

## 2.26 サブアクティブモードへの遷移

サブアクティブモードへの遷移	使用機能	低消費電力モード：サブアクティブモード
仕様	<p>(1) サブアクティブモードへの遷移を行ないます。</p> <p>(2) アクティブモードで、SLEEP命令を実行することによりスリープモードに遷移します。</p> <p>(3) スリープモードで、IRQ<sub>0</sub>端子に接続したスイッチ入力が入オンになるとIRQ<sub>0</sub>割込みが発生し、スリープモードは解除され、サブアクティブモードに遷移します。</p> <p>(4) サブアクティブモードでは、0.5secごとにタイマA割込み要求が発生し、タイマA割込み処理の中でLEDの制御を行ないます。LEDは0.5secごとに点灯、消灯を繰り返します。</p> <p>(5) サブアクティブモードで、IRQ<sub>1</sub>端子に接続したスイッチ入力が入オンになるとIRQ<sub>1</sub>割込みが発生し、IRQ<sub>1</sub>割込み終了後に、SLEEP命令を実行しアクティブモードに直接遷移します。</p> <p>(6) 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>(7) 図1にIRQ<sub>0</sub>、IRQ<sub>1</sub>端子に接続するスイッチの接続例を示します。</p> <div data-bbox="534 560 1061 907" style="text-align: center;"> <p>図1は、H8/3664のIRQ<sub>0</sub>とIRQ<sub>1</sub>端子の接続例を示しています。IRQ<sub>0</sub>端子にはプルアップ抵抗とスイッチSW1が接続されており、SW1は接地されています。IRQ<sub>1</sub>端子にもプルアップ抵抗とスイッチSW2が接続されており、SW2も接地されています。</p> </div> <p style="text-align: center;">図1 サブアクティブモードへの遷移におけるスイッチ接続例</p>	

使用機能説明	<p>(1) 本タスク例では、低消費電力モードのサブアクティブモードへの遷移を行ないます。図2にサブアクティブモードへのモード遷移図を示します。以下にサブアクティブモードの機能の説明を示します。</p> <ul style="list-style-type: none"> <li>スリープモードで割込み（タイマA、タイマV、タイマW、IRQ<sub>3</sub>～IRQ<sub>0</sub>、WKP<sub>5</sub>～WKP<sub>0</sub>、ウォッチドックタイマ、SCI3、IIC、A/D変換器）が発生したとき、SYSCR2のLSONが"1"ならば、サブアクティブモードに遷移します。また、サブスリープモードで割込み（タイマA、タイマW、IRQ<sub>3</sub>～IRQ<sub>0</sub>、WKP<sub>5</sub>～WKP<sub>0</sub>、ウォッチドックタイマ、IIC）が発生したとき、サブアクティブモードに遷移します。</li> <li>CCRのIビットが"1"の場合、また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、サブアクティブモードに遷移しません。</li> <li>サブアクティブモードの解除は、SLEEP命令またはRES端子入力により行われます。</li> <li>SLEEP命令による解除は、SYSCR1のSSBYが"X"、SYSCR2のDTONが"1"、SMSELが"X"、LSONが"0"の状態です。SLEEP命令を実行することによってアクティブモードへ直接遷移します。</li> <li>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。</li> <li>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"レベルを保持してください。</li> <li>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。</li> <li>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）</li> <li>CPUがプログラムを実行している動作モードにはアクティブモード、サブアクティブモードの2つのモードがあります。この2つの動作モードの間で、プログラムを停止する事なく遷移することを直接遷移と呼びます。直接遷移はSYSCR2のDTONを"1"にセットし、SLEEP命令を実行することにより可能です。アクティブモード、サブアクティブモードで動作周波数を変更する場合にも有効です。遷移後は直接遷移割込み例外処理を開始します。なお、割込みイネーブルレジスタ1により直接遷移割込みが禁止されている場合は、スリープモードまたはサブスリープモードへ遷移します。また、CCRのIビットを"1"の状態ですべての直接遷移を行なうとスリープモードまたはサブスリープモードに遷移し、遷移後のモードから割込みによる解除は不可能となります。</li> <li>サブアクティブモードからアクティブモードへの直接遷移は、サブアクティブモードでSYSCR1のSSBYを"X"、SYSCR2のLSONを"0"、SMSELを"X"、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に直接遷移します。</li> </ul> <div data-bbox="1061 1960 1476 2004" style="text-align: right; border: 1px solid black; padding: 2px;">             : Xは"1"、"0"どちらでも良い         </div>
--------	---

## 使用機能説明

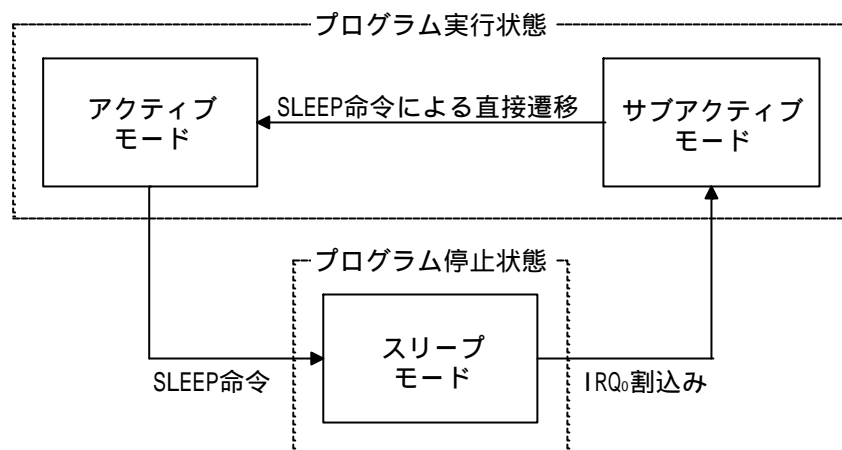


図2 サブアクティブモードへの遷移におけるモード遷移図

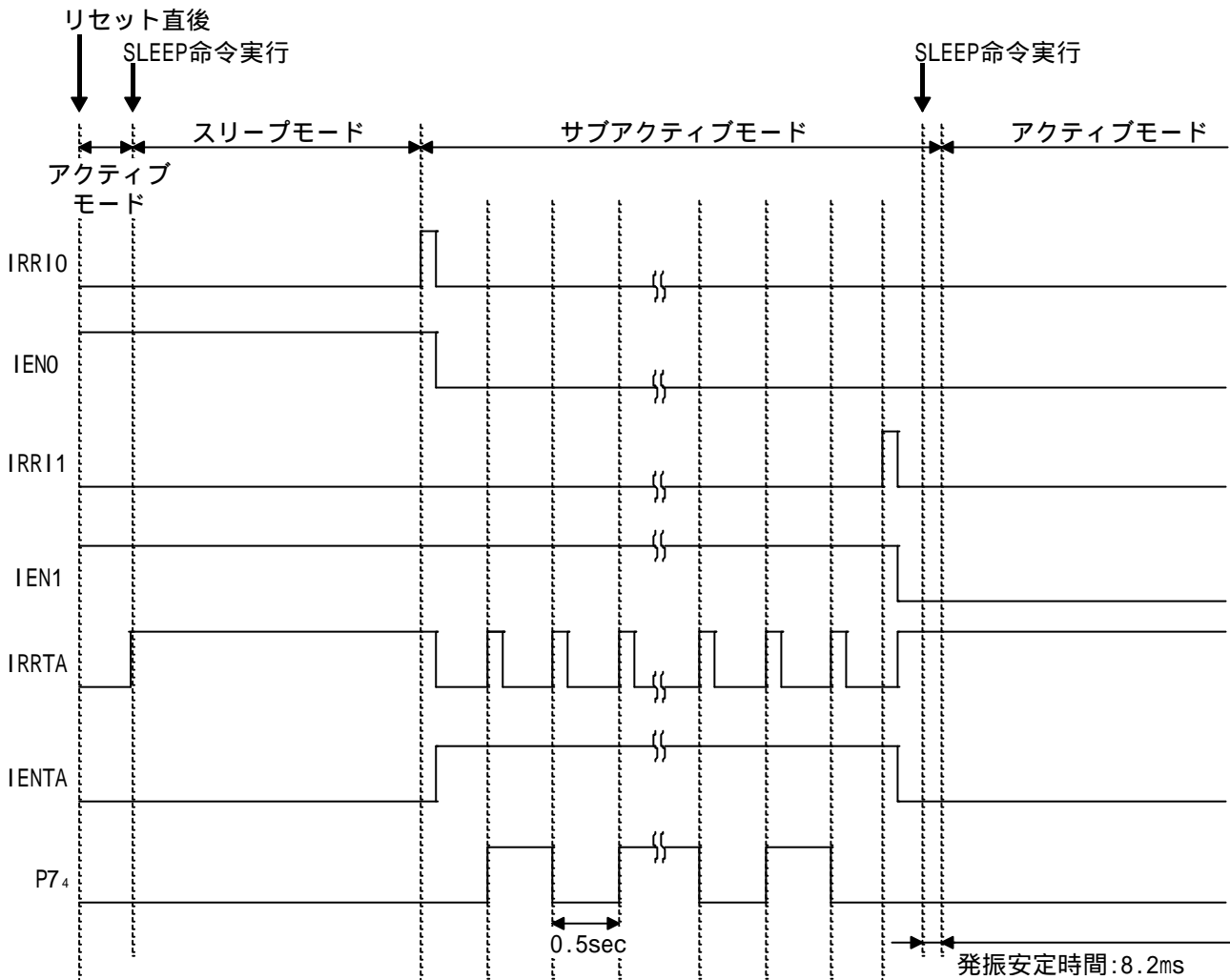
(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、サブアクティブモードへの遷移を行ないます。

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 <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>0</sub>	スイッチ1入力
IRR10	タイマA割込み要求の有無を反映
IEN10	タイマA割込み要求の許可、または禁止を設定
IRQ <sub>1</sub>	スイッチ2入力

動作原理

(1) 図3に動作原理を示します。図3に示すようにハードウェア処理、およびソフトウェア処理によりサブアクティブ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>0</sub> 割込みの設定
(b) IRQ <sub>0</sub> 割込み要求の許可
(c) IRQ <sub>1</sub> 割込みの設定
(d) IRQ <sub>1</sub> 割込み要求の禁止
(e) タイマAの設定
(f) タイマA割込み要求の禁止
(g) ポート7 <sub>4</sub> の設定
(h) 割込みの許可

ハードウェア処理
(a) IRR10を"1"にセット
(b) サブアクティブモードへ遷移
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) TCAをリセット
(e) タイマA割込み要求の許可
(f) IRQ <sub>1</sub> 割込み要求の許可

ハードウェア処理
(a) IRRTAを"1"にセット
ソフトウェア処理
(a) タイマA割込み処理の起動
(b) IRRTAを"0"にクリア
(c) IENTAを"1"にセットし、割込み要求を許可
(c) P7 <sub>4</sub> を反転し、LEDを点灯または消灯

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) タイマA割込み要求の禁止
(d) IRQ <sub>1</sub> 割込み要求の禁止

図3 サブアクティブ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>0</sub> 割込みの設定、IRQ <sub>1</sub> 割込みの設定、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、アクティブモードへの遷移を行なう
スイッチ1オン	IRQ0	IRQ <sub>0</sub> 割込み処理ルーチンで、IRQ <sub>0</sub> 割込み要求の禁止を行なう
スイッチ2オン	IRQ1	IRQ <sub>1</sub> 割込み処理ルーチンで、SWONFのセット、タイマA割込み要求の禁止、IRQ <sub>1</sub> 割込み要求の禁止を行なう
LED制御	taint	タイマA割込み処理ルーチンで、割込み要求の許可、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 <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	1
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="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"
	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、スリープ命令実行後、スリープモードを設定	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="1"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、サブアクティブモード (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、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 w/8に設定	H'FFF1 ビット1 ビット0	SA1="0" SA0="0"

## 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 : IRQ1_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) : IRQ0_SET="0"のとき、IRQ0端子を汎用入出力ポートに設定 : IRQ0_SET="1"のとき、IRQ0端子をIRQ0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) : IEG0="0"のとき、IRQ0端子入力の立ち下がりエッジを検出 : IEG0="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	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
	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	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
	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

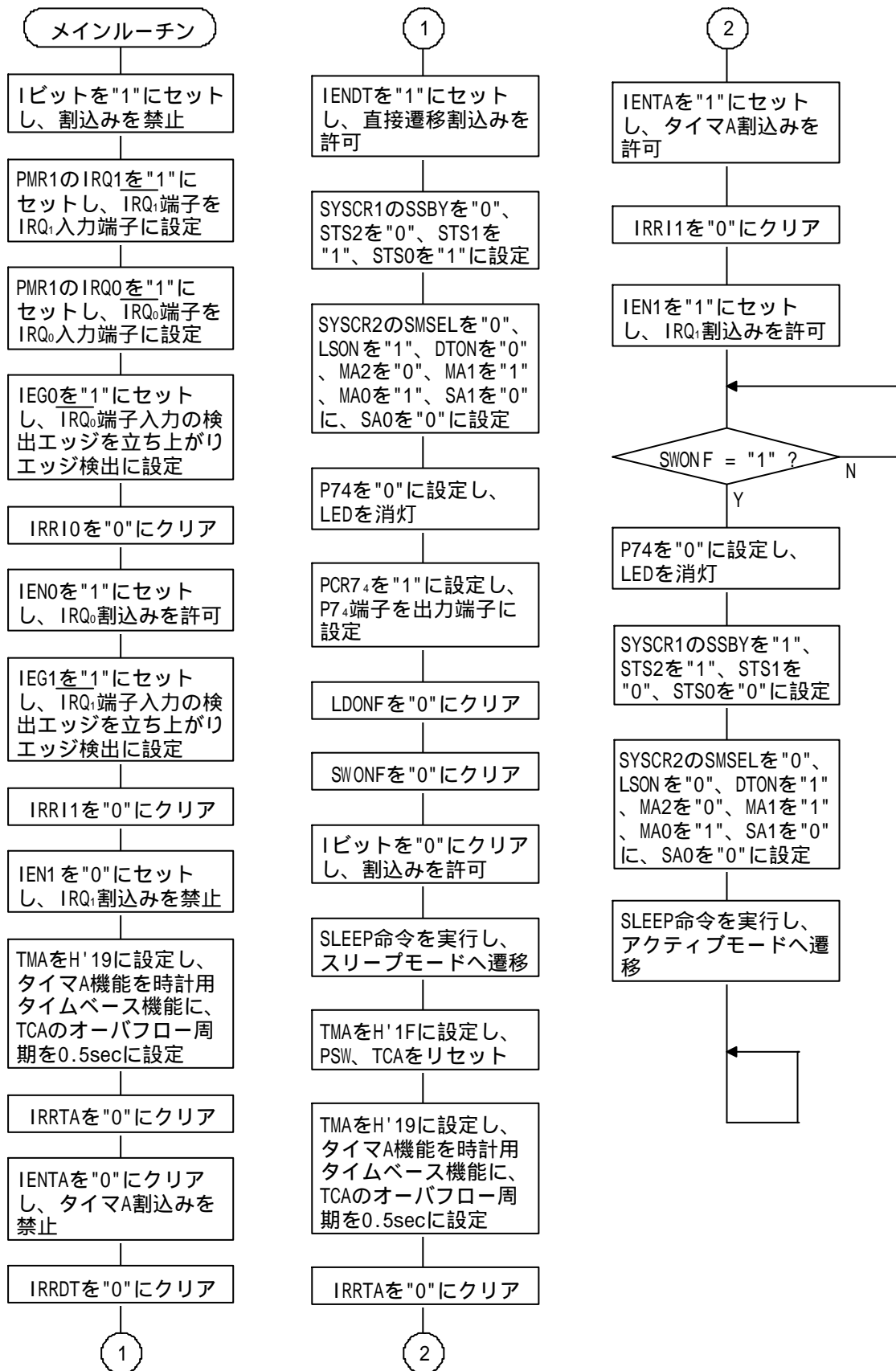
表4に本タスク例における使用RAM説明を示します。

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1	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0	LED制御

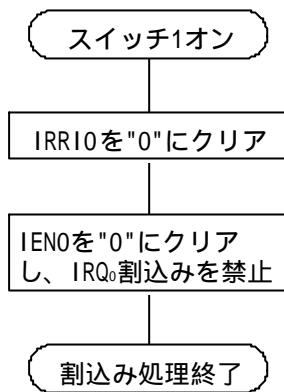
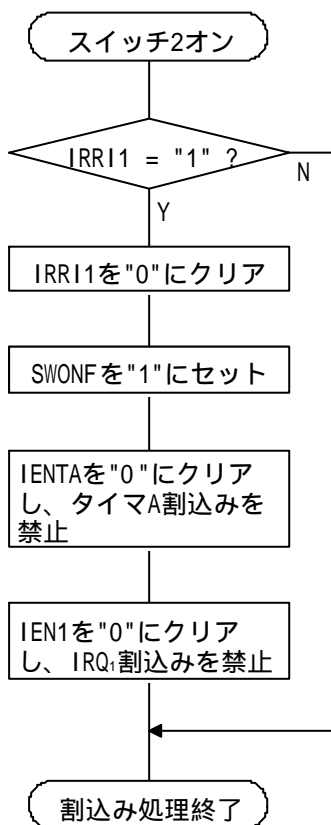
フローチャート

(a) メインルーチン

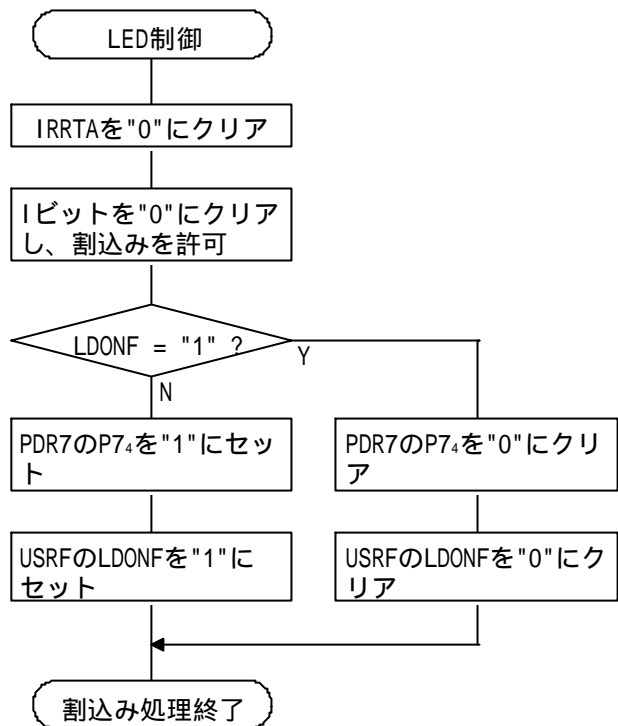


本例ではスタックポインタはINIT.SRC (アセンブリ言語) で設定してあります。

## フローチャート

(b) IRQ<sub>0</sub>割込み処理ルーチン(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



(e) 直接遷移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    #H'FF80,R7
LDC.B    #B'10000000,CCR
JMP      @_main
;
.END

/*****/
/* */
/* H8/300H Tiny Series -H8/3664- */
/* Application Note */
/* */
/* 'Transition to Subactive Mode' */
/* */
/* Function */
/* : Power-Down Mode */
/* Subactive 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 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 */
#define SA0    SYSCR2_BIT.b0 /* Subactive Mode Clock Select 0 */
#define IEGR1_BIT    (*(struct BIT *)0xFFF2) /* Interrupt Edge Select Register 1 */

```



サブアクティブモードへの遷移	使用機能	低消費電力モード：サブアクティブモード
プログラムリスト		
<pre> #define IEG1      IEGR1_BIT.b1          /* IRQ1 Edge Select          */ #define IEG0      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 IEN0      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 (dtint) #pragma interrupt (IRQ0) #pragma interrupt (IRQ1) #pragma interrupt (taint) /*****/ /* 関数定義 */ /*****/ extern void INIT ( void ); /* SP Set */ void main ( void ); void dtint ( void ); void IRQ0 ( void ); void IRQ1 ( void ); void taint ( void ); void wait ( void ); void sleep ( void );  /*****/ /* RAM define */ /*****/ unsigned char USRF; /* User Flag Area */  #define USRF_BIT (*(struct BIT *)&amp;USRF) #define SWONF     USRF_BIT.b1 /* Switch On Flag */ #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 /* Sleep Interrupt */ }; #pragma section V3 /* VECTOR SECTOIN SET */ void (*const VEC_TBL3[])(void) = {     IRQ0 /* IRQ0 Interrupt */ }; #pragma section V4 /* VECTOR SECTOIN SET */ void (*const VEC_TBL4[])(void) = {     IRQ1 /* IRQ1 Interrupt */ }; #pragma section V5 /* VECTOR SECTOIN SET */ void (*const VEC_TBL5[])(void) = {     taint /* timer A Interrupt */ }; #pragma section /* P */ /*****/ /* Main Program */ /*****/ void main ( void ) {     set_imask_ccr(1); /* Interrupt Disable */ </pre>		

## プログラムリスト

```

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

IEG0 = 1;              /* Set Rising Edge of IRQ0 Terminal Input */
IRR10 = 0;            /* Clear IRR10 */
IEN0 = 1;              /* IRQ0 Interrupt Enable */

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

TMA = 0x19;           /* Set TMA3 */

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

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

SYSCR1 = 0x30;        /* Initialize Function of Sleep Mode 1 */
SYSCR2 = 0x4C;        /* Initialize Function of Sleep Mode 2 */

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

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

set_imask_ccr(0);     /* Interrupt Enable */

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

TMA = 0x1F;           /* Reset PSW & TCA */
TMA = 0x19;           /* Initialize Timer A Function */

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

IRR11 = 0;            /* Clear IRR11 */
IEN1 = 1;              /* IRQ1 Interrupt Enable */

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

P74 = 0;              /* Turn off LED */

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

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

while(1) {
    ;
}

/*****
/*  IRQ0 Interrupt */
*****/
void IRQ0 ( void )
{
    IRR10 = 0;         /* Clear IRR10 */
    IEN0 = 0;          /* IRQ0 Interrupt Disable */
}

```

## プログラムリスト

```

/*****/
/*  IRQ1 Interrupt                               */
/*****/
void  IRQ1 ( void )
{
    if(IRR11 == 1){                               /* IRR11 = "1" ?           */
        IRR11 = 0;                               /* Clear IRR11            */
        SWONF = 1;                               /* Set SWONF              */
        IENTA = 0;                               /* Timer A Interrupt Disable */
        IEN1 = 0;                               /* IEN1 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              */
    }
}

/*****/
/*  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'001C
CV4	H'001E
CV5	H'0026
P	H'0100
B	H'FB80