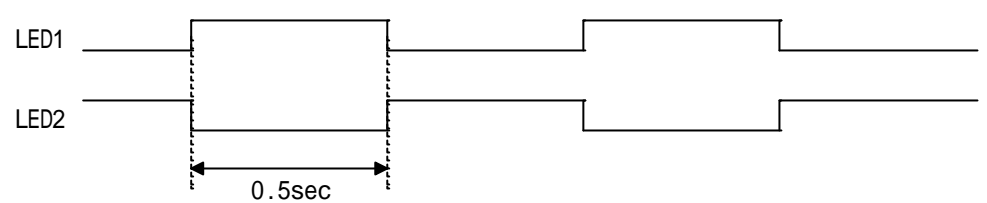
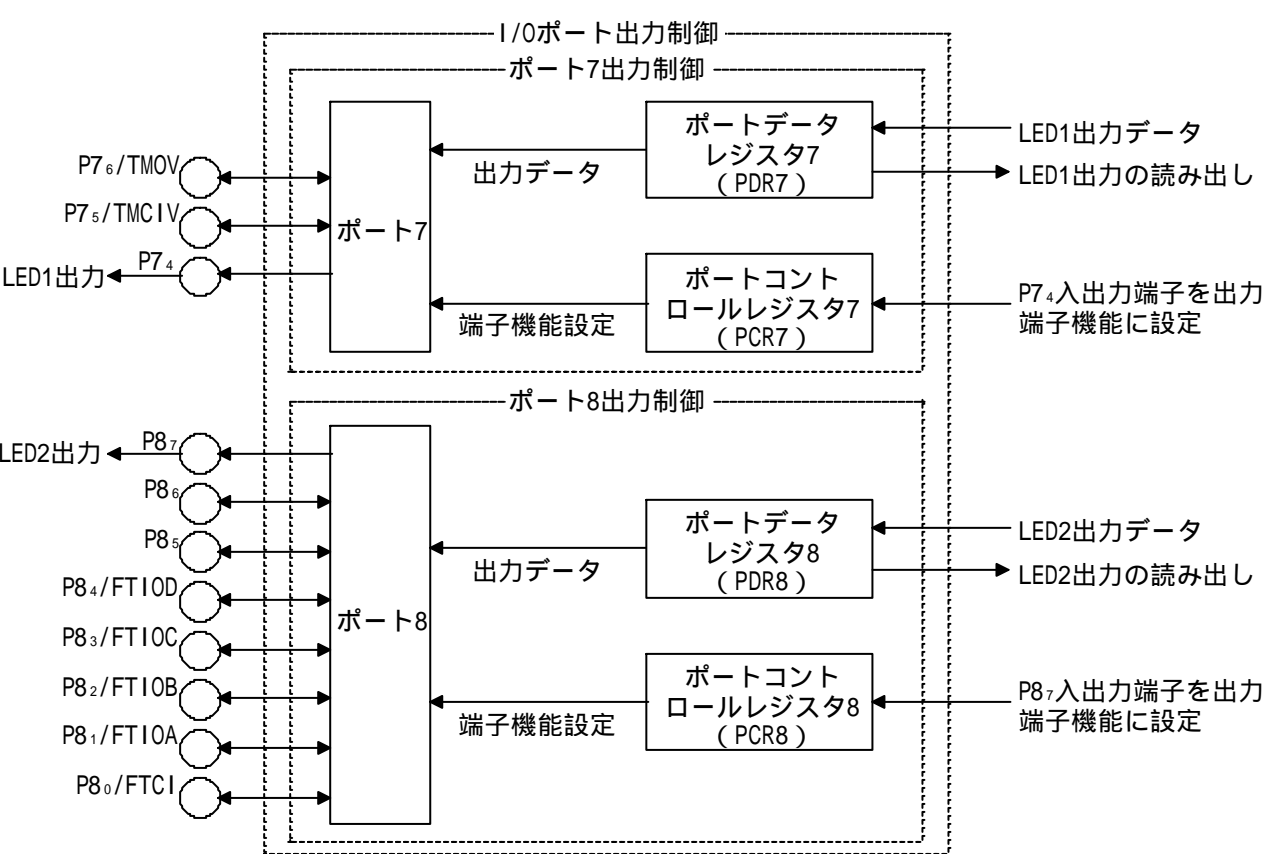


2.19 I/Oポートに接続したLEDの点滅動作

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
<p>仕様</p> <p>(1) 図1に示すように、ポートに接続した2つのLEDを交互に点灯/消灯させます。 (2) 点灯、および消灯の時間はタイマA時計用タイムベース機能を使用して0.5secに設定します。 (3) LED1はポート7のP7₄出力端子に、LED2はポート8のP8₇出力端子に接続されているものとします。</p>  <p style="text-align: center;">図1 LED点滅動作</p>		

使用機能説明
<p>(1) 本タスク例では、I/Oポートに接続したLEDの点滅動作を行いません。</p> <p>(a) 図2にI/Oポートのブロック図を示します。以下にI/Oポートのブロック図について説明します。</p> <ul style="list-style-type: none"> ・ポートコントロールレジスタ7 (PCR7) は、ポート7の各端子P7₆~P7₄の入出力をビットごとに制御します。PCR7₄に"1"をセットするとP7₄端子は出力端子となり、"0"にクリアすると入力端子となります。 ・ポートデータレジスタ7 (PDR7) は、ポート7の各端子P7₆~P7₄のデータを格納する8ビットのレジスタです。PCR7が"1"のとき、ポート7のリードを行なうと、PDR7の値を直接リードします。そのため端子状態の影響を受けません。PCR7が"0"のとき、ポート7のリードを行なうと、端子状態が読み出されます。 ・ポートコントロールレジスタ8 (PCR8) は、ポート8の各端子P8₇~P8₀の入出力をビットごとに制御します。PCR8₇に"1"をセットするとP8₇端子は出力端子となり、"0"にクリアすると入力端子となります。 ・ポートデータレジスタ8 (PDR8) は、ポート8の各端子P8₇~P8₀のデータを格納する8ビットのレジスタです。PCR8が"1"のとき、ポート8のリードを行なうと、PDR8の値を直接リードします。そのため端子状態の影響を受けません。PCR8が"0"のとき、ポート8のリードを行なうと、端子状態が読み出されます。  <p style="text-align: center;">図2 I/Oポート機能のブロック図</p>

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
---------------------	------	--------

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、I/Oポートに接続したLEDの点滅動作を行ないます。

表1 機能割付け

機能	機能割付け
PDR7	ポート7のP7 ₆ ～P7 ₄ のデータを格納
PDR8	ポート8のP8 ₇ ～P8 ₆ のデータを格納
PCR7	ポート7のP7 ₆ ～P7 ₄ 入出力端子機能の設定
PCR8	ポート8のP8 ₇ ～P8 ₆ 入出力端子機能の設定
P7 ₄	LED1の出力端子
P8 ₇	LED2の出力端子
PSW	32.768kHzを4分周したクロックを入力とする5ビットのカウンタ
TCA	PSW出力クロックを入力とする8ビットのカウンタ
TMA	タイマAを時計用タイムベース機能に設定、TCAオーバフロー周期を0.5secに設定

動作説明

(1) 図3に動作原理を示します。図3に示すようなハードウェア処理、およびソフトウェア処理によりI/Oポートに接続したLEDの点滅動作を行ないます。

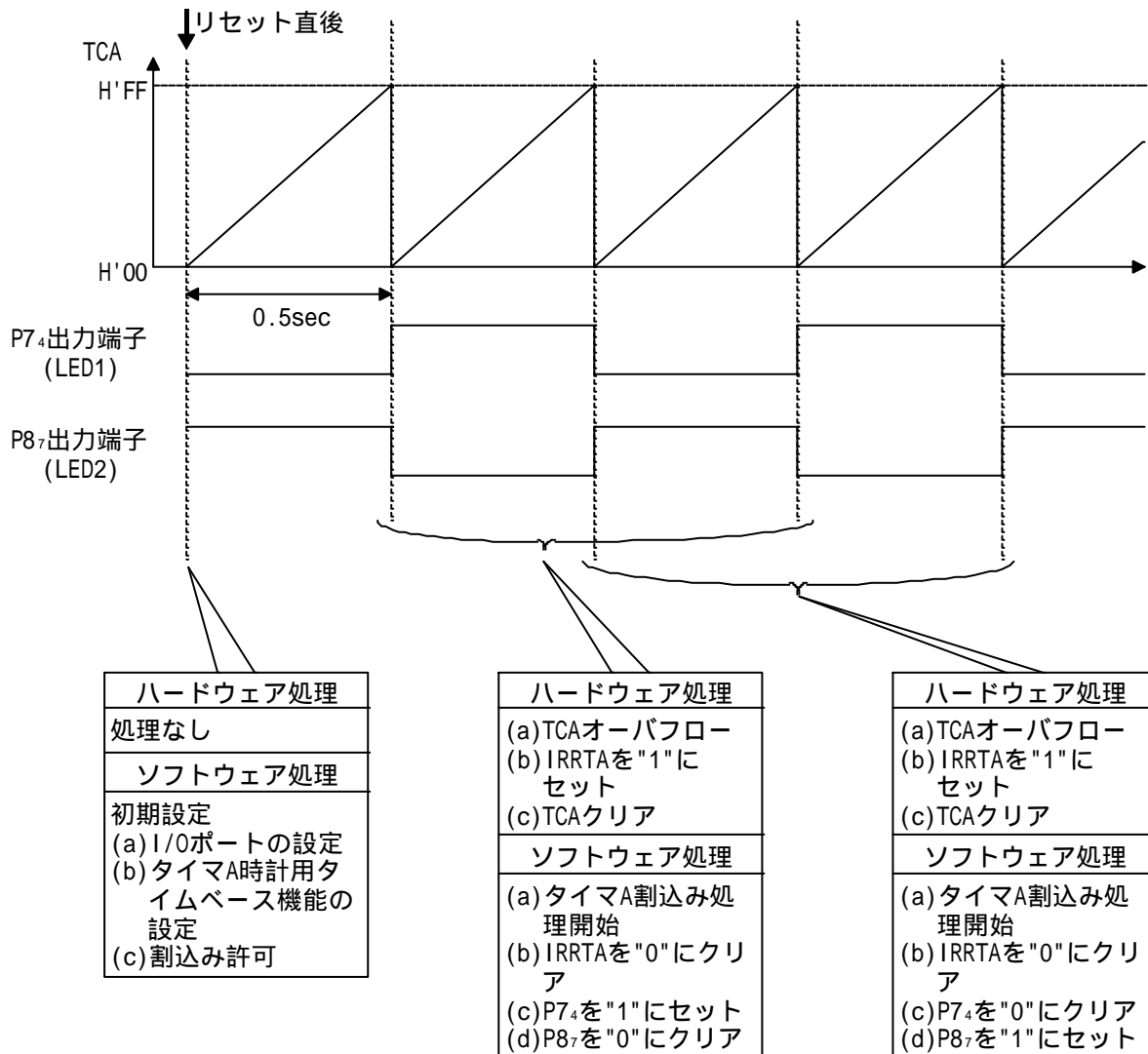


図3 I/Oポートに接続したLEDの点滅動作の動作原理

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
---------------------	------	--------

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時計用タイムベース機能の設定、I/Oポートの設定、および割込みの許可を行なう
ポート出力	taint	タイマA割込み処理ルーチンで、LED1、LED2の出力の判定、および出力の制御を行なう

(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

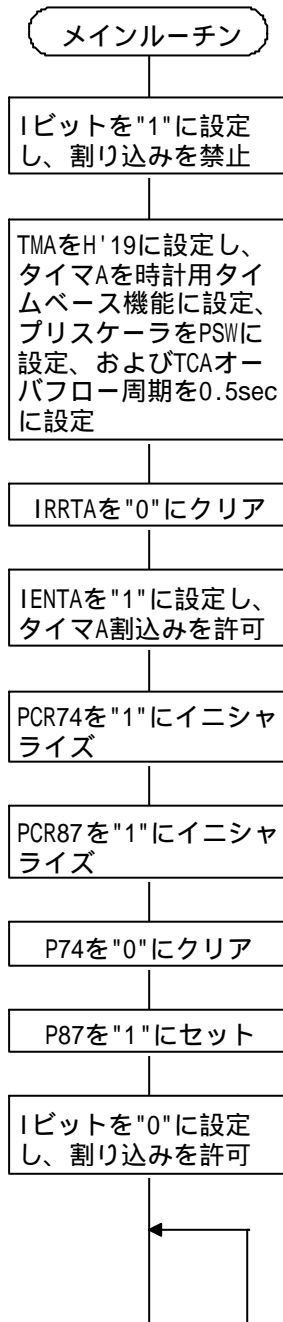
レジスタ名	機能	アドレス	設定値
TMA TMA3 TMA2 TMA1 TMA0	タイマモードレジスタA (内部クロックセレクト3~0) : TMA3="1"、TMA2="0"、TMA1="0"、TMA0="1"のとき、タイマA機能を時計用タイムベース機能に設定、プリスケアラをPSWに設定、およびTCAオーバフロー周期を0.5secに設定	H'FFA6 ビット3 ビット2 ビット1 ビット0	TMA3="1" TMA2="0" TMA1="0" TMA0="1"
TCA	タイマカウンタA : PSWの出力するクロックを入力とし、0.5secでオーバフローする8ビットのアップカウンタ	H'FFA7	H'00
PDR7 P7 ₄	ポートデータレジスタ7 (ポートデータレジスタ7 ₄) : P7 ₄ =0のとき、P7 ₄ 端子の出力レベルは"Low" : P7 ₄ =1のとき、P7 ₄ 端子の出力レベルは"High"	H'FFDA ビット4	0
PDR8 P8 ₇	ポートデータレジスタ8 (ポートデータレジスタ8 ₇) : P8 ₇ =0のとき、P8 ₇ 端子の出力レベルは"Low" : P8 ₇ =1のとき、P8 ₇ 端子の出力レベルは"High"	H'FFDB ビット7	1
PCR7 PCR7 ₄	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 ₄) : PCR7 ₄ ="1"のとき、P7 ₄ 入出力端子は、P7 ₄ 出力端子として機能	H'FFEA ビット4	1
PCR8 PCR8 ₇	ポートコントロールレジスタ8 (ポートコントロールレジスタ8 ₇) : PCR8 ₇ ="1"のとき、P8 ₇ 入出力端子は、P8 ₇ 出力端子として機能	H'FFEB ビット7	1
IENR1 IENTA	割込み許可レジスタ1 (タイマA割込みイネーブル) : 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 IRRTA	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRRTA="0"のとき、タイマA割込みが要求されていない : IRRTA="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

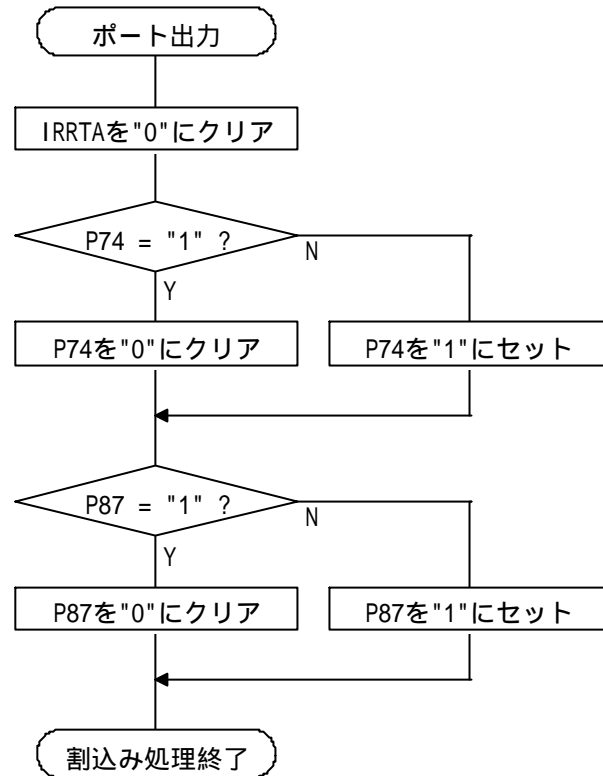
本タスク例では、引数は使用していません。

フローチャート

(a) メインルーチン



(b) タイマA割り込み処理ルーチン



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
/*
/*      'Flickering of LEDs Connected to I/O Port'
/*
/*      Function
/*      : I/O Port
/*
/*      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    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 74 */
#define    PDR8_BIT  (*(struct BIT *)0xFFDB) /* Port Data Register 8 */
#define    P87        PDR8_BIT.b7 /* Port Data Register 87 */
#define    PCR7_BIT  (*(struct BIT *)0xFFEA) /* Port Control Register 7 */
#define    PCR74     PCR7_BIT.b4 /* Port Control Register 74 */
#define    PCR8_BIT  (*(struct BIT *)0xFFEB) /* Port Control Register 8 */
#define    PCR87     PCR8_BIT.b7 /* Port Control Register 87 */
#define    IENR1_BIT (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1 */
#define    IENTA     IENR1_BIT.b6 /* Timer A Interrupt Enable */
#define    IRR1_BIT  (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1 */
#define    IRRTA     IRR1_BIT.b6 /* Timer A Interrupt Request Flag */

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

/*****
/*      Vector Address
*****/

```

I/Oポートに接続したLEDの点滅動作	使用機能	I/Oポート
---------------------	------	--------

プログラムリスト

```

#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) = {
taint /* 26 Timer A Interrupt */
};
#pragma section /* P */
/*****/
/* Main Program */
/*****/
void main ( void )
{
set_imask_ccr(1); /* Interrupt Disable */

TMA = 0x19; /* Initialize TCA Overflow Period */

IRRRTA = 0; /* Clear IRRRTA */
IENTA = 1; /* Timer A Interrupt Enable */

PCR74 = 1; /* Initialize P74 Output Terminal Function */
PCR87 = 1; /* Initialize P87 Output Terminal Function */

P74 = 1; /* Initialize P74 Terminal Output */
P87 = 0; /* Initialize P87 Terminal Output */

set_imask_ccr(0); /* Interrupt Enable */

while(1) {
;
}

/*****/
/* Timer A Interrupt */
/*****/
void taint ( void )
{
IRRRTA = 0; /* Clear IRRRTA */
if ( P74 == 1 ) { /* Turn on LED1 ? */
P74 = 0; /* Turn off LED1 */
}
else{
P74 = 1; /* Turn on LED1 */
}

if ( P87 == 1 ){ /* Turn on LED2 ? */
P87 = 0; /* Turn off LED2 */
}
else{
P87 = 1; /* Turn on LED2 */
}
}

```

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
P	H'0100