2.24 スリープモード→サブアクティブモードへの遷移

スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

仕様

(1) スリープモード→サブアクティブモードへの遷移を行ないます。
(2) アクティブモードでSYSCR1のSSBYが"0"、SYSCR2のSMSELが"0"、LSONが"1"、DTONが"0"のときSLEEP命令を実行することによりスリープモードへ遷移します。
(3) スリープモードに遷移後、タイマA割込み要求が発生することによりサブアクティブモードへ遷移します。
(4) サブアクティブモードで、タイマA割込み要求の回数をカウントし、再びスリープモードへ遷移します。
(5) タイマA割込み要求は、0.5secごとに発生します。また、タイマA割込み処理で0.5secごとにLEDを交互に点灯、消灯させます。
(6) タイマA割込みが120回要求され、60sec経過するとサブアクティブモードからアクティブモードへ直接遷移させて終了します。
(7) 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。

使用機能説明

(1) 本タスク例では、低消費電力モードのスリープモード→サブアクティブモードへの遷移を行ないます。図2にスリープモード→サブアクティブモードへのモード遷移図を示します。
以下にスリープモード、サブアクティブ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"0"、SYSCR2のSMSELが"0"、LSONが"1"、DTONが"0"のとき SLEEP命令を実行すると、スリープモードへ遷移します。
- スリープモードではCPUの動作は停止しますが、内蔵周辺モジュールはSYSCR2のMA2 、MA1、MA0で設定した周波数のクロックで動作します。なお、CPU のレジスタの内容は保持されます。
- スリープモードの解除は、すべての割込み（タイマA、タイマV、タイマW、IRQ_3～IRQ_0、WKP_5～WKP_0、ウォッチドックタイマ、SCI3、IIC、A/D変換器）、RES端子入力によって行われます。
- 割込みによる解除において、割込み要求が発生すると、スリープモードは解除され、割込み例外処理を開始します。
- 本タスク例では、スリープモードの解除にタイマA割込みを使用します。なお、スリープモードの解除後はサブアクティブモードへ遷移させます。
- サブアクティブモードでSYSCR1のSSBYを"X"、LSONを"0"、SYSCR2のSMSELが"X"、LSONが"0"、DTONを"1"にセットした状態でSLEEP命令を実行すると、SYSCR1のSTS2～STS0により設定された時間を経過した後、アクティブモードへ直接遷移します。
- サブアクティブ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。
- CCRのIビットが"1"の場合、あるいは割込みイネーブルレジスタ1により当該割込みの受け付けが禁止されている場合は、スリープモードは解除されません。
- RES端子による解除は、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はりセット例外処理を開始します。
- 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。
- 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)

※: Xは"1"、"0"どちらでも良い

使用機能説明

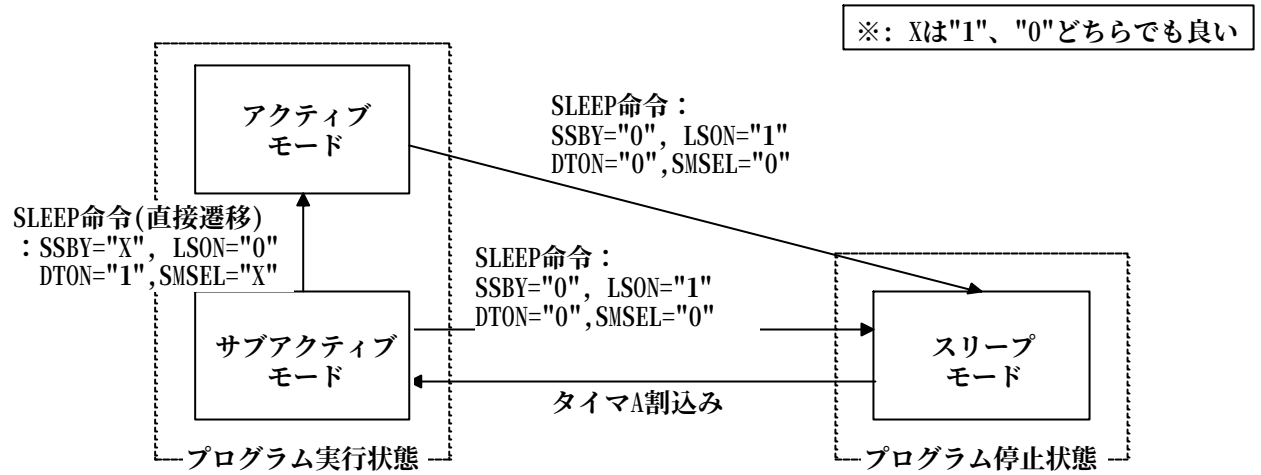


図2　スリープモード→サブアクティブモードへの遷移におけるモード遷移図

（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、スリープモード→サブアクティブモードへの遷移を行ないます。

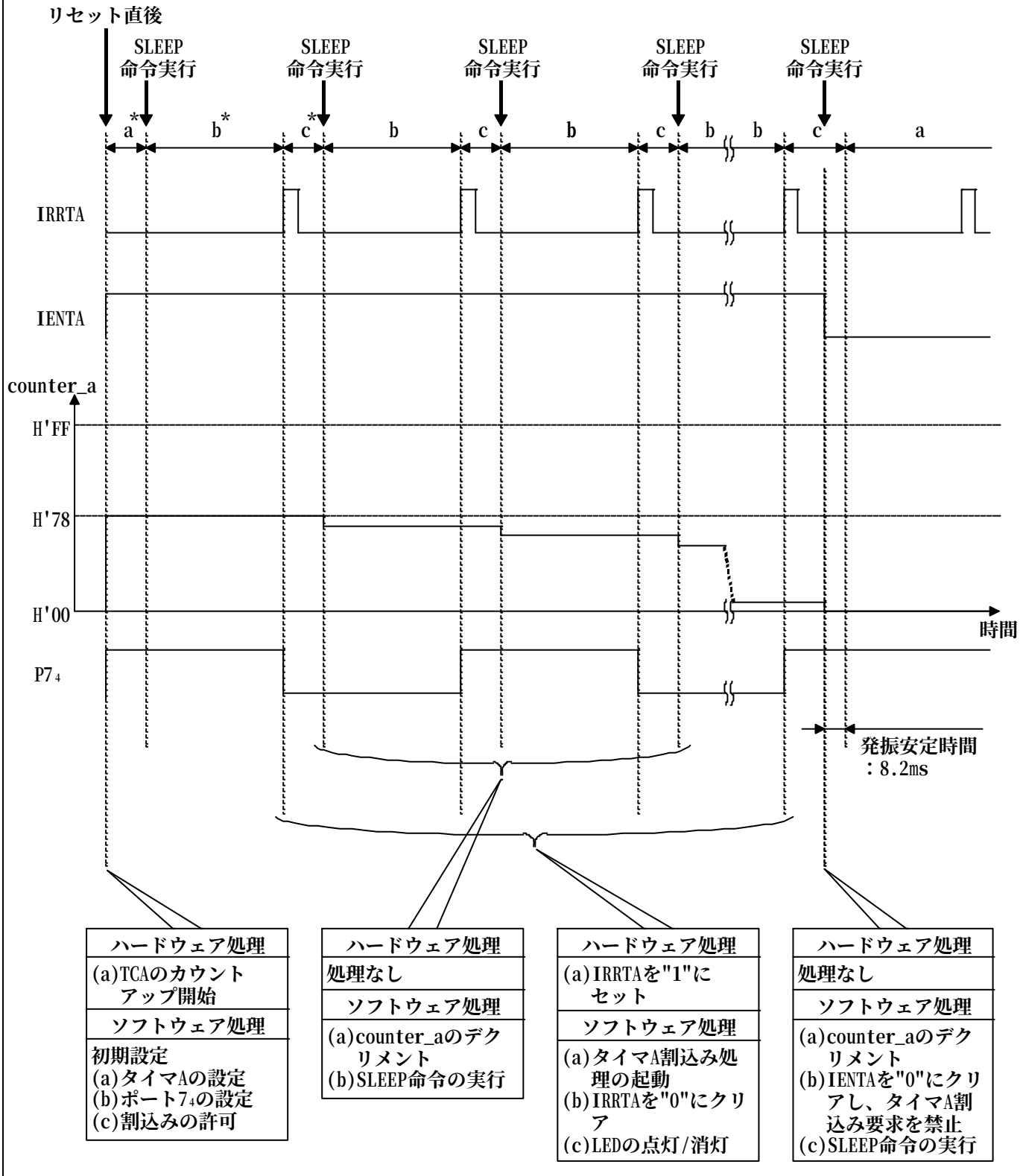
表1　機能割付け

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	$P7_4$出力端子機能の設定
PDR7	$P7_4$出力端子のデータの格納
$P7_4$	LED出力
TMA	タイマA時計用タイムベース機能、およびTCAオーバフロー周期の設定
TCA	時計用タイムベース機能により0.5secでオーバフローする8ビットのアップカウンタ
IRRTA	タイマA割込み要求の有無を反映
IENTA	タイマA割込み要求の許可、または禁止を設定

スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

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
スリープモード→サブアクティブモードへの遷移を行ないます。



【注】* a：アクティブモード
b：スリープモード
c：サブアクティブモード

図3　スリープモード→サブアクティブモードへの遷移の動作原理

スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

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、ポート7の設定、割込みの許可、スリープモード→サブアクティブモードへの遷移、タイマA割込みの禁止を行なう
LED制御	taint	タイマA割込み処理ルーチンで、LEDの制御を行なう
直接遷移	dtint	直接遷移割込み処理ルーチンで、直接遷移割込み要求フラグのクリアを行なう

(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'19のとき、タイマA機能を時計用タイムベース機能に、TCAのオーバフロー周期を0.5secに設定	H'FFA6	H'19
TCA		タイマカウンタA ：PSW出力クロックを入力とし、時計用タイムベース機能により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	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、$P7_4$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="0"のとき、アクティブモードでSLEEP命令実行後スリープモード、あるいはサブスリープモードに遷移。サブアクティブモードでSLEEP命令実行後アクティブモードに直接遷移	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"
SYSCR2	SMSEL	システムコントロールレジスタ2（スリープモード選択） ：SMSEL="0"のとき、アクティブモードで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の各1モードを設定	H'FFF1 ビット7	0
	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="1"のとき、アクティブモードで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、アクティブモード（直接遷移）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、φOSC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ϕ_W/8に設定	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"

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
IENR1	IENDT	割込みイネーブルレジスタ1（直接遷移割込みイネーブル） ：IENDT="0"のとき、直接遷移による割込み要求を禁止 ：IENDT="1"のとき、直接遷移による割込み要求を許可	H'FFF4 ビット7	1
	IENTA	割込みイネーブルレジスタ1（タイマA割込みイネーブル） ：IENTA="0"のとき、タイマA割込み要求を禁止 ：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	IRRDT	割込みフラグレジスタ1（直接遷移割込み要求フラグ） ：IRRDT="0"のとき、直接遷移による割込みが要求されていない ：IRRDT="1"のとき、直接遷移による割込みが要求されている	H'FFF6 ビット7	0
	IRRTA	割込みフラグレジスタ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

表4に本タスク例における使用RAM説明を示します。

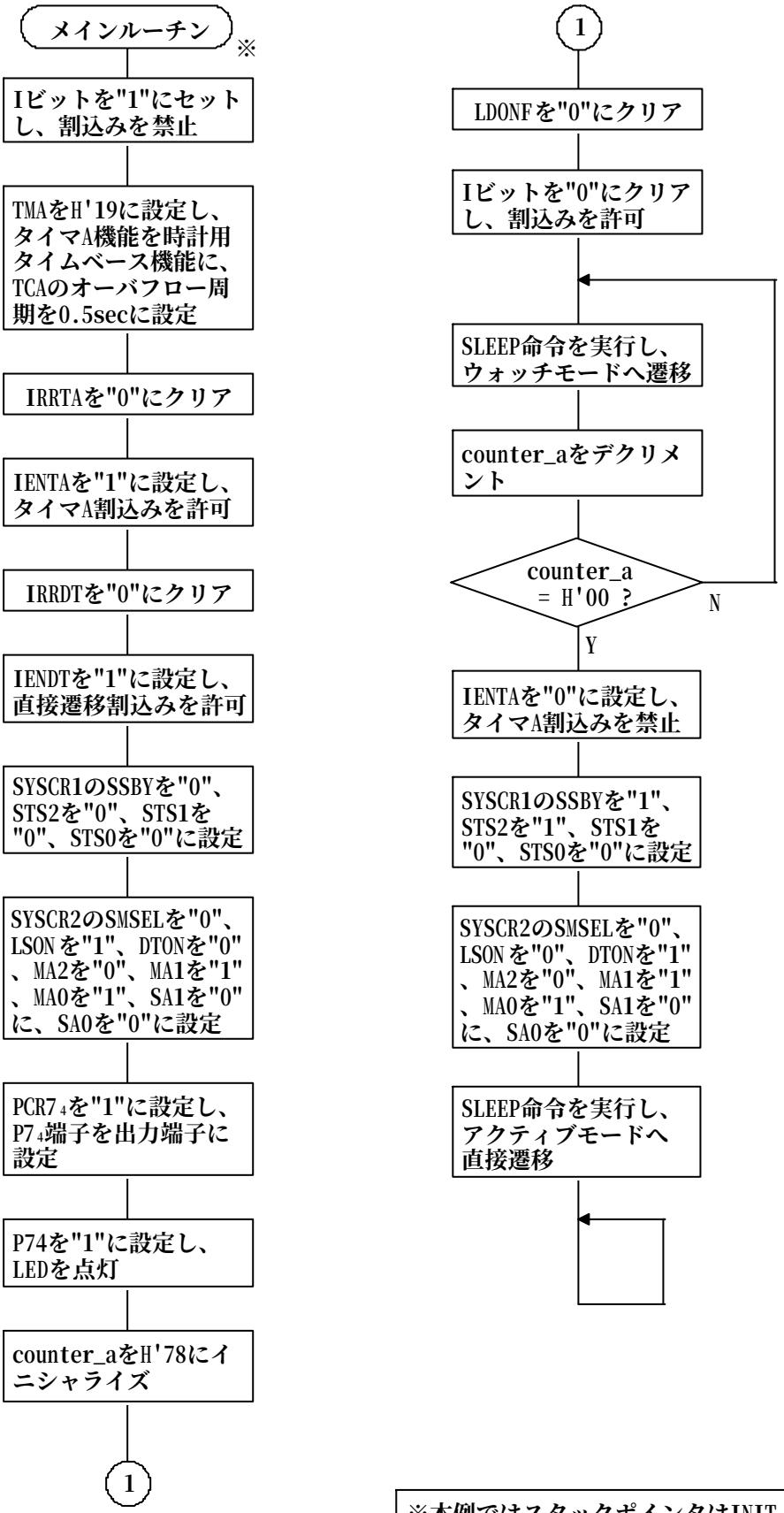
表4　使用RAM説明

ラベル名		機能	アドレス	使用モジュール名
counter_a		タイマAの割込みをカウントするダウンカウンタ	H'FB80	メインルーチン
USRF	LDONF	LEDのON/OFFを判定するフラグ	H'FB81 ビット0	LED制御

スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

フローチャート

(a) メインルーチン

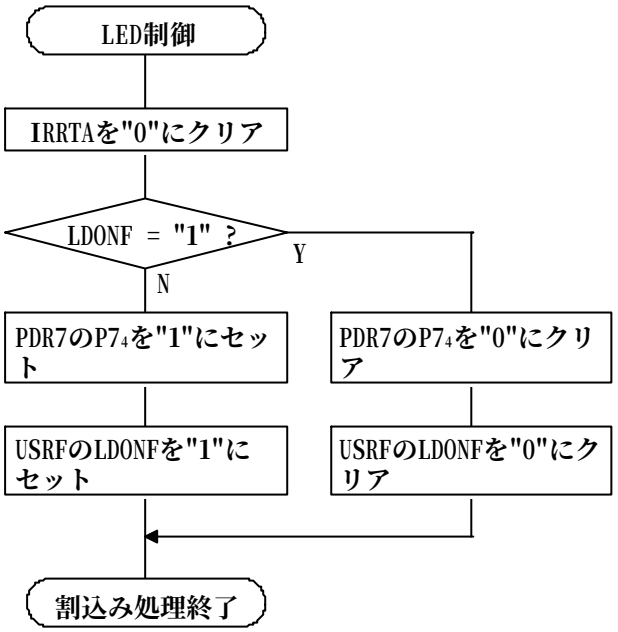


※本例ではスタックポインタはINIT.SRC（アセンブリ言語）で設定してあります。

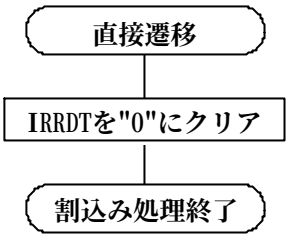
スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

フローチャート

(b) タイマA割込み処理ルーチン



(c) 直接遷移割込み処理ルーチン



スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ ： ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

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                                 */
/*                                                     */
/*    'Transition to Sleep Mode → Sub Active Mode '    */
/*                                                     */
/*    Function                                         */
/*    : Power-Down Mode                                */
/*       Sleep Mode → Sub Active Mode                  */
/*                                                     */
/*    External Clock : 16MHz                           */
/*    Internal Clock : 16MHz                           */
/*    Sub Clock      : 32.768kHz                       */
/*                                                     */
/*******************************************************/

#include    <machine.h>

/*******************************************************/
/*    Symbol Defnition                                 */
/*******************************************************/
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 7 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                  */
#define   SA0        SYSCR2_BIT.b0                     /* Subactive Mode Clock Select 0                  */
#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   IRR1_BIT   (*(struct BIT *)0xFFF6)           /* Interrupt Request Register 1                   */
#define   IRRDT      IRR1_BIT.b7                       /* Direct Transfer Interrupt Request Flag         */
```

スリープモード→サブアクティブモードへの遷移	使用機能	低消費電力ﾓｰﾄﾞ : ｽﾘｰﾌﾟﾓｰﾄﾞ、ｻﾌﾞｱｸﾃｨﾌﾞﾓｰﾄﾞ

プログラムリスト

```
#define   IRRTA      IRR1_BIT.b6                    /* Timer A Interrupt Request Flag            */

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

/************************************************************/
/*    RAM define                                            */
/************************************************************/
unsigned char   counter_a;
unsigned char   USRF;                               /* User Flag Erea            */

#define        USRF_BIT    (*(struct BIT *)&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) = {
    dtint                                           /* Direct Transfer  Interrupt */
};
#pragma    section        V3                        /* VECTOR SECTOIN SET        */
void (*const VEC_TBL3[])(void) = {
    taint                                           /* timer A  Interrupt        */
};
#pragma    section                                  /* P                         */
/************************************************************/
/*    Main Program                                          */
/************************************************************/
void      main ( void )
{

    set_imask_ccr(1);                               /* Interrupt Disable                      */

    TMA = 0x19;                                     /* Initialize Timer A Function            */

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

    IRRDT = 0;                                      /* Clear IRRDT                            */
    IENDT = 1;                                      /* Direct Transfer Interrupt Enable       */

    SYSCR1 = 0x00;                                  /* Initialize Function of Sleep Mode 1    */
    SYSCR2 = 0x4C;                                  /* Initialize Function of Sleep Mode 2    */

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

    counter_a = 0x78;                               /* Initialize 8bit Timer A Interrupt Counter */

    LDONF = 0;                                      /* Initialize LDONF                       */

    set_imask_ccr(0);                               /* Interrupt Enable                       */

    do{
        sleep();                                    /* Transition to Watch Mode               */

        counter_a--;                                /* Decrement 8bit Timer A Interrupt Counter  */
```

プログラムリスト

```
  }while(counter_a != 0x00);                    /* 8bit Timer A Interrupt Counter = H'00 ?   */

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

    SYSCR1 = 0xC0;                              /* Initialize Function of Active Mode 1      */
    SYSCR2 = 0x2C;                              /* Initialize Function of Active Mode 2      */

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

    while(1){
       ;
    }
}

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

    IRRTA = 0;                                  /* Clear IRRTA   */

    if(LDONF == 1){                             /* LDONF = "1" ? */
        P74 = 0;                                /* Turn Off LED  */
        LDONF = 0;                              /* Clear LDONF   */
    }
        else{
            P74 = 1;                            /* Turn On LED   */
            LDONF = 1;                          /* Set LDONF     */
        }
}

/*************************************************************/
/*    Direct Transfer Interrupt                              */
/*************************************************************/
void      dtint ( void )
{

    IRRDT = 0;                                  /* Clear IRRDT   */

}
```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'001A
CV3	H'0026
P	H'0100
B	H'FB80