

## 2.23 スタンバイモードへの遷移

スタンバイモードへの遷移	使用機能	低消費電力モード：スタンバイモード
--------------	------	-------------------

仕様

- (1) スタンバイモードへの遷移を行ないます。
- (2) アクティブモードでプログラム実行中に、 $\overline{\text{IRQ}}_1$ 端子に接続したスイッチ入力のオンにより $\text{IRQ}_1$ 割込みを発生させ、 $\text{IRQ}_1$ 割込み処理終了後にSLEEP命令の実行によりスタンバイモードへ遷移させます。アクティブモードではLEDの点滅動作を行ない、スタンバイモードへ遷移するとLEDは点灯します（P7<sub>4</sub>端子はスタンバイモード時、ハイインピーダンス状態）。
- (3) スタンバイモードで、 $\overline{\text{IRQ}}_0$ 端子に接続したスイッチ入力のオンにより $\text{IRQ}_0$ 割込みを発生させ、スタンバイモードを解除し、アクティブモードへ遷移します。アクティブモードへ遷移すると、再びLEDの点滅動作を行ないます。
- (4) LEDはポート7のP7<sub>4</sub>出力端子に接続されているものとします。
- (5) 図1に $\overline{\text{IRQ}}_1$ 、 $\overline{\text{IRQ}}_0$ 端子に接続するスイッチの接続例を示します。

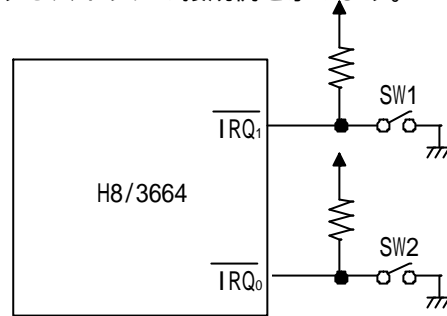


図1 スタンバイモードへの遷移におけるスイッチ接続例

使用機能説明

- (1) 本タスク例では、低消費電力モードのスタンバイモードへの遷移を行ないます。図2にスタンバイモードへのモード遷移図を示します。以下にスタンバイモードの機能の説明を示します。
  - ・アクティブモードで、SYSCR1のSSBYが"1"、DTONが"0"、およびSMSEL、LSONが"X"のときSLEEP命令を実行すると、スタンバイモードに遷移します。
  - ・スタンバイモードではクロック発振回路からのクロック供給を停止するため、CPUおよび内蔵周辺機能が停止します。規定の電圧が与えられている限り、CPUのレジスタと一部の内蔵周辺機能の内部レジスタ、および内蔵RAMのデータは保持されています。さらに、RAMデータ保持電圧で規定した電圧が与えられている限り、内蔵RAMのデータは保持されています。このとき、I/Oポートはハイインピーダンス状態となります。
  - ・スタンバイモードの解除は、割込み（ $\text{IRQ}_3 \sim \text{IRQ}_0$ 、 $\text{WKP}_5 \sim \text{WKP}_0$ ） $\overline{\text{RES}}$ 端子入力によって行われます。
  - ・割込みによるスタンバイモードの解除は、割込み要求が発生するとシステムクロックの発振が開始され、SYSCR1のSTS2～STS0により設定された時間が経過した後、安定したシステムクロックがLSI全体に供給されて、スタンバイモードは解除され、割込み例外処理を開始します。
  - ・CCRのIビットが"1"のとき、あるいは、割込み許可レジスタにより当該割込みの受け付けが禁止されている場合は、スタンバイモードは解除されません。
  - ・RES端子によるスタンバイモードの解除は、RES端子を"Low"レベルにすると、システムクロックの発振が開始されます。発振安定時間経過後、RES端子を"High"レベルにすると、CPUはリセット例外処理を開始します。なお、システムクロックの発振開始と同時にLSI全体にシステムクロックが供給されます。RES端子は、必ずシステムクロックの発振が安定するまで、"Low"レベルを保持しなければなりません。
  - ・スタンバイモード解除後の発振安定時間は、SYSCR1のSTS2～STS0により設定します。
  - ・本タスク例では、 $\text{IRQ}_0$ 割込みによりスタンバイモードを解除します。
  - ・本タスク例では、動作周波数に16MHzを使用し、待機ステート数を131.072ステートに設定します。（発振安定時間:8.2ms）

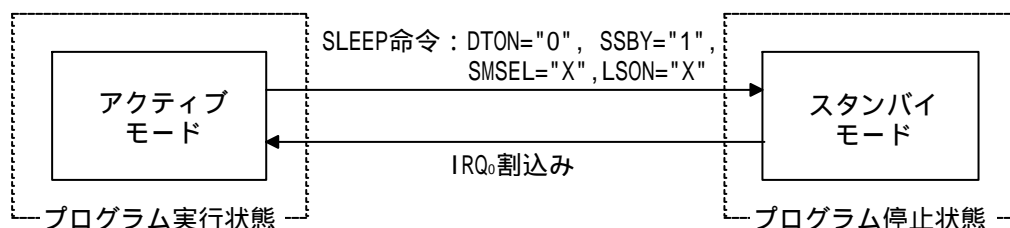


図2 スタンバイモードへの遷移におけるモード遷移図

: Xは"1"、"0"どちらでも良い

## 使用機能説明

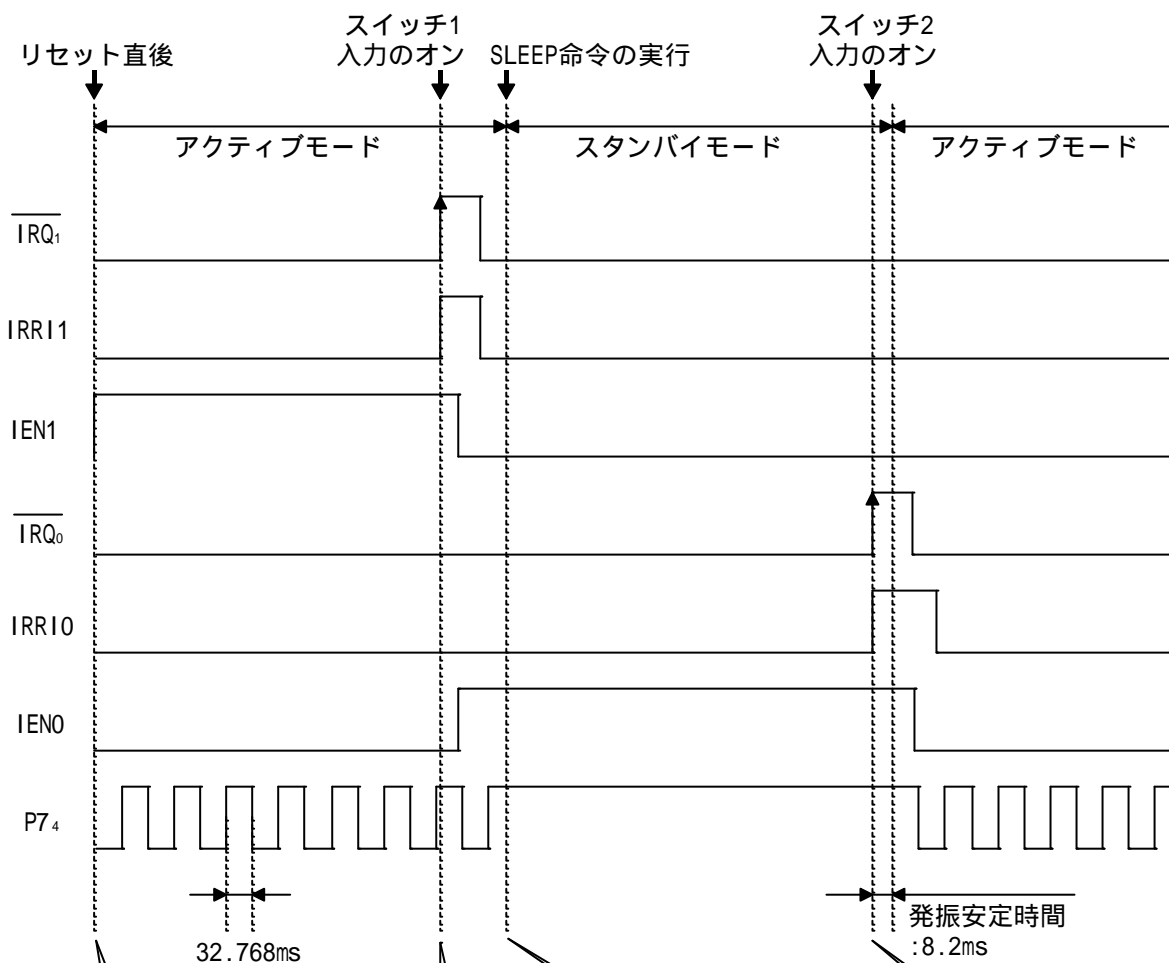
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、スタンバイモードへの遷移を行ないます。

表1 機能割付け

機能	機能割付け
SYSCR1	低消費電力モードの制御を行なう
SYSCR2	低消費電力モードの制御を行なう
PCR7	P7 <sub>4</sub> 出力端子機能の設定
PDR7	P7 <sub>4</sub> 出力端子のデータの格納
P7 <sub>4</sub>	LED出力
IRQ <sub>1</sub>	スイッチ入力1
IRQ <sub>0</sub>	スイッチ入力2

動作原理

(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スタンバイモードへの遷移を行ないます。



ハードウェア処理
処理なし
ソフトウェア処理
初期設定
(a) IRQ <sub>1</sub> 割込みの設定
(b) IRQ <sub>0</sub> 割込みの設定
(c) タイマA割込みの設定
(d) ポート7 <sub>4</sub> の設定
(e) 割込みの許可

ハードウェア処理
(a) IRR11を"1"にセット
ソフトウェア処理
(a) IRQ <sub>1</sub> 割込み処理の起動
(b) IRR11を"0"にクリア
(c) IRQ <sub>1</sub> 割込み要求の禁止
(d) タイマA割込み要求の禁止

ハードウェア処理
処理なし
ソフトウェア処理
(a) SLEEP命令の実行

ハードウェア処理
(a) IRR10を"1"にセット
ソフトウェア処理
(a) IRQ <sub>0</sub> 割込み処理の起動
(b) IRR10を"0"にクリア
(c) IRQ <sub>0</sub> 割込み要求の禁止
(d) タイマA割込み要求の許可

図3 スタンバイモードへの遷移の動作原理

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	IRQ <sub>i</sub> 割込みの設定、ポート7の設定、割込みの許可、LEDの制御、タイマ割込みの設定、スリープモードへの遷移を行なう
スイッチ1オン	IRQ1	IRQ <sub>i</sub> 割込み処理ルーチンで、SWONFのセット、IRQ <sub>i</sub> 割込みの禁止、タイマ割込みの禁止を行なう
スタンバイモード解除	IRQ0	IRQ <sub>o</sub> 割込み処理ルーチンで、IRQ <sub>o</sub> 割込みの禁止、タイマ割込みの許可を行なう。
LED制御	taint	タイマ割込み処理ルーチンで、LED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	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タイマ機能に、TCA入力クロックをシステムクロックの2048分周のクロックに設定	H'FFA6	H'12
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> 端子を出力端子に設定	H'FFEA ビット4	1
SYSCR1	SSBY システムコントロールレジスタ1 (ソフトウェアスタンバイ) : SSBY="1"のとき、アクティブモードでSLEEP命令実行後スタンバイモードに遷移	H'FFF0 ビット7	0
	STS2 STS1 STS0 システムコントロールレジスタ1 (スタンバイタイマセレクト2、1、0) : STS2="1", STS1="0", STS0="0"のとき、待機ステート数を131.072ステートに設定	H'FFF0 ビット6 ビット5 ビット4	STS2="1" STS1="0" STS0="0"
	SMSSEL システムコントロールレジスタ2 (スリープモード選択) : SMSSEL="X"のとき、スリープ命令実行後、スタンバイモード、サブスリープモード、サブアクティブモードの各1モードを設定	H'FFF1 ビット7	0
SYSCR2	LSON システムコントロールレジスタ2 (ロースピードオンフラグ) : LSON="X"のとき、スリープ命令実行後、スタンバイモード、サブスリープモード、サブアクティブモードの各1モードを設定	H'FFF1 ビット6	1
	DTON システムコントロールレジスタ2 (ダイレクトトランスファオンフラグ) : DTON="0"のとき、スリープ命令実行後、スリープモード、サブスリープモード、スタンバイモードの各1モードを設定	H'FFF1 ビット3	0
	MA2 MA1 MA0 システムコントロールレジスタ2 (アクティブモードクロックセレクト2~0) : MA2="0", MA1="1", MA0="1"、のとき、OSCをクロックに設定	H'FFF1 ビット4 ビット3 ビット2	MA2="0" MA1="1" MA0="1"
	SA1 SA0 システムコントロールレジスタ2 (サブアクティブモードクロックセレクト1、0) : SA1="0", SA0="0"のとき、サブアクティブモードのCPUの動作クロックを $\omega/8$ に設定	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"

: Xは"1"、"0"どちらでも良い

## ソフトウェア説明

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
PMR1	IRQ1 ポートモードレジスタ1 (IRQ1I/Oポートセレクト) : IRQ1_SET="0"のとき、IRQ1端子を汎用入出力ポートに設定 : IEG1_SET="1"のとき、IRQ1端子をIRQ1入力端子に設定	H'FFE0 ビット1	1
	IRQ0 ポートモードレジスタ1 (IRQ0I/Oポートセレクト) : IENO_SET="0"のとき、IRQ1端子を汎用入出力ポートに設定 : IENO_SET="1"のとき、IRQ1端子をIRQ1入力端子に設定	H'FFE0 ビット0	1
IEGR1	IEG1 割込みエッジセレクトレジスタ1 (IRQ1エッジセレクト) : IEG1="0"のとき、IRQ1端子入力の立ち下がりエッジを検出 : IEG1="1"のとき、IRQ1端子入力の立ち上がりエッジを検出	H'FFF2 ビット1	1
	IEG0 割込み許可レジスタ1 (IRQ0エッジセレクト) : IEO="0"のとき、IRQ0端子入力の立ち下がりエッジを検出 : IEO="1"のとき、IRQ0端子入力の立ち上がりエッジを検出	H'FFF2 ビット0	1
IENR1	IEN1 割込みイネーブルレジスタ1 (タイマ割込みイネーブル) : IEN1="0"のとき、タイマ割込み要求を禁止 : IEN1="1"のとき、タイマ割込み要求を許可	H'FFF4 ビット6	1
	IEN1 割込みイネーブルレジスタ1 (IRQ1割込み要求イネーブル) : IEN1="0"のとき、IRQ1端子による割込み要求を禁止 : IEN1="1"のとき、IRQ1端子による割込み要求を許可	H'FFF4 ビット1	1
	IENO 割込みイネーブルレジスタ1 (IRQ0割込み要求イネーブル) : IENO="0"のとき、IRQ0端子による割込み要求を禁止 : IENO="1"のとき、IRQ0端子による割込み要求を許可	H'FFF4 ビット0	1
IRR1	IRRTA 割込みフラグレジスタ1 (タイマ割込み要求フラグ) : IRRTA="0"のとき、タイマ割込みが要求されていない : IRRTA="1"のとき、タイマ割込みが要求されている	H'FFF6 ビット6	0
	IRRI1 割込みフラグレジスタ1 (IRQ1割込み要求フラグ) : IRRI1="0"のとき、IRQ1端子入力による割込みが要求されていない : IRRI1="1"のとき、 $\overline{\text{IRQ1}}$ 端子入力による割込みが要求されている	H'FFF6 ビット1	0
	IRRI0 割込みフラグレジスタ1 (IRQ0割込み要求フラグ) : IRRI0="0"のとき、IRQ0端子入力による割込みが要求されていない : IRRI0="1"のとき、 $\overline{\text{IRQ0}}$ 端子入力による割込みが要求されている	H'FFF6 ビット0	0

## (4) 使用RAM説明

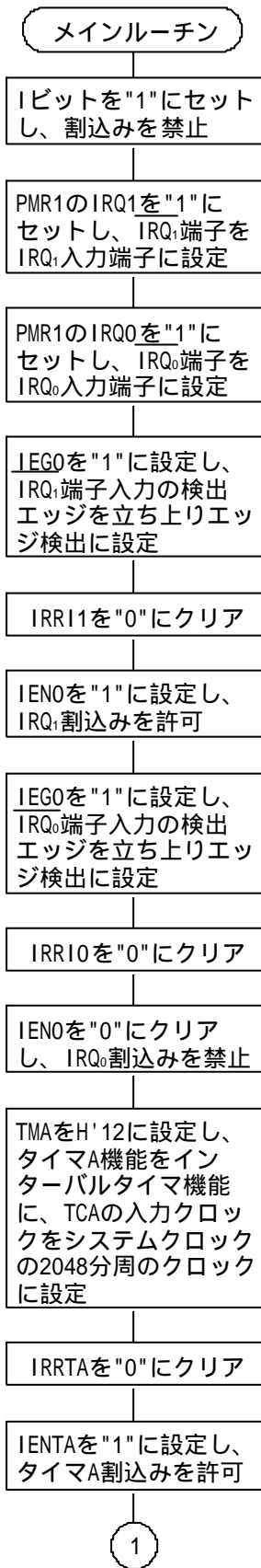
表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

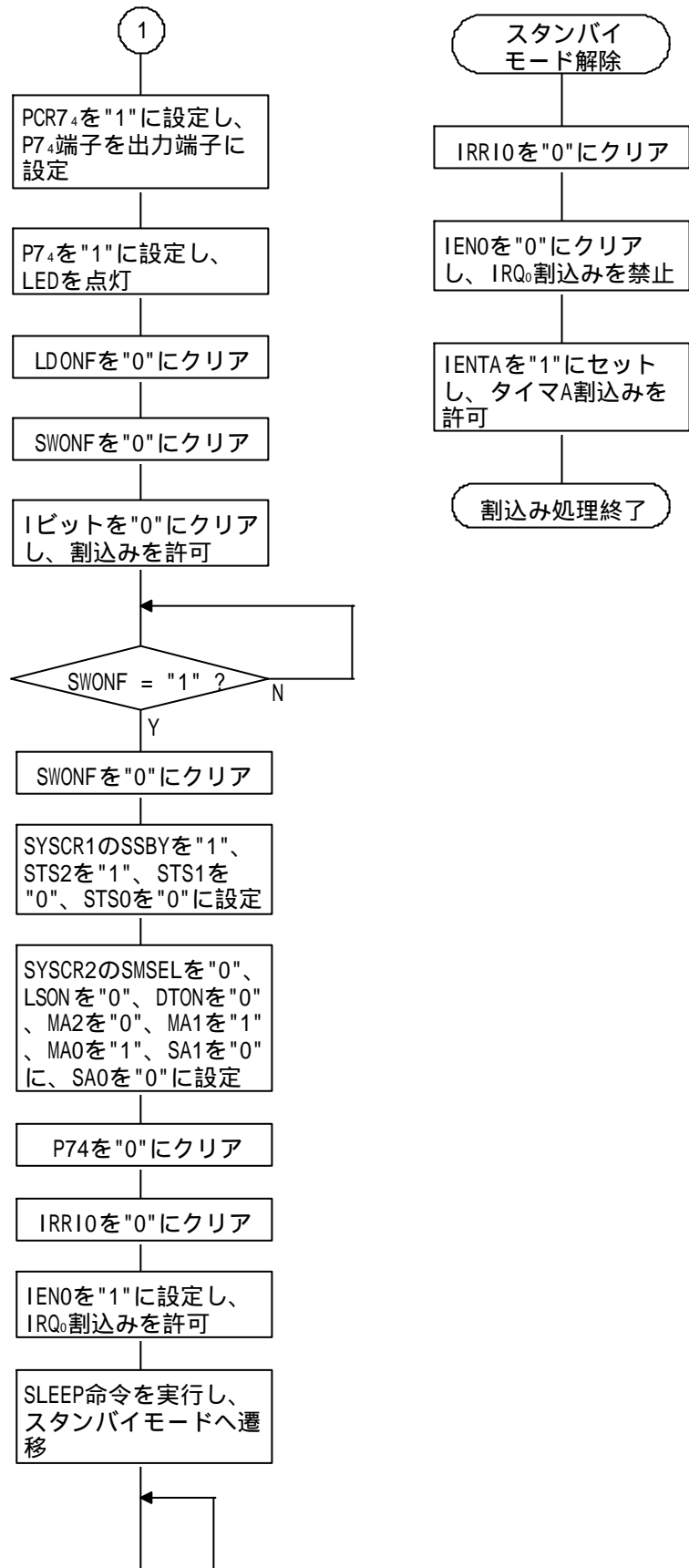
ラベル名	機能	アドレス	使用モジュール名
USRF	SWONF スイッチ2入力のON/OFFを判定するフラグ	H'FB80 ビット0	メインルーチン スイッチ2オン
	LDONF LEDのON/OFFを判定するフラグ	H'FB80 ビット1	LED制御

フローチャート

(a) メインルーチン



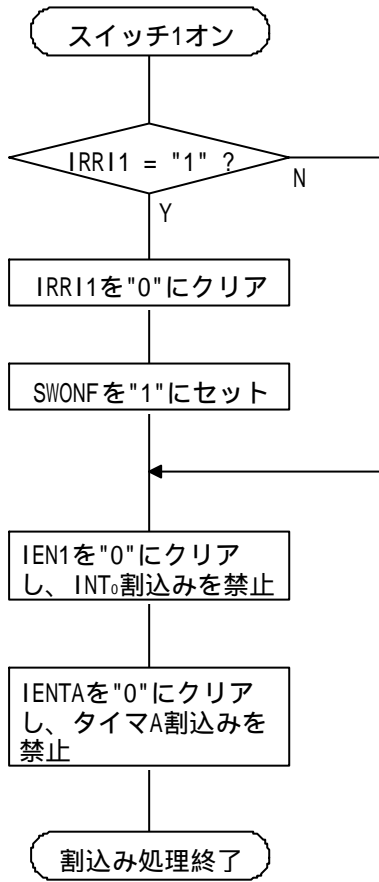
(b) IRQ0割込み処理ルーチン



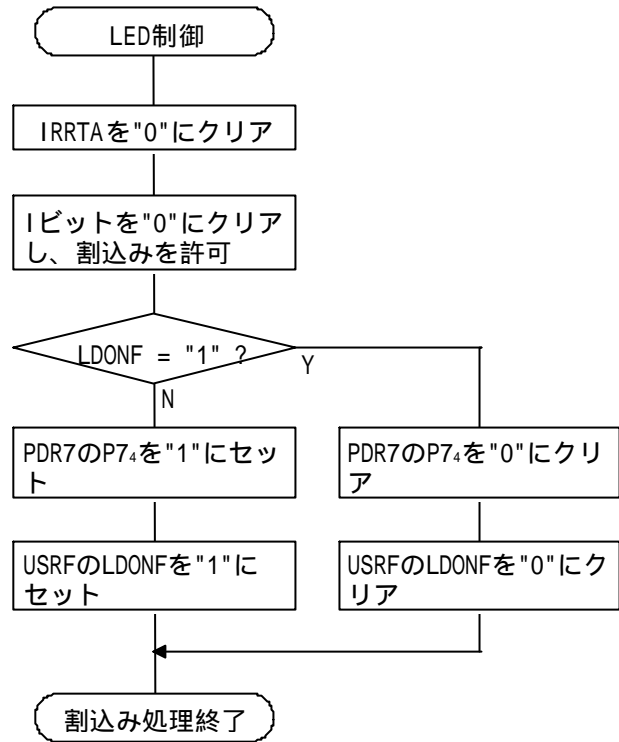
本例ではスタックポインタはINIT.SRC (アセンブリ言語) で設定してあります。

フローチャート

(c) IRQ<sub>1</sub>割込み処理ルーチン



(d) タイマA割込み処理ルーチン



## 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    #'FF80, R7
        LDC.B    #B'10000000, CCR
        JMP      @_main
;
        .END

/*****
/*
/*  H8/300H Tiny Series -H8/3664-
/*  Application Note
/*
/*  'Transition to Standby Mode'
/*
/*  Function
/*  : Power-Down Mode
/*    Standby Mode
/*
/*  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 TMA_BIT     (*(struct BIT *)0xFFA6)           /* Timer Mode Register A */
#define TMA3        TMA_BIT.b3                       /* Interval Time */
#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 PMR1_BIT    (*(struct BIT *)0xFFE0)           /* Port Mode Register 1 */
#define IRQ1_SET    PMR1_BIT.b5                       /* Port Mode Register 1 bit5 */
#define IRQ0_SET    PMR1_BIT.b4                       /* Port Mode Register 1 bit4 */
#define PCR7_BIT    (*(struct BIT *)0xFFEA)           /* Port Control Register 7 */
#define PCR74       PCR7_BIT.b4                       /* Port Control Register 7 bit4 */
#define SYSCR1      *(volatile unsigned char *)0xFFF0 /* System Control Register 1 */
#define SYSCR1_BIT  (*(struct BIT *)0xFFF0)           /* System Control Register 1 */
#define SSBY        SYSCR1_BIT.b7                     /* Software Standby */
#define STS2        SYSCR1_BIT.b6                     /* Standby Timer Select 2 */
#define STS1        SYSCR1_BIT.b5                     /* Standby Timer Select 1 */
#define STS0        SYSCR1_BIT.b4                     /* Standby Timer Select 0 */
#define NESEL       SYSCR1_BIT.b3                     /* Noise Elimination Sampling Frequency Select */
#define SYSCR2      *(volatile unsigned char *)0xFFF1 /* System Control Register 2 */
#define SYSCR2_BIT  (*(struct BIT *)0xFFF1)           /* System Control Register 2 */
#define LSON        SYSCR2_BIT.b6                     /* Low Speed On Flag */
#define DTON        SYSCR2_BIT.b5                     /* Direct Transfer On Flag */
#define MA1         SYSCR2_BIT.b3                     /* Active Mode Clock Select 1 */
#define MA0         SYSCR2_BIT.b2                     /* Active Mode Clock Select 0 */
#define SA1         SYSCR2_BIT.b1                     /* Subactive Mode Clock Select 1 */

```



## プログラムリスト

```

#define SA0          SYSCR2_BIT.b0          /* Subactive Mode Clock Select 0          */
#define IEGR1_BIT    (*(struct BIT *)0xFFF2) /* Interrupt Edge Select Register 1       */
#define IEQ1         IEGR1_BIT.b1          /* IRQ1 Edge Select                        */
#define IEQ0         IEGR1_BIT.b0          /* IRQ0 Edge Select                        */
#define IENR1_BIT    (*(struct BIT *)0xFFF4) /* Interrupt Enable Register 1            */
#define IENDT        IENR1_BIT.b7          /* Direct Transfer Interrupt Enable        */
#define IENTA        IENR1_BIT.b6          /* Timer A Interrupt Enable                */
#define IEN1         IENR1_BIT.b1          /* IRQ1 Interrupt Request Enable          */
#define IENO         IENR1_BIT.b0          /* IRQ0 Interrupt Request Enable          */
#define IRR1_BIT     (*(struct BIT *)0xFFF6) /* Interrupt Request Register 1           */
#define IRRDT        IRR1_BIT.b7          /* Direct Transfer Interrupt Request Flag  */
#define IRRTA        IRR1_BIT.b6          /* Timer A Interrupt Request Flag          */
#define IRR11        IRR1_BIT.b1          /* IRQ1 Interrupt Request Flag            */
#define IRR10        IRR1_BIT.b0          /* IRQ0 Interrupt Request Flag            */

#pragma interrupt (IRQ0)
#pragma interrupt (IRQ1)
#pragma interrupt (taint)
/*****
/* 関数定義 */
*****/
extern void INIT ( void ); /* SP Set */
void main ( void );
void IRQ0 ( void );
void IRQ1 ( void );
void taint ( void );
void sleep ( void );

/*****
/* RAM define */
*****/
unsigned char USRF; /* User Flag Area */

#define USRF_BIT (*(struct BIT *)&USRF)
#define LDONF USRF_BIT.b1 /* LED On Flag */
#define SWONF USRF_BIT.b0 /* Switch 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) = {
    IRQ0 /* IRQ0 Interrupt */
};
#pragma section V3 /* VECTOR SECTOIN SET */
void (*const VEC_TBL3[])(void) = {
    IRQ1 /* IRQ1 Interrupt */
};
#pragma section V4 /* VECTOR SECTOIN SET */
void (*const VEC_TBL4[])(void) = {
    taint /* timer A Interrupt */
};
#pragma section /* P */
/*****
/* Main Program */
*****/
void main ( void )
{
    set_imask_ccr(1); /* Interrupt Disable */

    IRQ1_SET = 1; /* Initialize IRQ1 Terminal Input */
    IRQ0_SET = 1; /* Initialize IRQ0 Terminal Input */

    IEQ1 = 1; /* Set Rising Edge of IRQ1 Terminal Input */

```

## プログラムリスト

```

IRQ0_SET = 1; /* Initialize IRQ0 Terminal Input */

IEG1 = 1; /* Set Rising Edge of IRQ1 Terminal Input */
IRRI1 = 0; /* Clear IRR11 */
IEN1 = 1; /* IRQ1 Interrupt Enable */

IEG0 = 1; /* Set Rising Edge of IRQ0 Terminal Input */
IRRI0 = 0; /* Clear IRR10 */
IENO = 0; /* IRQ0 Interrupt Disable */

TMA = 0x12; /* Initialize Timer A Function */
IRRTA = 0; /* Clear IRRTA */
IENTA = 1; /* Timer A Interrupt Enable */

P74 = 0; /* Initialize P74 */
PCR74 = 1; /* Initialize P74 Output Port */

LDONF = 0; /* Initialize LDONF */
SWONF = 0; /* Initialize SWONF */

set_imask_ccr(0); /* Interrupt Enable */

while(SWONF != 1){ /* SWONF = "1" ? */
    ;
}

SWONF = 0; /* Initialize SWONF */

SYSCR1 = 0xC0; /* Initialize Function of Standby Mode 1 */
SYSCR2 = 0x0C; /* Initialize Function of Standby Mode 2 */

P74 = 0; /* Turn Off LED */
IRRI0 = 0; /* Clear IRR10 */
IENO = 1; /* IRQ0 Interrupt Enable */

sleep(); /* Transition to Standby Mode */

while(1) {
    ;
}

/*****/
/* IRQ0 Interrupt */
/*****/
void IRQ0 ( void )
{

    IRR10 = 0; /* Clear IRR10 */

    IENO = 0; /* IRQ0 Interrupt Disable */

    IENTA = 1; /* Timer A Interrupt Enable */

}

/*****/
/* IRQ1 Interrupt */
/*****/
void IRQ1 ( void )
{

    if(IRRI1 == 1){ /* IRR11 = "1" ? */
        IRR11 = 0; /* Clear IRR11 */
        SWONF = 1; /* Set SWONF */
    }

    IEN1 = 0; /* IEN1 Interrupt Disable */
}

```

## プログラムリスト

```

    IENTA = 0;                                /* Timer A Interrupt Disable */
}

/*****
/*   Timer A Interrupt
*****/
void taint ( void )
{
    IRRTA = 0;                                /* Clear IRRTA          */
    set_imask_ccr(0);                          /* Interrupt Enable     */

    if(LDONF == 1){                            /* LDONF = "1" ?      */
        P74 = 0;                               /* Turn Off LED        */
        LDONF = 0;                             /* Clear LDONF         */
    }
    else{
        P74 = 1;                               /* Turn 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'001C
CV3	H'001E
CV4	H'0026
P	H'0100
B	H'FB80