

## 2.21 内部割込みによる多重割込み動作

内部割込みによる多重割込み動作	使用機能	内部割込み
<p>仕様</p> <p>(1) タイマA、およびタイマV割込みを使用して、多重割込み処理を行いません。  (2) タイマA割込み処理の中で、タイマV割込み要求を受け付けることにより、ソフトウェアでタイマV割込み要求の優先順位をタイマA割込み要求の優先順位より高くなるように設定します。  (3) タイマA割込み要求は、インターバル機能により32.768msごとに発生するように設定します。  (4) タイマV割込み要求は、インターバル機能により2.048msごとに発生するように設定します。  (5) LEDはポート7のP74出力端子に接続されているものとします。</p>		
<p>使用機能説明</p> <p>(1) 本タスク例では、内部割込みを使用してタイマAおよびタイマV割込みの多重割込み動作を行いません。</p> <p>(a) 以下に内部割込みについて説明します。</p> <ul style="list-style-type: none"> <li>・ 各内蔵周辺モジュールには割込み要求ステータスフラグとこれらの割込みイネーブルビットがあります。タイマA割込み要求とSLEEP命令実行によって発生する直接遷移割込みについてはこの機能はIRR1、IENR1に含まれています。内蔵周辺モジュールからの割込み要求が発生すると対応する割込み要求ステータスフラグが"1"にセットされCPU に対して割込みを要求します。これらの割込み要求は対応するイネーブルビットを0 にクリアすることにより禁止できます。</li> <li>・ CCRのIビットを"1"にセットすることにより、すべての割込みをマスクできます。</li> <li>・ 以下に割込みの動作を示します。 <ol style="list-style-type: none"> <li>(1) 割込みイネーブルレジスタ1の対応するビットが"1"にセットされている状態で、割込み要因が発生したとき、割込みコントローラに対して割込み要求信号が送られます。</li> <li>(2) 割込みコントローラに割込み要求信号が送られると、割込み要求フラグが"1"にセットされます。</li> <li>(3) 割込み要求フラグが"1"にセットされている割込みの中で、優先順位に従って最高位の割込み要求が選択され、その他は保留となります。</li> <li>(4) CCRのIビットを参照し、Iビットが"0"にクリアされている場合は、割込み要求は受け付けられませんが、Iビットが"1"にセットされている場合は割込み要求は保留となります。</li> <li>(5) 割込みが受け付けられると、そのとき実行中の命令の処理が終了した後、プログラムカウンタ(PC)とCCRがスタック領域に退避されます。スタックされるPCは、リターン後に実行する最初のアドレスを示しています。</li> <li>(6) CCRのIビットが"1"にセットされます。これにより、すべての割込みはマスクされます。</li> <li>(7) 受け付けた割込みに対応するベクタアドレスを生成し、そのアドレスの内容によって示されるアドレスから、割込み処理ルーチンの実行を開始します。</li> </ol> </li> <li>・ 割込みイネーブルレジスタ1を"0"にクリアすることにより割込みを禁止にする場合、または割込みフラグレジスタ1をクリアする場合は、必ず割込みをマスクした状態(I="1")で行いません。I="0"の状態で行なうと、当該操作命令の実行と当該割込みの発生が競合した場合に、当該操作命令の実行終了時に発生した割込みに対応する例外処理を実行します。</li> <li>・ 以下に本例におけるタイマA割込み周期とタイマV割込み周期の計算方法を示します。</li> </ul> $\text{タイマA割込み周期} = \frac{1}{\text{システムクロック} / 2048\text{分周}} \times 256$ $32.768\text{ms} = \frac{1}{16\text{MHz} / 2048\text{分周}} \times 256$ $\text{タイマV割込み周期} = 2 \times \frac{1}{\text{システムクロック} / 64\text{分周}} \times 256$ $2.048\text{ms} = 2 \times \frac{1}{16\text{MHz} / 64\text{分周}} \times 256$		

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

使用機能説明

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

表1 機能割付け

機能	機能割付け
OVIE	TCRSRVのOVFによる割込み要求の許可を行なう
IENTA	タイマA割込み要求の許可を行なう
OVF	TCNTVのオーバーフロー割込み要求の有無を反映
IRRTA	タイマA割込み要求の有無を反映
CCR 1ビット	すべての割込み要求の許可/禁止を行なう
P74	LED出力

動作説明

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

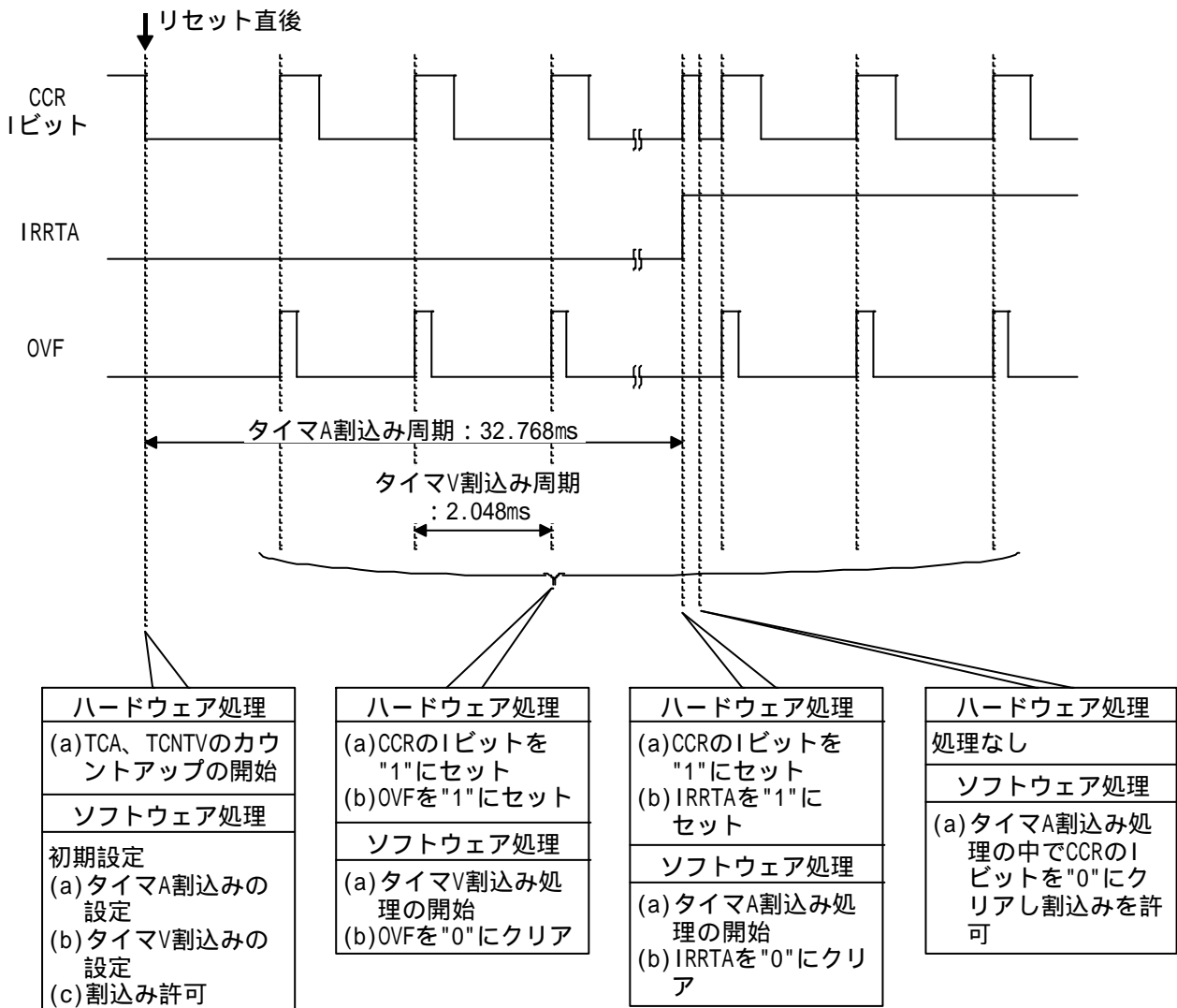


図1 内部割込みによる多重割込み動作の動作原理

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

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	タイマA割込みの設定、タイマV割込みの設定、ポート7の設定、割込みの許可を行なう
カウント	taint	タイマA割込み処理ルーチンで、割込みの許可、16ビットカウンタをインクリメントし、H'4000になったら終了
LED制御	tvint	タイマV割込み処理ルーチンで、LEDの点灯/消灯を行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3に本タスク例における使用内部レジスタ説明を示します。

表3 使用内部レジスタ説明

レジスタ名	機能	アドレス	設定値	
TCRV0	OVIE	タイマコントロールレジスタV0 (タイマオーバフローインタラプトイネーブル) : OVIE="1"のときTCNTVのオーバフロー割込みを許可	H'FFA0 ビット5	0
	CKS2 CKS1 CKS0	タイマコントロールレジスタV0 (クロックセレクト2~0) : CKS2="0"、CKS1="1"、CKS0="1"、ICKS0="0"の時、TCNTVは システムクロックの64分周のクロックの立ち上がりエッジでカウント	H'FFA0 ビット2 ビット1 ビット0	CKS2="0" CKS1="1" CKS0="1"
TCSR	OVF	タイマコントロール/ステータスレジスタV (タイマオーバフラグ) : OVF="0"のときTCNTVのオーバフロー割込みの要求されていない : OVF="1"のときTCNTVのオーバフロー割込みの要求されている	H'FFA1 ビット5	0
TCNTV		タイマカウンタV : システムクロックの64分周のクロックを入力とする8ビットのアップ カウンタ	H'FFA4	H'00
TCRV1	ICKS0	タイマコントロールレジスタV1 (インターナルクロックセレクト0) : TCRV0のCK2~CK0との組み合わせで、TCNTVに入力するクロックを選択する。	H'FFA5 ビット0	0
TMA		タイマモードレジスタA : TMA=H'12のとき、タイマA機能をインターバル機能に、TCA入力ク ロックソースをPSSに、プリスケラ分周比を2048分周に設定	H'FFA6	H'12
TCA		タイマカウンタA : システムクロックを2048分周したクロックを入力とする8ビットの アップカウンタ	H'FFA7	H'00
PDR7	P7 <sub>4</sub>	ポートデータレジスタ7 (ポートデータレジスタ7 <sub>4</sub> ) : P7 <sub>4</sub> =0のとき、P7 <sub>4</sub> 端子の出力レベルは"Low" : P7 <sub>4</sub> =1のとき、P7 <sub>4</sub> 端子の出力レベルは"High"	H'FFDA ビット4	0
PCR7	PCR7 <sub>4</sub>	ポートコントロールレジスタ7 (ポートコントロールレジスタ7 <sub>4</sub> ) : PCR7 <sub>4</sub> =1"のとき、P7 <sub>4</sub> 入出力端子は、P7 <sub>4</sub> 出力端子として機能	H'FFEA ビット4	1
IENR1	IENTA	割込みイネーブルレジスタ1 (タイマA割込み要求フラグ) : IENTA="1"のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
IRR1	IRRRTA	割込みフラグレジスタ1 (タイマA割込み要求フラグ) : IRRRTA="0"のとき、タイマA割込みが要求されていない : IRRRTA="1"のとき、タイマA割込みが要求されている	H'FFF6 ビット6	0

(4) 使用RAM説明

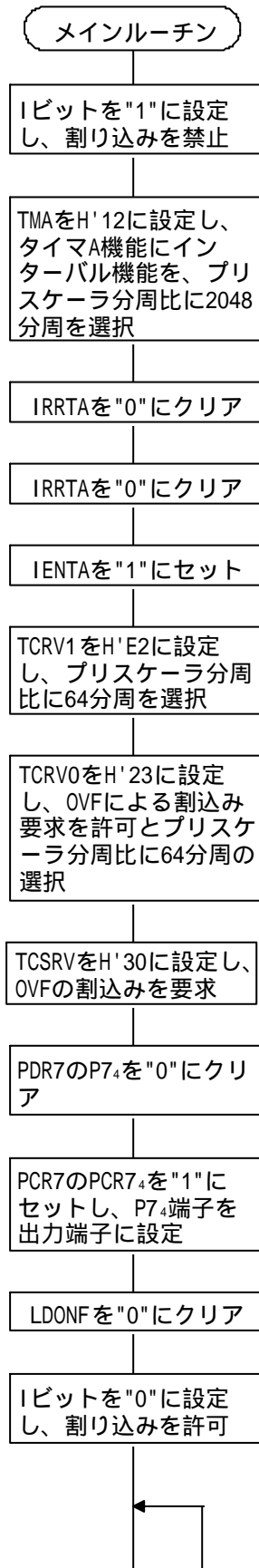
表4に本タスク例で使用するRAMの説明を示します。

表4 使用RAM説明

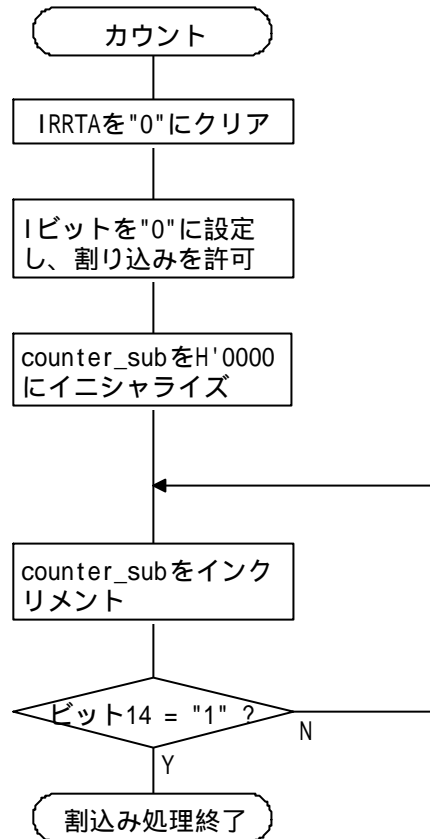
ラベル名	機能	アドレス	使用モジュール名
counter_sub	割込みルーチン (タイマA) の中でH'4000までカウント する16ビットアップカウンタ	H'FB80	割込みルーチン タイマA
USRF	LDONF	H'FB82 ビット0	LED制御

## フローチャート

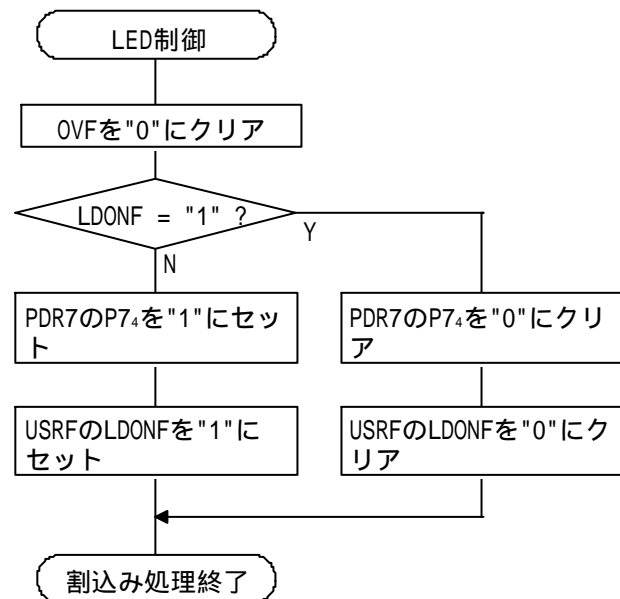
(a) メインルーチン



(b) タイマA割込み処理ルーチン



(c) タイマV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                  */
/*                                     */
/*   'Multiple Interrupt Operation by Internal */
/*     Interrupt'                       */
/*                                     */
/*   Function                           */
/*   : Internal Interrupt               */
/*                                     */
/*   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 TCRV0    *(volatile unsigned char *)0xFFA0 /* Timer Control Register V0 */
#define TCSRVS  *(volatile unsigned char *)0xFFA1 /* Timer Control/Status Register V */
#define TCSRVB  (*(struct BIT *)0xFFA1)          /* Timer Control/Status Register V */
#define OVF      TCSRVB.b5                       /* Timer Overflow Flag */
#define TCNTV    *(volatile unsigned char *)0xFFA4 /* Timer Counter V */
#define TCRV1    *(volatile unsigned char *)0xFFA5 /* Timer Control Register V1 */
#define TMA      *(volatile unsigned char *)0xFFA6 /* Timer Mode Register A */
#define TCA      *(volatile unsigned char *)0xFFA7 /* Timer Counter A */
#define PDR7_BIT (*(struct BIT *)0xFFDA)          /* Port Data Register 7 */
#define P74      PDR7_BIT.b4                     /* Port Data Register 7 bit4 */
#define PCR7_BIT (*(struct BIT *)0xFFEA)          /* Port Control Register 7 */
#define PCR74    PCR7_BIT.b4                     /* Port Control Register 7 bit4 */
#define IENR1_BIT (*(struct BIT *)0xFFE4)         /* Interrupt Enable Register 1 */
#define IENTA    IENR1_BIT.b6                    /* Timer A Interrupt Enable */
#define IENO      IENR1_BIT.b0                   /* IRQ0 Interrupt Enable */
#define IRR1_BIT (*(struct BIT *)0xFFE6)         /* Interrupt Request Register 1 */
#define IRRTA    IRR1_BIT.b6                    /* Timer A Interrupt Request Flag */
#define IRR10    IRR1_BIT.b0                    /* IRQ0 Interrupt Request Flag */
#define PMR1_BIT (*(struct BIT *)0xFFE0)         /* Port Mode Register 1 */
#define IRQO_SET PMR1_BIT.b4                     /* Port Mode Register 1 bit4 */
#define TMOW     PMR1_BIT.b0                     /* P10/TMOW Terminal Function Change */
#define IEGR1_BIT (*(struct BIT *)0xFFE2)        /* Interrupt Edge Select Register 1 */
#define IEGR1    IEGR1_BIT.b0                   /* IRQ0 Edge Select */

```

内部割込みによる多重割込み動作	使用機能	内部割込み
プログラムリスト		
<pre> #pragma      interrupt    (taint) #pragma      interrupt    (tvint) /***** /*   関数定義 *****/ extern void  INIT ( void );          /* SP Set          */ void      main ( void ); void      taint ( void ); void      tvint ( void );  /***** /*   RAM define *****/ unsigned int  counter_sub; unsigned char  USRF;                /* User Flag Erea */  #define USRF_BIT  (*(struct BIT *)&amp;USRF) #define LDONF     USRF_BIT.b0      /* LED On Flag     */  /***** /*   Vector Address *****/ #pragma section      V1          /* VECTOR SECTOIN SET */ void (*const VEC_TBL1[])(void) = {     INIT                /* 00 Reset          */ }; #pragma section      V2          /* VECTOR SECTOIN SET */ void (*const VEC_TBL2[])(void) = {     taint              /* 26 Timer A Interrupt */ }; #pragma section      V3          /* VECTOR SECTOIN SET */ void (*const VEC_TBL3[])(void) = {     tvint              /* 2C Timer V Interrupt */ };  #pragma section          /* P          */ /***** /*   Main Program *****/ void  main ( void ) {      set_imask_ccr(1);          /* Interrupt Disable */      TMA = 0x12;                /* Initialize TMA Function &amp; TCA Input Clock Period */     IRRTA = 0;                 /* Clear IRRTA          */     IENTA = 1;                 /* Timer A Interrupt Enable */      TCRV1 = 0xE2;              /* Initialize Time Control Register V1 */     TCRV0 = 0x23;              /* Initialize TCRV0 Function &amp; TCRV0 Input Clock Period */     TCSRv = 0x30;              /* Initialize Timer Control/Status Register V */      P74 = 0;                   /* Clear P74          */     PCR74 = 1;                 /* Initialize P74 Output Terminal */      LDONF = 0;                 /* Clear LDONF        */      set_imask_ccr(0);          /* Interrupt Enable    */      while(1){         ;     } } </pre>		

## プログラムリスト

```

/*****
/*   Timer A Interrupt
*****/
void taint ( void )
{

    IRRTA = 0;                /* Clear IRRTA          */

    set_imask_ccr(0);        /* Interrupt Enable    */

    counter_sub = 0x0000;    /* Initialize 16bit Counter */

    do{
        counter_sub++;      /* Increment 16bit Counter */
    }while((counter_sub & 0x4000) == 0); /* 16bit Counter bit14 = 1 ? */

}

/*****
/*   Timer V Interrupt
*****/
void tvint ( void )
{

    OVF = 0;

    if(LDONF == 1){
        P74 = 0;            /* LDONF = 1 ? */
        LDONF = 0;         /* Turn off LED */
        LDONF = 0;         /* Clear LDONF */
    }
    else{
        P74 = 1;            /* urn on LED */
        LDONF = 1;         /* Set LDONF */
    }

}

```

## リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0026
CV3	H'002C
P	H'0100
B	H'FB80