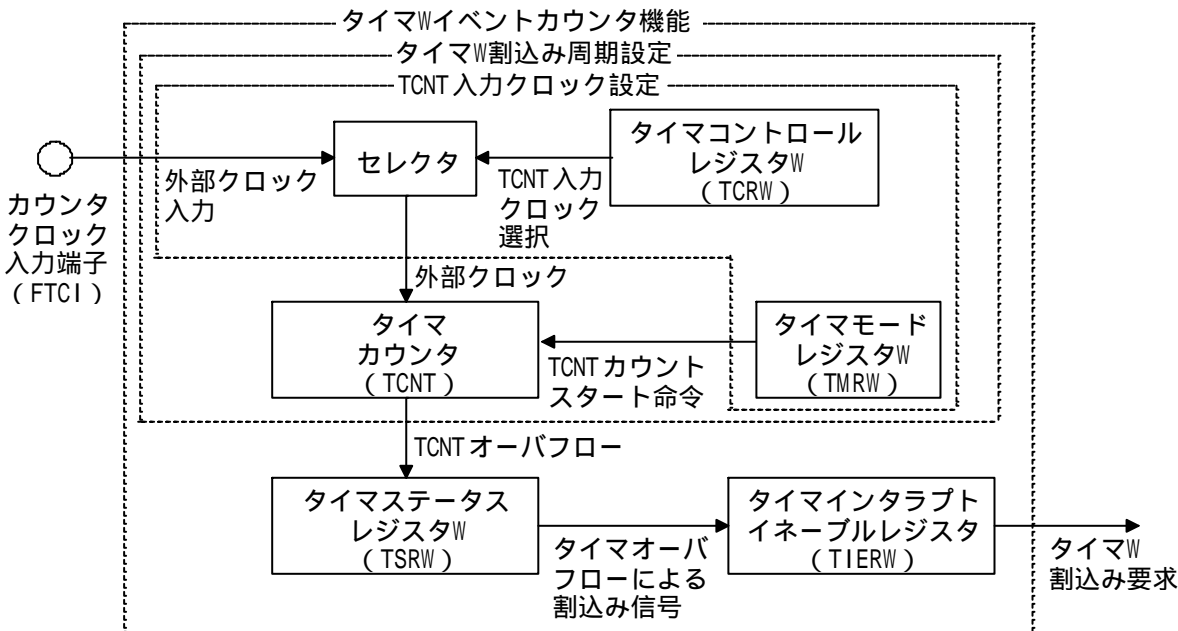


## 2.7 16ビットイベントカウンタ機能による入力パルス数の測定

16ビットイベントカウンタ機能による入力パルス数の測定	使用機能	タイマW : 16ビットイベントカウンタ機能
仕様	<p>(1) タイマWの16ビットイベントカウンタ機能を使用して、カウンタクロック入力端子(FTCI)に入力されたパルスの立ち上がりエッジをカウントします。</p> <p>(2) タイマカウンタ (TCNT) を外部クロックの立ち上がりエッジでカウントするように設定し、1024回の立ち上がりエッジをカウントするまでカウントアップを行ないます。</p> <p>(3) 1024回の立ち上がりエッジをカウントすると、TCNTの外部クロック入力を止め、終了します。</p>	
使用機能説明	<p>(1) 本タスク例では、タイマWイベントカウンタ機能を使用して、FTCI入力端子に入力したパルスのパルス数の測定を行ないます。</p> <p>(a) 図1にタイマWフリーランニング機能のブロック図を示します。以下にタイマWイベントカウンタ機能のブロック図について説明します。</p> <ul style="list-style-type: none"> <li>・タイマカウンタ (TCNT) は、16ビットのリード/ライト可能なアップカウンタで、入力する内部クロック/外部クロックによりカウントアップされます。入力するクロックはシステムクロックの2分周、4分周、8分周および外部クロックの計4種類のクロックより選択可能です。本タスク例では、TCNTの入力クロックに外部クロックを選択しています。</li> <li>・タイマコントロールレジスタW (TCRW) は、8ビットのリード/ライト可能なレジスタで、TCNTの入力クロックの選択を行ないます。</li> <li>・タイマステータスレジスタW (TSRW) は、8ビットのレジスタで、TCNTの割込み要求信号の制御を行ないます。</li> <li>・タイマインタラプトイネーブルレジスタW (TIERW) は、8ビットのリード/ライト可能なレジスタで、各割込み要求の許可/禁止を制御します。本タスク例では、タイマオーバーフローによる割込み要求を許可し、それ以外の割込みは禁止します。</li> <li>・カウンタクロック入力端子 (FTCI) より、外部クロックを入力します。</li> <li>・タイマモードレジスタW (TMRW) は TCNTのカウンタスタート命令を行ないます。</li> </ul>  <p style="text-align: center;">図1 イベントカウンタ機能ブロック図</p>	

使用機能説明

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

表1 機能割付け

機能	機能割付け
TCNT	外部クロックを入力とする16ビットのカウンタ
TCRW	TCNT入力クロックの設定
TSRW	タイマオーバフローによる割込み要求信号の制御
TIERW	タイマオーバフローによる割込み要求の許可
TMRW	TCNTカウントスタート設定
FTCI	パルスの入力端子

動作説明

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

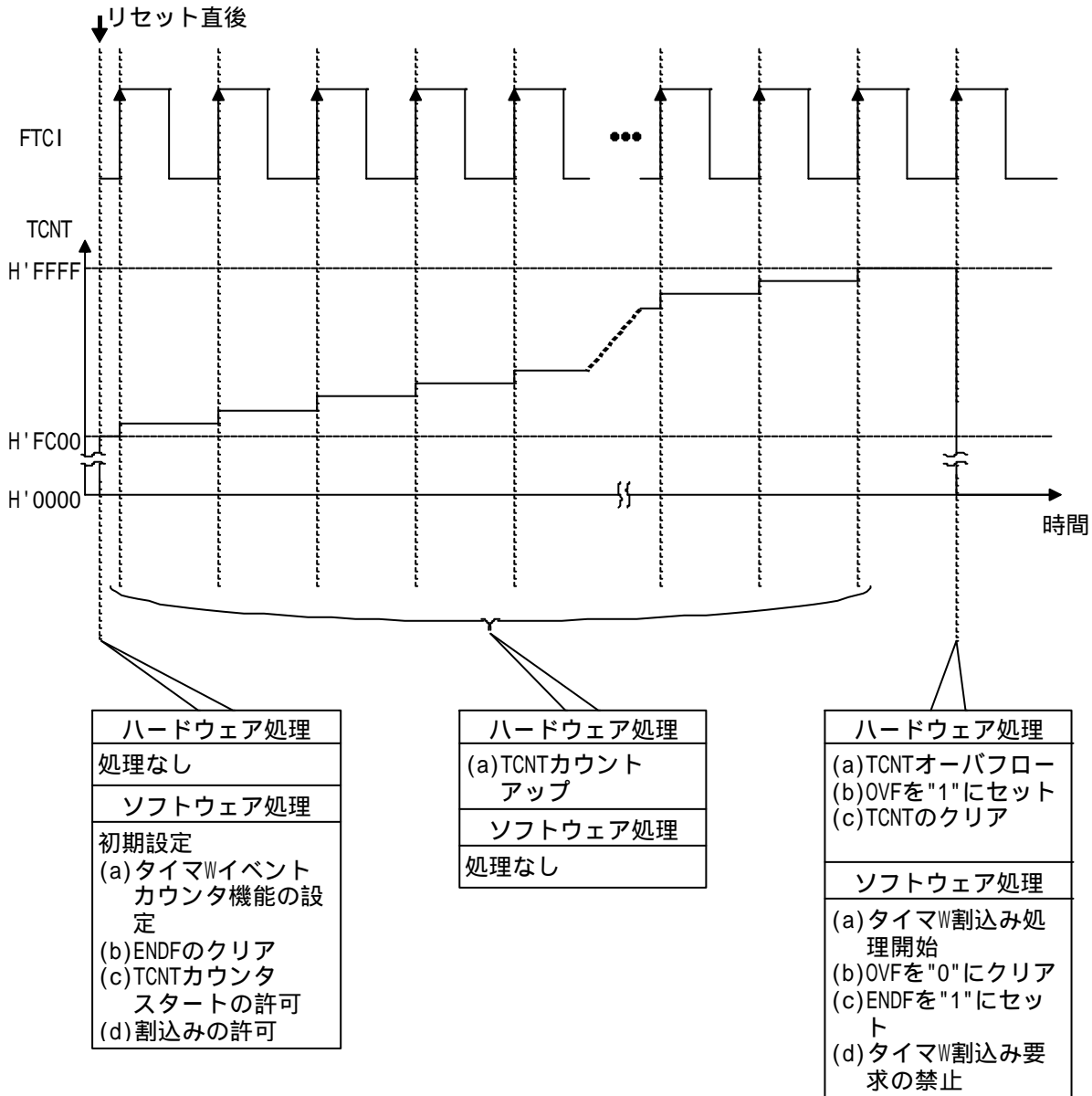


図2 タイマW16ビットイベントカウンタ機能による入力パルス数の測定動作原理

## ソフトウェア説明

## (1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

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

## (2) 引数の説明

本タスク例では、引数は使用していません。

## (3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値
TMRW CTS	タイマモードレジスタW (タイマカウンタスタート) : CTS="1"のとき、TCNTがカウンタ開始を示す : CTS="0"のとき、TCNTがカウンタ停止を示す	H'FF80 ビット7	1
TCRW CKS2 CKS1 CKS0	タイマコントロールレジスタW (クロックセレクト1、0) : CLS2="1"、CLS1="×"、CLS0="×"のとき、TCNT入力クロックを外部クロックに設定  注意："×"は1、0どちらでも良い	H'FF81 ビット6 ビット5 ビット4	CKS2="1" CKS1="×" CKS0="×"
TIERW OVIE	タイマインタラプトイネーブルレジスタW (タイマオーバーフロー割込みイネーブル) : OVIE="0"のとき、OVFによる割込み要求を禁止 : OVIE="1"のとき、OVFによる割込み要求を許可	H'FF82 ビット7	1
TSRW OVF	タイマコントロール/ステータスレジスタW (タイマオーバーフロー) : OVF="0"のとき、FRCがオーバーフローしていないことを示す : OVF="1"のとき、FRCがオーバーフローしたことを示す	H'FF83 ビット7	0
TCNT	タイマカウンタ : システムクロックの8分周のクロックを入力とする16ビットのアップカウンタ	H'FF86	H'00

## (4) 使用RAM説明

表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

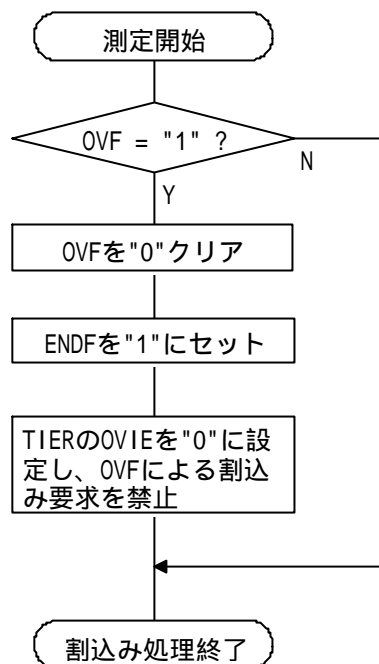
ラベル名	機能	アドレス	使用モジュール名
USRFB ENDF	入力パルスの立ち上がりエッジを1024回検出したことを示すフラグ	H'FB80 ビット0	メインルーチン 割込みカウント

## フローチャート

(a) メインルーチン



(b) タイマW割込み処理ルーチン



本例ではスタックポインタはINIT.SRC (アセンブリ言語) で設定してあります。

## プログラムリスト

## INIT.SRC (プログラムリスト)

```

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

/*****/
/* */
/* H8/300H Tiny Series -H8/3664- */
/* Application Note */
/* */
/* 'Measurement of Input Pulses by 16-bit */
/* Event Counter Function' */
/* */
/* Function */
/* : Timer W 16bit Event Counter */
/* */
/* External Clock : 16MHz */
/* Internal Clock : 16MHz */
/* Sub Clock : 32.768kHz */
/* */
/*****/

#include <machine.h>

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

#define TMRW *(volatile unsigned char *)0xFF80 /* Timer Mode Register W */
#define TCRW *(volatile unsigned char *)0xFF81 /* Timer Control Register W */
#define TCRW_BIT (*(struct BIT *)0xFF81) /* Timer Control Register W */
#define CKS1 TCRW_BIT.b5 /* Clock Select 1 */
#define CKS0 TCRW_BIT.b4 /* Clock Select 0 */
#define TIERW *(volatile unsigned char *)0xFF82 /* Timer Interrupt Enable Register */
#define TIERW_BIT (*(struct BIT *)0xFF82) /* Timer Interrupt Enable Register */
#define OVIE TIERW_BIT.b7 /* Timer Overflow Interrupt Enable */
#define TSRW *(volatile unsigned char *)0xFF83 /* Timer Status Register W */
#define TSRW_BIT (*(struct BIT *)0xFF83) /* Timer Status Register W */
#define OVF TSRW_BIT.b7 /* Timer Over flow */
#define TCNT *(volatile unsigned int *)0xFF86 /* Time Counter */
#define PCR8_BIT (*(struct BIT *)0xFFEB) /* Port Control Register 8 */
#define PCR80 PCR8_BIT.b0 /* FTCL Select 0 */

#pragma interrupt (twint)
/*****/
/* 関数定義 */
/*****/
extern void INIT ( void ); /* SP Set */
void main ( void );
void twint ( void );

```

## プログラムリスト

```

/*****
/*   RAM define
/*****

    unsigned char   USRF;                                /* User Flag Erea */

#define   USRF_BIT   (*(struct BIT *)&USRF)
#define   ENDF       USRF_BIT.b0                        /* End Flag */

/*****
/*   Vector Address
/*****
#pragma section V1                                     /* VECTOR SECTOIN SET */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
    INIT                                                /* 00 Reset */
};
#pragma section V2                                     /* VECTOR SECTOIN SET */
void (*const VEC_TBL2[])(void) = {
    twint                                               /* 2A Timer W Interrupt */
};
#pragma section P                                     /* P */
/*****
/*   Main Program
/*****
void main ( void )
{
    set_imask_ccr(1);                                  /* Interrupt Disable */

    PCR80 = 0;                                        /* Initialize FTCL Input Terminal */

    TCRW = 0x40;                                       /* Initialize TCNT Input Clock */

    TSRW = 0x70;                                       /* Clear OVF */

    TIERW = 0xF0;                                       /* OVF Interrupt Enable */

    TCNT = 0xFC00;                                       /* Initialize FRC(Timer Counter) */

    TMRW = 0xC8;                                       /* Timer Counter Count Start */

    ENDF = 0;                                           /* Initialize ENDF */

    set_imask_ccr(0);                                   /* Interrupt Enable */

    while(ENDF != 1){
        ;
    }

    TCRW = 0x10;                                       /* Initialize TCNT Input Clock */

    while(1){
        ;
    }
}

```

## プログラムリスト

```
/*
*****
/*   Timer W Interrupt
*****
void twint ( void )
{
    if ( OVF == 1 ){
        OVF = 0;           /* Clear OVF          */
        ENDF = 1;         /* Set ENDF           */
        OVIE = 0;         /* OVF Interrupt Disable */
    }
}
```

## リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'002A
P	H'0100
B	H'FB80