H のカウント動作を停止
b7 TGE = 0 タイマ G のカウント動作を停止
b6 TFE = 1 タイマ F のカウント動作を許可
b5 TEE = 0 タイマ E のカウント動作を停止
b4 TDE = 0 タイマ D のカウント動作を停止
b3 TCE = 0 タイマ C のカウント動作を停止
b2 TBE = 0 タイマ B のカウント動作を停止
b1 TAE = 1 タイマ A のカウント動作を許可
b0 PSCE = 1 プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0043; /* タイマ A, F およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : ICIF0
* Description : インพุットキャプチャ割り込み
* Arguments : none
* Return Value : none
*****/
void ICIF0(void)
{
/* Configure TSRF0
b7:4 Reserved
b3 OVFCF0 = 0 オーバフローフラグをクリア
b2 OVFBF0 = 0 オーバフロー/アンダフローフラグをクリア
b1 OVFAF0 = 0 オーバフローフラグをクリア
b0 ICFF0 = 0 インพุットキャプチャの検出フラグをクリア */
ATUF.SUBBLOCK[0].TSRF.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */

/* Configure GRBF0
b15:0 CDRFn[23:0] ECNTBF0 のデータ */
UD_cnt = ATUF.SUBBLOCK[0].GRBF; /* キャプチャした ECNTBF の値を変数に保存 */
} /* End of function ICIF0() */
```

2.20 動作例 19 コンペアマッチ割り込み発生[タイマ G]

2.20.1 概要

タイマカウントがコンペアマッチの値と一致した時、コンペアマッチ割り込みを発生させます。

2.20.2 使用機能説明

表 2.21に関連レジスタの機能割り付けを示します。

表 2.21 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケーラの動作を設定します。
	PSCR0	プリスケーラ 0 の分周比を設定します。
タイマ G レジスタ	TCRG0	タイマ G の各機能を設定します。
	OCRG0	コンペアマッチ値を設定します。
	TCNTG0	TCRG で設定したクロックバスでタイマカウントを行います。
	TSTRG	タイマ G サブクロック 0 のタイマカウンタをスタートさせます。
	TSRG0	オーバフロー、コンペアマッチが発生した場合にフラグがセットされます。
INTC レジスタ	IPR20	CMIG0 のコンペアマッチ割り込みの優先度を設定します。

2.20.3 動作説明

図 2.35に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってコンペアマッチを発生させます。タイマ G のカウント(TCNTG0)数は、コンペアマッチが発生するとクリアされます。

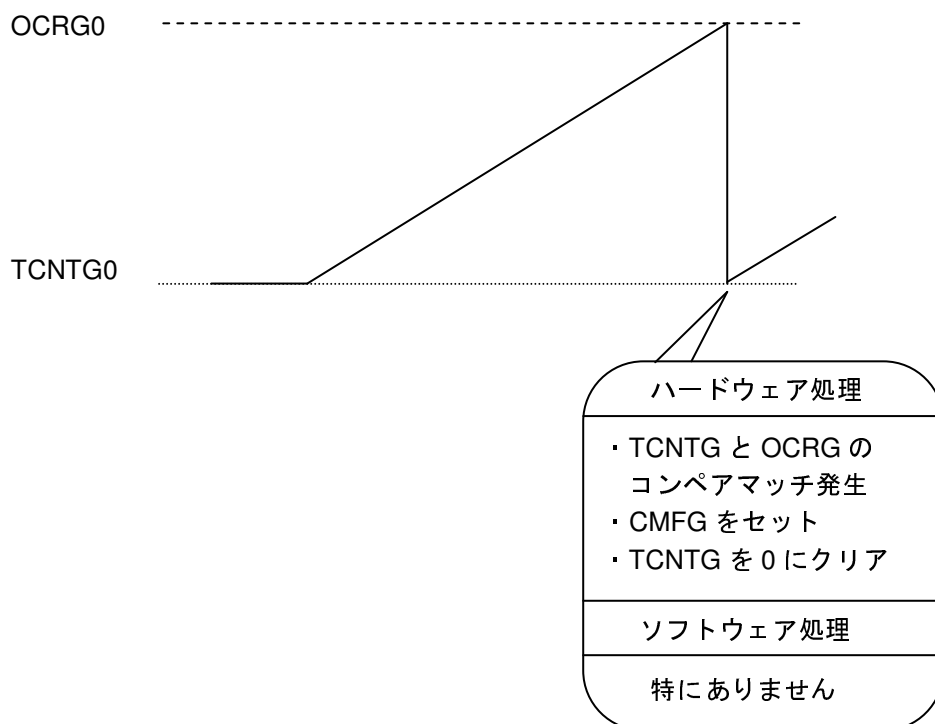


図 2.35 コンペアマッチ動作原理

2.20.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PSCR0	プリスケラ0の分周比を10に設定します。	0x0009	メインルーチン
ATUENR	ATU-IIIのチャンネルG及びプリスケラのカウンタ動作を許可します。	0x0081	
TSTRG	タイマGサブブロック0のカウンタ動作を許可します。	0x3F	
TCRG0	「クロックバス0でカウンタ」「コンペアマッチパルスを出力しない」「コンペアマッチ時の割り込みを許可」に設定します。	0x01	
OCRG0	TCNTG0とのコンペアマッチの値を設定します。	0x0400	
TSRG0	オーバフロー、コンペアマッチフラグをクリアします。	0x00	
IPR20	CMIF0の割り込み優先順位を15に設定します。	0xF000	

2.20.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description    : This is the main tutorial code.
 * Operation      : Compare match interrupt occurs.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value : none
 */

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
void main(void); /* メインルーチン */

/*****
* Function Name : main
* Description : The main loop
* Arguments : none
* Return Value : none
*****/
void main(void)
{
    /* ATU-IIIの設定 */
    /* Configure TCRG
    b7 Reserved
    b6:4 CKSELG0 = 0 クロックバス 0 をクロックソースに指定
    b3:2 Reserved
    b1 CMPOEG0 = 0 TCNTG0 と OCRG0 のコンペアマッチでパルス出力しない
    b0 CMEG0 = 1 CMFG0 による割り込みを許可 */
    ATUG.SUBBLOCK[0].TCRG.BYTE = 0x01;

    /* Configure OCRG
    b15:0 OCRG0 = H'0400 コンペアマッチの値を設定 */
    ATUG.SUBBLOCK[0].OCRG = 0x0400; /* タイマカウンタ 0 のコンペアマッチの値を
    0x0400(1024) に設定 */

    /* Configure TSRG
    b7:2 Reserved
    b1 OVFG0 = 0 オーバフローフラグをクリア
    b0 CMFG0 = 0 コンペアマッチフラグをクリア */
    ATUG.SUBBLOCK[0].TSRG.BYTE &= 0x00; /* 各ステータスフラグをクリア */

    /* Configure TSTRG
    b7:6 Reserved
    b5 STRG5 = 1 TCNTG5 のカウント動作を許可
    b4 STRG4 = 1 TCNTG4 のカウント動作を許可
    b3 STRG3 = 1 TCNTG3 のカウント動作を許可
    b2 STRG2 = 1 TCNTG2 のカウント動作を許可
    b1 STRG1 = 1 TCNTG1 のカウント動作を許可
    b0 STRG0 = 1 TCNTG0 のカウント動作を許可 */
    ATUG.TSTRG.BYTE = 0x3F; /* タイマ G サブブロック 0 のタイマカウンタをスタート*/

    /* Configure PSCR0
    b15:10 Reserved
    b9:0 PSCn = H'9 プリスケーラの分周比を 10(設定値は-1) に設定 */
    ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラの分周比を 10 に設定 */

    /* Configure IPR20
    b15:12 ATU-G(CMIG0-CMIG3) = H'F 対応する割り込みの優先順位を 15 に設定
    b11:8 ATU-G(CMIG4,CMIG5) = H'0 対応する割り込みの優先順位を 0 に設定
    b7:4 ATU-H(CMIH) = H'0 対応する割り込みの優先順位を 0 に設定
    b3:0 Reserved */
    INTC.IPR20.WORD = 0xF000; /* CMIG0 の割り込み優先順位を 15 に設定 */

    set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

    /* Configure ATUENR
    b15:10 Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b9     TJE = 0           タイマ J のカウント動作を停止
b8     THE = 0           タイマ H のカウント動作を停止
b7     TGE = 1           タイマ G のカウント動作を許可
b6     TFE = 0           タイマ F のカウント動作を停止
b5     TEE = 0           タイマ E のカウント動作を停止
b4     TDE = 0           タイマ D のカウント動作を停止
b3     TCE = 0           タイマ C のカウント動作を停止
b2     TBE = 0           タイマ B のカウント動作を停止
b1     TAE = 0           タイマ A のカウント動作を停止
b0     PSCE = 1         プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0081; /* タイマ G およびプリスケアラのカウントをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : CMIG0
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value : none
*****/
void CMIG0(void)
{
    /* Configure TSRG
    b7:2 Reserved
    b1     OVFG0 = 0     オーバフローフラグをクリア
    b0     CMFG0 = 0     コンペアマッチフラグをクリア */
    ATUG.SUBBLOCK[0].TSRG.BYTE &= 0x00; /* 各ステータスフラグをクリア */
} /* End of function CMIG0() */
```

2.21 動作例 20 コンペアマッチ発生による他モジュールの起動[タイマ G]

2.21.1 概要

コンペアマッチ発生時に 1 サイクルの負論理パルス信号を生成し、その信号をトリガとして A/D 変換器を起動します。

2.21.2 使用機能説明

表 2.22に関連レジスタの機能割り付けを示します。

表 2.22 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR1	プリスケアラ 1 の分周比を設定します。
タイマ G レジスタ	TCRG1	タイマ G の各機能を設定します。
	OCRG1	タイマ G のコンペアマッチの値を設定します。
	TCNTG1	TCRG で設定したクロックバスでタイマカウントを行います。
	TSTRG	タイマ G のタイマカウンタをスタートします。
	TSRG1	オーバーフロー、コンペアマッチが発生した場合にフラグがセットされます。
A/D 変換機 レジスタ	ADCSR0	A/D 変換機全体の設定を行います。
	ADCER0	変換したデータを保存するレジスタに関する設定を行います。
	ADANS0	A/D 変換対象チャネルを選択します。
	ADTRE0	割り込み変換要求を禁止/許可します。
	ADTRF0	割り込みトリガ終了フラグです。
	ADREF0	A/D 変換に関するフラグがセットされます。
INTC レジスタ	IPR20	CMIG1 のコンペアマッチ割り込みの優先度を設定します。

2.21.3 動作説明

図 2.36に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルスの周期測定を行います。コンペアマッチが発生するとコンペアマッチパルスが負論理パルスとして生成されます。この時生成された負論理パルスをトリガとして A/D 変換を行います。

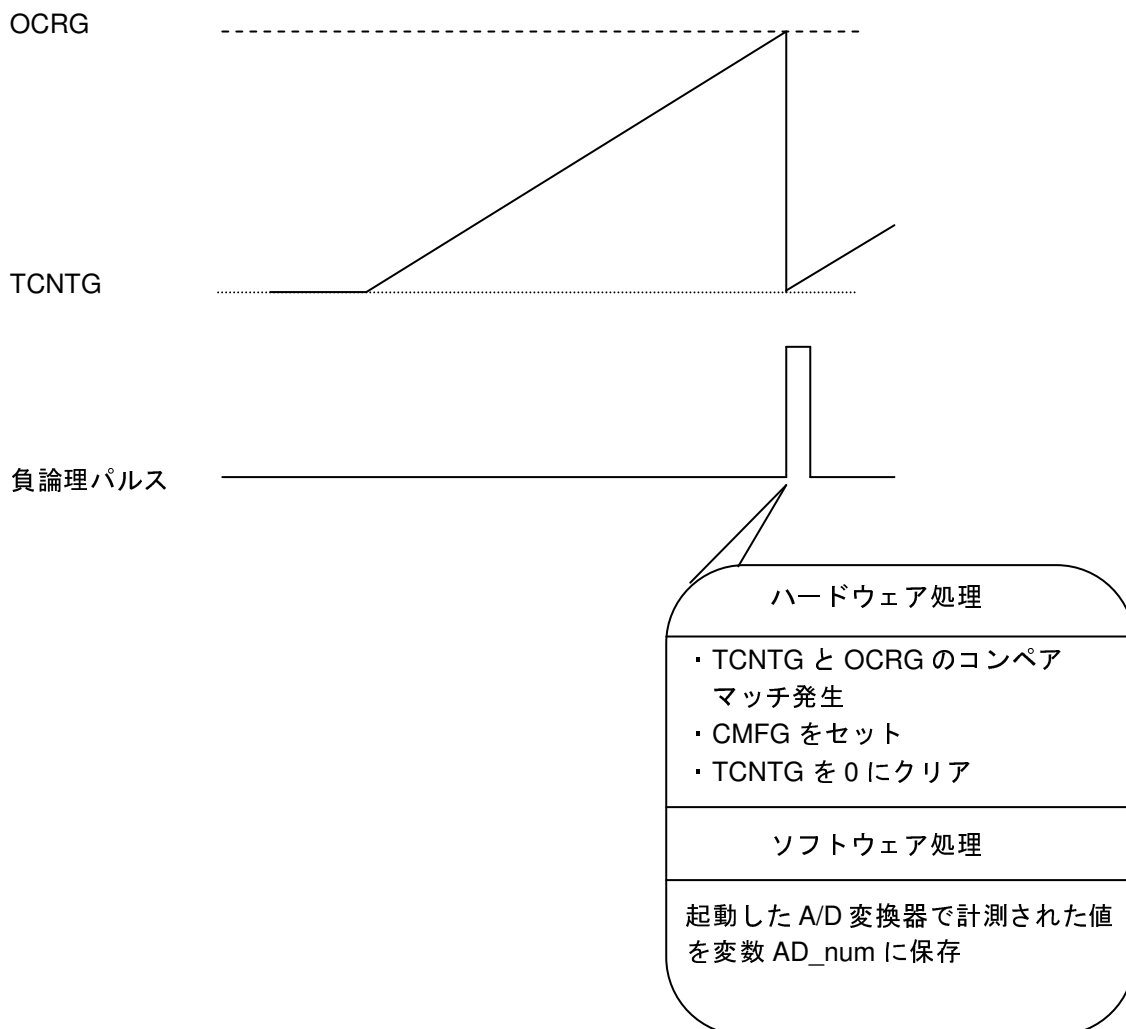


図 2.36 パルスの High 幅計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.21.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
A/D 設定ルーチン	AD_init	A/D 変換器の初期設定を行います。
コンペアマッチ割り込み	CMIG1	TCNTG1 と OCRG1 のコンペアマッチで割り込み、A/D 変換器で変換した値を保存します。

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
AD_num	A/D 変換器で変換した値を保存します。	unsigned long	コンペアマッチ 割り込み

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PSCR1	プリスケアラ 1 の分周比を 10 に設定します。	0x0009	メインルーチン
ATUENR	ATU-IIIのチャンネル G 及びプリスケアラのカウント動作を許可します。	0x0081	
TSTRG	タイマGサブブロック 0 のカウント動作を許可します。	0x02	
TCRG1	「クロックバス 0 でカウント」「コンペアマッチパルスを出力する」「コンペアマッチ時の割り込みを許可」に設定します。	0x03	
OCRG1	TCNTG1 とのコンペアマッチの値を設定します。	0x0400	
TSRG1	オーバフローフラグ、コンペアマッチフラグをクリアします。	0x00	
IPR20	コンペアマッチ割り込み(CMIG1)の優先度を設定します。	0xF000	
ADCSR0	「外部トリガによるスキャン起動を許可」「トリガは ATU-IIIのタイマトリガを使用」に設定します。	0xC2	A/D 変換器 設定ルーチン
ADCER0	「データレジスタのフォーマットは右詰」「オフセットキャンセルの処理を行う」「割り込み変換のトリガソースを ATU-IIIのタイマ G1 トリガ」に設定します。	0x8001	
ADANS0	チャンネル 0 を変換に使用するよう設定します。	0x0001	
ADTRE0	チャンネル 0 への割り込み要求を許可します。	0x0001	
ADREF0	A/D 変換に関するフラグがセットされます。	0x00	
ADTRF0	A/D 割り込みが終了した際にフラグがセットされず。	0x00	

2.21.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform  : SH7254R
* Description    : This is the main tutorial code.
* Operation     : Other modules start with compare match.
*****/
/*****
* History : DD.MM.YYYY Version Description
*         : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
unsigned long AD_num;
void main(void);            /* メインルーチン */
void AD_init(void);        /* A/D 変換設定ルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value  : none
*****/

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
*****/
void main(void)
{
  /* ATU-IIIの設定 */
  /* Configure TCRG
  b7      Reserved
  b6:4 CKSELG0 = 0      クロックバス 0 をクロックソースに指定
  b3:2 Reserved
  b1      CMPOEG0 = 1      TCNTG0 と OCRG0 のコンペアマッチでパルスを出力
  b0      CMEG0 = 1      CMFG0 による割り込みを許可 */
  ATUG.SUBBLOCK[1].TCRG.BYTE = 0x03; /* クロックバス 1 を使用、コンペアマッチ時にコン
  ペアマッチパルスを出力、
                                     コンペアマッチ時の割り込み動作を許可          */

  /* Configure OCRG
  b15:0  OCRG0 = H'0400 コンペアマッチの値を設定 */
  ATUG.SUBBLOCK[1].OCRG = 0x0400; /* タイマカウンタ 1 のコンペアマッチの値を
  0x0400(1024) に設定 */

  /* Configure TSRG
  b7:2 Reserved
  b1      OVFG0 = 0      オーバフローフラグをクリア
  b0      CMFG0 = 0      コンペアマッチフラグをクリア */
  ATUG.SUBBLOCK[1].TSRG.BYTE &= 0x00; /* 各ステータスフラグをクリア
  */

  /* Configure TSTRG
  b7:6 Reserved
  b5      STRG5 = 0      TCNTG5 のカウント動作を停止
  b4      STRG4 = 0      TCNTG4 のカウント動作を停止
  b3      STRG3 = 0      TCNTG3 のカウント動作を停止
  b2      STRG2 = 0      TCNTG2 のカウント動作を停止
  b1      STRG1 = 1      TCNTG1 のカウント動作を許可
  b0      STRG0 = 0      TCNTG0 のカウント動作を停止 */
  ATUG.TSTRG.BYTE = 0x02; /* タイマ G サブブロック 1 のタイマカウンタをスタート
  */

  /* Configure PSCR0
  b15:10 Reserved
  b9:0 PSCn = H'9      プリスケアラの分周比を 10 (設定値は-1) に設定 */
  ATUCTRL.PSCR1.WORD = 0x0009; /* プリスケアラの分周比を 10 に設定          */

  /* A/D 変換の設定 */
  AD_init(); /* A/D 変換設定ルーチン          */

  /* Configure IPR20
  b15:12 ATU-G(CMIG0-CMIG3) = H'F  対応する割り込みの優先順位を 15 に設定
  b11:8  ATU-G(CMIG4,CMIG5) = H'0  対応する割り込みの優先順位を 0 に設定
  b7:4  ATU-H(CMIH) = H'0      対応する割り込みの優先順位を 0 に設定
  b3:0 Reserved */
  INTC.IPR20.WORD = 0xF000; /* CMIG1 の割り込み優先順位を 15 に設定          */

  set_imask(0); /* 割り込みマスクレベルを 0 に設定          */

  /* Configure ATUENR
  b15:10 Reserved
  b9      TJE = 0      タイマ J のカウント動作を停止
```

```

b8     THE = 0     タイマ H のカウント動作を停止
b7     TGE = 1     タイマ G のカウント動作を許可
b6     TFE = 0     タイマ F のカウント動作を停止
b5     TEE = 0     タイマ E のカウント動作を停止
b4     TDE = 0     タイマ D のカウント動作を停止
b3     TCE = 0     タイマ C のカウント動作を停止
b2     TBE = 0     タイマ B のカウント動作を停止
b1     TAE = 0     タイマ A のカウント動作を停止
b0     PSCE = 1   プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0081; /* タイマ G およびプリスケアラのカウントをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : AD_init
* Description   : A/D 変換器設定
* Arguments    : none
* Return Value  : none
*****/
void AD_init(void)
{
    /* Configure ADCSR
b7     ADST = 1     スキャン変換開始
b6     ADCS = 1     連続スキャンモード
b5     Reserved
b4     ADIE = 0     スキャン終了後の ADI 割り込み発生の禁止
b3:2 Reserved
b1     TRGE = 1     外部トリガまたは ATU-III のタイマトリガによるスキャン変換の起動を許可
b0     EXTRG = 0    ATU-III のタイマトリガによるスキャン変換の起動 */
ADCA.ADCSR.BYTE = 0xC2; /* 外部トリガによるスキャン変換の起動を許可
                        外部トリガは ATU-III のタイマトリガを使用 */

    /* Configure ADCER
b15    ADRFMT = 1   A/D データレジスタのフォーマットを右詰めに設定
b14:12 Reserved
b11    DIAGM = 0    A/D 変換器の自己診断を行わない
b10    DIAGLD = 0   自己診断は自動的にローテーションして実施
b9:8   DIAGVAL = 0  リザーブ (未設定状態)
b7     CKS = 0     A/D 変換ステート数は 50 ステート
b6:1   Reserved
b0     ITTRGS = 1   ATU-III のタイマ G1 トリガで AN0 の割り込み変換の起動を行う */
ADCA.ADCER.WORD = 0x8001; /* データレジスタのフォーマットは右詰め
                        オフセットキャンセル処理の要求を行う
                        割り込み変換のトリガソースを ATU-III のタイマ G1 トリガに設定 */

    /* Configure ADANS0
b15:0  ANSn = H'0001 スキャン変換用チャネル ANS0 を使用 */
ADCA.ADANS0.WORD = 0x0001; /* チャネル 0 を変換に使用 */

    /* Configure ADTRE
b15:0  ANSn = H'0001 AN0 に ATU-III タイマ又はソフトトリガからの割り込み要求許可 */
ADCA.ADTRE.WORD = 0x0001; /* チャネル 0 への割り込み要求を許可 */

    /* Configure ADREF
b7     ADSCACT = 0   スキャン変換中フラグをクリア

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b6      ADITACT = 0      割り込み変換中フラグをクリア
b5:1 Reserved
b0      ADF = 0          スキャン変換終了状態をクリア */
ADCA.ADREF.BYTE  &= 0x00;      /* 各変換ステータスフラグをクリア (アイドル中) */

/* Configure ADTRF
b15:0  ADTFn = 0      割り込み変換終了状態をクリア */
ADCA.ADTRF.WORD  &= 0x0000;      /* 各割り込みトリガ終了フラグをクリア      */
} /* End of function AD_init() */

/*****
* Function Name : CMIG1
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value : none
*****/
void CMIG1(void)
{
    /* Configure TSRG
b7:2 Reserved
b1      OVFG0 = 0      TCNTG0 オーバフローフラグをクリア
b0      CMFG0 = 0      コンペアマッチフラグをクリア */
    ATUG.SUBBLOCK[1].TSRG.BYTE  &= 0x00;      /* 各ステータスフラグをクリア      */
    AD_num = ADCA.ADR0;          /* AD 変換器で保存した値を変数 AD_num に保存      */
} /* End of function CMIG1() */
```

2.22 動作例 21 コンペアマッチ発生による他モジュールの起動[タイマ G]

2.22.1 概要

- 1) コンペアマッチ発生時に、1 サイクルの負論理パルス信号を生成し、A/D 変換器を起動します。
- 2) コンペアマッチ割り込みで DMAC を起動し、A/D 変換した値を転送します。
- 3) DMAC の転送終了割り込みで DMAC の送信先アドレスを加算します。

2.22.2 使用機能説明

表 2.23に関連レジスタの機能割り付けを示します。

表 2.23 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケラ0の動作を設定します。
	PSCR0	プリスケラ0の分周比を設定します。
タイマ G レジスタ	TCRG1	タイマ G の各機能を設定します。
	OCRG1	タイマ G のコンペアマッチ値を設定します。
	TCNTG1	TCRG で設定したクロックパスでタイマカウントを行います。
	TSRG1	オーバーフロー、コンペアマッチが発生した場合にフラグがセットされます。
A/D 変換機 レジスタ	ADCSR0	A/D 変換機全体の設定を行います。
	ADCER0	変換したデータを保存するレジスタに関する設定を行います。
	ADREF0	A/D 変換に関するフラグがセットされます。
	ADTRF0	割り込み変換が終了した際にフラグがセットされます。
	ADANS0	変換するチャンネルを設定します。
	ADTRE0	各チャンネルへの割り込み要求の許可を設定します。
DMAC レジスタ	CHCR2	DMAC の各機能設定を行います。
	DMARS2	拡張リソースを設定します。
	SAR2	転送元のアドレスを設定します。
	RSAR2	リロード時の転送元アドレスを設定します。
	DAR2	転送先のアドレスを設定します。
	RDAR2	リロード時の転送先アドレスを設定します。
	DMATCR2	転送時のデータの転送回数を設定します。
	RDMATCR2	リロード時の転送回数を設定します。
	CHFR2	DMAC の転送終了フラグが格納されます。
	DMAOR	DMAC の動作許可を設定します。
INTC レジスタ	IPR03	DMAC の転送終了割り込みの優先度を設定します。

2.22.3 動作説明

図 2.37に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってパルスの周期測定を行います。タイマ G のカウント開始前に、A/D 変換器はタイマ G コンペアマッチ時に発生する負論理パルスで起動し、DMAC はタイマ G のコンペアマッチで起動するように設定します。

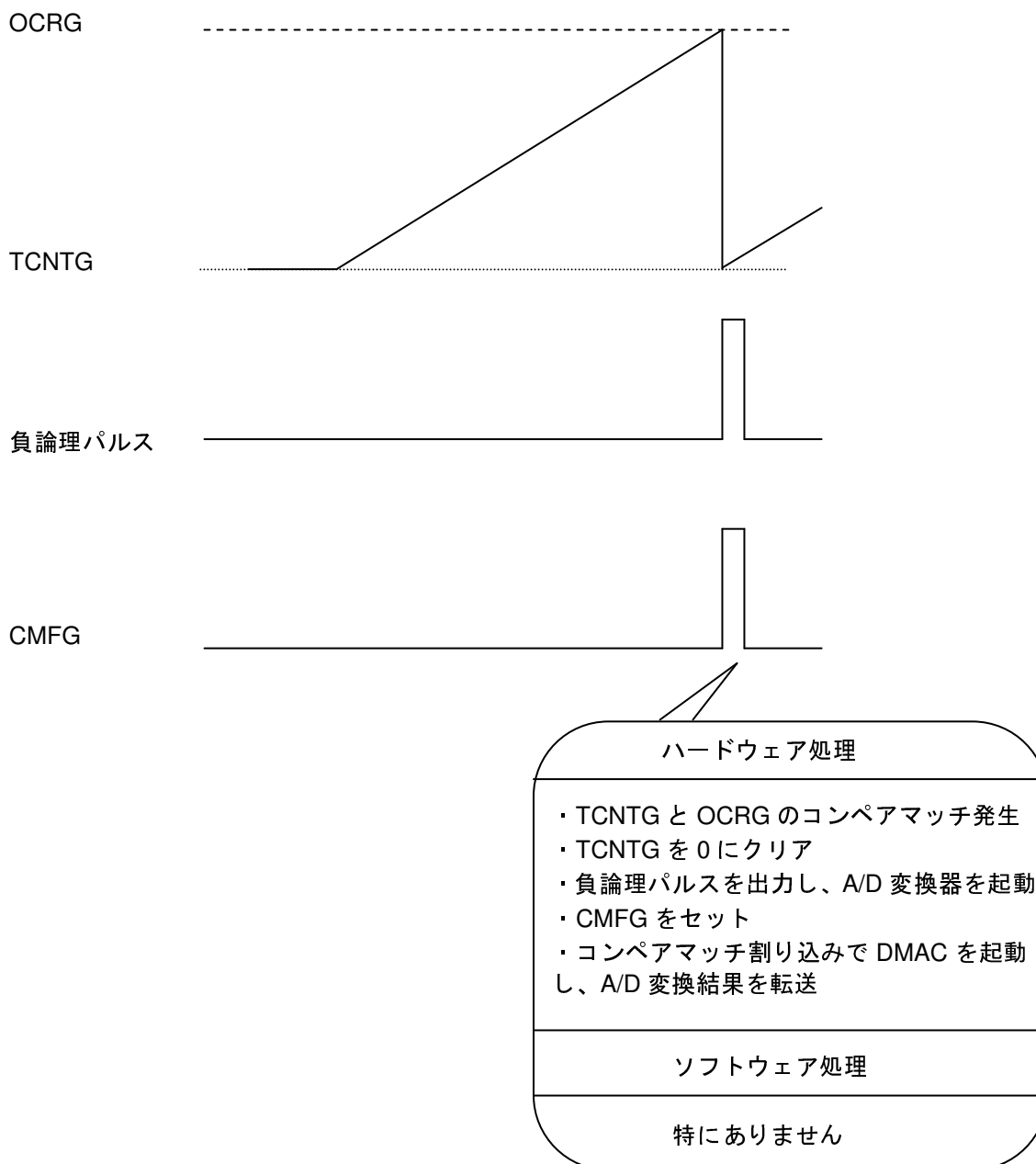


図 2.37 コンペアマッチからの A/D 変換器起動の動作原理

2.22.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
A/D 設定ルーチン	AD_init	A/D 変換器の初期設定を行います。
DMAC 設定ルーチン	DMA_init	DMA 転送の初期設定を行います。
DMA 転送終了 割り込み	DEI2	DMA 転送終了で割り込み。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	メインルーチン
ATUENR	ATU-IIIチャンネル G 及びプリスケアラのカウンタ動作を許可します。	0x0081	
TSTRG	タイマ G サブブロック 0 のカウンタ動作を許可します。	0x01	
TSRG1	オーバフローフラグと、コンペアマッチフラグをクリアします。	0x00	
TCRG1	「クロックバス 0 でカウント」「コンペアマッチパルスを出力する」「コンペアマッチ時の割り込みを許可」に設定します。	0x03	
OCRG1	TCNTG0 とのコンペアマッチの値を設定します。	0x0400	A/D 変換器設定 ルーチン
ADCSR0	「外部トリガによるスキヤン起動を許可」「トリガは ATU-III のタイマトリガを使用」に設定します。	0x82	
ADCER0	「データレジスタのフォーマットは右詰」「オフセットキャンセルの処理を行う」「割り込み変換のトリガソースを ATU-III のタイマ G1 トリガ」に設定します。	0x8001	
ADANS0	チャンネル 0 を変換に使用するよう設定します。	0x0001	
ADTRE0	チャンネル 0 への割り込み要求を許可します。	0x0001	
ADREF0	A/D 変換に関するフラグがセットされます。	0x00	
ADTRF0	A/D 割り込みが終了した際にフラグがセットされます。	0x0000	
IPR20	コンペアマッチ割り込み(CMIG0)の優先度を設定します。	0xF000	DMAC 設定 ルーチン
CHCR2	DMAC の転送モード設定としてワード転送、バースト転送、転送元/転送先のアドレスを設定します。	0x1000082C	
DMARS2	CH2 の拡張リソースを ATU III G1 に設定します。	0x0047	
SAR2, RSAR2	転送元として ADR0 のアドレスを設定します。	0xFFFFE840	
DAR2, RADR2	転送先として任意 RAM 領域のアドレスを設定します。	0xFFFF82000	
DMATCR2, RDMATCR2	データの転送回数を 1 回に設定します。	0x0001	
IPR03	DMA 転送終了割り込み(DEI2)の優先度を設定します。	0x00F0	
CHFR2	DMA の終了フラグをクリアします。	HE = 0	DMA 転送終了

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

		TE = 0	割り込み
TSRG1	タイマ G のコンペアマッチフラグをクリアします。	0x00	

2.22.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform   : SH7254R
* Description     : This is the main tutorial code.
* Operation      : Other modules start with compare match.
*****/
/*****
* History : DD.MM.YYYY Version Description
*          : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
void main(void);            /* メインルーチン */
void AD_init(void);         /* A/D 変換設定ルーチン */
void DMA_init(void);        /* DMAC 設定ルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value  : none
*****/

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
*****/
void main(void)
{
  /* ATUⅢの設定 */
  /* Configure TCRG
  b7      Reserved
  b6:4 CKSELG0 = 0      クロックバス 0 をクロックソースに指定
  b3:2 Reserved
  b1      CMPOEG0 = 1      TCNTG0 と OCRG0 のコンペアマッチでパルス出力
  b0      CMEG0 = 1      CMFG0 による割り込みを許可 */
  ATUG.SUBBLOCK[1].TCRG.BYTE = 0x03; /* クロックバス 0 を使用、コンペアマッチ時にコンペアマッ
  チパルスを出力
                                     コンペアマッチ時の割り込み動作を許可          */

  /* Configure OCRG
  b15:0  OCRG0 = H'0400 コンペアマッチの値を設定 */
  ATUG.SUBBLOCK[1].OCRG = 0x0400; /* タイマカウンタ 0 のコンペアマッチの値を
  0x0400(1024) に設定 */

  /* Configure TSRG
  b7:2 Reserved
  b1      OVFG0 = 0      TCNTG0 オーバフローフラグをクリア
  b0      CMFG0 = 0      コンペアマッチフラグをクリア */
  ATUG.SUBBLOCK[1].TSRG.BYTE &= 0x00; /* 各ステータスフラグをクリア          */

  /* Configure TSTRG
  b7:6 Reserved
  b5      STRG5 = 1      TCNTG5 のカウント動作を停止
  b4      STRG4 = 1      TCNTG4 のカウント動作を停止
  b3      STRG3 = 1      TCNTG3 のカウント動作を停止
  b2      STRG2 = 1      TCNTG2 のカウント動作を停止
  b1      STRG1 = 1      TCNTG1 のカウント動作を許可
  b0      STRG0 = 1      TCNTG0 のカウント動作を停止 */
  ATUG.TSTRG.BYTE = 0x02; /* タイマ G サブブロック 1 のタイマカウンタをスタート*/
  /* Configure PSCR0
  b15:10 Reserved
  b9:0 PSCn = H'9      プリスケアラの分周比を 10(設定値は-1) に設定 */
  ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケアラの分週比を 10 に設定          */

  /* 他モジュール(AD, DMAC)の設定 */
  AD_init(); /* AD 変換器設定ルーチンへ          */
  DMA_init(); /* DMAC 設定ルーチンへ          */

  set_imask(0); /* 割り込みマスクレベルを 0 に設定          */

  /* Configure ATUENR
  b15:10 Reserved
  b9      TJE = 0      タイマ J のカウント動作を停止
  b8      THE = 0      タイマ H のカウント動作を停止
  b7      TGE = 1      タイマ G のカウント動作を許可
  b6      TFE = 0      タイマ F のカウント動作を停止
  b5      TEE = 0      タイマ E のカウント動作を停止
  b4      TDE = 0      タイマ D のカウント動作を停止
  b3      TCE = 0      タイマ C のカウント動作を停止
  b2      TBE = 0      タイマ B のカウント動作を停止
  b1      TAE = 0      タイマ A のカウント動作を停止
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b0      PSCE = 1 プリスケアラのカウンタ動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0081; /* タイマ G およびプリスケアラのカウンタをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : AD_init
* Description   : A/D 変換器設定ルーチン
* Arguments    : none
* Return Value  : none
*****/
void AD_init(void)
{
    /* Configure ADCSR
b7      ADST = 1     スキャン変換開始
b6      ADCS = 1     1 サイクルスキャンモード
b5      Reserved
b4      ADIE = 0     スキャン終了後の ADI 割り込み発生の禁止
b3:2 Reserved
b1      TRGE = 1     外部トリガまたは ATU-III のタイマトリガによるスキャン変換の起動を許可
b0      EXTRG = 0    ATU-III のタイマトリガによるスキャン変換の起動 */
ADCA.ADCSR.BYTE = 0x82; /* 外部トリガによるスキャン変換の起動を許可
                        外部トリガは ATU-III のタイマトリガを使用 */

    /* Configure ADCER
b15     ADRFMT = 1   A/D データレジスタのフォーマットを右詰めに設定
b14:12 Reserved
b11     DIAGM = 0    A/D 変換器の自己診断を行わない
b10     DIAGLD = 0   自己診断は自動的にローテーションして実施
b9:8    DIAGVAL = 0  リザーブ (未設定状態)
b7      CKS = 0      A/D 変換ステータス数は 50 ステータス
b6:1    Reserved
b0      ITTRGS = 1   ATU-III のタイマ G1 トリガで AN0 の割り込み変換の起動を行う */
ADCA.ADCER.WORD = 0x8001; /* データレジスタのフォーマットは右詰め
                        オフセットキャンセル処理の要求を行う
                        割り込み変換のトリガソースを ATU-III のタイマ G1 トリガに設定 */

    /* Configure ADANS0
b15:0   ANSn = H'0001 スキャン変換用チャンネル ANS0 を使用 */
ADCA.ADANS0.WORD = 0x0001; /* チャンネル 0 を変換に使用 */

    /* Configure ADTRE
b15:0   ANSn = H'0001 AN0 に ATU-III タイマ又はソフトトリガからの割り込み要求許可 */
ADCA.ADTRE.WORD = 0x0001; /* チャンネル 0 への割り込み要求を許可 */

    /* Configure ADREF
b7      ADSCACT = 0   スキャン変換中フラグをクリア
b6      ADITACT = 0   割り込み変換中フラグをクリア
b5:1    Reserved
b0      ADF = 0       スキャン変換終了状態をクリア */
ADCA.ADREF.BYTE &= 0x00; /* 各変換ステータスフラグをクリア */

    /* Configure ADTRF
b15:0   ADTFn = 0    割り込み変換終了状態をクリア */
ADCA.ADTRF.WORD &= 0x0000; /* 各割り込みトリガ終了フラグをクリア */
} /* End of function AD_init() */
```

```

/*****
* Function Name : DMA_init
* Description   : DMAC 設定ルーチン
* Arguments    : none
* Return Value  : none
*****/
void DMA_init(void)
{
    /* Configure CHCR
    b31:30 TC = 10          一回の転送要求で、DMATCR の設定回数分 n 転送
    b29     Reserved
    b28     RLD1 = 0       リロード機能 1 無効
    b27:26 RLD2 = 00      リロード機能 2 無効
    b25     Reserved
    b24     IFT = 0        Integer→Floating-point 変換機能は無効
    b23:19 Reserved
    b18     HIE = 0        DMATCR =(転送前にセットした DMATCR)/2 の時、CPU への割り込み要求を禁止
    b17:16 Reserved
    b15:14 DM = 00         デスティネーションアドレスは固定 (16 バイト単位転送時は設定禁止)
    b13:12 SM = 00         ソースアドレスは固定 (16 バイト単位転送時は設定禁止)
    b11:8   RS = 1000     DMA 拡張リソースセクタ
    b7:6    Reserved
    b5      TB = 1         バーストモードで転送
    b4:3    TS = 01       ワード単位で転送
    b2      IE = 1        割り込み要求を許可
    b1      Reserved
    b0      DE = 0        DMA 転送を禁止 */
    DMAC2.CHCR.LONG = 0x1000082C;          /* 転送サイズ:ワード リソース:拡張リソースセクタ
                                           DMAC 転送終了割り込みを許可
                                           バースト転送 転送先のアドレスを固定 */

    /* Configure DMARS
    b15:10 CH3 MID = H'0    (default)DMA 転送要求元を ATU-III_C0 に指定
    b9:8   CH3 RID = H'0    (default)DMA 転送要求元を ATU-III_C0 に指定
    b7:2   CH2 MID = 0100   DMA 転送要求元を ATU-III_G1 に指定
    b1:0   CH2 RID = 0111   DMA 転送要求元を ATU-III_G1 に指定 */
    DMACCOM.DMARSx[1].DMARS.WORD = 0x0047; /* CH2 の拡張リソースを ATUIII_G1 に設定*/

    /* Configure SAR0
    b31:0   DMA ソースアドレス */
    DMAC2.SAR = 0xFFFFE840;          /* 転送元のアドレスを指定 (AD0 の値) */

    /* Configure DAR
    b31:0   DMA デスティネーションアドレス */
    DMAC2.DAR = 0xFFFF8200;          /* 転送先のアドレスを指定 */

    /* Configure DMATCR
    b31:24 Reserved
    b23:0   DMA 転送回数 = 1 */
    DMAC2.DMATCR = 0x0001;          /* DMA 転送の回数を設定 */

    /* Configure RSAR
    b31:0   DMA リロード時のソースアドレス */
    DMAC2.RSAR = 0xFFFFE840;          /* 各値のリロード値を設定 */

```

SH7254R グループ アドバンストタイムユニット-III (ATU-III) を使用したタイマ動作例

```
/* Configure DRAR
b31:0 DMA リロード時のデスティネーションアドレス */
DMAC2.RDAR = 0xFFFF82000;

/* Configure RDMATCR
b31:24 Reserved
b23:0 DMA リロード時の転送回数 = 1 */
DMAC2.RDMATCR = 0x0001;

/* Configure IPR03
b15:12 DMAC0 = H'0 対応する割り込みの優先順位を 0 に設定
b11:8 DMAC1 = H'0 対応する割り込みの優先順位を 0 に設定
b7:4 DMAC2 = H'F 対応する割り込みの優先順位を 15 に設定
b3:0 DMAC3 = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR03.WORD = 0x00F0; /* DMAC2 転送終了割り込みの優先順位を 15 に設定 */

/* Configure CHFR
b7:5 Reserved
b4 HE = 0 DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0 TE = 0 DMA トランスファエンドフラグをクリア*/
DMAC2.CHFR.BYTE ^= 0x11; /* DMAC の転送終了フラグをクリア */

/* Configure CHCR
DE = 1 DMA 転送を許可*/
DMAC2.CHCR.BIT.DE = 1; /* DMAC チャンネル 2 の動作を許可 */

/* Configure DMAOR
DME = 1 全てのチャンネルの DMA 転送を許可 */
DMACCOM.DMAOR.BIT.DME = 1; /* DMAC の動作を許可 */
} /* End of function DMA_init() */

/*****
* Function Name : DEI2
* Description : DMAC 転送終了割り込み
* Arguments : none
* Return Value : none
*****/
void DEI2(void)
{
/* Configure CHFR
b7:5 Reserved
b4 HE = 0 DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0 TE = 0 DMA トランスファエンドフラグをクリア*/
DMAC2.CHFR.BYTE ^= 0x11; /* DMAC の転送終了フラグをクリア */

/* Configure TSRG
b7:2 Reserved
b1 OVFG0 = 0 オーバフローフラグをクリア
b0 CMFG0 = 0 コンペアマッチフラグをクリア */
ATUG.SUBBLOCK[1].TSRG.BYTE &= 0x00; /* 各ステータスフラグをクリア */
} /* End of function DEI2() */
```

2.23 動作例 22 コンペアマッチ割り込み発生[タイマ H]

2.23.1 概要

タイマカウンタ 1H がコンペアマッチを発生する度に、タイマカウンタ 2H がカウントアップを行うことで、タイマカウンタ 1H のコンペアマッチ回数を計測します。

2.23.2 使用機能説明

表 2.24に 関連レジスタの機能割り付けを示します。

表 2.24 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ H レジスタ	TCRH	タイマ H の各機能を設定します。
	OCR1H	タイマカウンタ 1H のコンペアマッチの値を設定します。
	TSRH	オーバーフロー、コンペアマッチが発生した場合にフラグがセットされます。
	TCNT1H	選択したクロックバスでアップカウント動作を行います。
	TCNT2H	TCNT1H と OCR1H のコンペアマッチ発生によってアップカウント動作を行います。
INTC レジスタ	IPR20	CMFH のコンペアマッチ割り込みの優先度を設定します。

2.23.3 動作説明

図 2.38に動作原理を示します。これに従って SH72546 のハードウェアおよびソフトウェア処理によってコンペアマッチ回数の計測動作を発生させます。タイマカウンタ 2Hは、タイマカウンタ 1Hのコンペアマッチ発生ごとにインクリメントされます。

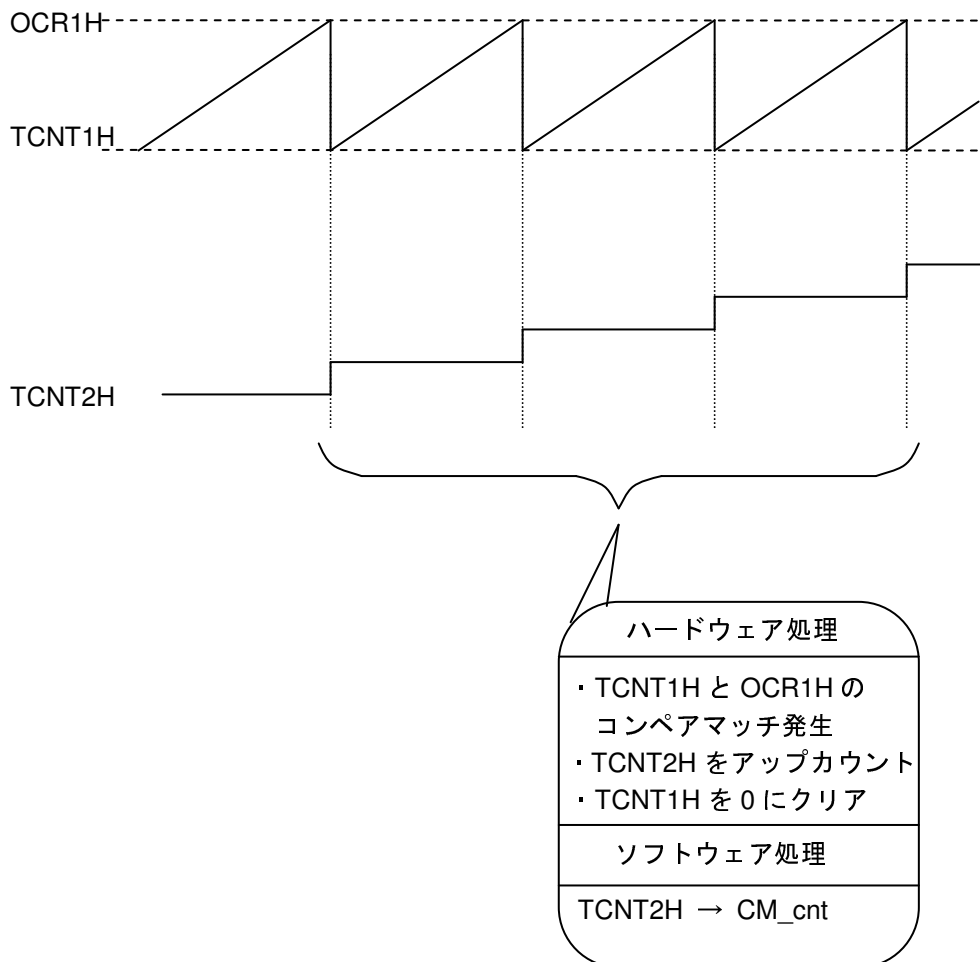


図 2.38 コンペアマッチ回数測定動作原理

2.23.4 ソフトウェア説明

• 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	ATU-IIIの初期設定を行います。
コンペアマッチ割り込み	CMIH	TCNT1H と OCR1H のコンペアマッチ発生で割り込み、TCNT2H の値を変数に保存します。

• 使用変数の説明

ラベル名	機能	データ長	使用モジュール名
CM_cnt	コンペアマッチの発生回数を保存します。	unsigned long	コンペアマッチ 割り込み

• 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PSCR0	プリスケアラ 0 の分周比 10 に設定します。	0x0009	メインルーチン
ATUENR	ATU-IIIチャンネルH 及びプリスケアラのカウンタ動作を許可します。	0x0101	
TCRH	「クロックバス 0 でカウンタ」「コンペアマッチ時の割り込みを許可」に設定します。	0x01	
IPR20	CMFH による割り込み(CMIH)の優先順位を 15 に設定します。	0x00F0	
OCR1H	TCNT1H とのコンペアマッチの値を設定します。	0x0300	
TSRH	タイマ H のコンペアマッチフラグをクリアします。	0x00	コンペアマッチ 割り込み

2.23.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Compare-match interrupt occurs.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned long CM_cnt;
void main(void);            /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
    /* ATU-IIIの設定 */
    /* Configure TCRH
    b7      Reserved
    b6:4 CKSELH[2:0] = 000 クロックバス 0
    b3:1 Reserved
    b0      CMEH = 1          CMFH による割り込みを許可 */
    ATUH.TCRH.BYTE = 0x01;    /* クロックバス 0 で動作 CMFH による割り込みを許可 */

    /* Configure OCR1H
    b15:0  OCR1H[15:0]      コンペアマッチ値を指定 */
    ATUH.OCR1H = 0x0300;    /* タイマカウンタ 1H のコンペアマッチの値を設定 */

    /* Configure PSCR0
    b15:10 Reserved
    b9:0 PSC0[9:0] = H'9     プリスケーラの分周比を 10 (設定値は-1) に設定 */
    ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラ 0 の分周比を 10 に設定 */

    /* Configure IPR20
    b15:12 ATU-G (CMIG0~CMIG3) = H'0     対応する割り込みの優先順位を 0 に設定
    b11:8  ATU-G (CMIG4、CMIG5) = H'0     対応する割り込みの優先順位を 0 に設定
    b7:4  ATU-H (CMIH) = H'F             対応する割り込みの優先順位を 15 に設定
    b3:0  Reserved */
    INTC.IPR20.WORD = 0x00F0; /* CMFH による割り込み (CMIH) の優先順位を 15 に設定 */

    set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

    /* Configure ATUENR
    b15:10 Reserved
    b9      TJE = 0     タイマ J のカウント動作を停止
    b8      THE = 1     タイマ H のカウント動作を許可
    b7      TGE = 0     タイマ G のカウント動作を停止
    b6      TFE = 0     タイマ F のカウント動作を停止
    b5      TEE = 0     タイマ E のカウント動作を停止
    b4      TDE = 0     タイマ D のカウント動作を停止
    b3      TCE = 0     タイマ C のカウント動作を停止
    b2      TBE = 0     タイマ B のカウント動作を停止
    b1      TAE = 0     タイマ A のカウント動作を停止
    b0      PSCE = 1   プリスケーラのカウント動作を許可 */
    ATUCTRL.ATUENR.WORD = 0x0101; /* タイマ H およびプリスケーラのカウントをスタート */

    while(1);
} /* End of function main() */

/*****
* Function Name : CMIH
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value  : none
*****/
void CMIH(void)
{
    /* Configure TSRH
    b7-3 Reserved
    b2      OV2H = 0     オーバフローフラグをクリア
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b1      OVF1H = 0    オーバフローフラグをクリア
b0      CMFH = 0     コンペアマッチフラグをクリア */
ATUH.TSRH.BYTE &= 0x00; /* コンペアマッチフラグをクリア */

/* Configure TCNT2H
b31-0  TCNT2H[31:0] TCNT1H と OCR1H とのコンペアマッチ発生回数 */
CM_cnt = ATUH.TCNT2H;
} /* End of function CMIH() */
```

2.24 動作例 23 FIFO レジスタへの値の格納[タイマ J]

2.24.1 概要

- 1) 外部からの信号を受信する度にカウンタの値を FIFO レジスタに保存します。
- 2) FIFO レジスタに格納したデータの個数が設定値以上になると、FIFO データフル割り込みで DMAC を起動し、格納された値を転送し、転送した FIFO レジスタに格納されていた値をクリアします。

2.24.2 使用機能説明

表 2.25に関連レジスタの機能割り付けを示します。

表 2.25 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ J レジスタ	TCRJ0	タイマ J の各機能を設定します。
	FGRJ0	FIFO レジスタの動作を設定します。
	TSTRJ0	タイマ J サブブロックの動作を許可します。
	TIERJ0	データフル時の割り込みを許可します。
	TCNTJ0	選択したクロックバスでアップカウント動作を行います。
	FIFOJ0	コンペアマッチ発生時に TCNTJ0 の値を保存します。
	FDNRJ0	FIFOJ0 に保存されている数の個数が読み出されます。
	OCRJ0	TCNTJ0 のコンペアマッチの値を設定します。
	TSRJ0	タイマ J サブブロック 0 のステータスレジスタです。
DMAC レジスタ	CHCR1	DMAC の各機能の設定をします。
	DMAOR	転送時の各チャンネルの動作を設定します。
	DMARS0	DMAC の拡張リソースを設定します。
	SAR1	DMAC 転送の転送先のアドレスを設定します。
	RSAR1	リロード時の転送先アドレス値を設定します。
	DAR1	DMAC 転送の転送元のアドレスを設定します。
	RDAR1	リロード時の転送元アドレス値を設定します。
	DMATCR1	DMAC 転送の転送回数を設定します。
	RDMATCR1	リロード時の転送回数を設定します。
CHFR1	DMAC の各フラグがセットされます。	
PFC レジスタ	PDGR2	ポート D の端子機能を設定します。
	PECR1	ポート E の端子機能を設定します。
INTC レジスタ	IPR03	DMAC の転送終了割り込みの優先度を設定します。
	IPR21	タイマ J データフル割り込みの優先度を設定します。

2.24.3 動作説明

図 2.39に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってコンペアマッチ回数の計測動作を発生させます。（図は FCRJ0 で設定した基準値が 4 の場合）

DMAC の転送元アドレスを FIFO レジスタに設定することで、FIFO に格納されたカウンタの値を取得します。

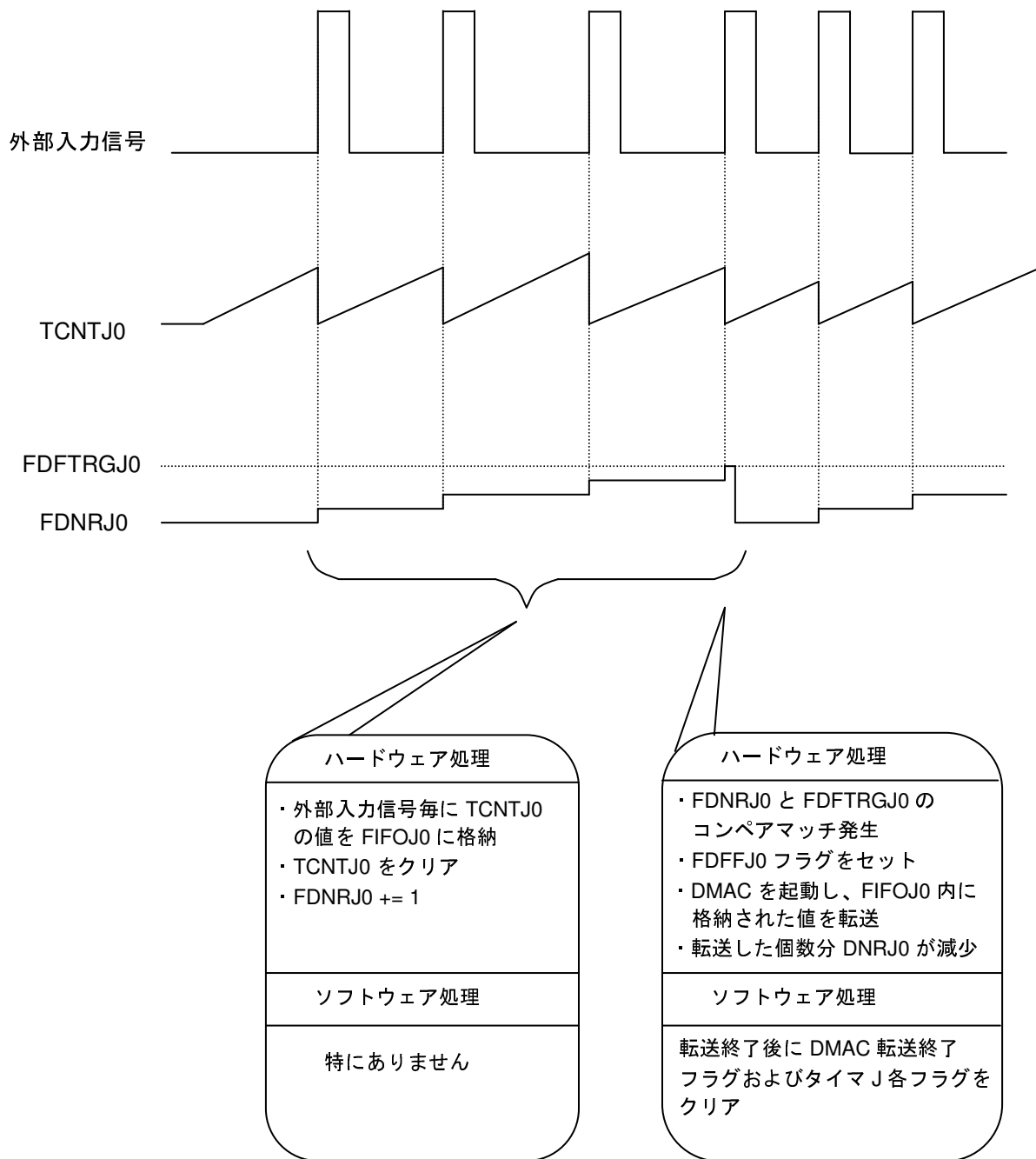


図 2.39 パルスの High 幅計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.24.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
DMAC 設定ルーチン	DMAC_init	TCNTJ0 と OCRJ0 のコンペアマッチ発生で割り込み要求が発生し、TCNT2H の値を変数に保存します。
DMAC 転送終了割り込み	DEI1	DMAC の転送が終了すると割り込み要求が発生し、各フラグのクリア、DMAC の送信先アドレスの増加を行います。

- 使用変数の説明

このタスクでは変数を使用していません。

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	メインルーチン
PDCR2	PD12 を TIJ0 入力に設定します。	0x0300	
PECR1	PE6 を TOE00 出力に設定します。	0x1000	
ATUENR	ATU-IIIのチャンネルJ及びプリスケアラのカウント動作を許可します。	0x0201	
TCRJ0	「クロックバス 0 でカウント」「ノイズキャンセルを行わない」「対立ち上がりエッジ」に設定します。	0x01	
FCRJ0	「FIFO レジスタを常に有効」「FIFO レジスタに格納する値の個数を 6 個」に設定します	0x81	
TSRJ0	各ステータスフラグをクリアします。	0x00	
TIERJ0	FIFO データフル時の割り込みを許可します。	0x01	
TSTRJ	タイマ J サブブロック [0] を動作させる。	0x01	
CHCR1	「転送サイズは word 単位」「転送開始リソースは拡張リソースセレクタ」「転送方式はバースト転送」「MATCR をリロード」「転送先のアドレスを転送毎に増加」「転送終了割り込みを許可」「このチャンネルの動作を許可」に設定します。	0x9000482D	DMAC 設定ルーチン
DMARS0	DMAC チャンネル 1 の拡張リソースを「ATU-III J0」に設定します。	0xE300	
SAR1	DMA 転送元アドレスを設定します。	0xFFFFFFFF9C	
DAR1	DMA 転送先アドレスを設定します。	0xFFF80000	
DMATCR1	DMA 転送の回数を設定します。	0x0006	
RSAR1	DMA 転送元アドレスのリロード値を設定します。	0xFFFFFFFF9C	
RDAR1	DMA 転送先アドレスのリロード値を設定します。	0xFFF80000	
RDMATCR1	DMA 転送の回数のリロード値を設定します	0x006	
CHFR1	DMA 転送に関する各フラグがセットされます。	—	
DMAOR	DMAC 全体の動作の許可を設定します	0x0001	
TSRJ0	各ステータスフラグをクリアします。	0x00	DMAC 転送終了 割り込み
CHFR1	DMAC の転送終了フラグをクリアします。	HE = 0 TE = 0	

2.24.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform  : SH7254R
* Description    : This is the main tutorial code.
* Operation     : Store a value into FIFO register.
*****/
/*****
* History : DD.MM.YYYY Version Description
*         : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
void main(void);             /* メインルーチン */
void DMAC_init(void);        /* DMA 設定ルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value  : none
*****/

```

```

void main(void)
{
  /* ポートの設定 */
  /* Configure PDCR2
  b15:12 Reserved
  b11:10 PD13MD[1:0] = 00 PD13 入出力 (ポート)
  b9:8 PD12MD[1:0] = 11 TIJ0 入力 (ATU-III)
  b7:6 PD11MD[1:0] = 00 PD11 入出力 (ポート)
  b5:4 PD10MD[1:0] = 00 PD10 入出力 (ポート)
  b3:2 PD9MD[1:0] = 00 PD9 入出力 (ポート)
  b1:0 PD8MD[1:0] = 00 PD8 入出力 (ポート) */
  PORTD.CR2.WORD = 0x0300; /* PD12 を TIJ0 入力に設定 */

  /* Configure PECCR1
  b15:14 PE7MD[1:0] = 00 PE7 入出力 (ポート)
  b13:12 PE6MD[1:0] = 01 TOE00 出力 (ATU-III)
  b11 Reserved
  b10 PE5MD = 0 PE5 入出力 (ポート)
  b9 Reserved
  b8 PE4MD = 0 PE4 入出力 (ポート)
  b7 Reserved
  b6 PE3MD = 0 PE3 入出力 (ポート)
  b5:4 PE2MD[1:0] = 00 PE2 入出力 (ポート)
  b3:2 PE1MD[1:0] = 00 PE1 入出力 (ポート)
  b1 Reserved
  b0 PE0MD = 0 PE0 入出力 (ポート) */
  PORTE.CR1.WORD = 0x1000; /* PE6 を TOE00 出力に設定 */

  /* ATU-III の設定 */
  /* Configure TCRJ0
  b7 Reserved
  b6:4 CKSELJ0[2:0] = 000 クロックバス 0
  b3 Reserved
  b2 NCEJ0 = 0 TIJ0 入力のノイズキャンセラ機能を無効に
  b1:0 IOJ0[1:0] = 01 TIJ0 の立ち上がりでインプットキャプチャ */
  ATUJ.SUBBLOCK[0].TCRJ.BYTE = 0x01; /* タイマサブブロック 0 を「クロックバス 0」「ノイズ
  キャンセル無し」
  「立ち上がりエッジでキャプチャ」に設定 */

  /* Configure FCRJ0
  b7 FIFOENJ0 = 1 FIFO レジスタ J0 は有効
  b6 Reserved
  b5 FVCRENJ0 = 0 OCRJ0 でのコンペアマッチで FIFO を有効にしない
  b4 FRSTJ0 = 0 何の動作もしない
  b3:2 Reserved
  b1:0 FDFTRGJ0[1:0] = 01 タイマ J ステータスレジスタのフラグセット基準データ数は 6 個 */
  ATUJ.SUBBLOCK[0].FCRJ.BYTE = 0x81; /* FIFO レジスタを常に有効に、FIFO に格納するデー
  タ数を 6 に設定 */

  /* Configure TSRJ0
  b7:5 Reserved
  b4 FVLDFJ0 = 0 FIFO レジスタが有効状態でない
  b3 CMFJ0 = 0 OCRJ0 と TCNTJn のコンペアマッチの検出なし
  b2 OVFJ0 = 0 TCNTJ0 にオーバーフローなし
  b1 FDOVFJ0 = 0 FIFO 内のデータは正常であることを示す

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```

b0      FDFEJ0    = 0      FIFO 内のデータ数が FDFTRGJ の設定値より少ないことを示す */
ATUJ.SUBBLOCK[0].TSRJ.BYTE &= 0; /* 各ステータスフラグをクリア */

/* Configure TIERJ0
b7:3 Reserved
b2      OVEJ0    = 0      OVEJ0 による割り込みを禁止
b1      FDOVEJ0  = 0      FDOVEJ0 による割り込みを禁止
b0      FDFEJ0    = 1      FDFEJ0 による割り込みを許可 */
ATUJ.SUBBLOCK[0].TIERJ.BYTE = 0x01; /* FIFO データフル時の割り込みを許可 */

/* Configure TSTRJ
b7:2 Reserved
b1      STRJ1    = 0      TCNTJn のカウント動作を停止
b0      STRJ0    = 1      TCNTJn のカウント動作を許可 */
ATUJ.TSTRJ.BYTE = 0x01; /* タイマ J サブブロック 0 を動作 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCR0[9:0] = H'9   プリスケアラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケアラ 0 の分周比を 10 に設定 */

DMAC_init(); /* DMAC 設定ルーチンへ */

/* Configure IPR21
b15:12 ATU-J (DFIJ0、DFIJ1) = H'F   対応する割り込みの優先順位を 15 に設定
b11:8  ATU-J (OVIJ0、OVIJ1) = H'0   対応する割り込みの優先順位を 0 に設定
b7:4  ATU-J (DOVIJ0、DOVIJ1) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 Reserved*/
INTC.IPR21.WORD = 0xF000; /* DFIJ0 の割り込み優先順位を 15 に設定 */

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9      TJE = 1      タイマ J のカウント動作を許可
b8      THE = 0      タイマ H のカウント動作を停止
b7      TGE = 0      タイマ G のカウント動作を停止
b6      TFE = 0      タイマ F のカウント動作を停止
b5      TEE = 0      タイマ E のカウント動作を停止
b4      TDE = 0      タイマ D のカウント動作を停止
b3      TCE = 0      タイマ C のカウント動作を停止
b2      TBE = 0      タイマ B のカウント動作を停止
b1      TAE = 0      タイマ A のカウント動作を停止
b0      PSCE = 1     プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0201; /* タイマ J およびプリスケアラのカウントをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : DMAC_init
* Description   : DMAC 設定ルーチン (FIFO レジスタに格納したデータの転送)
* Arguments    : none
* Return Value  : none
*****/

```

```

void DMAC_init(void)
{
  /* Configure CHCR1
b31:30 TC[1:0] = 10   1 回の転送要求で DMATCR の設定回数転送
b29   Reserved
b28   RLD1 = 1       リロード機能 1 は有効 (ON)
b27:26 RLD2[1:0] = 00   リロード機能 2 は無効 (OFF)
b25   Reserved
b24   IFT = 0        Integer→Floating-point 変換機能は無効 (OFF)
b23:19 Reserved
b18   HIE = 0        DMATCR = (転送前にセットした DMATCR) /2 で、割り込み要求を禁止
b17:16 Reserved
b15:14 DM[1:0] = 01   デスティネーションアドレスは増加
b13:12 SM[1:0] = 00   ソースアドレスは固定 (16 バイト単位転送時は設定禁止)
b11:8  RS[3:0] = 1000   DMA 拡張リソースセクタ
b7:6   Reserved
b5     TB = 1         バーストモード
b4:3 TS[1:0] = 01   ワード (2 バイト) 単位
b2     IE = 1        割り込み要求を許可
b1     Reserved
b0     DE = 0        DMA 転送を禁止 */
DMAC1.CHCR.LONG = 0x9000482C;   /* 転送サイズ:ワード リソース:拡張リソースセクタ

                                MATCR をリロード DMAC 転送終了割り込みを許可
                                バースト転送 転送先のアドレスを増加 */

  /* Configure DMARS0
b15:10 CH1 MID[5:0] = 111000DMA0 の転送要求元は ATU-III_J0
b9:8  CH1 RID[1:0] = 11   同上
b7:2  CH0 MID[5:0] = 000000default
b1:0  CH0 RID[1:0] = 00   同上 */
DMACCOM.DMARSx[0].DMARS.WORD = 0xE300; /* CH1 の拡張リソースを ATU-III J0 に設定
*/

  /* Configure SAR0
b31:0  DMA ソースアドレス */
DMAC1.SAR = 0xFFFFFFFF9C;   /* 転送元のアドレスを指定 (FIFO レジスタ) */

  /* Configure DAR
b31:0  DMA デスティネーションアドレス */
DMAC1.DAR = 0xFFFF80000;   /* 転送先のアドレスを指定 */

  /* Configure DMATCR
b31:24 Reserved
b23:0  DMA 転送回数 */
DMAC1.DMATCR = 0x0006;   /* DMA 転送の回数を設定 */

  /* Configure RSAR
b31:0  DMA リロード時のソースアドレス */
DMAC1.RSAR = 0xFFFFFFFF9C;   /* 各アドレスの値のリロード値を設定 */

  /* Configure DRAR
b31:0  DMA リロード時のデスティネーションアドレス */
DMAC1.RDAR = 0xFFFF80000;

```

```

/* Configure RDMATCR
b31:24 Reserved
b23:0 DMA リロード時の転送回数 */
DMAC1.RDMATCR = 0x0006;

/* Configure IPR03
b15:12 DMAC0 = H'0 対応する割り込みの優先順位を 0 に設定
b11:8 DMAC1 = H'F 対応する割り込みの優先順位を 15 に設定
b7:4 DMAC2 = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 DMAC3 = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR03.WORD = 0x0F00; /* DMAC1 転送終了割り込みの優先順位を 15 に設定 */

/* Configure CHFR1
b7:5 Reserved
b4 HE = 0 DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0 TE = 0 DMA トランスファエンドフラグをクリア*/
DMAC1.CHFR.BYTE ^= 0x11; /* DMAC の転送終了フラグをクリア */

/* Configure CHCR
b0 DE = 1 DMA 転送を許可 */
DMAC1.CHCR.BIT.DE = 1; /* DMA 転送を許可 */

/* Configure DMAOR
b0 DME = 1 全てのチャンネルの DMA 転送を許可 */
DMACCOM.DMAOR.BIT.DME = 1; /* DMAC の動作を許可 */
} /* End of function DMAC_init() */

/*****
* Function Name : DEI1
* Description : DMAC 転送終了割り込み
* Arguments : none
* Return Value : none
*****/
void DEI1(void)
{
/* Configure TSRJ0
b7:5 Reserved
b4 FVLDFJ0 = 0 FIFO レジスタが有効状態でない
b3 CMFJ0 = 0 OCRJ0 と TCNTJ0 のコンペアマッチの検出なし
b2 OVJ0 = 0 TCNTJ0 にオーバーフローなし
b1 FDOVFJ0 = 0 FIFO 内のデータは正常であることを示す
b0 FDFJ0 = 0 FIFO 内のデータ数が FDFTRGJ の設定値より少ないことを示す */
ATUJ.SUBBLOCK[0].TSRJ.BYTE &= 0x00; /* 各ステータスフラグをクリア */

/* Configure CHFR1
b7:5 Reserved
b4 HE = 0 DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0 TE = 0 DMA トランスファエンドフラグをクリア*/
DMAC1.CHFR.BYTE ^= 0x11; /* DMAC の転送終了フラグをクリア */
} /* End of function DEI1() */

```

2.25 動作例 24 FIFO レジスタへの値の格納[タイマ J]

2.25.1 概要

- 1) 一定期間 (TCNTJ0 と OCRJ0 のコンペアマッチ発生) 毎に外部信号の受信を有効にします。
- 2) 外部信号エッジを受信する度にタイマカウンタの値を FIFO レジスタに保存します。
- 3) FIFO レジスタに格納したデータの個数が設定値以上になると、FIFO データフル割り込みで DMAC を起動し、格納された値を別の場所に保存し、FIFO レジスタをクリアします。また、外部信号の受信を無効に設定し、次のコンペアマッチまで待機します。

2.25.2 使用機能説明

表 2.26に関連レジスタの機能割り付けを示します。

表 2.26 ATU-III機能割り付け

関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
タイマ J レジスタ	TCRJ0	タイマ J の各機能を設定します。
	FCRJ0	FIFO レジスタの動作を設定します。
	TSTRJ	タイマ J 各サブブロックの動作を許可します。
	TIERJ0	データフル時の割り込みを許可します。
	TCNTJ0	選択したクロックバスでアップカウント動作を行います。
	OCRJ0	TCNTJ0 に対するコンペアマッチの値を設定します。
	FIFOJ0	コンペアマッチ発生時に TCNTJ0 の値を保存します。
	FDNRJ0	FIFOJ0 に保存されている数の個数が読み出されます。
	TSRJ0	タイマ J のステータスフラグが格納されます。
DMAC レジスタ	CHCR1	DMAC の各機能の設定をします。
	DMAOR	転送時の各チャネルの動作を設定します。
	DMARS0	DMAC の拡張リソースを設定します。
	SAR1	DMAC 転送の転送先のアドレスを設定します。
	RSAR1	リロード時の DMAC 転送先アドレスを設定します。
	DAR1	DMAC 転送の転送元のアドレスを設定します。
	RDAR1	リロード時の DMAC 転送元アドレスを設定します。
	DMATCR1	DMAC 転送の転送回数を設定します。
	RDMATCR1	リロード時の転送回数を設定します。
CHFR1	DMAC の各フラグがセットされます。	
PFC レジスタ	PDCR2	ポート D の端子機能を設定します。
INTC レジスタ	IPR03	DMAC の転送終了割り込みの優先度を設定します。
	IPR21	タイマ J データフル割り込みの優先度を設定します。

2.25.3 動作説明

図 2.40に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によってコンペアマッチ回数の計測動作を発生させます。（図は FCRJ0 で設定した基準値が 4 の場合）

DMAC の転送元アドレスを FIFO レジスタに設定することで、FIFO に格納されたカウンタの値を取得します。

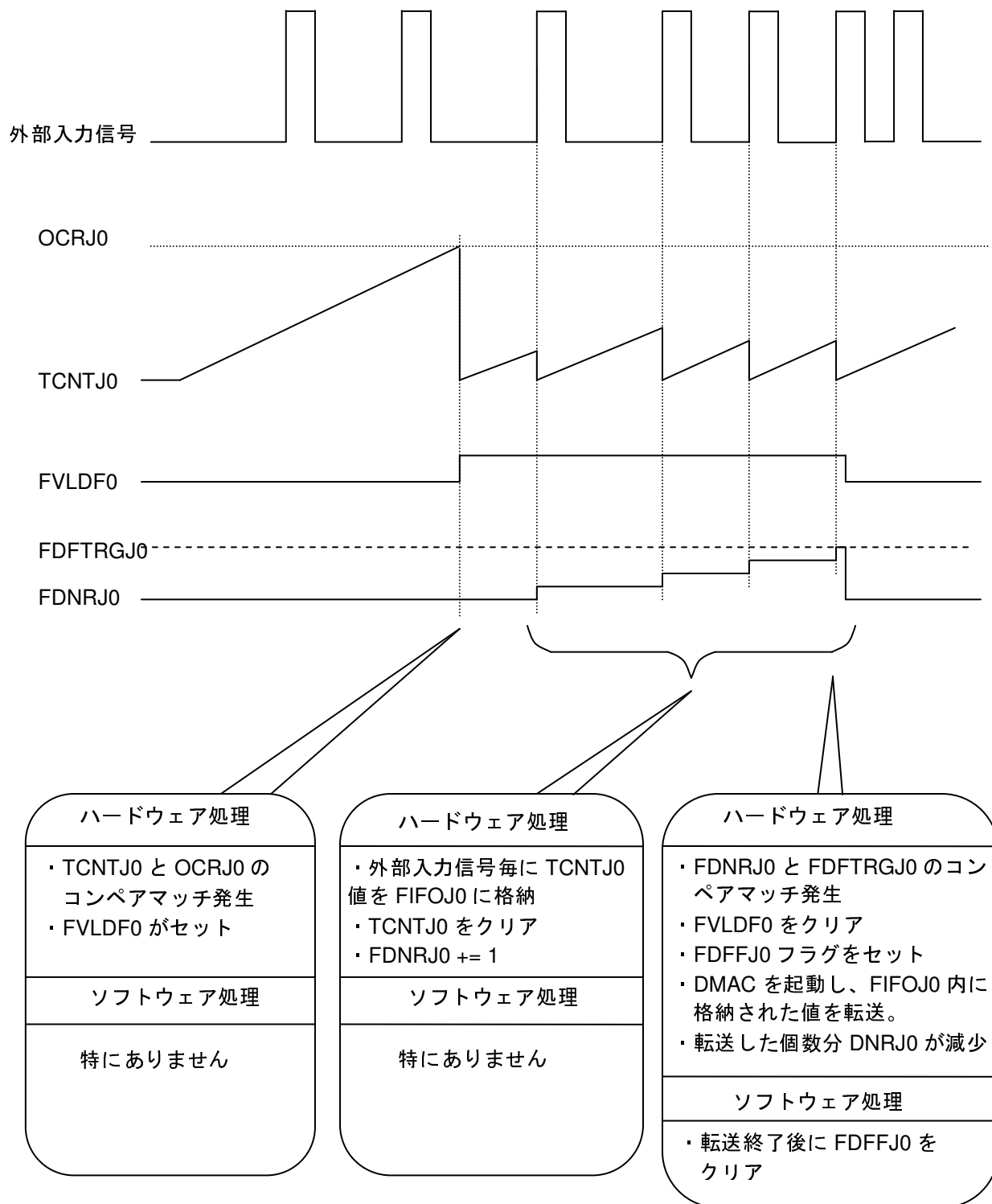


図 2.40 パルスの High 幅計測動作原理

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

2.25.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
DMAC 設定ルーチン	DMAC_init	TCNTJ0 と OCRJ0 のコンペアマッチで割り込み要求を発生し、TCNT2H の値を変数に保存します。
DMAC 転送終了割り込み	DEI1	DMAC の転送が終了すると割り込み要求を発生し、各フラグのクリア、DMAC の送信先アドレスの増加を行います。

- 使用変数の説明

このタスクでは変数を使用していません。

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PDCR2	PD12 を TIJ0 入力に設定します。	0x0300	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIチャンネル J 及びプリスケアラのカウンタ動作を許可します。	0x0201	
TCRJ0	「クロックバス 0 でカウント」「ノイズキャンセルを行わない」「対立ち上がりエッジ」に設定します。	0x01	
FCRJ0	「FIFO レジスタをコンペアマッチ後のみ有効」「FIFO に格納する値の個数は 6 個」に設定します。	0x21	
TSRJ0	各ステータスフラグがセットされます。	—	
TCNTJ0	TCRJ で設定したクロックバスでアップカウントを行います。	—	
OCRJ0	TCNTJ0 と対応するコンペアマッチの値を設定します。	0x00C8	
TIERJ0	FIFO データフル時の割り込みを許可します。	0x01	
TSTRJ	タイマ J サブブロック[0]を動作させる。	0x01	
CHCR1	DMAC の各設定を「転送サイズは word 単位」「転送開始リソースは拡張リソースセクタ」「転送方式はバースト転送」「MATCR をリロード」「転送先のアドレスを転送毎に増加」「転送終了割り込みを許可」「このチャンネルの動作を許可」に設定します。	0x9000482D	DMAC 設定ルーチン
DMARS0	DMAC チャンネル 1 の拡張リソースを「ATU-III J0」に設定します。	0xE300	
SAR1	DMA 転送元アドレスを設定します。	0xFFFFFFFF9C	
DAR1	DMA 転送先アドレスを設定します。	0xFFF80000	
DMATCR1	DMA 転送の回数を設定します。	0x0006	
RSAR1	DMA 転送元アドレスのリロード値を設定します。	0xFFFFFFFF9C	
RDAR1	DMA 転送先アドレスのリロード値を設定します。	0xFFF80000	
RDMATCR1	DMA 転送の回数のリロード値を設定します	0x006	
CHFR1	DMA 転送に関する各フラグがセットされます。	—	
DMAOR	DMAC 全体の動作の許可を設定します	0x0001	
TSRJ0	J のステータスフラグをクリアします。	0x00	DMAC 転送終了割り込み
CHFR1	DMAC の転送終了フラグをクリアします。	HE = 0 TE = 0	

2.25.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform   : SH7254R
* Description    : This is the main tutorial code.
* Operation     : Store a value into FIFO register.
*****/
/*****
* History : DD.MM.YYYY Version Description
*         : 12.01.2012 1.00   First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"       /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
void main(void);           /* メインルーチン */
void DMAC_init(void);     /* DMA 設定ルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value : none
*****/

```

```

void main(void)
{
  /* ポートの設定 */
  /* Configure PDCR2
  b15:12 Reserved
  b11:10 PD13MD[1:0] = 00 PD13 入出力 (ポート)
  b9:8 PD12MD[1:0] = 11 TIJ0 入力 (ATU-III)
  b7:6 PD11MD[1:0] = 00 PD11 入出力 (ポート)
  b5:4 PD10MD[1:0] = 00 PD10 入出力 (ポート)
  b3:2 PD9MD[1:0] = 00 PD9 入出力 (ポート)
  b1:0 PD8MD[1:0] = 00 PD8 入出力 (ポート) */
  PORTD.CR2.WORD = 0x0300; /* PD12 を TIJ0 入力に設定 */

  /* ATU-III の設定 */
  /* Configure TCRJ0
  b7 Reserved
  b6:4 CKSELJ0[2:0] = 000 クロックバス 0
  b3 Reserved
  b2 NCEJ0 = 0 TIJ0 入力のノイズキャンセラ機能を無効に
  b1:0 IOJ0[1:0] = 01 TIJ0 の立ち上がりでインプットキャプチャ */
  ATUJ.SUBBLOCK[0].TCRJ.BYTE = 0x01; /* タイマサブブロック 0 を「クロックバス 0」
  「ノイズキャンセル無し」「立ち上がりエッジでキャプチャ」に設定 */

  /* Configure FCRJ0
  b7 FIFOENJ0 = 0 FIFO レジスタ J0 は無効
  b6 Reserved
  b5 FVCRENJ0 = 1 OCRJ0 のコンペアマッチ後、FIFO フルまで FIFO レジスタを有効に
  b4 FRSTJ0 = 0 何の動作もしない
  b3:2 Reserved
  b1:0 FDFTRGJ0[1:0] = 01 FDFFFJ0 フラグをセットする基準データ数を 6 個に設定 */
  ATUJ.SUBBLOCK[0].FCRJ.BYTE = 0x21; /* FIFO レジスタをコンペアマッチ発生後から FIFO フル
  までの間のみ有効に設定
  FIFO に格納するデータ数を 6 に設定 */

  /* Configure OCRJ0
  b15:0 OCRJ0[15:0] TCNTJ0 に対するアウトプットコンペア */
  ATUJ.SUBBLOCK[0].OCRJ = 0x00C8; /* コンペアマッチの値を設定 */

  /* Configure TSRJ0
  b7:5 Reserved
  b4 FVLDFJ0 = 0 FIFO レジスタが有効状態でない
  b3 CMEJ0 = 0 OCRJ0 と TCNTJ0 のコンペアマッチの検出なし
  b2 OVVFJ0 = 0 TCNTJ0 にオーバフローなし
  b1 FDOVFJ0 = 0 FIFO 内のデータは正常であることを示す
  b0 FDFFFJ0 = 0 FIFO 内のデータ数が FDFTRGJ の設定値より少ないことを示す */
  ATUJ.SUBBLOCK[0].TSRJ.BYTE &= 0; /* 各ステータスフラグをクリア */

  /* Configure TIERJ0
  b7:3 Reserved
  b2 OVEJ0 = 0 OVEJ0 による割り込みを禁止
  b1 FDOVEJ0 = 0 FDOVEJ0 による割り込みを禁止
  b0 FDFEJ0 = 1 FDFEJ0 による割り込みを許可 */
  ATUJ.SUBBLOCK[0].TIERJ.BYTE = 0x01; /* FIFO データフル時の割り込みを許可
  */
}

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```

/* Configure TSTRJ
b7:2   Reserved
b1     STRJ1 = 0      TCNTJ1 のカウント動作を停止
b0     STRJ0 = 1      TCNTJ0 のカウント動作を許可 */
ATUJ.TSTRJ.BYTE = 0x01;          /* タイマ J サブブロック 0 を動作          */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSC0[9:0] ' H'9   プリスケーラの分周比を 10 (設定値は-1) に設定  */
ATUCTRL.PSCR0.WORD = 0x0009;    /* プリスケーラの分週比を 10 に設定          */

DMAC_init();                    /* DMAC 設定ルーチンへ          */

/* Configure IPR21
b15:12 ATU-J (DFIJ0、DFIJ1) ' H'F   対応する割り込みの優先順位を 15 に設定
b11:8  ATU-J (OVIJ0、OVIJ1) ' H'0   対応する割り込みの優先順位を 0 に設定
b7:4 ATU-J (DOVIJ0、DOVIJ1) ' H'0  対応する割り込みの優先順位を 0 に設定
b3:0 Reserved */
INTC.IPR21.WORD = 0xF000;       /* DFIJ0 の割り込み優先順位を 15 に設定    */

set_imask(0);                  /* 割り込みマスクレベルを 0 に設定        */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 1      タイマ J のカウント動作を許可
b8     THE = 0      タイマ H のカウント動作を停止
b7     TGE = 0      タイマ G のカウント動作を停止
b6     TFE = 0      タイマ F のカウント動作を停止
b5     TEE = 0      タイマ E のカウント動作を停止
b4     TDE = 0      タイマ D のカウント動作を停止
b3     TCE = 0      タイマ C のカウント動作を停止
b2     TBE = 0      タイマ B のカウント動作を停止
b1     TAE = 0      タイマ A のカウント動作を停止
b0     PSCE = 1     プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0201;   /* タイマ J およびプリスケーラのカウントをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : DMAC_init
* Description   : DMAC 設定ルーチン (FIFO レジスタに格納したデータ) 転送)
* Arguments    : none
* Return Value : none
*****/
void DMAC_init(void)
{
    /* Configure CHCR1
    b31:30 TC[1:0] = 10   1 回の転送要求で DMATCR の設定回数転送
    b29     Reserved
    b28     RLD1 = 1      リロード機能 1 は有効 (ON)
    b27:26 RLD2[1:0] = 00   リロード機能 2 は無効 (OFF)
    b25     Reserved
    b24     IFT = 0       Integer→Floating-point 変換機能は無効 (OFF)
    b23:19 Reserved

```

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b18      HIE = 0          DMATCR = (転送前にセットした DMATCR) /2 で、割り込み要求を禁
止
b17:16  Reserved
b15:14  DM[1:0] = 01     デスティネーションアドレスは増加
b13:12  SM[1:0] = 00     ソースアドレスは固定 (16 バイト単位転送時は設定禁止)
b11:8   RS[3:0] = 1000   DMA 拡張リソースセクタ
b7:6    Reserved
b5      TB = 1          バーストモード
b4:3    TS[1:0] = 01     ワード (2 バイト) 単位
b2      IE = 1          割り込み要求を許可
b1      Reserved
b0      DE = 0          DMA 転送を禁止 */
DMAC1.CHCR.LONG = 0x9000482C; /* 転送サイズ:ワード リソース:拡張リソースセレ
クタ

                                MATCR をリロード DMAC 転送終了割り込みを許可
                                バースト転送 転送先のアドレスを増加 */

/* Configure DMARS0
b15:10  CH1 MID[5:0] = 111000DMA0 の転送要求元は ATU-III_J0
b9:8    CH1 RID[1:0] = 11     同上
b7:2    CH0 MID[5:0] = 000000default
b1:0    CH0 RID[1:0] = 00     同上 */
DMACCOM.DMARSx[0].DMARS.WORD = 0xE300; /* CH1 の拡張リソースを ATU-III J0 に設定
*/

/* Configure SAR0
b31:0   DMA ソースアドレス */
DMAC1.SAR = 0xFFFFFFFF9C; /* 転送元のアドレスを指定 (FIFO レジスタ) */

/* Configure DAR
b31:0   DMA デスティネーションアドレス */
DMAC1.DAR = 0xFFFF80000; /* 転送先のアドレスを指定 */

/* Configure DMATCR
b31:24  Reserved
b23:0   DMA 転送回数 */
DMAC1.DMATCR = 0x0006; /* DMA 転送の回数を設定 */

/* Configure RSAR
b31:0   DMA リロード時のソースアドレス */
DMAC1.RSAR = 0xFFFFFFFF9C; /* 各アドレスの値のリロード値を設定 */

/* Configure DRAR
b31:0   DMA リロード時のデスティネーションアドレス */
DMAC1.RDAR = 0xFFFF80000;

/* Configure RDMATCR
b31:24  Reserved
b23:0   DMA リロード時の転送回数 */
DMAC1.RDMATCR = 0x0006;

/* Configure IPR03
b15:12  DMAC0 ' H'0     対応する割り込みの優先順位を 0 に設定
b11:8   DMAC1 ' H'F     対応する割り込みの優先順位を 15 に設定
b7:4    DMAC2 ' H'0     対応する割り込みの優先順位を 0 に設定
b3:0    DMAC3 ' H'0     対応する割り込みの優先順位を 0 に設定 */
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
INTC.IPR03.WORD = 0x0F00;          /* DMAC1 転送終了割り込みの優先順位を 15 に設定 */

/* Configure CHFR1
b7:5 Reserved
b4      HE = 0          DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0      TE = 0          DMA トランスファエンドフラグをクリア*/
DMAC1.CHFR.BYTE ^= 0x11;          /* DMAC の転送終了フラグをクリア */

/* Configure CHCR
b0      DE = 1          DMA 転送を許可 */
DMAC1.CHCR.BIT.DE = 1;          /* DMA 転送を許可 */

/* Configure DMAOR
b0      DME = 1          全てのチャンネルの DMA 転送を許可 */
DMACCOM.DMAOR.BIT.DME = 1;      /* DMAC の動作を許可 */
} /* End of function DMAC_init() */

/*****
* Function Name : DEI1
* Description   : DMAC 転送終了割り込み
* Arguments    : none
* Return Value : none
*****/
void DEI1(void)
{
    /* Configure TSRJ0
b7:5 Reserved
b4      FVLDFJ0 = 0      FIFO レジスタが有効状態でない
b3      CMFJ0 = 0        OCRJ0 と TCNTJ0 のコンペアマッチの検出なし
b2      OVFJ0 = 0        TCNTJ0 にオーバフローなし
b1      FDOVFJ0 = 0      FIFO 内のデータは正常であることを示す
b0      FDFJ0 = 0        FIFO 内のデータ数が FDFTRGJ の設定値より少ないことを示す */
    ATUJ.SUBBLOCK[0].TSRJ.BYTE &= 0x10; /* 各ステータスフラグをクリア */

    /* Configure CHFR1
b7:5 Reserved
b4      HE = 0          DMA ハーフエンドフラグをクリア
b3:1 Reserved
b0      TE = 0          DMA トランスファエンドフラグをクリア*/
    DMAC1.CHFR.BYTE ^= 0x11;          /* DMAC の転送終了フラグをクリア */
} /* End of function DEI1() */
```

2.26 動作例 25 倍周クロックの生成[タイマ B]

2.26.1 概要

- 1) 図 2.41に示すように外部信号の周期を計測し、それに対する倍周クロックを生成します。
- 2) 倍周クロックの周期は次の式で求められます。

[倍周クロックの周期(ns)=外部入力信号の周期÷パルスインターバルマルチプライヤレジスタ (PIMR)の設定値]

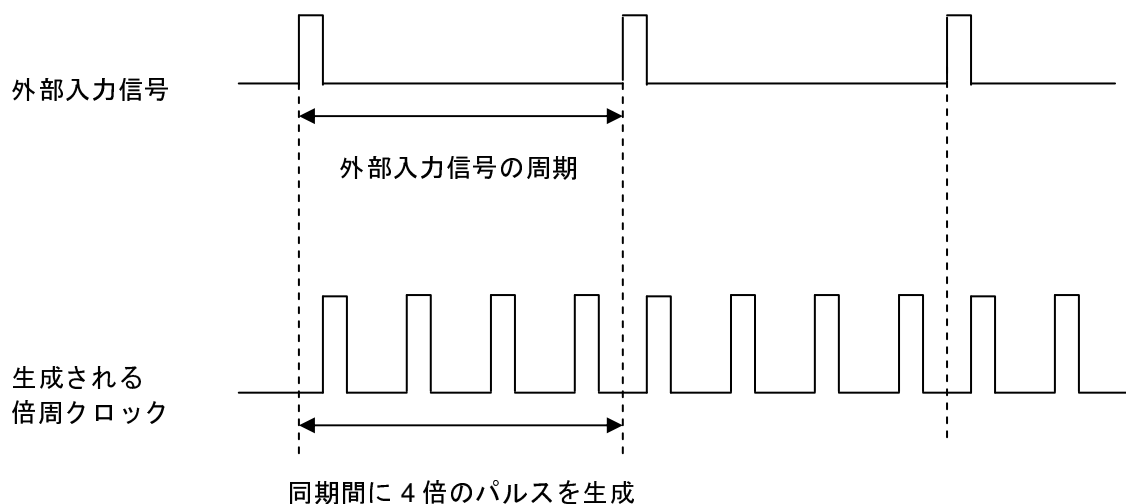


図 2.41 倍周クロックの生成(PIMR=4 の場合)

2.26.2 使用機能説明

表 2.27に使用端子及び関連レジスタの機能割り付けを示します。

表 2.27 ATU-III機能割り付け

使用端子		機能
端子	TIA00	測定するパルスをこの端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
	CBCNT	クロックバス 5 のクロックソースを設定します。
タイマ A レジスタ	TCRA	タイマ A の動作クロック、外部入力エッジのイベント出力を設定します。
	TIOR1A	TIA0 の立ち上がりエッジを検出する。
タイマ B レジスタ	TCRB	TCNTB0、TCNTB2 のカウントソースを設定します。
	TIORB	カウント動作の許可等、倍周クロック生成に関する設定を行います。
	PIMR	外部入力信号に対して生成する、倍周クロックの通倍比を設定します。
	TIERB	インプットキャプチャ/コンペアマッチの割り込み要求の許可/禁止を制御します。
	TCNTB3,6	入力エッジ間計測カウンタです。
	OCRB6	TCNTB6 へのコンペアマッチの値を設定します。

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

タイマ D レジスタ	TCRD	タイマ D の動作を設定します。
	TIOR1D0	タイマ D の入出力を設定します。
	TSRD0	タイマ D のステータスレジスタです。
	OCRD0	TCNT1D0 へのコンパママッチの値を設定します。
	TIERD0	タイマ D 出力反転の設定を行います。
	TCNT1D0	タイマ D のカウンタの値を設定します。
	DCNTD00 ~02	タイマ D のダウンカウンタの値を設定します。
	DSTRD0	ダウンカウンタの開始/停止を設定します。
PFC レジスタ	PECR1	端子機能を設定します。
INTC レジスタ	IPR07	CMIB6 の割り込み優先順位を設定します。
	IPR10	OCRD00 の割り込み優先順位を設定します。

2.26.3 動作説明

図 2.42に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって入力信号に対する倍周クロックの生成を行います。エッジ間計測ブロックの入力エッジ間計測カウンタ B0 (TCNTB0) からのキャプチャ値に対して、リロード付きのダウンカウントを行います。ダウンカウンタのリロード時に倍週クロック用パルスを生成します。

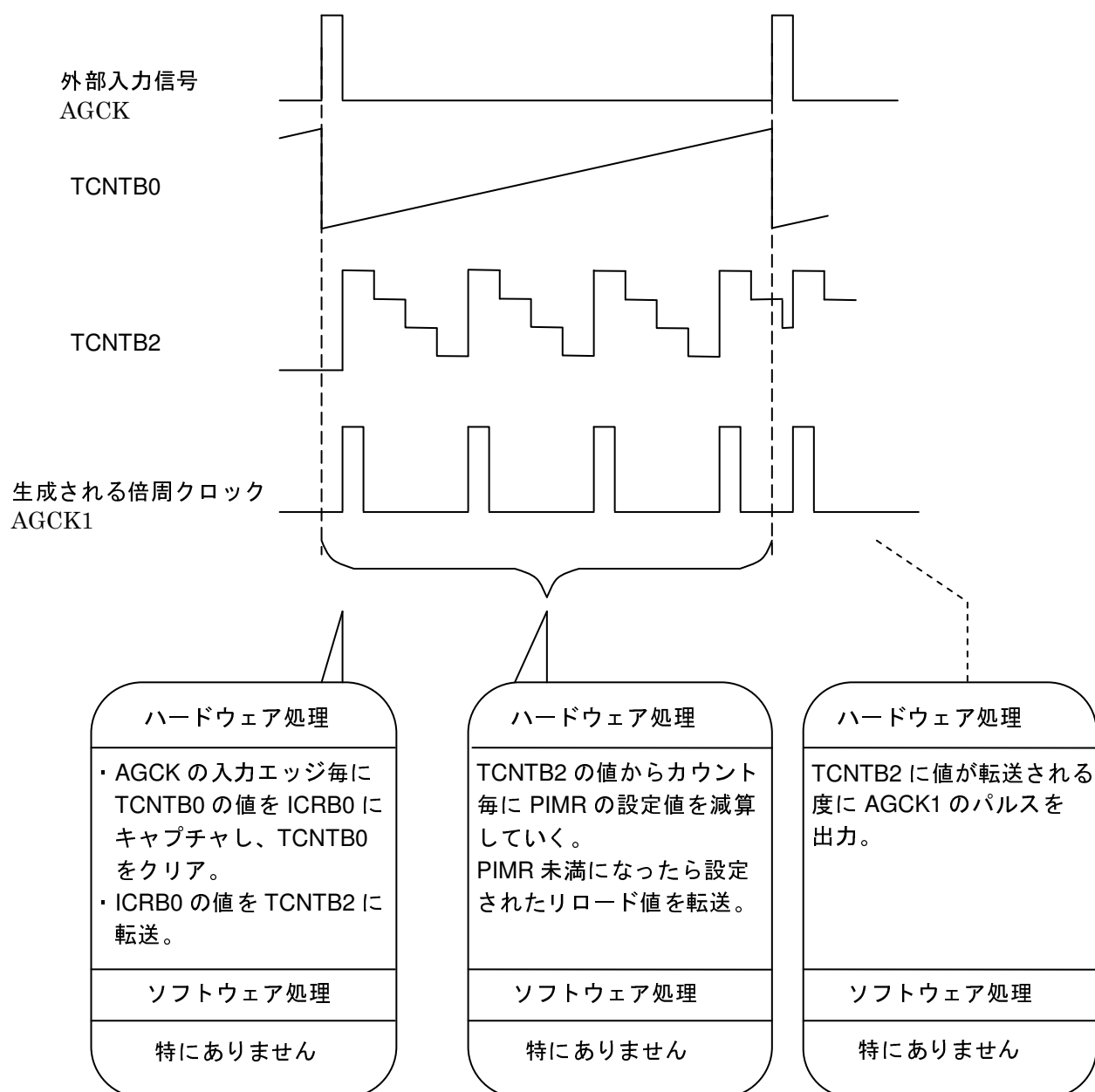


図 2.42 パルス周期計測動作原理(PIMR=4 の場合)

2.26.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。

● 使用変数の説明

ラベル名	機能	データ長	使用モジュール
cnt	ワンショットパルス出カルーチン参照の回数を記録します。	unsigned char	ワンショットパルス出カルーチン

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE0 端子を入力端子 TIA0 に設定します。	0x0001	メインルーチン
PSCR0	プリスケラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIチャンネルA,B及びプリスケラのカウンタ動作を許可します。	0x0007	
TCRA	TIA0 への入力信号をイベントとしてタイマ B に出力します。	0x08	
TIOR1A	TIA0 への入力信号の立ち上がりエッジを検出する。	0x0001	
TIORB	TCNTB2 のリロード値を ICRB0 の値に設定、外部イベントの入力を許可します。	0x21	
PIMR	外部入力周期に対する倍周クロックの通倍比を 100 に設定します。	0x0004	
TCRB	TCNTB0,TCNTB2 の使用クロックをクロックバス 0 に設定します。	0x00	
TIERB	TCNTB6,OCRB6 のコンペアマッチ発生のタイミングでの割り込み要求を許可します。	0x18	
TCNTB6	TCNTB6 の初期値を 0 に設定します。	0x00000000	
OCRB6	アウトプットコンペア値を 4 に設定します。	0x00002000	
TCRD0	TCNT1D をクロックバス 5 でアップカウントし、DCNTD0 をクロックバス 5 でダウンカウント	0x0055	
TIOR1D	OCRD00 のコンペアマッチ A 発生時、TOD00A 端子からトグル出力します。	0x0103	
TSRD	コンペアマッチ、アンダフローフラグをクリアします。	0x0000	
OCRD	OCRD00 に TCNT1D0 に対応するコンペアマッチ A の条件を 100 に設定します。	0x00006400	
TIERD	OCRD00 のコンペアマッチによる割り込みを許可します。	0x0010	
IPR10	CMID00-CMID03 の割り込み優先順位を 15 に設定します。		
IPR07	CMIB6 のコンペアマッチ割り込み優先順位を 15 に設定します。	0x0F00	
ATUENR	タイマ A,B,D およびプリスケラのカウンタをスタートします。	0x0017	

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

TCNTB3	入力エッジ間計測カウント値を 0 に設定します。	0x00000000	ワンショットパルス出カルーチン
TCNT1D0	タイマカウンタ値を 0 に設定します。	0x00000000	
DCNTD00~ DCNTD02	ダウンカウンタ値を設定します。	0x00FA<<8	
DSTRD0	ダウンカウンタ 00 を開始します。 ダウンカウンタ 01 を開始します。 ダウンカウンタ 02 を開始します。	0x01 0x02 0x04	

2.26.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform  : SH7254R
* Description    : This is the main tutorial code.
* Operation     : Generate multiple clock.
*****/
/*****
* History : DD.MM.YYYY Version Description
*          : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Private global variables and functions
*****/
unsigned char cnt;
void main(void);            /* メインルーチン */

/*****
* Function Name : main
* Description   : The main loop
* Arguments    : none
* Return Value : none
*****/

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PECR1
b15:14 PE7MD = 0   PE7 入出力 (ポート)
b13:12 PE6MD = 0   PE6 入出力 (ポート)
b11     Reserved
b10     PE5MD = 0   PE5 入出力 (ポート)
b9      Reserved
b8      PE4MD = 0   PE4 入出力 (ポート)
b7      Reserved
b6      PE3MD = 0   PE3 入出力 (ポート)
b5:4 PE2MD = 0   PE2 入出力 (ポート)
b3:2 PE1MD = 0   PE1 入出力 (ポート)
b1      Reserved
b0      PE0MD = 1   TIA00 入力 (ATU-III) */
PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定
*/

  /* ATU-III (A) の設定 */
  /* Configure TCRA
b7      EVOSEL2A = 0 イベント出力 2A セレクト
b6      EVOSEL2B = 0 イベント出力 2B セレクト
b5:3 EVOSEL1 = 001 タイマ B へ TIA00 の入力エッジを出力
b2:0 CKSELA = 000 カウントクロックにクロックバス 0 (プリスケアラ 0) を選択 */
ATUA.TCRA.BYTE = 0x08; /* タイマ A をクロックバス 0 のカウントクロックで動作
                        TIA00 の入力をイベントとしてタイマ B に出力 */

  /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00   TIA のインプットキャプチャ禁止
b9:8 IOA4 = 00   TIA のインプットキャプチャ禁止
b7:6 IOA3 = 00   TIA のインプットキャプチャ禁止
b5:4 IOA2 = 00   TIA のインプットキャプチャ禁止
b3:2 IOA1 = 00   TIA のインプットキャプチャ禁止
b1:0 IOA0 = 01   TIA の立ち上がりで ICRA にキャプチャ */
ATUA.TIOR1A.WORD = 0x0001; /* TIA00 入力の立ち上がりエッジで ICRA にインプットキャ
                           プチャ */

  /* ATU-III (B) の設定 */
  /* Configure TIORB
b7      LDSEL = 0   TCNTB2 にロードする値に ICRB0 を用いる
b6      CTCNTB5 = 0   TCNTB5 のカウントを許可
b5      EVCNTB = 1   外部イベントの入力を許可
b4      LDEN = 0    リロードレジスタの更新を許可
b3      CCS = 0     TCNTB=TCNTB4 のとき、TCNTB4 のカウント動作を停止しない
b2:1 Reserved
b0      IOB6 = 1    TCNTB6 と OCRB6 とのコンペアマッチを許可 */
ATUB.TIORB.BYTE = 0x21;

  /* Configure PIMR
b15:12 Reserved
b11:0 PIM = H'0004 クロックの通倍比を 4 倍に設定 */
ATUB.PIMR.WORD = 0x0004; /* 倍周クロックの通倍比を 4 倍に設定 */

  /* Configure TCRB
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b7:2 Reserved
b1:0 CKSELB = 00      カウントソースにクロックバス 0 を選択 */
  ATUB.TCRB.BYTE = 0x00; /* TCNTB0,TCNTB2 をクロックバス 0 のカウントクロックで動作 */

/* Configure TIERB
b7:6 Reserved
b5:4 IREG = 01      CMFB6 有効後、次の外部イベント入力タイミングで割り込み要求を出力
b3      CMEB6 = 1      コンペアマッチ割り込み B6 要求の出力を許可
b2      CMEB1 = 0      コンペアマッチ割り込み B1 要求の出力を禁止
b1      ICEB0 = 0      インプットキャプチャ割り込み B0 要求の出力を禁止
b0      CMEB0 = 0      コンペアマッチ割り込み B0 要求の出力を禁止 */
ATUB.TIERB.BYTE = 0x18; /* TCNTB6,OCRB6 のコンペアマッチ発生タイミングでの割り込み要求を許可 */

/* Configure TCNTB
b31:0  CNTB0 = 0x00000000  入力エッジ間計測カウント動作数を 0 クリア */
ATUB.TCNTB6.LONG = 0x00000000; /* TCNTB6 の初期値を 0 に設定 */
*/

/* Configure OCRB6
b31:12 OCB6 = H'00002      アウトプットコンペア値を、2 に設定
b11:0  Reserved */
  ATUB.OCRB6.LONG = 0x00002000; /* TCNTB6 に対応するコンペアマッチの値を 4 に設定 */
*/

/* タイマ D に関する設定 */
/* Configure TCRD
b15      Reserved
b14      OBREDn = 0          OSBRDn のインプットキャプチャ動作を禁止
b13      C2CEDn = 0          タイマ B からの TCNT2Dn カウンタ値クリア要求を禁止
b12      C1CEDn = 0          タイマ B からの TCNT1Dn カウンタ値クリア要求を禁止
b11      Reserved
b10:8    CKSEL2Dn = 000      クロックバス 0 で TCNT2Dn をアップカウント
b7      Reserved
b6:4    CKSEL1Dn = 101      クロックバス 5 で TCNT1Dn をアップカウント
b3      Reserved
b2:0    DCSELDn = 101      クロックバス 5 で DCNTDnm をダウンカウント */
ATUD.SUBBLOCKA[0].TCRD.WORD = 0x0055; /* TCNT1Dn をクロックバス 5 でアップカウント
                                         DCNTD0n をクロックバス 5 でダウンカウント */

/* Configure TIOR1D
b15:14  OSSDn3 = 00      TODn3A 端子出力なし
b13:12  OSSDn2 = 00      TODn2A 端子出力なし
b11:10  OSSDn1 = 00      TODn1A 端子出力なし
b9:8    OSSDn0 = 01      TODn0A 端子出力なし
b7:6    IOADn3 = 00      OCRDn3 コンペアマッチ禁止
b5:4    IOADn2 = 00      OCRDn2 コンペアマッチ禁止
b3:2    IOADn1 = 00      OCRDn1 コンペアマッチ禁止
b1:0    IOADn0 = 11      OCRDn0 コンペアマッチ時、トグル出力*/
ATUD.SUBBLOCKB[0].TIOR1D.WORD = 0x0103; /* OCRD00 のコンペアマッチ A 発生時、TOD00A
端子からトグル出力 */

/* Configure TSRD
b15:14  Reserved
b13      OVF2Dn = 0      オーバフローフラグをクリア
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```

b12  OVF1Dn = 0   オーバフローフラグをクリア
b11  UDFDn3 = 0   アンダーフローフラグをクリア
b10  UDFDn2 = 0   アンダーフローフラグをクリア
b9   UDFDn1 = 0   アンダーフローフラグをクリア
b8   UDFDn0 = 0   アンダーフローフラグをクリア
b7   CMFADn3 = 0  コンペアマッチフラグをクリア
b6   CMFADn2 = 0  コンペアマッチフラグをクリア
b5   CMFADn1 = 0  コンペアマッチフラグをクリア
b4   CMFADn0 = 0  コンペアマッチフラグをクリア
b3   CMFBDn3 = 0  コンペアマッチフラグをクリア
b2   CMFBDn2 = 0  コンペアマッチフラグをクリア
b1   CMFBDn1 = 0  コンペアマッチフラグをクリア
b0   CMFBDn0 = 0  コンペアマッチフラグをクリア */
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダフローフラグをク
リ
ア
*/

/* Configure OCRD
b31:8   アウトプットコンペアレジスタを H'64(D'100) に設定
b7:0   Reserved */
ATUD.SUBBLOCKB[0].OCRD[0] = 0x00006400; /* OCRD00 に TCNT1D0 に対応するコンペアマッ
チ
A
の条件を 100 に設定 */

/* Configure TIERD
b15:14 Reserved
b13   OVE2Dn = 0   OVF2Dn による割り込み要求を禁止
b12   OVE1Dn = 0   OVF1Dn による割り込み要求を禁止
b11   UDEDn3 = 0   UDFDn3 による割り込み要求を禁止
b10   UDEDn2 = 0   UDFDn2 による割り込み要求を禁止
b9    UDEDn1 = 0   UDFDn1 による割り込み要求を禁止
b8    UDEDn0 = 0   UDFDn0 による割り込み要求を禁止
b7    CMEADn3 = 0  CMFADn3 による割り込み要求を禁止
b6    CMEADn2 = 0  CMFADn2 による割り込み要求を禁止
b5    CMEADn1 = 0  CMFADn1 による割り込み要求を禁止
b4    CMEADn0 = 1  CMFADn0 による割り込み要求を許可
b3    CMEBDn3 = 0  CMFBDn3 による割り込み要求を禁止
b2    CMEBDn2 = 0  CMFBDn2 による割り込み要求を禁止
b1    CMEBDn1 = 0  CMFBDn1 による割り込み要求を禁止
b0    CMEBDn0 = 0  CMFBDn0 による割り込み要求を禁止 */
ATUD.SUBBLOCKB[0].TIERD.WORD = 0x0010; /* OCRD00 のコンペアマッチによる割り込みを
許可
*/

/* Configure IPR10
b15:12 ATU-D0 (CMID00-CMID03) = H'F   対応する割り込みの優先順位を 15 に設定
b11:8  ATU-D0 (OVI1D0-OVI2D0) = H'0   対応する割り込みの優先順位を 0 に設定
b7:4   ATU-D0 (UDID00-UDID03) = H'0   対応する割り込みの優先順位を 0 に設定
b3:0   ATU-D1 (CMID10-CMID13) = H'0   対応する割り込みの優先順位を 0 に設定 */
INTC.IPR10.WORD = 0xF000; /* OCRD00 の割り込み優先順位を 15 に設定
*/

/* ATU-III (共通) の設定 */
/* Configure PSCR0
b15:10 Reserved
b9:0   PSCn = H'9   プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラ 0 の分周比を 10 に設定
*/

```

```

/* Configure IPR07
b15:12 ATU-B(CMIB0,CMIB1) = H'0   対応する割り込みの優先順位を 0 に設定
b11:8  ATU-B(CMIB6,ICIB0) = H'F   対応する割り込みの優先順位を 15 に設定
b7:4  ATU-C0(IMIC0-IMIC3) = H'0   対応する割り込みの優先順位を 0 に設定
b3:0  ATU-C0(OVIC0) = H'0         対応する割り込みの優先順位を 0 に設定 */
INTC.IPR07.WORD = 0x0F00; /* TCNTB6 のコンペアマッチの割り込み優先順位を 15 に設定
*/

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0      タイマ J のカウント動作を停止
b8     THE = 0      タイマ H のカウント動作を停止
b7     TGE = 0      タイマ G のカウント動作を停止
b6     TFE = 0      タイマ F のカウント動作を停止
b5     TEE = 0      タイマ E のカウント動作を停止
b4     TDE = 1      タイマ D のカウント動作を許可
b3     TCE = 0      タイマ C のカウント動作を停止
b2     TBE = 1      タイマ B のカウント動作を許可
b1     TAE = 1      タイマ A のカウント動作を許可
b0     PSCE = 1     プリスケアラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0017; /* タイマ A,B,D およびプリスケアラのカウントをスタート
*/
while(1);
} /* End of function main() */

/*****
* Function Name : CMIB6
* Description   : コンペアマッチ割り込み
* Arguments    : none
* Return Value  : none
*****/
void CMIB6(void)
{
/* Configure TSRB
b7:4 Reserved
b3     CMFB6 = 0     コンペアマッチフラグをクリア
b2     CMFB1 &= 1    コンペアマッチフラグを維持
b1     ICFB0 &= 1    コンペアマッチフラグを維持
b0     CMFB0 &= 1    コンペアマッチフラグを維持 */
ATUB.TSRB.BYTE &= 0xF7; /* コンペアマッチフラグをクリア */
} /* End of function CMIB6() */

/*****
* Function Name : CMID00
* Description   : ワンショットパルス出力ルーチン
* Arguments    : none
* Return Value  : none
*****/
void CMID00(void)
{
/* Configure TSRD
b15:14 Reserved
b13     OVF2Dn = 0   オーバフローフラグをクリア

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b12  OVF1Dn = 0   オーバフローフラグをクリア
b11  UDFDn3 = 0   アンダーフローフラグをクリア
b10  UDFDn2 = 0   アンダーフローフラグをクリア
b9   UDFDn1 = 0   アンダーフローフラグをクリア
b8   UDFDn0 = 0   アンダーフローフラグをクリア
b7   CMFADn3 = 0  コンペアマッチフラグをクリア
b6   CMFADn2 = 0  コンペアマッチフラグをクリア
b5   CMFADn1 = 0  コンペアマッチフラグをクリア
b4   CMFADn0 = 0  コンペアマッチフラグをクリア
b3   CMFBdn3 = 0  コンペアマッチフラグをクリア
b2   CMFBdn2 = 0  コンペアマッチフラグをクリア
b1   CMFBdn1 = 0  コンペアマッチフラグをクリア
b0   CMFBdn0 = 0  コンペアマッチフラグをクリア */
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダフローフラグを
クリア */

/* Configure TCNTB
b31:0  入力エッジ間計測カウント値 = 0 */
ATUB.TCNTB3.LONG =0x00000000;

/* Configure TCNA1D
b31:8  タイマカウンタ値 = 0
b7:0 Reserved */
ATUD.SUBBLOCKA[0].TCNT1D = 0x00000000;

cnt ++;
if(cnt == 1)
{
  /* Configure DCNTD00
b31:8  タイマダウンカウンタ値
b7:0  Reserved */
  ATUD.SUBBLOCKB[0].DCNTD[0] = 0x00FA << 8; /* ダウンカウンタ 00 を 250 にセット
  */

  /* Configure DSTRD0
b7:4  Reserved
b3   DSTD03=0   DCNTD03 ダウンカウント停止
b2   DSTD02=0   DCNTD02 ダウンカウント停止
b1   DSTD01=0   DCNTD01 ダウンカウント停止
b0   DSTD00=1   DCNTD00 ダウンカウント開始 */
  ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x01; /* ダウンカウンタ 00 のダウンカウントをスタート
  */
}
else if(cnt == 4)
{
  /* Configure DCNTD01
b31:8  タイマダウンカウンタ値
b7:0  Reserved */
  ATUD.SUBBLOCKB[0].DCNTD[1] = 0x00FA<<8; /* ダウンカウンタ 01 を 250 にセット
  */

  /* Configure DSTRD0
b7:4  Reserved
b3   DSTD03=0   DCNTD03 ダウンカウント停止
b2   DSTD02=0   DCNTD02 ダウンカウント停止
b1   DSTD01=0   DCNTD01 ダウンカウント開始
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b0    DSTD00=1    DCNTD00 ダウンカウント停止 */
ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x02; /* ダウンカウンタ 01 のダウンカウントをスタート
*/
}
else if(cnt == 8)
{
    /* Configure DCNTD02
b31:8 タイマダウンカウンタ値
b7:0  Reserved */
    ATUD.SUBBLOCKB[0].DCNTD[2] = 0x00FA << 8; /* ダウンカウンタ 02 を 250 にセット
        */

    /* Configure DSTRD0
b7:4  Reserved
b3    DSTD03=0    DCNTD03 ダウンカウント停止
b2    DSTD02=0    DCNTD02 ダウンカウント開始
b1    DSTD01=0    DCNTD01 ダウンカウント停止
b0    DSTD00=1    DCNTD00 ダウンカウント停止 */
    ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x04; /* ダウンカウンタ 02 のダウンカウントをスタート
*/
}
else if(cnt == 10)
{
    cnt = 0;
}
} /* End of function CMID00() */
```

2.27 動作例 26 倍周クロックを用いた入力信号の欠け歯の検出[タイマ B]

2.27.1 概要

- 1) 図 2.43に示すように外部信号の周期を計測し、それに対する倍周クロックを生成します。
- 2) 生成した倍周クロックのカウントから、外部入力信号の欠け歯を検出します。

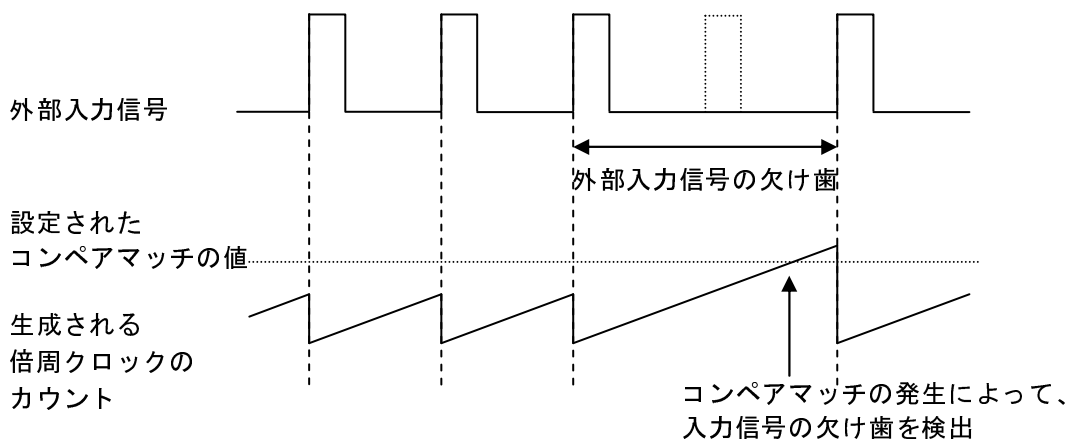


図 2.43 倍周クロックの欠け歯の検出

2.27.2 使用機能説明

表 2.28に使用端子及び関連レジスタの機能割り付けを示します。

表 2.28 ATU-III機能割り付け

使用端子		機能
端子	TIA00	測定するパルスを端子に入力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
	CBCNT	クロックバス 5 のクロックソースを設定します。
タイマ A レジスタ	TCRA	タイマ A の動作クロック、外部入力エッジのイベント出力を設定します。
	TIOR1A	TIA00 の立ち上がりエッジを検出する。
タイマ B レジスタ	TCRB	TCNTB0、TCNTB2 のカウントソースを設定します。
	TIORB	カウント動作の許可等、倍周クロック生成に関する設定を行います。
	PIMR	外部入力信号に対して生成する、倍周クロックの逡倍比を設定します。
	TIERB	割り込み要求の許可を設定します。
	TCNTB6	生成した倍周クロックでカウント動作を行います。
	TCCLRB	補正カウンタクリア値を設定します。
	OCRB6	欠け歯として検出する入力エッジ間隔を設定します。
	TSRB	タイマ B のステータスが格納されます。
PFC レジスタ	PECR1	使用端子の機能を設定します。
INTC レジスタ	IPR07	CMIB6 のコンペアマッチの優先度を設定します。

2.27.3 動作説明

図 2.44に動作原理を示します。これに従って SH72546R のハードウェアおよびソフトウェア処理によって入力信号の欠け歯の検出を行います。コンペアマッチが発生した時点で、外部クロックに欠け歯があることが判明します。

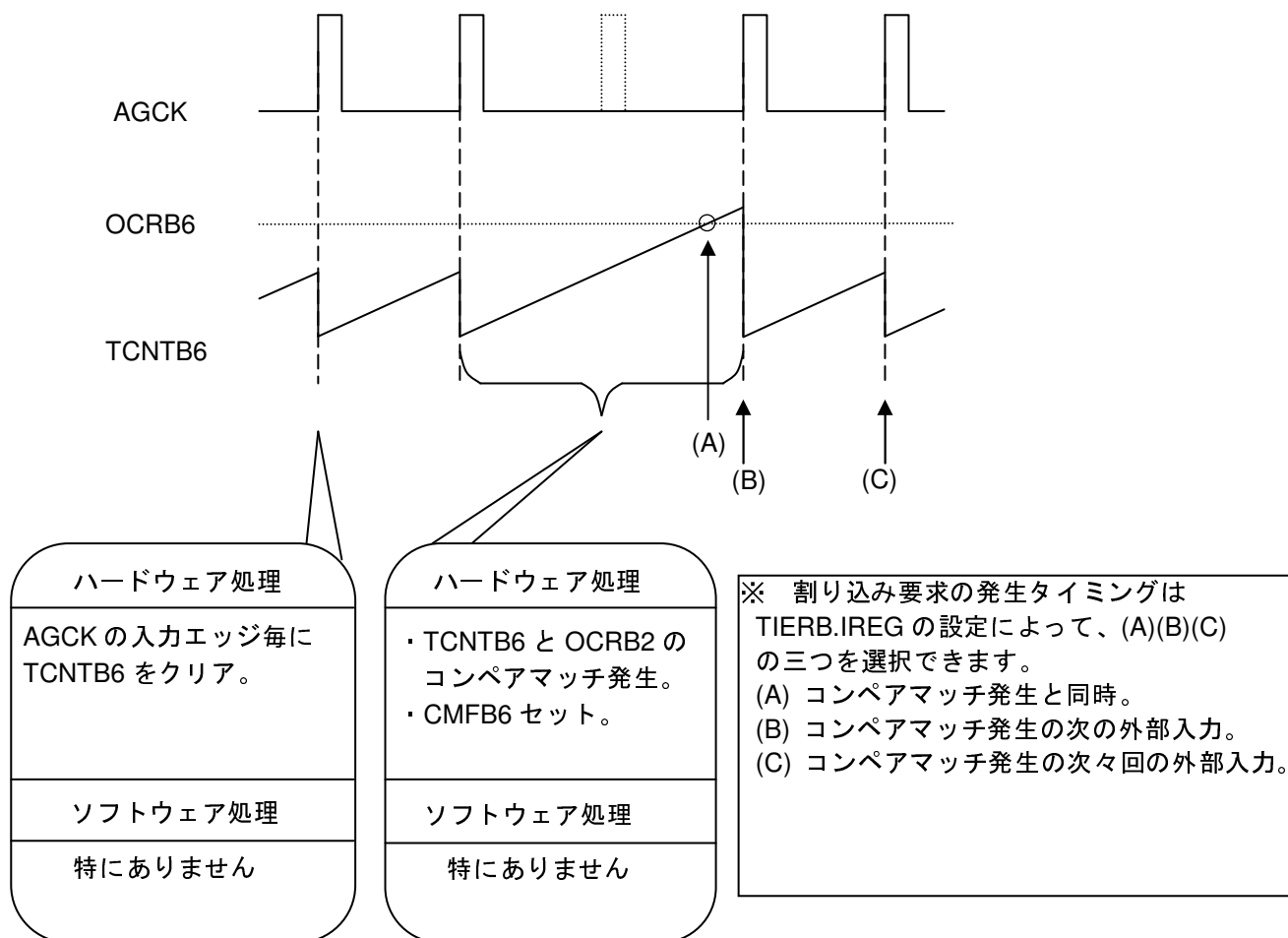


図 2.44 欠け歯検出動作原理

2.27.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
欠け歯検出ルーチン	CMIB6	欠け歯を検出するとこのルーチンにジャンプします。

● 使用変数の説明

本タスクでは変数は使用していません。

● 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE0 端子を入力端子 TIA0 に設定します。	0x0001	メインルーチン
PSCR0	プリスケアラ 0 の分周比を 10 に設定します。	0x0009	
ATUENR	ATU-IIIのチャンネル A,B 及びプリスケアラのカウント動作を許可します。	0x0007	
CBCNT	クロックバス 5 のクロックソースをタイマ B の倍周補正クロックに設定します。	0x04	
TCRA	TIA0 への入力信号をイベントとしてタイマ B に出力します。	0x08	
TIOR1A	TIA0 への入力信号の立ち上がりエッジを検出する。	0x0001	
TIORB	TCNTB2 のリロード値を ICRB0 の値に設定、外部イベントの入力を許可します。	0x21	
PIMR	外部入力周期に対する倍周クロックの逡倍比を 4 に設定します。	0x0004	
TCRB	TCNTB0,TCNTB2 の使用クロックをクロックバス 0 に設定します。	0x00	
TIERB	OCRB6 コンペアマッチ時の割り込みを許可、また割り込み出力のタイミングを設定します。	0x18	
TCNTB6	入力エッジ間計測カウント TCNTB6 の初期値を 0 に設定します。	0x00000000	
OCRB6	欠け歯検出を判断する、TCNTB5 とのコンペアマッチの値を 2 に設定します。※	0x00002000	
TCNTB5	入力エッジ間計測カウント TCNTB5 の初期値を 0 に設定します。	0x00000000	
TCCLRB	TCNTB5 の上限を 150 に設定します。	0x00096000	
CBCNT	クロックバス 5 をタイマ B からの倍周補正クロックに設定	0x04	
IPR07	TCNTB6 のコンペアマッチの割り込み優先順位を 15 に設定します。	0x0F00	
TSRB	コンペアマッチフラグをクリアします。	0x00	
TCNTB3	入力エッジ間計測カウント TCNTB3 をクリアします。	0x00000000	

※ サンプルプログラムは 1 歯欠けから検出するように値を設定しています。

2.27.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS              : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Detection of missing teeth using multiple clock input signal.
 */
/*
 * History : DD.MM.YYYY Version Description
 *          : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
void main(void);           /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PECCR1
b15:14 PE7MD = 0   PE7 入出力 (ポート)
b13:12 PE6MD = 0   PE6 入出力 (ポート)
b11     Reserved
b10     PE5MD = 0   PE5 入出力 (ポート)
b9      Reserved
b8      PE4MD = 0   PE4 入出力 (ポート)
b7      Reserved
b6      PE3MD = 0   PE3 入出力 (ポート)
b5:4 PE2MD = 0   PE2 入出力 (ポート)
b3:2 PE1MD = 0   PE1 入出力 (ポート)
b1      Reserved
b0      PE0MD = 1   TIA00 入力 (ATU-III)  */
PORTE.CR1.WORD = 0x0001; /* PE0 を TIA00 入力に設定
*/

  /* ATU-III (A) の設定 */
  /* Configure TCRA
b7      EVOSEL2A = 0 イベント出力 2A セレクト
b6      EVOSEL2B = 0 イベント出力 2B セレクト
b5:3 EVOSEL1  = 001 タイマ B へ TIA00 の入力エッジを出力
b2:0 CKSELA= 000 カウントクロックにクロックバス 0 (プリスケアラ 0) を選択 */
ATUA.TCRA.BYTE = 0x08; /* タイマ A をクロックバス 0 のカウントクロックで動作
                        TIA00 の入力をイベントとしてタイマ B に出力          */

  /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00   TIA のインプットキャプチャ禁止
b9:8 IOA4 = 00   TIA のインプットキャプチャ禁止
b7:6 IOA3 = 00   TIA のインプットキャプチャ禁止
b5:4 IOA2 = 00   TIA のインプットキャプチャ禁止
b3:2 IOA1 = 00   TIA のインプットキャプチャ禁止
b1:0 IOA0 = 01   TIA の立ち上がりで ICRA にキャプチャ  */
ATUA.TIOR1A.WORD = 0x0001; /* TIA00 入力の立ち上がりエッジで ICRA にインプットキャプ
チャ          */

  /* ATU-III (B) の設定 */
  /* Configure TIORB
b7      LDSEL = 0   TCNTB2 にロードする値に ICRB0 を用いる
b6      CTCNTB5 = 0   TCNTB5 のカウントを許可
b5      EVCNTB = 1   外部イベントの入力を許可
b4      LDEN = 0    リロードレジスタの更新を許可
b3      CCS = 0     TCNTB=TCNTB4 のとき、TCNTB4 のカウント動作を停止しない
b2:1 Reserved
b0      IOB6 = 1    TCNTB6 と OCRB6 とのコンペアマッチを許可 */
ATUB.TIORB.BYTE = 0x21; /* TCNTB2 にロードする値を ICRB0 に設定。TCNTB5 のカウントを
許可
                        外部イベントの入力を許可、リロードレジスタの更新を許可
                        TCNTB6 と OCRB6 とのコンペアマッチを許可          */

  /* Configure PIMR
b15:12 Reserved
b11:0 PIM = H'0004 クロックの通倍比を 4 倍に設定 */
ATUB.PIMR.WORD = 0x0004; /* 倍周クロックの通倍比を 4 倍に設定          */
}
```

```
/* Configure TCRB
b7:2 Reserved
b1:0 CKSELB = 00      カウントソースにクロックバス 0 を選択 */
ATUB.TCRB.BYTE = 0x00; /* TCNTB0,TCNTB2 をクロックバス 0 のカウントクロックで動作
*/

/* Configure TIERB
b7:6 Reserved
b5:4 IREG = 01      CMFB6 有効後、次の外部イベント入力タイミングで割り込み要求を出力
b3      CMEB6 = 1      コンペアマッチ割り込み B6 要求の出力を許可
b2      CMEB1 = 0      コンペアマッチ割り込み B1 要求の出力を禁止
b1      ICEB0 = 0      インพุットキャプチャ割り込み B0 要求の出力を禁止
b0      CMEB0 = 0      コンペアマッチ割り込み B0 要求の出力を禁止 */
ATUB.TIERB.BYTE = 0x18; /* TCNTB6,OCRB6 のコンペアマッチ発生のタイミングでの割り込み
要求を許可 */

/* Configure TCNTB
b31:0 入力エッジ間計測カウント値 = 0 */
ATUB.TCNTB6.LONG = 0x00000000; /* TCNTB6 の初期値を 0 に設定
*/

/* Configure OCRB6
b31:12 OCB6 = H'00002 アウトプットコンペア値を、2 に設定
b11:0  Reserved */
ATUB.OCRB6.LONG = 0x00002000; /* TCNTB6 に対応するコンペアマッチの値を 2 に設定
*/

/* Configure TCNTB
b31:0 入力エッジ間計測カウント値 = 0 */
ATUB.TCNTB5.LONG = 0x00000000; /* TCNTB5 の初期値を 0 に設定
*/

/* Configure TCCLRB
b31:12 CCLRB = 96      補正カウンタクリア値 B = 150
b11:0  Reserved */
ATUB.TCCLRB.LONG = 0x00096000; /* TCNTB5 の上限を 150 に設定
*/

/* ATU-III (共通) の設定 */
/* Configure CBCNT
b7:6 Reserved
b5:4 CB4EG = 00      クロックバス 4:外部クロックのエッジ検出をしない
b3      Reserved
b2      CB5SEL = 1      クロックバス 5:タイマ B 出力の倍周補正クロック
b1:0 CB5EG = 00      クロックバス 5:外部クロックのエッジ検出をしない */
ATUCTRL.CBCNT.BYTE = 0x04; /* クロックバス 5 をタイマ B からの倍周補正クロック () に
設定 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'9      プリスケーラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケーラ 0 の分周比を 10 に設定
*/

/* Configure IPR07
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:12 ATU-B(CMIB0,CMIB1) = H'0   対応する割り込みの優先順位を 0 に設定
b11:8  ATU-B(CMIB6,ICIB0) = H'F   対応する割り込みの優先順位を 15 に設定
b7:4  ATU-C0(IMIC0-IMIC3) = H'0   対応する割り込みの優先順位を 0 に設定
b3:0  ATU-C0(OVIC0) = H'0         対応する割り込みの優先順位を 0 に設定 */
INTC.IPR07.WORD = 0x0F00;   /* TCNTB6 のコンペアマッチの割り込み優先順位を 15 に設定
*/

set_imask(0);                /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0               タイマ J のカウント動作を停止
b8     THE = 0               タイマ H のカウント動作を停止
b7     TGE = 0               タイマ G のカウント動作を停止
b6     TFE = 0               タイマ F のカウント動作を停止
b5     TEE = 0               タイマ E のカウント動作を停止
b4     TDE = 0               タイマ D のカウント動作を停止
b3     TCE = 0               タイマ C のカウント動作を停止
b2     TBE = 1               タイマ B のカウント動作を許可
b1     TAE = 1               タイマ A のカウント動作を許可
b0     PSCE = 1             プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0007; /* タイマ A,B およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
* Function Name : CMIB6
* Description   : コンペアマッチ割り込み
* Arguments     : none
* Return Value  : none
*****/
void CMIB6(void)
{
  /* Configure TSRB
  b7:4 Reserved
  b3     CMFB6 = 0   コンペアマッチフラグをクリア
  b2     CMFB1 = 0   コンペアマッチフラグをクリア
  b1     ICFB0 = 0   コンペアマッチフラグをクリア
  b0     CMFB0 = 0   コンペアマッチフラグをクリア */
  ATUB.TSRB.BYTE &= 0;   /* コンペアマッチフラグをクリア */

  /* Configure TCNTB
  b31:0  入力エッジ間計測カウント値 = 0 */
  ATUB.TCNTB3.LONG = 0x00000000; /* TCNTB3 をクリア */
} /* End of function CMIB6() */
```

2.28 動作例 27 倍周補正クロック信号の生成[タイマ B]

2.28.1 概要

TCNTB3、TCNTB4、TCNTB5、TCCLR B の各レジスタを使用する事で、一定の外部入力区間で生成するクロックの平均を、入力信号の変化に関わらず、PIMR の値に補正したクロックを生成します。生成されたパルスを倍周補正クロックと言い、他のタイマでも使用できるよう、クロックバス 5 のクロックソースとして設定します。

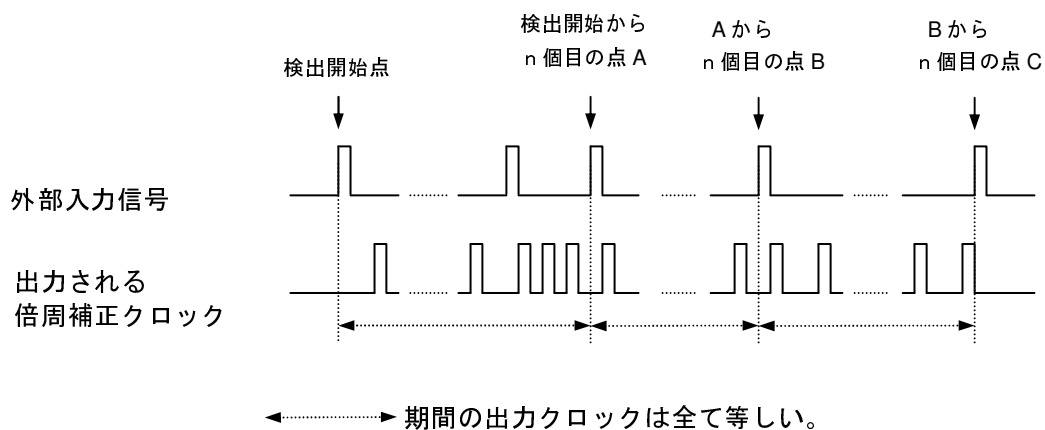


図 2.45 倍周補正クロック信号の生成(概要)

2.28.2 使用機能説明

表 2.29に使用端子及び関連レジスタの機能割り付けを示します。

表 2.29 ATU-III機能割り付け

使用端子		機能
端子	TIA00	倍周クロックの生成に使用する外部入力信号をこの端子に入力します。
	TOD00~02B	ワンショットパルスをこの端子から出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCRO	プリスケアラ0の分周比を設定します。
	CBCNT	クロックバス5のクロックソースを設定します。
タイマA レジスタ	TCRA	タイマAの動作クロック、外部入力エッジのイベント出力を設定します。
	TIOR1A	TIA1の検出エッジを設定します。
タイマB レジスタ	TIERB	割り込み許可の設定をします。
	TCRB	TCNTB0、TCNTB2のカウントソースを設定します。
	TIORB	カウント動作の許可等、倍周クロック生成に関する設定を行います。
	PIMR	生成する倍周クロックの通倍比を設定します。
	TCNTB3	外部入力信号毎にTCNTB4に値を転送し、PIMRの値が加算されます。
	TCNTB4	外部入力信号毎にTCNTB3の値を読み込み、倍周クロックでアップカウント動作を行います。
	TCNTB5	TCNTB4の値より小さいときに、PΦクロックでアップカウント動作を行います。
	TCCLRB	TCNTB5の上限値として機能し、このレジスタとTCNTB5が一致すると、TCNTB5はクリアされます。
	TCNTB6	入力エッジ間計測カウントです。
	OCRB6	TCNTB6に対応するコンペアマッチの値を設定します。
TSRB	各コンペアマッチの発生状況を示します。	
PFCレジスタ	PECR1	PE1をTIA01に設定します。
	PF0R1	PF0,PF1,PF2をTOD出力00B,01B,02Bに設定します。
INTCレジスタ	IPR06	割り込みの優先順位を設定します。
	IPR07	割り込みの優先順位を設定します。
タイマD レジスタ	TCRD	タイマDの各カウンタの動作クロックバスを設定します。
	TIOR1D0	各コンペアマッチの発生の有無及び、発生時の動作を設定します。
	TSRD0	各コンペアマッチの発生状況を示します。
	DCNTD00~02	ダウンカウンタの値を設定します。
	DSTRD0	ダウンカウンタの動作許可を設定します。

2.28.3 動作説明

(1) 動作原理

図 2.46に減速時、図 2.47に加速時の動作原理を示します。これに従って SH72546R のハードウェア処理によって入力信号に対する倍周クロックの生成を行います。

[共通処理]

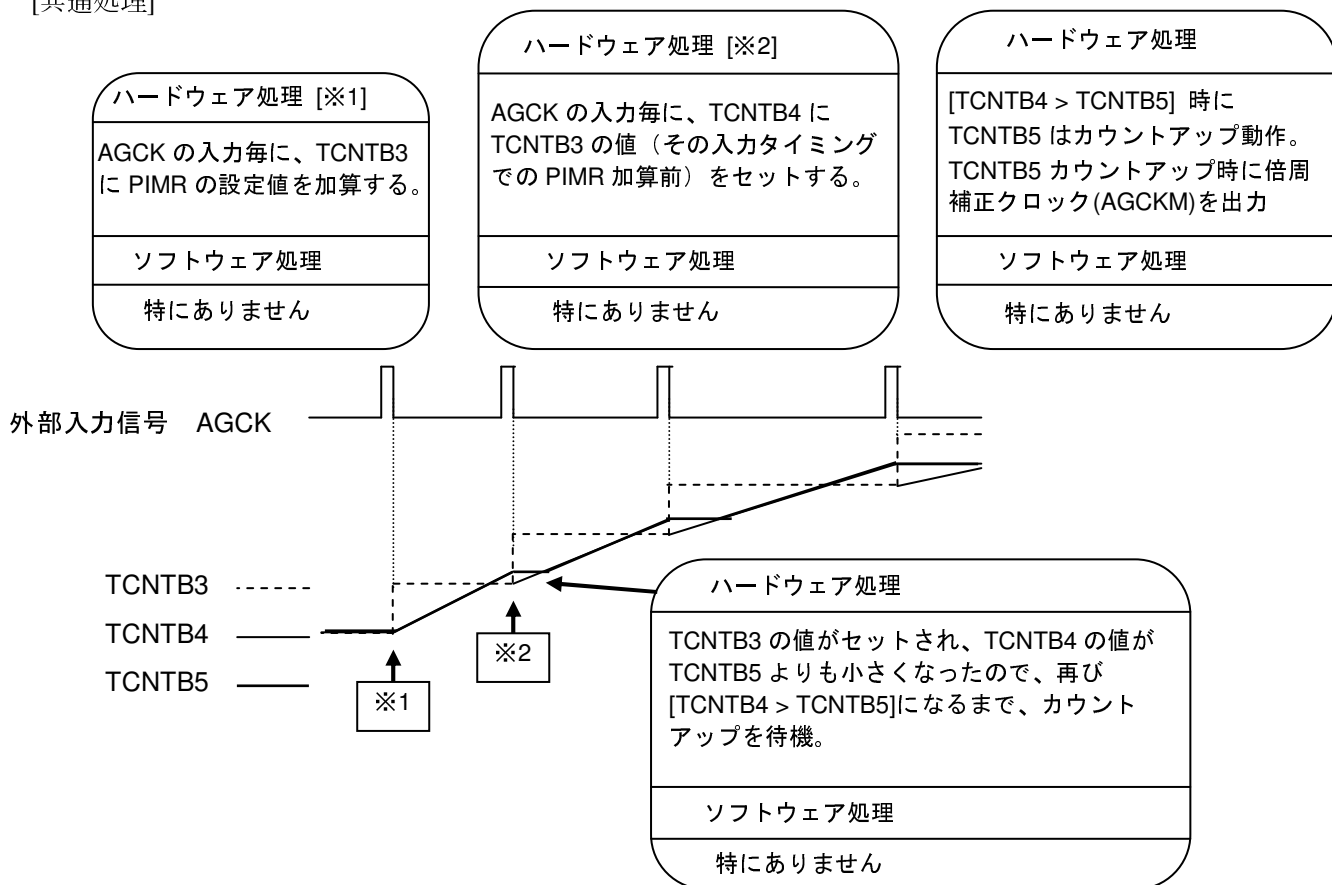


図 2.46 倍周補正クロック出力原理(減速時)

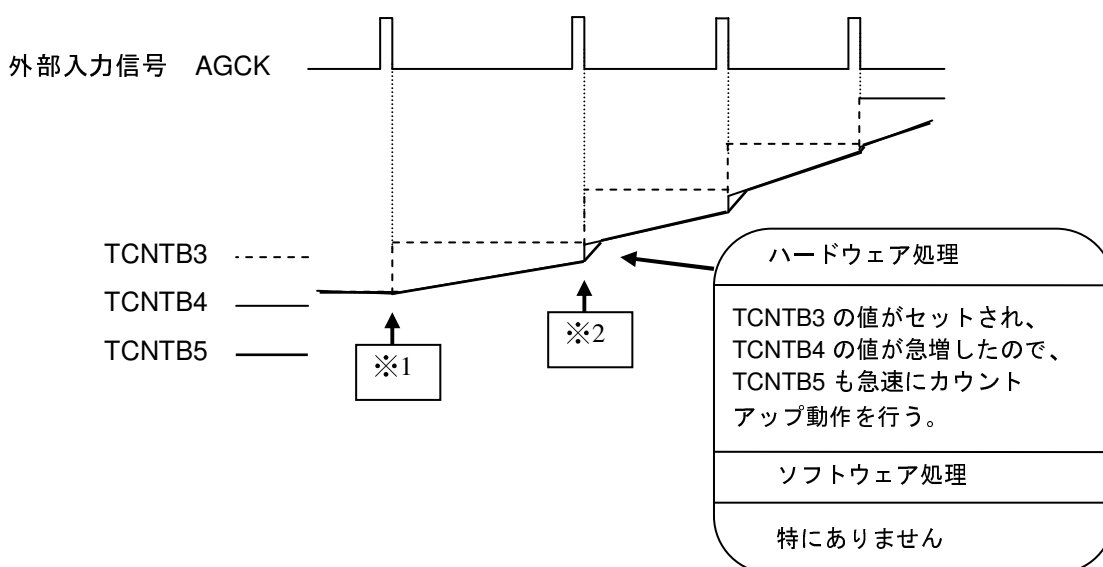


図 2.47 倍周補正クロック出力原理(加速時)

(2) 使用例

周期の変化する外部入力信号に対して、その信号から倍周クロックを生成する場合、補正を用いない場合と、用いる場合では下図のように動作が異なります。それぞれをクロックに用いて、例えば「ある外部入力から 12 カウント（外部入力 3 歯分）だけパルスを出力する」というソフトウェアを実行した場合の波形を図 2.48、図 2.49 に示します。（PIM の値が 4 の場合）

[1] 入力信号の間隔が増加していく場合（減速時）

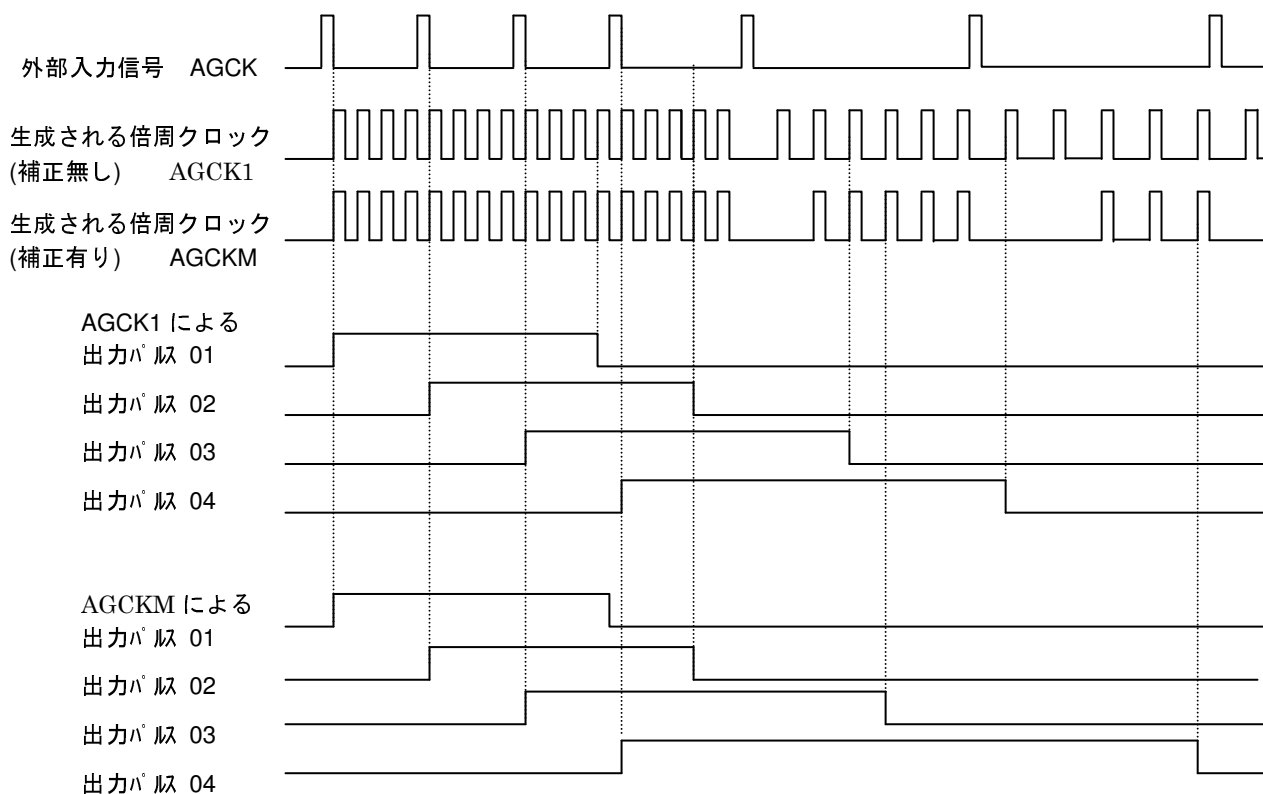


図 2.48 倍周機能使用に関する比較(減速時、PIM=4 の場合)

[2]入力信号の間隔が減少していく場合（加速時）

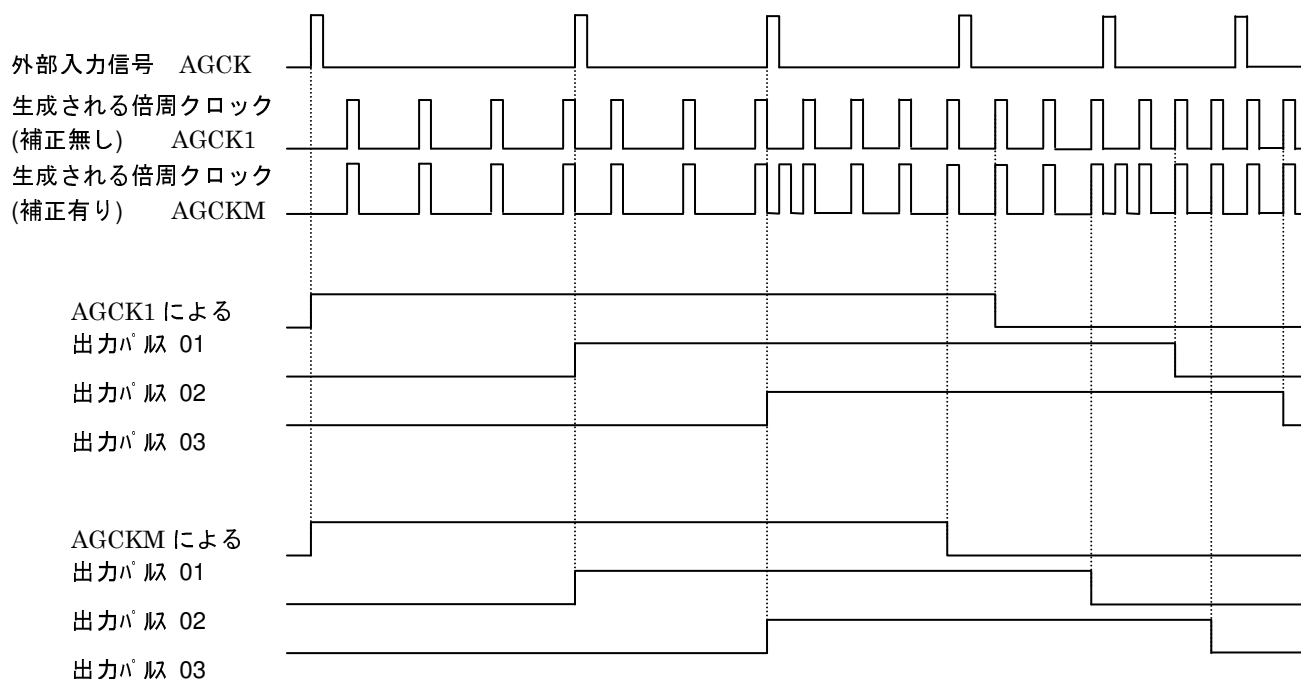


図 2.49 倍周機能使用に関する比較(加速時、PIM=4 の場合)

このように、倍周クロックを用いたカウント動作を基準にする事で、入力信号の波長が変化した場合に、波長の変化に対するソフト的な処理を行わなくても、波長に追従して処理を行う事ができます。そして、補正機能を使用する事により、より正確に外部入力信号の間隔に対応します。

2.28.4 ソフトウェア説明

- 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	ATU-IIIの初期設定を行います。
インプットキャプチャ割り込み	CMIB6	TCTB6 と OCRB6 のコンペアマッチ割り込みが発生したら、コンペアマッチフラグをクリアします。
パルス出カルーチン	CMID00	TCNTB1 と OCRB1 のコンペアマッチが発生したら、ワンショットパルスを出力します。

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

- 使用変数の説明

ラベル名	機能	データ長	使用モジュール
cnt	ワンショットパルス出力ルーチン参照の回数を記録します。	unsigned char	ワンショットパルス出力ルーチン

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名	
PECR1	ポート E の PE0 端子を入力端子 TIA001 に設定します。	0x0001	メインルーチン	
PFCR1	ポート F0,F1,F2 を TOD 出力 00B,01B,02B に設定します。	0x0015		
PSCR0	プリスケアラ 0 の分周比を 100 に設定します。	0x0063		
ATUENR	ATU-IIIチャネル A,B,及びプリスケアラのカウント動作を許可します。	0x0007		
CBCNT	クロックバス 5 のクロックソースをタイマ B の倍周補正クロックに設定します。	0x04		
TCRA	タイマ B へ TIA00 の入力エッジを出力します。	0x08		
TIERA	割り込みを許可します。	0x02		
TIOR1A	TIA00 への入力信号の立ち上がりエッジを検出する。	0x0001		
TIORB	TCNTB2 のリロード値に ICRB0 の値を使用、外部イベントの入力を許可、そして、TCNTB3=TCNTB4 の時に TCNTB4 のカウントを停止するよう設定します。	0x21		
PIMR	外部入力周期に対する倍周クロックの逡倍比を 10 に設定します。	0x000A		
TCRB	TCNTB0,TCNTB2 の使用クロックをクロックバス 0 に設定します。	0x00		
TCCLRB	TCNTB5 の上限値を 104 に設定します。	0x00068000		
TIERB	コンペアマッチ B6 の割り込み要求を許可します。	0x08		
TCNTB5	TCNTB5 の初期値を 0 に設定します。	0x00000000		
TCNTB6	TCNTB6 の初期値を 0 に設定します。	0x00000000		
OCRB6	TCNTB6 のコンペアマッチの値を 15 に設定します。	0x0000F000		
IPR07	タイマ B に関する割り込みの優先度を設定します。	0x0F00		
TSRB	コンペアマッチフラグ CMFB6 をクリアします。	0x00		コンペアマッチ割り込みルーチン
TSRD0	コンペアマッチ、アンダーフローフラグをクリアします。	0x0000		ワンショットパルス出力ルーチン
TCNTB3	補正イベントカウント B3 をクリアします。	0x00000000		
TCNT1D0	タイマカウンタ 1D0 をクリアします。	0x00000000		
DCNTD00~02	ダウンカウンタ 00~02 に 250 をセットします。	0x00FA<<8		
DSTRD0	ダウンカウンタの停止/開始を設定します。	0x01 0x02 0x04		

2.28.5 サンプルプログラム

```
/*
 * DISCLAIMER
 * This software is supplied by Renesas Electronics Corporation and is only
 * intended for use with Renesas products. No other uses are authorized. This
 * software is owned by Renesas Electronics Corporation and is protected under
 * all applicable laws, including copyright laws.
 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
 * THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
 * LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
 * AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
 * ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
 * ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
 * BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
 * Renesas reserves the right, without notice, to make changes to this software
 * and to discontinue the availability of this software. By using this software,
 * you agree to the additional terms and conditions found by accessing the
 * following link:
 * http://www.renesas.com/disclaimer *
 * Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
 */
/*
 * File Name      : SH7254R_ATU.c
 * Version        : 1.00
 * Device(s)      : SH72546R
 * Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
 *                 : C/C++ compiler package for the SuperH RISC engine family
 *                 : (Ver.9.04 Release00).
 * OS             : None
 * H/W Platform   : SH7254R
 * Description     : This is the main tutorial code.
 * Operation      : Multiple correction clock signal generation.
 */
/*
 * History : DD.MM.YYYY Version Description
 *         : 12.01.2012 1.00 First Release
 */
/*
 * Includes <System Includes> , "Project Includes"
 */
#include <machine.h> /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h" /* 周辺レジスタ定義ヘッダファイル */

/*
 * Private global variables and functions
 */
unsigned char cnt;
void main(void); /* メインルーチン */

/*
 * Function Name : main
 * Description   : The main loop
 * Arguments    : none
 * Return Value  : none
 */

```

```
void main(void)
{
  /* ポートの設定 */
  /* Configure PECR1
b15:14 PE7MD = 0    PE7 入出力 (ポート)
b13:12 PE6MD = 0    PE6 入出力 (ポート)
b11    Reserved
b10    PE5MD = 0    PE5 入出力 (ポート)
b9     Reserved
b8     PE4MD = 0    PE4 入出力 (ポート)
b7     Reserved
b6     PE3MD = 0    PE3 入出力 (ポート)
b5:4 PE2MD = 0    PE2 入出力 (ポート)
b3:2 PE1MD = 0    PE1 入出力 (ポート)
b1     Reserved
b0     PE0MD = 1    TIA00 入力 (ATU-III)  */
PORTE.CR1.WORD = 0x0001;    /* PE0 を TIA00 入力に設定 */

  /* Configure PFCR1
b15:14 PF7MD = 00    PF7 モード = PF7 入出力 (ポート)
b13:12 PF6MD = 00    PF6 モード = PF6 入出力 (ポート)
b11:10 PF5MD = 00    PF5 モード = PF5 入出力 (ポート)
b9:8 PF4MD = 00    PF4 モード = PF4 入出力 (ポート)
b7:6 PF3MD = 00    PF3 モード = PF3 入出力 (ポート)
b5:4 PF2MD = 01    PF2 モード = TOD22B 出力 (ATU-III)
b3:2 PF1MD = 01    PF1 モード = TOD21B 出力 (ATU-III)
b1:0 PF0MD = 01    PF0 モード = TOD20B 出力 (ATU-III)  */
PORTF.CR1.WORD = 0x0015;    /* PF0,1,2 を TOD00B,01B,02B 出力に設定 */

  /* ATU-III (A) の設定 */
  /* Configure TCRA
b7     EVOSEL2A = 0 イベント出力 2A セレクト
b6     EVOSEL2B = 0 イベント出力 2B セレクト
b5:3 EVOSEL1  = 001 タイマ B へ TIA00 の入力エッジを出力
b2:0 CKSELA= 000 カウントクロックにクロックバス 0 (プリスケアラ 0) を選択 */
ATUA.TCRA.BYTE = 0x08;    /* タイマ A をクロックバス 0 のカウントクロックで動作
                             TIA00 の入力をイベントとして、タイマ B に出力 */

  /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00    TIA05 のインプットキャプチャ禁止
b9:8 IOA4 = 00    TIA04 のインプットキャプチャ禁止
b7:6 IOA3 = 00    TIA03 のインプットキャプチャ禁止
b5:4 IOA2 = 00    TIA02 のインプットキャプチャ禁止
b3:2 IOA1 = 00    TIA01 のインプットキャプチャ禁止
b1:0 IOA0 = 01    TIA00 の立ち上がりで ICRA にキャプチャ */
ATUA.TIOR1A.WORD = 0x0001;    /* TIA00 の立ち上がりエッジでインプットキャプチャ */

  /* ATU-III (B) の設定 */
  /* Configure TIORB
b7     LDSEL = 0    TCNTB2 にロードする値に ICRB0 を用いる
b6     CTCNTB5 = 0    TCNTB5 のカウントを許可
b5     EVCNTB = 1    外部イベントの入力を許可
b4     LDEN = 0    リロードレジスタの更新を許可
b3     CCS = 0    TCNTB=TCNTB4 のとき、TCNTB4 のカウント動作を停止しない
b2:1 Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b0      IOB6 = 1      TCNTB6 と OCB6 とのコンペアマッチを許可 */
ATUB.TIORB.BYTE = 0x21;          /* TCNTB2 にロードする値を ICRB0 に設定。TCNTB5 のカ
ウントを許可

                                     外部イベントの入力を許可、リロードレジスタの更新を許可
                                     TCNTB6 と OCB6 とのコンペアマッチを許可          */

/* Configure PIMR
b15:12 Reserved
b11:0   PIM = H'000Aクロックの通倍比を 10 倍に設定 */
ATUB.PIMR.WORD = 0x000A;        /* 倍周クロックの通倍比を 10 倍に設定          */

/* Configure TCRB
b7:2 Reserved
b1:0 CKSELB = 00      カウントソースにクロックバス 0 を選択 */
ATUB.TCRB.BYTE = 0x00;          /* TCNTB0, TCNTB2 をクロックバス 0 のカウントクロックで動
作          */

/* Configure TIERB
b7:6 Reserved
b5:4 IREG = 00      CMFB6 が有効になったタイミングで割り込み要求出力
b3      CMEB6 = 1    コンペアマッチ割り込み B6 要求の出力を許可
b2      CMEB1 = 0    コンペアマッチ割り込み B1 要求の出力を禁止
b1      ICEB0 = 0    インพุットキャプチャ割り込み B0 要求の出力を禁止
b0      CMEB0 = 0    コンペアマッチ割り込み B0 要求の出力を禁止 */
ATUB.TIERB.BYTE = 0x08;          /* TCNTB6, OCB6 のコンペアマッチ発生タイミングでの割り込み要
求を許可 */

/* Configure TCNTB
b31:0   入力エッジ間計測カウント値 = 0 */
ATUB.TCNTB6.LONG = 0x00000000; /* TCNTB6 の初期値を 0 に設定
*/

/* Configure OCB6
b31:12 OCB6 = H'0000F アウトプットコンペア値を、15 に設定
b11:0   Reserved */
ATUB.OCRB6.LONG = 0x0000F000; /* TCNTB6 に対応するコンペアマッチの値を 15 に設定
*/

/* Configure TCNTB
b31:0   入力エッジ間計測カウント値 = 0 */
ATUB.TCNTB5.LONG = 0x00000000; /* TCNTB5 の初期値を 0 に設定
*/

/* Configure TCCLRB
b31:12 CCLRB = H'68補正カウンタクリア値 B = D'104
b11:0   Reserved */
ATUB.TCCLRB.LONG = 0x00068000; /* TCNTB5 の上限を 104 に設定
*/

/* ATU-III (共通) の設定 */
/* Configure CBCNT
b7:6 Reserved
b5:4 CB4EG = 00      クロックバス 4:外部クロックのエッジ検出をしない
b3      Reserved
b2      CB5SEL = 1    クロックバス 5:タイマ B 出力の倍周補正クロック
b1:0 CB5EG = 00      クロックバス 5:外部クロックのエッジ検出をしない*/
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUCTRL.CBCNT.BYTE = 0x04;      /* クロックバス 5 をタイマ B からの倍周補正クロック () に
設定 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'63 プリスケーラの分周比を 100 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0063; /* プリスケーラ 0 の分周比を 100 に設定 */

/* Configure IPR07
b15:12 ATU-B(CMIB0,CMIB1) = H'0   対応する割り込みの優先順位を 0 に設定
b11:8  ATU-B(CMIB6,ICIB0) = H'F   対応する割り込みの優先順位を 15 に設定
b7:4  ATU-C0(IMIC0-IMIC3) = H'0   対応する割り込みの優先順位を 0 に設定
b3:0  ATU-C0(OVIC0) = H'0         対応する割り込みの優先順位を 0 に設定 */
INTC.IPR07.WORD = 0x0F00; /* TCNTB6 のコンペアマッチの割り込み優先順位を 15 に設定
*/

set_imask(0); /* 割り込みマスクレベルを 0 に設定 */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0   タイマ J のカウント動作を停止
b8     THE = 0   タイマ H のカウント動作を停止
b7     TGE = 0   タイマ G のカウント動作を停止
b6     TFE = 0   タイマ F のカウント動作を停止
b5     TEE = 0   タイマ E のカウント動作を停止
b4     TDE = 0   タイマ D のカウント動作を停止
b3     TCE = 0   タイマ C のカウント動作を停止
b2     TBE = 1   タイマ B のカウント動作を停止
b1     TAE = 1   タイマ A のカウント動作を許可
b0     PSCE = 1   プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0007; /* タイマ A,B およびプリスケーラのカウントをスタート */

while(1);
} /* End of function main() */

/*****
* Function Name : CMIB6
* Description   : コンペアマッチ割り込み (B6)
* Arguments    : none
* Return Value  : none
*****/
void CMIB6(void)
{
    /* Configure TSRB
    b7:4 Reserved
    b3     CMFB6 = 0   コンペアマッチフラグをクリア
    b2     CMFB1 = 0   コンペアマッチフラグをクリア
    b1     ICFB0 = 0   コンペアマッチフラグをクリア
    b0     CMFB0 = 0   コンペアマッチフラグをクリア */
    ATUB.TSRB.BYTE &= 0; /* コンペアマッチフラグをクリア */
} /* End of function CMIB6() */

/*****
* Function Name : CMID00
* Description   : ワンショットパルス出力ルーチン
* Arguments    : none
*****/
```

```

* Return Value : none
*****/
void CMID00(void)
{
  /* Configure TSRD
  b15:14 Reserved
  b13   OVF2Dn = 0   TCNT2Dn フラグをオーバフローなしにクリア
  b12   OVF1Dn = 0   TCNT1Dn フラグをオーバフローなしにクリア
  b11   UDFDn3 = 0   UDFDn3 フラグをアンダーフローなしにクリア
  b10   UDFDn2 = 0   UDFDn2 フラグをアンダーフローなしにクリア
  b9    UDFDn1 = 0   UDFDn1 フラグをアンダーフローなしにクリア
  b8    UDFDn0 = 0   UDFDn0 フラグをアンダーフローなしにクリア
  b7    CMFADn3 = 0   CMFADn3 フラグをコンペアマッチなしにクリア
  b6    CMFADn2 = 0   CMFADn2 フラグをコンペアマッチなしにクリア
  b5    CMFADn1 = 0   CMFADn1 フラグをコンペアマッチなしにクリア
  b4    CMFADn0 = 0   CMFADn0 フラグをコンペアマッチなしにクリア
  b3    CMFBDn3 = 0   CMFBDn3 フラグをコンペアマッチなしにクリア
  b2    CMFBDn2 = 0   CMFBDn3 フラグをコンペアマッチなしにクリア
  b1    CMFBDn1 = 0   CMFBDn3 フラグをコンペアマッチなしにクリア
  b0    CMFBDn0 = 0   CMFBDn3 フラグをコンペアマッチなしにクリア*/
  ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダフローフラグをクリ
  ア */

  /* Configure TCNTB
  b31:0 入力エッジ間計測カウント値 = 0 */
  ATUB.TCNTB3.LONG = 0x00000000;

  /* Configure TCNA1D
  b31:8 タイマカウンタ値 = 0
  b7:0 Reserved */
  ATUD.SUBBLOCKA[0].TCNT1D = 0x00000000;

  cnt ++;
  if(cnt == 1)
  {
    /* Configure DCNTD00
    b31:8 タイマダウンカウンタ値
    b7:0 Reserved */
    ATUD.SUBBLOCKB[0].DCNTD[0] = 0x00FA << 8; /* ダウンカウンタ 00 を 250 にセット
    */

    /* Configure DSTRD0
    b7:4 Reserved
    b3   DSTD03=0   DCNTD03 ダウンカウント停止
    b2   DSTD02=0   DCNTD02 ダウンカウント停止
    b1   DSTD01=0   DCNTD01 ダウンカウント停止
    b0   DSTD00=1   DCNTD00 ダウンカウント開始 */
    ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x01; /* ダウンカウンタ 00 のダウンカウントを開始 */
  }
  else if(cnt == 4)
  {
    /* Configure DCNTD01
    b31:8 タイマダウンカウンタ値
    b7:0 Reserved */
    ATUD.SUBBLOCKB[0].DCNTD[1] = 0x00FA << 8; /* ダウンカウンタ 01 を 250 にセット
    */
  }
}

```

```
/* Configure DSTRD0
b7:4 Reserved
b3 DSTD03=0 DCNTD03 ダウンカウント停止
b2 DSTD02=0 DCNTD02 ダウンカウント停止
b1 DSTD01=1 DCNTD01 ダウンカウント開始
b0 DSTD00=0 DCNTD00 ダウンカウント停止 */
ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x02; /* ダウンカウンタ 01 のダウンカウントを開始 */
}
else if(cnt == 8)
{
/* Configure DCNTD02
b31:8 タイマダウンカウンタ値
b7:0 Reserved */
ATUD.SUBBLOCKB[0].DCNTD[2] = 0x00FA << 8; /* ダウンカウンタ 02 を 250 にセット
*/

/* Configure DSTRD0
b7:4 Reserved
b3 DSTD03=0 DCNTD03 ダウンカウント停止
b2 DSTD02=1 DCNTD02 ダウンカウント開始
b1 DSTD01=0 DCNTD01 ダウンカウント停止
b0 DSTD00=0 DCNTD00 ダウンカウント停止 */
ATUD.SUBBLOCKB[0].DSTRD.BYTE = 0x04; /* ダウンカウンタ 02 のダウンカウントを開始 */
}
else if(cnt == 10)
{
cnt = 0;
}
} /* End of function CMID00() */
```

2.29 動作例 28 倍周補正クロック信号の生成[タイマ B]

2.29.1 概要

欠け歯の含まれた外部入力信号に対して、その外部入力信号で生成した倍周補正クロック信号でダウンカウント動作を行うワンショットパルスを出力する事で、欠け歯を基準として、一定の位置でパルスを出力するような出力信号を生成します。

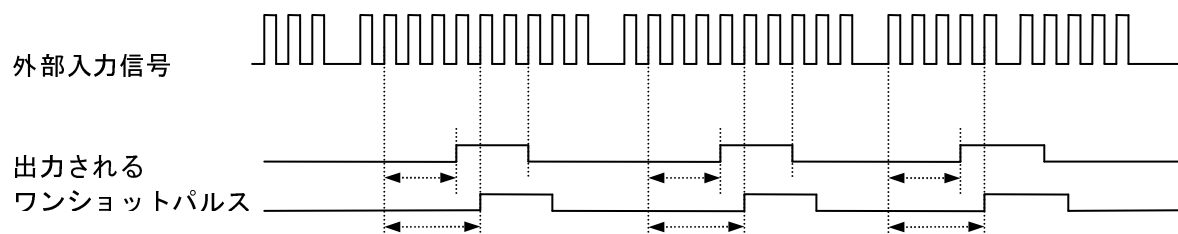


図 2.50 倍周補正クロック信号の生成(概要)

2.29.2 使用機能説明

表 2.30に使用端子及び関連レジスタの機能割り付けを示します。

表 2.30 ATU-III機能割り付け

使用端子		機能
端子	TIA01	倍周クロックの生成に使用する外部入力信号をこの端子に入力します。
	TOD00~02B	ワンショットパルスはこの端子から出力します。
関連レジスタ		機能
ATU-III 共通レジスタ	ATUENR	各タイマやプリスケアラの動作を設定します。
	PSCR0	プリスケアラ 0 の分周比を設定します。
	CBCNT	クロックバス 5 のクロックソースを設定します。
タイマ A レジスタ	TCRA	タイマ A の動作クロック、外部入力エッジのイベント出力を設定します。
	TIOR1A	TIA00 の検出エッジを設定します。
	TSRA	インプットキャプチャフラグが格納されます。
	TIERA	インプットキャプチャ割り込みの許可/禁止を設定します。
タイマ B レジスタ	TCNTB2	外部イベント入力により値が更新されます。
	LDB	TCNTB2、RLDB の更新に用いられます。
	RLDB	ICRB0 または、LDB の値を更新します。
	TIERB	割り込みの許可をします。
	TCNTB6	AGCKM によりカウントアップします。
	OCRB6	TCNTB6 と比較され一致したら、割り込みを発生します。
	TCRB	TCNTB0、TCNTB2 のカウントソースを設定します。
	TIORB	カウント動作の許可等、倍周クロック生成に関する設定を行います。
	PIMR	生成する倍周クロックの通倍比を設定します。
	TCNTB3	外部入力信号毎に TCNTB4 に値を転送し、PIMR の値が加算されます。
	TCNTB4	外部入力信号毎に TCNTB3 の値を読み込み、倍周クロックでアップカウント動作を行います。
	TCNTB5	TCNTB4 の値より小さいときに、PΦクロックでアップカウント動作を行います。
	TCCLR	TCNTB5 の上限値として機能し、このレジスタと TCNTB5 が一致すると、TCNTB5 はクリアされます。
	TCNTB6	OCRB6 と一致することで、コンペアマッチ割り込みを発生します。
	OCRB6	TCNTB6 に対応するコンペアマッチの値を設定します。
TSRB	タイマ B 各コンペアマッチ発生状況を示します。	
INTC レジスタ	IPR10	割り込みの優先順位を設定します。
	IPR06	割り込みの優先順位を設定します。
	IPR07	割り込みの優先順位を設定します。
タイマ D レジスタ	TCRD0	タイマ D 各カウンタの動作クロックバスを設定します。
	TIOR1D0	各コンペアマッチの発生の有無及び、発生時の動作を設定します。
	DCRD0	ダウンカウンタの動作開始、停止の条件を設定します。
	TSRD0	各コンペアマッチの発生状況を示します。
	OCRD00~02	OCRD00 と TCNT1D0、OCRD01 と TCNT1D1、OCRD02 と TCNT1D2 に対応するコンペアマッチ A の条件を設定。
	TIERD0	OCRD00~02 とのコンペアマッチ発生による割り込み要求を設定します。
	TSTRD	タイマ D のカウントをスタートします。

2.29.3 動作説明

(1) 動作原理

図 2.51に減速時の、図 2.52に加速時の動作原理を示します。これに従って SH72546R のハードウェア処理によって入力信号に対する倍周クロックの生成を行います。

[共通処理]

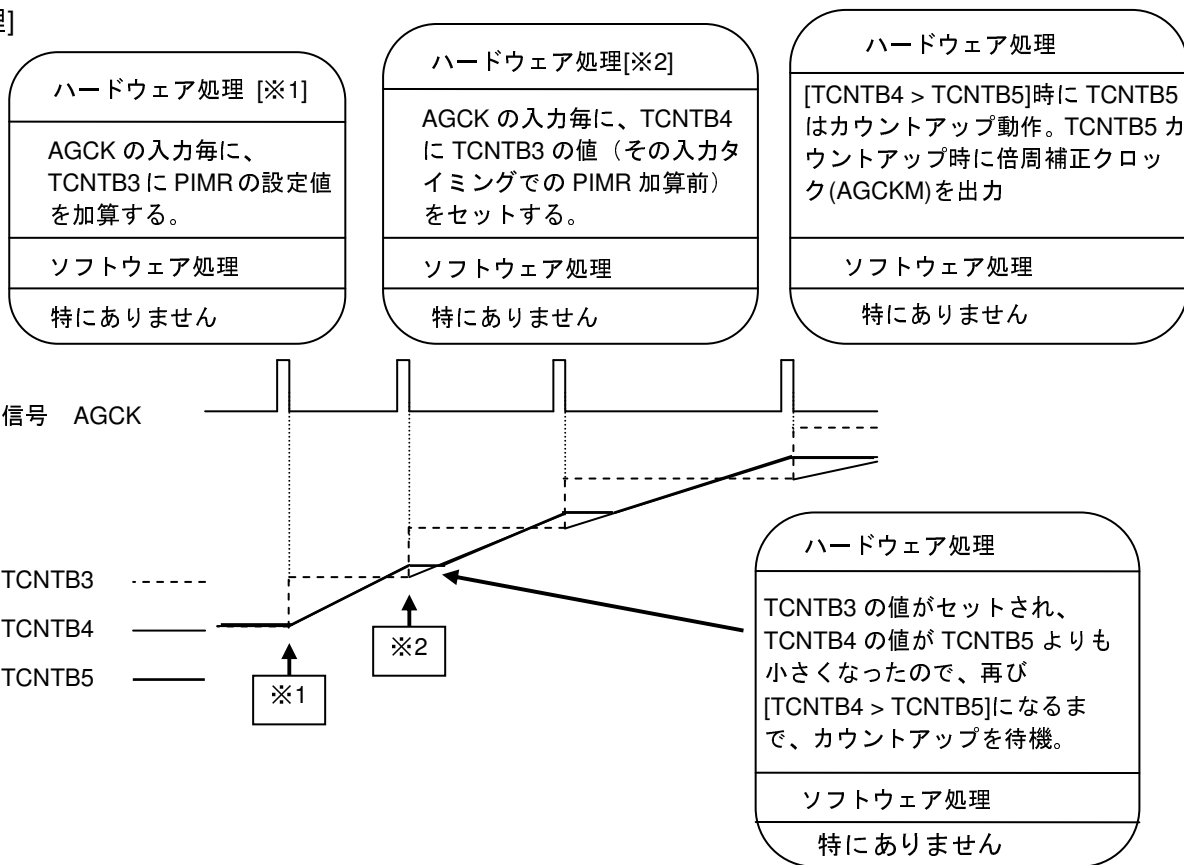


図 2.51 倍周補正クロック出力原理(減速時)

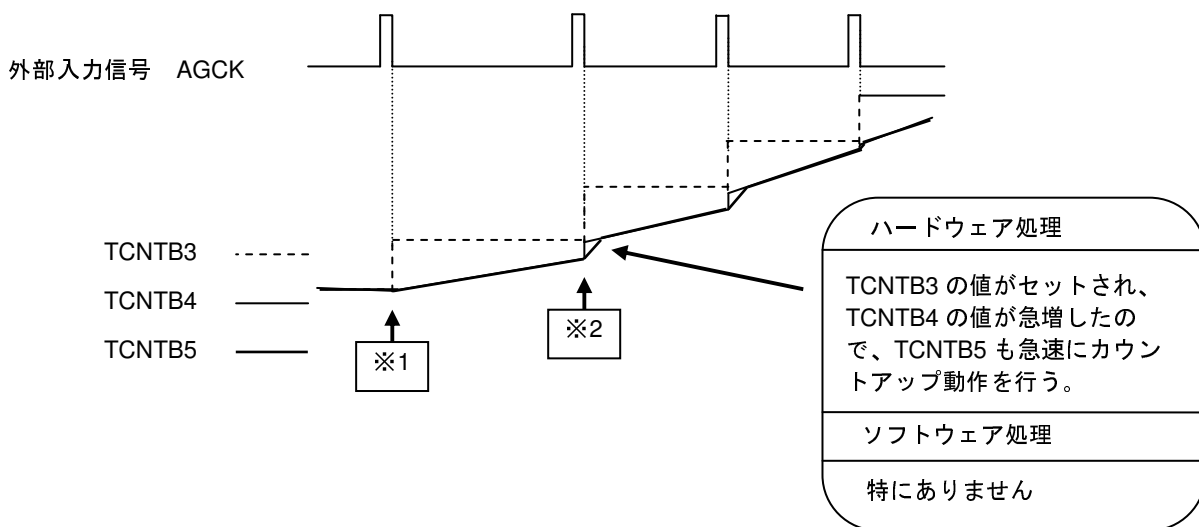


図 2.52 倍周補正クロック出力原理(加速時)

(2) 使用例

周期の変化する外部入力信号に対して、その信号から倍周クロックを生成する場合、補正を用いない場合と、用いる場合では下図のように動作が異なります。それぞれをクロックに用いて、例えば「ある外部入力から 12 カウント（外部入力 3 歯分）だけパルスを出力する」というソフトウェアを実行した場合の波形を図 2.53、図 2.54 に示します。（PIM の値が 4 の場合）

[1] 入力信号の間隔が増加していく場合（減速時）

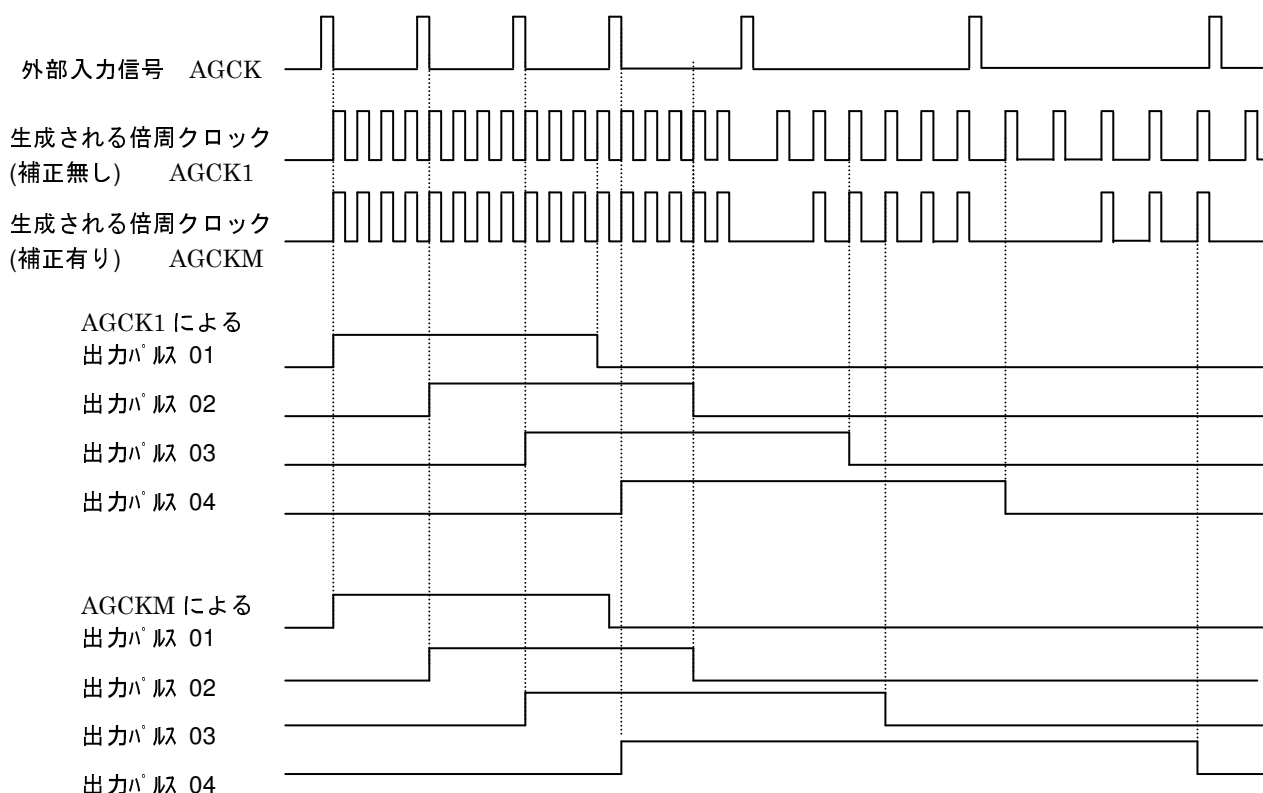


図 2.53 倍周機能使用に関する比較(減速時、PIM=4 の場合)

[2]入力信号の間隔が減少していく場合 (加速時)

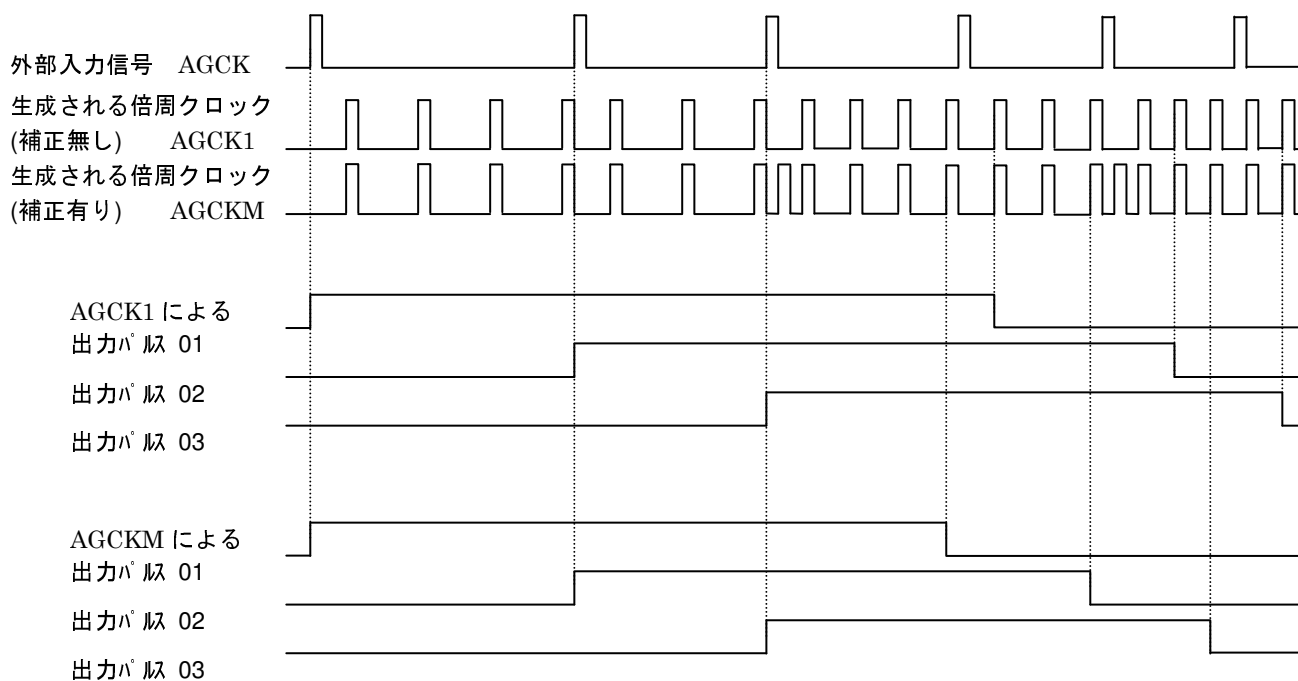


図 2.54 倍周機能使用に関する比較(加速時、PIM=4 の場合)

このように、倍周クロックを用いたカウント動作を基準にする事で、入力信号の波長が変化した場合に、波長の変化に対するソフト的な処理を行わなくても、波長に追従して処理を行う事ができます。そして、補正機能を使用する事により、より正確に外部入力信号の間隔に対応します。

2.29.4 ソフトウェア説明

● 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	ATU-IIIの初期設定を行います。
外部入力検知ルーチン	ICIA1	外部入力検知したら本ルーチンを行います。
フラグクリアルーチン	CMID00	TCNTD1 と OCRD のコンペアマッチが発生したら、本ルーチンを行います。
パルス出力ルーチン	CMIB6	TCNTB6 と OCRB6 のコンペアマッチが発生したら、ワンショットパルスを出力します。

● 使用変数の説明

本タスクでは変数は使用していません。

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

- 使用レジスタの説明

レジスタ名	機能	設定値	使用モジュール名
PECR1	ポート E PE1 端子を入力端子 TIA1 に設定します。	0x0004	メインルーチン
PFCR1	ポート F0,F1,F2 を TOD 出力 00B,01B,02B に設定します。	0x0015	
PSCR0	プリスケアラ 0 の分周比 10 に設定します。	0x0009	
ATUENR	ATU-IIIチャネル A,B,D 及びプリスケアラのカウンタ動作を許可します。	0x0017	
CBCNT	クロックバス 5 のクロックソースをタイマ B の倍周補正クロックに設定します。	0x04	
TCRA	TIA1 への入力信号をイベントとしてタイマ B に出力します。	0x10	
TIOR1A	TIA1 への入力信号の立ち上がりエッジを検出する。	0x0004	
TSRA	インプットキャプチャフラグをクリアします。	0x00	
TIERA	TIA00 のインプットキャプチャ割り込みを許可します。	0x01	
TIORB	TCNTB2 のリロード値に ICRB0 の値を使用、外部イベントの入力を許可、そして、TCNTB3=TCNTB4 の時に TCNTB4 のカウンタを停止しないよう設定します。	0x21	
PIMR	外部入力周期に対する倍周クロックの通倍比を 4 に設定します。	0x0004	
OCRB6	TCNTB6 に対応するコンペアマッチの値を 6 に設定	0x00006000	
LDB	LDB の値を 0x7D0 に設定します。	0x000007D0	
TCRB	TCNTB0,TCNTB2 の使用クロックをクロックバス 0 に設定 します。	0x00	
TCCLRB	TCNTB5 の上限値を 0x28(= PIMR × 設定する外部入力数)に 設定します。	0x00028000	
TIERB	コンペアマッチ B6 の割り込み要求を許可します。	0x08	
TCNTB5	倍周補正クロック生成カウンタ B5 を 0 に設定します。	0x00000000	
TCNTB6	倍周クロックカウンタ B6 の初期値を 0 に設定します。	0x00000000	
TCRD0	TCNT1Dn、DCNTD0n が、それぞれクロックバス 5 でアップ/ダウンカウントを行うよう設定します。	0x1555	
DCRD0	コンペアマッチ A 発生時、ダウンカウント開始します。	0x0222	
TSRD0	コンペアマッチフラグをクリアします。	0x0000	
OCRD00 ~02	オフセット値を設定します。	-	
TIERD	割り込みの許可をします。	0x0010	
TSTRD	タイマ D のサブブロック 0 のカウンタをスタートします。	0x01	
TIOR1D0	OCRD00、01 のコンペアマッチを許可します。	0x153F	
IPR06	タイマ A に関する割り込みの優先度を設定します。	0xF000	
IPR10	タイマ D に関する割り込みの優先度を設定します。	0xF000	
IPR07	タイマ B に関する割り込みの優先度を設定します。	0x0F00	

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

TSRB	タイマ B のコンペアマッチフラグをクリアします。	0x00	パルス出力ルーチン
TSRA	タイマ A のコンペアマッチフラグをクリアします。	0x00	
TIERA	TIA01 のインプットキャプチャ割り込みを許可します。	0x02	
TCNTB3	補正イベントカウンタ値を 0 に設定します。	0x00000000	
TIORB	TCNTB2 のロードデータ、RLDB のロードデータの計算に LDB 値を用いるよう設定します。	LDSEL=0x1	
DCNTD00~02	ダウンカウンタをセットします。	0x0800	フラグクリアルーチン
TSRD0	コンペアマッチ、アンダーフローフラグをクリアします。	0x0000	
TSRA	インプットキャプチャフラグをクリアします。	0x00	外部入力検知ルーチン
TIERA	TIA のインプットキャプチャ割り込みを禁止します。	0x00	
TIORB	TCNTB2 のロードデータ、RLDB のロードデータの計算に ICRB0 値を用いるよう設定します。	LDSEL=0x0	

2.29.5 サンプルプログラム

```

/*****
* DISCLAIMER
* This software is supplied by Renesas Electronics Corporation and is only
* intended for use with Renesas products. No other uses are authorized. This
* software is owned by Renesas Electronics Corporation and is protected under
* all applicable laws, including copyright laws.
* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES REGARDING
* THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT
* LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
* AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY DISCLAIMED.
* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR
* ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS AFFILIATES HAVE
* BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
* Renesas reserves the right, without notice, to make changes to this software
* and to discontinue the availability of this software. By using this software,
* you agree to the additional terms and conditions found by accessing the
* following link:
* http://www.renesas.com/disclaimer *
* Copyright (C) 2011 Renesas Electronics Corporation. All rights reserved.
*****/
/*****
* File Name      : SH7254R_ATU.c
* Version        : 1.00
* Device(s)      : SH72546R
* Tool-Chain     : High-performance Embedded Workshop (Ver.4.09.00.007).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 : (Ver.9.04 Release00).
* OS             : None
* H/W Platform   : SH7254R
* Description    : This is the main tutorial code.
* Operation      : Multiple correction clock signal generation.
*****/
/*****
* History : DD.MM.YYYY Version Description
*          : 12.01.2012 1.00 First Release
*****/

/*****
Includes <System Includes> , "Project Includes"
*****/
#include <machine.h>          /* ライブラリ関数用ヘッダファイル */
#include "iodefine.h"        /* 周辺レジスタ定義ヘッダファイル */

/*****
Macro definitions
*****/
#define ofset0 0x0800        /* オフセット */
#define ofset1 0x0F00        /* オフセット */
#define ofset2 0x1400        /* オフセット */
#define puls_w 0x0800        /* パルス幅 */

/*****
Private global variables and functions
*****/

```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
void main(void); /* メインルーチン */

/*****
* Function Name : main
* Description : The main loop
* Arguments : none
* Return Value : none
*****/
void main(void)
{
    /* ポートの設定 */
    /* Configure PECR1
b15:14 PE7MD = 0 PE7 入出力 (ポート)
b13:12 PE6MD = 0 PE6 入出力 (ポート)
b11 Reserved
b10 PE5MD = 0 PE5 入出力 (ポート)
b9 Reserved
b8 PE4MD = 0 PE4 入出力 (ポート)
b7 Reserved
b6 PE3MD = 0 PE3 入出力 (ポート)
b5:4 PE2MD = 0 PE2 入出力 (ポート)
b3:2 PE1MD = 01 TIA01 入力 (ATU-III)
b1 Reserved
b0 PE0MD = 0 PE0 入出力 (ポート) */
    PORTE.CR1.WORD = 0x0004; /* PE1 を TIA01 入力に設定
    */

    /* Configure PFCR1
b15:14 PF7MD = 00 PF7 モード = PF7 入出力 (ポート)
b13:12 PF6MD = 00 PF6 モード = PF6 入出力 (ポート)
b11:10 PF5MD = 00 PF5 モード = PF5 入出力 (ポート)
b9:8 PF4MD = 00 PF4 モード = PF4 入出力 (ポート)
b7:6 PF3MD = 00 PF3 モード = PF3 入出力 (ポート)
b5:4 PF2MD = 01 PF2 モード = TOD22B 出力 (ATU-III)
b3:2 PF1MD = 01 PF1 モード = TOD21B 出力 (ATU-III)
b1:0 PF0MD = 01 PF0 モード = TOD20B 出力 (ATU-III) */
    PORTF.CR1.WORD = 0x0015; /* PF0, 1, 2 を TOD00B, 01B, 02B 出力に設定
    */

    /* ATU-III (A) の設定 */
    /* Configure TCRA
b7 EVOSEL2A = 0 イベント出力 2A セレクト
b6 EVOSEL2B = 0 イベント出力 2B セレクト
b5:3 EVOSEL1 = 010 TIA01 の入力エッジを出力
b2:0 CKSELA = 000 カウントクロックにクロックバス 0 (プリスケアラ 0) を選択 */
    ATUA.TCRA.BYTE = 0x10; /* タイマ A をクロックバス 0 のカウントクロックで動作
    TIA01 の入力をイベントとして、タイマ B に出力 */

    /* Configure TIOR1A
b15:12 Reserved
b11:10 IOA5 = 00 TIA05 のインプットキャプチャ禁止
b9:8 IOA4 = 00 TIA04 のインプットキャプチャ禁止
b7:6 IOA3 = 00 TIA03 のインプットキャプチャ禁止
b5:4 IOA2 = 00 TIA02 のインプットキャプチャ禁止
b3:2 IOA1 = 01 TIA01 TIA の立ち上がりで ICRA にキャプチャ
b1:0 IOA0 = 00 TIA00 のインプットキャプチャ禁止 */
}
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUA.TIOR1A.WORD = 0x0004;          /* TIA01 の立ち上がりエッジでインプットキャプ  
チャ */  
  
/* Configure TSRA  
b7      OVFA = 0      インプットキャプチャフラグをクリア  
b6      Reserved  
b5      ICFA5 = 0     インプットキャプチャフラグをクリア  
b4      ICFA4 = 0     インプットキャプチャフラグをクリア  
b3      ICFA3 = 0     インプットキャプチャフラグをクリア  
b2      ICFA2 = 0     インプットキャプチャフラグをクリア  
b1      ICFA1 = 0     インプットキャプチャフラグをクリア  
b0      ICFA0 = 0     インプットキャプチャフラグをクリア */  
ATUA.TSRA.BYTE &= 0x00;          /* インプットキャプチャフラグをクリア  
*/  
  
/* Configure TIERA  
b7      OVEA = 0     オーバフロー割り込み A 要求の出力を禁止  
b6      Reserved  
b5      ICEA5 = 0     インプットキャプチャ割り込み A5 要求の出力を禁止  
b4      ICEA4 = 0     インプットキャプチャ割り込み A4 要求の出力を禁止  
b3      ICEA3 = 0     インプットキャプチャ割り込み A3 要求の出力を禁止  
b2      ICEA2 = 0     インプットキャプチャ割り込み A2 要求の出力を禁止  
b1      ICEA1 = 0     インプットキャプチャ割り込み A1 要求の出力を禁止  
b0      ICEA0 = 1     インプットキャプチャ割り込み A0 要求の出力を許可      */  
ATUA.TIERA.BYTE = 0x01;          /* TIA00 のインプットキャプチャ割り込みを許可  
*/  
  
/* Configure IPR06  
b15:12 ATU-A(ICIA0,1) = H'F      対応する割り込みの優先順位を 15 に設定  
b11:8  ATU-A(ICIA2,3) = H'0      対応する割り込みの優先順位を 0 に設定  
b7:4  ATU-A(ICIA4,5) = H'0      対応する割り込みの優先順位を 0 に設定  
b3:0  ATU-A(OVIA) = H'0          対応する割り込みの優先順位を 0 に設定      */  
INTC.IPR06.WORD = 0xF000;          /* TIA00 の割り込み (ICIA0) の優先順位を 15 に設定  
*/  
  
/* ATU-III (B) の設定 */  
/* Configure TIORB  
b7      LDSEL = 0     TCNTB2 にロードする値に ICRB0 を用いる  
b6      CTCNTB5 = 0   TCNTB5 のカウントを許可  
b5      EVCNTB = 1    外部イベントの入力を許可  
b4      LDEN = 0      リロードレジスタの更新を許可  
b3      CCS = 0       TCNTB=TCNTB4 のとき、TCNTB4 のカウント動作を停止しない  
b2:1 Reserved  
b0      IOB6 = 1      TCNTB6 と OCB6 とのコンペアマッチを許可*/  
ATUB.TIORB.BYTE = 0x21;          /* TCNTB2 にロードする値を ICRB0 に設定。TCNTB5  
のカウントを許可  
  
外部イベントの入力を許可、リロードレジスタの更新を許可  
TCNTB6 と OCB6 とのコンペアマッチを許可      */  
  
/* Configure PIMR  
b15:12 Reserved  
b11:0  PIM = H'0004 クロックの通倍比を 4 倍に設定 */  
ATUB.PIMR.WORD = 0x0004;          /* 倍周クロックの通倍比を 4 倍に設定      */  
  
/* Configure TCRB  
b7:2 Reserved
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b1:0 CKSELB = 00      カウントソースにクロックバス 0 を選択 */
ATUB.TCRB.BYTE = 0x00;          /* TCNTB0,TCNTB2 をクロックバス 0 のカウントクロック
で動作 */

/* Configure TIERB
b7:6 Reserved
b5:4 IREG = 00      CMFB6 が有効になったタイミングで割り込み要求出力
b3      CMEB6 = 1      コンペアマッチ割り込み B6 要求の出力を許可
b2      CMEB1 = 0      コンペアマッチ割り込み B1 要求の出力を禁止
b1      ICEB0 = 0      インพุットキャプチャ割り込み B0 要求の出力を禁止
b0      CMEB0 = 0      コンペアマッチ割り込み B0 要求の出力を禁止 */
ATUB.TIERB.BYTE = 0x08;          /* TCNTB6,OCRB6 のコンペアマッチ発生のタイミングでの割り込み要
求を許可 */

/* Configure TCNTB
b31:0  入力エッジ間計測カウント値 = 0  */
ATUB.TCNTB6.LONG = 0x00000000;    /* TCNTB6 の初期値を 0 に設定
*/

/* Configure OCRB6
b31:12 OCB6 = H'00006 アウトプットコンペア値を、6 に設定
b11:0  Reserved */
ATUB.OCRB6.LONG = 0x00006000;    /* TCNTB6 に対応するコンペアマッチの値を 6 に設
定 */

/* Configure LDB
b31:24 Reserved
b23:0  LDVAL = 0007D0 タイマロード値 = 2000 */
ATUB.LDB.LONG=0x000007D0;

/* Configure TCNTB
b31:0  入力エッジ間計測カウント値 = 0  */
ATUB.TCNTB5.LONG = 0x00000000;    /* TCNTB5 の初期値を 0 に設定
*/

/* Configure TCCLRB
b31:12 CCLRB = H'28補正カウンタクリア値 B = D'40
b11:0  Reserved */
ATUB.TCCLRB.LONG = 0x00028000;    /* TCCLRB の初期値を 40 に設定
*/

/* ATU-III (D) の設定 */
/* Configure TCRD
b15    Reserved
b14    OBREDn = 0      OSBRDn のインพุットキャプチャ動作を禁止
b13    C2CEDn = 0      タイマ B からの TCNT2Dn カウンタ値クリア要求を禁止
b12    C1CEDn = 1      タイマ B からの TCNT1Dn カウンタ値クリア要求を許可
b11    Reserved
b10:8  CKSEL2Dn = 101 クロックバス 5 で TCNT2Dn をアップカウント
b7     Reserved
b6:4  CKSEL1Dn = 101 クロックバス 5 で TCNT1Dn をアップカウント
b3     Reserved
b2:0  DCSELDn = 101   クロックバス 5 で DCNTDnm をダウンカウント */
ATUD.SUBBLOCKA[0].TCRD.WORD = 0x1555; /* TCNT1Dn をクロックバス 5 でアップカウント
DCNTD0n をクロックバス 5 でダウンカウント */

/* Configure TIOR1D
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b15:14 OSSDn3 = 00    TODn3A 端子出力なし
b13:12 OSSDn2 = 01    コンペアマッチ A 発生時 TODn2A 端子に I/O コントロールビット A で
の設定値を出力
b11:10 OSSDn1 = 01    コンペアマッチ A 発生時 TODn1A 端子に I/O コントロールビット A で
の設定値を出力
b9:8 OSSDn0 = 01     コンペアマッチ A 発生時 TODn0A 端子に I/O コントロールビット A での設
定値を出力
b7:6 IOADn3 = 00     OCRDn3 コンペアマッチ禁止
b5:4 IOADn2 = 11     OCRDn2 コンペアマッチ時、トグル出力
b3:2 IOADn1 = 11     OCRDn1 コンペアマッチ時、トグル出力
b1:0 IOADn0 = 11     OCRDn0 コンペアマッチ時、トグル出力*/
ATUD.SUBBLOCKB[0].TIOR1D.WORD = 0x153F; /* OCRD00 のコンペアマッチ A 発生時、TOD00A
端子から 0 出力 */

/* Configure DCRD
b15    Reserved
b14:12 TRGSELDn3 = 000    D3 ダウンカウントスタート/ストップトリガ選択 : なし
b11    Reserved
b10:8  TRGSELDn2 = 010    D2 ダウンカウントスタート/ストップトリガ選択 : コンペアマッチ
A
b7     Reserved
b6:4  TRGSELDn1 = 010    D1 ダウンカウントスタート/ストップトリガ選択 : コンペアマッチ A
b3     Reserved
b2:0  TRGSELDn0 = 010    D0 ダウンカウントスタート/ストップトリガ選択 : コンペアマッチ A
*/
ATUD.SUBBLOCKB[0].DCRD.WORD = 0x0222; /* コンペアマッチ A 発生時にダウンカウントスター
ト */

/* Configure TSRD
b15:14 Reserved
b13    OVF2Dn = 0        オーバフローフラグをクリア
b12    OVF1Dn = 0        オーバフローフラグをクリア
b11    UDFDn3 = 0        アンダーフローフラグをクリア
b10    UDFDn2 = 0        アンダーフローフラグをクリア
b9     UDFDn1 = 0        アンダーフローフラグをクリア
b8     UDFDn0 = 0        アンダーフローフラグをクリア
b7     CMFADn3 = 0       コンペアマッチフラグをクリア
b6     CMFADn2 = 0       コンペアマッチフラグをクリア
b5     CMFADn1 = 0       コンペアマッチフラグをクリア
b4     CMFADn0 = 0       コンペアマッチフラグをクリア
b3     CMFBDn3 = 0       コンペアマッチフラグをクリア
b2     CMFBDn2 = 0       コンペアマッチフラグをクリア
b1     CMFBDn1 = 0       コンペアマッチフラグをクリア
b0     CMFBDn0 = 0       コンペアマッチフラグをクリア */
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダフローフラグをクリ
ア */

/* Configure OCRD
b31:8   アウトプットコンペアレジスタ設定
b7:0    Reserved */
ATUD.SUBBLOCKB[0].OCRD[0] = (ofset0); /* OCRD00 に TCNT1D0 に対応するコンペアマッチ A
の条件を設定 */
ATUD.SUBBLOCKB[0].OCRD[1] = (ofset1); /* OCRD01 に TCNT1D1 に対応するコンペアマッチ A
の条件を設定 */
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUD.SUBBLOCKB[0].OCRD[2] = (offset2); /* OCRD02 に TCNT1D2 に対応するコンペアマッチ A
の条件を設定 */

/* Configure TIERD
b15:14 Reserved
b13 OVE2Dn = 0 OVF2Dn による割り込み要求を禁止
b12 OVE1Dn = 0 OVF1Dn による割り込み要求を禁止
b11 UDEDn3 = 0 UDFDn3 による割り込み要求を禁止
b10 UDEDn2 = 0 UDFDn2 による割り込み要求を禁止
b9 UDEDn1 = 0 UDFDn1 による割り込み要求を禁止
b8 UDEDn0 = 0 UDFDn0 による割り込み要求を禁止
b7 CMEADn3 = 0 CMFADn3 による割り込み要求を禁止
b6 CMEADn2 = 0 CMFADn2 による割り込み要求を禁止
b5 CMEADn1 = 0 CMFADn1 による割り込み要求を禁止
b4 CMEADn0 = 1 CMFADn0 による割り込み要求を許可
b3 CMEBDn3 = 0 CMFBDn3 による割り込み要求を禁止
b2 CMEBDn2 = 0 CMFBDn2 による割り込み要求を禁止
b1 CMEBDn1 = 0 CMFBDn1 による割り込み要求を禁止
b0 CMEBDn0 = 0 CMFBDn0 による割り込み要求を禁止 */
ATUD.SUBBLOCKB[0].TIERD.WORD = 0x0010; /* OCRD00 のコンペアマッチによる割り込みを
許可 */

/* Configure TSTRC
b7:5 Reserved
b4 STRD4 = 0 TCNTD4 のカウント動作を停止
b3 STRD3 = 0 TCNTD3 のカウント動作を停止
b2 STRD2 = 0 TCNTD2 のカウント動作を停止
b1 STRD1 = 0 TCNTD1 のカウント動作を停止
b0 STRD0 = 1 TCNTD0 のカウント動作を許可 */
ATUD.TSTRD.BYTE = 0x01; /* タイマ D のサブブロック 0 のカウントをスタート
*/

/* ATU-III (共通) の設定 */
/* Configure CBCNT
b7:6 Reserved
b5:4 CB4EG = 0 クロックバス 4:外部クロックのエッジ検出をしない
b3 Reserved
b2 CB5SEL = 1 クロックバス 5:タイマ B 出力の倍周補正クロック
b1:0 CB5EG = 0 クロックバス 5:外部クロックのエッジ検出をしない*/
ATUCTRL.CBCNT.BYTE = 0x04; /* クロックバス 5 をタイマ B からの倍周補正クロック () に
設定 */

/* Configure PSCR0
b15:10 Reserved
b9:0 PSCn = H'9 プリスケアラの分周比を 10 (設定値は-1) に設定 */
ATUCTRL.PSCR0.WORD = 0x0009; /* プリスケアラ 0 の分周比を 10 に設定
*/

/* Configure IPR07
b15:12 ATU-B (CMIB0, CMIB1) = H'0 対応する割り込みの優先順位を 0 に設定
b11:8 ATU-B (CMIB6, ICIB0) = H'F 対応する割り込みの優先順位を 15 に設定
b7:4 ATU-C0 (IMIC0-IMIC3) = H'0 対応する割り込みの優先順位を 0 に設定
b3:0 ATU-C0 (OVIC0) = H'0 対応する割り込みの優先順位を 0 に設定 */
INTC.IPR07.WORD = 0x0F00; /* TCNTB6 のコンペアマッチの割り込み優先順位を 15 に設
定 */
```

```
/* Configure IPR10
b15:12 ATU-D0 (CMID00-CMID03) = H'F   対応する割り込みの優先順位を 15 に設定
b11:8  ATU-D0 (OVI1D0-OVI2D0) = H'0   対応する割り込みの優先順位を 0 に設定
b7:4  ATU-D0 (UDID00-UDID03) = H'0   対応する割り込みの優先順位を 0 に設定
b3:0  ATU-D1 (CMID10-CMID13) = H'0   対応する割り込みの優先順位を 0 に設定*/
INTC.IPR10.WORD = 0xF000;           /* OCRD00 の割り込み優先順位を 15 に設定
*/

set_imask(0);                       /* 割り込みマスクレベルを 0 に設定          */

/* Configure ATUENR
b15:10 Reserved
b9     TJE = 0     タイマ J のカウント動作を停止
b8     THE = 0     タイマ H のカウント動作を停止
b7     TGE = 0     タイマ G のカウント動作を停止
b6     TFE = 0     タイマ F のカウント動作を停止
b5     TEE = 0     タイマ E のカウント動作を停止
b4     TDE = 1     タイマ D のカウント動作を許可
b3     TCE = 0     タイマ C のカウント動作を停止
b2     TBE = 1     タイマ B のカウント動作を許可
b1     TAE = 1     タイマ A のカウント動作を許可
b0     PSCE = 1   プリスケーラのカウント動作を許可 */
ATUCTRL.ATUENR.WORD = 0x0017;     /* タイマ A, B, D およびプリスケーラのカウントをスタート
*/

while(1);
} /* End of function main() */

/*****
*
* Function Name : CMIB6
* Description   : コンペアマッチ割り込み (B6)
* Arguments    : none
* Return Value : none
*****/
/
void CMIB6(void)
{
/* Configure TSRB
b7:4 Reserved
b3     CMFB6 = 0   コンペアマッチフラグをクリア
b2     CMFB1 = 0   コンペアマッチフラグをクリア
b1     ICFB0 = 0   コンペアマッチフラグをクリア
b0     CMFB0 = 0   コンペアマッチフラグをクリア */
ATUB.TSRB.BYTE  = 0;               /* コンペアマッチフラグをクリア          */

/* Configure TSRA
b7     OVFA = &0   オーバフローフラグをクリア
b6     Reserved
b5     ICFA5 = &0   インพุットキャプチャフラグをクリア
b4     ICFA4 = &0   インพุットキャプチャフラグをクリア
b3     ICFA3 = &0   インพุットキャプチャフラグをクリア
b2     ICFA2 = &0   インพุットキャプチャフラグをクリア
b1     ICFA1 = &0   インพุットキャプチャフラグをクリア
b0     ICFA0 = &0   インพุットキャプチャフラグをクリア */
```

SH7254R グループ アドバンストタイマユニット-III (ATU-III) を使用したタイマ動作例

```
ATUA.TSRA.BYTE &= 0x00;          /* インพุットキャプチャフラグをクリア
*/

/* Configure TIERA
b7      OVEA = 0      オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0     インพุットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0     インพุットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0     インพุットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0     インพุットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 1     インพุットキャプチャ割り込み A1 要求の出力を許可
b0      ICEA0 = 0     インพุットキャプチャ割り込み A0 要求の出力を禁止 */
ATUA.TIERA.BYTE = 0x02;          /* TIA01 のインพุットキャプチャ割り込みを許可
*/

/* Configure TCNTB
b31:0   入力エッジ間計測カウンタ値 = 0 */
ATUB.TCNTB3.LONG = 0x00000000;

/* Configure TIORB
b7      LDSEL = 1     TCNTB2 のロードデータ、RLDB のロードデータの計算に LDB 値を用いる
*/
ATUB.TIORB.BIT.LDSEL=0x1;

/* Configure DCNTD
b31:8   タイマダウンカウンタ値
b7:0   Reserved */
ATUD.SUBBLOCKB[0].DCNTD[0] = puls_w; /* ダウンカウンタ 00 を 4 にセット
*/
ATUD.SUBBLOCKB[0].DCNTD[1] = puls_w; /* ダウンカウンタ 01 を 4 にセット
*/
ATUD.SUBBLOCKB[0].DCNTD[2] = puls_w; /* ダウンカウンタ 02 を 4 にセット
*/
} /* End of function CMIB6() */

/*****
* Function Name : CMID00
* Description    : コンペアマッチ割り込み (D0)
* Arguments     : none
* Return Value  : none
*****/
void CMID00(void)
{
    /* Configure TSRD
b15:14 Reserved
b13     OVF2Dn = 0     オーバフローフラグをクリア
b12     OVF1Dn = 0     オーバフローフラグをクリア
b11     UDFDn3 = 0     アンダーフローフラグをクリア
b10     UDFDn2 = 0     アンダーフローフラグをクリア
b9      UDFDn1 = 0     アンダーフローフラグをクリア
b8      UDFDn0 = 0     アンダーフローフラグをクリア
b7      CMFADn3 = 0     コンペアマッチフラグをクリア
b6      CMFADn2 = 0     コンペアマッチフラグをクリア
b5      CMFADn1 = 0     コンペアマッチフラグをクリア
b4      CMFADn0 = 0     コンペアマッチフラグをクリア
b3      CMFBDn3 = 0     コンペアマッチフラグをクリア
```

SH7254R グループ アドバンスドタイマユニット-III (ATU-III) を使用したタイマ動作例

```
b2      CMFBDn2 = 0      コンペアマッチフラグをクリア
b1      CMFBDn1 = 0      コンペアマッチフラグをクリア
b0      CMFBDn0 = 0      コンペアマッチフラグをクリア */
ATUD.SUBBLOCKB[0].TSRD.WORD &= 0x0000; /* コンペアマッチ、アンダフローフラグをク
リ
ア */
} /* End of function CMID00() */

/*****
* Function Name : ICIA1
* Description   : インพุットキャプチャ割り込み
* Arguments     : none
* Return Value  : none
*****/
void ICIA1(void)
{
    /* Configure TSRA
b7      OVFA = 0      オーバフローフラグをクリア
b6      Reserved
b5      ICFA5 = 0      インพุットキャプチャフラグをクリア
b4      ICFA4 = 0      インพุットキャプチャフラグをクリア
b3      ICFA3 = 0      インพุットキャプチャフラグをクリア
b2      ICFA2 = 0      インพุットキャプチャフラグをクリア
b1      ICFA1 = 0      インพุットキャプチャフラグをクリア
b0      ICFA0 = 0      インพุットキャプチャフラグをクリア*/
ATUA.TSRA.BYTE &= 0x00; /* インพุットキャプチャフラグをクリア */

    /* Configure TIERA
b7      OVEA = 0      オーバフロー割り込み A 要求の出力を禁止
b6      Reserved
b5      ICEA5 = 0      インพุットキャプチャ割り込み A5 要求の出力を禁止
b4      ICEA4 = 0      インพุットキャプチャ割り込み A4 要求の出力を禁止
b3      ICEA3 = 0      インพุットキャプチャ割り込み A3 要求の出力を禁止
b2      ICEA2 = 0      インพุットキャプチャ割り込み A2 要求の出力を禁止
b1      ICEA1 = 0      インพุットキャプチャ割り込み A1 要求の出力を禁止
b0      ICEA0 = 0      インพุットキャプチャ割り込み A0 要求の出力を禁止 */
ATUA.TIERA.BYTE = 0x00; /* TIA01 のインพุットキャプチャ割り込みを禁止 */

    /* Configure TIORB
b7      LDSEL = 0      TCNTB2 のロードデータ、RLDB のロードデータの計算に ICRB0 値を用いる
*/
ATUB.TIORB.BIT.LDSEL=0x0;
} /* End of function ICIA1() */
```

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
<http://japan.renesas.com/>
- お問い合わせ先
<http://japan.renesas.com/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 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・システムの設計において、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て、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。
2. 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、誤りがな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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
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、産業用ロボット等
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、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、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いません。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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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。
9. 本資料に記載されている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・システムに使用することはできません。また、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情報に使用しないで行ってください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>